

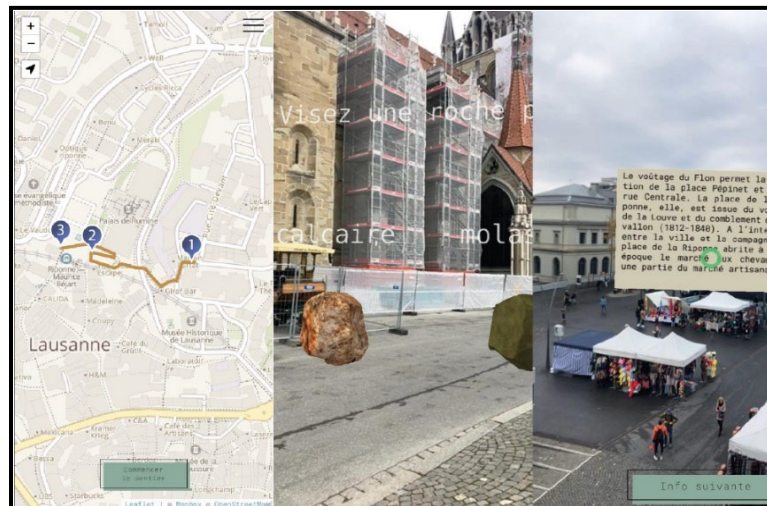
Master of Science in Geography

Implémentation, limites et perspectives d'une application web mobile en réalité augmentée

Adaptation du GéoGuide Lausanne pour une expérience touristique augmentée

Theo Gerritsen

Sous la direction du Prof. Christian Kaiser



Ce travail n'a pas été rédigé en vue d'une publication, d'une édition ou diffusion. Son format et tout ou partie de son contenu répondent donc à cet état de fait. Les contenus n'engagent pas l'Université de Lausanne. Ce travail n'en est pas moins soumis aux règles sur le droit d'auteur. À ce titre, les citations tirées du présent mémoire ne sont autorisées que dans la mesure où la source et le nom de l'auteur sont clairement cités. La loi fédérale sur le droit d'auteur est en outre applicable.

Implémentation, limites et perspectives d'une application web mobile en réalité augmentée

Adaptation du GéoGuide Lausanne pour une expérience touristique augmentée

Contenu du travail :

- Rédaction intégrale du travail de mémoire
- Application réalisée dans le cadre de ce travail (https://theogerritsen.github.io/AR_project/)
- Répertoire GitHub de l'application réalisée (https://github.com/theogerritsen/AR_project)

Source de l'image de la page de garde :
Capture d'écran de l'application réalisée

Résumé

La réalité augmentée est une technologie aujourd'hui largement utilisée et fait partie intégrante du quotidien de bon nombre de personnes. Avec plusieurs milliards d'utilisateurs de smartphones à travers le monde et l'évolution constante de la qualité de leurs capteurs intégrés, les téléphones mobiles sont devenus le médium idéal pour vivre des expériences en réalité augmentée. Croisé avec la possibilité de déployer une application en réalité augmentée directement sur le web, sans devoir passer par le téléchargement d'une application spécifique au préalable, ces deux facteurs ont permis l'émancipation rapide de la réalité augmentée durant cette dernière décennie. Cette étude dresse tout d'abord un cadre théorique concernant la réalité augmentée mobile sur le web. Elle propose des définitions, un historique et l'état de l'art de cette technologie. Une analyse est ensuite faite concernant le statut de la réalité augmentée dans les domaines du tourisme, du géotourisme et de l'éducation. Puis, dans une approche plus pratique, nous élucidons les méthodes d'implémentation d'une telle technologie, pour finalement nous concentrer sur le développement d'une application web mobile en réalité augmentée. Cette dernière reprend le GéoGuide Lausanne de l'Université de Lausanne, qui propose un sentier didactique à travers la ville de Lausanne, et présente trois de ses étapes en réalité augmentée. Nous explorons grâce à AR.js, une librairie JavaScript libre d'accès spécifiquement conçue pour la réalité augmentée sur le web, deux types de réalité augmentée, à savoir la réalité augmentée basée sur la localisation et la réalité augmentée basée sur la reconnaissance d'éléments naturels. Ce cadre pratique nous aidera par la suite à dresser les limites et perspectives de la réalité augmentée web mobile.

Abstract

Augmented reality has become a widely used technology for a variety of people and industries around the world. With several billion smartphone users across the world, mobile devices, thanks to the constant evolution of their components, are now the ideal medium to live augmented reality experiences. Furthermore, combined with the possibility to deploy such an experience directly on the web, without requiring any additional download of a native application, these two factors allowed a fast emancipation of augmented reality during the last decade. This study first establishes a theoretical background of mobile web augmented reality, giving definitions, a historical context and the state of the art of this technology. We further analyze the current status of augmented reality applied to tourism, geotourism and education. We then set up a more practical framework by giving the different possible implementation approaches for mobile web augmented reality applications, with their advantages and inconveniences. We finally develop our own augmented reality application, based on the Lausanne GeoGuide of the University of Lausanne, which consists of an educational trail through the city of Lausanne. Using AR.js, a JavaScript library specifically developed for augmented reality on the web, we explore two augmented reality solutions (location based AR and image tracking AR) that can be used for outdoor purposes. This practical framework will later help us to show the limits and future perspectives of mobile web augmented reality.

Table des matières

1. Introduction.....	7
1.1 Problématique	7
1.2 Questions et hypothèses de recherche.....	8
2. Définitions.....	10
2.1 Lexique	10
2.2 La réalité augmentée	10
2.3 La réalité augmentée sur téléphone mobile.....	11
2.4 La réalité augmentée mobile sur le web.....	11
3. État de l’art.....	13
3.1 La réalité augmentée	13
3.2 La réalité augmentée sur mobile	13
3.3 La réalité augmentée et le web.....	13
3.4 La réalité augmentée dans l’industrie	14
3.4.1 Le tourisme	14
3.4.2 L’éducation	15
3.5 Exemples d’application.....	15
Quelques sentiers didactiques interactifs en réalité augmentée en Suisse romande	15
4. Histoire de la réalité augmentée.....	16
5. La réalité augmentée dans l’industrie	19
5.1 La réalité augmentée et le tourisme	19
5.2 La réalité augmentée et le géotourisme.....	20
5.3 La réalité augmentée et l’éducation	20
6. La réalité augmentée mobile et le web.....	22
6.1 La réalité augmentée	22
6.2 Principes.....	22
6.2.1 Types de réalité augmentée.....	22
6.3 Algorithmes disponibles	23
6.3.1 Marker based augmented reality / Image tracking augmented reality	23
6.3.2 Performances des algorithmes	25
6.3.3 Comparaison des algorithmes	26
6.3.4 Discussion de la performance des algorithmes	27
6.4 Solutions d’implémentation de la réalité augmentée sur le web	28
6.4.1 Types d’implémentations.....	28
6.4.2 Comparaison des différentes solutions.....	30
6.4.3 Résultats et discussion des différences de coûts	30

7.	Méthodologie	33
7.1	Contexte	33
7.2	Environnements, librairies et API	33
7.2.1	Définitions.....	33
7.2.2	Ressources utilisées.....	34
7.3	Objectifs de l'application	36
7.3.1	Prototype	37
7.4	Langages et librairies utilisées	38
7.5	Architecture de l'application.....	40
7.6	Déploiement.....	41
8.	Résultats.....	42
8.1	Contenu de l'application	43
8.1.1	Écran d'accueil.....	43
8.1.2	Étape 1	45
8.1.3	Étape 2	46
8.1.4	Étape 3	47
8.1.5	Barre latérale de navigation	48
8.2	Interface utilisateur	48
9.	Discussion.....	49
9.1	Performances de l'application.....	49
9.2	La réalité augmentée web mobile dans le tourisme, le géotourisme et l'éducation.....	51
9.2.1	La réalité augmentée et l'éducation	51
9.2.2	La réalité augmentée appliquée au tourisme et au géotourisme.....	52
9.3	Limites et perspectives de la réalité augmentée mobile sur le web	52
9.3.1	Limites technologiques	52
9.3.2	Limites dans les domaines de l'éducation, du tourisme et du géotourisme	53
9.3.3	Perspectives d'avenir pour la réalité augmentée mobile sur le web	54
10.	Conclusion	56
11.	Références.....	57
11.1	Références en ligne	61

1. Introduction

La réalité augmentée est la possibilité de représenter des objets virtuels par-dessus le monde réel, par le biais d'un dispositif informatique physique. Il ne s'agit pas d'une technologie nouvelle, mais l'émancipation du téléphone mobile, plus particulièrement du smartphone, comme medium disposé à utiliser cette technologie a permis à la réalité augmentée de connaître un essor fulgurant durant cette dernière décennie. De plus, il existe aujourd'hui une volonté croissante de converger vers un développement des applications web plutôt que natives. La réalité augmentée web mobile a le potentiel de révolutionner la manière dont nous interagissons avec notre environnement. Elle est aujourd'hui largement utilisée dans diverses industries et fait déjà partie intégrante de nos vies. En vue des perspectives qu'offre la réalité augmentée, ainsi que l'évolution continue des téléphones mobiles et du Web, il ne fait aucun doute qu'elle prendra une place de plus en plus importante dans le quotidien des gens.

Cette étude explore en premier lieu les différentes facettes de la réalité augmentée. Partant d'un cadre général, nous resserrons peu à peu notre vision quant à cette technologie pour nous concentrer sur son utilisation directement sur un navigateur web, plus particulièrement sur téléphone mobile. Nous proposons ici un cadre théorique expliquant, de la manière la plus claire possible, ce qu'est la réalité augmentée mobile sur le web, d'où elle vient et ce qu'elle deviendra. Nous expliquons notamment ses diverses utilisations, ses solutions d'implémentation et les technologies utilisées pour son usage. Puis, dans une approche plus pratique, nous développerons notre propre application web mobile en réalité augmentée en utilisant une librairie JavaScript libre d'accès se nommant AR.js. Ce cadre pratique nous permettra de discuter des inconvénients, des avantages et des perspectives d'avenir d'une telle technologie.

1.1 Problématique

La réalité augmentée sur téléphone mobile a connu un essor fulgurant ces dernières années en lien avec son accessibilité facilitée pour les développeurs (outils de développement simplifiés) et pour les utilisateurs, grâce notamment aux progrès technologiques sans précédent des smartphones. L'évolution des caméras et des écrans à haute résolution, ainsi que l'amélioration des capteurs intégrés (le système de positionnement global (GPS), gyroscope, accéléromètre, etc...) rend son utilisation plus accessible, aisée et fluide (Paucher & Turk, 2010 ; Kamarainen et al., 2018). Cette technologie est aujourd'hui utilisée dans de nombreux domaines et industries tels que le tourisme, le géotourisme, l'éducation, le marketing, la manufacture et d'autre encore (Kanade & Prasad, 2021). Dans le monde académique, la réalité augmentée implémentée directement sur le web a également gagné une attention accrue durant la dernière décennie (Qiao et al., 2019a). Cette forme de réalité augmentée permet de dépasser les limites concernant les coûts et les outils de développement d'une application native et de faciliter la compatibilité entre les différents systèmes d'exploitation (iOS et Android) (Qiao et al., 2018). En revanche, elle souffre encore de quelques problèmes de performances, souvent liés aux algorithmes complexes qu'elle utilise (Karami et al, 2017), ainsi qu'à l'utilisation constante de l'ensemble des capteurs intégrés aux smartphones (Federov et al., 2016). Certaines solutions d'implémentation, qui transfèrent les calculs complexes sur le Cloud par exemple, adressent ce problème et améliorent grandement certaines performances de la réalité augmentée web mobile (Qiao et al., 2019b). Il ne fait aucun doute que l'évolution des algorithmes, l'amélioration des capteurs des téléphones mobiles, mais surtout l'utilisation de plus en plus fréquente du réseau 5G, plus rapide et fluide que ses prédécesseurs, donneront l'opportunité pour la réalité augmentée mobile sur le web de devenir une technologie plus attrayante, que ce soit pour les utilisateurs ou pour les fournisseurs.

Nous dressons ici un cadre théorique concernant la réalité augmentée, son historique, l'état de l'art actuel et ses diverses opportunités d'application. Nous nous intéresserons à l'utilisation de cette technologie dans les industries de l'éducation, du tourisme et dans la branche du géotourisme. Depuis plusieurs années et profitant de son attrait grandissant et son accessibilité facilitée, ces industries tirent

des avantages de la réalité augmentée en l'utilisant pour faciliter la visite d'un lieu ou pour rendre un apprentissage plus interactif et ludique. De nombreuses études montrent les bienfaits de l'utilisation de la réalité augmentée en lien avec le tourisme (Jung & tom Dieck, 2017) et l'éducation (Kamarainen et al., 2018). Nous comparons ensuite une partie des algorithmes disponibles et utilisés dans le cadre de la réalité augmentée afin de nous faire une meilleure idée quant à leur performance. Finalement, afin de conclure sur notre cadre théorique, nous montrons les différentes possibilités d'implémentation de la réalité augmentée mobile sur le web et les différences de performance qui leur sont liées. Trois solutions d'implémentation sont possibles (purement *front-end*, basé sur le Cloud et utilisant l'*edge computing* (Qiao et al., 2019a)) et présentent des avantages et des inconvénients selon le type de réalité augmentée utilisé. Nous tenterons de définir quelle est la meilleure approche pour l'étude réalisée ici.

Dans un second temps, nous rapporterons cet apprentissage dans un travail plus pratique en tentant de développer et de déployer une application web mobile en réalité augmentée (Mobile Web Augmented Reality, MWAR) dans le cadre d'un sentier didactique interactif, le GéoGuide Lausanne¹. L'application en question sera réalisée dans les langages informatiques de développement web typiques, à savoir HTML, CSS et JavaScript. Nous nous appuyons également sur des bibliothèques et des infrastructures logicielles (*framework*) JavaScript déjà existantes pour faciliter la mise en place de la réalité augmentée sur le web, à savoir AR.js et A-Frame. Le sentier didactique que nous tenterons de transformer en une expérience en réalité augmentée sera le GéoGuide Lausanne de l'Université de Lausanne qui traverse la ville de Lausanne en racontant des faits historiques, géomorphologiques et géologiques sur les lieux que visite l'utilisateur. Quelques étapes de ce sentier seront transformées en une expérience de réalité augmentée, rendant la visite plus interactive et impliquant au mieux l'utilisateur dans la découverte de la ville de Lausanne. Ce cadre pratique nous aidera à valider certaines hypothèses et à répondre à nos questions de recherche concernant la réalité augmentée mobile sur le web concernant ses performances, ses limites et perspectives d'avenir, ainsi que son utilité dans les domaines du tourisme, du géotourisme et de l'éducation.

1.2 Questions et hypothèses de recherche

Cette étude, ainsi que notre application, tentent de répondre aux questions de recherche suivantes :

1. Quels types de réalité augmentée peuvent être utilisées aujourd'hui ? Quels en sont leurs avantages et leurs inconvénients ?
2. Quelles sont les performances des algorithmes utilisés dans la réalité augmentée ?
3. Quels outils sont nécessaires afin de développer une application web mobile en réalité augmentée ?
4. De manière plus générale, quelles sont les performances de la réalité augmentée mobile sur le web ?
5. Quels sont les apports de la réalité augmentée dans des domaines purement informatifs, à savoir le tourisme, le géotourisme et l'éducation ?
6. Quelles sont les limites actuelles de la réalité augmentée sur le web ?
7. L'évolution actuelle du web présente-t-elle des perspectives d'avenir pour la réalité augmentée mobile sur le web ?
8. Les smartphones sont-ils aujourd'hui des mediums efficaces pour la visualisation de contenu en réalité augmentée ?

En nous penchant sur ces diverses questions, nous pouvons d'ores et déjà avancer quelques hypothèses que nous tenterons d'expliquer au fil de cette étude.

1. Il est possible, pour les domaines de l'éducation, du tourisme et du géotourisme, de tirer des bienfaits de la réalité augmentée web mobile, et ce pour diverses raisons. Selon Lin et al. (2013), la possibilité de visualiser, mais également de manipuler des phénomènes complexes par-dessus

¹ <https://igd.unil.ch/geoguidelsne/> consulté le 03.01.2023

le monde réel facilite leur compréhension. Un apprentissage ludique et interactif permet donc une acquisition des connaissances plus aisée. Ensuite, selon Jung & tom Dieck (2017), la possibilité pour l'utilisateur d'interagir avec son environnement crée des expériences uniques, plus mémorables et facilite grandement la visite de lieux inconnus, sans devoir passer par les connaissances d'un guide local.

2. Nous pouvons également émettre l'hypothèse que le smartphone représente aujourd'hui le medium idéal pour les expériences en réalité augmentée, par rapport à des mediums spécifiques à cette technologie. Ils sont aujourd'hui un objet indispensable dans la vie de la large majorité des personnes autour du globe. Ils ne sont sans doute pas aussi performants que les dispositifs spécifiques à la réalité augmentée, mais, selon Geiger et al. (2014), leur large émancipation augmente considérablement le nombre de personnes capables d'utiliser cette technologie. De plus, les dispositifs spécifiques à la réalité augmentée ayant perdus un vif intérêt ces dernières années et les smartphones connaissant une évolution de leurs performances fulgurante (Kanade & Prasad, 2021), ces derniers resteront, sans doute, le moyen privilégié de vivre des expériences en réalité augmentée durant les prochaines années à venir. Nous pouvons également avancer l'hypothèse qu'une application web, proposant un contenu utilisable directement dans un navigateur et supprimant ainsi la nécessité de télécharger une application supplémentaire, représente un argument de taille en terme d'attractivité pour les utilisateurs (Qiao et al., 2019b).
3. Finalement, nous pouvons penser que la réalité augmentée web mobile souffre encore aujourd'hui de problèmes de performances, notamment en lien avec les lourdes tâches computationnelles et les capacités encore limitées de certains composants intégrés aux smartphones, mais que certaines technologies permettront, dans un avenir proche, de produire des expériences toujours plus fluides. En effet, selon Qiao et al. (2019a), l'émergence progressive du réseau 5G, grâce aux performances qu'il offre, permettrait de surpasser les problèmes de latence en transférant certaines tâches computationnelles sur le Cloud, ou d'utiliser de manière plus efficace une technique appelée l'*edge computing*, qui cherche à effectuer l'ensemble des calculs au plus proche de l'utilisateur (soit directement sur le smartphone, soit sur une machine virtuelle à proximité de l'utilisateur). De plus, nous pouvons également émettre l'hypothèse que, dans certains cas, les faibles performances de la réalité augmentée sur le web peuvent être liées à l'efficacité des différents algorithmes utilisés. Il serait sans doute possible d'employer des algorithmes plus performants et moins lourds, grâce notamment à l'évolution rapide des algorithmes utilisant les réseaux de neurones (Morikawa et al., 2021).

2. Définitions

Dans cette section, nous allons définir certaines notions concernant la réalité augmentée (1), la réalité augmentée sur téléphone mobile (2) et la réalité augmentée mobile déployée sur un navigateur web (3). Chaque sujet sera ensuite traité plus en profondeur dans les chapitres suivants.

2.1 Lexique

Il est nécessaire, en premier lieu, de définir un lexique afin de mieux comprendre l'ensemble des termes en lien avec la réalité augmentée. En effet, la plupart des termes en lien avec cette technologie sont en anglais et difficilement traductible en français. Afin d'éviter d'utiliser des termes francophones à rallonge qui ne feraient qu'augmenter la complexité des thèmes abordés, nous avons choisi d'utiliser certains termes anglophones, plus explicites, tout au long de cette étude. La traduction la plus proche de chacun de ces termes est donc disponible dans ce lexique :

- AR : Augmented Reality, réalité augmentée.
- MAR : Mobile Augmented Reality, réalité augmentée utilisée grâce à un téléphone mobile.
- MWAR : Mobile Web Augmented Reality, réalité augmentée déployé sur le web, visualisée grâce à un téléphone mobile.
- Location based augmented reality: réalité augmentée basée sur la localisation, utilisant les capteurs intégrés dans un smartphone tels que le système GPS, l'accéléromètre, le gyroscope.
- Marker based augmented reality : réalité augmentée basée sur la reconnaissance d'un marqueur prédéfini. Le marqueur doit être unique, avec des géométries simples et des couleurs contrastées (par ex. un QR Code).
- Markerless augmented reality ou image tracking augmented reality: réalité augmentée qui utilise un algorithme permettant de reconnaître des éléments naturels et de les suivre, calculant constamment la position d'un élément par rapport à la caméra de l'utilisateur, et superposant ainsi du contenu en réalité augmentée par-dessus cet élément.
- Cloud : serveurs distants accessibles sur internet
- Cloud computing : méthode de calcul computationnel consistant à transférer les données sur le Cloud et effectuer les calculs liés à ces données directement sur le Cloud, réduisant ainsi la charge computationnelle des dispositifs informatiques.
- Edge computing : méthode de calcul computationnel visant à, d'une part, réduire la charge computationnelle des dispositifs informatiques, et d'autre part réduire de le temps de latence engendré par le transfert des données sur le Cloud, en effectuant cette tâche au plus proche de l'utilisateur (une définition complète est donnée au chapitre 6.4.2)

2.2 La réalité augmentée

D'une manière générale, la réalité augmentée peut être décrite comme étant une technologie qui combine des informations virtuelles au monde réel (Chen et al., 2019). Elle permet à l'utilisateur de visualiser ou d'intégrer des objets 3D virtuels sur le monde réel, en temps réel, par le biais d'une caméra intégrée aux appareils informatiques (smartphone, tablette ou autres dispositifs spécifique à l'AR) (Gatis, 2019) et vise à améliorer la compréhension que l'utilisateur se fait de l'environnement alentours. Elle correspond en quelque sorte au monde virtuel qui interagit avec le monde réel (Vyas & Bhatt, 2017).

Selon Azuma (1997), un système utilisant la réalité augmentée doit remplir trois critères afin d'être défini comme tel :

- Le système combine des éléments du monde réel et virtuel
- L'utilisateur peut interagir avec le système en temps réel
- Les éléments virtuels affichés interagissent et sont intégrés aux objets réels

Il existe aujourd'hui trois types de réalité augmentée qui peuvent être différenciés (Vyas & Bhatt, 2017) :

- 1) La réalité augmentée basée sur la géolocalisation d'éléments virtuels, utilisant le système GPS et les capteurs de mouvement intégrés aux dispositifs informatiques (*location based AR*)
- 2) La réalité augmentée basée sur la reconnaissance d'un marqueur prédéfini (*marker based AR*)
- 3) La réalité augmentée basée sur la reconnaissance d'éléments naturels environnants (*marker-less AR* ou *image tracking AR*)

Deux solutions d'implémentation sont possibles afin d'utiliser et de visualiser un contenu en réalité augmentée (Vyas & Bhatt, 2017):

- La réalité augmentée visualisée à travers un écran : elle utilise la caméra du dispositif informatique afin d'afficher à l'écran de l'utilisateur sa vue réelle ainsi que les éléments supplémentaires en réalité augmentée.
- La réalité augmentée visualisée à travers un dispositif informatique transparent, type lunettes : elle permet de visualiser des éléments en réalité augmentée en les superposant au monde réel sans devoir passer par l'utilisation d'une caméra.

2.3 La réalité augmentée sur téléphone mobile

Avec l'évolution sans précédent des smartphones durant ces deux dernières décennies (Kanade & Prasad, 2021), ces dispositifs sont devenus un moyen efficace et bon marché pour utiliser des applications en réalité augmentée. La réalité augmentée mobile (MAR, *Mobile Augmented Reality*) a rendu cette technologie accessible à tous, lui permettant et lui promettant un essor fulgurant. Elle prend l'avantage des différents capteurs directement intégrés aux smartphones (GPS, accéléromètre, gyroscope, caméra, etc...) (Federov et al., 2016). Selon Chatzopoulos et al. (2017), une application mobile peut être considérée comme une MAR si elle remplit les caractéristiques suivantes :

- Entrée (*input*) : l'application considère et utilise les différents types de capteurs du smartphone
- Traitement : l'application détermine le type d'information qui va être affiché à l'écran. Ces informations sont calculées et stockées soit localement, soit sur un serveur distant.
- Sortie (*output*) : l'application projette des informations supplémentaires sur l'écran du téléphone avec la vue actuelle de l'utilisateur. Elle augmente la réalité de l'utilisateur.

De manière similaire à la réalité augmentée et comme vu précédemment, une MAR peut être implémentée selon trois mécanismes distincts (Qiao et al., 2019) : (1) basé sur l'utilisation des capteurs du smartphone (*sensor based AR*) ; (2) basé sur la vision de l'utilisateur (*vision based AR*), ce mécanisme peut soit être *marker based AR*, soit être *marker-less AR*; (3) basé sur une utilisation hybride des deux mécanismes précédents.

2.4 La réalité augmentée mobile sur le web

Deux solutions peuvent être utilisées afin d'implémenter une application mobile en réalité augmentée (Qiao et al., 2019) : (1) développer une application native, (2) développer une application web.

- 1) Le développement d'une application native passe par l'utilisation de plateformes et d'outils dédiés (SDK, Software Development Kit) selon le système d'exploitation (Chen et al., 2019) :
 - a. ARKit pour iOS, lancé en 2017
 - b. ARCore pour Android, lancé en 2018
 - c. Vuforia, qui permet de développer une application en AR pour les différents systèmes d'exploitation

Bien que plus stable et plus robuste que la deuxième option, cette solution souffre en revanche d'un problème de compatibilité, puisqu'une application développée pour iOS, par exemple, ne sera pas utilisable en tant que telle sur un autre système d'exploitation. De plus, publier une application sur les plateformes de téléchargement respectives (AppStore & PlayStore) engendre d'importants coûts pour le développeur (Qiao et al., 2019).

- 2) Cette solution, appelée Mobile Web Augmented Reality (MWAR) permet un développement simple par le biais des langages de développement web typiques (HTML, CSS, JavaScript) et d'un éditeur de texte. L'hébergement d'une application web ne présente (presque) aucun coût monétaire. Elle ne requiert pas le téléchargement d'une application spécifique au préalable pour l'utilisateur. En revanche, certains facteurs tels que l'efficacité énergétique des smartphones, la connectivité au réseau internet ou encore la capacité de calcul des langages informatiques pour le développement web représentent encore des défis et des limites pour l'implémentation de telles applications (Chatzopoulos et al., 2017). L'émergence du réseau 5G et l'évolution continue et progressive des smartphones permet aujourd'hui d'implémenter ce genre de solution de manière efficace et stable.

3. État de l'art

3.1 La réalité augmentée

Azuma cadre, en 1997 déjà, les caractéristiques d'un système utilisant la réalité augmentée et les différents horizons pour une telle technologie (Azuma, 1997). Vyas & Bhatt (2017) divisent la réalité augmentée en trois modes de fonctionnement : la réalité augmentée basée sur un système GPS et les capteurs de mouvement (*location based AR*), la réalité augmentée basée sur la reconnaissance d'un marqueur prédéfini (*marker based AR*) et la réalité augmentée basée sur la reconnaissance d'éléments naturels (*markerless AR* ou *image tracking AR*). Gatis (2019) explique les différences principales, les avantages et les inconvénients de ces différents types de réalité augmentée. Billinghurst et al. (2015) dressent un historique de la réalité augmentée, partant du milieu des années 60, dans des cadres purement académiques ou gouvernementaux et allant jusqu'à 2015, où la réalité augmentée est largement utilisée et accessible à tous.

3.2 La réalité augmentée sur mobile

Le principe de la réalité augmentée sur téléphone mobile est de superposer des informations virtuelles complémentaires par-dessus le monde réel en utilisant les différents capteurs intégrés des smartphones (Qiao et al., 2019).

La volonté de transférer la réalité augmentée sur les téléphones mobiles est arrivée aussitôt que ces derniers étaient équipés d'une caméra, comme le montre l'étude de Mohring et al. (2004). Liarokapis et al. (2005) tentent de croiser cette technologie avec des données SIG sur téléphone mobile dès 2005. La MAR est aujourd'hui largement utilisée, résultat de plusieurs facteurs. L'émancipation des smartphones permet à des milliards de personnes autour du globe de posséder un medium permettant d'afficher du contenu en réalité augmentée (Billinghurst et al., 2015 ; Chatzopoulos et al., 2017). L'évolution rapide des capteurs intégrés tels que l'accéléromètre, le système GPS, le capteur de rotation, l'amélioration de la résolution de l'écran et de la caméra, a permis aux smartphones de rapidement s'imposer comme un medium bon marché, accessibles à tous pour l'utilisation quotidienne de la réalité augmentée (Geiger et al., 2014 ; Paucher Turk, 2010 ; Kanade & Prasad, 2021). Chatzopoulos et al. (2017) donnent une vue d'ensemble de la MAR en mettant en avant les domaines d'application, la mise en place d'une application MAR et les avantages et les limites d'une telle application. Bien que nos téléphones mobiles évoluent sans cesse avec du matériel informatique toujours plus puissant et une accessibilité et une fiabilité grandissante du réseau internet (Chatzopoulos et al., 2017 ; Geiger et al., 2014), les limites que constituent la puissance de calcul et les limites énergétiques d'un smartphone représentent encore de nos jours de nombreux défis dans l'élaboration d'une application mobile en réalité augmentée (Fedorov et al., 2016 ; Gatis, 2019). Ces applications sont très versatiles et peuvent tout aussi bien remplir un rôle fonctionnel que ludique. Pour n'en citer que deux, MapLens (Morrison et al., 2011) est une application MAR permettant de superposer du contenu géographique additionnel sur une carte papier traditionnelle. Pokémon GO² est un jeu en réalité augmentée qui utilise le système GPS et les différents capteurs d'un smartphone pour intégrer des monstres virtuels dans le monde réel.

3.3 La réalité augmentée et le web

Dans le monde académique, la réalité augmentée implémentée directement sur le web (MWAR, Mobile Web Augmented Reality) a également gagné de plus en plus d'attention durant la dernière décennie (Qiao et al., 2019a). Cette forme de réalité augmentée permet de dépasser les limites concernant les coûts et les outils de développement d'une application et de rendre la compatibilité entre systèmes d'exploitation (iOS et Android) plus aisée (Qiao et al., 2018). Plusieurs implémentations (*pure front-end*, *browser-kernel-based-extension*, *cloud computing*) sont possibles pour des applications web mobiles en réalité augmentée (Qiao et al., 2018), mais chacune pose ses propres limites. Ahn et al. (2014) proposent une marche à suivre afin de développer une application MWAR basée sur la

² <https://pokemongolive.com/>, consulté le 11.01.2023

reconnaissance d'éléments naturels en HTML, CSS et JavaScript. Qiao et al. (2018) mettent en avant les avantages de l'utilisation du *cloud computing* afin de réaliser de la réalité augmentée sur le web, transférant ainsi la charge de calcul du smartphone à un serveur externe afin d'optimiser le temps de calcul et l'efficacité énergétique. Renius (2019) teste les limites du WebXR Device API qui permet de développer et de visualiser efficacement des applications en réalité augmentée sur le web en prenant en charge les différents capteurs intégrés au matériel informatique utilisé. Une étude récente de Qiao et al. (2019a) montre que l'émergence progressive du réseau 5G offre des opportunités remarquables en ce qui concerne les applications MWAR. Son utilisation permettrait d'alterner entre une solution *pure front-end* et *cloud-based* afin d'optimiser l'application en termes d'images par seconde (FPS, frames per second), de latence et de consommation d'énergie. Ils mettent également en avant les possibilités qu'offre une solution utilisant l'*edge computing*, une technique qui pourrait s'avérer être la plus performante.

3.4 La réalité augmentée dans l'industrie

La popularité croissante de la réalité augmentée dans diverses industries est notamment due à son accessibilité et sa mise en place facilitée, de par les outils de développement mais aussi les avancées technologiques des smartphones. Les domaines du tourisme (Han et al., 2013 ; Yovcheva et al., 2013 ; Benyon et al., 2014 ; Jung et al., 2015 ; Jung & tom Dieck, 2017 ; Kumar et al., 2019 ; Cranmer et al., 2020), du géotourisme, (González-Delgado et al., 2020 ; Marino Alfonso et al., 2021), de l'éducation (Perry, 2015 ; Fedorov et al., 2016 ; Kumarainen et al., 2018 ; Yildiz, 2021) ou encore de la manufacture (Gavish et al., 2015 ; Pai et al., 2016) peuvent aisément mettre en place des applications en réalité augmentée qui leur ajouteront une plus-value notable.

3.4.1 Le tourisme

La réalité augmentée dans le tourisme permet de créer des expériences uniques et améliore grandement l'interaction de l'utilisateur avec son environnement (Jung & tom Dieck, 2017 ; InnovatAR, 2017). Elle ajoute une plus-value au monde environnant, laissant l'utilisateur interagir avec son environnement et explorer, de manière aisée, des lieux inconnus (Cranmer et al., 2020). Le développement de la réalité augmentée a transformé la manière dont les touristes peuvent vivre une destination, motivés par des expériences plus interactives, diversifiées et profondes (Jung et al., 2015), mais également la manière dont les gens planifient leur prochaine destination (InnovatAR, 2017). La visite de lieux devient plus dynamique et les touristes ne sont plus dépendants de guides locaux. La superposition d'informations supplémentaires, invisibles à l'œil nu, comme le contexte géologique ou des faits historiques, permet aux touristes une meilleure immersion dans leur lieu de visite. En revanche, Benyon et al. (2014) présentent les limites de la réalité augmentée pour le tourisme en mettant en avant la liberté de l'utilisateur de se déplacer sans devoir regarder sans cesse son écran.

3.4.1.1 Le géotourisme

Le géotourisme est une branche du tourisme qui se concentre sur la promotion d'éléments géologiques et de processus géomorphologiques qu'il est possible de trouver dans la nature (Ólafsdóttir, 2019). D'une manière générale, il consiste à visiter des lieux avec des paysages naturels particulièrement intéressants (Chylińska & Kołodziejczyk, 2018). Il a émergé au début des années 90, tout d'abord comme un champ du tourisme destiné aux personnes déjà informées quant aux géosciences. Au fil des années, cette facette du tourisme a suscité de plus en plus d'intérêt et s'adresse maintenant à un large public, allant du touriste lambda au spécialiste avisé (Gordon, 2018). Il représente aujourd'hui le domaine le plus fleurissant de l'industrie du tourisme. Le géotourisme propose la visite de sites comprenant des faits géologiques ou géomorphologiques d'intérêt particulier, appelé « géosites » (Dowling & Newsome, 2018). Il promeut, entre autres, la conservation des géosites en sensibilisant le public quant à son environnement, notamment en divulguant une interprétation quant à la géologie du site. De plus, le géotourisme contribue en général au développement durable de la région comprenant le site en question, notamment en tentant de protéger le paysage alentour via des panneaux informatifs, des points de vues spécifiques et des sentiers à respecter (Dowling & Newsome, 2018). Chylińska &

Kołodziejczyk (2018) soulignent l'intérêt du géotourisme en milieu urbain, qui permet la compréhension de processus géologiques ou géomorphologiques qui ont influencé la disposition et le développement d'une ville, notamment avec les matériaux de construction utilisés ou encore l'influence des formes héritées sur la planification du territoire. La réalité augmentée mobile devient aujourd'hui un outil intéressant pour le géotourisme puisqu'il permet à l'utilisateur de s'informer directement sur le terrain, de manière ludique et interactive (González-Delgado et al., 2020 ; Marino Alfonso et al., 2021 ; Martínez-Graña et al., 2022).

3.4.2 L'éducation

Plusieurs études montrent également les nombreux bénéfices qu'offre l'apprentissage à travers la réalité augmentée (Billinghurst et al., 2015 ; Kamarainen et al., 2018). En augmentant l'environnement alentours grâce à la superposition de plus amples informations, elle permet aux étudiants de comprendre des concepts abstraits qui requièrent une certaine interprétation des relations spatiales sur le terrain. Selon Lin et al. (2013), la modélisation d'objets en trois dimensions par-dessus le monde réel facilite la compréhension de certains phénomènes complexes par rapport à la lecture de textes descriptifs ou de simples graphiques en deux dimensions. L'immersion et l'interactivité qu'offre l'apprentissage par la réalité augmentée permettent également de mieux impliquer les étudiants, rendant l'apprentissage plus attrayant et ludique (Yen et al., 2013 ; Akçayir et al., 2016).

Kanade et Prasad (2021) donnent une vue d'ensemble des différents autres domaines dans lesquels la réalité augmentée peut être utilisée, comme le journalisme, la navigation, la localisation de points d'intérêts (restaurants, bars, cafés), les travaux sur le terrain (géologie, ingénierie), les parcs d'attraction, etc ...

3.5 Exemples d'application

Quelques sentiers didactiques interactifs en réalité augmentée en Suisse romande

- *Carcatchou ... ciel et espace*³ est une application web mobile en réalité augmentée recréant la voûte céleste au-dessus de Froideville (VD). Elle utilise la réalité augmentée basée sur la localisation de l'utilisateur. Ils mettent à disposition un QR Code, sur leur site, qui vérifie si le smartphone de l'utilisateur est compatible avec la réalité augmentée.
- *Totemi*⁴ est une application native pour smartphone qui regroupe de nombreux sentiers didactiques en réalité augmentée à travers des villes ou des régions de Suisse romande. Il suffit d'installer l'application via l'AppStore ou le PlayStore, et l'interface utilisateur permet de naviguer parmi les différents sentiers. Le type de réalité augmentée varie selon le sentier didactique. Afin de créer un sentier personnalisé à publier sur cette plateforme, il est nécessaire de contacter Totemi et leur faire pourvoir l'idée en question (scénario, public cible, etc...) et ils se chargeront de créer un sentier didactique interactif en réalité augmentée.
- *Novilé*⁵ est une application mobile native qui propose un tour guidé, agrémenté par une chasse au trésor, autour du Lac de Gruyère. Cette application utilise la plupart du temps la réalité augmentée basée sur la vision, en superposant des informations supplémentaires par-dessus des panneaux informatifs répartis autour du Lac de Gruyère. Un système de récompenses est également implémenté.

³ <https://carcatchou.cn-froideville.ch/> consulté le 01.02.2023

⁴ <https://totemi.ch/> consulté le 01.02.2023

⁵ <https://www.novile.ch/> consulté le 01.02.2023

4. Histoire de la réalité augmentée

Cette section décrit l'histoire de la réalité augmentée. Nous nous concentrerons ici sur les recherches majeures et sur l'évolution de la réalité augmentée mobile et web. Billinghurst et al. (2015) présentent dans leur papier une histoire exhaustive de la réalité augmentée.

Les premières expériences de réalité augmentée en utilisant des outils informatiques remontent à 1968, lorsque Ivan Sutherland, un ancien chercheur du MIT (Massachusetts Institution of Technology), développe à l'Université de Harvard le premier prototype de réalité augmentée avec l'aide de Bob Sproull (Billinghurst et al., 2015). Il s'agissait d'une pièce portable montée sur la tête de l'utilisateur qui calculait en temps réel les coordonnées de sa tête à l'intérieur d'un lieu clos. Un ordinateur transformait ces coordonnées pour afficher des informations virtuelles supplémentaires, sur des lentilles optiques, selon le point de vue de l'utilisateur (Sutherland, 1968). La plupart des recherches sur la réalité augmentée furent ensuite développées dans le cadre de recherches gouvernementales. En 1969, Tom Furness entame, pour la *United States Air Force*, des études concernant un projet nommé *Super-Cockpit*, qui cherchait à développer un outil intégré aux casques des pilotes afin d'afficher des informations de vols supplémentaires (Furness, 1986). Il se focalise alors déjà sur des éléments propres à la réalité augmentée, notamment l'interactivité. Le système permettait d'afficher du contenu stabilisé du monde réel afin d'aider le pilote dans certaines situations. Les années 80 furent consacrées, en majeure partie, au développement de la réalité virtuelle. Une compagnie américaine nommée VPL (Virtual Programming Languages) Research développe du matériel informatique comme le DataGlove, le EyePhone ou encore le DataSuit qui visait à contrôler un ordinateur avec des mouvements du corps (Billinghurst et al., 2015). Ces technologies débloquent de nouveaux horizons à la fin des années 80 et permirent à la réalité augmentée d'ouvrir ses portes aux mondes industriel et académique.

Divers groupes au sein d'universités différentes consacrent des recherches plus approfondies sur cette technologie. Au milieu des années 90, des chercheurs au CMU (Carnegie Mellon University) et au MIT commencent à proposer des instruments utilisant la réalité augmentée qui peuvent être portés toute une journée. En 1997, Steven Feiner développe, avec un groupe de chercheurs à l'Université de Columbia, un des premiers prototypes portables de réalité augmentée basé sur la localisation (Feiner et al., 1997). L'instrument, pesant 18 kg, était composé d'un ordinateur à écran tactile que l'utilisateur tenait dans sa main, ainsi qu'un deuxième ordinateur porté dans son dos, couplé de capteurs (accéléromètre, GPS) qui servait à calculer en temps réel la réalité augmentée à afficher. L'utilisateur était équipé de lunettes à travers lesquelles il voyait des informations complémentaires sur les bâtiments du campus de l'université. Ces informations complémentaires pouvaient être cliquées à l'aide de l'écran tactile portatif. À la même époque, une équipe du MIT se concentre sur l'implémentation d'un algorithme de reconnaissance d'images appliqué à la réalité augmentée (Starnes et al., 1997). Cet algorithme est capable de reconnaître les points clés d'un visage humain, enregistré à l'aide d'une caméra, et de les comparer avec 8000 autres visages stockés dans une base de données. Cela correspond aux premières tentatives d'une réalité augmentée portable en utilisant l'*image tracking*. La réalité augmentée basée sur la reconnaissance de marqueur commence également à être un standard. Billinghurst et Kato développent en 1999 ARToolKit, la première librairie Open Source de réalité augmentée, permettant ainsi de grandement faciliter l'implémentation de la réalité augmentée pour les chercheurs (Billinghurst et al., 2015).

En parallèle aux recherches de réalité augmentée portable, le milieu des années 90, avec l'introduction et l'émancipation des téléphones portables, montre un intérêt grandissant pour la réalité augmentée mobile. Les premières études utilisaient des caméras munies d'un écran qui étaient reliées à un ordinateur. Wagner & Schmalstieg proposent en 2003 un premier prototype de réalité augmentée portable autosuffisant (sans devoir passer par un ordinateur tiers) basée sur la librairie ARToolKit. Certaines tâches de calcul étaient transférées sur un serveur, ce qui montre également les premiers signes de fonctionnalités basées sur le Cloud pour la réalité augmentée. L'instrument était capable de

reconnaître des marqueurs prédéfinis ainsi que certains éléments naturels afin de superposer des éléments en réalité augmentée. 2004 marque le début de la réalité augmentée sur téléphone mobile. Mohring et al. (2004) développent un algorithme permettant de détecter un marqueur prédéfini afin d'afficher du contenu en réalité augmentée, en utilisant la caméra et l'écran d'un téléphone mobile (Fig. 1). L'algorithme est capable de détecter, sur deux lignes ou colonnes consécutives, un changement attendu de couleur jusqu'à déceler un marqueur prédéfini. Il est alors possible d'y superposer un objet virtuel en 3D.

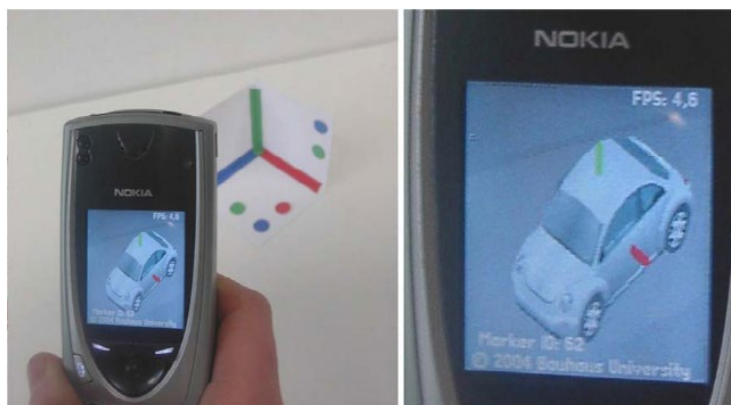


Figure 1, Réalité augmentée sur téléphone mobile, selon Mohring et al. (2004)

Henrysson & Ollila (2004) transposent en parallèle la librairie ARToolKit sur téléphone mobile. Bien que ces avancées impliquaient la possibilité d'utiliser la réalité augmentée pour tous les utilisateurs possédant un téléphone mobile, la première expérience en réalité augmentée réellement disponible au grand public fût *The Eye of Judgement*, un jeu vidéo de plateau sorti en 2007 sur la console PlayStation 3 de Sony (Billinghurst et al., 2015). L'intérêt pour la réalité augmentée ne cesse d'augmenter jusqu'en 2009 en lien avec le développement de la réalité augmentée sur smartphone et l'émergence de la réalité augmentée utilisant Flash. Adobe Flash Player⁶ était un module d'extension (*plugin*) de navigateur web développé par Adobe en 1996 qui permettait d'intégrer du contenu multimédia sur le web. Avec l'intégration du support des caméras par Flash, la réalité augmentée sur le web a connu un réel essor à partir de 2008. L'intégration du GPS et du compas par Android font du smartphone la plateforme idéale pour une expérience en réalité augmentée à l'extérieur. En 2009, les développeurs d'ARToolKit améliorent leur librairie afin de transposer la réalité augmentée au web (Billinghurst et al., 2015). Depuis lors, des plateformes de réalité augmentée sur le web, telles que Wikitude⁷ et Layar⁸, pour ne nommer que les plus connus, ont vu le jour. Les applications de réalité augmentée mobiles se multiplient d'année en année, remplissant un rôle ludique, éducationnel, publicitaire ou informatif (Arth et al., 2015). L'intérêt des développeurs grandit également grâce aux améliorations des performances des smartphones (Qiao et al., 2019a). La réalité augmentée sur le web se développe davantage, avec notamment des bibliothèques JavaScript permettant d'implémenter cette technologie rapidement et efficacement, telles que awe.media⁹ en 2016 et AR.js¹⁰ en 2017, qui utilise la librairie ARToolKit. Apple et Google sortent respectivement ARKit¹¹ pour iOS en 2017 et ARCore¹² pour Android en 2018, des SDK (Software Development Kits, kits de développement logiciel) en libre accès qui permettent de créer des applications natives en réalité augmentée sur les plateformes respectives (Oufkir et al., 2020). Limitées par les ressources qu'offre JavaScript et les performances des smartphones (connectivité,

⁶ <https://www.adobe.com/products/flashplayer/end-of-life.html> consulté le 03.01.2023

⁷ <https://www.wikitude.com/> consulté le 03.01.2023

⁸ <https://www.layar.com/> consulté le 03.01.2023

⁹ <https://awe.media/create> consulté le 03.01.2023

¹⁰ <https://ar-js-org.github.io/AR.js-Docs/> consulté le 03.01.2023

¹¹ <https://developer.apple.com/augmented-reality/> consulté le 03.01.2023

¹² <https://developers.google.com/ar> consulté le 03.01.2023

capteurs, caméras), les recherches concernant la réalité augmentée sur le web s'orientent aujourd'hui vers une réalité augmentée basée sur le Cloud (Qiao et al., 2018 ; Chen et al., 2019). L'émergence du réseau 5G et les opportunités qu'elle offre en terme de latence et de temps de calcul promet également une utilisation plus immersive de la réalité augmentée sur le web (Qiao et al., 2019a).

La réalité augmentée a donc connu un développement rapide et une attention particulière au fil des décennies. Les intérêts ont varié entre l'utilisation d'un matériel spécifique à la réalité augmentée et la réalité augmentée sur téléphone mobile. Avec l'amélioration constante des smartphones et l'implémentation facilitée de la réalité augmentée sur ceux-ci, cette branche de la réalité augmentée a connu un essor fulgurant et une attention accrue. De plus, avec le *cloud computing* et l'émergence du réseau 5G, la réalité augmentée sur le web est aujourd'hui de plus en plus performante, stable et facile à implémenter.

5. La réalité augmentée dans l'industrie

Le tourisme, le géotourisme, l'éducation, mais également la manufacture, la médecine ou encore le marketing présentent aujourd'hui plusieurs possibilités d'utiliser la réalité augmentée. De plus, avec l'amélioration grandissante des smartphones, ces derniers sont devenus le médium idéal pour des expériences en réalité augmentée sans coûts supplémentaires liés au matériel informatique. Nous allons, dans ce chapitre, discuter des plus-values, mais également des inconvénients, que peut amener la réalité augmentée dans les domaines du tourisme, du géotourisme et de l'éducation.

5.1 La réalité augmentée et le tourisme

L'attrait pour la réalité augmentée connaît sans cesse une croissance d'utilisation que ce soit pour les organisations qui les proposent ou pour les consommateurs (Cranmer et al., 2020). La réalité augmentée offre de nombreuses opportunités afin d'ajouter une plus-value à l'environnement, permettant ainsi au consommateur, ici le touriste, d'interagir avec les objets qui l'entourent et ainsi mieux comprendre son environnement. L'implémentation de contenu en réalité augmentée dans le domaine du tourisme a transformé la manière dont les touristes visitent, vivent une destination, en proposant des expériences toujours plus interactives, immersives, uniques et diversifiées.

Selon Jung & tom Dieck (2017) et Cranmer et al. (2020), l'efficacité d'une attraction touristique repose sur un concept nommé la co-création de valeur. Elle est définie comme « la création jointe de valeur ajoutée par la compagnie productrice et le consommateur » (Cranmer et al., 2020 p.3). En d'autres termes, afin de créer une attraction touristique efficace, la participation du consommateur est toute aussi importante que celle du producteur. Le consommateur joue alors un rôle actif crucial dans le processus d'innovation d'un produit touristique. Cette participation permet d'assurer le fait que la valeur ajoutée est développée selon le point de vue des utilisateurs, qui seront le public cible du produit (Jung & tom Dieck, 2017). Selon des études, ce processus crée des expériences « plus mémorables et riches » (Jung & tom Dieck, 2017 p.4). Ceci peut être réalisé en récupérant les données de l'utilisateur, ou en faisant participer l'utilisateur à un questionnaire de satisfaction quant à l'innovation en question. Selon ce concept de co-création de valeur, une visite guidée d'un lieu en réalité augmentée permet une meilleure implication du consommateur. Ce dernier est capable d'interagir directement avec son environnement, d'explorer des lieux inconnus, sans passer par une personne tierce. Les producteurs peuvent extraire des données sur l'utilisation de leur produit ainsi que le comportement des utilisateurs et ainsi développer leur produit davantage, en prenant en compte le point de vue du consommateur.

Jung & tom Dieck (2017) avancent également l'idée que l'implémentation de la réalité augmentée dans le domaine du tourisme permet de séparer une visite en trois phases : avant, pendant et après la visite du consommateur. Puisque ce dernier ne se trouve pas réellement sur les lieux durant la première phase, nous parleront plutôt de réalité virtuelle (VR) ou réalité mixte (XR). L'utilisation de la réalité mixte avant la visite d'un lieu permet de supprimer les a priori et d'obtenir une meilleure attention lors de la visite réelle puisque l'utilisateur possède déjà quelques informations essentielles quant à la compréhension du lieu. L'utilisation de la réalité augmentée durant une visite permet de proposer un nombre important d'informations à l'utilisateur en les superposant aux éléments du monde réel. En laissant l'opportunité à l'utilisateur d'interagir avec son environnement, il est alors possible de lui faire comprendre des processus spatiaux complexes qui nécessiteraient de longues descriptions durant une visite traditionnelle (Kamarainen et al., 2018). Il est toutefois important de souligner qu'un équilibre doit être réalisé entre le contenu proposé en réalité augmentée et le monde réel. En effet, selon Benyon et al. (2014) qui mettent en avant le concept de « disparition de la médiation », un danger de cette utilisation accrue de la réalité augmentée pour visiter des lieux touristiques est la perte d'attention pour les objets réels de l'environnement. La réalité augmentée participerait à une diminution du vécu authentique d'un lieu en éloignant la présence de l'utilisateur de l'endroit où il se trouve réellement. Il est donc essentiel de trouver un équilibre entre le contenu virtuel affiché à l'écran et le monde réel.

5.2 La réalité augmentée et le géotourisme

Le domaine du géotourisme, bien qu'existant depuis plus de trente ans, connaît depuis peu une effervescence particulière en lien avec la réduction des distances grâce aux vols aériens internationaux meilleurs marché (Ólafsdóttir, 2019). Le géotourisme est aujourd'hui la branche la plus fleurissante de l'industrie du tourisme et invite toujours plus de touristes à visiter des lieux montrant des paysages d'intérêts particuliers. Cette augmentation soudaine de visiteurs demande une attention particulière en ce qui concerne la conservation des géosites et requiert de sensibiliser le public concerné quant aux problématiques environnementales de la région. Il est également nécessaire d'installer des infrastructures spécifiques à la protection de la nature (aires d'observation, barrières, panneaux directionnels, etc.) (Dowling & Newsome, 2018). L'étude portée ici adresse plus particulièrement le géotourisme en milieu urbain. Cette forme de géotourisme concerne la visite de lieux urbanisés recelant des processus géologiques et géomorphologiques intéressants, cachés aujourd'hui sous les infrastructures urbaines (Tscheu & Buhalis, 2016 ; Chylińska & Kołodziejczyk, 2018). L'objectif de cette branche du géotourisme est de proposer une visite montrant les paysages hérités qui ont façonné la morphologie et le développement d'une ville. Les matériaux de construction locaux, utilisés la plupart du temps dans les anciennes constructions, permettent également de retracer l'histoire d'une ville (Chylińska & Kołodziejczyk, 2018). La réalité augmentée connaît un attrait grandissant dans le géotourisme, notamment en lien avec son accessibilité facilitée mais également avec les avantages qu'elle offre, que ce soit pour l'utilisateur, le fournisseur ou la nature (Tscheu & Buhalis, 2016). Premièrement, la réalité augmentée appliquée au géotourisme permet à l'utilisateur, ici les touristes, de vivre des expériences plus immersives et mémorables (Tscheu & Buhalis, 2016). La possibilité pour l'utilisateur d'interagir avec son environnement facilite la compréhension de processus géologiques et morphologiques complexes (Jung & tom Dieck, 2017). La réalité augmentée offre également l'opportunité de révéler des informations invisibles à l'œil nu. Appliquée à une géovisite géologique, par exemple, il est possible de superposer les différentes couches géologiques par-dessus le paysage réel (Marino Alfonso et al., 2021). La réalité augmentée en tant que produit géotouristique propose également des avantages conséquents pour les fournisseurs. Les touristes voulant vivre des expériences de plus en plus mémorables et uniques, cette technologie représente un bon moyen, pour les destinations, de se différencier d'autres attractions touristiques (Tscheu & Buhalis, 2016). Elle limite également les infrastructures nécessaires lors de l'élaboration de tours géotouristiques. Il n'est alors plus nécessaire de passer par l'installation de matériel spécifique au géotourisme (panneaux informatifs, directionnels, etc...), puisque seul le téléphone portable de l'utilisateur est requis afin de vivre l'expérience. En plus de réduire les coûts de mise en place, cela permet aussi de limiter la modification d'un paysage naturel (dans un milieu non urbain) avec des infrastructures supplémentaires.

5.3 La réalité augmentée et l'éducation

La réalité augmentée appliquée au domaine de l'éducation n'est pas une chose nouvelle, notamment avec les disciplines liées à la médecine qui ont très vite, dès les années 1990, vu les opportunités d'apprentissage qu'une telle technologie représentait (Billinghurst et al., 2015). De manière similaire à l'industrie du tourisme, l'émancipation et l'évolution des smartphones a permis une utilisation plus spontanée de la réalité augmentée dans l'éducation, sans devoir passer par du matériel informatique coûteux et encombrant (Kamarainen et al., 2018). La réalité augmentée offre de nombreuses opportunités d'apprentissage en impliquant davantage l'étudiant dans l'acquisition de connaissances nouvelles et en proposant un contenu interactif. Selon certaines études, l'utilisation de la réalité augmentée dans le processus d'apprentissage permet aux étudiants de mieux comprendre des phénomènes spatiaux complexes, abstraits, invisibles à l'œil nu ou encore inimaginables à l'échelle humaine (Wu et al., 2013 ; Kamarainen et al., 2018). En effet, représenter virtuellement ces phénomènes, tout en permettant à l'étudiant d'interagir avec ceux-ci, participe grandement à une meilleure compréhension et à un apprentissage facilité. Dans les disciplines des sciences naturelles, comme l'écologie ou la géologie, il est possible d'afficher, en temps réel et superposés au monde réel, des biomes ou des affleurements rocheux invisibles à l'œil nu, cachés, par exemple, par une végétation

dense (Kamarainen et al., 2018). Dans le cadre de disciplines plus abstraites, comme la physique ou la chimie, la réalité augmentée permet de simuler des phénomènes complexes non reproductibles dans le monde réel, comme la rotation de la terre ou la réaction de procédés chimiques dangereux (Majeed & Ali, 2020). De plus, la manipulation et l'interaction avec des phénomènes intangibles facilite la compréhension de leur fonctionnement. D'une manière générale, les recherches montrent une progression importante pour les étudiants utilisant la réalité augmentée dans leur divers apprentissages (Kamarainen et al., 2018).

Cependant, une utilisation inadéquat de la réalité augmentée dans le cadre de l'apprentissage entrerait grandement son efficacité. Majeed & Ali (2020) séparent l'utilisation de la réalité augmentée dans le domaine de l'éducation en cinq catégories :

1. L'apprentissage basé sur la découverte : utilisé en histoire ou en astronomie, du contenu virtuel additionnel est affiché à l'écran selon l'endroit visité. Ce mode d'apprentissage permet de superposer des informations complémentaires concernant un objet ou un lieu et d'ainsi rendre l'apprentissage de certains sujets plus excitant et dynamique.
2. La modélisation d'objet : il s'agit de modéliser un objet en trois dimensions afin de permettre à l'étudiant de le manipuler et de l'observer sous différents points de vue. Cette méthode est souvent utilisée en architecture ou en anatomie humaine.
3. Les livres en réalité augmentée (Fig. 2): certaines applications permettent d'augmenter le contenu d'un livre afin de le rendre interactif. Cette méthode est notamment bénéfique aux étudiants ayant plus de peine à acquérir des connaissances uniquement sur la base de textes (Billinghurst et al., 2015).
4. L'apprentissage basé sur les jeux : l'utilisation de jeux vidéo est devenue, au fil des années, un outil d'apprentissage puissant, permettant de grandement simplifier des concepts et phénomènes complexes (Diegmann et al., 2015). La réalité augmentée peut ainsi être utilisée pour la création d'un jeu vidéo se déroulant dans le monde réel, rendant l'apprentissage plus dynamique et ludique.
5. L'entraînement des compétences : cette méthode consiste simplement à utiliser la réalité augmentée afin de suivre, par exemples, différentes étapes menant à la maintenance d'un certain objet. Elle est souvent utilisée dans les apprentissages liées aux industries de manufacture, en montrant les différents stades de réparation ou de mise en place d'un produit spécifique.



Figure 2, Livre augmenté, selon Billinghurst et al. (2015)

La réalité augmentée offre donc de nombreuses opportunités en termes d'apprentissage, pour peu qu'elle soit utilisée de manière adaptée. Selon Diegmann et al. (2015), elle offre une gamme d'apprentissage unique et diversifiée, comme l'apprentissage collaboratif ou l'apprentissage centré sur l'étudiant, qui consiste à le rendre plus indépendant vis-à-vis de l'acquisition de nouvelles connaissances, ne reposant plus entièrement sur le partage du savoir d'un professeur. Il faut en revanche savoir équilibrer la quantité de contenus proposée en réalité augmentée afin de ne pas surcharger les étudiants, au risque de complexifier certains apprentissages (Wu et al., 2013).

6. La réalité augmentée mobile et le web

6.1 La réalité augmentée

Avec l'émancipation de la réalité augmentée durant cette dernière décennie, la réalité augmentée mobile sur le web (MWAR, Mobile Web Augmented Reality) a connu un intérêt croissant en lien avec sa facilité d'implémentation pour les développeurs et son accessibilité pour les utilisateurs. Contrairement à une application native, une application web ne requiert pas d'installation au préalable. Développer une application web permet d'unifier le développement entre plateformes et supprime les soucis de compatibilité entre systèmes d'exploitation (Qiao et al., 2019b). Plusieurs manières d'implémenter une MWAR sont possibles : une solution purement *front-end*, une solution basée sur le Cloud et une solution qui requiert de modifier le noyau (*kernel*) du navigateur.

Nous allons, dans cette section, discuter des différents systèmes et caractéristiques d'implémentation d'une application web mobile en réalité augmentée en mettant également en avant les limites qu'elles représentent et les perspectives pour l'avenir.

6.2 Principes

6.2.1 Types de réalité augmentée

Ce chapitre élucide les différents types de réalité augmentée qu'il est possible d'implémenter, répondant ainsi à notre première question de recherche. De manière similaire à la réalité augmentée et comme vu précédemment, une MWAR peut être implémentée selon trois mécanismes distincts (Qiao et al., 2019a):

- 1) Basée sur l'utilisation des capteurs du smartphone (*location based AR* ou *sensor based AR*) : l'application utilise les capteurs intégrés aux smartphones (accéléromètre, gyroscope, compas, magnétomètre, GPS) afin de calculer la position et les mouvements de l'utilisateur et de placer sur le monde réel des éléments en réalité augmentée. Cette solution ne demande que très peu de ressources au niveau des calculs du smartphone, mais est largement dépendante de la précision des capteurs intégrés, ce qui a pour conséquence des objets souvent faussement géolocalisés (Gatis, 2019).
- 2) Basée sur la vision de l'utilisateur (*vision based AR*) : l'application capture des éléments de l'environnement et superpose du contenu en réalité augmentée. Ce type de réalité augmentée peut être soit basé sur la reconnaissance d'un marqueur contrasté prédéfini (*marker based AR*) soit sur la reconnaissance d'éléments naturels préalablement établis (*marker less AR / image tracking AR*) à l'aide d'algorithmes type NFT (Natural Feature Tracking). Contrairement à la solution précédente, elle permet une expérience souvent très stable et sans imprécisions, mais, dans le cas de l'*image tracking AR*, elle est extrêmement énergivore en raison des algorithmes compliqués qu'elle nécessite (Gatis, 2019).
- 3) Basée sur une version hybride des deux méthodes mentionnées ci-dessus.

6.2.1.1 Location based augmented reality

Cette méthode repose sur le placement de points d'intérêt selon des coordonnées géographiques. Elle utilise les capteurs intégrés au smartphone afin de définir le comportement de l'utilisateur. Plusieurs capteurs sont utilisés (Paucher & Turk, 2010) :

- Le système de positionnement global (GPS) du smartphone pour une application en extérieur, ou une reconnaissance des éléments naturels à travers un algorithme NFT (Natural Feature Tracking), expliqué dans la section 7.3 pour une application intérieure
- L'accéléromètre
- Le gyroscope
- Le magnétomètre

En premier lieu, pour une application extérieure, des points d'intérêt avec des coordonnées géographiques précises sont stockés dans une base de données. La position de l'utilisateur est extraite

en utilisant les coordonnées GPS acquises par le biais du smartphone. Les points d'intérêt peuvent alors être affichés sur l'écran de l'utilisateur. Afin de répondre aux différents mouvements de l'utilisateur lors de l'utilisation de l'application, l'accéléromètre, le gyroscope et le magnétomètre sont utilisés.

L'accéléromètre du smartphone donne trois vecteurs principaux (x, y, z) qui représentent l'accélération de l'utilisateur par rapport à ces trois axes. Le gyroscope est utilisé pour déterminer l'inclinaison du smartphone selon trois angles, qui représentent la quantité de rotation autour des trois axes du smartphone. Finalement, le magnétomètre est utilisé afin de déterminer la rotation de l'utilisateur lorsqu'il est en mouvement. Ce capteur n'utilise qu'un paramètre, qui est l'angle de rotation par rapport à un vecteur indiquant le Nord magnétique (Geiger et al., 2014).

En combinant les paramètres de ces quatre capteurs, il est possible d'afficher des objets virtuels en réalité augmentée sur l'écran de l'utilisateur et de répondre à ses moindres mouvements. L'implémentation de ce type de réalité augmentée est aisée et rapide, mais l'utilisation de ces capteurs et le calcul constant du comportement de l'utilisateur reste relativement énergivore (Qiao et al., 2019a).

6.2.1.2 Marker based augmented reality

Cette méthode repose sur la reconnaissance d'un marqueur (en général binaire, en noir et blanc) prédéfini (Vyas & Bhatt, 2017). Il existe donc deux entrées : la scène en temps réel que capture la caméra, et le marqueur prédéfini. Afin d'accélérer le processus et de mieux détecter le marqueur, l'image acquise par la caméra est transformée en ton de gris. Depuis cette image, l'algorithme détecte les arrêtes du marqueur et les délimitations externes des contours. Le marqueur est ensuite divisé en une grille plus fine. Chaque valeur de cellule (0 ou 1) est comparée au marqueur prédéfini. Les coordonnées du marqueur sont ensuite calculées, relativement à la position et au point de vue de la caméra.

6.2.1.3 Image tracking augmented reality

La méthode de l'image tracking AR est la plus lourde au niveau computationnel. Plusieurs approches sont possibles afin d'implémenter un algorithme capable de reconnaître des éléments naturels dans une scène donnée. Parmi les plus utilisés nous pouvons nommer la méthode SIFT (Scale Invariant Feature Transform, (Lowe, 2004)) et la méthode de Ferns (Wagner et al., 2008). D'autres algorithmes sont aujourd'hui disponibles et plus performants dans certains cas. Wagner et al. (2010) proposent une alternative à ces deux algorithmes, spécifiquement développés pour la détection d'image sur smartphone. Nous présentons ici la méthode SIFT, utilisée dans le cadre de notre étude. Nous comparons l'efficacité des différents algorithmes disponibles dans le chapitre suivant.

6.3 Algorithmes disponibles

6.3.1 Marker based augmented reality / Image tracking augmented reality

Un algorithme de détection d'image est divisé en quatre étapes distinctes (Wagner et al., 2010):

- 1) La détection : l'algorithme identifie les points clés
- 2) La description : la région autour de chaque point clé est décrite. Idéalement, cette région est décrite de manière à ce qu'elle soit invariante aux changements de luminosité, d'échelle et de rotation.
- 3) La correspondance : les descripteurs sont comparés à travers les images afin d'identifier des éléments similaires entre images.
- 4) Le rejet des données aberrantes (*outliers*) : les éléments clés qui n'ont pas trouvé de correspondance sont omis.

En ce qui concerne la réalité augmentée, une cinquième étape consiste à superposer un objet virtuel sur l'image lorsque les points clés ont été détectés.

De nombreux algorithmes pour la détection d'images sont aujourd'hui disponibles et applicables dans le cadre de la réalité augmentée (Ou et al., 2020). Outre SIFT et Ferns, nous pouvons également nommer

SURF (Speed Up Robust Features) et ORB (Oriented Fast and Rotated Binary Robust Independent Elementary Features). Wagner et al. (2010) proposent une méthode adaptée afin de rendre les approches SIFT et Ferns plus accessibles et efficaces sur téléphone mobile, nommés PhonySIFT et PhonyFern. Aujourd'hui, des algorithmes bien plus performants, basés sur des techniques de détection utilisant des réseaux de neurones, ont été développés (Morikawa et al., 2021). Ces algorithmes n'étant pas encore implémentés dans les outils de développement destinés à la réalité augmentée, nous ne les traiteront pas ici.

Dans cette section, nous nous concentrerons sur l'algorithme SIFT, qui est intégré dans ARToolKit, la librairie qu'utilise AR.js, employée dans notre projet. Il s'agit d'un algorithme relativement simple à implémenter et robuste, développé en 2004 par Lowe, à l'Université de Columbia (Tyagi, 2020). Nous nous pencherons ensuite sur d'autres algorithmes, également utilisés dans le cadre de la réalité augmentée, afin de comparer leurs performances et leur utilité.

6.3.1.1 SIFT (Scale Invariant Feature Transform)

SIFT, l'algorithme le plus renommé (Tareen & Saleem, 2018), a été présenté en 2004 par Lowe. Il s'agit d'un algorithme de reconnaissance d'image qui a la particularité d'être performant quelle que soit la rotation et l'échelle de l'image observée (Fig. 3). La méthode SIFT peut être séparée en cinq étapes principales (Tyagi, 2020) :

1. Sélection des pics dans une échelle spatiale : la première étape consiste à détecter des points clés qui sont reconnaissables à plusieurs échelles. Cela se fait en prenant une image de base et en y appliquant un flou Gaussien progressif. Cette image est donc dupliquée avec un flou de plus en plus important. Ces images floutées sont ensuite utilisées pour générer un nouvel ensemble d'images nommées Différence de Gaussiens (DoG). Une image DoG est générée en soustrayant deux images possédant un flou gaussien différent. Il est alors possible de trouver les points clés qui sont invariants quelle que soit l'échelle. Un pixel d'une image est ensuite comparé à ses huit pixels voisins, ainsi qu'aux neuf pixels de l'image de l'échelle d'en dessus et d'en dessous. De cette manière, un total de vingt-six vérifications sont faites. Si le pixel vérifié représente un extrema local, il est alors un candidat pour un point clé. Cela veut dire que ce point clé est le mieux représenté à cette échelle. Cette étape se nomme Laplacien de Gaussien (LoG).
2. Localisation des points clés : les points clés générés dans l'étape précédente sont ensuite utilisés pour définir les points clés les plus importants. Certains points clés se trouvent sur le bord de l'image, ou ont un contraste trop faible, ils ne sont donc pas utiles en tant que points clés.
3. Affectation d'orientation : le voisinage est pris autour du point clé en fonction de l'échelle. Le gradient de magnitude et de direction est calculé dans cette région. Par exemple, le gradient de direction à un certain point est de 18.89 degrés ; il ira donc dans l'intervalle 10-19 degré. La quantité qui est ajoutée à cette intervalle est proportionnelle à la magnitude du gradient à ce point. Une fois cette étape réalisée pour tous les points clés, il est possible d'extraire un histogramme des valeurs. Les pics de l'histogramme, et toute valeur au-dessus de 80%, sont considérés pour calculer l'orientation. Cela crée des points clés avec la même localisation et échelle, mais avec des orientations différentes.
4. Description d'éléments : tous les éléments ont donc une localisation, une échelle et une orientation. La prochaine étape est de générer un descripteur pour les régions des images pour chaque point clé qui est hautement distinct et invariant aux changements de points de vue et d'illumination. Pour cela, une fenêtre de 16x16 est prise autour de chaque point clé.
5. Correspondance de points clés : finalement, les points clés des différentes images peuvent être mis en correspondance en comparant leur descripteur.

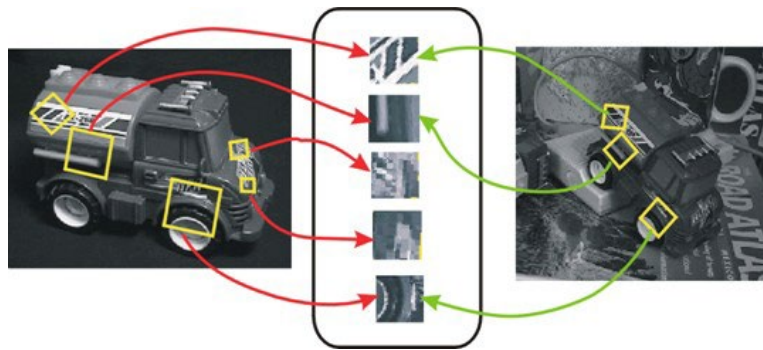


Figure 3, Correspondance des points clés en utilisant SIFT, selon Tyagi (2020)

6.3.2 Performances des algorithmes

Nous allons ici, afin de traiter notre deuxième question de recherche, comparer trois algorithmes largement utilisés dans la reconnaissance d'images en lien avec la réalité augmentée :

- SIFT, décrit ci-dessus.
- SURF (Speed Up Robust Features), dont l'intérêt principal est sa rapidité d'exécution.
- ORB (Oriented FAST and Rotated BRIEF), un algorithme en libre accès, principalement développé comme alternative à SIFT et SURF qui sont des algorithmes brevetés.

6.3.2.1 SURF (Speed Up Robust Features)

Cet algorithme, développé en 2006, vise à améliorer la rapidité d'exécution de l'algorithme SIFT. Cette méthode est composée de quatre étapes principales :

- 1) Extraction d'éléments qui détecte les points clés sur l'image : une grille est générée sur l'image et chaque cellule (pixel) se voit attribuer une valeur selon son intensité. Une image intégrale est générée sur la base de ces cellules. Une image intégrale est une représentation numérique de l'image originale. Chaque pixel se voit attribuer un chiffre, qui correspond à la somme de tous les pixels qui se situent à gauche et au-dessus du pixel en question (Tyagi, 2020). Au lieu de directement recalculer la valeur des cellules après un flou gaussien, comme SIFT, une estimation du flou gaussien selon l'échelle est faite, ce qui accélère fortement le processus, puisqu'il n'y a pas besoin d'itérer sur la même image pour chaque échelle différente. Des images DoG sont utilisées afin de créer une nouvelle gamme d'images appelée « représentation de l'espace d'échelle d'une image » (*scale-space representation*). Ce nouveau lot d'images contient donc la représentation des images générées avec différents niveaux d'échelle et de flou. Les maxima et minima locaux dans l'espace d'échelle sont identifiés comme des points clés potentiels.
- 2) Localisation des points clés : la position, l'échelle et l'orientation des points clés est estimée en ajustant une fonction parabolique en 2D aux extrema dans l'espace d'échelle.
- 3) Description d'élément : l'orientation de chaque point clé est déterminée dans l'image. L'algorithme regarde la région autour de chaque point clé et construit un histogramme des orientations des gradients. Il regarde ensuite quelle orientation montre la plus grande valeur dans l'historgramme et lui assigne cette orientation. Cette dernière étape assure que chaque point clé est orienté de la même manière.
- 4) Correspondances des points clés : de manière similaire à SIFT, les points clés des différentes images sont comparés et mis en correspondance selon leur descripteur.

6.3.2.2 ORB (Oriented Fast and Rotated Binary Robust Independent Elementary Features)

ORB est une combinaison d'algorithmes rapides pour la détection d'images. Il montre une réelle efficacité pour la détection d'images tout en présentant une rapidité exemplaire (Tareen & Saleem, 2018). Il utilise l'algorithme FAST pour la détection de points clés et l'algorithme BRIEF pour la description. Ces deux algorithmes sont utilisés car ils sont performants et peu coûteux.

- 1) Détection des points clés : L'algorithme FAST calcule la luminosité pour un pixel donné. Il dessine ensuite un petit cercle autour de ce pixel et calcule la luminosité des 16 pixels qui touchent ce cercle. Si 8 pixels sont plus ou moins lumineux que le pixel de base, ce dernier est alors choisi comme un point clé. ORB crée en premier lieu une pyramide composée d'une même image à des échelles différentes. Des points clés, utilisant FAST, sont détectés. Selon le changement d'intensité autour d'un point clé, ORB lui assigne une rotation.
- 2) Rotation invariante : ORB utilise la même méthode que SURF, à savoir la construction d'un histogramme des gradients d'orientation pour chaque point clé, pour construire des descripteurs robustes quant aux changements d'orientation.
- 3) Description d'élément : ORB utilise une variante de l'algorithme BRIEF, qui prend tous les points clés et les convertit en vecteurs binaires. Il sélectionne ensuite une paire aléatoire de pixels dans un voisinage défini autour d'un point clé. Après un calcul, la paire de pixels se voit assigner un chiffre (0 ou 1), selon la différence de luminosité avec leur voisin. BRIEF effectue cette opération 128 fois autour d'un point clé. Un vecteur est ainsi créé avec ces valeurs pour chaque point clé dans l'image.
- 4) Correspondance des points clés : les points clés sont finalement comparés grâce à leur descripteur et sont mis en correspondance.

6.3.3 Comparaison des algorithmes

Karami et al. (2017), Tareen & Saleem (2018) et Ou et al. (2020) proposent une comparaison de la performance et la robustesse de ces trois différents algorithmes.

Les algorithmes sont soumis à des tests qui transforment l'orientation, la luminosité, l'échelle, la distorsion ou encore le bruit d'une image de base (Fig. 4, 5 & 6). Des indicateurs sont ensuite calculés selon leurs performances, en prenant en compte la durée de détection de points clés, le nombre de points clés détectés, le temps nécessaire pour faire correspondre deux images et le nombre de points clés correspondant entre les deux images. Nous présentons ici les résultats de la durée complète des algorithmes, le nombre de points clés détectés, ainsi que le taux de correspondance pour les différents tests. Nous regardons également les coûts computationnels requis par point clé, ce qui nous donne une idée de la performance des différents algorithmes.

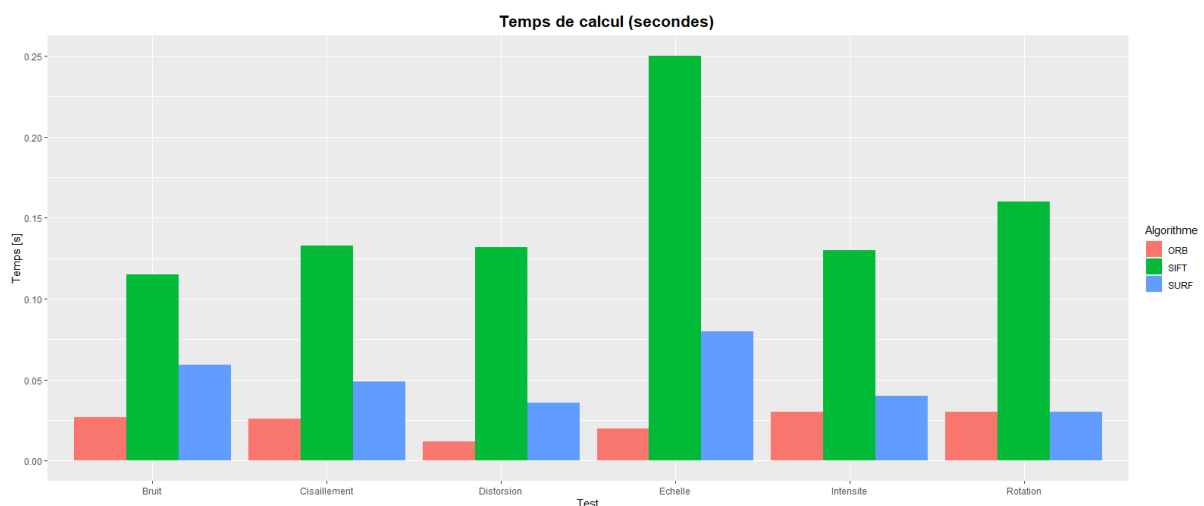


Figure 4, Temps de calcul (selon Karami et al., 2017)

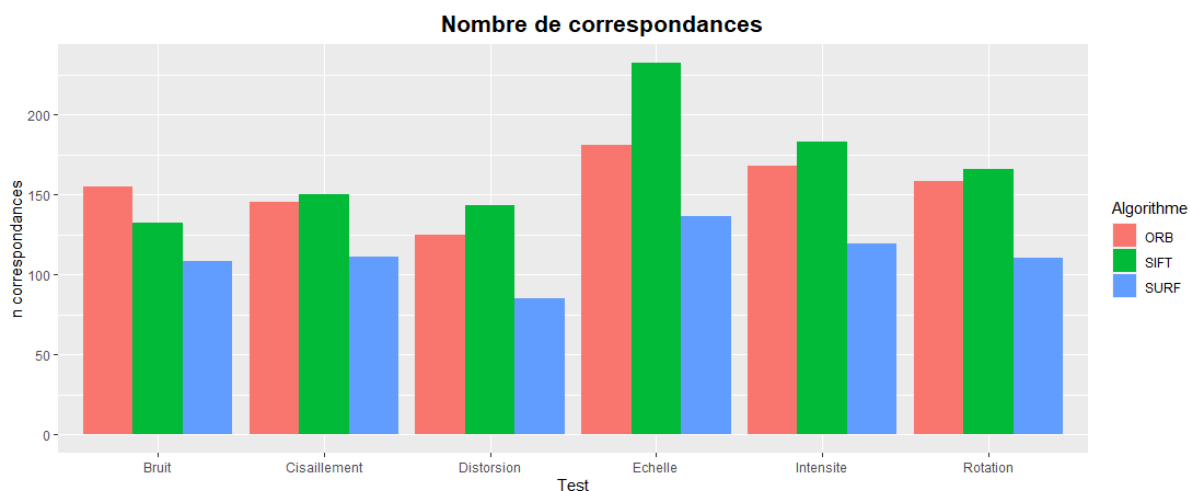


Figure 5, Nombre de correspondances (selon Karami et al., 2017)

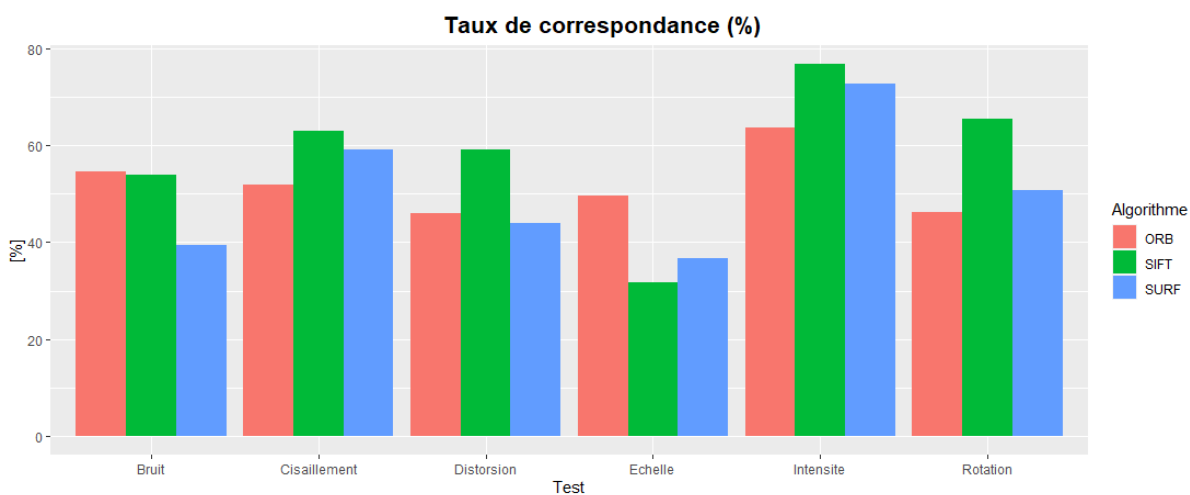


Figure 6, Taux de correspondance (selon Karami et al., 2017)

Algorithme	Durée moyenne de détection-description d'élément par point clé (µs)		Durée moyenne de correspondance par point clé (µs)
	1 ^{ère} image	2 ^{ème} image	
SIFT	90.44	85.15	142.02
SURF	42.78	42.22	168.55
ORB	13.51	13.92	11.82

Table 1, Durée de détection selon l'algorithme (selon Tareen & Saleem, 2018)

6.3.4 Discussion de la performance des algorithmes

SIFT et SURF détectent en général le plus grand nombre de points d'intérêt. Cela influence leur robustesse pour la correspondance entre deux images, mais rend également la durée de détection plus longue. Bien que le temps de calcul de SURF soit bien moins important que SIFT, il reste plus élevé que pour ORB. Les points d'intérêts que SIFT et SURF détectent sont éparpillés de manière égale à travers l'image, alors qu'ORB détecte mieux les coins des éléments (Tareen & Saleem, 2018). Les points clés qu'ORB détecte semblent être plus pertinents.

SURF montre également le taux de correspondance le moins élevé parmi les trois algorithmes. SIFT est capable de faire correspondre le plus grand nombre de points clés à travers l'ensemble des tests. En conséquence, il semble être le plus robuste et le plus consistant, mais son nombre de points clés détectés

élevé influence fortement sur le temps de calcul total. ORB présente le temps de détection et de correspondance le moins élevé parmi l'ensemble des tests.

En ce qui concerne les coûts computationnels, ORB est de loin le plus efficace pour la détection de points clés. Pour la correspondance d'images, ORB reste le plus efficace, suivi de SIFT, puis SURF. Pour le coût total de l'ensemble des processus, ORB est le plus rapide, puis SIFT et SURF, qui varie selon les paramètres indiqués.

SIFT semble être l'algorithme le plus robuste et consistant, quelques soient les modifications apportées à l'image. Il est en revanche le plus lent. ORB montre d'excellents résultats en termes de rapidité et de coûts computationnels, ainsi qu'un taux de correspondance assez élevé. SURF montre des résultats moins bons dans les deux cas de figure.

En conclusion, dans le cadre de la réalité augmentée sur mobile, il serait préférable d'opter pour l'algorithme ORB. Présentant des résultats satisfaisants dans le taux de correspondance, son faible coût computationnel permettrait de réaliser une application en réalité augmentée en réduisant la consommation excessive de batterie.

6.4 Solutions d'implémentation de la réalité augmentée sur le web

6.4.1 Types d'implémentations

Il existe différentes solutions afin d'implémenter une application mobile en réalité augmentée. La solution purement *front-end*, la solution visant à modifier le noyau (*kernel*) du navigateur web, et la solution basée sur le *Cloud*.

Dans ce chapitre, nous allons tout d'abord définir chacune de ces solutions avec des exemples d'implémentation. Nous les comparerons ensuite en mettant en avant leurs avantages et leurs inconvénients au niveau de leurs performances et facilité d'implémentation. Nous concluons finalement en évoquant les différences de coûts monétaires qu'elles engendrent, que ce soit pour les développeurs ou pour les utilisateurs, selon la solution choisie.

a. Solution en modifiant le noyau (*kernel*) du navigateur

Certains navigateurs cherchent à étendre leur kernel afin de supporter directement des expériences en réalité augmentée et en réalité virtuelle (Qiao et al., 2019a). Il est alors possible de développer une application par le biais d'un kit de développement (SDK) dédié (ARKit pour iOS et ARCore pour Android), puis de l'afficher sur le web. Cette solution, en utilisant au mieux les capacités computationnelles du support, permet d'obtenir des performances proches d'une application native.

Mozilla et Google, qui possèdent les navigateurs les plus utilisés, ont chacun développé ces extensions afin de faciliter l'intégration de contenu en réalité augmentée sur un navigateur. L'Immersive Web Community Group, composé de contributeurs comme Google, Microsoft, Meta et Mozilla, sortent WebXR Device API¹³ en février 2018, une API présentant une nouvelle interface afin d'inclure du contenu AR et VR sur le web, en utilisant WebGL. Cette API n'est aujourd'hui que partiellement supportée par l'ensemble des navigateurs web (Fig. 7).

Les détails concernant la définition d'une API, ainsi que la description et fonctionnalités des différentes API détaillées dans ce chapitre sont disponibles dans le chapitre 7.2.

¹³ <https://immersiveweb.dev/> consulté le 03.01.2023

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-64				10-51												
65-78	12-18	3.1-12.1	2-76	52-65				4-11.2								
79-107	79-107	13-16.0	77-106	66-91	6-10		3.2-16.0	12.0-18.0		12-12.1		2.1-4.4.4				
108	108	16.1	107	92	11	108	16.1	19.0	all	72	13.4	108	107	13.1	13.18	2.5
109-111		16.2-TP	108-109													

Figure 7, Support actuel du WebXR Device API (en rouge, les navigateurs ne le supportant pas; en vert foncé, les navigateurs le supportant partiellement) (selon caniuse.com, consulté le 13.12.2022)

Mozilla développe WebXR Viewer¹⁴ en 2017, une application mobile, uniquement disponible pour iOS, permettant de créer et d'afficher du contenu en réalité augmentée. Elle est basée sur le kit de développement d'Apple, ARKit, un logiciel permettant de facilement créer, d'interagir et de publier du contenu en réalité augmentée via une interface graphique dédiée. Google développe de son côté WebARonARCore¹⁵, basé sur ARCore, leur plateforme de création d'expérience en réalité augmentée, et WebARonARKit¹⁶, basé sur ARKit. Ces deux navigateurs expérimentaux permettent aux développeurs de facilement tester leur contenu. Malheureusement, ces extensions de navigateurs sont encore en phase de test et ne sont pas, à ce jour, directement intégrés dans les navigateurs de base proposés par Mozilla et Google.

b. Solution purement front-end

Il consiste ici à utiliser une librairie JavaScript spécifiquement conçue pour implémenter de la réalité augmentée sur un navigateur web en s'appuyant sur les langages typiques de développement web (HTML, CSS et JavaScript) (Qiao et al., 2019a). Parmi ces librairies, nous pouvons nommer JSARToolKit, Awe.js et AR.js. Ces librairies utilisent différents frameworks (environnements) et API afin de développer une application en réalité augmentée en quelques lignes de code (Qiao et al., 2018). Il est commun de devoir passer par les API WebGL pour le rendu d'objets 3D et WebRTC pour le support des périphériques en entrée (caméra, microphone, capteurs).

L'intégralité des calculs spécifiques à la réalité augmentée est réalisée directement sur le téléphone mobile. En raison des capacités limitées de calcul de matrice de JavaScript, nécessaire pour l'*image tracking AR* (Qiao et al., 2018), cette solution est extrêmement énergivore et dépend grandement de la qualité et des performances des composants du support utilisé (Qiao et al., 2019a). Des calculs complexes résultent en une consommation excessive de la batterie et un nombre d'images par seconde relativement bas (Qiao et al., 2019b).

c. Solution transférant les calculs vers le Cloud

Il s'agit ici de partir d'une application développée en JavaScript et de transférer la tâche de calcul au Cloud afin d'alléger le travail du téléphone mobile. Cette solution a l'avantage de non seulement rendre une application en réalité augmentée moins énergivore pour le téléphone mobile, mais également de rendre l'expérience plus immersive en effectuant les calculs sur un Cloud dont les capacités computationnelles sont plus puissantes que ceux des smartphones (Qiao et al., 2019a). En revanche, sous les réseaux 3G et 4G, cette solution montre un temps de latence relativement élevé en raison du temps de communication entre le téléphone mobile et le Cloud. Qiao et al. (2019a ; 2019b) soutiennent l'idée que l'émergence du réseau 5G promet une nouvelle ère de la réalité augmentée mobile sur le web, en réduisant considérablement le temps de latence tout en conservant les performances d'une application purement *front-end*.

¹⁴ <https://labs.mozilla.org/projects/webxr-viewer/> consulté le 03.01.2023

¹⁵ <https://github.com/google-ar/WebARonARCore> consulté le 03.01.2023

¹⁶ <https://github.com/google-ar/WebARonARKit> consulté le 03.01.2023

6.4.2 Comparaison des différentes solutions

Nous allons, dans cette section, comparer en premier lieu les performances des trois solutions disponibles pour la réalité augmentée, à savoir la solution purement front-end et la solution basée sur le Cloud, élucidées précédemment, ainsi que la solution de l'informatique en périphérie (*edge computing*). Pour ce faire, nous nous appuyerons sur leurs caractéristiques suivantes, pour une même application en réalité augmentée : images par seconde (FPS, Frame Per Second), temps de latence et consommation d'énergie. Cette comparaison a été réalisée sur la base de trois études élaborées par Qiao et al. (2018, 2019a et 2019b). Nous finirons par analyser les différences de coûts que représentent ces trois solutions, que ce soit pour l'utilisateur ou pour le développeur.

Trois solutions pour le calcul intensif de la réalité augmentée sont proposées :

- 1) La solution purement front-end
- 2) La solution d'*edge computing* (informatique en périphérie)
- 3) La solution du *cloud computing* (informatique en nuage)

L'*edge computing* est une méthode d'optimisation du *cloud computing* qui vise à effectuer l'ensemble des calculs nécessaires (ici à la réalité augmentée) au plus proche de l'utilisateur (Qiao et al., 2018). Dans les méthodes de *cloud computing* traditionnelles, les données et calculs spécifiques à une application sont envoyés sur une machine virtuelle qui se trouve à endroit précis dans le monde. Afin de réduire la latence en utilisant cette méthode, un développeur va choisir une machine virtuelle proche de sa localisation. Or, le développeur ne sera pas le seul à utiliser son application. Le temps de latence augmente en fonction de la distance à la machine virtuelle. L'*edge computing* tente de réduire la latence en traitant les données directement sur le smartphone, sur un dispositif informatique proche ou sur une machine virtuelle se trouvant à une faible distance de l'utilisateur. Cette proximité permet de réduire le temps de transfert de données et ainsi obtenir un temps de latence moins important que la méthode de *cloud computing* traditionnelle.

6.4.3 Résultats et discussion des différences de coûts

6.4.3.1 Coûts computationnels

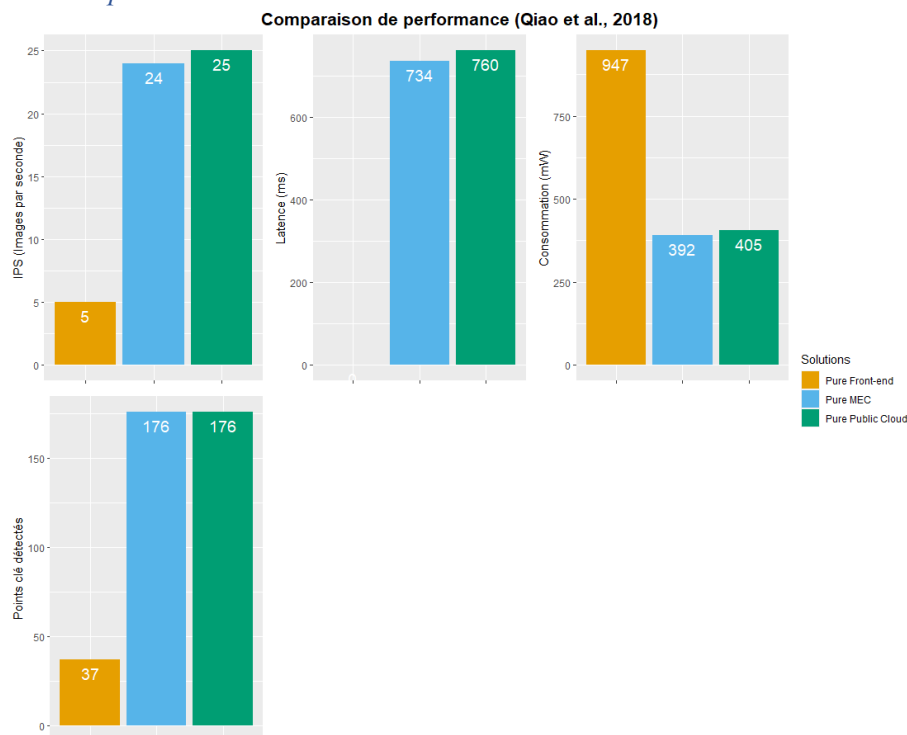


Figure 8, Performances avec une solution simple (selon Qiao et al., 2018)

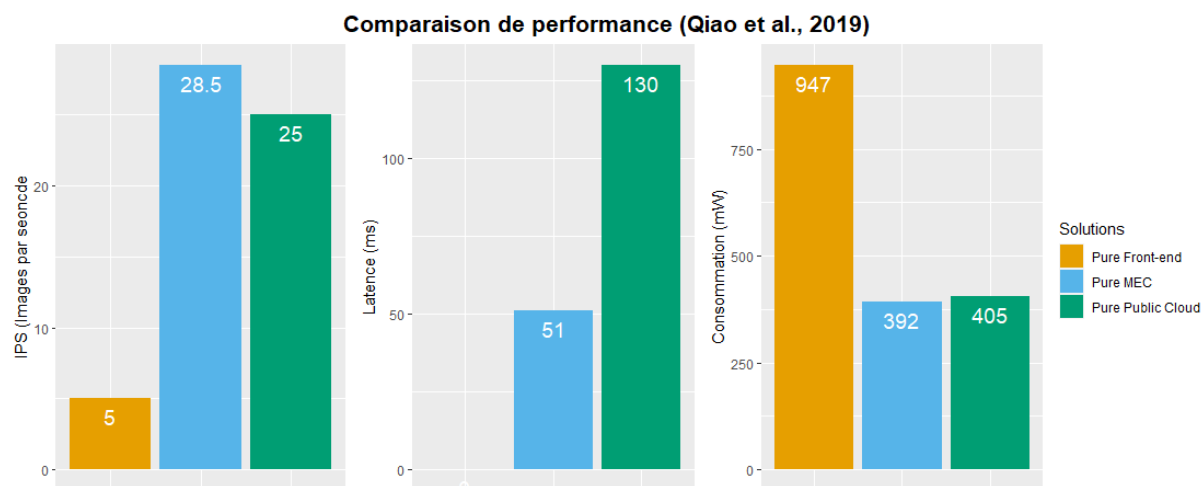


Figure 9, Performances avec une solution simple (selon Qiao et al., 2019b)

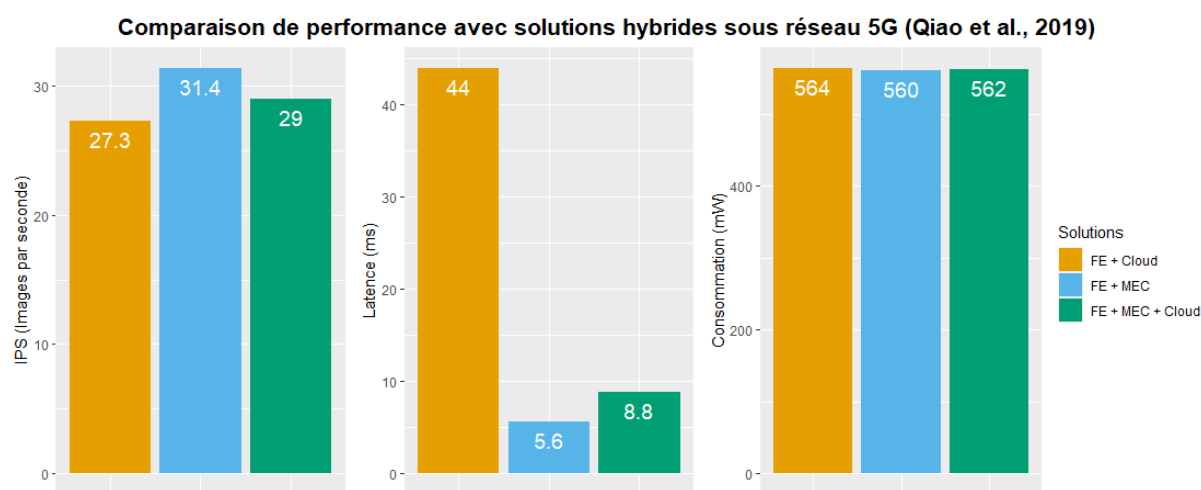


Figure 10, Performances avec une solution hybride (selon Qiao et al., 2019a)

Les deux premières comparaisons (Fig. 8 & Fig. 9) réalisées par Qiao et al. (2018 & 2019b) ont été effectuées sur des solutions uniques, c'est-à-dire en ne croisant pas les différentes technologies afin d'en tirer les meilleures performances.

En ce qui concerne la solution purement front-end, il est évident qu'elle présente les moins bons résultats au niveau de la fluidité de rendu (images par seconde). La capacité limitée des navigateurs web à effectuer des calculs difficiles ainsi que les faibles performances de JavaScript pour des tâches complexes (ici les algorithmes de reconnaissance d'images) (Qiao et al., 2019b) impliquent une réduction considérable de la fluidité du rendu affiché à l'écran. Puisque l'ensemble des calculs est effectué directement sur le téléphone mobile, la latence que présente cette solution est nulle. Aucun transfert de données ne doit être effectué. En revanche, ces lourdes tâches computationnelles effectuées de manière locale impactent grandement la consommation d'énergie pour cette solution.

Les deux autres solutions présentent une fluidité de rendu similaire, mais la solution du *edge computing* montre un temps de latence relativement plus faible, puisque les calculs sont effectués plus proche de l'utilisateur. Ces deux solutions sont également nettement moins énergivores, ne dépendant pas entièrement des performances du téléphone mobile et du navigateur afin d'effectuer les tâches compliquées.

D'une manière générale, la solution du *mobile edge computing* semble être la plus adéquate afin d'atteindre les meilleures performances pour une application web mobile en réalité augmentée. Elle profite des avantages que présentent les deux autres solutions, c'est-à-dire un nombre d'images par seconde et un temps de latence exemplaires en effectuant les calculs proches de l'utilisateur, tout en ayant une consommation assez faible. Qiao et al. (2019a ; 2019b) mettent également en avant des solutions hybrides (Front-End + Cloud, Front-End + Mobile Edge Computing, Front-End + Mobile Edge Computing + Cloud) qui voient leurs performances croître grâce à l'émergence et l'émancipation du réseau 5G (Fig. 10), qui a pour principal avantage de réduire considérablement le temps de latence lors de transfert de données entre un téléphone mobile et le Cloud.

6.4.3.2 Coûts monétaires

En ce qui concerne la différence de coûts financiers entre une solution purement *front-end* et une solution basée sur le Cloud, la première solution reste beaucoup moins chère. Les plus grands fournisseurs de Cloud à l'heure actuelle sont Amazon, avec leur service AWS, Google, avec leur service Google Cloud Platform et Microsoft, avec leur service Azure. Chacun de ces fournisseurs proposent un tarif mensuel pour le *cloud computing* dépendant des performances souhaitées. Une comparaison a été faite pour trois types de machines : (1) usage général, (2) à haute capacité de mémoire, (3) à haute capacité du processeur, basée sur les résultats de Simform¹⁷ et cast.ai¹⁸ :

Type d'instance	Tarif horaire (\$/heure)		
	AWS	Google	Azure
Usage général	0.154	0.134	0.166
Optimisé en terme de calcul	0.136	0.208	0.169
Optimisé en terme de mémoire	0.202	6.293	0.252

Table 2, Comparaison des tarifs horaires pour différents services basés sur le cloud (selon Simform et cast.ai, consulté le 02.01.2023)

Les tarifs (table 2), dans l'ensemble, sont assez similaires pour les trois fournisseurs. Le tarif horaire pour les machines virtuelles à haute capacité de mémoire de Google Cloud est beaucoup plus élevé car il utilise 40 processeurs contre 4 pour les deux autres services.

En prenant l'option optimisée en termes de calcul afin de tirer les meilleures performances pour une application web mobile en réalité augmentée, le tarif mensuel revient à environ 100\$ par mois. Nous nous basons ici sur les tarifs mis à jour en date du 2 janvier 2023. Les tarifs étant en constante fluctuation et évolution, il est difficile d'obtenir un chiffre précis. Il est important à noter que les trois fournisseurs mentionnés ci-dessus proposent des rabais allant jusqu'à 60% pour des abonnements annuels, ce qui revient à environ 40\$ par mois.

Il est relativement aisé aujourd'hui d'héberger gratuitement un site internet par le biais de différents fournisseurs. Pour l'hébergement de sites web plus complexes, ou pour des services spécifiques (comme la volonté d'avoir un nom de domaine personnalisé), les tarifs horaires dépassent rarement les 10\$ mensuels. Si notre unique intérêt était ici de réduire au maximum les coûts de développement, de déploiement et d'hébergement de notre application web, la solution purement front-end resterait la moins coûteuse.

¹⁷ <https://www.simform.com/blog/compute-pricing-comparison-aws-azure-googlecloud/> consulté le 10.01.2023

¹⁸ <https://cast.ai/blog/ultimate-cloud-pricing-comparison-aws-vs-azure-vs-google-cloud-in-2021/> consulté le 10.01.2023

7. Méthodologie

Ce chapitre explique les différentes étapes nécessaires à l'implémentation d'une application web en réalité augmentée. Elle vise à clarifier certaines notions clés et présenter les outils utilisés pour l'élaboration du produit final.

7.1 Contexte

L'application en question est basée sur le GéoGuide Lausanne¹⁹, un sentier didactique interactif permettant aux utilisateurs d'étendre leurs nouvelles connaissances concernant l'environnement à Lausanne. Développé par la Faculté des Géosciences et l'Environnement de l'Université de Lausanne, elle vise à valoriser le patrimoine de la ville de Lausanne en guidant l'utilisateur à travers divers lieux présentant des intérêts géologiques, géomorphologiques, glaciologiques, hydrologiques, urbanistiques et historiques. Plongé dans un contenu interactif et informatif, et suivant un sentier prédéfini, l'utilisateur est capable d'en apprendre davantage sur des phénomènes invisibles à l'œil nu. L'application est disponible directement sur le web, ou en téléchargement pour iOS et Android.

La réalité augmentée sur téléphone mobile a permis de changer la manière dont les utilisateurs visitent une destination. Rendant la visite plus interactive, immersive et diversifiée, elle permet de créer des expériences uniques et personnalisées (Jung et al., 2015). Il a également été montré que la réalité augmentée facilite la compréhension de phénomènes plus complexes, en proposant un environnement interactif en trois dimensions (Kamarainen et al., 2018). Jouissant d'un héritage géomorphologique complexe et de zones densément bâties, la ville de Lausanne présente de nombreux paysages hérités invisibles à l'œil nu. En superposant du contenu en réalité augmentée par-dessus le monde réel, il est alors possible de révéler ces différents processus cachés aux utilisateurs.

Le but de l'application réalisée dans cette étude est de susciter l'intérêt de l'utilisateur, en l'immergeant totalement dans l'environnement grâce à la réalité augmentée. Révélant des phénomènes invisibles, reconstruisant l'héritage géomorphologique de la ville de Lausanne et permettant l'interaction en temps réel avec l'environnement, l'application laisse à l'utilisateur l'opportunité de s'informer, de manière immersive, quant à un lieu. L'application est conçue comme un jeu, une chasse au trésor. L'utilisateur suit un itinéraire prédéfini, marqué par différentes étapes. Chaque étape propose d'informer l'utilisateur grâce à du contenu en réalité augmentée, souvent développé sous forme de mini-jeu interactif. Une fois le mini-jeu réalisé et les informations acquises, la prochaine étape se débloque.

7.2 Environnements, librairies et API

Nous allons, dans ce chapitre, passer en revue les différents outils utilisés afin d'implémenter du contenu en réalité augmentée sur le web, en vue notamment de traiter notre troisième question de recherche. Il ne s'agit en aucun cas d'une liste exhaustive, mais uniquement des outils utilisés dans le cadre de cette étude.

7.2.1 Définitions

1) Librairie

Une librairie est un ensemble de fonctions préalablement établies et écrites que les développeurs peuvent utiliser afin d'optimiser certaines tâches, gagnant ainsi du temps et réduisant considérablement le nombre de lignes de code nécessaires.

¹⁹ <https://igd.unil.ch/geoguide/fr/> consulté le 03.01.2023

2) Framework web (infrastructure logicielle web ou environnement)

Un *framework*²⁰ web est un ensemble de ressources qui permet de construire et de maintenir un site ou une application web. Il vise à faciliter le travail d'un développeur en proposant un environnement regroupant plusieurs bibliothèques, fonctions et API spécifiquement utiles à la tâche du développeur.

3) API

Une interface de programmation d'application²¹ (API, Application Programming Interface) offre un ensemble de solutions pour la communication entre une application consommatrice d'un service et une application tierce productrice de ce service. Elle permet donc aux développeurs d'utiliser, par le biais d'une bibliothèque logicielle, des services fournis par une application tierce en quelques lignes de commande. L'application *consommatrice* peut ainsi communiquer avec l'application *productrice* afin d'utiliser ses services. En ce qui concerne l'affichage de contenu en réalité augmentée sur une page web, plusieurs API sont nécessaires.

7.2.2 Ressources utilisées

a) ARToolKit

ARToolKit (Augmented Reality Tool Kit)²² est une bibliothèque logicielle, en langage de programmation C, permettant l'affichage et la manipulation d'objets en réalité augmentée. Elle utilise OpenGL pour le rendu d'image, GLUT (OpenGL Utility ToolKit)²³ pour l'interaction et une API simple pour les différentes plateformes sur lesquelles l'application va être exécutée. Elle fut la première bibliothèque en libre accès capable de faire du traçage d'image performant, participant grandement à l'essor progressif de la réalité augmentée dès les années 2000.

b) OpenGL

OpenGL²⁴ (Open Graphic Library), lancé en 1992, est une API donnant accès à un large panel de fonctions permettant de manipuler des graphiques et images en 2D et 3D, sous forme de primitives (points, vecteurs, polygones, bitmaps ou textures).

c) WebGL

WebGL²⁵ (Web Graphic Library) est une API JavaScript pour l'affichage d'un rendu graphique interactif en 2D ou 3D sans passer par l'utilisation d'un plugin. Ses fonctionnalités, dérivées d'OpenGL, fonctionnent dans un contexte HTML en utilisant son élément `<canvas>`. Il s'agit d'une API de bas niveau, ce qui veut dire qu'une quantité de code assez conséquent est nécessaire afin de créer un objet simple. WebGL est prise en charge par tous les navigateurs modernes (Fig. 11).

²⁰ https://developer.mozilla.org/fr/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction consulté le 13.01.2023

²¹ https://developer.mozilla.org/fr/docs/Learn/JavaScript/Client-side_web_APIs/Introduction consulté le 13.01.2023

²² <http://www.hitl.washington.edu/artoolkit/documentation/devframework.htm> consulté le 03.01.2023

²³ https://www.opengl.org/resources/libraries/glut/glut_downloads.php consulté le 03.01.2023

²⁴ <https://www.opengl.org/> consulté le 03.01.2023

²⁵ https://www.khronos.org/webgl/wiki/Getting_Started consulté le 03.01.2023

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-7		3.1-5	2-3.6	10-11.5												
8-32	12-18	5.1-7.1	4-23	12.1-18			3.2-7.1									
33-106	79-106	8-16.0	24-105	19-90	6-10		8-16.0	4-17.0		12-12.1		2.1-4.4.4				
107	107	16.1	106	91	11	107	16.1	18.0	all	64	13.4	107	106	13.1	13.18	2.5
108-110		16.2-TP	107-108													

Figure 11, Support de WebGL (selon caniuse.com, consulté le 14.12.2022)

d) WebRTC

WebRTC²⁶ (Web Real Time Communication) est une API JavaScript pour l’implémentation de fonctionnalités de communication (vidéo, audio) en temps réel. Elle permet donc à la page web d’accéder aux systèmes audio (microphone, haut-parleurs) et vidéo (caméra intégrée) de l’appareil. WebRTC est prise en charge par la plupart des navigateurs modernes (Fig. 12).

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-22	12-14		2-21	10-17												
23-55	15-18	3.1-10.1	22-43	18-42			3.2-10.3									
56-106	79-106	11-16.0	44-105	43-90	6-10		11-16.0	4-17.0		12-12.1		2.1-4.4.4				
107	107	16.1	106	91	11	107	16.1	18.0	all	64	13.4	107	106	13.1	13.18	2.5
108-110		16.2-TP	107-108													

Figure 12, Support de WebRTC (selon caniuse.com, consulté le 14.12.2022)

e) JSARToolKit5 (ARToolkit.js)

JSARToolKit²⁷ (JavaScript Augmented Reality Tool Kit) est une librairie JavaScript, utilisant l’API WebRTC afin d’accéder aux différents capteurs du systèmes, ainsi que WebGL pour le rendu d’objets en 3D. Cette librairie est construite sur la base d’ARToolkit. Les images capturées grâce à WebRTC sont stockées dans des éléments <canvas>. Chacun de ces <canvas> est ensuite analysé par JSARToolKit afin de détecter un supposé marqueur, pour finalement afficher du contenu en réalité augmentée par-dessus.

f) Three.js

Three.js²⁸ est une librairie JavaScript permettant d’implémenter du contenu 3D sur une page web. Basée sur WebGL, elle vise à faciliter grandement la création de scène et d’objet en 3D pour les développeurs, en passant par un nombre de lignes de code extrêmement réduit. Il est possible, comme dans un logiciel 3D, de créer différents objets complexes, scènes, lumières, ombres, etc. à l’aide de balises prédéfinies. Il est également possible d’importer des objets 3D créés sur un logiciel tiers.

g) A-Frame

A-Frame²⁹ est un *framework* web spécifiquement conçu pour construire du contenu en réalité augmentée et en réalité virtuelle. A-Frame est une extension de Three.js, permettant la création de réalité augmentée en passant uniquement par l’écriture de balises prédéfinies. Il propose également un environnement de travail directement intégré dans le navigateur web permettant d’interagir avec les objets sans modifier une ligne de code.

²⁶ <https://webrtc.org/> consulté le 03.01.2023

²⁷ <https://web.dev/webgl-jsartoolkit-webrtc/> consulté le 03.01.2023

²⁸ <https://threejs.org/> consulté le 03.01.2023

²⁹ <https://aframe.io/docs/1.3.0/introduction/#what-is-a-frame> consulté le 03.01.2023

h) AR.js

AR.js³⁰, développé par Jérôme Etienne en 2017, est une librairie JavaScript open source pour l'implémentation de contenu en réalité augmentée sur le Web. Elle est basée sur JSARToolKit pour la reconnaissance de marqueur ou d'éléments naturels. Elle utilise three.js et A-Frame (au choix) afin de réaliser du contenu en 3D. Elle nécessite des navigateurs qui supportent WebGL et WebRTC. Cette librairie est capable d'afficher un rendu en 60 images par seconde, ce qui n'avait pas encore été fait pour du contenu en réalité augmentée sur le web jusque-là. AR.js propose diverses manières d'implémenter du contenu en réalité augmentée : basé sur la reconnaissance d'un marqueur prédéfini (*marker based AR*), basé sur la localisation de l'utilisateur (*location based AR*) et basé sur la reconnaissance d'éléments naturels (*image tracking AR*).

7.3 Objectifs de l'application

L'une des premières étapes lors de l'élaboration d'un contenu de géovisualisation est la définition des objectifs principaux du projet, ainsi que la création d'un prototype de l'application.

L'application a été pensée comme un medium de contenu interactif visant à informer les utilisateurs, de manière ludique, quant aux différents processus et phénomènes environnementaux qui ont façonné la ville de Lausanne. Ces phénomènes sont, pour la plupart, invisibles à l'œil nu, et particulièrement invisibles pour les gens qui ne sont pas éduqués à propos de ces différents sujets. Le choix de l'utilisation de la réalité augmentée a été fait afin de surpasser certaines barrières que représentent les applications informatives standards.

Le contenu en réalité augmentée peut être visualisé, manipulé et vécu. Il permet ainsi une plus grande immersion dans l'expérience qu'une application informative standard. Selon Kamarainen et al. (2018), ce type d'expérience facilite l'acquisition de nouvelles connaissances. En liant du contenu virtuel au monde réel, l'utilisateur peut alors visualiser des processus complexes directement en temps réel, permettant ainsi une compréhension facilitée.

Le but premier de notre application était donc de repenser le sentier didactique du GéoGuide Lausanne sous forme de jeu interactif en réalité augmentée. Naviguant à travers la ville de Lausanne dans un monde entre le virtuel et la réalité, l'utilisateur est capable d'interagir avec son environnement afin d'en apprendre davantage quant à divers faits géologiques, géomorphologiques et historiques. L'application propose, à chaque étape, des notions basiques quant à divers sujets. Le public cible sera donc en premier lieu les touristes, étrangers et locaux, souhaitant en apprendre davantage sur la ville de Lausanne. L'application part du principe que l'utilisateur n'a aucune notion préalable à propos des différents sujets abordés. Elle ne fait en rien la distinction d'une personne habitant à Lausanne et une personne venant de l'extérieur. L'ensemble des notions sont reprises à zéro.

Si nous avons ciblé un public précis en relation avec les notions présentées dans l'application, il est en revanche plus délicat de définir un public cible en lien avec l'utilisation de son interface graphique. En effet, les capacités à se familiariser et utiliser une interface graphique est propre à chacun. D'autant plus que la manipulation de contenu en réalité augmentée n'est pas une pratique quotidienne pour la plupart des gens. Il est donc essentiel d'accompagner les utilisateurs tout au long de leur expérience, donnant les instructions les plus claires et concises possibles pour l'utilisation générale de l'application et pour chaque étape. En proposant une marche à suivre trop longue, cela pourrait avoir pour effet de perdre l'attention de l'utilisateur. Au contraire, en donnant des instructions trop courtes, les utilisateurs pourraient se perdre sans savoir comment utiliser certaines fonctionnalités. Il faut donc trouver un juste milieu permettant l'utilisation aisée de l'application par le plus grand nombre de personnes possible, indépendamment de leur familiarité avec la technologie utilisée. Chaque étape propose donc une

³⁰ <https://ar-js-org.github.io/AR.js-Docs/> consulté le 03.01.2023

description des fonctionnalités implémentées. Au moyen d'un menu de navigation rétractable, il est possible de lire à nouveau chaque instruction si nécessaire.

Le choix d'une application web s'est faite afin de toucher un grand nombre d'utilisateurs. En effet, ne requérant pas une installation au préalable, et permettant une compatibilité entre système d'exploitation aisée (Qiao et al., 2019b), une application web est le choix idéal afin de proposer un GéoGuide facilement accessible à tous.

7.3.1 Prototype

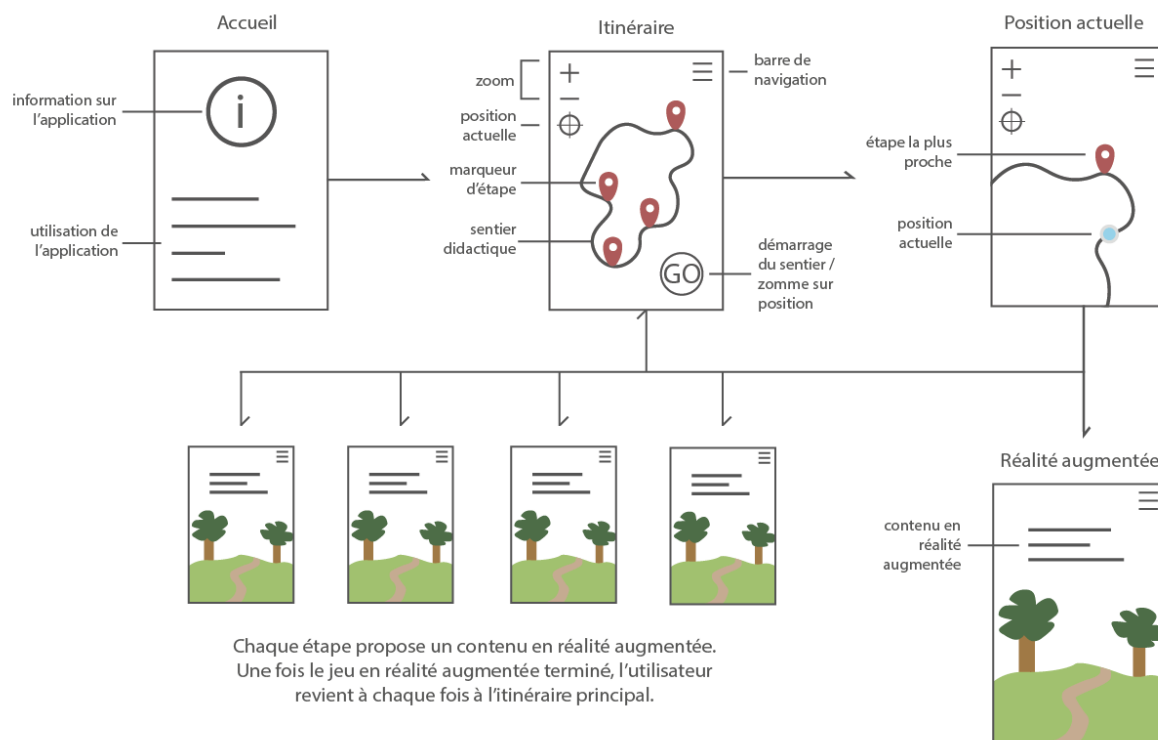


Figure 13, Premier prototype de l'application

Le prototype (Fig. 13) proposé vise à offrir l'expérience d'un sentier didactique en réalité augmentée de la manière la plus simple possible.

- 1) Accueil : l'utilisateur arrive tout d'abord sur un écran d'accueil qui affiche des informations relatives aux buts de l'application et de son utilisation. Cette page d'information peut ensuite être fermée afin d'afficher le sentier didactique en question.
- 2) Itinéraire : ce sentier est représenté par une ligne superposée à un fond de carte simple. Des marqueurs informent l'utilisateur quant à la localisation des différentes étapes à visiter. Une barre de navigation rétractable permet d'afficher les informations essentielles à nouveau, si besoin. Un bouton permet de commencer le sentier, ce qui revient en pratique à géolocaliser l'utilisateur et zoomer sur sa position actuelle.
- 3) Itinéraire actif : la position actuelle de l'utilisateur est croisée avec la position de l'étape à visiter. Si l'utilisateur est assez proche, il peut alors commencer le tour en réalité augmentée.
- 4) Réalité augmentée : le tour en réalité augmentée utilise la caméra du téléphone mobile de l'utilisateur, ainsi que ses différents capteurs intégrés, afin de superposer diverses informations supplémentaires par-dessus l'environnement réel.
- 5) Itinéraire actif : une fois le tour en réalité augmentée terminé, l'utilisateur est ramené à l'étape 3 (itinéraire actif) afin de se rendre à la prochaine étape.

7.4 Langages et bibliothèques utilisées

L'application a été développée à l'aide de trois langages informatiques : HTML, CSS et JavaScript.

- 1) HTML (HyperText Markup Language) : le langage HTML est le langage basique pour la création et la structure de pages web. Son système de balise permet de structurer efficacement un fichier en indiquant si le contenu à l'intérieur de ces balises est un paragraphe, une liste, un titre, etc. Un élément est délimité par un ensemble de deux balises (qui ouvre l'élément, représenté par `<balise>`, et qui le ferme, représenté par `</balise>`) ainsi que son contenu. Un élément peut également contenir un attribut, définissant sa classe, son identifiant, son style, etc. La balise `<div>` permet de créer des éléments a priori invisibles avant de leur appliquer un style.
Un fichier HTML représente le corps, la structure d'une page web. Il peut y être indiqué le chemin d'accès des autres fichiers nécessaires au fonctionnement d'une application, comme le fichier CSS ou le fichier JavaScript. Un fichier HTML se voit attribuer l'extension de fichier `.html`.
- 2) CSS (Cascading Style Sheet) : le langage CSS permet de mettre en forme une page web. En reprenant les éléments spécifiés dans les éléments HTML, il est possible de définir le style des éléments, comme par exemple afficher du texte en vert, changer la couleur de fond en bleu, etc. Un fichier CSS se voit attribuer l'extension de fichier `.css`.
- 3) JavaScript : JavaScript est un langage de programmation permettant de dynamiser et de rendre une page web interactive. À l'aide de variables, de structures conditionnelles et de fonctions, il est possible de récupérer les éléments décrits dans le fichier HTML et de les rendre interactifs. En s'appuyant sur des API directement intégrés aux navigateurs, il est possible de récupérer et manipuler certaines données comme des flux vidéos, des données de capteurs intégrés, etc. Étant un langage relativement compact, il est aisé d'utiliser des bibliothèques JavaScript élaborées par d'autres développeurs en y faisant référence dans le fichier HTML. La plupart des boutons et des événements relatifs aux documents HTML ont été créés à l'aide de JQuery, réduisant effectivement la quantité de lignes de code nécessaire.

En plus de ces trois langages basiques, nous avons utilisé, pour le développement de notre application, plusieurs bibliothèques JavaScript afin de faciliter une certaine partie du travail :

- 1) AR.js : il s'agit de la bibliothèque principale utilisée dans notre projet. Comme mentionné dans le chapitre 7.2, il s'agit d'une bibliothèque JavaScript permettant l'implémentation de la réalité augmentée directement sur un navigateur web supportant les API WebGL et WebRTC. AR.js propose trois types de réalité augmentée :
 - a. Basée sur la localisation
 - b. Basée sur la reconnaissance d'éléments naturels
 - c. Basée sur la reconnaissance d'un marqueur

Dans le cadre de notre application, nous avons choisi d'utiliser uniquement les deux premières solutions, la troisième n'étant pas adaptée à une application axée sur un sentier didactique se déroulant à l'extérieur. AR.js laisse le choix d'utiliser `three.js` ou `A-Frame` pour la création d'objets 3D. Nous avons choisi `A-Frame` en raison de sa simplicité et de son environnement de développement et de débogage directement intégré dans le navigateur web (Fig. 14).

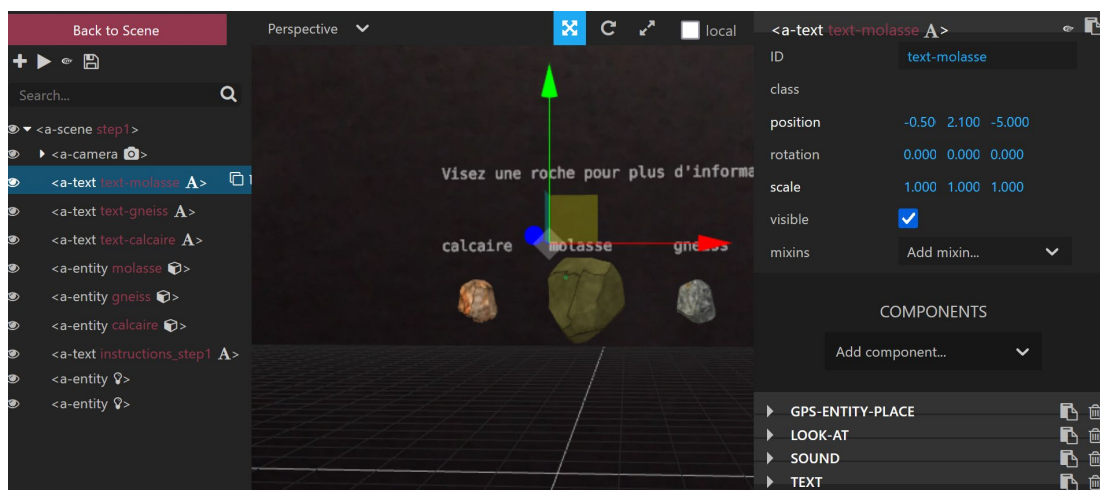


Figure 14, Environnement de développement de A-Frame (intégré au navigateur)

L'implémentation d'objets virtuels 3D se fait simplement en indiquant la balise adéquate selon l'objet souhaité : `<a-box>` pour un cube, `<a-sphere>` pour une sphère, `<a-cylinder>` pour un cylindre, `<a-asset>` pour l'importation de modèles 3D externes ou d'autres fichiers. Dans notre cas, nous avons, la plupart du temps, importé des modèles 3D que nous avons créés grâce au logiciel Blender. Ceux-ci peuvent être enregistrés en format glTF ou GLB afin d'être utilisés sur le web. Un fichier glTF est un fichier contenant les informations d'un modèle 3D en format JSON. Il est assez volumineux et accompagné d'un fichier JPEG contenant les textures à appliquer au modèle 3D. Un fichier GLB est simplement la version binaire d'un fichier glTF. Plus léger et compact, il montre un temps de chargement moins important pour un modèle 3D que le format glTF.

Afin d'afficher les objets en réalité augmentée, il est nécessaire de créer un nouvel élément dans le fichier HTML avec la balise `<a-entity>`. Il est alors possible d'ajouter des attributs à cet élément selon l'objet que nous voulons afficher (Fig. 15). Dans le cas de la réalité augmentée basée sur la localisation, il est nécessaire d'indiquer la latitude et longitude de l'objet que nous voulons afficher. D'autres attributs, comme un identifiant (id), un événement, une échelle, une rotation, etc ... peuvent également être ajoutés.

```

130 let molasse = document.createElement('a-entity');
131     molasse.setAttribute('id', 'molasse');
132     molasse.setAttribute('gps-entity-place', `latitude: ${latitude}; longitude: ${longitude};`);
133     molasse.setAttribute('onclick', 'toggleActive(this.id)');
134     molasse.setAttribute('gltf-model', '#molasse');
135     molasse.setAttribute('scale', '.3 .3 .3');
136     molasse.setAttribute('position', '0 1.3 -5');
137     molasse.setAttribute('rotation', '0 0 0');

```

Figure 15, Ajout dynamique avec JavaScript d'un élément en réalité augmentée basée sur la localisation et ses différents attributs

En ce qui concerne l'implémentation de la réalité augmentée basée sur la reconnaissance d'éléments naturels (NFT, Natural Feature Tracking), un nombre plus important d'étapes sont requises. Il est tout d'abord nécessaire de définir les points clés sur l'image utilisée en entrée. Le programme NFT Marker Creator³¹, proposé par Daniel Fernandes, permet de réaliser cette tâche et de stocker les informations relatives aux points dans des fichiers FSET. Un élément qui utilise la réalité augmentée basée sur la reconnaissance d'éléments naturels est définie avec la balise `<a-nft>`. Il faut ensuite faire référence à ces fichiers, à l'aide d'un attribut « url » (Fig. 16), pour que l'image capturée par la caméra du téléphone mobile soit croisée avec les points

³¹ <https://github.com/Carnaux/NFT-Marker-Creator> consulté le 13.01.2023

clés générés. Les fichiers FSET doivent être appelés avec une URL avec un protocole sécurisé (HTTPS). Il n'est pas possible de les utiliser dans le cas contraire. Si les points clés correspondent, il est alors possible de superposer du contenu à l'élément naturel reconnu. L'algorithme détecte sans cesse les points clés en question afin de définir les mouvements de l'utilisateur par rapport à l'objet. Le contenu s'adapte selon ces mouvements, lui permettant ainsi de toujours être face à l'utilisateur.

```

26 let nft = document.createElement('a-nft');
27 nft.setAttribute('markerhandler', '');
28 nft.setAttribute('registerevents', '' );
29 nft.setAttribute('name', 'panneau2');
30 nft.setAttribute('type', 'nft');
31 nft.setAttribute('url', 'https://raw.githubusercontent.com/theogerritsen/AR_project/main/ar_files/ar_gltf/step3/etape3_p2');
    
```

Figure 16, Ajout dynamique avec JavaScript d'un élément en réalité augmentée basée sur la reconnaissance d'éléments naturels et ses différents attributs

- 2) Leaflet.js³²: il s'agit de la librairie JavaScript utilisée pour l'affichage de la carte interactive montrant le sentier didactique et les différentes étapes. En indiquant, à l'aide de fichiers externes, la géolocalisation des étapes et du sentier, il est extrêmement aisé de les afficher directement sur une carte à l'aide de Leaflet.js. Cette librairie open source donne également la possibilité d'utiliser des extensions proposées par d'autres développeurs. Nous avons notamment utilisé ici une extension permettant de géolocaliser l'utilisateur en temps réel.
- 3) JQuery³³: JQuery est une librairie JavaScript qui simplifie grandement la création d'événements pour une page web. Elle implémente des fonctions simples qui permettent de faire référence rapidement à un identifiant ou une classe, supprimant ainsi certaines lignes de code redondantes lors de la manipulation d'événements sur des éléments HTML.

7.5 Architecture de l'application

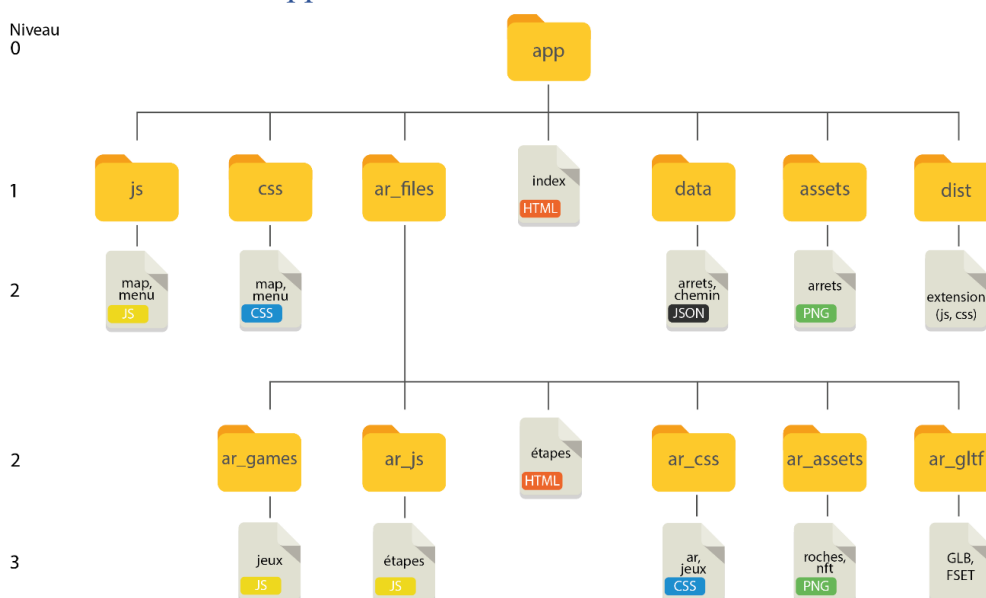


Figure 17, Schéma d'architecture de l'application

L'application possède une architecture hiérarchique simple (Fig. 17) qui permet de facilement retourner à la page d'accueil, tâche qui est réalisée après chaque étape en réalité augmentée. Le dossier principal contient le fichier *index.html* (la page d'accueil), ainsi que tous les autres dossiers contenant les fichiers

³² <https://leafletjs.com/> consulté le 15.01.2023

³³ <https://jquery.com/> consulté le 15.01.2023

nécessaires au fonctionnement de l'application web. La page d'accueil montre chaque marqueur d'étape, qui mène donc à une expérience en réalité augmentée. Le dossier *ar_files* (niveau 1) contient l'ensemble des fichiers proposant des étapes en réalité augmentée. Chaque étape est construite avec son propre fichier HTML et JavaScript (niveau 2). En général, un fichier CSS permet de styliser l'ensemble de ces étapes. Le dossier *ar_gltf* contient les fichiers d'objets en 3D (.glb), modélisés sur Blender³⁴, ainsi que les fichiers contenant les points clés nécessaires à la reconnaissance d'éléments naturels (.FSET). Les fichiers .FSET ont été générés à l'aide du programme NFT-Marker-Creator. Les fichiers FSET ne sont pas directement utilisés avec un chemin relatif, mais sont appelés depuis l'URL de leur contenu qui est stocké en ligne. Après la réalisation d'une étape, l'application redirige l'utilisateur vers la page d'accueil (niveau 1), permettant ainsi d'accéder à nouveau à l'ensemble des étapes en réalité augmentée (niveau 2).

7.6 Déploiement

L'application réalisée dans cette étude a été réalisée sous Windows 11 et testée sur iOS 16 (iPhone SE 2022) avec Safari et XRViewer (v2.0).

Plusieurs outils ont été nécessaires afin de développer et de déployer l'application web en question. Nous présentons ici les différentes étapes ayant permis de développer une application web mobile en réalité augmentée en utilisant AR.js. Uniquement les étapes présentant certaines difficultés sont explicitées ici. Il ne s'agit en rien d'un tutoriel exhaustif pour la création d'une application web mobile.

La première étape fut de déployer un serveur local permettant d'afficher et de travailler sur une application mobile. Les étapes en réalité augmentée qui utilisent les algorithmes de détection d'éléments naturels (NFT) nécessitent le passage par un protocole sécurisé (HTTPS, HyperText Transfer Protocol Secure) afin de faire le transfert de données entre les différents sites web. En effet, les divers fichiers FSET doivent être stockés sur un site web externe, car les stocker dans un fichier local ne fonctionne pas. Afin d'afficher notre application web sur un téléphone mobile, il a donc fallu monter un serveur local possédant un protocole de transfert sécurisé. Cette tâche a été réalisée grâce à Node.js³⁵, un environnement d'exécution (runtime) permettant d'exécuter du code JavaScript sur un serveur. Node.js permet de déployer un serveur HTTPS en local en indiquant le chemin des fichiers du certificat et de la clé SSL en une ligne de commande :

```
http-server -S -C [chemin jusqu'au certificat SSL] -K [chemin jusqu'à la clé SSL]
```

L'application web est alors disponible à trois adresses différentes. L'une de ces adresses correspond à l'adresse IPv4 du réseau local, ce qui permet de visionner le site web sur n'importe quel medium en entrant `https:// [adresse IPv4] : [numéro du port]` dans la barre de recherche du navigateur, à condition qu'il soit connecté au même réseau que celui sur lequel le serveur a été déployé. Cette solution permet d'apporter des modifications à nos différents fichiers et d'en apercevoir directement le résultat sur un téléphone mobile.

En revanche, puisqu'il s'agit d'une application utilisée en extérieur, l'ensemble de l'application a dû être stockée sur un ordinateur avec une adresse IP publique et un nom de domaine associé, permettant ainsi son utilisation depuis internet. Nous avons choisi de publier sur GitHub Pages³⁶, un service proposé par GitHub³⁷ qui permet de publier un répertoire existant directement sur le web. Le déploiement dure environ cinq minutes, ce qui justifie la première étape passant par Node.js pour les changements mineurs et les modifications apportées à l'interface utilisateur, ne nécessitant pas d'être sur place. L'application web est disponible à l'adresse https://theogerritsen.github.io/AR_project.

³⁴ <https://www.blender.org/> consulté le 15.01.2023

³⁵ <https://nodejs.org/en/> consulté le 03.01.2023

³⁶ <https://pages.github.com/> consulté le 03.01.2023

³⁷ <https://github.com/> consulté le 13.01.2023

8. Résultats

L'application finale se présente comme un sentier didactique en réalité augmentée, proposée sous forme de chasse au trésor. L'utilisateur a le choix de participer à l'expérience sous deux approches différentes : (1) en débloquant au fur et à mesure les étapes lorsqu'un jeu est terminé ou (2) en ayant un aperçu du sentier complet directement. L'application couvre trois étapes du GéoGuide Lausanne, numérotées de 1 à 3 (Fig. 18) (respectivement les étapes 9 à 11 du GéoGuide Lausanne), allant de la Cathédrale de Lausanne jusqu'à la rivière de la Louve, située sous la Place de la Riponne. Nous avons choisi ces trois étapes car elles permettent de proposer deux des trois types de réalité augmentée incluses dans AR.js, à savoir la réalité augmentée basée sur la localisation, utilisée dans les étapes 1 et 2, et la réalité augmentée basée sur la reconnaissance d'éléments naturels, utilisée dans l'étape 3.

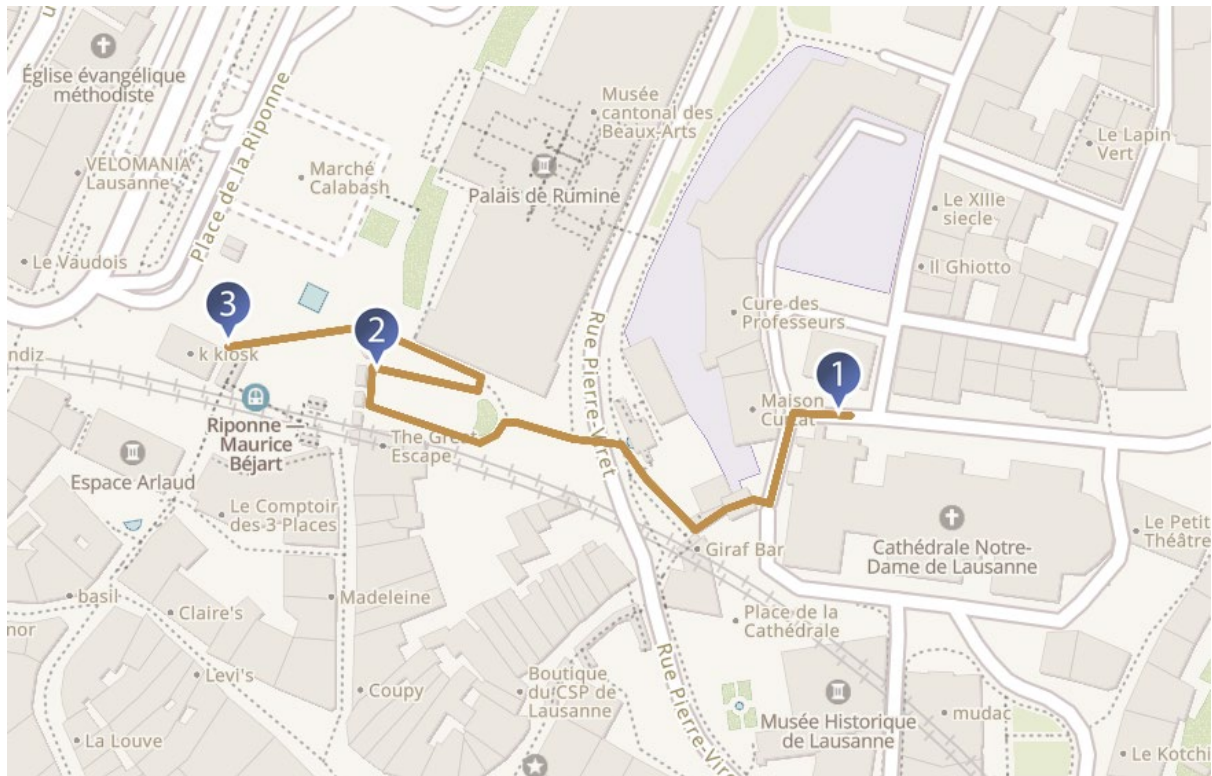


Figure 18, Sentier didactique

8.1 Contenu de l'application

8.1.1 Écran d'accueil

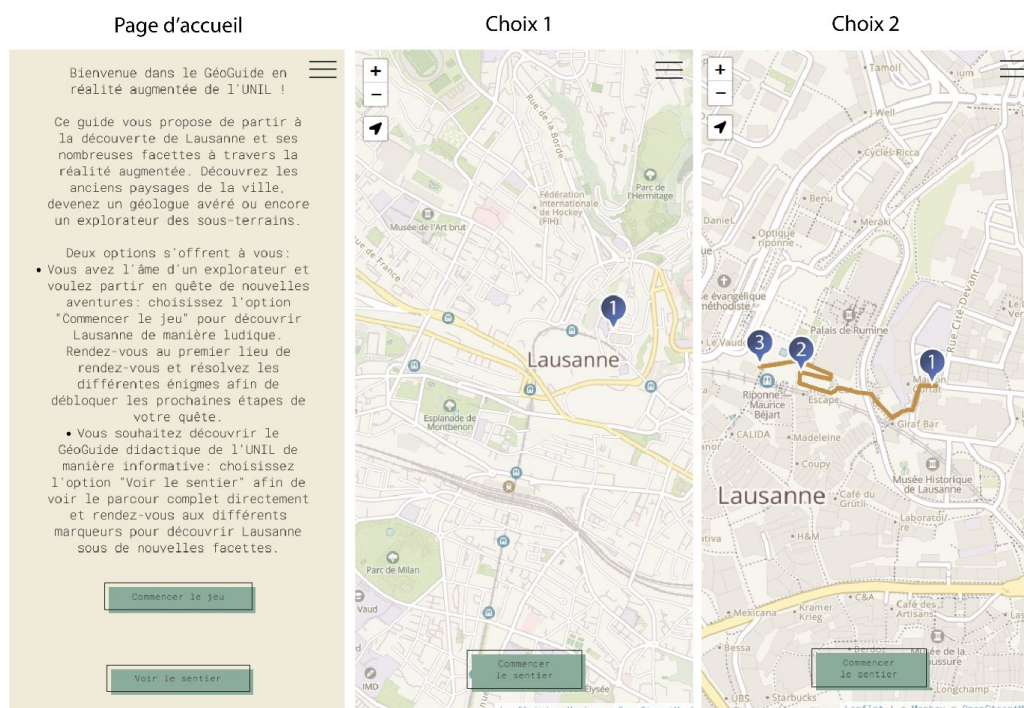


Figure 19, Écran d'accueil de l'application

Lors de l'ouverture de l'application, l'utilisateur est dirigé sur une page d'accueil qui l'informe à propos de son utilisation. Deux options s'offrent à lui :

- 1) Découvrir le sentier didactique sous la forme d'une « chasse au trésor » (Fig. 19, choix 1): si cette option est sélectionnée, seule la localisation de la première étape est affichée. L'utilisateur doit s'y rendre et compléter le premier jeu afin de débloquent la seconde étape.
- 2) Découvrir le sentier didactique à titre informatif (Fig. 19, choix 2): si cette option est sélectionnée, l'ensemble des étapes et du sentier est directement affiché à l'écran. Cela permet à l'utilisateur de ne pas forcément réaliser les étapes dans un ordre précis.

Un bouton « Commencer le sentier » permet de géolocaliser l'utilisateur et de zoomer sur sa position. La localisation en direct a été implémentée grâce à l'extension Leaflet.Locate³⁸, proposée par Dominik Moritz, qui récupère, avec une précision d'environ 35 mètres sous réseau 4G, les coordonnées de l'utilisateur. Sa position est mise à jour automatiquement toutes les secondes. L'application calcule ensuite la distance entre l'utilisateur et l'étape la plus proche. Les coordonnées (WGS 84) de l'utilisateur et de l'étape la plus proche sont croisées. À une latitude de 45° (qui correspond environ à la latitude de la Suisse, Lausanne se situant à 46.5205, 6.6331), une coordonnée avec trois décimales présente une précision de 78 mètres (OpenStreetMap Wiki³⁹). En croisant les coordonnées (en Degré - Décimal, DD) de l'utilisateur et des marqueurs, avec trois décimales, il est donc possible de calculer si l'utilisateur se trouve à moins de 78 mètres d'une étape. Si l'utilisateur se trouve à plus de 78 mètres d'une étape, l'application lui indique de s'approcher davantage de celle-ci (Fig. 20a). Si l'utilisateur se trouve à moins de 78 mètres d'une étape, un bouton « Commencer le tour en réalité augmentée » apparaît, permettant ainsi de commencer l'expérience en réalité augmentée (Fig. 20b).

³⁸ <https://github.com/domoritz/leaflet-locatecontrol> consulté le 13.01.2023

³⁹ https://wiki.openstreetmap.org/wiki/Precision_of_coordinates consulté le 15.01.2023

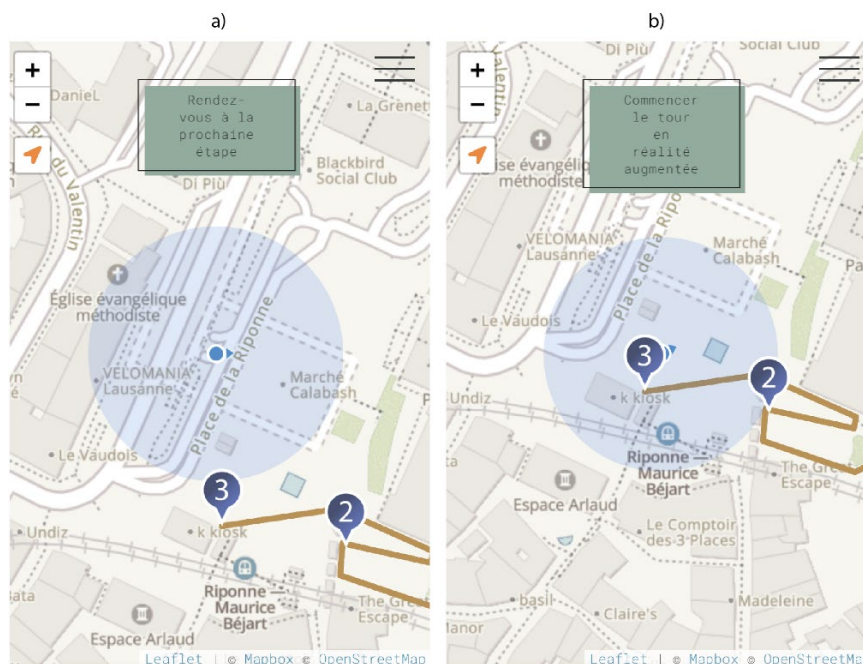


Figure 20, Boutons selon la proximité de l'utilisateur à une étape : (a) trop éloigné, (b) à proximité

Il est également possible d'appuyer sur le marqueur de l'étape que nous voulons afficher. Un pop-up apparaît alors, proposant deux options (Fig. 21):

- 1) Commencer le tour en réalité augmentée pour l'étape en question
- 2) Montrer les informations relatives à cette étape. Cette option redirige l'utilisateur vers le GéoGuide Lausanne et montre les informations concernant l'étape en question.

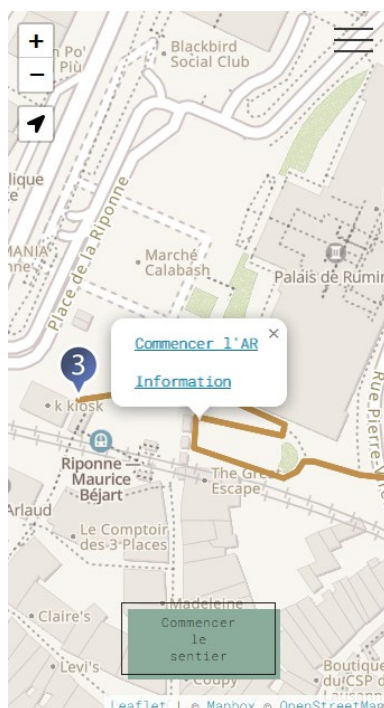


Figure 21, Pop-up permettant de commencer le tour en réalité augmentée ou d'afficher le GéoGuide Lausanne officiel

L'intégration du pop-up a été implémentée afin de répondre aux problèmes liés à la précision des coordonnées GPS sur lequel nous revenons dans la section 9.3. Si l'application ne détecte pas

correctement la localisation de l'utilisateur, il lui est quand même possible de participer à l'expérience en réalité augmentée s'il sait qu'il se situe au bon endroit.

Si le premier mode de jeu (« chasse au trésor ») est sélectionné, une variable globale à l'application, contenant le nombre d'étapes réalisées, se met à jour automatiquement. Elle permet ainsi d'afficher le bon nombre de marqueur lorsque l'utilisateur retourne à l'aperçu du sentier après la réalisation d'une étape.

8.1.2 Étape 1

La première étape se déroule autour de la Cathédrale de Lausanne. Les trois roches les plus utilisées dans les diverses constructions de Lausanne sont affichées en réalité augmentée (gneiss, calcaire et molasse) (Fig. 22). Ces roches ont été modélisées sur Blender. L'utilisateur est invité à viser chacune d'entre elles afin d'en apprendre davantage sur leur apparence et leur utilisation. La position d'un viseur, fixé au milieu de l'écran, est croisée avec la position des objets en réalité augmentée. Il est alors possible de détecter s'il y a eu une collision entre le viseur et un objet. Si c'est le cas, cela indique que l'utilisateur a visé une roche ; des informations relatives (images et texte) à cette roche sont affichées à l'écran.

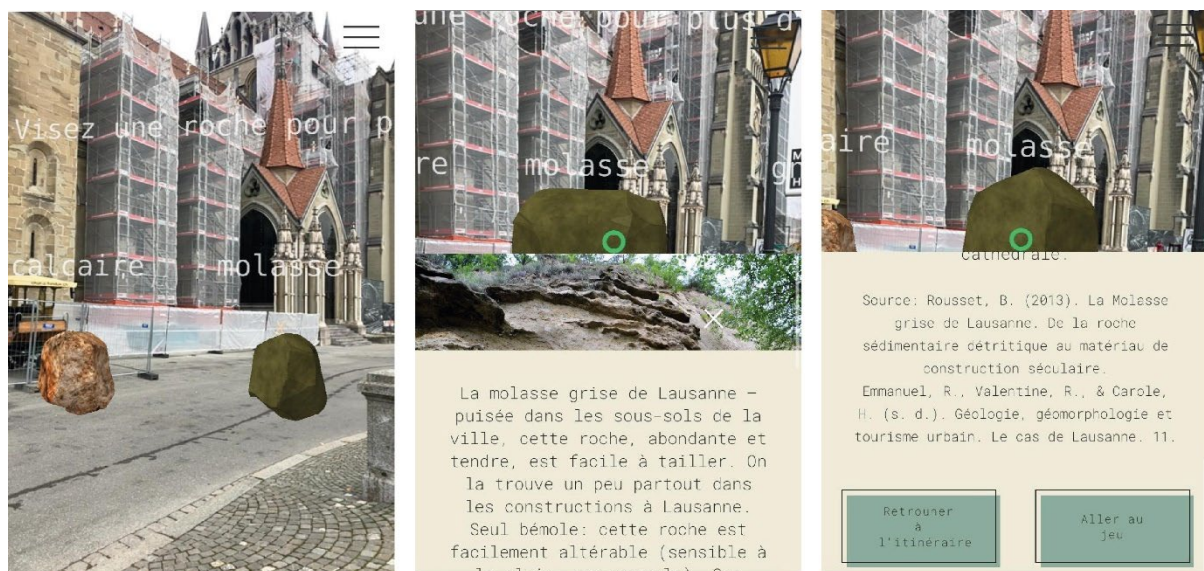


Figure 22, 1^{ère} étape

Une fois que l'utilisateur s'estime assez informé, il peut participer au premier jeu en réalité augmentée (Fig. 23). L'utilisateur est accueilli par des informations relatives au jeu (Fig. 23a). Il s'agit de repérer les trois marqueurs localisant les trois différents types de roche autour de la cathédrale (Fig. 23b). L'utilisateur doit alors s'approcher d'un des marqueurs. S'il se trouve à plus de 15m du marqueur, l'application lui indique de s'approcher davantage. Il doit alors identifier la roche indiquée par le marqueur et tenter de deviner le type de roche dont il s'agit, basé sur les connaissances acquises lors de la première phase de cette étape. Par le même processus de collision entre un viseur et un objet, il est possible de détecter quel marqueur l'utilisateur a visé. Lorsque l'utilisateur vise un marqueur, trois boutons apparaissent, correspondant aux trois choix de roche possibles (Fig. 23c). Il doit alors deviner de quelle roche il s'agit (Fig. 23d & Fig. 23e). Le jeu est complété une fois que les trois roches ont été correctement identifiées (Fig. 23f). Le score est indiqué en haut à gauche de l'écran et se met à jour à chaque bonne réponse. L'utilisateur est redirigé sur la carte présentant les marqueurs d'étape et invité à se diriger vers la deuxième étape.

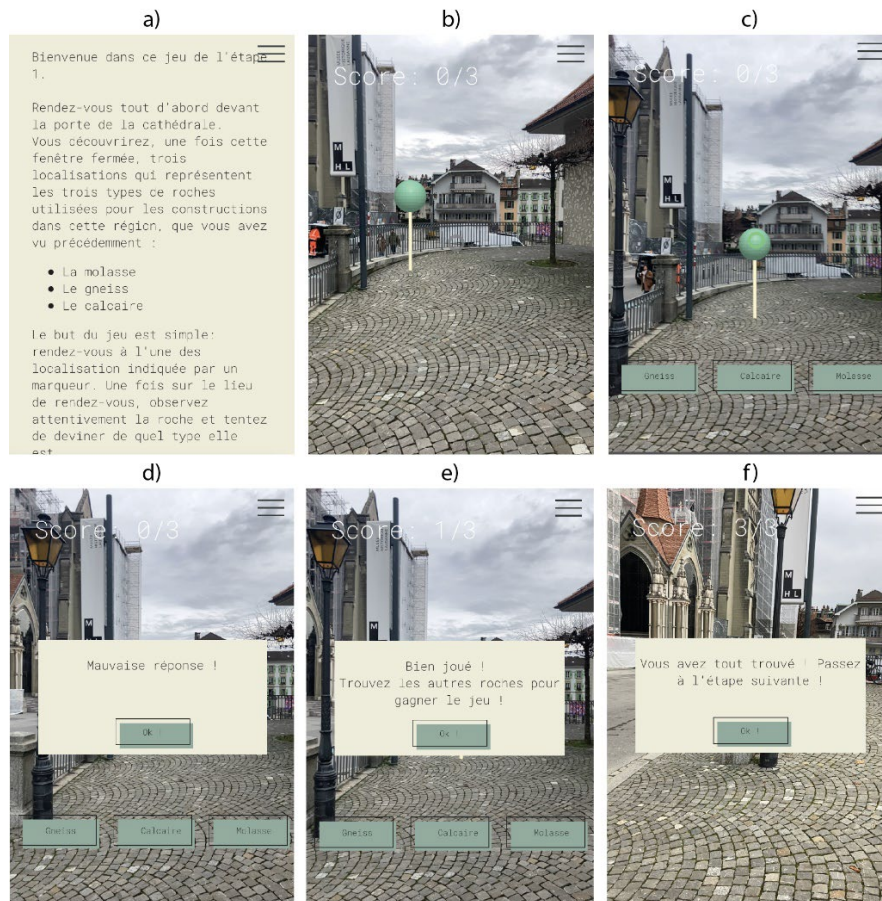


Figure 23, 1^{er} jeu : (a) Instructions du jeu, (b) Marqueur de localisation, (c) Boutons de choix, (d) Message pour mauvaise réponse, (e) Message pour bonne réponse, (f) Message si le jeu est complété

8.1.3 Étape 2

La deuxième étape se déroule sur la Place de la Madeleine qui surplombe la Place de la Riponne (Fig. 21). L'étape a été pensée comme une expérience informative en réalité augmentée plutôt qu'un jeu. L'application indique à l'utilisateur le lieu précis où se rendre (Fig. 24a). Il est alors invité à commencer l'expérience en réalité augmentée en utilisant la caméra de son smartphone afin de filmer la Place de la Riponne qui se trouve en dessous de lui. Des informations en réalité augmentée lui sont présentées :

- Des marqueurs indiquant la localisation de la Place de la Riponne, la Place Pépinet et la Grand Rue
- Un panneau informatif interactif permettant d'avoir de plus amples informations quant à l'environnement alentour. Lorsque le panneau est visé, un bouton apparaît permettant de changer les informations affichées (Fig. 24b)
- Un modèle en 3D de la Louve (la rivière passant sous la Place de la Riponne) montrant son tracé avant qu'il ne soit enterré (Fig. 24c)

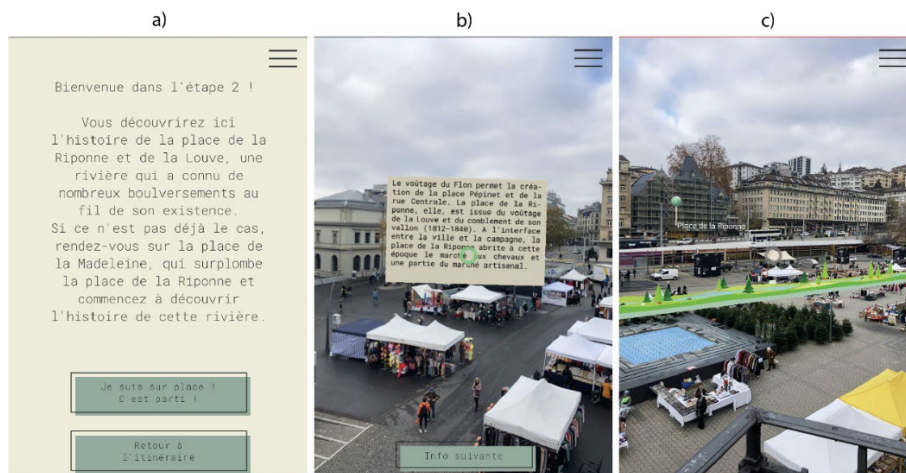


Figure 24, 2^{ème} étape : (a) Message d'accueil et instructions, (b) Panneau informatif interactif, (c) Modèle 3D de la Louve

8.1.4 Étape 3

La troisième étape se déroule dans l'abri de la Place de la Riponne donnant l'accès au parking souterrain. L'utilisateur est invité à faire une chasse au trésor qui le guidera jusqu'au regard qui observe la Louve circulant sous son voûtage. Une fois arrivé devant l'abri, une image est proposée à l'utilisateur qu'il devra ensuite trouver en explorant les différents étages de l'abri du parking souterrain. Une fois que ce dernier pense avoir trouvé la bonne image, il utilise sa caméra et scanne l'image en question (Fig. 25). Un algorithme SIFT est utilisé, intégré dans la librairie AR.js, afin de reconnaître l'image prise par l'utilisateur. Si elle correspond en effet à l'image qu'il doit trouver, une information concernant le voûtage de la Louve lui est donnée (Fig. 25a). Un viseur, comme à l'étape 1, est également présent. Il permet de calculer s'il y a collision entre le viseur et le panneau d'information (si l'utilisateur vise le panneau d'information). Si tel est le cas, un bouton « image suivante » apparaît. Si ce bouton est appuyé, le panneau d'information disparaît et laisse place à la prochaine image que l'utilisateur doit trouver (Fig. 25b). Son score est affiché dans le coin supérieur gauche de l'écran, lui indiquant le nombre d'images scannées par rapport au nombre total d'images à trouver. Une fois toutes les images correctement découvertes, l'utilisateur se trouvera devant le regard montrant la Louve souterraine, qui correspond à l'arrivée de la troisième étape.

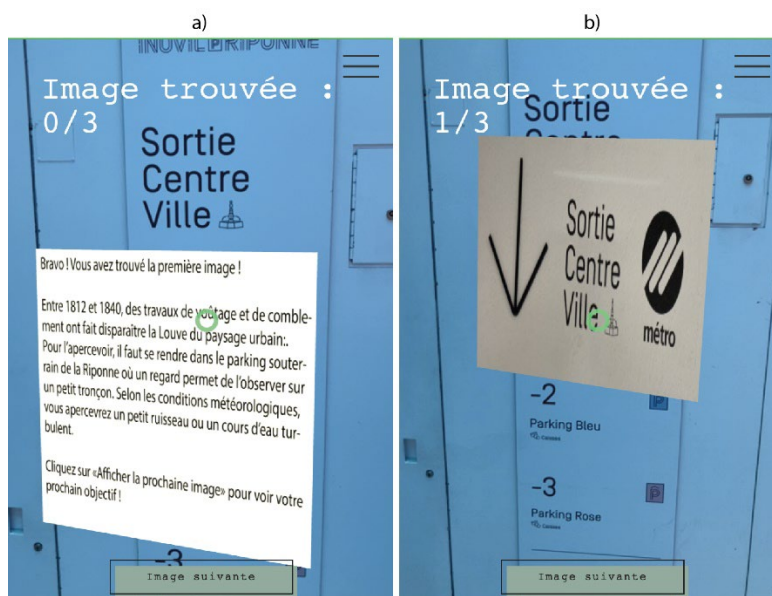


Figure 25, 3^{ème} étape : (a) Panneau informatif concernant la Louve si la bonne image est scannée, (b) Prochaine image à trouver lorsque l'utilisateur a cliqué sur « Image suivante »

8.1.5 Barre latérale de navigation

L'ensemble de l'application a été développée sous forme de jeu interactif en réalité augmentée. Elle a été pensée afin de faciliter au mieux la navigation des utilisateurs à travers les étapes. Chaque étape (y compris la page d'accueil) est agrémentée d'une brève explication de son fonctionnement. Une barre latérale de navigation (Fig. 26), présente à toutes les étapes, a également été implémentée. Elle a été développée afin de faciliter davantage la navigation et l'utilisation de l'application.

Elle présente donc plusieurs fonctionnalités :

- Affichage de l'étape à laquelle l'utilisateur se trouve
- Possibilité d'afficher à nouveau les informations relatives à l'étape
- Retour à la carte contenant le sentier et les marqueurs d'étape
- Retour à l'écran principal, permettant de choisir le mode de jeu
- Lien vers le GéoGuide Lausanne

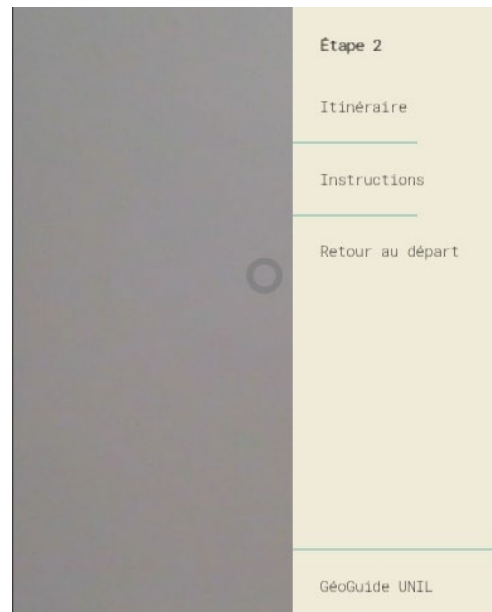


Figure 26, Barre latérale de navigation

8.2 Interface utilisateur

L'application a été développée pour l'ensemble des appareils mobiles. Elle peut, en théorie, être utilisée dans tous les navigateurs supportant WebGL, bien qu'AR.js présente quelques limites que nous discuterons dans le chapitre 9.1. L'interface graphique a été conçue afin de répondre aux différentes tailles et résolutions que présentent la variété de modèles de smartphone (sous Android ou iOS). L'adaptation du contenu selon la taille de l'écran de l'utilisateur a été réalisée en utilisant des unités relatives dans les différents fichiers CSS. CSS3 propose des unités de base permettant de récupérer la taille de l'écran de l'appareil qui utilise l'application. Il est ainsi aisé d'adapter le contenu à ce dernier en calculant la taille de manière relative plutôt qu'absolue. Les unités en question, utilisées dans le cadre de notre application sont :

- %, pourcentage relatif à l'élément parent
- vw, relatif au 1% de la largeur de la fenêtre du navigateur
- vh, relatif au 1% de la hauteur de la fenêtre du navigateur
- em, employé pour les polices, relatif à la taille de la police de l'élément, 16pt par défaut

La réactivité de l'application (c'est-à-dire la manière dont l'interface graphique s'adapte selon l'appareil) a été testée sur l'ensemble des formats de téléphones mobiles proposés par les outils de développement de Firefox et de Google Chrome (iPhone X à 13 (+SE) sur iOS 14.6, iPad sur iPadOS 14.7.1, Galaxy S8+ / S10 / S20 / Note20 / Fold sur Android) et présente des résultats satisfaisants sur l'intégralité de ces appareils.

9. Discussion

9.1 Performances de l'application

Plusieurs facteurs peuvent agir sur les performances de l'application et ainsi influencer l'expérience globale de l'utilisateur. Nous discutons ici, démontrant ainsi partiellement notre deuxième hypothèse et répondant à notre quatrième question de recherche, des performances observées lors du développement et de l'utilisation de notre application. L'interface réalisée dans le cadre de notre application peut être séparée en deux catégories distinctes, à savoir l'interface de base (sans réalité augmentée) et l'expérience en réalité augmentée.

L'application fonctionne sur l'ensemble des smartphones testés (iPhone SE 2020 (iOS 14), iPhone SE 2022 (iOS 16), Xiaomi Redmi 9A (Android) et Huawei). Bien que la réalité augmentée sur le web prône la compatibilité aisée entre systèmes d'exploitation et navigateurs (à condition qu'ils supportent WebGL et WebRTC), les tests que nous avons effectués relèvent le contraire. Il est impossible d'utiliser l'application sur iOS avec un navigateur fondé sur Chromium. Le système iOS empêche l'apparition des fenêtres de permissions permettant à l'application d'accéder aux différents capteurs (caméra, GPS, etc.). Safari et XRViewer permettent en revanche une expérience fluide sur iOS. Les navigateurs fondés sur Chromium fonctionnent sur Android.

Premièrement, l'interface graphique de base a été entièrement réalisée en HTML5, CSS3 et JavaScript. L'interface de base a été développée pour agir directement comme une application web pour téléphone mobile. Chaque fonctionnalité a donc été préalablement testée et ajustée pour les appareils mobiles. L'interface a été conçue afin d'être facile d'utilisation. Les fonctionnalités interactives relatives aux boutons ont également été conçues pour réagir aux écrans tactiles. Un problème majeur de l'interface de base est le calcul de la position GPS de l'utilisateur. Celui-ci est réalisé lorsque l'utilisateur se déplace le long du sentier ainsi que lorsque l'application propose de la réalité augmentée basée sur la localisation. Dans le premier cas, le plugin utilisé indique une précision d'environ 35 mètres avec une mise à jour faite toutes les secondes. Cette précision se remarque lors de la navigation en utilisant la carte, avec la localisation de l'utilisateur qui fluctue fortement et saute de lieu en lieu, à chaque mise à jour de sa position. Cela influence fortement le calcul du croisement des positions de l'utilisateur et des différentes étapes. Il est possible que l'utilisateur se trouve précisément à l'endroit indiqué, mais qu'il ne puisse pas commencer le tour en réalité augmentée car l'application ne détecte pas correctement sa position.

Ensuite, les deux types de réalité augmentée implémentées dans notre application (basée sur la localisation et basée sur la reconnaissance d'éléments naturels) montrent des résultats satisfaisants d'une manière générale, mais posent de nombreux problèmes de performances, réduisant ainsi la qualité de l'expérience de l'utilisateur.

1) Réalité augmentée basée sur la localisation

Dans le cas de la réalité augmentée basée sur la localisation, et similairement aux problèmes des coordonnées GPS expliqués précédemment, la précision et la mise à jour constante de la position de l'utilisateur posent plusieurs problèmes. La position des marqueurs du premier jeu sont fixés selon une latitude et longitude donnée. La position (latitude / longitude) de l'utilisateur calculée influencera donc l'endroit où les marqueurs apparaîtront. Chaque mise à jour de sa position entraînera un repositionnement des marqueurs relativement à la position de l'utilisateur, causant un saut des marqueurs à chaque mise à jour. Ce saut a été réduit en indiquant un facteur de lissage entre chaque saut. Il n'en reste pas moins que les marqueurs et les objets virtuels sont positionnés différemment à chaque utilisation, puisque la position de base de l'utilisateur est prise en compte. Cela ne pose pas de problèmes pour des objets lointains. Les éléments en réalité augmentée de l'étape 2, se situant à plusieurs dizaines de mètres de l'utilisateur, ne sont que peu influencés par ce

phénomène. En revanche, les marqueurs de l'étape 1 sont davantage impactés, rendant la position précise des roches à localiser difficile.

Le système GPS est le seul capteur qui semble poser des problèmes quant à l'expérience en réalité augmentée. Les objets répondent bien aux différents changements directionnels de l'utilisateur.

2) Réalité augmentée basée sur la reconnaissance d'éléments naturels

Les performances de l'*image tracking AR* varient énormément selon plusieurs facteurs. Premièrement, la qualité de l'image donnée en entrée au programme NFT-Marker-Creator joue un rôle crucial pour la détection de points clés pertinents. La densité de points par pouce (DPI) analysée influencera la stabilité de l'image superposée. Une image avec une résolution élevée ainsi qu'un DPI élevé (300 DPI) donnera les meilleurs résultats, mais les points clés prendront beaucoup plus de temps à être définis. Il est également essentiel de choisir une image possédant des éléments contrastés, la luminosité pouvant varier sur le terrain. Deuxièmement, la qualité de la caméra du téléphone mobile influence également la stabilité de l'image superposée. Une qualité plus grande induit une meilleure stabilité.

L'application montre des résultats satisfaisant lors de l'*image tracking AR*. L'ensemble de l'étape qui utilise ce type de réalité augmentée se déroule à l'intérieur, limitant ainsi les variations de luminosité. L'application reconnaît les images données en entrée. En revanche, les images superposées souffrent d'une stabilité faible en raison de la reconnaissance des points clés. L'algorithme fonctionne mieux pour une image délimitée par un cadre. Puisque ce n'est pas le cas ici, il faut parfois faire varier la distance entre le smartphone et l'image afin que l'algorithme fonctionne.

Finalement, il a été démontré que l'utilisation de la réalité augmentée sur le web engendre une utilisation excessive des composants du téléphone mobile. Nous concluons ce chapitre en analysant le taux d'utilisation du processeur, de la mémoire vive et du réseau lors de l'utilisation de notre application. Cette étape a été réalisée avec l'application Lirium Info pour iOS, qui permet de monitorer l'utilisation des différents composants d'un smartphone.

La figure 24a montre l'utilisation du processeur (CPU, Central Processing Unit) lors de l'ouverture de l'application et de l'utilisation de la réalité augmentée basée sur la localisation. Nous pouvons observer une utilisation prononcée du CPU lors de la première ouverture et du chargement des diverses fonctions JavaScript relatives à la carte et la localisation de l'utilisateur (Fig. 27a, 1^{er} pic à 100% d'utilisation). L'utilisation se stabilise ensuite jusqu'au chargement de la première expérience en réalité augmentée (Fig. 27a, 2^{ème} pic à 60% d'utilisation). Une fois les éléments chargés et affichés à l'écran, la consommation reste plutôt faible.

Les figures 24b et 24c montrent l'utilisation du processeur, de la mémoire vive et du réseau lors d'une expérience en réalité augmentée basée sur la reconnaissance d'éléments naturels. Comme énoncé précédemment, les algorithmes de reconnaissance tendent à être extrêmement énergivores. Chaque pic correspond à une reconnaissance d'image. L'application doit sans arrêt recalculer la localisation et l'orientation des points clés afin de superposer correctement l'objet virtuel à l'image de base. Les capacités du processeur sont utilisées à 100% pour chaque reconnaissance. Les fichiers FSET qui contiennent les points clés des images à reconnaître sont hébergés sur un site internet externe. Nous pouvons donc observer que le réseau est fortement sollicité au début de chaque reconnaissance.

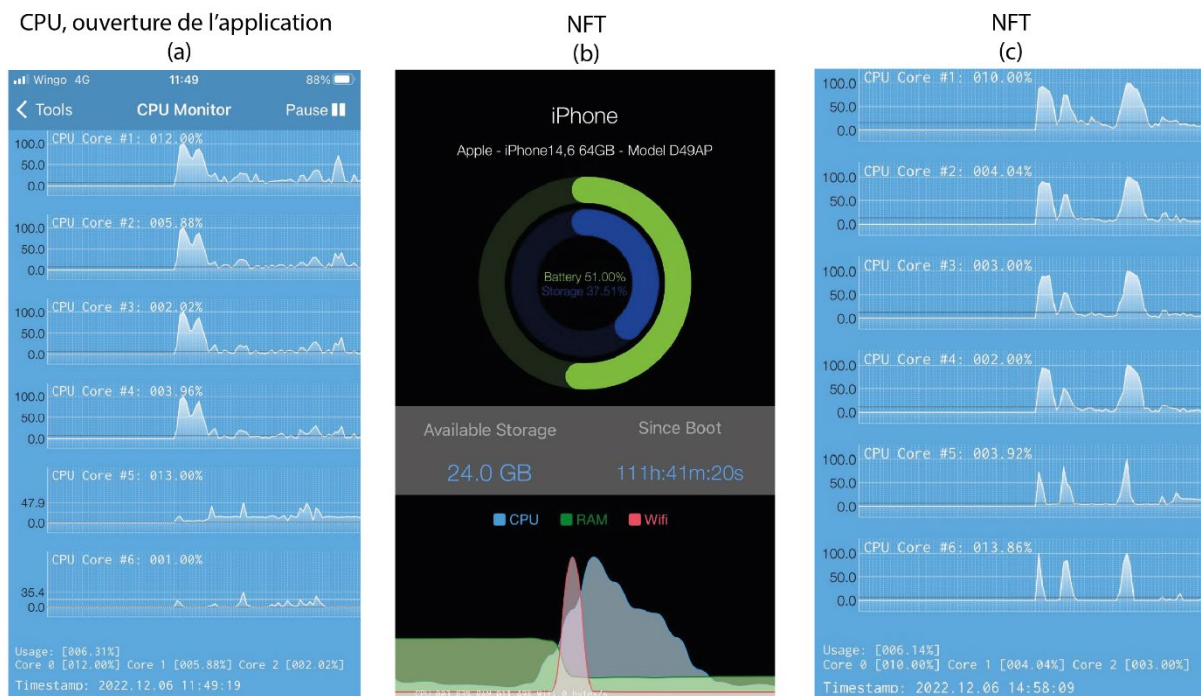


Figure 27, (a) utilisation du CPU à l'ouverture de l'application, (b) utilisation du CPU, de la mémoire vive et du wifi lors de la reconnaissance d'éléments naturels (NFT), (c) utilisation du CPU lors d'un NFT

9.2 La réalité augmentée web mobile dans le tourisme, le géotourisme et l'éducation

Nous revenons ici sur l'apport de la réalité augmentée dans les domaines du tourisme, du géotourisme et de l'éducation. Nous traiterons plus particulièrement de l'utilisation de la réalité augmentée web mobile appliquée à ceux-ci, afin de notamment répondre à notre cinquième question de recherche et notre première hypothèse concernant l'apport de la réalité augmentée dans ces différents domaines.

En premier lieu, nous avons effectivement vu qu'il est possible pour ces domaines de tirer de réels bénéfices de la réalité augmentée. Elle représente un outil ludique, facile d'utilisation et accessible à tout le monde possédant un smartphone. En effet, l'unique fait d'être une application « web mobile » représente de nombreux avantages. Tout d'abord, l'application utilise un objet du quotidien de la majeure partie de la population : un smartphone. Il n'est donc pas nécessaire de passer par l'achat (pour l'éducation) de matériel spécifique à la réalité augmentée, la plupart des écoles étant aujourd'hui équipée de tablettes. Venant aux touristes, ils sont donc capables de vivre une expérience en réalité augmentée sans louer du matériel supplémentaire ou en étant dépendant d'une agence touristique ou d'un guide local. Ensuite, elle ne requiert aucun téléchargement au préalable. Un simple navigateur, pour autant qu'il supporte les API nécessaires au fonctionnement de la réalité augmentée, suffit afin d'utiliser l'application web.

9.2.1 La réalité augmentée et l'éducation

En termes d'éducation, la réalité augmentée offre de nouvelles opportunités d'acquisition de connaissances et représente un outil d'apprentissage puissant. Les méthodes d'enseignement évoluent parallèlement aux nouvelles technologies disponibles, et l'utilisation de matériel informatique est de plus en plus courante et proposée très tôt dans les cursus scolaires. Il est essentiel pour les étudiants d'être formés sur les nouvelles technologies, puisqu'ils y seront sans doute confrontés tout au long de leur vie. Avec l'essor de la réalité augmentée, il ne fait aucun doute qu'elle intégrera peu à peu le quotidien des gens. Il a également été démontré que la réalité augmentée permet un apprentissage simplifié pour certaines branches spécifiques. Tout d'abord, la possibilité de visualiser et de manipuler des phénomènes complexes facilite grandement leur compréhension. L'apprentissage de certains

concepts théoriques, difficiles à se représenter par la lecture ou par des images statiques, devient interactif, immersif et ludique. Cette méthode d'apprentissage encourage l'implication et l'attention des étudiants et facilite donc l'acquisition de certaines connaissances. Utiliser la réalité augmentée dans le cadre de l'enseignement montre donc un double avantage : une attention et une compréhension accrue des étudiants, mais aussi une formation essentielle sur les nouvelles technologies qui feront sans doute partie intégrante de leur vie à l'avenir.

9.2.2 La réalité augmentée appliquée au tourisme et au géotourisme

Nous avons également vu que la réalité augmentée offre de nombreuses opportunités pour les domaines du tourisme et du géotourisme, autant pour les utilisateurs que pour les compagnies productrices. Les touristes cherchent aujourd'hui à visiter des lieux mémorables, qui sortent des chemins battus, à vivre des expériences inédites et originales. La réalité augmentée converge vers l'ensemble de ces points par le biais de plusieurs facteurs. L'immersion que propose la réalité augmentée permet tout d'abord aux touristes de vivre des expériences uniques et personnalisées, notamment selon le processus de co-création. Il est alors possible pour les compagnies productrices de se rapprocher au mieux des attentes des touristes et de proposer un contenu en fonction leurs attentes. La superposition de contenu en réalité augmentée par-dessus le monde réel permet également de déceler des informations supplémentaires (historiques, géologiques, géomorphologiques, etc...) concernant un lieu. Il n'est donc pas nécessaire de passer par du matériel physique spécifique à la visite d'un lieu, l'ensemble de l'expérience se passant directement sur leur téléphone mobile. La réalité augmentée peut également être utilisée comme moyen de navigation, chose que nous n'avons pas développé ici. Elle peut guider les touristes le long de sentiers de manière continue, à l'aide de flèches directionnelles ou d'un chemin en réalité augmentée indiquant les directions à prendre. Elle réduit ainsi la probabilité pour les touristes de se perdre et leur offre la possibilité d'explorer des zones inconnues. Dans le cas de cette étude, le géotourisme en milieu urbain est tout particulièrement concerné. La réalité augmentée se prête spécialement bien pour cette branche du géotourisme puisque l'urbanisation des villes a aujourd'hui complètement caché les différents processus qui ont formé le paysage. Il est alors possible de montrer aux touristes des processus géologiques et géomorphologiques complexes, présents sous leurs pieds, en les superposant au monde réel. Dans notre cas, nous avons eu l'occasion d'informer les utilisateurs quant aux matériaux de construction locaux utilisés dans les constructions de Lausanne et de montrer, en partie, l'ancien tracé de la Louve avant son comblement. Il est alors plus facile pour l'utilisateur de se représenter ce à quoi ressemblait la ville autrefois. Il aurait été intéressant toutefois de réitérer l'expérience en montrant les formes héritées de l'ancien paysage glaciaire (moraines) à Lausanne, puisque ces processus ont grandement influencé la morphologie de la ville.

9.3 Limites et perspectives de la réalité augmentée mobile sur le web

Nous discuterons ici, afin notamment de traiter notre sixième question de recherche et de compléter la confirmation de notre deuxième hypothèse, des différentes limites et perspectives que présente l'implémentation de la réalité augmentée sur le web, plus précisément pour l'utilisation sur téléphone mobile. Nous commencerons par expliquer les limites techniques et technologiques utilisées lors du développement et des divers tests du produit terminé. Nous concluons en mettant en avant les perspectives qu'offrent les technologies émergentes actuelles, à savoir le transfert des tâches computationnellement lourdes sur le Cloud et l'évolution permanente des smartphones.

9.3.1 Limites technologiques

Comme élucidé dans les chapitres précédents, la réalité augmentée mobile sur le web souffre encore de nombreux problèmes liés aux limites technologiques des navigateurs, des librairies utilisées et des téléphones mobiles. En premier lieu, la consommation énergétique est réellement excessive. La réalité augmentée demande l'utilisation de l'ensemble des capteurs (GPS, caméra, gyroscope, accéléromètre) intégrés dans les téléphones mobiles modernes. De plus, la consommation dépend du type de réalité augmentée utilisé ainsi que le contenu affiché. Nous avons démontré que la réalité augmentée basée sur la localisation n'est pas extrêmement énergivore. Seul le placement des entités au moment du

chargement de l'expérience montre un pic d'utilisation du processeur. La répétition du calcul de leur position selon le mouvement de l'utilisateur demande assez peu de ressources. En limitant donc le nombre d'entité, il est possible d'introduire une expérience en réalité augmentée limitée en termes d'énergie utilisée. En revanche, l'utilisation d'algorithmes lourds (ici SIFT) induit une consommation beaucoup plus importante. L'application doit tout d'abord faire appel à des fichiers qui ne sont pas stockés dans l'architecture principale de l'application, demandant ainsi des ressources en réseau importantes. L'algorithme doit ensuite reconnaître les éléments naturels captés par la caméra, croisant en continu les points clés avec les images données en entrée. Une fois les points clés détectés, leur position est sans cesse recalculée afin de superposer correctement l'objet virtuel. De plus, l'affichage de contenu en 3D, ainsi qu'une possible interactivité avec ce contenu, génère de lourdes tâches computationnelles pour le processeur du smartphone. Ces diverses sollicitations engendrent une consommation énergétique excessive, menant à une perte rapide du niveau de batterie.

Les lourdes tâches computationnelles restent un problème général concernant l'ensemble des technologies proposant de la réalité augmentée, mais certaines limites présentées dans le cadre de cette étude sont spécifiques à la librairie utilisée ici (AR.js). Bien qu'étant une librairie open source, son répertoire GitHub n'a pas été modifié depuis plus de trois ans. Certains problèmes que les développeurs rencontrent aujourd'hui n'ont donc pas été traités. AR.js utilise un algorithme de reconnaissance d'image extrêmement énergivore car cette librairie est basée sur ARToolKit, qui a été développée en 1999. Bien que performante, elle n'entre pas en concurrence avec les nouvelles technologies émergentes, notamment les algorithmes utilisant les réseaux de neurones. AR.js permet une approche aisée à la réalité augmentée sur le web mais montre des résultats relativement peu satisfaisants au niveau des performances pour les deux approches de la réalité augmentée utilisées ici. Elle a été développée, à la base, pour la réalité augmentée basée sur la reconnaissance de marqueurs ; une approche que nous n'avons pas traitée dans le cadre de cette étude. Il ne fait nul doute qu'il serait possible d'obtenir de meilleures performances avec d'autres bibliothèques plus récentes, mises à jour régulièrement, ou en développant une application native.

Contrairement à ce que la réalité augmentée sur le web prône, à savoir une compatibilité entre systèmes d'exploitation, les résultats énoncés dans cette étude démontrent en partie le contraire, bien que ce problème soit spécifique à la librairie AR.js utilisée. Le développement d'une application web permet en effet d'unifier les langages utilisés, les limitant à HTML5, CSS3 et JavaScript. Il n'est donc pas nécessaire de passer par un SDK (Kit de Développement Logiciel) spécifique au système d'exploitation et cette solution limite grandement les coûts qu'engendrerait la publication d'une application native sur les plateformes relatives (PlayStore et AppStore). Or, les résultats montrent que les navigateurs actuels souffrent encore d'un problème de compatibilité, selon le système d'exploitation utilisé, mais également selon son support des API spécifiques à la réalité augmentée sur le web. Certains efforts sont développés afin de construire un navigateur permettant les expériences en réalité mixte (augmentée et virtuelle) à travers les diverses plateformes.

9.3.2 Limites dans les domaines de l'éducation, du tourisme et du géotourisme

Finalement, lorsque nous nous concentrons sur les domaines du tourisme et du géotourisme, nous pouvons mettre en avant plusieurs problèmes liés aux performances d'une application web mobile en réalité augmentée. Tout d'abord, la forte consommation de batterie du smartphone est un souci majeur. Les téléphones mobiles représentent un outil essentiel pour les touristes en termes de navigation. Une application drainant de manière excessive la batterie n'a rien d'attrayant et peut fortement influencer l'appréciation de l'utilisateur. De plus, les attentes des utilisateurs concernant les performances d'une application sont aujourd'hui extrêmement élevées. Une simple latence, un délai de chargement ou une imprécision de quelques mètres de la localisation de l'utilisateur ou des points d'intérêt dégraderaient fortement l'expérience touristique. Nous avons vu que même dans le cadre d'une expérience en réalité augmentée se déroulant en ville, où la connexion internet est souvent à son maximum, de nombreuses imprécisions sont notables. La localisation des points d'intérêt est souvent imprécise, et la

reconnaissance d'image reste assez bancal. Il est primordial de noter que ces faiblesses sont spécifiques à la librairie utilisée ici, et ne sont en aucun cas générales à la réalité augmentée. Il n'en reste que la qualité de l'expérience se voit détériorée et l'application ne montre pas forcément un résultat satisfaisant en tant que produit touristique. De plus, l'utilisation de la réalité augmentée web mobile dans des lieux à faible connexion internet, en pleine montagne ou dans des endroits reculés par exemple, engendrerait des résultats encore moins satisfaisants. Concernant les problèmes de compatibilité entre navigateurs, cela représente également une limite principale pour les domaines du tourisme et du géotourisme. La plupart des gens utilisent le navigateur qui leur convient le mieux. Si celui-ci ne supporte pas les API nécessaires au bon fonctionnement de la réalité augmentée sur le web, il serait extrêmement délicat de leur demander de télécharger un autre navigateur afin de vivre l'expérience. Il est arrangeant que les deux navigateurs permettant d'utiliser AR.js soient ceux natifs aux deux différents systèmes d'exploitation (Safari pour iOS et Chrome pour Android).

9.3.3 Perspectives d'avenir pour la réalité augmentée mobile sur le web

Le développement d'une application web en réalité augmentée représente aujourd'hui encore de nombreux défis, comme vu dans le chapitre précédant, que des technologies émergentes tentent peu à peu de résoudre. Nous répondons ici à notre septième question de recherche en mettant en avant les différentes perspectives d'avenir qu'offre l'évolution de la technologie.

L'émergence des diverses librairies qui permettent de développer des applications web en réalité augmentée ou en réalité virtuelle offrent aujourd'hui aux développeurs un moyen efficace et aisé de produire de telles technologies. Elles permettent, en premier lieu, de ne plus devoir passer par des plateformes de développement spécifiques aux systèmes d'exploitation. Il est dorénavant possible de créer du contenu augmenté en quelques lignes de code, sans devoir se soucier des diverses API requises pour l'affichage d'un tel contenu sur un navigateur. Nous avons exploré ici les possibilités de la librairie AR.js. Cette librairie tente de simplifier au maximum l'implémentation de la réalité augmentée en utilisant d'autres librairies et environnement de développement spécifiques. Il en résulte qu'il est possible, en quelques lignes de code, d'obtenir une expérience en réalité augmentée. Il existe, bien entendu, d'autres outils permettant l'implémentation de réalité augmentée, certains passant par une interface graphique, ne requérant aucune notion de programmation. Il est donc évident que les efforts convergent aujourd'hui vers une accessibilité facilitée de cette technologie, permettant ainsi une émancipation accélérée.

Ensuite, le transfert des tâches computationnellement lourdes sur le Cloud, ou la plus récente technologie de l'*edge computing*, semblent présenter les meilleures perspectives en termes de performances pour la réalité augmentée web sur téléphone mobile. Bien que ces solutions présentaient encore certaines limites en raison du temps de latence élevé lors de transfert de calcul, l'émergence progressive du réseau 5G permettrait d'implémenter ce genre de technologie tout en gardant une faible latence et en augmentant les performances des différents algorithmes utilisés en réalité augmentée. En raison de l'acceptation lente de ce réseau 5G, les solutions hybrides semblent aujourd'hui être les plus adéquates et les plus performantes. Les tâches computationnelles légères peuvent alors être effectuées directement sur le téléphone mobile, alors que les plus lourdes sont transférées sur une machine virtuelle distante. Dans le cas de notre application, il serait judicieux de garder une approche *front-end* pour les expériences en réalité augmentée basée sur la localisation, puisqu'elles sont peu énergivores. En revanche, il serait approprié de transférer les calculs liés à la réalité augmentée basée sur la reconnaissance d'éléments naturels sur le Cloud, réduisant ainsi les lourdes tâches computationnelles du téléphone mobile. Nous avons également mis en avant que des algorithmes plus efficaces existaient aujourd'hui pour la reconnaissance d'éléments naturels.

Finalement, nous voyons aujourd'hui une volonté de convergence des navigateurs vers une intégration native des expériences en réalité augmentée et en réalité virtuelle. La réalité augmentée sur le web repose encore aujourd'hui sur le support des API WebGL et WebRTC des navigateurs. Or, les étapes

de développement d'une expérience augmentée, ainsi que son intégration sur le web, dépendent entièrement de la plateforme utilisée (ARKit, ARCore, AR.js, etc.). Le développement du WebXR Device API par le Immersive Web Community Group vise à corriger les problèmes de compatibilité en proposant une plateforme unifiée et universelle pour l'ensemble des navigateurs et systèmes d'exploitation. Mais l'intégration d'une troisième dimension au web, ainsi que les manipulations qui lui sont liées, n'est pas chose aisée. Il faudra encore un certain temps avant que ces technologies puissent être utilisées de manière fluide, mais l'intérêt grandissant pour la réalité mixte par les diverses industries convergeront sans doute les efforts pour son intégration universelle.

En conclusion à ce chapitre, et afin de reprendre nos deux dernières hypothèses (2 & 3) portant sur l'efficacité des smartphones en tant que medium pour la réalité augmentée sur le web, ainsi que notre huitième question de recherche, il nous est possible d'affirmer que les smartphones sont un moyen efficace d'utiliser la réalité augmentée, mais ils souffrent encore de grands problèmes de performances. D'une part, le fait qu'un nombre extrêmement élevé de personnes possèdent un smartphone le rend le medium idéal. En effet, il n'est pas nécessaire de posséder un dispositif informatique spécifique à la réalité augmentée, ce qui est le cas en revanche pour la réalité virtuelle. Son accessibilité est donc grandement facilitée et à la portée de toutes et tous. D'autre part, les énormes coûts computationnels de la réalité augmentée, en particulier pour les algorithmes de reconnaissance d'éléments naturels, couplé avec la faible précision du placement des objets géoréférencés, diminuent la fluidité de l'expérience de l'utilisateur. Les perspectives qu'offrent le réseau 5G et les solutions hybrides élucidées précédemment pourraient, en revanche, atténuer ce problème et assurer un futur prometteur pour la réalité augmentée mobile sur le web.

10. Conclusion

La réalité augmentée, bien que n'étant pas une technologie récente, connaît aujourd'hui un intérêt grandissant. Deux facteurs principaux en sont la cause : la possibilité d'utiliser un objet aussi accessible qu'un téléphone mobile pour vivre une telle expérience, et la volonté de transférer cette technologie directement sur le web, supprimant ainsi les inconvénients relatifs aux applications natives.

Le téléphone mobile est apparu, au fil de cette étude, comme le candidat idéal en tant que médium pour la réalité augmentée. Il s'agit d'un objet largement répandu, que la plupart des gens possèdent. Proposer une expérience en réalité augmentée sur téléphone mobile permet donc de toucher un nombre élevé de personnes, sans engendrer de coûts supplémentaires en lien avec du matériel spécifique à la réalité augmentée. Les industries du tourisme et de l'éducation, ainsi que la branche du géotourisme, peuvent tirer de nets avantages de la réalité augmentée mobile sur le web. Elle représente une technologie attrayante, accessible à tous et reste instinctive pour la plupart des utilisateurs. La possibilité d'interagir avec l'environnement facilite la compréhension de certains processus complexes et offre, autant dans l'éducation que dans le tourisme, une bonne manière d'assimiler certaines informations. Les étudiants disposent alors d'un outil d'apprentissage novateur et ludique, et les touristes peuvent vivre des expériences plus mémorables et uniques. Ensuite, le développement facilité d'une application en réalité augmentée directement sur le web a largement participé à son émancipation. Les langages informatiques répandus que sont HTML5, CSS3 et JavaScript, permettent à un nombre important de développeurs d'accéder à cette technologie et de l'intégrer dans leurs projets. Finalement, il n'est pas nécessaire pour les utilisateurs de passer par le téléchargement d'une application native, mais uniquement de rentrer une URL prédéfinie dans leur navigateur. Ceci représente un argument de taille, les utilisateurs étant de moins en moins enclins à télécharger du nouveau contenu sur leur téléphone mobile, surtout s'il ne sera utilisé que temporairement.

La réalité augmentée web mobile souffre encore aujourd'hui de nombreux problèmes en lien avec la qualité du matériel informatique intégré aux smartphones, les lourdes tâches computationnelles et les bibliothèques utilisées. Les composants des smartphones ne cessent d'évoluer, mais l'usage continu de l'ensemble des capteurs intégrés, ainsi que l'utilisation d'algorithmes complexes, engendrent une forte consommation de la batterie. Appliqué au domaine du tourisme, il est délicat de proposer une activité qui drainerait si rapidement la batterie des utilisateurs. De plus, comme nous l'avons vu, l'expérience en réalité augmentée proposée dans le cadre de cette étude montre de nombreuses imprécisions, que ce soit avec la réalité augmentée basée sur la localisation ou la réalité augmentée basée sur la reconnaissance d'images. Avec les attentes en performances que les gens demandent aujourd'hui, de telles imprécisions dégradent fortement l'expérience des utilisateurs. Il est évident que certains problèmes rencontrés ici sont spécifiques à la bibliothèque utilisée (AR.js), notamment la géolocalisation approximative, mais les coûts computationnels importants peuvent être appliqués à l'ensemble des technologies utilisant la réalité augmentée.

La réalité augmentée évolue aujourd'hui rapidement et montre un réel potentiel en lien avec la manière dont nous interagissons avec le monde digital. Cette technologie a su trouver sa place comme un outil puissant dans diverses industries. Avec un support grandissant de la réalité augmentée par les navigateurs, il est probable que de plus en plus de sites web proposeront un tel contenu dans un futur proche. L'évolution des outils permettant de développer du contenu en réalité augmentée et l'acceptation progressive du réseau 5G faciliteront d'autant plus son accessibilité pour les industries et pour les utilisateurs. Les performances qu'offre *l'edge computing* influenceront également grandement la manière dont évoluera la réalité augmentée. Les perspectives d'avenir de la réalité augmentée mobile sur le web sont prometteuses, et offrent de nouvelles opportunités. Cette technologie continuera sans doute d'évoluer et influencera la manière dont nous interagissons avec l'environnement et le monde digital.

11. Références

- Ahn, S., Ko, H., & Yoo, B. (2014). Webizing mobile augmented reality content. *New Review of Hypermedia and Multimedia*, 20(1), 79-100. <https://doi.org/10.1080/13614568.2013.857727>
- Akçayır, M., Akçayır, G., Pektaş, H. M., & Ocak, M. A. (2016). Augmented reality in science laboratories : The effects of augmented reality on university students' laboratory skills and attitudes toward science laboratories. *Computers in Human Behavior*, 57, 334-342. <https://doi.org/10.1016/j.chb.2015.12.054>
- Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A., & Wagner, D. (2015). *The History of Mobile Augmented Reality*.
- Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385. <https://doi.org/10.1162/pres.1997.6.4.355>
- Benyon, D., Quigley, A., O'Keefe, B., & Riva, G. (2014). Presence and digital tourism. *AI & SOCIETY*, 29(4), 521-529. <https://doi.org/10.1007/s00146-013-0493-8>
- Billinghurst, M., Clark, A., & Lee, G. (2015). A Survey of Augmented Reality. *Foundations and Trends® in Human-Computer Interaction*, 8(2-3), 73-272. <https://doi.org/10.1561/11000000049>
- Chatzopoulos, D., Bermejo, C., Huang, Z., & Hui, P. (2017). Mobile Augmented Reality Survey : From Where We Are to Where We Go. *IEEE Access*, 5, 6917-6950. <https://doi.org/10.1109/ACCESS.2017.2698164>
- Chen, Y., Wang, Q., Chen, H., Song, X., Tang, H., & Tian, M. (2019). An overview of augmented reality technology. *Journal of Physics: Conference Series*, 1237(2), 022082. <https://doi.org/10.1088/1742-6596/1237/2/022082>
- Chylińska, D., & Kołodziejczyk, K. (2018). Geotourism in an urban space? *Open Geosciences*, 10(1), 297-310. <https://doi.org/10.1515/geo-2018-0023>
- Cranmer, E. E., tom Dieck, M. C., & Fountoulaki, P. (2020). Exploring the value of augmented reality for tourism. *Tourism Management Perspectives*, 35, 100672. <https://doi.org/10.1016/j.tmp.2020.100672>
- Diegmann, P., Schmidt-Kraepelin, M., Eynden, S., & Basten, D. (2015). Benefits of Augmented Reality in Educational Environments—A Systematic Literature Review. *Wirtschaftsinformatik Proceedings 2015*. <https://aisel.aisnet.org/wi2015/103>
- Dowling, R., & Newsome, D. (2018). Geotourism : Definition, characteristics and international perspectives. *Handbook of Geotourism*, 1-22.
- Emmanuel, R., Valentine, R., & Carole, H. (s. d.). (2001). *Géologie, géomorphologie et tourisme urbain. Le cas de Lausanne*. 11.
- Fedorov, R., Frajberg, D., & Fraternali, P. (2016). A Framework for Outdoor Mobile Augmented Reality and Its Application to Mountain Peak Detection. In L. T. De Paolis & A. Mongelli (Éds.), *Augmented Reality, Virtual Reality, and Computer Graphics* (p. 281-301). Springer International Publishing. https://doi.org/10.1007/978-3-319-40621-3_21
- Feiner, S., MacIntyre, B., Höllerer, T., & Webster, A. (1997). A touring machine : Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal Technologies*, 1(4), 208-217. <https://doi.org/10.1007/BF01682023>

- Furness, T. A. (1986). The Super Cockpit and its Human Factors Challenges. *Proceedings of the Human Factors Society Annual Meeting*, 30(1), 48-52.
<https://doi.org/10.1177/154193128603000112>
- Gavish, N., Gutiérrez, T., Webel, S., Rodríguez, J., Peveri, M., Bockholt, U., & Tecchia, F. (2015). Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6), 778-798.
<https://doi.org/10.1080/10494820.2013.815221>
- Geiger, P., Schickler, M., Pryss, R., Schobel, J., & Reichert, M. (2014). *Location-based Mobile Augmented Reality Applications : Challenges, Examples, Lessons Learned*. 383-394.
<http://dbis.eprints.uni-ulm.de/1028/>
- González-Delgado, J. Á., Martínez-Graña, A., Holgado, M., Gonzalo, J. C., & Legoinha, P. (2020). Augmented Reality as a Tool for Promoting the Tourist Value of the Geological Heritage Around Natural Filming Locations : A Case Study in “Sad Hill” (The Good, the Bad and the Ugly Movie, Burgos, Spain). *Geoheritage*, 12(2), 34. <https://doi.org/10.1007/s12371-020-00457-4>
- Gordon, J. E. (2018). Geoheritage, Geotourism and the Cultural Landscape : Enhancing the Visitor Experience and Promoting Geoconservation. *Geosciences*, 8(4), Art. 4.
<https://doi.org/10.3390/geosciences8040136>
- Henrysson, A., & Ollila, M. (2004). UMAR : Ubiquitous Mobile Augmented Reality. *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, 41-45.
<https://doi.org/10.1145/1052380.1052387>
- Jung, T., Chung, N., & Leue, M. C. (2015). The determinants of recommendations to use augmented reality technologies : The case of a Korean theme park. *Tourism Management*, 49, 75-86.
<https://doi.org/10.1016/j.tourman.2015.02.013>
- Jung, T. H., & tom Dieck, M. C. (2017). Augmented reality, virtual reality and 3D printing for the co-creation of value for the visitor experience at cultural heritage places. *Journal of Place Management and Development*, 10(2), 140-151. <https://doi.org/10.1108/JPM-07-2016-0045>
- Kamarainen, A., Reilly, J., Metcalf, S., Grotzer, T., & Dede, C. (2018). Using Mobile Location-Based Augmented Reality to Support Outdoor Learning in Undergraduate Ecology and Environmental Science Courses. *Bulletin of the Ecological Society of America*, 99(2), 259-276.
- Kanade, P., & Prasad, J. P. (2021). Mobile and Location Based Service using Augmented Reality : A Review. *European Journal of Electrical Engineering and Computer Science*, 5(2), Art. 2.
<https://doi.org/10.24018/ejece.2021.5.2.303>
- Karami, E., Prasad, S., & Shehata, M. (2017). *Image Matching Using SIFT, SURF, BRIEF and ORB : Performance Comparison for Distorted Images* (arXiv:1710.02726). arXiv.
<https://doi.org/10.48550/arXiv.1710.02726>
- Kumar, P., Mantri, A., & Anireddy, S. (2019). *Innovations in Tourism Industry & Development Using Augmented Reality (AR), Virtual Reality (VR)*. <https://doi.org/10.1109/TENCON.2019.8929478>
- Liarokapis, F., Greatbatch, I., Mountain, D., Gunesh, A., Brujic-Okretic, V., & Raper, J. (2005). Mobile augmented reality techniques for GeoVisualisation. *Ninth International Conference on Information Visualisation (IV'05)*, 745-751. <https://doi.org/10.1109/IV.2005.79>

- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110.
<https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- Majeed, Z., & Ali, H. (2020). A review of augmented reality in educational applications. *International Journal of Advanced Technology and Engineering Exploration*, 7, 20-27.
<https://doi.org/10.19101/IJATEE.2019.650068>
- Marino Alfonso, J. L., Poblete Piedrabuena, M. Á., Beato Bergua, S., & Herrera Arenas, D. (2021). Geotourism Itineraries and Augmented Reality in the Geomorphosites of the Arribes del Duero Natural Park (Zamora Sector, Spain). *Geoheritage*, 13(1), 16. <https://doi.org/10.1007/s12371-021-00539-x>
- Martínez-Graña, A. M., Díez, T., González-Delgado, J. Á., Gonzalo-Corral, J. C., & Merchán, L. (2022). Geological Heritage in the “Arribes del Duero” Natural Park (Western, Spain) : A Case Study of Introducing Educational Information via Augmented Reality and 3D Virtual Itineraries. *Land*, 11(11), Art. 11. <https://doi.org/10.3390/land11111916>
- Mohring, M., Lessig, C., & Bimber, O. (2004). Video See-Through AR on Consumer Cell-Phones. *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 252-253. <https://doi.org/10.1109/ISMAR.2004.63>
- Morikawa, C., Kobayashi, M., Satoh, M., Kuroda, Y., Inomata, T., Matsuo, H., Miura, T., & Hilaga, M. (2021). Image and video processing on mobile devices : A survey. *The Visual Computer*, 37(12), 2931-2949. <https://doi.org/10.1007/s00371-021-02200-8>
- Morrison, A., Mulloni, A., Lemmelae, S., Oulasvirta, A., Jacucci, G., Peltonen, P., Regenbrecht, H., & Schmalstieg, D. (2011). Collaborative use of mobile augmented reality with paper maps. *Computers & Graphics*, 35, 789-799.
- Ólafsdóttir, R. (2019). Geotourism. *Geosciences*, 9(1), Art. 1.
<https://doi.org/10.3390/geosciences9010048>
- Ou, Y., Cai, Z., Lu, J., Dong, J., & Ling, Y. (2020). *Evaluation of Image Feature Detection and Matching Algorithms*. 220-224. <https://doi.org/10.1109/ICCCS49078.2020.9118480>
- Pai, Y. S., Yap, H. J., Dawal, S., Ramesh, S., & Phoon, S. (2016). Virtual Planning, Control, and Machining for a Modular-Based Automated Factory Operation in an Augmented Reality Environment. *Scientific Reports*, 6, 27380. <https://doi.org/10.1038/srep27380>
- Paucher, R., & Turk, M. (2010). Location-based augmented reality on mobile phones. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 9-16.
<https://doi.org/10.1109/CVPRW.2010.5543249>
- Perry, B. (2015). Gamifying French Language Learning : A Case Study Examining a Quest-based, Augmented Reality Mobile Learning-tool. *Procedia - Social and Behavioral Sciences*, 174, 2308-2315. <https://doi.org/10.1016/j.sbspro.2015.01.892>
- Qiao, X., Ren, P., Dustdar, S., & Chen, J. (2018). A New Era for Web AR with Mobile Edge Computing. *IEEE Internet Computing*, 22(4), 46-55.
<https://doi.org/10.1109/MIC.2018.043051464>

- Qiao, X., Ren, P., Dustdar, S., Liu, L., Ma, H., & Chen, J. (2019). Web AR : A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights. *Proceedings of the IEEE*, 107(4), 651-666. <https://doi.org/10.1109/JPROC.2019.2895105>
- Qiao, X., Ren, P., Nan, G., Liu, L., Dustdar, S., & Chen, J. (2019). Mobile web augmented reality in 5G and beyond : Challenges, opportunities, and future directions. *China Communications*, 16(9), 141-154. <https://doi.org/10.23919/JCC.2019.09.010>
- Renius, O. (2019). *A Technical Evaluation of the WebXR Device API for Developing Augmented Reality Web Applications*. <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-157773>
- Rousset, B. (2013). *La Molasse grise de Lausanne. De la roche sédimentaire détritique au matériau de construction séculaire*.
- Starnes, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R. W., & Pentland, A. (1997). Augmented Reality through Wearable Computing. *Presence: Teleoperators and Virtual Environments*, 6(4), 386-398. <https://doi.org/10.1162/pres.1997.6.4.386>
- Sutherland, I. E. (1968). A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 757-764. <https://doi.org/10.1145/1476589.1476686>
- Tareen, S. A. K., & Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1-10. <https://doi.org/10.1109/ICOMET.2018.8346440>
- Tscheu, F., & Buhalis, D. (2016). Augmented Reality at Cultural Heritage sites. In A. Inversini & R. Schegg (Éds.), *Information and Communication Technologies in Tourism 2016* (p. 607-619). Springer International Publishing. https://doi.org/10.1007/978-3-319-28231-2_44
- Vyas, D. A., & Bhatt, D. N. (2017). Augmented Reality (AR) Applications : A survey on Current Trends, Challenges, & Future Scope. *International Journal of Advanced Research in Computer Science*, 8(5), Art. 5. <https://doi.org/10.26483/ijarcs.v8i5.4059>
- Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., & Schmalstieg, D. (2008). Pose tracking from natural features on mobile phones. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 125-134. <https://doi.org/10.1109/ISMAR.2008.4637338>
- Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., & Schmalstieg, D. (2010). Real-Time Detection and Tracking for Augmented Reality on Mobile Phones. *IEEE transactions on visualization and computer graphics*, 16, 355-368. <https://doi.org/10.1109/TVCG.2009.99>
- Wu, H.-K., Lee, S. W.-Y., Chang, H.-Y., & Liang, J.-C. (2013). Current status, opportunities and challenges of augmented reality in education. *Computers & Education*, 62, 41-49. <https://doi.org/10.1016/j.compedu.2012.10.024>
- Yen, J.-C., Tsai, C.-H., & Wu, M. (2013). Augmented Reality in the Higher Education : Students' Science Concept Learning and Academic Achievement in Astronomy. *Procedia - Social and Behavioral Sciences*, 103, 165-173. <https://doi.org/10.1016/j.sbspro.2013.10.322>
- Yildiz, E. P. (2021). *Augmented Reality Research and Applications in Education*. IntechOpen. <https://doi.org/10.5772/intechopen.99356>

Yovcheva, Z., Buhalis, D., & Gatzidis, C. (2013). Engineering Augmented Tourism Experiences. In L. Cantoni & Z. (Phil) Xiang (Éds.), *Information and Communication Technologies in Tourism 2013* (p. 24-35). Springer. https://doi.org/10.1007/978-3-642-36309-2_3

11.1 Références en ligne

How Augmented Reality Travel Will Re-shape the Tourism Industry. (2017, juin 29). *InnovatAR*. Consulté le 14.01.2023 à l'adresse <https://innovatar.io/how-augmented-reality-travel-will-re-shape-the-tourism-industry/>

Gatis, Z. (2019). *Marker-based vs markerless augmented reality : Pros & cons | Overly app*. Overly. Consulté le 03.02.2023 à l'adresse <https://overlyapp.com/blog/marker-based-vs-markerless-augmented-reality-pros-cons-examples/>

Perera, S. A. (2018, août 18). A Comparison of SIFT , SURF and ORB. *Medium*. Consulté le 14.01.2023 à l'adresse <https://medium.com/@shehan.a.perera/a-comparison-of-sift-surf-and-orb-333d64bcaaea>