République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
Ecole nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

Mémoire de Master

Pour l'obtention du diplôme Master II en Informatique

Option : Systèmes Informatiques

Thème :

Apprentissage profond pour le Tracking en Réalité Augmentée

Réalisé par :

- Samir SI-MOHAMMED

Encadré par :

- Pr. Karima BENATCHBA

Promotion : 2019/2020

ESI (Ecole nationale Supérieure d'Informatique) B.P 68 M Oued Smar Alger
Tel: +213 023 93 91 32 Fax : +213 023 93 91 34
Site web: http://www.esi.dz

Remerciements

Je tiens avant tout à adresser mes remerciements les plus sincères et les plus vifs à mon encadrante Mme Karima BENATCHBA pour ses conseils, le temps qu'elle m'a consacré et pour la confiance qu'elle m'a témoignée. J'espère avoir été à la hauteur de ses attentes.

Je remercie l'ensemble du Staff de l'Ecole Nationale Supérieure d'Informatique, qui font un travail formidable et qui chaque jour contribuent à nous rendre fiers de notre école. Particulièrement Mme Dahbia AIT ALI YAHIA pour sa gentillesse, sa patience et pour avoir toujours été à notre écoute.

Je tiens aussi à remercier les membres du Jury, pour l'intérêt qu'ils ont témoigné à mes travaux et pour le temps qu'ils passeront à me lire.

Je remercie ensuite toutes celles et ceux, que je n'ai pas pu citer mais qui ont contribué à la réalisation de ce modeste travail.

Enfin je remercie les personnes les plus chères à mes yeux, mon grand frère, qui ne cesse de me rendre fier et surtout de me placer la barre de plus en plus haut..., ma mère, qui n'a cessé de me soutenir durant ces longs mois, et enfin, celui qui fut en même temps mon plus grand supporter, et surtout mon plus grand modèle, mon père.

Abstract

Augmented Reality is a field of research in full swing, supported by many application areas such as education, or health. It aims to include virtual elements in a real environment, which requires an integration that makes the user believe that these coexist perfectly with the real environment. This major component of object integration is called "Tracking". Tracking is therefore very important in Augmented Reality, and this is what makes the user experience in Augmented Reality a success by making the user feel immersed in it. To do this, tracking must be carried out efficiently and quickly, to avoid integration errors, as well as delays in placing objects which may be unpleasant for the user, and which may distort their immersion. The number of images to be processed per second being important, it would be wise to apply deep learning, which has recently demonstrated its effectiveness in imaging problems.

In this report, we are interested by the application of deep learning techniques to improve the tracking process in Augmented Reality. We present the study of Augmented Reality in general, deep learning, as well as the types of tracking and the different methods used in the state of the art. We also propose a research hypothesis on the theme. We are interested in this report, in the application of deep learning techniques to improve the tracking process in Augmented Reality. We present the study of Augmented Reality in general, deep learning, as well as the types of tracking and the different methods used in the state of the art. We also propose a research hypothesis on the subject.

Keywords : Augmented Reality, Deep Learning, Tracking, Computer Vision.

Résumé

La Réalité Augmentée est un domaine de recherche en pleine effervescence, soutenu par de nombreux domaines d'application tels que l'éducation, ou la santé. Elle vise à inclure des éléments virtuels dans un environnement réel. Cela nécessite une intégration transparente à l'utilisateur qui voit une coexistence parfaite de ces éléments avec l'environnement réel. Cette composante majeure d'intégration des objets est appelée « Tracking ». Il est donc très important en Réalité Augmentée, car il permet à l'utilisateur de se sentir en immersion. Pour ce faire, le tracking doit y être effectué de façon efficace et rapide, pour éviter les erreurs d'intégration, ainsi que les délais de placement des objets qui qui peuvent fausser son immersion et peuvent être désagréable pour l'utilisateur. Le nombre d'images à traiter à la seconde étant important, il serait judicieux d'y appliquer l'apprentissage profond, qui a dernièrement démontré son efficacité dans les problèmes d'imagerie.

Nous nous intéressons dans ce rapport, à l'application des techniques d'apprentissage profond pour l'amélioration du processus de tracking en Réalité Augmentée. Nous présentons l'étude de la Réalité Augmentée de façon générale, l'apprentissage profond, ainsi que les types de tracking et les différentes méthodes utilisées dans l'état de l'art. Nous proposons aussi une hypothèse de recherche sur le thème.

Mots-clés : Réalité Augmentée, Deep Learning, Tracking, Vision par Ordinateur

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

La Réalité Augmentée fait partie des domaines les plus dynamiques de ces dernières années. Cette technologie a rapidement conquis une place incontournable dans l'informatique contemporaine, si bien qu'elle a fini par toucher une multitude d'autres domaines comme la médecine ou les jeux-vidéo. Pour faire de l'expérience utilisateur une expérience plus immersive, il faut améliorer une étape cruciale de la Réalité Augmentée, qui est le Tracking, et qui consiste à garder la trace d'un l'objet dans une image en détectant sa position, et ce pour permettre une meilleure intégration des objets virtuels dans l'environnement réel.

Le tracking traitant donc d'une branche de l'imagerie, divers chercheurs ont trouvé judicieux d'y appliquer le Deep Learning, qui a déjà montré sa grande efficacité en ce qui concerne la détection d'objets. En effet, les réseaux de neurones, qui sont largement utilisés pour la reconnaissance d'images, devraient s'avérer efficaces pour le Tracking en Réalité Augmentée, surtout que celui-ci doit se faire en temps réel. Ensuite, le nombre de données à traiter durant le tracking est important (nombre considérable d'images à traiter à la seconde). Or, le Deep Learning est connu pour le traitement de données massives de manière efficace.

L'objectif de ce rapport est donc d'analyser certaines méthodes de l'état de l'art de l'application du Deep Learning en Tracking, sur des vidéos diverses en général, et plus particulièrement dans des systèmes de Réalité Augmentée.

Dans le premier chapitre, nous définissons la Réalité Augmentée de façon générale. Nous décrivons ensuite de son historique ainsi que le lien étroit qui la relie à la Réalité Virtuelle. Nous évoquons ensuite quelques concepts primordiaux de cette technologie, de l'environnement matériel nécessaire pour le déroulement d'une expérience Réalité Augmentée, ainsi que ses domaines d'application et les défis qui lui font face.

Nous présentons dans le deuxième chapitre le Tracking, en évoquant ses étapes, ses types, ainsi que les méthodes qui y sont utilisées dans l'état de l'art. Nous précisons les conditions nécessaires pour l'application de chaque méthode, ses avantages ainsi que ses inconvénients.

Dans le troisième chapitre, nous analysons trois méthodes où le Deep Learning a été utilisé pour effectuer du tracking, que nous avons choisies pour leur efficacité ainsi que leur correspondance avec notre problématique. Nous détaillons chaque méthode, son mode de fonctionnement, ses avantages ainsi que ses inconvénients, tout en précisant l'architecture du réseau de neurones utilisé dans chaque cas. Nous finissons en évoquant des perspectives d'évolution de la technologie en général.

En annexe I, nous présentons le Deep Learning, en détaillant ses applications, ainsi que les réseaux de neurones en général. Etant donné que nous avons besoin de métriques pour comparer les méthodes, nous présentons aussi en Annexe II les métriques les plus importantes qui sont aujourd'hui utilisées pour l'évaluation de méthodes de tracking

1. Chapitre I : Réalité augmentée

Parmi les branches de l'informatique qui ont connu le plus d'évolution ces dernières années se trouve la Réalité Augmentée. En effet, cette technologie s'avère aujourd'hui omniprésente dans plusieurs domaines d'applications tels que la médecine, l'éducation, le divertissement…etc. Nous proposons dans ce qui suit certaines définitions de la Réalité Augmentée, ainsi que ses types, et nous évoquons brièvement le lien étroit qu'il existe entre elle et la Réalité Virtuelle. Ensuite, nous présentons l'histoire de cette technologie à travers les dernières décennies, son fonctionnement ainsi que les étapes par lesquelles passe l'utilisation d'un système RA. Puis, nous citons certaines notions de RA, avant de passer au matériel physique utilisé en RA. Les domaines d'application de cette technologie ainsi que les challenges auxquels elle fait face sont présentées par la suite, avant de conclure le chapitre.

1.1 Définition

Selon (Azuma et al., 1997) on peut définir la Réalité Augmentée (RA) comme étant une variante de la Réalité Virtuelle (RV), qui est elle-même une technologie permettant de plonger l'utilisateur en immersion totale dans un environnement synthétique. La RA quant à elle permet à l'utilisateur de voir le monde réel, avec des objets virtuels superposés au-dessus de ceux du monde réel. Par conséquent, la RA enrichit la réalité, au lieu de la remplacer complètement, en intégrant des éléments virtuels à l'environnement réel de façon à ce qu'ils paraissent coexister.

La différence la plus évidente entre la RA et la RV est l'équipement lui-même. Une expérience RV devrait être vue à travers un casque, qui requiert une puissance de calcul importante permettant d'afficher des mondes virtuels entiers sans décalages entre les images. En somme, la RV donne à l'utilisateur la sensation d'être dans une toute autre situation que la sienne. Il n'a donc pas besoin de se trouver à un endroit précis, il lui suffit de porter un dispositif pour avoir l'impression d'y être physiquement transporté. La RA, elle, ajoute de l'information et des éléments au monde physique de l'utilisateur et lui permet d'interagir avec eux.

Toujours selon (Azuma et al., 1997), la RA peut être définie par trois caractéristiques :

1. La combinaison des contenus virtuel et réel
2. L'interaction en temps réel
3. Intégration 3D des objets virtuels

Lorsqu'on évoque la réalité augmentée, on a souvent tendance aussi à penser à un casque imposant porté sur la tête, avec des lunettes et divers dispositifs. Cependant la RA fait partie de notre quotidien. Ainsi, toutes les applications de photographie où l'on applique des filtres à nos vidéos sont en fait de la réalité augmentée. Elles superposent à la réalité (qui est dans ce cas la vidéo 3D) divers effets virtuels (les filtres en question).

L'image la plus commune que l'on se fait de la réalité augmentée est qu'elle permet d'altérer la vision de l'utilisateur, alors que cette technologie peut agir sur les cinq sens, en ajoutant par exemple des sons artificiels à un environnement sonore (Harma et al., 2004), ou encore des odeurs stimulant le sens olfactif de l'utilisateur (Wang et al., 2018).

Selon (Azuma et al., 2001), la RA peut être divisée en trois types :

1. ***Video See-Through :*** Les images sont d'abord capturées par un dispositif comme une caméra ou une tablette, puis sont projetées sur un écran après avoir été « augmentées » en y intégrant les objets virtuels adéquats.
2. ***Optical See-Through :*** C'est le type standard auquel est souvent associée la RA. Les objets sont directement intégrés dans l'environnement réel que l'utilisateur a en face de lui, à travers un casque comportant un écran semi-transparent.
3. ***Projective AR :*** La RA spatialement augmentée. Le contenu virtuel est projeté directement sur un objet de l'environnement réel.

1.2 Historique

Le terme Réalité Augmentée a été utilisé la première fois par (Caudell et Mizell, 1992). Ingénieurs à Boeing, ils travaillaient sur un casque qui aiderait les ingénieurs en aviation dans différents schémas de câblages complexes. Selon leur rapport, l'objectif de la réalité augmentée était de permettre d'augmenter l'efficacité de certains travaux humains manuels dans la construction aéronautique, tout en réduisant les coûts.

La RA partage d'une certaine façon la même histoire que la Réalité Virtuelle. En effet ces deux technologies ont le même ancêtre commun : L'épée de Damoclès. Conçue en 1968, l'épée de Damoclès (Figure 1) a été créée par un chercheur informaticien, **Ivan Sutherland**. Son objectif était de créer un affichage parfait, c'est-à-dire une interface digitale capable d'altérer le monde physique. Le prototype était cependant si imposant qu'il devait être suspendu par un bras mécanique. Le résultat final était un simple affichage d'une pièce étroite difficilement explorable. C'était cependant l'une des premières expériences où l'Homme a tenté de remplacer le monde physique par un monde virtuel ou digital (Azuma et al., 2001).

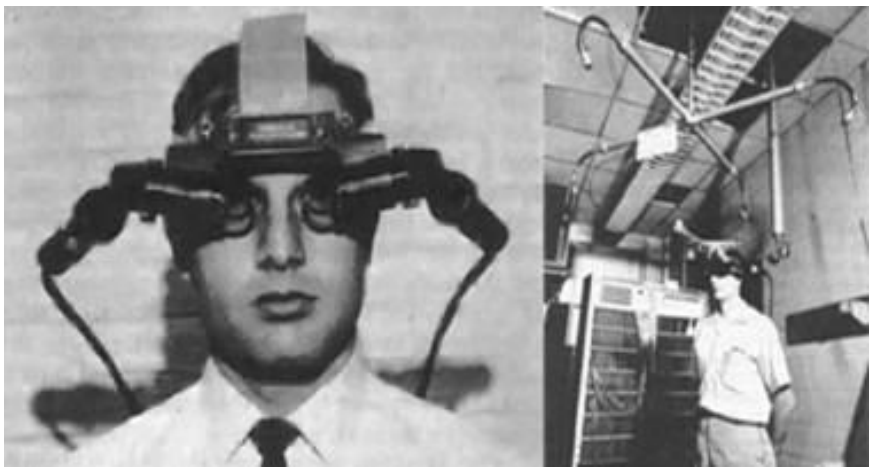

Figure 1 - Epée de Damoclès (360natives, 2018)

Après cette expérience et depuis les années 90 en particulier, beaucoup de scientifiques ont tenté de compléter et de parfaire la vision de Sutherland. Aujourd'hui, nous n'avons plus besoin de

bras mécaniques pour suspendre l'équipement nécessaire. Les casques, nettement plus puissants que l'épée de Damoclès, peuvent être portés sur la tête comme des lunettes, et même être transportés dans nos poches. La plupart des applications utilisées pour la RA sont d'ailleurs sur smartphone, comme par exemple : Pokémon GO.

L'industrie des smartphones a en effet contribué à la croissance de l'industrie RA (Youcheva et al., 2012). Ceci est dû au fait que certains équipements nécessaires pour faire fonctionner les smartphones comme les gyroscopes, accéléromètres…etc. sont aussi nécessaires pour les casques RA/RV. La demande massive en smartphones durant les dernières années a donc provoqué un développement considérable des équipements physiques utilisés, ainsi qu'une réduction notable de leur coût, ce qui a inéluctablement provoqué le développement de l'industrie RA, comme présenté dans (Schmalstieg et Wagner, 2008).

1.2.1 Fonctionnement

La façon dont les dispositifs précédents sont utilisés pour créer une expérience RA est décrite dans la Figure 2 :

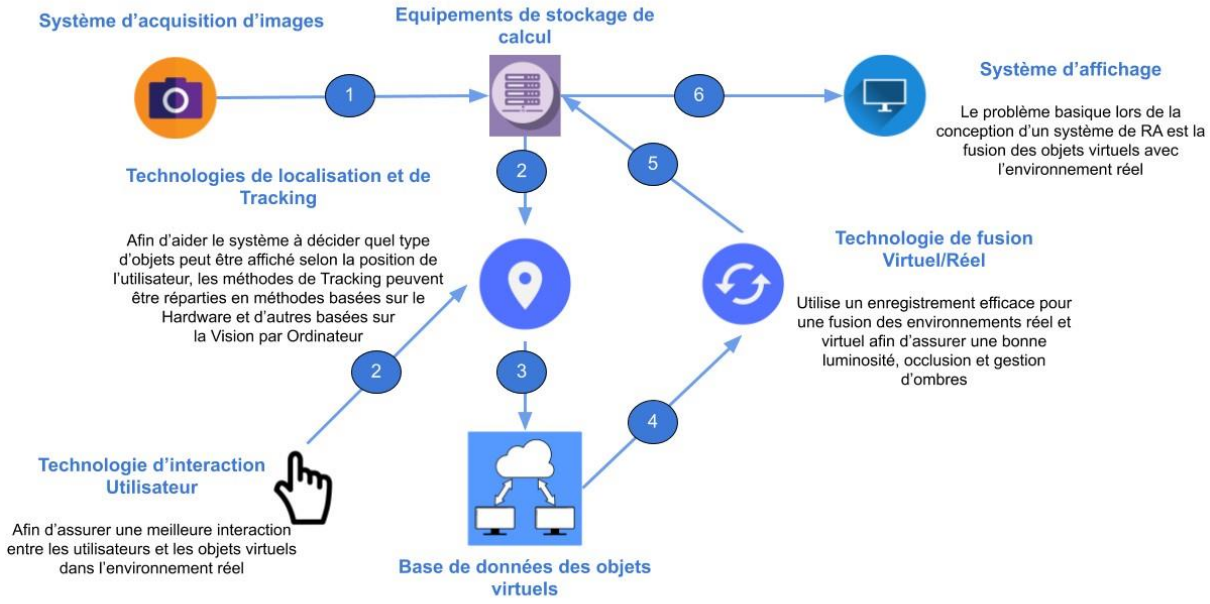

Figure 2 - Processus de la RA (Juan Cheng et al., 2018)

Les étapes majeures pour le bon déroulement de l'utilisation d'un système RA sont présentées dans la Figure 2, qui sont les suivantes :

I. **Acquisition des images :** Les caméras servent en premier lieu à capturer l'environnement tel qu'il est, et le transférer au dispositif.
II. **Tracking et Localisation :** L'image est ensuite analysée et les objets sont détectés pour faciliter le placement d'objets virtuels par-dessus.

III. **Accès à l'objet virtuel :** Le système accède à la base de données contenant les objets virtuels pouvant être ajoutés.

IV. **Fusion :** Des algorithmes sont exécutés pour assurer une bonne occlusion et un bon éclairage, pour faire confondre du mieux possible les mondes réel et virtuel.

V. **Affichage des objets virtuels :** La dernière étape est l'application des résultats des algorithmes exécutés afin d'intégrer de la meilleure façon les objets virtuels choisis.

1.3 Quelques notions de RA

Nous recensons dans ce qui suit les problématiques auxquelles fait face la RA afin d'assurer une bonne expérience utilisateur RA, ainsi que le fonctionnement technique de celle-ci. On peut définir une expérience RA comme étant l'utilisation d'un système RA qui plonge l'utilisateur dans le monde physique doté d'objets virtuels superposés.

1.3.1 Occlusion

L'occlusion représente la situation où un objet est bloqué par un autre (Figure 3), c'est une obstruction effectuée d'un objet sur un autre. Dans le cas où par exemple un utilisateur en pleine expérience RA se déplace et se met derrière un mur. S'il arrive toujours à voir les objets virtuels derrière le mur, cela voudrait dire que l'occlusion n'est pas bien gérée, et que l'immersion de l'application est insuffisante. Sans une bonne gestion de ce phénomène, l'utilisateur aura donc l'impression que l'objet réel est plus loin qu'il ne l'est vraiment, ce qui fausse l'immersion. (Tian et al., 2010).

Figure 3 - Exemple d'occlusion (GuidiGO, 2018)

1.3.2 Eclairage

L'éclairage est un paramètre extrêmement important pour assurer l'immersion dans une expérience RA. C'est l'évolution des couleurs et des ombres engendrées par un objet virtuel relativement à l'environnement dans lequel ils sont (Figure 4). Par un exemple un objet virtuel placé dans deux environnements, l'un éclairé et l'autre sombre ne devrait pas donner la même représentation.

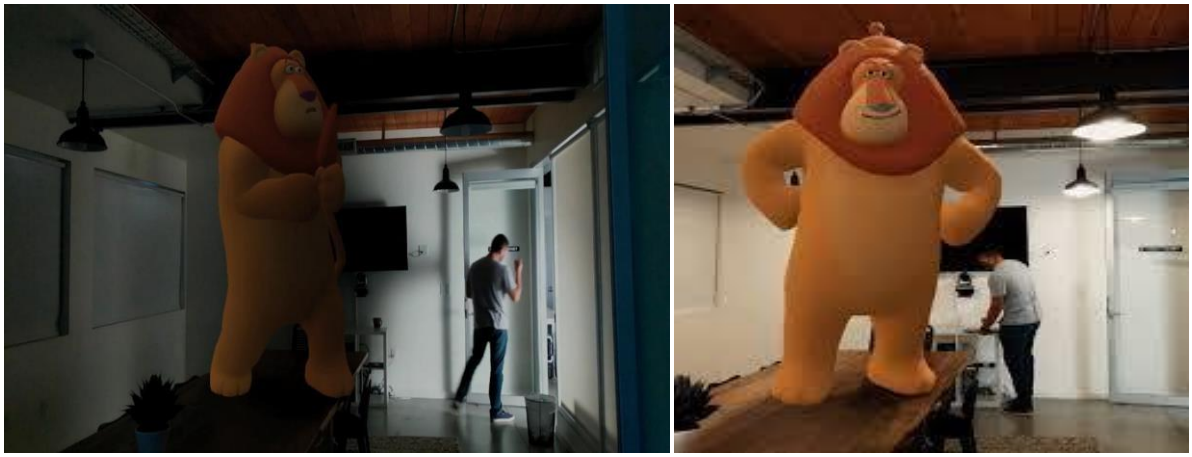

Figure 4 - Impact de l'éclairage dans l'expérience RA (Coursera.com, 2019)

1.4 Environnement matériel

Nous présentons dans ce qui suit les équipements nécessaires pour une expérience RA, pour le calcul (Processeur), l'affichage (Casque, ou lunettes), ainsi que les capteurs qui sont plus spécifiques aux application RA sur mobile. Ces capteurs peuvent prendre la forme de systèmes micro-électromécaniques (Accéléromètre, GPS, gyroscope…), qui permettent de localiser avec précision la localisation ainsi que la rotation des dispositifs mobiles, afin d'y ajouter les objets virtuels de façon plus efficace.

1.4.1 Processeur

Son rôle est de coordonner et d'analyser les entrées visuelles (ou sonores…) via les capteurs, stocker et accéder aux informations, gérer les tâches de chaque dispositif du système et enfin générer les signaux appropriés pour l'afficheur (Diguet et al., 2015). Les processeurs peuvent différer selon leur capacité de calcul et selon le dispositif principal, que ce soit un smartphone, un ordinateur de bureau, un système distribué de machines…etc. Dans tous les cas, le processeur doit avoir assez de capacité de calcul pour effectuer les tâches dont il est responsable en temps réel, et ce pour assurer l'immersion de l'utilisateur dans l'expérience.

La Figure 5 décrit l'architecture d'un exemple de processeur (Kim et al., 2015). Les différents composants sont chacun responsable d'une partie de l'expérience RA, comme le calcul de la position des objets, un détecteur de points d'intérêt dans l'image, l'extraction des caractéristiques par le Description Generation Cluster…etc.

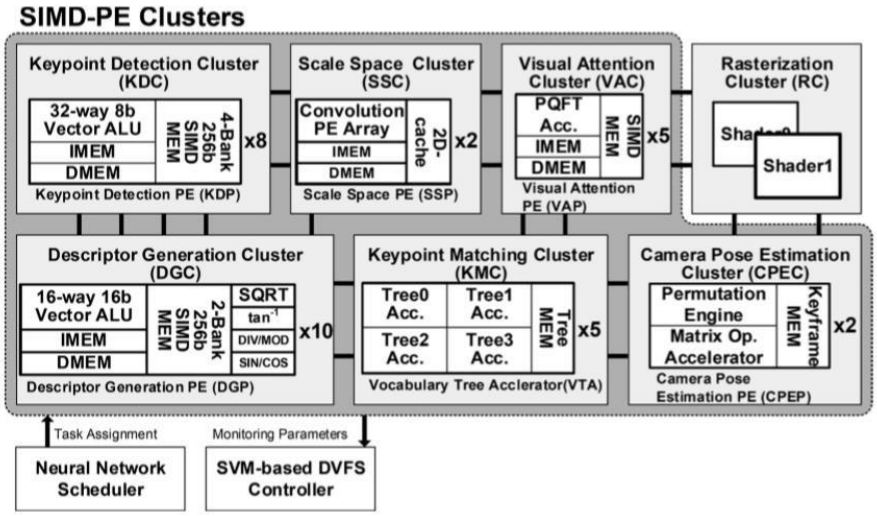

Figure 5 - Architecture Processeur RA (Kim et al., 2015)

1.4.2 Afficheur

Plusieurs technologies sont utilisées pour l'affichage dans un système RV. La plus connue est le dispositif HMD (Figure 6) « Head Mounted Display » [1]. C'est un dispositif porté sur la tête comme un casque, qui place les images du monde réel et virtuel dans le champ de vision de l'utilisateur. Il offre « 6 degrés de liberté » à l'utilisateur, c'est-à-dire qu'il lui permet de déplacer la tête sur les 3 axes X, Y et Z, tout en changeant l'orientation sur chacun de ces axes. Il offre donc une mobilité totale et complète. Les premiers à avoir proposé un HMD pour la RA sont (Claudell et Mizell, 1968).

Figure 6 - Head Mounted Display (Iccsl, 2019)

[1] https://electronics.howstuffworks.com/gadgets/other-gadgets/VR-gear1.htm

1.4.3 Capteurs

Afin d'intégrer les objets virtuels de la meilleure façon possible sur l'environnement réel, ce dernier doit être bien assimilé et reconnu par le système RA, qui, pour ce faire, s'appuie sur de nombres capteurs de position, de vitesse…etc. La Figure 7 représente l'ensemble de ces capteurs, que nous détaillons dans ce qui suit, ainsi que leur rôle.

1.4.3.1 Accéléromètre

Il permet de mesurer l'accélération du dispositif en mouvement, c'est-à-dire le changement de vélocité (Carmigniani et al., 2011). Cette accélération peut être continue ou dynamique (Ex : Vibration du dispositif) : Par exemple si l'utilisateur se déplace pendant l'expérience à vitesse constante, en gardant son dispositif (Casque ou Smartphone) dans la même position, son accélération sera donc fixe, tandis que s'il secoue la tête, son accélération sera dynamique, ce qui peut influer sur le positionnement des objets environnants : Une accélération constante faciliterait cette tâche.

1.4.3.2 Gyroscope

Il permet de mesurer l'orientation et la vitesse angulaire du dispositif et de faire en sorte que l'application RA réponde bien à l'orientation du champ de vision de l'utilisateur. Si par exemple, l'utilisateur fait pencher le dispositif dans une certaine direction, les objets virtuels devraient garder la même position sans effectuer la même rotation.

1.4.3.3 Magnétomètre

Il fournit au dispositif (Smartphone dans ce cas) une orientation par rapport au champ magnétique terrestre. Ainsi l'appareil sait toujours où se situe le Nord, permettant ainsi l'orientation automatique des cartes virtuelles utilisées dans l'application selon l'orientation physique de l'utilisateur.

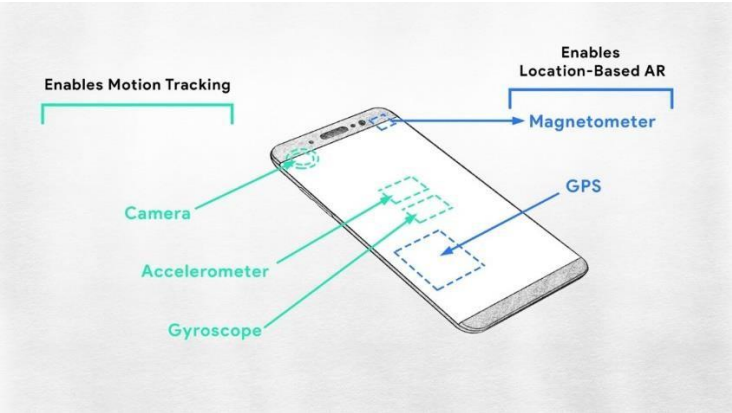

Figure 7 - Capteurs sur smartphone (Coursera.com, 2019)

1.4.4 Base de données

La base de données contient un ensemble d'objets virtuels pouvant être ajoutés à l'affichage pour augmenter le monde physique. Ceux-ci peuvent être sous forme Vidéo, images 2D ou modèles 3D. Cette base de données permet de stocker les objets virtuels avec lesquels la réalité sera augmentée durant l'expérience, et de les mettre à disposition du processeur qui les placera sur l'affichage de l'utilisateur.

1.5 Domaines d'application

La réalité augmentée possède une multitude de domaines d'applications possibles, c'est d'ailleurs ce qui rend cette technologie extrêmement populaire au sein du grand public.

1.5.1 Réseaux sociaux / Vente en ligne

Snapchat est la première plateforme à avoir instauré de la RA dans son contenu (Forbes.com, 2019). Les filtres, qui sont en fait des effets visuels s'ajoutant sur les visages dans des photos prises, sont par ailleurs devenus viraux dans toutes les autres plateformes de réseaux sociaux. La popularisation de la réalité augmentée dans les réseaux sociaux crée un nouveau marché cible en ligne. Les marques et marketeurs cherchent maintenant des moyens pour exploiter cette technologie. Le but est de faire connaître leurs produits et services auprès de leurs clients par le biais de la communication interactive. Par exemple, la RA est aujourd'hui utilisée pour faciliter l'achat en ligne, en simulant le positionnement d'un objet acheté dans une pièce quelconque d'une maison (Figure 8). Cela permet donc à l'utilisateur de percevoir si l'objet en question a les dimensions qu'il faut, si l'objet lui plaît vraiment une fois placé…etc. sans avoir à se déplacer et l'acheter sans l'avoir essayé, comme le montre la figure suivante, où l'on visualise le positionnement d'une veilleuse virtuelle sur une table :

Figure 8 - RA pour la vente en ligne (TechCrunch, 2018)

1.5.2 Jeux-Vidéos

Le premier jeu-vidéo utilisant la RA et ayant connu un succès planétaire est Pokémon GO en 2016. Les utilisateurs devaient se déplacer dans le monde entier pour trouver certains endroits où sont cachés des trésors, directement superposés aux paysages réels. Depuis, plusieurs jeux faisant appel à la RA ont vu le jour (Pokémon GO (Figure 9), Jurassic World…) car l'idée d'implanter cette expérience dans le jeu ne peut qu'augmenter l'immersion des joueurs. Toutefois, si celle-ci est mal faite l'immersion en sera grandement affectée.

Figure 9 - La RA pour les jeux-vidéo (Niantic, 2018)

(Nilsen et al, 2002) proposent dans leur papier intitulé « Motivations for Augmented Reality Gaming » une étude sur les motivations qui devraient pousser les développeurs de jeux-vidéo à y intégrer la RA, en faisant référence à l'aspect émotionnel, mental, social…etc. des joueurs.

1.5.3 Education

Plusieurs application RA ont déjà été développées à des fins pédagogiques. Elles permettent aux élèves de mieux visualiser du contenu quel qu'il soit. Par exemple, ils peuvent visualiser une séquence ADN en détail, et même interagir avec. Ils peuvent également visiter des monuments historiques comme s'ils y étaient. C'est souvent plus efficace qu'avoir recours à leur imagination.

(Mark Bilinghurst, 2002) dans son papier « Augmented Reality in Education » soulève la problématique de l'application de la RA en éducation en termes de recherche, où il affirme que le fait que les enfants soient directement immiscés dans l'expérience les pousse à apprendre plus facilement que s'ils étaient derrière un écran.

1.5.4 Santé

La RA est déjà très utilisée dans la médecine (Madison, 2018). La visualisation consiste par exemple à augmenter l'image qu'ils ont d'un membre d'un corps humain, en montrant la position exacte d'une pathologie sur ce membre. Pour les médecins en apprentissage, la Réalité Augmentée permet à la formation médicale de devenir beaucoup plus interactive et immédiate. Les applications RA peuvent être utilisées pour afficher des informations anatomiques sur un

squelette humain imprimé en 3D, ou offrir une illustration 3D du cœur humain, permettant ainsi une meilleure compréhension de la façon dont le corps humain fonctionne. Dans la figure suivante on peut voir des ossements humains apparents permettant aux médecins de visualiser plus précisément le positionnement de chaque os.

Figure 10 - La RA pour la santé (OneYoungWorld, 2017)

La RA a été utilisée dans beaucoup d'applications de la médecine, dont la pathologie anatomique par (Hanna et al., 2018) dans le papier intitulé « Augmented Reality Technology Using Microsoft Hololens in Anatomic Pathology » où les auteurs proposent une application directe des casques Hololens de Microsoft pour visualiser des pathologies au niveau microscopique de manière plus précise et plus interactive.

1.6 Challenges

Etant une technologie relativement récente, la RA fait face aujourd'hui à un certain nombre de contraintes et des challenges qu'elle devra surmonter si on veut qu'elle connaisse un essor plus considérable. Dans ce qui suit, nous allons présenter certains des challenges auxquels la RA doit faire face.

1.6.1 Interface Utilisateur

La plupart des interfaces utilisateur développées pour la RA sont difficilement intuitives (Poupyrev et al., 2002). De plus, les procédés utilisés lors du développement des applications mobiles standards ne sont plus d'actualité. Par exemple, comment naviguer sur un menu ? L'utilisateur devrait-il utiliser ses yeux, ou alors ses mains ? Il n'y a aujourd'hui que peu de modes d'utilisation conventionnels adoptés par les développeurs RA (comme par exemple 3D UI), c'est-à-dire qu''en RA il n'y a pas de mode d'utilisation conventionnel comme c'est le cas pour les ordinateurs classiques avec le clavier et la souris.

1.6.2 Matériel

La taille des dispositifs est en premier lieu très contraignante (Mekni et Lemieux, 2014), surtout si ces derniers doivent être portés sur la tête. Ceci est dû à la taille importante des processeurs, vu la capacité de calcul nécessaire. Ensuite, avec autant de calculs effectués en temps réel, la consommation d'énergie croît de manière importante (Liu et al., 2013), et donc la chaleur dégagée par les processeurs devient vite un problème difficile à résoudre.

1.6.3 Limites de la reconnaissance d'images

Un autre challenge qui peut être évoqué est celui qui nous intéresse le plus dans notre travail, car il traite de la reconnaissance d'images appliquée à la RA. Pour rappel, la reconnaissance d'images est le terme donné au fait de doter les machines de la capacité à reconnaître des formes et des objets dans une image, et donc en quelque sorte de « comprendre » l'environnement physique. Dans le cas de la RA, la machine doit être capable de comprendre au mieux la réalité pour pouvoir la doter d'objets virtuels appropriés (Behringer, 1999). Parmi les algorithmes qui sont aujourd'hui utilisés dans ce contexte on peut citer « Simultaneous Localization And Mapping » (SLAM), qui ont été implémentés pour aider la machine à inférer la position des objets autour d'elle (Thrun et Leonard, 2008).

1.7 Conclusion

La réalité augmentée est une technologie dont les domaines d'applications sont divers et variés, et dont le potentiel et les débouchés sont considérables. Elle n'a toutefois toujours pas atteint son apogée, et ceci est dû au fait qu'elle soit relativement récente. Même si elle a connu un essor considérable durant ces dernières années, elle fait toujours face à certains défis que la communauté des chercheurs n'a toujours pas réussi à surmonter.

Dans ce chapitre, nous avons défini la Réalité Augmentée de façon générale, en évoquant son historique, ainsi que ses différences et similitudes avec la Réalité Virtuelle. Nous avons aussi parlé des concepts fondamentaux de la RA, et de l'environnement matériel nécessaire au déroulement d'une expérience RA. Nous avons clôturé le chapitre en évoquant es différents domaines et les défis majeurs auxquels elle fait face.

Dans le chapitre suivant, nous parlerons de l'apprentissage profond, ou « Deep Learning », en définissant la structure de base qui y est utilisée : le réseau de neurones, tout en énumérant ses types, et son fonctionnement. Nous parlerons aussi de la vision par ordinateur et de la manière dont les réseaux de neurones peuvent y être appliqués.

2. Chapitre II : Tracking

Dans ce qui précède, nous avons parlé de la Réalité Augmentée, de ses types, du matériel nécessaire pour concevoir un système RA, ainsi que le fonctionnement de ce dernier. Nous avons aussi vu que parmi les étapes pour le bon déroulement d'une expérience RA, l'étape de Tracking occupe une place très importance, étant donné que c'est durant cette étape que les objets constituant le monde physique sont détectés et suivis, afin d'assurer la meilleure superposition possible avec les objets virtuels ajoutés. Nous présentons dans ce qui suit le Tracking de façon détaillée, ses types, ainsi que son processus d'exécution.

2.1 Définition

Durant l'utilisation d'un système RA, l'étape de capture, reconnaissance, segmentation et analyse des informations environnementales est appelée tracking. C'est donc grâce au tracking que le dispositif semble « reconnaître » ce qui l'entoure, et peut ainsi placer de manière efficace des objets virtuels. Il existe en RA deux types de Tracking (Figure 11) que nous allons présenter dans ce qui suit.

2.1.1 Outside-In Tracking

Le Outside-In Tracking utilise des caméras et d'autre types de capteurs placés dans une position fixe et orientée vers l'objet cible à traquer (Exemple : Casque) qui peut se déplacer librement dans la zone d'intersection des champs de vision des caméras. L'objet est donc observé depuis l'extérieur par le traqueur. Il est souvent important que l'objet ait un ensemble de marqueurs lui permettant d'être facilement repéré et facilitant le calcul de sa position relativement aux capteurs. De plus, bien que ce type de suivi de position puisse être obtenu à l'aide du spectre de la lumière visible, il est courant d'utiliser des marqueurs infrarouges (IR) et des caméras capables de détecter ce type de lumière.

Les performances et la précision liées à ce type de tracking sont souvent dépendantes de plusieurs facteurs comme la qualité des capteurs, la manière dont les objets sont traqués, la puissance de calcul et les algorithmes de Tracking (Langley, 2017).

2.1.2 Inside-Out Tracking

Le Inside-Out Tracking est une méthode extrêmement utilisée, et ceci est dû au fait que les HMD sont souvent utilisés lors des expériences AR. La différence avec le Outside-In Tracking est la position des caméras qui effectuent la traque. Elles sont maintenant placées dans le casque que l'utilisateur porte. Les objets dans l'environnement sont donc traqués en calculant leur position relative à chaque fois que la caméra (le casque) bouge, tandis que les capteurs sont toujours placés au niveau de l'environnement de manière fixe.

Le dispositif en question cherche donc à déterminer la position des objets en calculant le changement de sa position relativement à l'environnement. Lors du mouvement du casque, les capteurs réajustent sa position et l'environnement virtuel répond en temps réel.

Les caméras placées dans le HMD détectent les caractéristiques de l'environnement autour. Les marqueurs sont conçus de sorte à être facilement détectables par les dispositifs de Tracking, et sont placés de manière précise dans l'environnement. Ils peuvent être des points, des formes, des cercles…. Le suivi de position dans ce genre de Tracking peut également être réalisé à l'aide de marqueurs infrarouges (IR) et d'une caméra sensible à ce type de lumière.

En cas d'utilisation de marqueurs, le système ne fonctionne que dans la mesure où il peut les détecter. Si ceux-ci sont hors de son champ de vision, le suivi de position sera affecté. Les marqueurs peuvent être aussi quelque fois optionnels, dans ce qu'on appelle le « Tracking sans marqueurs ».

Par contre le Tracking sans marqueurs est une méthode basée sur les caractéristiques naturelles : Le système utilise les caractéristiques distinctives (Par exemples les zones fixes, les objets mouvants…etc.) existant à l'origine dans l'environnement pour déterminer la position et l'orientation. Les algorithmes du système identifient des images ou des formes spécifiques et les utilisent pour calculer la position de l'appareil dans l'espace. Les données des accéléromètres et des gyroscopes peuvent également être utilisées pour augmenter la précision du suivi de position (Langley, 2017).

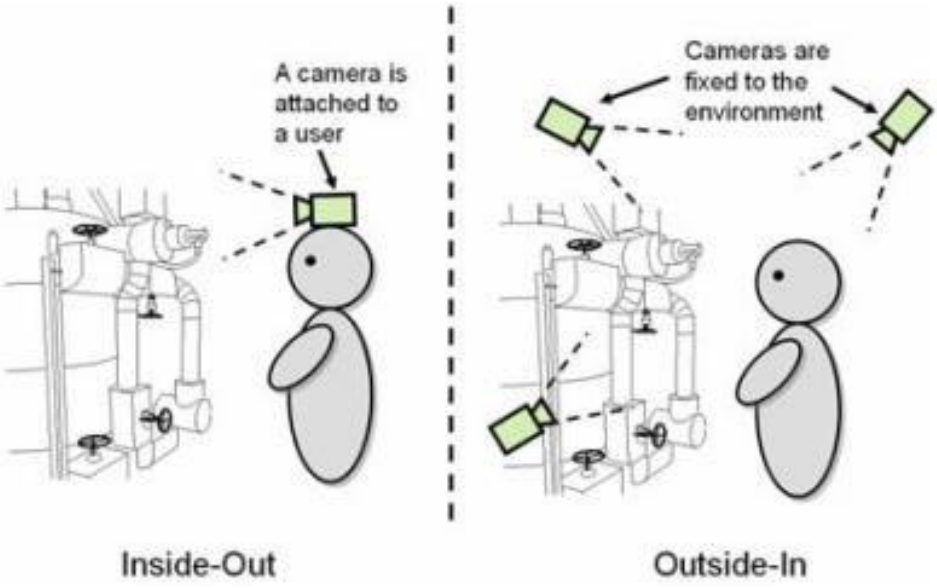

Figure 11 - Types de Tracking (Ishii, 2010)

2.2 Processus de Tracking

Traquer un objet dans une vidéo (le champ de vision de l'utilisateur dans le cas d'une expérience RA) revient à retracer son parcours et ses mouvements, en précisant la localisation de l'objet dans chaque image composant la vidéo. La difficulté réside dans le cas où l'objet a une forme assez complexe, ou que la représentation de l'objet peut varier selon l'éclairage de l'image, selon la couleur de l'objet, mais aussi la vélocité avec laquelle l'objet se déplace dans la vidéo. C'est pourquoi il est nettement plus simple de traquer un objet dont la vitesse de déplacement est faible (Jemilda et al., 2017).

Selon (Parekh et al., 2014), pour concevoir un traqueur d'objets il faut d'abord passer par certaines étapes que nous présentons dans ce qui suit.

2.2.1 Représentation de l'objet

La première condition nécessaire pour traquer un objet est que ce dernier soit clairement représenté, et d'une façon à ce que le 'Traqueur' reconnaisse et comprenne cette représentation clairement. On peut représenter un objet par sa **forme** ou son **apparence** (ou les deux en même temps dans certains cas) (Yilmaz et al., 2006). Chacune de ces deux méthodes s'effectue comme suit :

- **La forme :** Pour la définir clairement, on peut utiliser :
    - Les points : Les formes des objets peuvent être représentées soit par un point unique, soit par une multitude de point. Pour ce dernier, cas il est difficile de maintenir le Tracking dans une vidéo, dans le cas où il y a une occlusion avec d'autres objets, on perd donc certains points ou on les confond avec des points d'autres objets. C'est pourquoi il est préférable (quand c'est possible) de représenter l'objet par un seul point.
    - Les formes géométriques : Cela consiste à utiliser une forme géométrique qui approxime la forme de l'objet, tel qu'un ovale ou un rectangle, même si elle n'est pas précise.
    - Silhouette/Contour : Cela consiste à utiliser une/des formes géométriques qui contienne/contiennent l'objet.
    - Formes articulées : Cela consiste à considérer l'objet comme étant une connexion de différentes parties indépendantes, en établissant la forme de chacune.
    - Squelette : On tente de reproduire l'ossature d'un objet. Cette méthode est plus utilisée lorsqu'il s'agit de Tracking d'un corps humain.

La figure 12 résume les différentes représentations possibles d'un objet en tracking :

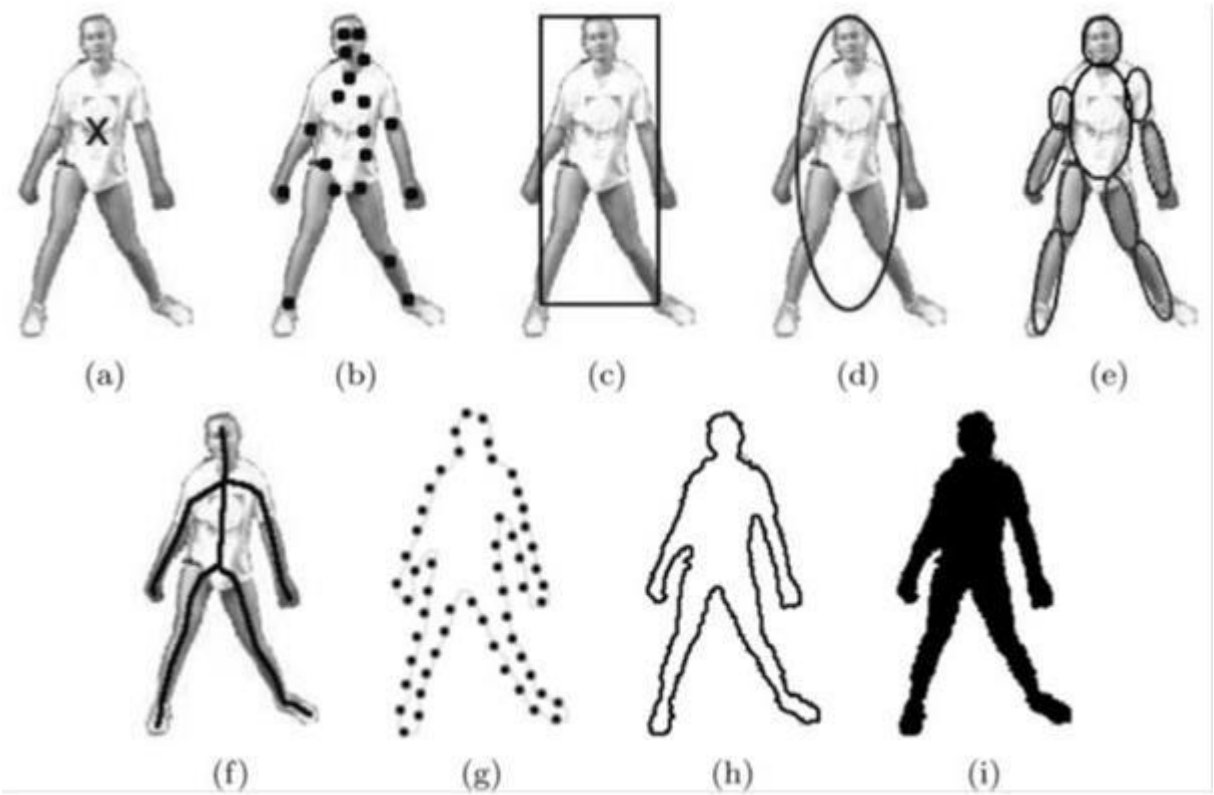

Figure 12 - Formes d'un Objet (Yilmaz, 2006)

(a) : Point unique
(b) : Points multiples
(c) : Formes géométriques
(d) : Formes géométriques
(e) : Formes articulées
(f) : Squelette
(g) : Silhouette/Contour
(h) : Silhouette/Contour
(i) : Silhouette

- **L'apparence :** Cela consiste à définir l'apparence, intérieure ou extérieure (contours) des objets en utilisant :
    - Densités probabilistiques : On peut prendre la région intérieure de l'objet (à l'intérieur de son contour par exemple) et y calculer une densité de probabilité qui représentera la distribution de points qui définiront la couleur ou la texture.
    - Modèle : On peut construire des modèles définis à partir de silhouettes et de simples formes géométriques, et essayer de trouver le meilleur modèle pour un

objet donné. Cette méthode est efficace quand l'objet n'est pas sujet à des distorsions dans la vidéo, c'est-à-dire quand il n'y a pas de changement de couleur, de formes, ou même de points de vue différents (Angle de vision).

- Modèles multi-vues : Cette méthode cherche à encoder différentes vues d'un objet, par exemple en appliquant une ACP (Analyse en Composantes Principales) sur la représentation matricielle d'une image, et donc générer la même image dans un espace de dimension moindre.

2.2.2 Sélection des caractéristiques (Classification)

L'étape qui vient après la représentation des objets est la sélection des caractéristiques susceptibles de distinguer un objet de la meilleure façon d'autres objets. C'est donc la sélection des caractéristiques qui le représentent au mieux. Les caractéristiques les plus courantes sont les suivantes :

- **Bordures :** Très faciles à distinguer par l'œil humain, elles permettent de distinguer les objets entre eux si ceux-ci ne sont pas en occlusion. Elles sont beaucoup moins sensibles à la variation de la lumière ou la couleur, ce qui facilite le tracking (Yilmaz et al, 2006).
- **Flux optique :** Phénomène qui représente le mouvement apparent des objets dû au changement de point de vue relatif d'une personne. On cherche donc à identifier les pixels dans chaque image de la vidéo pour mieux étudier le mouvement. Ces pixels malgré leur mouvement ont une luminosité identique, c'est ce qui permet de mieux visualiser le mouvement (Sun et Srdjan, 2016).
- **Couleur :** L'inconvénient avec les couleurs est qu'elles sont très sensibles au changement de luminosité. La couleur est généralement représentée en RGB (Degrés de rouge, vert et bleu), ou par HSV (Teinte, saturation, valeur) (Yilmaz et al., 2006).
- **Texture :** La texture des objets peut s'avérer utile pour distinguer les objets. En observant les irrégularités dans une image on peut déduire qu'il s'agit d'objets différents (Laws, 1980).

2.2.3 Détection d'objets

Après avoir correctement représenté l'objet, il faut désormais détecter la présence de cette représentation dans la vidéo à travers le Tracking. Voici quelques méthodes utilisées :

- **Détection de points :** Cette méthode consiste en la détection des points d'intérêt de l'image. On prend généralement les points d'angle comme point d'intérêt, un point d'angle étant un point dont le voisinage possède deux bordures dans des directions différentes, ou alors l'extremum d'une forme curviligne. Les points d'intérêt choisis se doivent d'être le moins sensibles aux changements de luminosité (Willis et Sui, 2009).

Pour la détection de points d'intérêt plusieurs méthodes existent, dont celle appelée « **Détecteur de Harris** », qui sera présentée plus en détail dans 4.1.1.

- **Soustraction d'arrière-plan :** Il s'agit de séparer les objets apparents au premier plan de ceux apparaissant dans l'arrière-plan de l'image. On peut ainsi détecter les mouvements des objets en termes de déviation par rapport au modèle référence en appliquant sur l'image un modèle de l'arrière-plan en question (Piccardi, 2004).
- **Segmentation :** Il s'agit de découper l'image et de la séparer en différentes régions (Ou segments) en fonction de la texture ou de la couleur… (Figure 13). Cela s'avère utile pour la détection d'objets ou de frontières entre objets. L'une des méthodes utilisées pour cela est dite « contours actifs » (Figure 14), et consiste à initialiser un genre de séparateur sur la bordure d'un objet dans l'image, puis se déplace et s'étend selon la texture et la couleur des pixels pour tenter de séparer l'image en régions (Saha et al., 2010).

Figure 13 - Segmentation d'image (Dwivedi, 2019)

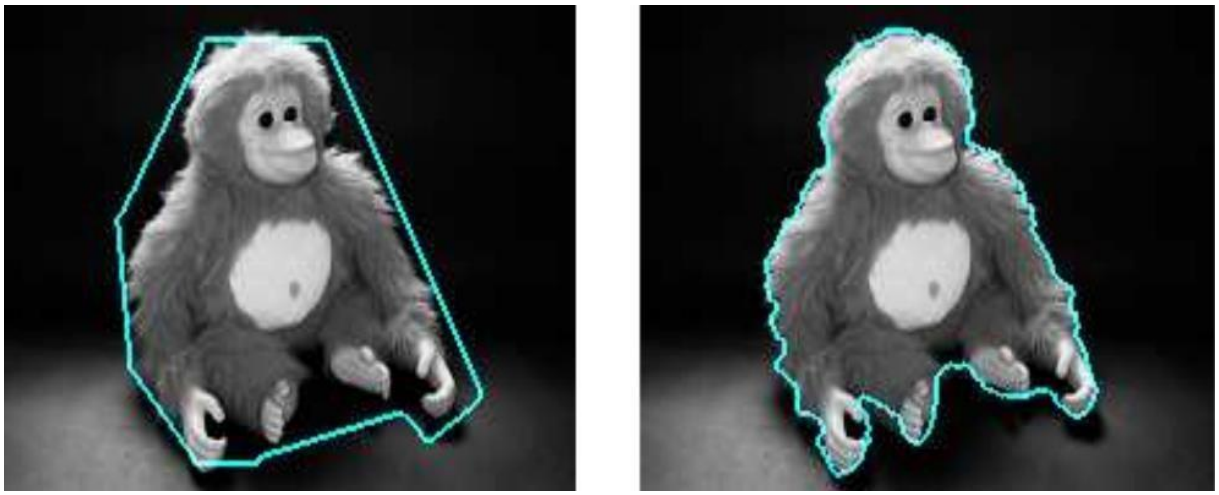

Figure 14 - Méthode Contours actifs (Dambreville S., et al., 2008)

- **Apprentissage supervisé :** On peut naturellement concevoir un modèle pour la détection capable de reconnaître des objets en l'entraînant avec un nombre suffisant

d'images labelisées au préalable. Diverses méthodes de classification peuvent être utilisées, comme le SVM, les arbres de décision, les réseaux de neurones…etc. (Yilmaz et al., 2006).

2.2.4 Tracking d'objets

Une fois toutes ces étapes accomplies, nous pouvons désormais passer à l'étape finale qui est le tracking. Les algorithmes qui effectuent le Tracking d'objets peuvent être répartis en trois types :

A. **Tracking de point :** Utilisé lorsque les objets sont représentés par des points. Il s'agit d'associer des points entre les images de la vidéo, en fonction de l'état (Position et mouvement) de l'objet dans chaque image. La fonction qui associe à un objet *i* dans l'image t-1 un objet *j* dans l'image t (Yilmaz et al., 2006) est appelée « coût de correspondance ». On cherche donc à minimiser cette fonction, pour tenter de prédire le point dans l'image t qui correspond le plus vraisemblablement à l'objet en question dans l'image t-1. Ceci se fait en respectant quelques contraintes, comme le fait qu'un point ne doit pas changer drastiquement de position entre deux images, ou que deux points d'un même objet auront toujours la même distance dans toutes les images (Rigidité de l'objet) …etc. Un exemple de modèle qui a été conçu off-line pour prévoir le déplacement de personnes, est traceur de chemin sur une image (Scovanner et Tappen, 2009), qui respecte les contraintes suivantes :

1- La différence entre la position d'une personne entre deux images est moindre.
2- La vitesse d'une personne et la direction prise sont identiques à un certain degré entre deux images.
3- Le chemin parcouru devrait amener la personne à l'endroit escompté.
4- Le mouvement des personnes tend à éviter les collisions avec les autres personnes.

Ce modèle s'avère bien sur très approximatif dans la plupart des cas, mais peut s'avérer efficace dans certains cas comme le suivant dans la figure 15, où le mouvement réel est en noir, et le mouvement prédit en rouge :

Figure 15 - Traque mouvement piétons (Scovanner et Tappen., 2009)

Pour minimiser le coût de correspondance on peut utiliser la méthode :

- Déterministe : L'approche déterministe revient à énoncer toutes les correspondances entre les points, et choisir la minimale à l'aide d'une méthode gloutonne par exemple (Yilmaz et al., 2006).
- Stochastique : L'approche stochastique prend aussi en compte les bruits dans les mouvements et les perturbations éventuelles avec une certaine probabilité pour concevoir le modèle. Elle est donc plus précise mais plus complexe (Chau et al., 2013).

B. **Tracking de noyau :** Méthode utilisée lorsque l'objet est représenté par un modèle, dans le cas où l'objet est représenté par son apparence (Voir 2.2.1.), ou par une forme géométrique (Voir 2.2.1).
   - Modèle : Lorsqu'il s'agit de traquer un seul objet, on peut former un modèle à partir de la couleur et l'intensité de l'image, puis on cherche ce modèle dans chaque image composant la vidéo, en le mettant à jour à chaque étape. Quand il s'agit par contre de traquer plusieurs objets, on a souvent recours à la méthode qui consiste à créer pour chaque objet une couche, qui représentera le modèle qui s'occupera de traquer l'objet en question de la même manière (Adaptive Vision, Template Matching, 2016).
   - Multi-Vues : Cette méthode cherche à résoudre le problème de perception de l'objet. Si un objet change d'orientation dans l'image on pourrait perdre sa trace lors de la traque. Pour pallier cette difficulté on peut reproduire l'image d'un objet sous un certain point de vue à partir d'un autre en utilisant les valeurs propres de la matrice de diffusion (Réduction de dimension de l'espace) (Black et Jepson, 1998).

C. **Tracking de silhouette :** Utilisé lorsque l'objet est représenté à l'aide d'un squelette, silhouette, contour ou autre forme d'articulations. Il se fait comme suit :
   - Matching de forme : De la même manière que le tracking sur les objets représentés en modèle, on cherche à établir une forme, généralement à base des bordures des objets, dans chaque image de la vidéo. On détecte ensuite celle qui lui correspond le plus dans l'image suivante, en mettant à jour la forme pour répondre aux changements de luminosité…etc. (Veltkamp, 2001).
   - Evolution de contour : Très similaire à la segmentation (2.2.3), on initialise dans une image une forme superposée aux contours d'un objet, puis à chaque image cette forme est adaptée au changement de position des contours de l'objet selon sa vitesse et son déplacement. Cette méthode est insensible aux variations de luminosité car elle ne s'intéresse pas aux couleurs mais aux bordures, ce qui la rend plus robuste (Jacob et Anitha, 2012). Elle est toutefois handicapée par le fait qu'elle ne puisse traquer plus d'un objet à la fois, et aussi du fait que le contour initial doive être assez proche de l'objet (Bendaoud, 2017).

L'ensemble de ces étapes est résumé dans la figure 16.

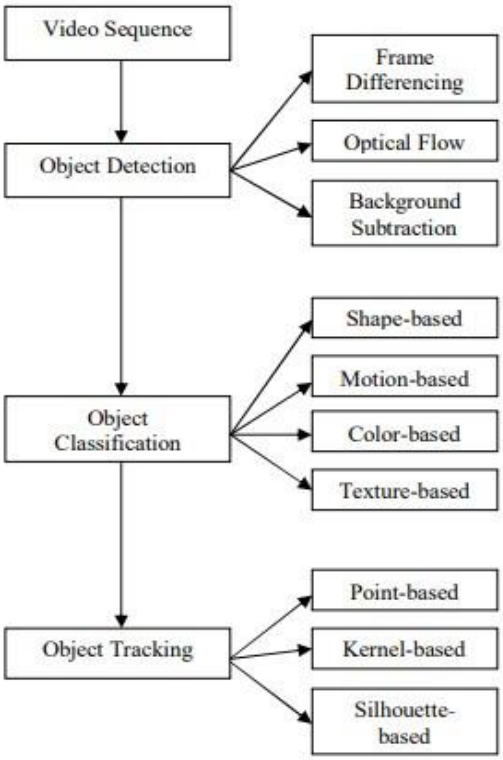

Figure 16 - Etapes du Tracking (Lee et Yu, 2011)

2.3 Récapitulatif des méthodes de Tracking

Le tableau 1 présente un récapitulatif des différentes méthodes de Tracking présentées dans ce chapitre. Pour chaque méthode, nous présentons les cas d'utilisation auxquels elle peut répondre, le type d'approche qu'elle suit, ses avantages et ses inconvénients. En effet, certaines méthodes s'avèrent efficaces quand des conditions sont définies, comme surtout la façon de représenter les objets. Mais encore, l'approche à suivre, qu'elle soit déterministe ou stochastique…etc. peut influer sur le choix de la méthode du Tracking à appliquer. Bien entendu, les avantages et les inconvénients diffèrent entre les méthodes. Tous ces critères sont présentés dans le tableau 1.

Tableau 1 - Comparaison des méthodes de Tracking

Type	A utiliser quand	Approche	Convient à	Avantages	Inconvénients
Tracking de points	Les objets sont représentés par des points	Déterministe	Objets dont la trajection est facilement prévisible	Peut être exécuté parallèlement pour chaque objet	Les mouvements complexes ne sont pas pris en compte pour la prédiction
		Stochastique	Bruit dans les mouvements	Prédiction plus précise du déplacement	Modèle plus complexe à concevoir
Tracking de noyaux	Les objets sont représentés par des formes primitives	Modèle	Objets ne changeant pas drastiquement de position, et ne changeant pas d'apparence	Simple à mettre en place	Peut consommer beaucoup de temps pour les modèles complexes, gère mal les occlusions
		Multi-Vues	Objets avec plusieurs apparences selon le point de vue	Flexibilité concernant l'apparence de l'objet	Approche plus complexe à concevoir que les modèles
Tracking de silhouettes	Les objets sont représentés par leurs silhouette ou contour	Matching de formes	Objets devant être représentés complètement, par exemple quand la couleur est utilisée comme caractéristique	Possibilité de combiner avec la reconnaissance d'objets	Peut consommer beaucoup de temps
		Evolution de contours	Objets devant être représentés uniquement par leurs contours	Insensible aux variations de la luminosité	Gère mal les occlusions

2.4 Conclusion

Comme nous avons pu le voir tout au long de ce chapitre, le tracking peut être réparti en plusieurs types, en fonction de la façon de placer la caméra durant l'expérience RA. Ensuite, nous avons constaté que le tracking comporte plusieurs étapes. La première, qui est la représentation de l'objet peut se faire de deux manières différentes, par la forme ou par l'apparence. Ensuite vient la sélection des caractéristiques, qui consiste en la sélection des caractéristiques susceptibles de différencier les objets et de les distinguer, telles que la couleur ou la texture…etc. La détection d'objets vient après ça, où diverses méthodes ont été proposées (Détecteur de Harris, contours actifs…), selon la méthode choisie de représentation des objets Enfin vient la méthode du tracking, qui elle aussi dépend de la représentation des objets, et qui peut se faire de façon déterministe ou stochastique.

Nous avons donc vu dans ce chapitre en quoi consistait le tracking en façon générale, et en réalité augmentée plus particulièrement. Nous avons détaillé ses étapes, ainsi que les différentes méthodes possibles pour effectuer le tracking. Nous avons conclu par un tableau récapitulatif et comparatif de celles-ci.

Dans le chapitre suivant, nous parlerons de trois méthodes où le Deep Learning a été utilisé pour effectuer du tracking, pour certains cas sur des vidéos diverses, et pour d'autres sur des expériences de RA.

3. Chapitre III : Deep Learning pour le Tracking

La plupart des méthodes de tracking dans la littérature, utilisant le deep learning, ne sont pas destinées à la Réalité augmentée, mais au simple suivi d'objets. Il est toutefois clair qu'une fois ces méthodes établies il est aisé de les utiliser pour les systèmes RA, à condition de bien surveiller leur temps d'exécution ainsi que leur consommation d'énergie, qui sont deux facteurs importants pour le matériel RA. Parmi les méthodes qui sont spécifiquement dédiées à la RA, nous avons étudié une des plus citées, qui est DeepAR. Ensuite, nous avons étudié deux autres méthodes de tracking qui sont GOTURN et Deep SORT. Elles ne sont certes pas appliquées à la Réalité Augmentée, mais elles présentent néanmoins d'intéressants avantages quant à l'efficacité de l'exécution.

La première méthode de DeepAR est spécialement dédiée au Tracking en RA, en utilisant des réseaux de neurones, ainsi qu'un détecteur d'angle qui sert à la détection d'objets. La deuxième méthode (GOTURN) quant à elle est une méthode Open Source utilisant des réseaux de neurones convolutifs, de façon à permettre la prédiction des positions des objets dans des images successives. La troisième et dernière méthode que nous présentons est Deep SORT. Elle permet le tracking simultané de plusieurs objets, en utilisant un procédé souvent employé en imagerie, qui est le filtre de Kalman. Les deux méthodes DeepAR et Deep SORT permettent d'effectuer le tracking de plusieurs objets de façon simultanée, ce qui n'est pas le cas pour GOTURN, qui n'est valable que pour un seul objet à traquer.

3.1 DeepAR

Nous allons énoncer le principe d'un algorithme utilisant le Deep Learning pour le tracking en RA (Akgul et al., 2016). On distingue dans les algorithmes de Tracking basés sur les modèles deux étapes majeures. La première est la détection de la cible dans la vidéo en entrée, et la deuxième est le suivi de la cible durant toute la vidéo jusqu'à ce que celle-ci disparaisse. Le tracking est généralement moins coûteux en puissance de calcul que la détection, car la détection consiste à reconnaître l'objet dans le champ de vision de l'utilisateur à un instant donné et le positionner, tandis que le Tracking consiste à le suivre et à tracer son parcours durant toute l'expérience RA.

Avant d'énoncer le fonctionnement de la méthode, nous présentons un procédé de détection ainsi que l'architecture du réseau utilisé :

3.1.1 Détecteur de Harris

Cette méthode a été développée par (Harris et Stephens, 1988). Elle permet de détecter de manière très efficace les points d'angle dans une image. L'intuition derrière la méthode est très simple : Dans une image, un angle peut être reconnu par une zone où la forme tracée dessus change de direction de manière drastique, l'algorithme fait dont déplacer au niveau de toute l'image une grille de détection, sur les deux axes X et Y. Cet angle représentera dans l'image un point critique, qui associé à d'autres angles dans l'image, pourront former les contours de l'objet cible à traquer. Pour chaque zone balayée, il calcule la variation dans la direction de la zone tracée (comme dans la figure ci-dessous), et lorsque la variation est assez grande (selon

un certain seuil) sur un seul axe, on détecte une bordure, tandis que si c'est sur les deux axes que la variation se fait observer, nous avons un point d'angle (Figure 17).

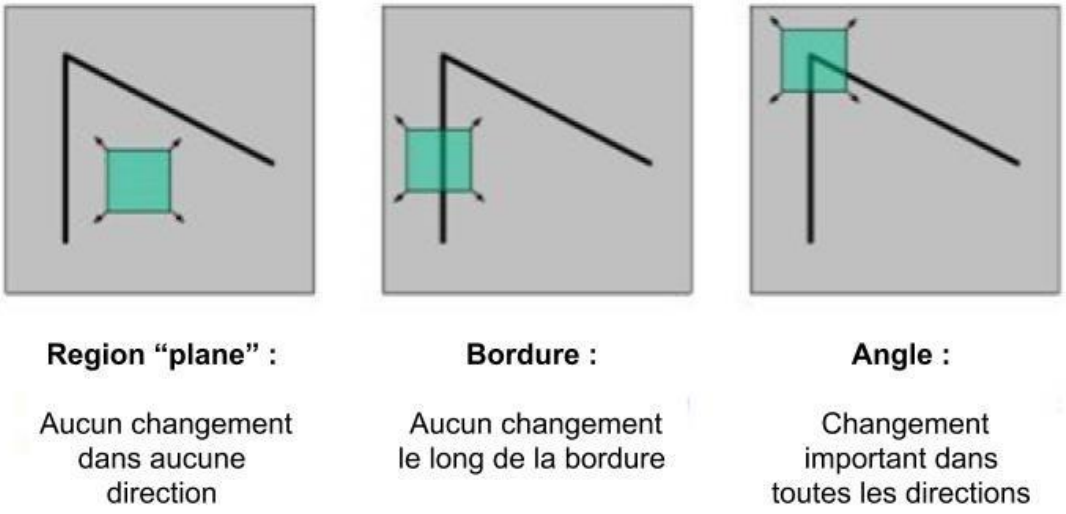

Figure 17 - Détecteur de Harris (Jieyang Hu, 2015)

Après cette approche intuitive, Harris a modélisé cela sous la forme de l'équation mathématique (1) :

$$E(u, v) = \sum_{x,y} w(x, y)\ [I(x + u, y + v) - I(x, y)]^2 \qquad (1)$$

Où :

- x et y sont les coordonnées actuelles de la grille
- u et v sont les déplacements par rapport à la position actuelle
- w(x,y) est une fonction représentant la position de la grille
- I(x,y) l'intensité de l'image à la position actuelle
- I(x+u, y+v) l'intensité de l'image après avoir appliqué le changement de direction u, v

La fonction E cherche dont à déterminer la différence entre les intensités avant et après les variations. Dans le cas où cette fonction a une grande valeur, cela veut dire simplement que la différence entre les intensités est grande, et donc que la zone entourée est un angle selon la définition précédente.

Il s'agit donc de maximiser la fonction E afin de déterminer les directions qui provoquent le plus grand changement d'intensité. Pour ce faire, et à cause de la complexité de la fonction, on passe par le développement de Taylor. Après quelques calculs nous obtenons la formule (2) sous forme matricielle :

$$u$$

$$E(u, v) = [u \quad v] \, \mathrm{M} \begin{bmatrix} u \\ v \end{bmatrix} \qquad (2)$$

Où la matrice M est (3) :

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \qquad (3)$$

Où Ix et Iy sont les dérivées de l'image dans les deux directions X et Y respectivement. Ensuite à partir de cette matrice, on calcule ce score (4), qui représente une façon mathématique de déterminer quelle région produit de grandes variations dans les directions :

$$R = \det(M) - k(Trace(M))^2 \qquad (4)$$

Où det(M) est le déterminant de M, et trace(M) sa trace, et k un hyperparamètres du modèle fixé à l'avance.

A partir de ce score, la décision se fait comme suit :

- Si |R| est petit, la région ne contient ni angle ni bordure.
- Si R<0, la région contient une bordure.
- Si R est grand, la région contient un point d'angle.

Le détecteur de Harris a été implémenté dans la bibliothèque OpenCV pour Python (https://pypi.org/project/opencv-python/).

3.1.2 AlexNet :

AlexNet est le modèle qui a remporté le prix ILSVRC 2012, sur le célèbre dataset ImageNet. C'est une compétition consistant à proposer des modèles qui effectuent la classification de différents objets sur un dataset totalement épars, et où l'objectif est d'obtenir le score le plus faible d'erreur sur les prédictions.

Le modèle prend en entrée des images de diverses tailles, les redimensionne et les découpe avant d'avoir des images de 256x256 contenant uniquement l'objet majeur à classifier. L'architecture du réseau consiste en 8 couches au total (Figure 18) : 5 couches de convolution, qui servent rappelons-le à extraire les caractéristiques les plus significatives de l'image, et 3 couches totalement connectées (Krizhevsky et al., 2012).

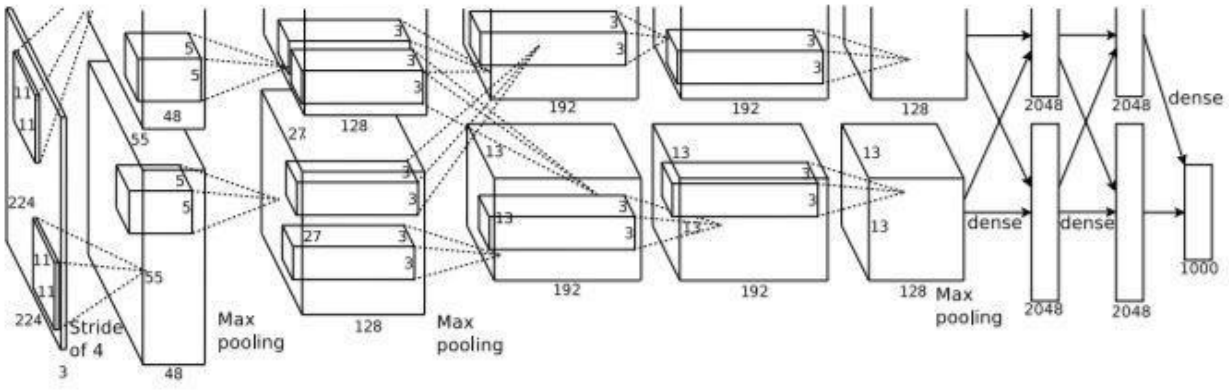

Figure 18 - Architecture AlexNet (Krizhevsky et al., 2012)

La fonction d'activation au niveau des neurones est le ReLU, car avec cette fonction les CNN effectuent l'entraînement beaucoup plus vite qu'avec la tangente hyperbolique par exemple. Le réseau a été entraîné pour des raisons de rapidité et de disponibilité de la mémoire sur deux GPU, c'est pourquoi dans le schéma qui suit les traitements sur les couches de convolution se font en parallèle sur deux niveaux.

Pour la réduction du **Sur-Apprentissage**, c'est-à-dire le fait que le modèle soit efficace uniquement sur les données avec lesquelles il a été entraîné, mais qu'il ne le soit pas avec de nouvelles données, plusieurs techniques ont été utilisées par les développeurs :

- **Augmentation de données :** Augmenter les données utilisées (les entrées) en appliquant diverses transformations. Par exemple, à partir d'une image du dataset, produire l'image miroir, cela permettra de rendre le modèle plus flexible en changeant le point de vue du même objet. Une autre transformation possible serait de produire des rognages différents de la même image, mais s'arranger pour que l'objet à reconnaître y figure toujours.
- **Elimination (Drop-out) :** L'élimination consiste à éliminer de temps à autre, suivant une probabilité de 0.5, un neurone du réseau (Figure 19). Ce neurone ne participera donc plus ni à la propagation avant, ni à la rétropropagation. Cela réduit le sur-appretissage du modèle comme l'indique (G.E. Hinton, 2012). Il est vrai que l'élimination peut retarder la convergence par un facteur de 2, mais elle empêche le modèle de tomber dans le sur-apprentissage.

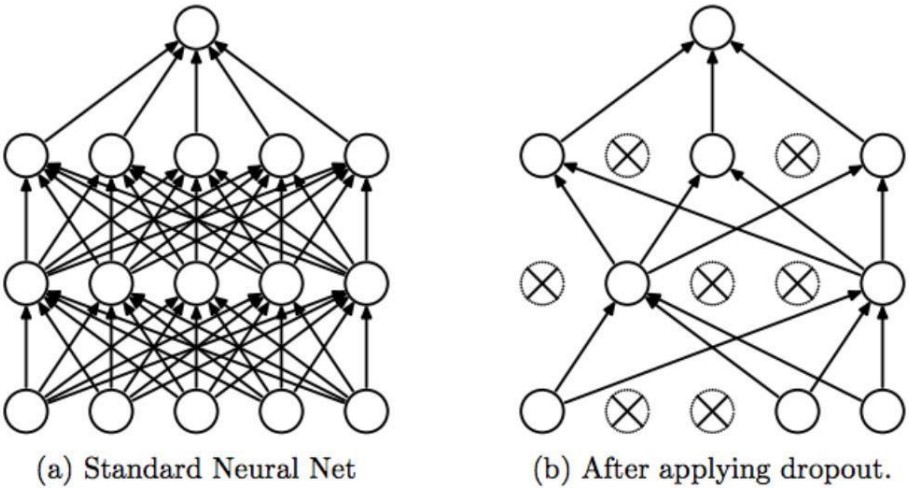

Figure 19 - Dropout (Srivastava et al., 2014)

3.1.3 Fonctionnement

Le modèle de détection sera entraîné de la manière suivante :

1- Les images pour l'entraînement en entrée sont passées à travers un extracteur de points d'intérêt dans chaque image.

2- Les images contenant les points d'intérêt sont ensuite passées à travers un extracteur de caractéristiques.

Ainsi le modèle sera capable d'établir une correspondance entre une image et les points d'intérêt la composant et leurs caractéristiques. Lors de la détection pour une image, celle-ci est passée à travers le mêmes extracteur de points d'intérêt et de caractéristiques, en produisant ainsi un Matching entre l'image en entrée et les images étudiées lors de l'entraînement. Ce dernier consiste à donner en entrée du réseau un grand nombre d'images, avec les points d'intérêt spécifiés pour chacune d'entre elles (Grâce au détecteur de Harris exécuté au préalable sur chaque image), puis corriger les coefficients du réseau en calculant l'erreur associée à chaque prédiction de points d'intérêt. Ce processus est répété plusieurs fois. Après cette étape, le modèle est capable d'effectuer ses propres prédictions sur une image donnée en entrée. L'ensemble de ces étapes est illustré dans la figure 20.

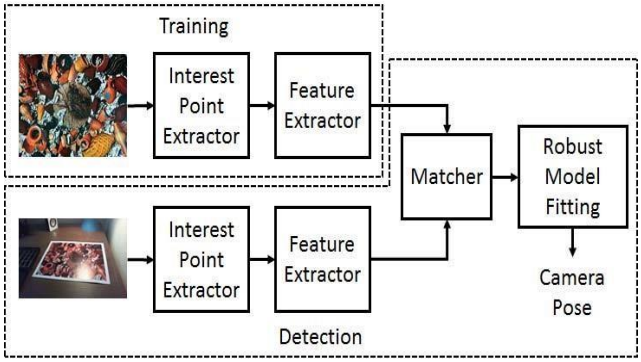

Figure 20- Etapes de détection en Tracking RA (Akgul et al., 2016)

Le procédé de détection que l'on vient de présenter est ensuite appliqué aux images d'entraînement. La méthode DeepAR consiste dans un premier temps à extraire depuis les images en entrée les points d'intérêt (points d'angle) en utilisant le détecteur de Harris. Puis, des grilles de 15x15 pixels sont extraites autour de chaque point d'intérêt. Afin de prendre en compte les différences dans la représentation, on génère certaines images à partir de l'image originale en simulant des variations de luminosité et d'échelle, afin de permettre au modèle d'apprendre le plus de formes possibles du point d'intérêt. Chaque grille représentera ensuite une classe (ou un objet), ainsi le modèle devrait faire la correspondance entre la grille et le point d'intérêt correspondant, que l'on aura déjà intégré dans l'entraînement du modèle. Pour l'entraînement, ont été utilisées des grilles de taille 50. Chaque classification durant 100 itérations, 80% des données ont été dédiées à l'entraînement et le reste pour le test, sur le réseau AlexNet. La figure 21 résume le fonctionnement de cette méthode.

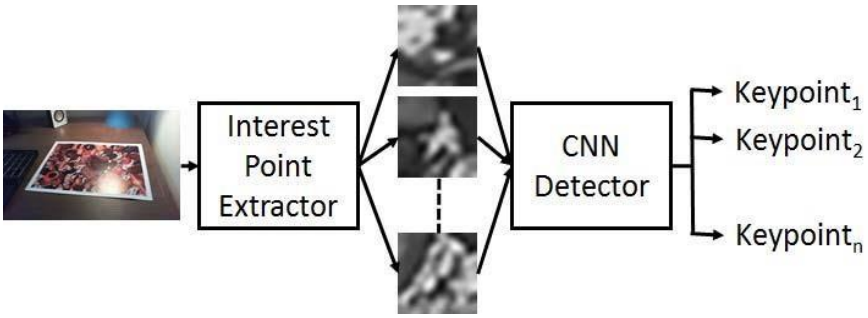

Figure 21 - Fonctionnement DeepAR (Akgul et al., 2016)

L'algorithme proposé a été testé et comparé à des méthodes de la littérature utilisant des données réelles. Deux types d'images ont été utilisées représentant des cailloux et des poteries afin d'entraîner le détecteur. Pour chaque image, environ 6000 images sont extraites en modifiant l'ancle de vue, la luminosité…etc. comme l'indique la figure 22.

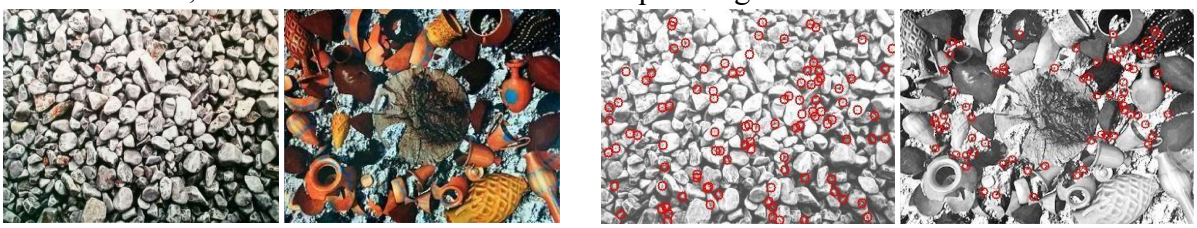

Figure 22 - Images utilisées pour l'entraînement de DeepAR (Akgul et al., 2016)

L'algorithme a été testé contre un algorithme de détection ORB (Rublee et al., 2011), en utilisant une métrique connue sous le nom d'erreur de reprojection, qui correspond à la distance d'image entre un point projeté et un point mesuré, c'est-à-dire entre la position prédite des objets par l'algorithme et celle réelle. Les résultats montrent que l'algorithme DeepAR est plus efficace que ORB pour ce genre de données, comme l'indique le tableau 2.

Tableau 2 - Performances DeepAR - ORB (Akgul et al., 2016)

	DeepAR	ORB
Cailloux 1	0.852	1.542
Cailloux 2	3.245	1.163
Cailloux 3	0.866	4.327
Poteries 1	0.821	0.754
Potterie 2	52.087	53.978
Potterie 3	0.992	54.389

Toutefois, comme constatons que les données utilisées pour l'entraînement ainsi que pour le test ne correspondent pas tout à fait à l'environnement de la Réalité Augmentée. Certes, les images utilisées, contenant une multitude d'objets divers (Figure 22), rendent la détection ardue. Néanmoins, il faudrait tester l'algorithme sur une vidéo de paysage comme ça peut être le cas dans la vision d'un utilisateur lors d'une expérience RA afin de tester l'efficacité.

3.2 GOTURN

GoTurn est un autre algorithme de Tracking utilisant le Deep Learning, développé par David (Held et al., 2016). Ce qui rend l'algorithme célèbre en plus de son efficacité, est le fait que contrairement à beaucoup d'autres l'entraînement du modèle se fait en offline. Pendant l'exécution il n'est pas nécessaire d'entraîner le modèle à chaque fois, ce qui diminue considérablement le temps d'exécution.

3.2.1 CaffeNet

CaffeNet est en fait la version GPU unique de AlexNet, c'est-à-dire qu'au lieu de fonctionner sur deux GPU, cette architecture est mise en place sur un seul GPU. Les deux niveaux parallèles sont donc fusionnés en un seul comme le montre la figure 23.

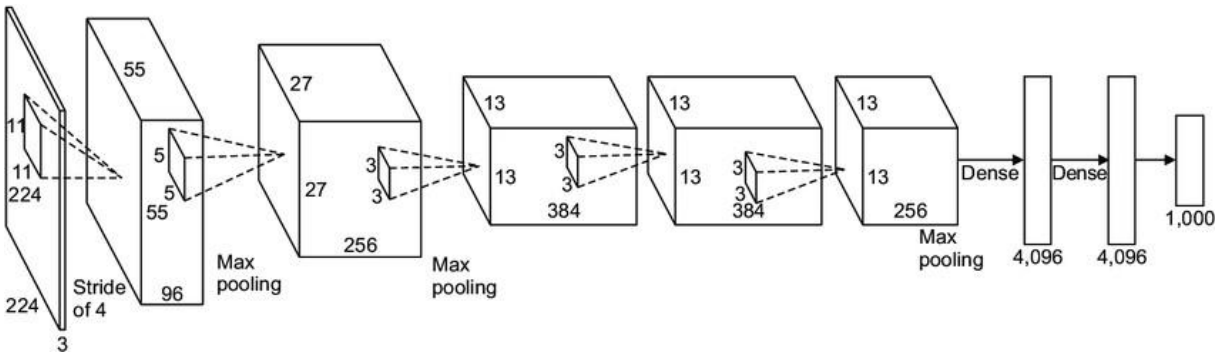

Figure 23 - Architecture CaffeNet (Hyung Lee, 2018)

3.2.2 Fonctionnement

La méthode consiste en un réseau CNN qui prend en entrée deux images (frames) de la vidéo : L'une ayant la cible entourée par un rectangle (donc traquée) et l'autre image contenant la cible que l'on cherche à entourer de la même manière que la première image. En fait, la première image représente l'image précédant la deuxième. On cherche donc à prédire la position de l'objet au moment t à partir de la position précédente au moment t-1. L'entraînement se fait donc en soumettant au réseau deux images successives chacune ayant la cible correctement entourée. La sortie du réseau représente les coordonnées 2D du rectangle entourant la cible.

Le réseau possède deux entrées (Figure 24) :

1. L'image actuelle dont il faut cerner la cible : Celle-ci passe à travers 5 couches de Convolution, qui permettent de réduire l'image, ensuite à travers 3 couches totalement connectées qui permettent d'effectuer l'apprentissage. Les 5 couches de convolution sont les couches du réseau CaffeNet.
2. L'image précédente : Celle-ci passe à travers 5 autres couches de Convolution puis par les mêmes 3 couches totalement connectées. Le résultat de ces 3 couches sera l'image actuelle avec la cible entourée.

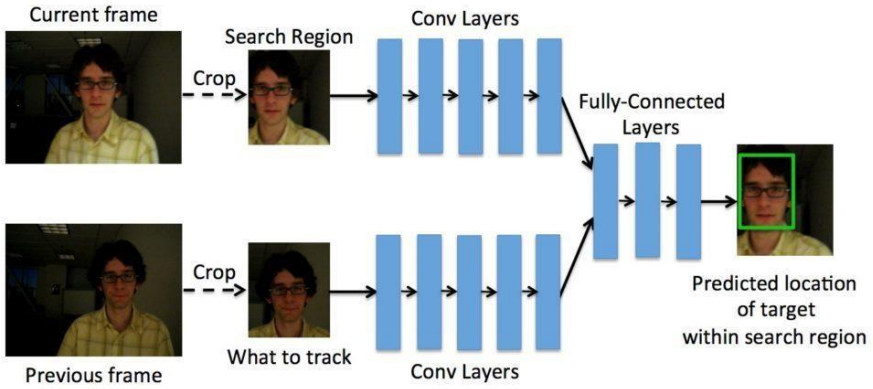

Figure 24 - Architecture GOTURN (LearnOpenCv, 2018)

Le réseau est entraîné sur des vidéos provenant d'ALOV300++ (Smeulders et al., 2014), qui est une collection de 314 séquences vidéo. Parmi celles-ci sont retirées 7 vidéos qui se chevauchent avec celles de test, ce qui laisse 307 vidéos à utiliser pour l'entraînement. Dans cet ensemble de données, environ une image sur cinq de chaque vidéo a été étiquetée avec l'emplacement d'un objet suivi. Ces vidéos sont généralement courtes, allant de quelques secondes à quelques minutes. Ces vidéos sont divisées en 251 pour l'entraînement et 56 pour la validation (hyperparameter tuning). Le dataset d'entraînement se compose d'un total de 13 082 images de 251 soit une moyenne de 52 images par objet. Le set de validation comprend 2 795 images de 56 objets différents.

Après avoir choisi les hyperparamètres, le test est effectué sur un set constitué des 25 vidéos du challenge de Tracking de VOT (Kristan et al., 2014), qui est une référence de tracking qui permet de comparer un tracker à une grande variété de trackers de pointe. Les trackers sont évalués selon deux mesures de suivi standard : la précision (A) et la robustesse (R), qui vont de 0 à 1. Sont calculées également l'erreur de précision (1 - A), l'erreur de robustesse (1 - R), et l'erreurs globale 1 - (A + R)/2. Chaque image de la vidéo est annotée avec un certain nombre d'attributs : occlusion, changement d'éclairage, changement de mouvement, changement de taille et mouvement de la caméra. Les trackers sont également classés en précision et en robustesse séparément pour chaque attribut, et les classements sont ensuite mis en moyenne sur l'ensemble des attributs pour obtenir une précision moyenne finale et le classement de la robustesse pour chaque tracker. Les classements de précision et de robustesse font l'objet d'une moyenne pour obtenir un classement général moyen.

Les résultats obtenus montrent que GOTURN a une bonne robustesse et une excellente précision. Par ailleurs, le classement global (calculé comme la moyenne de la précision et de la robustesse) surpasse tous les autres trackers sur ce point. L'importance de l'entraînement hors ligne pour améliorer les performances de suivi a aussi été démontré. En outre, ces Les résultats ont été obtenus après une formation sur seulement 307 courtes vidéos. Le tracker peut cependant s'avérer peu efficace dans certains cas en raison d'occlusions ou d'un sur-ajustement aux objets de l'ensemble de formation.

3.3 Deep SORT

Un autre algorithme est Deep Sort (Deep Simple Real Time Tracker) qui est une extension de l'algorithme SORT en utilisant l'apprentissage profond. SORT (Bewley et al., 2017) est un algorithme de Tracking de multiples objets simultanément. L'algorithme se fait en deux temps : Un premier entraînement Offline se fait en utilisant un réseau CNN, dont le but est d'apprendre la détection de d'objets à travers un dataset donné, et la deuxième étape se fait lors de l'application où le tracking se fait en temps réel.

3.3.1 Filtre de Kalman

L'algorithme fait appel à une méthode appelée **filtre de Kalman** en imagerie, qui sert en général d'estimation de mesure. Dans notre cas en particulier, elle servira à prédire la position actuelle d'un objet à partir de ses positions précédentes. Le principe du filtre de Kalman est le suivant :

Supposons que l'on veuille suivre le déplacement d'un objet dans une série d'images, et que l'on ait un détecteur qui effectue cette tâche avec un taux acceptable d'erreur. La situation idéale afin de prédire correctement la prochaine position de l'objet serait dans le cas où la vitesse de ce dernier reste constante. Notre modèle fonctionne en considérant cette propriété comme vérifiée, et fonctionne donc pour la situation idéale. Néanmoins, en pratique c'est rarement le cas, car nous avons deux types de bruit :

1. **Bruit de processus :** Bruit correspondant à la variation de la vitesse de l'objet dans les images.
2. **Bruit de mesure :** Etant donné que même le détecteur correspondant au modèle idéal a aussi sa marge d'erreur, on peut y avoir un bruit que l'on appellera bruit de mesure.

La figure 25 représente le processus de prédiction avec un filtre de Kalman :

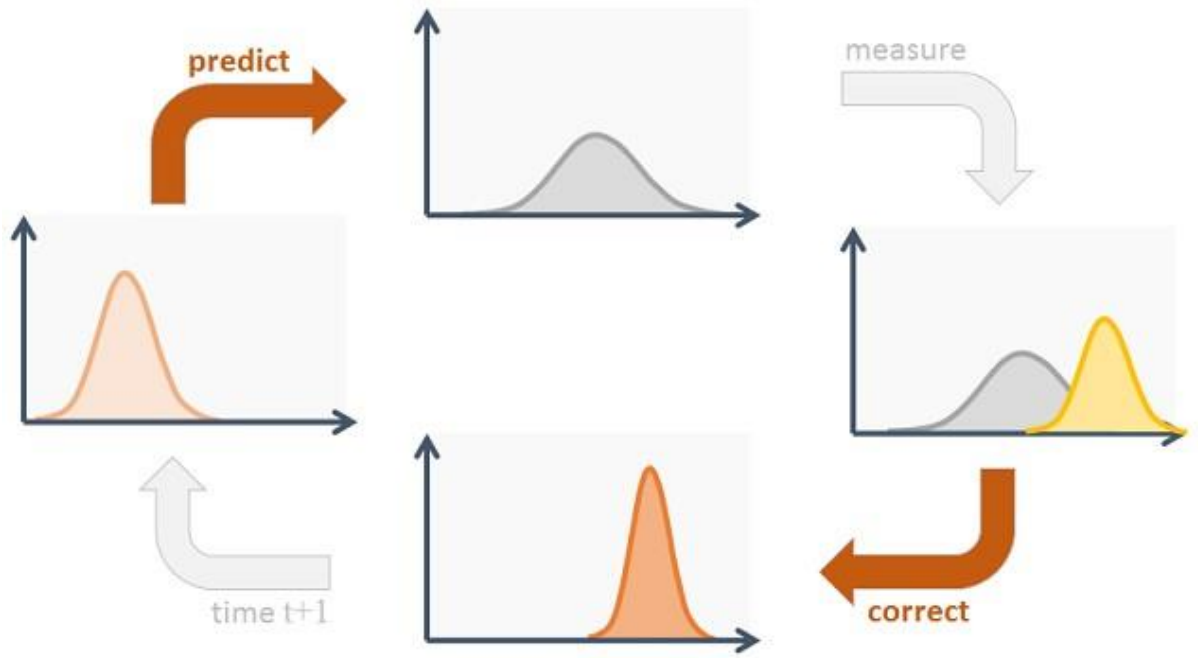

Figure 25 - Fonctionnement Filtre de Kalman (Jurić, 2015)

3.3.2 Fonctionnement

La figure 28 présente le principe du filtre de Kalman, qui consiste tout d'abord à effectuer une détection, ensuite mesurer les bruits correspondants à celle-ci, effectuer les corrections sur la détection, et prédire la détection au moment suivant, et ainsi de suite de façon récursive. Dans le cas de détection d'objets, les bruits énoncés précédemment suivent souvent une distribution Gaussienne, ce qui permet à la méthode de fonctionner de manière efficace. Le détecteur fournit donc une détection de manière tout à fait normale, ensuite le filtre agit sur la correction de cette détection en modélisant les bruits associés, ce qui procure bien entendu plus de précision. Le filtre associe à chaque détection un 'Track' contenant des informations relatives à la détection,

dont par exemple la dernière fois que l'objet en question a été correctement détecté dans la vidéo, ce qui permet donc de deviner les objets qui ont quitté la scène si ceux-ci n'ont pas été détectés depuis un certain laps de temps.

L'association entre le 'Track' et la détection se fait en utilisant un algorithme d'association de données appelé '**Hungarian Algorithm'**, proposé en 1955 par **Harold Kuhn.** L'aspect Deep Learning dans ce procédé réside dans le fait qu'un réseau CNN est mis en place de manière offline pour entraîner le modèle à reconnaître des personnes sur un dataset contenant plus de 1 millions d'images de 1261 piétons. Le réseau consiste en 2 couches de Convolution, suivies de 6 blocs résiduel (Un **bloc résiduel** est une succession de couches d'un réseau de neurones où certaines connexions entre deux couches successives sont omises, c'est-à-dire qu'une couche peut être connectée avec une autre couche qui ne lui est pas directement voisine. L'intérêt est d'accélérer l'apprentissage et la précision).

Le réseau comptabilise en tout plus de 2 500 000 paramètres, et a été entraîné sur des processeurs graphiques Nvidia GeForce GTX 1050 mobile GPU. Le tableau 3 illustre les caractéristiques du réseau.

Tableau 3 - Architecture CNN DeepSORT (Wojke et al., 2017)

Name	Patch Size/Stride	Output Size
Conv 1	3 × 3/1	32 × 128 × 64
Conv 2	3 × 3/1	32 × 128 × 64
Max Pool 3	3 × 3/2	32 × 64 × 32
Residual 4	3 × 3/1	32 × 64 × 32
Residual 5	3 × 3/1	32 × 64 × 32
Residual 6	3 × 3/2	64 × 32 × 16
Residual 7	3 × 3/1	64 × 32 × 16
Residual 8	3 × 3/2	128 × 16 × 8
Residual 9	3 × 3/1	128 × 16 × 8
Dense 10		128
Batch and l_2 normalization		128

3.4 Conclusion

Nous avons présenté dans ce chapitre trois méthodes de l'état de l'art, où le Deep Learning est utilisé pour effectuer du tracking. Pour chaque méthode, nous avons décrit la méthodologie proposée et suivie par les auteurs, ainsi que l'architecture du réseau de neurones utilisé.

Nous avons vu que la plupart des méthodes faisaient appel à des notions mathématiques (Méthode de Harris, Filtre de Kalman) pour améliorer l'algorithme de tracking. Nous avons vu aussi que les architectures des réseaux pouvaient être assez différentes les unes des autres, même si dans plusieurs cas, les chercheurs préfèrent utiliser des architectures déjà prêtes (AlexNet), qu'ils adaptent selon leur besoin.

Toutes les méthodes que nous avons citées utilisent le Deep Learning pour le Tracking d'objets dans une vidéo, mais la plupart d'entre-elles ne s'applique directement à la Réalité Augmentée. Il serait intéressant de tenter d'appliquer ces méthodes sur une expérience RA, en les implémentant directement sur le Traqueur au niveau du processeur du dispositif. La vidéo sur laquelle il faudra traquer les objets sera donc simplement le champ de vision de l'utilisateur, affiché à travers le HMD par exemple. Le Tracking se ferait donc en temps réel, en détectant et traquant l'objet à chaque instant de l'expérience.

La complexité de l'algorithme dépendra de l'équipement utilisé, par exemple si l'expérience RA est destinée aux smartphones, il faudra que l'algorithme ne consomme pas trop de ressources, et que l'architecture du réseau ne soit pas trop complexe.

Le principe reviendrait donc en plus de localiser les différents objets composant la scène durant l'expérience, à estimer les endroits susceptibles d'être augmentés par un objet virtuel. Par exemple, dans le chapitre 1.6.1., la veilleuse virtuelle est placée sur la surface plane qu'est la table, il faut donc que le système comprenne ou déduise au préalable la planitude de la surface en question. Ceci est déjà fait aujourd'hui de manière assez efficace, comme par exemple le ARCore de Google, comme le montre la figure 26 :

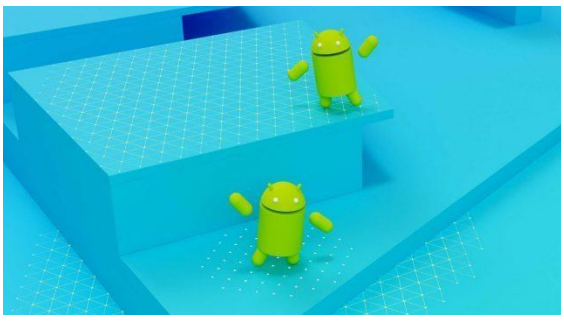

Figure 26 - Détection de surfaces planes par ARCore(Hruska, 2017)

Il est clair que dans certains cas, les formes à augmenter ne sont pas planes, mais peuvent avoir des formes curvilignes par exemple. Le traqueur devra prendre ça en considération. L'intérêt d'avoir recours au Deep Learning est justement de pouvoir spécifier les sorties du réseau à avoir. Dans ce cas par exemple, nous pouvons concevoir notre propre architecture de réseau, qui spécifiera en sortie les contours des objets, permettant ainsi de pourvoir y superposer les objets virtuels de manière plus précise.

Conclusion

La Réalité Augmentée est un des domaines de l'informatique les plus prolifiques de ces dernières années, elle commence petit à petit à occuper une place incontournable dans notre quotidien, quelque que soit le domaine où l'on évolue. Elle nous a permis de repousser les limites de temps et d'espace en immergeant les utilisateurs dans des environnements réels augmentés, qui leur donnent la sensation d'être littéralement transportés ailleurs en peu de temps. Par ailleurs, ce domaine a la capacité fondamentale d'influer sur de nombreux autres domaines, et de révolutionner la façon dont on côtoie l'éducation, le tourisme, la médecine…etc. Le tracking, quant à lui, est une partie importante de la RA. Il permet d'assurer une bonne fusion entre l'environnement physique dans lequel l'utilisateur se trouve, et les objets virtuels ajoutés par le système RA. Ayant de nombres types, cette étape fait pourtant face à de nombreux challenges, étroitement liés à ceux de la vision par ordinateur, pour lesquels le Deep Learning s'est avéré très efficace.

Au cours de ce travail, nous avons d'abord commencé par présenter la Réalité Augmentée, en retraçant son historique et en détaillant son fonctionnement, ainsi que le matériel auquel un système RA fait appel. Nous avons ensuite présenté plus en détail le tracking, évoqué ses types et les différentes méthodes qui y sont utilisées, et effectué un comparatif de celles-ci selon le contexte. Après ça, nous avons énoncé trois méthodes d'application de Deep Learning en Tracking, présentes dans la littérature, en détaillant pour chacune les procédés utilisés, les type et architecture des réseaux de neurones utilisés. Nous avons remarqué que dans toutes les méthodes citées, les réseaux de neurones utilisés sont les réseaux de neurones convolutifs (CNN), et ceci est dû à leur efficacité démontrée dans la reconnaissance d'images et de la vision par ordinateur. Or, cette efficacité a été avérée avec les images, tandis que dans notre cas ce sont des vidéos que nous avons à traiter (Champ de vision de l'utilisateur durant l'expérience). Il est clair que les CNN restent efficaces par la simple raison que l'on peut considérer une vidéo comme étant une succession d'images (Frames), mais dans certains cas ces images sont traitées comme si elles étaient indépendantes, c'est-à-dire que beaucoup de méthodes n'exploitent pas assez le fait que les images se suivent et qu'il y ait une relation de précédence entre elles, contrairement à ce qui fait dans l'algorithme GOTURN présenté dans 3.2.

C'est pour cette raison que nous proposons l'utilisation des réseaux de neurones récurrents (RNN), qui sont, rappelons-le, des architectures de réseaux neuronaux qui prennent à chaque instant t le résultat à l'instant t-1 comme entrée (Voir Annexe I). Cela nous permettrait d'exploiter la succession des images, c'est-à-dire que la position d'un objet à l'instant t influera inéluctablement sur sa position à l'instant t+1. Certaines recherches ont d'ores et déjà été effectuées dans ce domaine, comme (Fang, 2016) qui propose l'utilisation de RNN pour la détection de personnes mobiles. Néanmoins sa solution présente quelques lacunes lorsque les personnes sont habillées de la même couleur par exemple, ou plus généralement ont la même apparence. Une autre étude est celle de (Milan et al., 2017) qui utilise une architecture assez connue des RNN qui est Long Short Term Memory (LSTM) qui est surtout utilisée pour la reconnaissance de caractères et de paroles. Le réseau développé présente toutefois des résultats considérés satisfaisants.

Références bibliographiques

Akgul, O., Penekli, H. I., & Genc, Y. (2017). Applying Deep Learning in Augmented Reality Tracking. *Proceedings - 12th International Conference on Signal Image Technology and Internet-Based Systems, SITIS 2016*, 47–54. https://doi.org/10.1109/SITIS.2016.17

Aulenta, F., & Lens, P. (2011). Recent advances in environmental biotechnology. *New Biotechnology*, *29*(1), 1. https://doi.org/10.1016/S1871-6784(11)00246-9

Aulenta, F., & Lens, P. (2011). Recent advances in environmental biotechnology. *New Biotechnology*, *29*(1), 1. https://doi.org/10.1016/S1871-6784(11)00246-9

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). *Recent Advances in Augmented Reality Same as Report (SAR)*. (December).

Baek, A. R., Lee, K., & Choi, H. (2013). CPU and GPU parallel processing for mobile Augmented Reality. *Proceedings of the 2013 6th International Congress on Image and Signal Processing, CISP 2013*, *1*(Cisp), 133–137. https://doi.org/10.1109/CISP.2013.6743972

Balaban, S. (2015). Deep learning and face recognition: the state of the art. *Biometric and Surveillance Technology for Human and Activity Identification XII*, *9457*, 94570B. https://doi.org/10.1117/12.2181526

Behringer, R. (1999). Registration for outdoor augmented reality applications using computer vision techniques and hybrid sensors. *Proceedings - Virtual Reality Annual International Symposium*, (May), 244–251. https://doi.org/10.1109/vr.1999.756958

Billinghurst, M., Clark, A., & Lee, G. (2014). A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, *8*(2–3), 73–272. https://doi.org/10.1561/1100000049

Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, *51*(1), 341–377. https://doi.org/10.1007/s11042-010-0660-6

Chau, D. P., Bremond, F., & Thonnat, M. (2013). *Object Tracking in Videos: Approaches and Issues*. (1). Retrieved from http://arxiv.org/abs/1304.5212

Collins, R. (n.d.). *CSE486: Harris Corner Detector*.

Dandachi, G., Assoum, A., Elhassan, B., & Dornaika, F. (2015). Machine learning schemes in augmented reality for features detection. *2015 5th International Conference on Digital Information and Communication Technology and Its Applications, DICTAP 2015*, (May), 101–105. https://doi.org/10.1109/DICTAP.2015.7113179

Dieter Schmalstieg, D. W. (2008). Mobile Phones as a Platform for Augmented Reality Abstract: Handheld Augmented Reality (AR) running on. *IEEE VR 2008 Workshop on*

Software Engineering and Architectures for Realtime Interactive Systems, 43–44. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.144.6950

Diguet, J. P., Bergmann, N., & Morgère, J. C. (2015). Dedicated object processor for mobile augmented reality - Sailor assistance case study. *Eurasip Journal on Embedded Systems*, *2015*(1), 1–17. https://doi.org/10.1186/s13639-014-0019-6

Drahansky, M., Paridah, M. ., Moradbak, A., Mohamed, A. Z., abdulwahab taiwo Owolabi, F., Asniza, M., & Abdul Khalid, S. H. P. (2016). We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %. *Intech*, *i*(tourism), 13. https://doi.org/http://dx.doi.org/10.5772/57353

Ellakany, H., Fábián, K., Németh, I., & Stipkovits, L. (1998). Antibody response detected by immunoblot in respiratory tract washings of chickens after infection with Mycoplasma gallisepticum. *Avian Pathology*, *27*(6), 547–554. https://doi.org/10.1080/03079459808419382

Ellis, T. (2002). Performance metrics and methods for tracking in surveillance. *3rd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, (January 2002), 26–31. Retrieved from http://dircweb.king.ac.uk/papers/Ellis_T.J.2002_853150/Performance Metrics.pdf

Erb, R. J. (1993). Introduction to Backpropagation Neural Network Computation. *Pharmaceutical Research: An Official Journal of the American Association of Pharmaceutical Scientists*, Vol. 10, pp. 165–170. https://doi.org/10.1023/A:1018966222807

Fang, K. (2016). Track-RNN: Joint Detection and Tracking Using Recurrent Neural Networks. *29th Conference on Neural Information Processing Systems (NIPS 2016)*, (Nips). Retrieved from https://web.stanford.edu/class/cs231a/prev_projects_2016/final_report (7).pdf

Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, Jie Yu, Di Niu, & Bo Li. (2013). Gearing resource poor mobile devices with powerful clouds. *IEEE Wireless Communications*, *20*(June), 14–22. https://doi.org/10.1109/MWC.2013.6549279

Gordon, D., Farhadi, A., & Fox, D. (2018). Re 3 : Real-time recurrent regression networks for visual tracking of generic objects. *IEEE Robotics and Automation Letters*, *3*(2), 788–795. https://doi.org/10.1109/LRA.2018.2792152

Hanna, M. G., Ahmed, I., Nine, J., Prajapati, S., & Pantanowitz, L. (2018). Augmented reality technology using microsoft hololens in anatomic pathology. *Archives of Pathology and Laboratory Medicine*, *142*(5), 638–644. https://doi.org/10.5858/arpa.2017-0189-OA

Härmä, A., Jakka, J., Tikander, M., Karjalainen, M., Lokki, T., Hiipakka, J., & Lorho, G. (2004). Augmented reality audio for mobile and wearable appliances. *AES: Journal of the Audio Engineering Society*, *52*(6), 618–639.

Harris, C., & Stephens, M. (2013). *A Combined Corner and Edge Detector*. 23.1--23.6. https://doi.org/10.5244/c.2.23

Held, D., Thrun, S., & Savarese, S. (2016). GOTURN: Learning to Track at 100 FPS with Deep. *Eccv 2016*, 749–765. Retrieved from http://davheld.github.io/GOTURN/GOTURN.html

Jacob, A., & Anitha, J. (2012). Inspection of Various Object Tracking Techniques. *Ijeit.Com*, *2*(6), 118–124. Retrieved from http://ijeit.com/vol 2/Issue 6/IJEIT1412201212_21.pdf

Janin, A. L., Mizell, D. W., & Caudell, T. P. (1993). Calibration of head-mounted displays for augmented reality applications. *1993 IEEE Annual Virtual Reality International Symposium*, 246–255. https://doi.org/10.1109/vrais.1993.380772

Jansen, K., & Zhang, H. (2007). Scheduling malleable tasks. *Handbook of Approximation Algorithms and Metaheuristics*, 16–45. https://doi.org/10.1201/9781420010749

Jemilda, G. (2017). Tracking Moving Objects in Video. *Journal of Computers*, *12*(3), 221–229. https://doi.org/10.17706/jcp.12.3.221-229

Juan, C., YuLin, W., Tjondronegoro Dian, W., & Wei, S. (2018). Construction of interactive teaching system for course of mechanical drawing based on mobile augmented reality technology. *International Journal of Emerging Technologies in Learning*, *13*(2), 126–139. https://doi.org/10.3991/ijet.v13i02.7847

Khandelwal, P. (2015). Detection of Features to Track Objects and Segmentation using GrabCut for Application in Marker-less Augmented Reality. *Procedia - Procedia Computer Science*, *58*, 698–705. https://doi.org/10.1016/j.procs.2015.08.090

Kim, G., Lee, K., Kim, Y., Park, S., Hong, I., Bong, K., & Yoo, H. J. (2015). A 1.22 tops and 1.52 mW/MHz augmented reality multicore processor with neural network noc for HMD applications. *IEEE Journal of Solid-State Circuits*, *50*(1), 113–124. https://doi.org/10.1109/JSSC.2014.2352303

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Madison, D. (2018). The future of augmented reality in healthcare. *HealthManagement.Org*, *18*(1), 42–43. Retrieved from https://developer.apple.com/arkit/

Mekni, M., & Lemieux, A. (2014). Augmented Reality : Applications , Challenges and Future Trends. *Applied Computational Science Anywhere*, 205–214.

Mele, B., & Altarelli, G. (1993). Lepton spectra as a measure of b quark polarization at LEP. *Physics Letters B*, *299*(3–4), 345–350. https://doi.org/10.1016/0370-2693(93)90272-J

Milan, A., Rezatofighi, S. H., Dick, A., Reid, I., & Schindler, K. (2017). Online multi-target tracking using recurrent neural networks. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4225–4232.

Neumann, U., & Cho, Y. (n.d.). *A Self-Tracking Augmented Reality System*. 1–7.

Nilsen, T., Linton, S., & Looser, J. (2004). Motivations for augmented reality gaming. *Proceedings of FUSE 4*, (May), 86–93.

O'Shea, P. M. (2011). Augmented reality in education: Current trends. *International Journal of Gaming and Computer-Mediated Simulations*, *3*(1), 91–93. https://doi.org/10.4018/jgcms.2011010108

Pandey, M., Wadhwa, M., & Nair, M. P. (2014). *Tracking Algorithm for Augmented Reality System*. *3*(6), 6610–6614.

Pelc, A., & Raynal, M. (2005). Lecture Notes in Computer Science: Preface. In *Lecture Notes in Computer Science* (Vol. 3499).

Piccardi, M. (2004). Background subtraction techniques: A review. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, *4*, 3099–3104. https://doi.org/10.1109/ICSMC.2004.1400815

Poupyrev, I., Tan, D. S., Billinghurst, M., Kato, H., Regenbrecht, H., & Tetsutani, N. (2002). Developing a generic augmented-reality interface. *Computer*, *35*(3), 44–50. https://doi.org/10.1109/2.989929

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June-2015*, 815–823. https://doi.org/10.1109/CVPR.2015.7298682

Scovanner, P., & Tappen, M. F. (2009). Learning pedestrian dynamics from the real world. *Proceedings of the IEEE International Conference on Computer Vision*, 381–388. https://doi.org/10.1109/ICCV.2009.5459224

Sebastian, P., Voon, Y. V., & Comley, R. (2011). Performance evaluation metrics for video tracking. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, *28*(6), 493–502. https://doi.org/10.4103/0256-4602.90759

Si-Mohammed, H., Argelaguet, F., Casiez, G., Roussel, N., & Lécuyer, A. (2017). Braincomputer interfaces and augmented reality: A state of the art. *Graz Brain-Computer Interface Conference*. https://doi.org/10.3217/978-3-85125-533-1-82

Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, *9*(1). https://doi.org/10.1186/s41074-017-0027-2

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11141 LNCS*, 270–279. https://doi.org/10.1007/978-3-030-01424-7_27

Thrun S, Leonard JJ. Simultaneous localization and mapping. Springer handbook of robotics. Springer. (2008), 871–889.

Tian, Y., Guan, T., & Wang, C. (2010). Real-time occlusion handling in augmented reality based on an object tracking approach. *Sensors*, *10*(4), 2885–2900. https://doi.org/10.3390/s100402885

Uchiyama, H., & Marchand, E. (n.d.). *Object Detection and Pose Tracking for Augmented Reality : Recent Approaches*. 1–8.

Veltkamp, R. C. (2001). Shape matching: Similarity measures and algorithms. *Proceedings - International Conference on Shape Modeling and Applications, SMI 2001* , (June 2001), 188–197. https://doi.org/10.1109/SMA.2001.923389

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, *2018*. https://doi.org/10.1155/2018/7068349

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*(3), 355–368. https://doi.org/10.1109/TVCG.2009.99

Wang, J., Erkoyuncu, J., & Roy, R. (2018). A Conceptual Design for Smell Based Augmented Reality: Case Study in Maintenance Diagnosis. *Procedia CIRP*, *78*, 109–114. https://doi.org/10.1016/j.procir.2018.09.067

Wojke, N., Bewley, A., & Paulus, D. (n.d.). *SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC Nicolai Wojke † , Alex Bewley , Dietrich Paulus † University of Koblenz-Landau † , Queensland University of Technology*.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, *38*(4). https://doi.org/10.1145/1177352.1177355

Yovcheva, Z., Buhalis, D., & Gatzidis, C. (2012). Overview of smartphone augmented reality applications for tourism. *E-Review of Tourism Research*, *10*(2), 63–66.

Zhang, L., Gao, G., Zhou, C., Cui, Z., & Wang, L. (2018). An efficient feature extraction scheme for mobile anti-shake in augmented reality. *Tehnicki Vjesnik*, *25*(4), 1119–1124. https://doi.org/10.17559/TV-20180408120907

Zhou, F., Dun, H. B. L., & Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008* , 193–202. https://doi.org/10.1109/ISMAR.2008.4637362

Annexe I : Deep Learning

Le Deep Learning ou apprentissage profond est une branche de l'intelligence artificielle, dérivé du Machine Learning (apprentissage automatique), qui lui est un procédé où la machine est capable « d'apprendre » par elle-même. Par intelligence artificielle, nous entendons généralement un programme informatique capable de reproduire certaines tâches naturelles pour l'intelligence humaine, telles que la reconnaissance de modèles d'images ou de sons.

1. Définition

Selon (Tan et al., 2018), les algorithmes de Deep Learning tentent d'apprendre des caractéristiques de haut niveau à partir de données de masse. Ils peuvent extraire automatiquement les caractéristiques de données par un algorithme d'apprentissage de fonctionnalités (non supervisé ou semi-supervisé) et faire l'extraction d'entités hiérarchiques.

Selon (LeCun, 2015), le Deep Learning exhibe une structure complexe dans de grands ensembles de données en utilisant l'algorithme de rétropropagation pour indiquer comment une machine devrait changer ses paramètres internes, qui sont eux utilisés pour calculer la représentation dans chaque couche à partir de la représentation dans la couche précédente.

Si l'on veut percevoir le Deep Learning de façon intuitive, on peut considérer qu'il s'appuie sur un réseau de neurones artificiels s'inspirant directement du cerveau humain, où celui-ci est composé voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

La structure de base traitée dans le Deep Learning est le réseau de neurones artificiel, qui est en fait une structure de graphe, où chaque neurone peut être vu comme le nœud et où les connections sont des arêtes.

2. Réseaux de neurones artificiels

Le terme « profond » dans apprentissage profond revient à la structure des réseaux de neurones, qui sont la structure de base utilise dans ce type d'apprentissage. Un réseau de neurone est constitué d'un ensemble de couches successives, où plus le nombre de couches augmente et plus le réseau est « profond ». Nous détaillons cette structure dans ce qui suit.

3.4.1 Définition

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base d'informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau. Il y a trois notions essentielles à connaître pour comprendre le fonctionnement d'un réseau de neurones :

- **Neurone :** Unité de traitement qui reçoit des données en entrée, sous la forme d'un vecteur, et produit une sortie réelle.
- **Connexion :** La valeur numérique du poids associé à une connexion entre deux unités reflète la force de la relation entre ces deux unités. Si cette valeur est positive, la connexion est dite 'excitatrice', sinon elle est dite inhibitrice.
- **Couche :** Superpositions de neurones indépendants, et chaque couche est reliée à une autre à travers des connexions.
- **Fonction d'activation :** Fonction de transfert qui transforme une entrée en sortie à travers une fonction. Elle agit sur les entrées, et les poids des arêtes reliées à un neurone, en leur appliquant une fonction.

Un réseau de neurones classique, contenant toutes ces structures, est illustré dans la Figure 27.

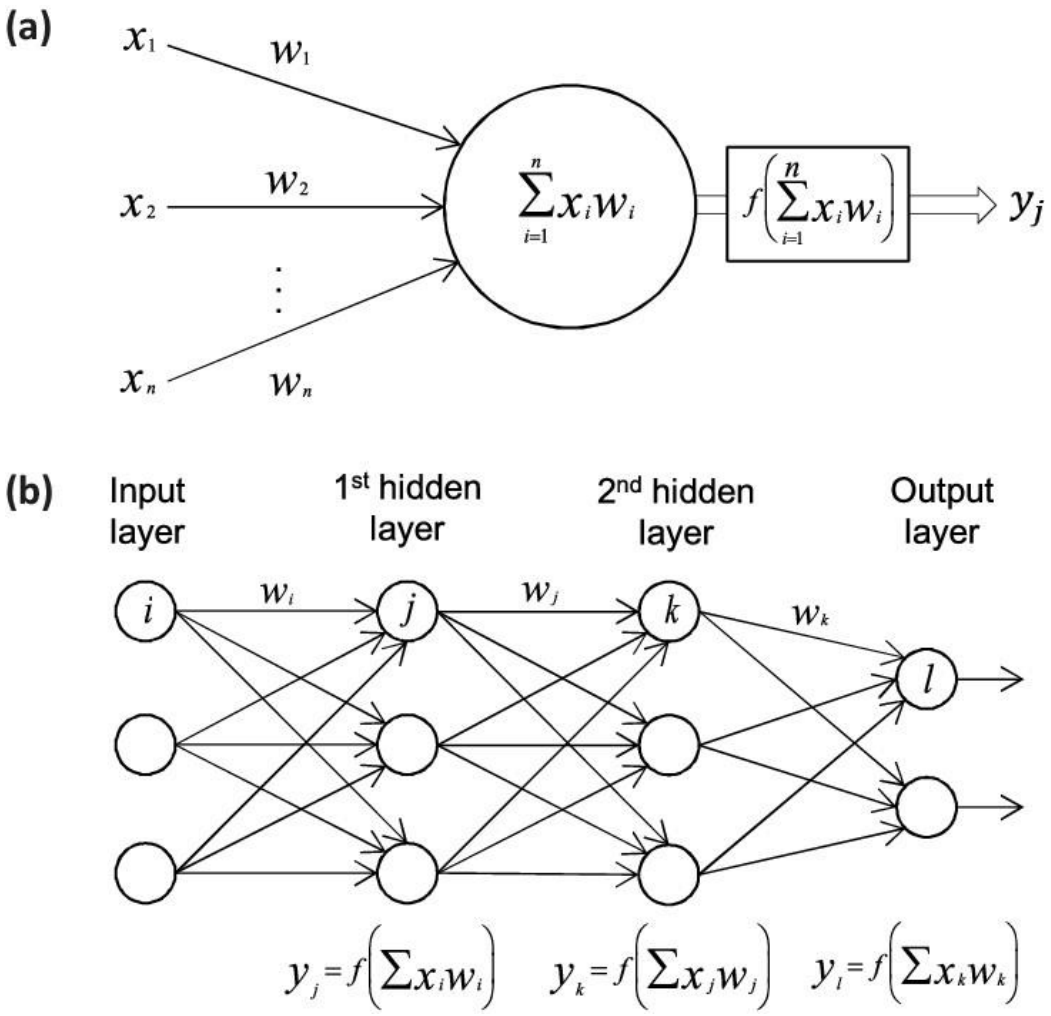

Figure 27 - Structure d'un réseau de neurones (Vieira et al., 2017)

3. Fonctionnement

Les utilisations les plus communes des réseaux de neurones sont la classification et la régression. La classification peut être définie comme étant un processus permettant au réseau de neurones de prédire la classe à laquelle appartient une certaine donnée en entrée, selon les données avec lesquelles il a été entraîné. La régression quant à elle est le fait de prédire une valeur en sortie à partir de certaines valeurs en entrée. Pour ce genre d'applications, les données d'apprentissage sont généralement labelisées, c'est-à-dire que sorties (Classe pour la classification, valeur pour la régression) sont connues. Les réseaux de neurones fonctionnent comme suit : Initialement, les poids sur les arêtes du graphe sont générés de manière aléatoire. La première couche du réseau est la couche d'entrée, elle est alimentée par les données d'entraînement du modèle, généralement sous forme de vecteurs. Puis, les données sur lesquelles on applique la fonction d'activation transitent à travers les différentes arêtes pondérées, avec les poids associés, jusqu'à arriver à la sortie finale au niveau de la dernière couche. Le phénomène d'apprentissage se produit ensuite, et consiste simplement à minimiser l'erreur, qui est la différence entre la valeur prédite au niveau de la sortie, et celle réelle correspondant à la donnée en entrée. C'est donc un problème d'optimisation qui consiste à minimiser une fonction de coût, et ce en utilisant par exemple la méthode du 'gradient descendant'. Cette fonction est donc minimisée en agissant sur les poids associés aux arêtes, ainsi certaines connexions seront plus importantes que d'autres.

Ce processus est répété un bon nombre de fois, et avec un grand nombre de données. Le fait de changer les coefficients des arêtes en faisant répercuter l'erreur à minimiser est appelé Rétropropagation (Figure 28) (Back-Propagation) (Erb, 1993).

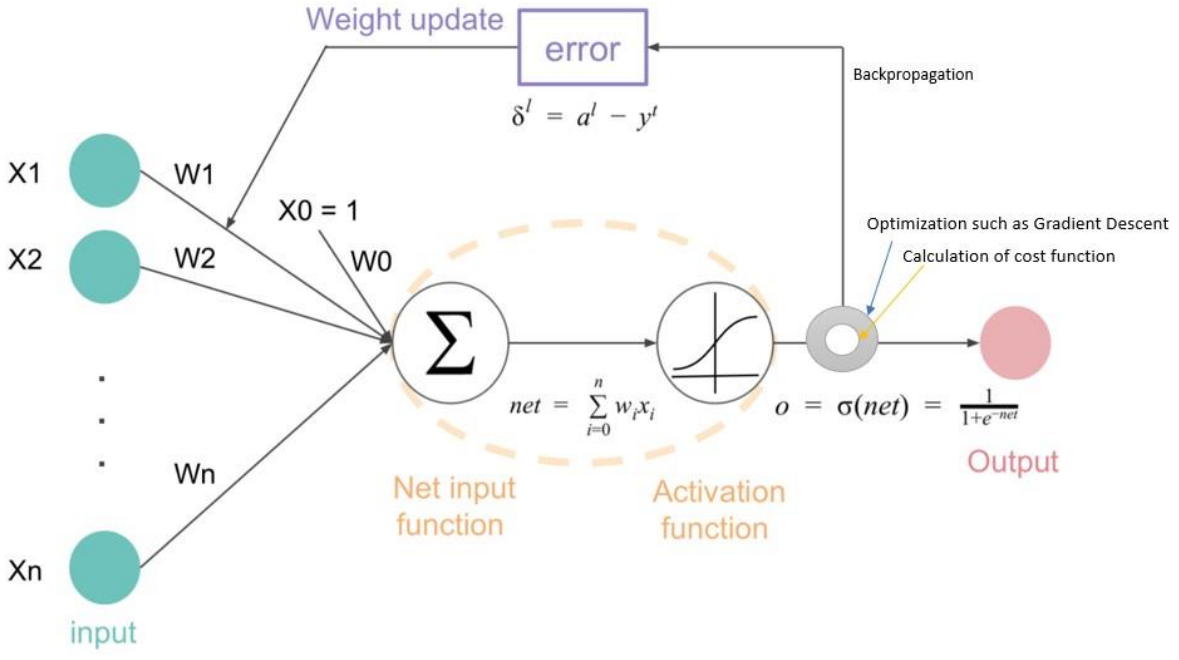

Figure 28 – Rétropropagation (StackExchange 2019)

4. Types des réseaux de neurones

Les réseaux de neurones peuvent être divisés en plusieurs types, selon leur architecture, et leurs applications de prédilection.

a. Réseau neuronal à propagation avant (Feed-Forward)

Type le plus simple et le plus général des réseaux de neurones, ils sont aussi appelés perceptrons multicouches. Ils sont composés d'une ou plusieurs couches de neurones, reliées aux entrées ou aux couches précédentes, et une couche de sortie, reliée aux neurones cachés. On les appelle feed-forward car l'information ne peut aller que des entrées aux sorties, sans revenir en arrière.

b. Réseau neuronal convolutif

C'est un type de réseau de neurones artificiels utilisé dans la reconnaissance et le traitement des images, et spécialement conçu pour l'analyse des pixels. Ce type de réseau de neurones représente une puissante application de l'intelligence artificielle (IA) au traitement des images, qui s'appuie sur l'apprentissage profond pour effectuer des tâches descriptives et génératives. Ils exploitent souvent la vision artificielle, notamment la reconnaissance de la vidéo et des images, les systèmes de recommandation et le traitement automatique du langage naturel (Natural Language Processing).

Un CNN (Convolutional Neural Network) utilise un système comparable au perceptron multicouche, mais conçu pour réduire le nombre de calculs. La structure d'un CNN consiste en une succession de couches ayant chacune un rôle précis. A l'issue d'une classification, les sorties d'un réseau CNN constituent les différents objets constituant l'image en entrée.

La logique que suivent les réseaux CNN est que les images en entrée étant de taille considérable, il faudrait les réduire en gardant une représentation fidèle des données portées par l'image, mais de taille moindre. Les couches constituant le réseau sont les suivantes (Figure 29) :

- Convolution (CONV) qui traite les données du champ récepteur.
- Pooling (POOL) qui permet de compresser l'information en réduisant la taille de l'image intermédiaire.
- Correction (ReLU).
- Entièrement Connectée (FC), qui est un réseau de neurones classique fait pour l'apprentissage.
- Perte (LOSS), qui spécifie de quelle manière quelques arêtes seront pénalisées par rapport à d'autres lors du calcul de l'erreur.

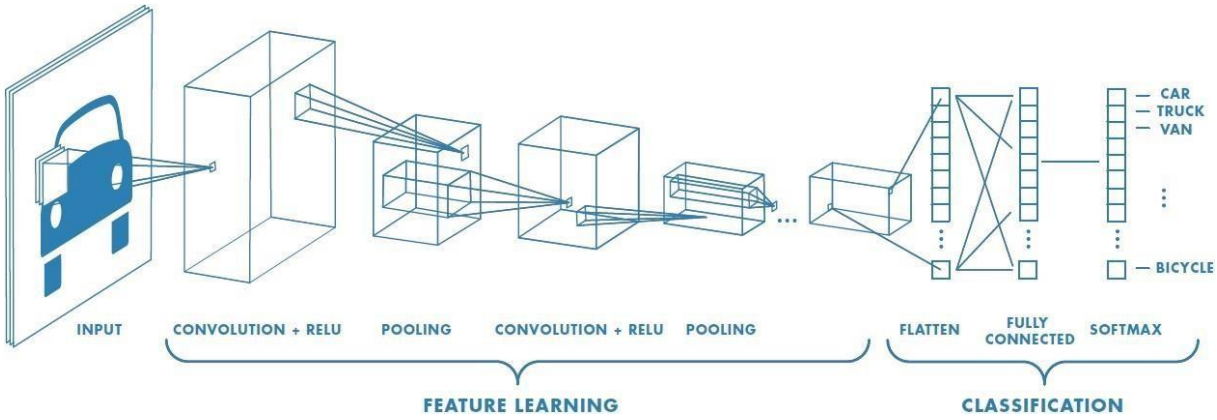

Figure 29 - Architecture Réseau CNN (Saha, 2018)

c. Réseau neuronal récurrent

Les réseaux récurrents (ou RNN pour Recurrent Neural Networks) sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches (Figure 30). Ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés. Pour cette raison, les RNNs sont particulièrement adaptés aux applications faisant intervenir le contexte. Plus particulièrement, ils sont efficaces sur le traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres. En effet, les RNNs « classiques » ne sont capables de mémoriser que le passé dit proche, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ.

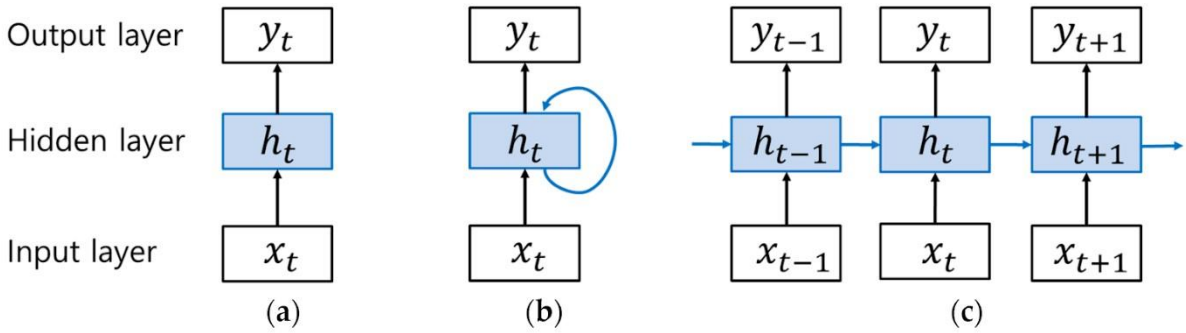

Figure 30 - Architecture Réseau RNN (Yangseon Kim et al., 2017)

Les premiers à avoir introduit ce concept sont (Mikilov et al., 2010. Ils ont défini une nouvelle architecture de réseaux de neurones pour le traitement concernant la reconnaissance de la parole (Speech Recognition).

5. Vision par Ordinateur

Il est nécessaire d'évoquer ce domaine car c'est le principe même du Tracking en RA, le fait de pouvoir reconnaître les objets dans une expérience RA et de les positionner correctement relevant de la vision par ordinateur. Nous présentons dans ce qui suit la vision par ordinateur, certains défis auxquels fait face cette branche de l'imagerie, ainsi que le lien étroit qu'il y a entre elle et le Deep Learning.

a. Définition

Le terme de « Computer Vision » ou « vision par ordinateur » désigne les différentes techniques permettant aux ordinateurs de voir et de « comprendre » le contenu d'images. Il s'agit d'une sous-catégorie d'intelligence artificielle et de Machine Learning.

Ce domaine regroupe de multiples techniques issues de divers champs d'ingénierie ou d'informatique. De manière générale, les différentes méthodes ont pour but de reproduire la vision humaine. Pour comprendre le contenu des images, les machines doivent être capables d'en extraire une caractéristique : un objet, une description, un modèle 3D…

Certains systèmes de vision par ordinateur peuvent aussi nécessiter un traitement de l'image, à savoir une simplification ou une augmentation de son contenu. En guise d'exemple, on peut citer la normalisation des propriétés photométriques de l'image, le rognage de ses contours ou la suppression du « bruit » tels que les artefacts numériques induits par un faible éclairage.

b. Défis

La Vision par Ordinateur, étant un domaine assez complexe, fait face aujourd'hui à de nombreuses problématiques, qui ralentissent le développement de ce domaine de recherche. On peut citer parmi ces problématiques :

- **Complexité de la vision humaine** : Pour égaler la vision humaine il faudrait d'abord que l'on sache comment fonctionne exactement la vision de l'humain. Or cette tâche demeure très complexe, nous devons d'abord comprendre exactement comment se fait la perception par l'œil humain, ensuite comment cette perception est traitée par le cerveau.
- **Complexité du monde visuel :** Les images que nous voulons analyser par ordinateur sont généralement très complexes, et la détection de formes y est extrêmement difficile, car un objet peut être perçu de distances différentes, sous des angles divers, et avec des luminosités différentes.
- **L'infinité de catégorie d'objets à reconnaître :** Etablir une liste exhaustive des objets à reconnaître pour une machine serait tout bonnement inconcevable, car celle-ci serait quasiment interminable.

c. Domaines d'application

Les domaines d'application de la vision par ordinateur sont nombreux. On peut citer :

- **Traitement de textes :** Certains algorithmes peuvent aujourd'hui reconnaître les caractères écrits de manière manuscrite.
- **Imagerie médicale :** Des algorithmes de détection de tumeurs à partir d'images d'échographie sont aujourd'hui utilisés dans quelques hôpitaux pour assister les médecins.
- **Détection d'objets :** Détection de la position d'un objet cible dans une image.
- **Traque d'objets :** Suivi du mouvement d'un objet cible dans une vidéo.
- **Surveillance automatique :** Détection automatique d'anomalies dans des vidéosurveillance.
- **Reconnaissance faciale :** Détection automatique d'un visage humain, que ce soit pour des raisons de sécurité (Verrouillage smartphone) ou autre…

6. Deep Learning pour la Vision par Ordinateur

Notre objectif étant d'examiner l'application du Deep Learning au tracking en RA, il est intéressant de voir comment l'on applique le Deep Learning dans la vision par ordinateur, puisque le tracking en découle directement.

a. Motivations

Durant les dernières années les algorithmes de Deep Learning ont surclassé les algorithmes de Machine Learning dans divers domaines, avec la reconnaissance d'images comme domaine majeur (Tan et al., 2018). Ceci est dû d'un côté au fait que les images sont des matrices de taille imposante, et le Deep Learning est connu pour le traitement de données massives. Par ailleurs, ceci est aussi dû au fait que les méthodes de Deep Learning sont beaucoup plus efficaces quand les données traitées sont en même temps complexes et de différentes sources.

b. Utilisations

Il existe diverses applications du Deep Learning dans le domaine très vaste qu'est la reconnaissance d'image. Nous présentons deux exemples majeurs de l'utilisation de la vision par ordinateur, ainsi que certaines solutions qui ont été proposées pour ces applications.

1- **Détection d'objets :** L'approche la plus commune afin de détecter la présence d'un objet dans une image en utilisant l'apprentissage profond est de découper l'image en un très grand nombre de fenêtres, et faire passer chacune d'elle à travers un réseau CNN. Il faut ensuite classifier les caractéristiques issues du passage par le réseau (Généralement les objets composant l'image, voir l'image qui suit) à l'aide d'un classifieur quelconque (exemple SVM). Cette méthode offre de bons résultats de précision, et répond bien à la question de la présence d'un objet ou non dans l'image, mais n'arrive pas à précisément déterminer sa position (LeCun, 2015).

2- **Reconnaissance faciale :** Application très répandue du Deep Learning. L'approche la plus simple est la même que pour la détection d'objets : Extraire les caractéristiques à partir d'une image d'un visage humain, puis de les faire passer à travers un classifieur. Google et Facebook ont récemment développé chacun leur propre réseau de neurones pour la reconnaissance faciale, comme le réseau FaceNet de (Schroff et al., 2015) de la firme Google. Le tableau 4 contient les méthodes les plus connues et les plus efficaces pour la reconnaissance faciale, testées sur la base de données LFW (Labelled Faces in the Wild) :

Tableau 4 - Méthodes Reconnaissance Faciale (Balaban, 2015)

Name	**Method**	**Images (Millions)**	**Accuracy**
Baidu[23](Announced)	CNN	-	0.9985 ± -
Google FaceNet[5]	**CNN**	**200.0**	**0.9963 ± 0.0009**
DeepID3[24]	CNN	0.29	0.9953 ± 0.0010
MFRS[25]	CNN	5.0	0.9950 ± 0.0036
DeepID2+[26]	CNN	0.29	0.9947 ± 0.0012
DeepID2[3]	CNN	0.16	0.9915 ± 0.0013
DeepID[27]	CNN	0.2	0.9745 ± 0.0026
DeepFace[6]	CNN	4.4	0.9735 ± 0.0025
FR+FCN[28]	CNN	0.087	0.9645 ± 0.0025
TL Joint Bayesian[29]	Joint Bayesian	0.099	0.9633 ± 0.0108
High-dim LBP[30]	LBP	0.099	0.9517 ± 0.0113

Annexe II : Evaluation des méthodes

Une fois développés, les algorithmes de Tracking, doivent être mesurés pour déterminer s'ils sont efficaces dans la tâche à laquelle ils ont été conçus. Pour évaluer ces algorithmes, on utilise différentes métriques qui ont été proposées par les chercheurs ces dernières années, que nous allons examiner dans les parties qui suivent. Ces métriques doivent être capables d'évaluer la précision, la consistance et le temps d'exécution des algorithmes.

D'une façon générale, les algorithmes de Tracking ont pour objectif de détecter ou à maintenir la détection d'un objet cible dans une vidéo, ou dans l'expérience RA en ce qui nous concerne. Ces métriques doivent donc mesurer l'habilité de l'algorithme à effectuer la détection. Cependant, chaque métrique ayant ses atouts et ses inconvénients, le choix de la métrique à utiliser dépend de la situation et du contexte.

La méthode standard à laquelle on compare les algorithmes de Tracking est le Ground Truth, décrit dans ce qui suit.

1. Ground Truth

C'est un terme utilisé dans le domaine de la photographie aérienne, imagerie satellite…etc. Il désigne le fait d'extraire un ensemble d'informations objectives concernant le contenu d'une région quelconque, comme la localisation, la taille, la couleur…etc. afin de comparer celles-ci avec les informations générées par des images satellite par exemple. Un exemple d'utilisation serait d'effectuer une identification manuelle de l'utilisation d'une zone d'agriculture, les types de récoltes et autres, et comparer avec les images générées par un satellite (Ellis, 2002).

Ce procédé peut être sujet à certaines erreurs qu'il faut prendre en considération. Il faut être par exemple sûr que l'identification manuelle se fait durant la même saison où l'image satellite a été prise, car l'environnement peut changer drastiquement d'une saison à une autre.

Les défis auxquels fait face cette méthode peuvent être divisés selon (Sebastian et al., 2014) en trois catégories :

1. Déterminer la position d'un objet à partir d'un point de ce dernier comme son sommet, ou son centroïde…
2. Déterminer la bordure d'un objet par une forme géométrique précise.
3. Déterminer le type de l'objet à partir d'une classification.

2. Alternatives au Ground Truth

Déterminer la position d'un objet ou sa bordure peut s'avérer extrêmement complexe si elle se fait de manière manuelle, surtout en ce qui concerne le temps de traitement, et si le nombre d'objets à identifier est très grand. C'est pourquoi on se passe quelque fois de la méthode du Ground Truth, pour des raisons d'efficacité et de performance. On peut avoir recours par exemple à :

- Implanter un récepteur GPS mobile au niveau de chaque objet cible, qui enregistre et envoie des coordonnées 4D. Ces coordonnées pourront être ainsi comparées aux résultats de l'image satellite directement.
- Générer des séquences d'images synthétiques qui tenteraient de reproduire de la même façon les paysages réels, tout en simulant un bruit d'erreur. • Générer des séquences pseudo-synthétiques en concaténant plusieurs sous-parties d'une séquence réelle dans une vidéo.

3. Valeurs à estimer

L'objectif des métriques est d'estimer le nombre de valeurs correctes prédites d'un algorithme par rapport aux valeurs correctes du Ground Truth. On peut estimer durant le Tracking dans une vidéo les différentes valeurs suivantes :

1. **Segmentation :** Consiste à déterminer la qualité de la forme finale de la cible prédite.
2. **Performance de détection :** La capacité à préciser si l'objet cible est présent dans l'image ou pas.
3. **Complétude du Tracking :** Evaluer la qualité du suivi de l'objet dans la vidéo.
4. **Classification de la cible :** Déterminer la classe à laquelle l'objet cible appartient.
5. **Détection d'activité :** Le fait de comprendre les évènements qui se produisent dans la scène.

Ainsi, nous aurons des métriques pour estimer chacune des valeurs citées ci-dessus.

4. Métriques

Les métriques peuvent être divisées en deux catégories : La première est la simple comparaison statistique entre deux populations de valeurs, qui sont dans notre cas les valeurs prédites par l'algorithme, et les valeurs du Ground Truth (qui sont dans ce cas les valeurs correctes). La deuxième est de passer à travers d'autres valeurs calculées comme les erreurs moyennes, nombre moyen de frames durant le Tracking…etc. Ces valeurs sont donc calculées pour l'algorithme ainsi que pour le Ground Truth et sont ainsi comparées.

Dans les deux cas, la comparaison donne lieu à une matrice de taille 2x2 qui est très utilisée dans le domaine de l'apprentissage automatique, que l'on appelle matrice de contingence. Nous devons d'abord définir quelques éléments avant de présenter le contenu de la matrice :

- **Nombre de Vrais positifs :** Nombre de prédictions correctes (Valeurs de l'algorithme correspondant à celles du Ground Truth).
- **Nombre de Faux positifs :** Nombre de prédictions faites dans l'algorithmes mais non présentes dans le Ground Truth.
- **Nombre de Vrais négatifs :** Nombre de prédictions rejetées (où l'algorithme indique que l'objet cible n'est pas présent) et non présentes dans le Ground Truth.
- **Nombre de Faux négatifs :** Nombre de prédictions rejetées mais pourtant présentes dans le Ground Truth.

Ces quatre valeurs consistent donc en le contenu de la matrice de contingence. Elles sont extrêmement utilisées pour évaluer un modèle de prédiction, et la table (tableau 5) est conçue comme suit :

Tableau 5 - Matrice de Contingence (Ellis, 2002)

	Ground truth	
Observations	Positive	Negative
Positive	N_{tp}	N_{fp}
Negative	N_{fn}	N_{tn}

Ntp : Nombre de Vrais Positifs

Nfp : Nombre de Faux Positifs

Nfn : Nombre de Vrais Négatifs

Ntn : Nombre de Faux Négatifs

A partir de ces valeurs, nous pouvons calculer les métriques dérivées suivantes (N étant le nombre de prédictions totales) :

Tableau 6 - Métriques de Tracking (Ellis, 2002)

Name	**Index**
Detection rate (sensitivity)	$N_{tp}/(N_{tp}+N_{fn})$
Specificity	$N_{tn}/(N_{tn}+N_{fp})$
Accuracy	$(N_{tn}+N_{tp})/N$
Positive Predictive value	$N_{tp}/(N_{tp}+N_{fp})$
False Negative rate	$N_{fn}/(N_{tp}+N_{fn})$
False Positive rate	$N_{fp}/(N_{fp}+N_{tn})$
Negative Predictive value	$N_{tn}/(N_{tn}+N_{fn})$

- **Sensitivité :** Proportion des valeurs positives correctement prédites. C'est donc la capacité du modèle à détecter un phénomène quand il se produit.
- **Spécificité :** Proportion des valeurs négatives correctement prédites. Capacité du modèle à prédire correctement l'absence d'un phénomène.
- **Précision (Accuracy) :** Proportion de bonnes prédictions effectuées (Positives ou négatives). Capacité du modèle à prédire de bons résultats, que ça soit absence ou présence d'un phénomène. On dit souvent que si cette valeur est grande alors le modèle

est efficace, cependant cette valeur est considérée déterminante dans le cas où le dataset est symétrique quant au nombre de faux positifs et faux négatifs.

- **Valeur positive prédite :** Proportion des vrais positifs sur l'ensemble des positifs. Capacité à prédire correctement des absences de phénomène.
- **Ratio de faux négatifs :** Proportion de faux négatifs sur la somme des faux positifs et vrais négatifs.
- **Ratio de faux positifs :** Proportion de faux positifs sur la somme des faux positifs et vrais négatifs.
- **Valeur négative prédite :** Proportion des vrais négatifs sur l'ensemble des négatifs. Capacité à prédire correctement des absences de phénomène.

Ainsi chacune de ces métriques sera calculée pour les méthodes de Tracking, et le choix de la meilleure se fera selon la nature du dataset, en essayant de trouver le meilleur compromis entre les métriques. La qualité de l'immersion de l'utilisateur dans l'expérience dépendra donc de la qualité de la méthode de Tracking utilisée.