Université du Québec à Chicoutimi
Département Informatique et Mathématique (DIM)

APPROCHE HYBRIDE POUR LA SEGMENTATION D'OBJETS ET LA RECONSTRUCTION 3D EN RÉALITÉ MIXTE : INTÉGRATION DE MODÈLES PROFONDS ET TRAITEMENT DE NUAGES DE POINTS RGB-D

Par Abdoul-Wahabou HAROUNA TIAMBOU

Mémoire présentée à l'Université Québec à Chicoutimi en vue de l'obtention du grade de du Maître en Sciences (M. Sc.) en informatique.

Québec, Canada

RÉSUMÉ

La capacité à reconstruire des objets en trois dimensions à partir d'images capturées dans le monde réel constitue un enjeu central pour le développement de systèmes intelligents en environnement de réalité mixte. Ce mémoire présente une approche hybride permettant de détecter, segmenter et reconstruire automatiquement des objets à partir d'images et données de profondeur.

Notre solution repose sur une application développée pour le casque HoloLens 2, qui permet de capturer des scènes sous forme de séquences d'images. Ces données sont ensuite transmises à un serveur distant où un traitement automatisé est effectué. Ce traitement comprend deux étapes principales : la segmentation des objets dans les images, puis la reconstruction de leur forme en trois dimensions.

Trois méthodes d'apprentissage automatique ont été mises en œuvre et comparées pour la segmentation des objets : un modèle fondé sur des détections rapides, un autre basé sur des régions masquées, et une méthode récente conçue pour segmenter tout type d'objet sans entraînement spécifique. Ces méthodes ont été évaluées selon leur capacité à identifier correctement les objets, même dans des situations complexes impliquant des objets de petite taille, des occultations partielles ou des variations de luminosité.

Les segments extraits ont ensuite été utilisés pour produire des représentations tridimensionnelles sous forme de maillages, en s'appuyant sur les informations de profondeur et des techniques de regroupement spatial.

Les résultats montrent que cette approche permet de produire des représentations visuelles cohérentes et exploitables, ouvrant la voie à des applications concrètes dans des domaines variés tels que la robotique, l'archivage numérique ou la réalité augmentée.

L'IA générative a été utilisée uniquement pour des suggestions ou des corrections mineures (orthographe, grammaire) ainsi que pour l'amélioration d'éléments visuels, tels que l'ajout d'un titre ou d'une légende, ou encore la traduction d'éléments présents dans une image.

Mots clés :
Réalité mixte, Segmentation d'objets, Reconstruction 3D, RGB-D, HoloLens 2, SAM, Faster R-CNN, YOLO, Occupancy Network.

ABSTRACT

The ability to recnstruct objects in three dimensions from images captured in the real world is a central challenge in the development of intelligent systems for mixed reality environments. This thesis presents a hybrid approach for automatically detecting, segmenting, and reconstructing objects from images and depth data.

Our solution is based on an application developed for the HoloLens 2 headset, which enables the capture of scenes as sequences of images. These data are then transmitted to a remote server where automated processing takes place. This processing consists of two main steps: object segmentation in the images and the subsequent reconstruction of their three-dimensional shapes.

Three machine learning methods were implemented and compared for object segmentation: a model based on fast detection, another based on masked regions, and a recent approach designed to segment any type of object without specific training. These methods were evaluated based on their ability to accurately identify objects, even in complex scenarios involving small objects, partial occlusions, or varying lighting conditions.

The extracted segments were then used to generate three-dimensional mesh representations, leveraging depth information and spatial clustering techniques.

The results demonstrate that this approach enables the production of coherent and usable visual representations, paving the way for practical applications in fields such as robotics, digital archiving, and augmented reality.

Generative AI was used solely for minor suggestions or corrections (spelling, grammar), as well as for enhancing visual elements, such as adding a title or caption, or translating content within images.

Keywords:
Mixed reality, Object segmentation, 3D Reconstruction, RGB-D, HoloLens 2, SAM, Faster R-CNN, YOLO, Occupancy Network.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ACRONYMES

2D : Deux dimensions
3D : Trois dimensions
API : Interface de programmation d'applications (Application Programming Interface)
AR : Réalité augmentée (Augmented Reality)
ASPP : Regroupement spatial pyramidal avec convolutions dilatées (Atrous Spatial Pyramid Pooling)
BBOX : Boîte englobante (Bounding Box)
COCO : Objets communs dans le contexte (Common Objects in Context)
CNN : Réseau de neurones convolutifs (Convolutional Neural Network)
CUDA : Architecture unifiée de calcul pour GPU (Compute Unified Device Architecture)
DBSCAN : Regroupement spatial basé sur la densité avec bruit (Density-Based Spatial Clustering of Applications with Noise)
FPN : Réseau pyramidal de caractéristiques (Feature Pyramid Network)
FPS : Images par seconde (Frames Per Second)
GPU : Processeur graphique (Graphics Processing Unit)
HMD : Afficheur monté sur tête (Head-Mounted Display)
IoU : Intersection sur union (Intersection over Union)
JSON : Notation d'objet JavaScript (JavaScript Object Notation)
LiDAR : Détection et télémétrie par laser (Light Detection and Ranging)
MLP : Perceptron multicouche (Multi-Layer Perceptron)
mAP : Précision moyenne (mean Average Precision)
MR : Réalité mixte (Mixed Reality)
R-CNN : Réseau convolutif basé sur les régions (Region-based Convolutional Neural Network)
RGB : Rouge – Vert – Bleu (Red – Green – Blue)
RGB-D : Rouge – Vert – Bleu – Profondeur (Red – Green – Blue – Depth)
RoI : Région d'intérêt (Region of Interest)
SAM : Modèle Segment Anything (Segment Anything Model)
SLAM : Localisation et cartographie simultanées (Simultaneous Localization and Mapping)
SLIC : Regroupement linéaire itératif simple (Simple Linear Iterative Clustering)
SSD : Disque électronique (Solid-State Drive)
UQAC : Université du Québec à Chicoutimi
VR : Réalité virtuelle (Virtual Reality)
YOLO : Vous ne regardez qu'une seule fois (You Only Look Once)

DÉDICACES

Je dédie ce mémoire à mon père, un homme d'une intégrité rare, dont la sagesse, la rigueur et la droiture m'ont profondément inspiré. Tu es pour moi un modèle de persévérance et de noblesse d'esprit. Merci, papa, pour ton soutien constant, discret mais inébranlable, et pour les valeurs précieuses que tu m'as transmises.

À ma mère, dont l'amour inconditionnel, la patience infinie et la présence rassurante ont été mon pilier à chaque étape. Tu as su illuminer mes incertitudes et me porter dans les moments les plus difficiles. Merci, maman, pour ton écoute, ton courage et ta tendresse inépuisable.

À mon frère Moctar, mon allié de toujours. Plus qu'un frère, tu es un ami fidèle, un confident précieux, et un soutien de chaque instant. Merci pour ta présence, ton humour, ton énergie et ta loyauté sans faille.

Et à moi-même, pour avoir persévéré malgré les doutes, les sacrifices et les longues heures de travail. Ce mémoire est le reflet d'un parcours personnel marqué par la résilience, la discipline et la passion. Je suis fier de ce chemin parcouru — et conscient qu'il ne fait que commencer.

REMERCIEMENTS

Au dénouement de ce projet, je désire exprimer ma profonde reconnaissance envers ma famille et mes amis, pour leur soutien moral et affectif tout au long de cette expérience. Vos encouragements, vos mots doux et votre présence ont été un véritable moteur dans les moments les plus difficiles, et j'ai été touché par votre implication et votre bienveillance.

Je souhaiterais également adresser ma profonde reconnaissance envers Monsieur Kevin Bouchard, mon directeur de recherche, pour sa confiance, sa disponibilité et ses précieux conseils tout au long de ce projet. Son encadrement rigoureux, son expertise et son regard critique ont grandement contribué à la qualité de ce mémoire. Merci pour votre accompagnement bienveillant et vos encouragements constants.

Enfin, un grand merci à Rani Baghezza pour son aide précieuse, ses retours constructifs et sa générosité tout au long de cette aventure.

CHAPITRE I

INTRODUCTION

Au cours de la dernière décennie, les technologies immersives ont connu un essor sans précédent, transformant en profondeur notre manière d'interagir avec le monde numérique. Parmi ces technologies, la réalité virtuelle (VR), la réalité augmentée (AR) et, plus récemment, la réalité mixte (MR), ont élargi les frontières de l'expérience utilisateur, en brouillant la distinction entre le monde physique et le monde numérique. Cette révolution a été rendue possible grâce aux avancées conjuguées en vision par ordinateur, en traitement du signal, en calcul embarqué et en intelligence artificielle (Billinghurst et al., 2015; Kim et al., 2018). Ces technologies ont progressivement investi une grande variété de secteurs, allant de la formation à la médecine, en passant par la visualisation scientifique, l'ingénierie industrielle ou encore le divertissement (Craig, 2013).

La réalité mixte, en particulier, se distingue des autres formes d'immersion par sa capacité à combiner, en temps réel, des éléments virtuels interactifs avec l'environnement physique de l'utilisateur. Définie initialement par Milgram et Kishino (1994) comme un point d'un continuum allant du réel au virtuel, la réalité mixte englobe toutes les expériences dans lesquelles les mondes réel et numérique interagissent dynamiquement. Cette capacité à intégrer de manière contextuelle et fluide les contenus numériques est aujourd'hui rendue possible par des dispositifs comme le HoloLens 2 de Microsoft, qui offre un affichage holographique spatialement ancré, un suivi précis du regard, des gestes et de la voix, ainsi qu'un mapping spatial en 3D (Microsoft, 2020). Ces caractéristiques en font une plateforme de choix pour des applications avancées telles que la chirurgie guidée, la télé-opération robotique ou la muséologie interactive.

1.1. CONTEXTE DE LA RECHERCHE

Cependant, malgré les avancées matérielles spectaculaires et les scénarios d'usage de plus en plus sophistiqués, un défi majeur demeure : celui de la compréhension automatique de l'environnement réel. En effet, pour permettre une interaction véritablement fluide, précise et contextuelle entre l'utilisateur et les objets virtuels, il est indispensable que le système de réalité mixte perçoive, analyse et comprenne la scène physique qui l'entoure. Cela implique plusieurs capacités fondamentales : la reconnaissance d'objets, l'estimation de leur position spatiale, la segmentation d'instances et, dans certains cas, leur reconstruction tridimensionnelle (He et al., 2017; Newcombe et al., 2011).

Cette perception intelligente constitue un socle essentiel pour le bon fonctionnement des interactions homme-machine dans les environnements mixtes. Sans elle, les objets numériques risquent de flotter sans cohérence dans l'espace, d'ignorer les surfaces physiques, ou d'interagir de manière contre-intuitive avec le monde réel. Des études récentes montrent que les systèmes capables de modéliser leur environnement en temps réel améliorent la précision des interactions tout en renforçant l'engagement et la présence de l'utilisateur (Rhee et al., 2017).

Ainsi, pour que la réalité mixte déploie tout son potentiel, elle doit s'appuyer sur une convergence efficace entre des dispositifs immersifs performants, une perception environnementale fine, et une intelligence contextuelle avancée (Milgram et Kishino., 1994; Rhee et al., 2017). Dans cette optique, l'intégration de modèles d'apprentissage profond capables d'assurer, en temps réel, la segmentation sémantique et la reconstruction 3D représente une perspective particulièrement prometteuse. Des approches récentes, comme le Segment Anything Model (Kirillov et al., 2023), illustrent les avancées notables dans ce domaine. Finalement, la qualité des expériences immersives repose autant sur la précision

de l'affichage que sur la capacité du système à comprendre et interpréter son environnement de manière cohérente et contextuelle (He et al., 2017; Newcombe et al., 2011).

1.1.1. LES DÉFIS DE LA PERCEPTION VISUELLE DANS LE MONDE RÉEL

La modélisation automatique des scènes du monde réel constitue aujourd'hui l'un des défis centraux de la vision par ordinateur, en particulier dans des environnements non contraints, dynamiques et riches en incertitudes. L'identification, la segmentation et la reconstruction des objets dans un espace tridimensionnel sont des tâches complexes, exacerbées par des conditions d'acquisition souvent défavorables : objets de petite taille, occultations partielles, faible texture, éclairage variable, bruit sensoriel ou encore mouvement rapide de la caméra (He et al., 2017; Everingham et al., 2015). Ces limitations se manifestent notamment dans les contextes réels tels que la robotique mobile, la réalité mixte ou la conduite autonome, où la perception doit être robuste, en temps réel et capable de s'adapter à des scènes imprévisibles (Geiger et al., 2012).

Dans les environnements encombrés ou désordonnés, la segmentation d'objets devient particulièrement délicate. Les méthodes traditionnelles, souvent fondées uniquement sur des images RGB, se révèlent insuffisantes pour fournir une compréhension complète de la scène. En effet, bien qu'elles permettent l'extraction de caractéristiques visuelles telles que la couleur, la texture ou les contours, elles n'intègrent aucune information de profondeur ou de structure géométrique. Cette limitation est critique pour les systèmes qui requièrent une perception tridimensionnelle, comme la manipulation robotique, la navigation autonome, ou l'ancrage spatial d'objets virtuels dans un environnement réel (Szeliski, 2010; Lowe, 2004; Redmon et al., 2016).

Plusieurs études ont mis en évidence les faiblesses de l'approche purement 2D dans des tâches telles que la reconnaissance d'objets partiellement visibles ou positionnés à

différentes distances du capteur (Song et al., 2015). Par exemple, dans des environnements faiblement éclairés ou où les objets sont similaires en apparence (couleur, texture), les modèles basés uniquement sur l'image optique échouent souvent à distinguer les entités présentes dans la scène.

C'est dans ce contexte que l'intégration de données de **profondeur** (issues de capteurs RGB-D, LIDAR, stéréo ou time-of-flight) s'est imposée comme une évolution incontournable des pipelines de perception visuelle. Ces données enrichissent les représentations en offrant une information géométrique précieuse, permettant d'estimer les distances relatives, les volumes, et la disposition spatiale des objets. Elles facilitent également la désambiguïsation de scènes complexes en apportant un signal supplémentaire pour la segmentation d'objets, notamment lorsque les indices visuels ne suffisent pas (Gupta et al., 2014).

La capacité à représenter un objet non seulement sur le plan visuel, mais aussi sur le plan géométrique, devient dès lors essentielle dans les systèmes nécessitant une interaction physique ou spatiale avec le monde réel. C'est le cas notamment dans la réalité augmentée/mixte, où l'alignement spatial entre objets virtuels et surfaces réelles repose sur une compréhension fine de la géométrie environnante (Newcombe et al., 2011; Zollhöfer et al., 2018). De même, en robotique, la planification de trajectoires et la préhension d'objets nécessitent des reconstructions précises des volumes dans l'espace (Wang et al., 2019).

Par conséquent, les modèles récents combinent de plus en plus les signaux RGB et de profondeur, dans une approche multimodale, souvent pilotée par des réseaux neuronaux convolutifs 3D, des graphes sur nuages de points, ou des architectures de type transformer adaptées aux représentations spatiales (Thomas et al., 2019; Zhao et al., 2021). Ces avancées permettent d'atteindre une perception plus robuste, capable de modéliser des scènes en trois

dimensions tout en tenant compte des détails fins nécessaires aux interactions immersives et intelligentes.

1.1.2. CAPTEURS RGB-D ET VISION PAR ORDINATEUR : UNE CONVERGENCE PROMETTEUSE

Les caméras RGB-D, en combinant l'imagerie couleur avec des mesures de profondeur par pixel, ont ouvert de nouvelles perspectives en vision par ordinateur. Elles fournissent une représentation enrichie de la scène, dans laquelle chaque point de l'image est associé à une distance par rapport à la caméra. Cette capacité permet non seulement d'améliorer la segmentation d'objets, mais aussi de générer des nuages de points représentant la forme géométrique des objets dans l'espace (Zollhöfer et al., 2018).

En parallèle, l'essor de l'apprentissage profond a permis des avancées majeures dans la reconnaissance et la segmentation d'objets. Des modèles comme Mask R-CNN ou Segment Anything (He et al., 2017; Kirillov et al., 2023) se sont imposés comme des références pour extraire des masques précis, même dans des scènes complexes.

L'association entre ces deux technologies — capteurs RGB-D et modèles profonds — ouvre la voie à une perception 3D automatisée et riche en détails. Elle permet de concevoir des pipelines capables d'effectuer une segmentation fine et de projeter les résultats dans l'espace pour reconstruire les objets. Une telle convergence s'avère particulièrement adaptée aux contraintes de la réalité mixte, qui exige des traitements rapides et une grande fiabilité spatiale.

1.1.3. UNE APPROCHE HYBRIDE POUR LA RECONSTRUCTION 3D D'OBJETS

L'un des objectifs centraux de ce mémoire est de concevoir une méthode capable de reconstruire automatiquement des objets en trois dimensions à partir d'images capturées

dans le monde réel. Pour cela, nous avons mis en place une approche dite hybride, qui combine des techniques d'intelligence artificielle et des méthodes géométriques pour obtenir des représentations 3D précises et réalistes.

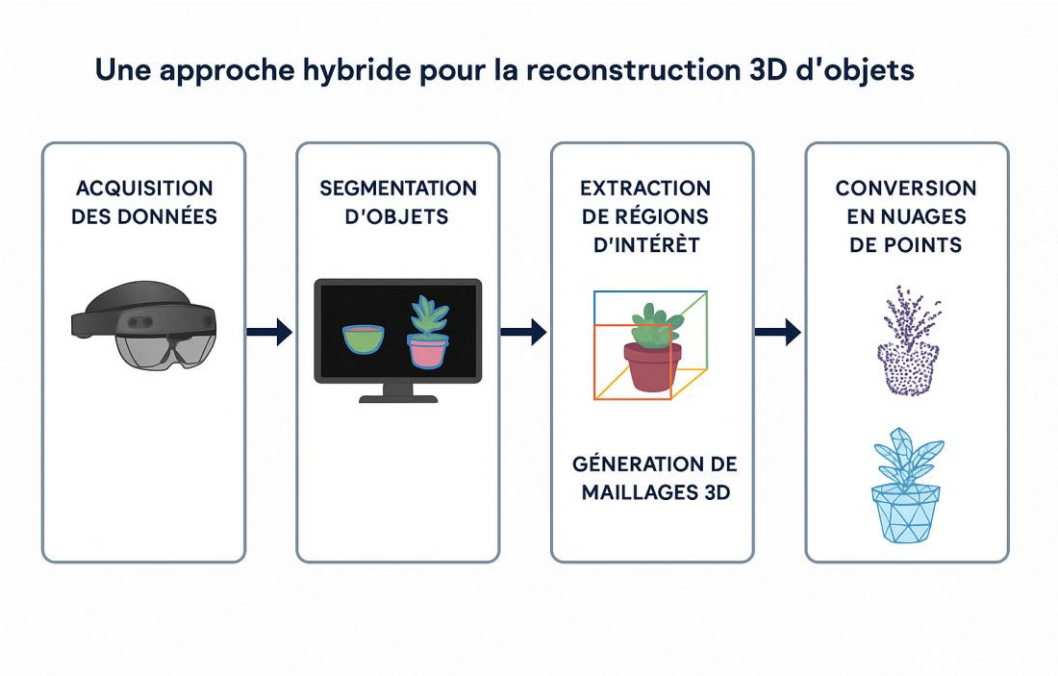

Figure 1.1 — Schéma général du pipeline hybride de segmentation et de reconstruction 3D à partir d'images RGB-D.
© Abdoul-Wahabou H. Tiambou

Le processus commence par la capture d'images à l'aide du casque HoloLens 2, qui fournit à la fois des images RGB et des informations sur la profondeur de chaque pixel. Une application développée sur mesure permet de prendre des photos et de les envoyer automatiquement vers un serveur, où elles sont stockées et prêtes à être analysées.

Dans un premier temps, les objets présents dans les images sont identifiés à l'aide de trois modèles de segmentation : YOLO (Jocher et al., 2023), Mask R-CNN (He et al., 2017), et un modèle hybride combinant Faster R-CNN (Ren et al., 2015) avec le Segment Anything Model (SAM) (Kirillov et al., 2023). Ces approches ont été sélectionnées pour leurs performances complémentaires en détection, précision des masques, et capacité de

généralisation. Leur entraînement a été réalisé sur un jeu de données personnalisé capturé avec HoloLens 2 et annoté manuellement via l'outil LabelMe (Russell et al., 2008).

Une fois les objets détectés et séparés, la seconde étape consiste à reconstruire leur forme en 3D. Pour cela, nous utilisons les données de profondeur afin de créer un nuage de points, c'est-à-dire un ensemble de points représentant la surface des objets dans l'espace. Ces points sont ensuite utilisés pour générer un maillage, c'est-à-dire une surface composée de petits triangles qui recréent la forme des objets.

Contrairement aux méthodes classiques, nous avons choisi une approche plus flexible qui utilise un réseau de neurones pour apprendre la forme globale des objets. Cette méthode permet d'obtenir des maillages plus lisses et plus cohérents, même lorsque les données sont incomplètes ou bruitées.

Cette approche hybride permet ainsi de combiner le meilleur des deux mondes : la capacité des modèles d'apprentissage à bien détecter les objets, et la puissance des méthodes géométriques pour les représenter en 3D. Elle ouvre la voie à des usages concrets dans des domaines variés comme la réalité augmentée, la robotique ou encore l'archivage numérique.

1.2. PROBLÉMATIQUE ET OBJECTIFS DE RECHERCHE

La compréhension automatique de scènes tridimensionnelles est devenue un enjeu fondamental dans la mise en œuvre de systèmes immersifs intelligents. Dans le domaine de la réalité mixte, la qualité des interactions entre les objets virtuels et le monde physique dépend fortement de la capacité du système à percevoir, interpréter et reconstruire en 3D les scènes réelles en temps réel (Billinghurst et al., 2015). Bien que des dispositifs comme le HoloLens 2 permettent aujourd'hui de superposer des hologrammes dans l'espace réel, ces

derniers reposent sur des mécanismes de perception encore limités, notamment en ce qui concerne la segmentation sémantique et la modélisation géométrique des objets capturés.

Un décalage persiste entre les capacités avancées des modèles de segmentation 2D, tels que Mask R-CNN ou YOLO, et les besoins réels en reconstruction 3D dans des contextes non structurés et dynamiques. Ces modèles sont conçus majoritairement pour traiter des images RGB et n'intègrent pas nativement les données de profondeur. De plus, la plupart des solutions actuelles en reconstruction 3D opèrent de façon découplée de la segmentation, utilisant des méthodes géométriques pures ou des techniques de SLAM (Simultaneous Localization and Mapping), qui n'offrent pas une détection d'objets adaptée à des scènes complexes (Salas-Moreno et al., 2013).

Dans ce contexte, il devient légitime de poser la question suivante : **comment concevoir un pipeline cohérent, modulaire et automatisé permettant de passer de données RGB-D capturées à une reconstruction 3D fiable et exploitable, en s'appuyant sur la segmentation d'objets par apprentissage profond ?**

Ce travail de recherche vise à répondre à cette interrogation par la mise en œuvre d'une approche hybride, reposant à la fois sur les avancées de la vision par ordinateur et sur les techniques de reconstruction 3D. Les objectifs spécifiques sont les suivants :

1. **Développer une application de capture de données RGB-D** pour le casque HoloLens 2, permettant une acquisition synchronisée d'images couleur et de profondeur en contexte réel.
2. **Concevoir un pipeline de traitement automatisé**, depuis la segmentation d'objets jusqu'à la reconstruction 3D, intégrant des modèles d'apprentissage profonds tels que YOLO, Mask R-CNN et SAM (Segment Anything Model).

3. **Évaluer les performances comparées des modèles de segmentation**, en utilisant des métriques telles que la mAP, la précision et le rappel, sur un ensemble personnalisé annoté du jeu de données.
4. **Transformer les résultats de segmentation en nuages de points 3D enrichis**, et proposer un mécanisme de clustering 4D (profondeur + couleur) par DBSCAN, suivi d'une triangulation via la méthode de Delaunay (Delaunay, 1934) avec filtrage alpha.
5. **Valider la robustesse et la pertinence du pipeline sur des scènes réelles**, en analysant les qualités géométriques des maillages produits, les limites du système et les pistes d'amélioration future.

L'ambition de ce projet est de proposer une architecture de traitement intégrée, reproductible et adaptable à différents contextes applicatifs, allant de la réalité mixte à la robotique, en passant par la numérisation patrimoniale ou la simulation interactive.

1.3. CONTRIBUTIONS DE LA RECHERCHE

Ce travail de recherche s'inscrit dans la volonté de rapprocher les technologies de vision par ordinateur, les capteurs RGB-D et les méthodes modernes de reconstruction 3D, en vue d'une intégration cohérente au sein d'environnements de réalité mixte. À travers une approche hybride mêlant apprentissage profond et modélisation implicite, ce mémoire présente quatre contributions principales, visant à renforcer l'automatisation, la robustesse et la pertinence des processus de capture, de segmentation et de reconstruction d'objets.

La première contribution concerne le développement d'une application embarquée pour le casque Microsoft HoloLens 2. Cette application, conçue sur mesure, permet la capture synchronisée d'images RGB et de profondeur, tout en automatisant la transmission des données vers un serveur distant. Elle initie également le déclenchement d'un processus de traitement backend dès la fin d'une session de capture. Cette solution permet de générer

un jeu de données structuré, encodé dans un format temporel, facilitant ainsi le traitement ultérieur et la reproductibilité des expériences. L'ensemble de l'architecture client-serveur a été optimisé pour fonctionner de manière fluide dans des conditions réelles d'acquisition mobile (Zhang et al., 2020).

La deuxième contribution est la mise en œuvre d'un pipeline hybride de traitement, combinant les résultats de segmentation issus de réseaux de neurones convolutifs avec des techniques de reconstruction géométrique. Le pipeline est constitué de plusieurs étapes successives : application de modèles de segmentation (YOLO, SAM combiné à Faster RCNN, Mask R-CNN) et génération d'un maillage 3D. Ce type d'intégration répond à la nécessité croissante de traitements automatisés et modulaires dans les systèmes de perception avancés (Edelsbrunner & Mücke, 1994).

La troisième contribution réside dans l'évaluation comparative rigoureuse des modèles de segmentation susmentionnés, entraînés sur le jeu de données personnalisé décrit précédemment. Cette analyse repose sur des métriques standard et met en lumière les points forts et les limites de chaque architecture dans des scénarios variés, incluant des objets de petite taille, des scènes partiellement occultées et des conditions de capture réalistes. En confrontant les résultats des modèles YOLO, SAM couplé à Faster RCNN et Mask R-CNN sur les mêmes séquences d'images, il a été possible de dégager des tendances significatives concernant leur efficacité dans des environnements mixtes, avec des objets de tailles variées, partiellement occultés ou faiblement texturés (Lin et al., 2014).

Enfin, la quatrième contribution consiste à valider l'ensemble du pipeline sur des cas concrets de reconstruction d'objets capturés avec le HoloLens 2. Les résultats montrent que la méthode proposée permet d'obtenir des maillages 3D plausibles et exploitables, avec un bon compromis entre fidélité géométrique et légèreté de calcul.

Ces contributions s'inscrivent dans une perspective plus large visant à combler le fossé entre les systèmes de perception automatisée, l'analyse intelligente des scènes et la reconstruction numérique réaliste. Le cadre proposé peut être adapté à d'autres cas d'usage comme la manipulation robotique, l'archivage numérique ou la conception de jumeaux numériques intelligents dans des espaces augmentés.

1.4. ORGANISATION DU DOCUMENT

Ce mémoire s'articule en cinq parties conçues pour guider le lecteur de façon progressive, depuis les fondements théoriques jusqu'à la validation expérimentale du pipeline de reconstruction 3D.

La première partie plante le décor en exposant le contexte scientifique et technologique : elle présente les enjeux de la réalité mixte, les défis de la perception visuelle dans le monde réel et la motivation qui sous-tend le développement de systèmes intelligents capables d'interagir avec des scènes réelles.

La deuxième partie offre un état de l'art détaillé, abordant successivement les concepts et technologies immersives, la segmentation d'objets par apprentissage profond (sémantique, instances, universelle, multimodale et RGB-D), les capteurs RGB-D et les méthodes de reconstruction et de génération de maillages 3D.

La troisième partie décrit la démarche méthodologique : elle détaille le cadre expérimental (LIARA et environnement matériel), la conception de l'application HoloLens 2, l'organisation des traitements automatisés sur le serveur backend, la constitution et le prétraitement du jeu de données, le choix des algorithmes de segmentation ainsi que le protocole d'entraînement et de fine-tuning.

La quatrième partie présente les expérimentations menées et analyse les résultats obtenus. On y décrit les modèles de segmentation entraînés, les métriques d'évaluation, le processus de génération des maillages 3D, puis on compare les performances, on évalue la qualité des reconstructions et on discute des limites et des pistes d'amélioration.

La cinquième partie synthétise les apports majeurs de la recherche, propose des orientations pour de futurs travaux et ouvre sur les applications potentielles du système conçu en contexte réel.

CHAPITRE II

REVUE DE LA LITTÉRATURE

La modélisation 3D d'objets à partir d'images RGB-D est un champ de recherche actif, situé à l'intersection de la vision par ordinateur, de l'apprentissage profond et des technologies immersives. L'un des objectifs de cette partie est de faire un tour d'horizon des concepts, méthodes et algorithmes qui ont marqué ces domaines, en mettant l'accent sur les approches que nous jugeons particulièrement pertinentes pour notre problématique de recherche : la segmentation d'objets et leur reconstruction 3D dans un contexte de réalité mixte.

Nous débuterons cette revue par une présentation des concepts fondamentaux liés aux réalités étendues. La Section **2.1** introduira les notions de réalité virtuelle (VR), réalité augmentée (AR) et réalité mixte (MR), en retraçant leur évolution et leurs usages actuels. Nous y préciserons également la place qu'occupe le casque HoloLens 2 dans cet écosystème, en tant que dispositif de capture de données et de rendu immersif.

La Section **2.2** abordera ensuite la segmentation d'objets, une tâche essentielle dans la compréhension automatique de scènes visuelles. Nous présenterons d'abord les approches traditionnelles issues de la vision par ordinateur, avant de nous concentrer sur les méthodes d'apprentissage profond qui ont récemment dominé le domaine, telles que YOLO, Mask R-CNN ou encore SAM (Segment Anything Model).

Enfin, la Section **2.3** sera consacrée à la reconstruction 3D et à la génération de maillages. Nous y décrirons les techniques fondées sur les images de profondeur, les méthodes de triangulation géométrique, ainsi que les approches basées sur le clustering spatial. Une attention particulière sera portée à la transition vers des méthodes de

reconstruction implicite, qui permettent de produire des surfaces 3D continues à partir de réseaux neuronaux formant une représentation dense.

Le chapitre se conclura par une analyse critique de la littérature, mettant en lumière les limites des approches existantes et les besoins auxquels notre recherche tente de répondre.

2.1. LES RÉALITÉS ÉTENDUES : CONCEPTS ET TECHNOLOGIES

Le concept de **réalités étendues** *(Extended Reality, ou XR)* regroupe un ensemble de technologies immersives visant à altérer ou enrichir la perception du monde réel par l'intégration de contenus numériques. Cette terminologie englobe trois formes principales : la réalité virtuelle (VR), la réalité augmentée (AR) et la réalité mixte (MR). Ces trois approches sont classiquement situées sur un continuum de virtualité, proposé par Milgram et Kishino (1994), allant de l'environnement entièrement réel à un environnement entièrement virtuel.

« Any particular configuration of real and virtual objects, presented to the user in a way such that they are perceived as coexisting in the same spatial context, may be described as a Mixed Reality (MR) display. » (Milgram & Kishino, 1994)

(Traduction libre : « Toute configuration particulière d'objets réels et virtuels, présentée à l'utilisateur de manière à ce qu'ils soient perçus comme coexistants dans le même contexte spatial, peut être qualifiée d'affichage en réalité mixte. »)

- La réalité virtuelle (VR) est définie comme une immersion totale de l'utilisateur dans un monde numérique généré artificiellement, le coupant de toute perception sensorielle directe de son environnement physique (Craig, 2013). Elle permet à l'utilisateur de naviguer ou d'interagir dans un espace 3D simulé, souvent à l'aide de casques immersifs.

- La réalité augmentée (AR) consiste à superposer des éléments virtuels — tels que du texte, des images ou des objets — à la vue du monde réel. Selon Azuma (1997), une AR est un système qui combine des éléments réels et virtuels en temps réel et de manière interactive, et qui est aligné spatialement avec l'environnement réel.

« AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. » (Azuma, 1997)

(Traduction libre : « La réalité augmentée permet à l'utilisateur de voir le monde réel, avec des objets virtuels superposés ou combinés à celui-ci. »)

- La réalité mixte (MR), quant à elle, représente une fusion dynamique entre les éléments réels et virtuels. Contrairement à l'AR, les objets numériques ne sont pas simplement superposés, mais interagissent en temps réel avec l'environnement physique, grâce à une compréhension de la géométrie, des surfaces et du contexte. Elle s'appuie sur des systèmes de perception avancés (vision par ordinateur, capteurs, spatial mapping) pour permettre une intégration fluide.

Cette typologie est aujourd'hui à la base des recherches en interaction homme-machine, simulation immersive et systèmes intelligents. Elle permet d'identifier les spécificités technologiques et les niveaux d'intégration nécessaires pour développer des expériences numériques cohérentes dans des contextes hybrides.

2.1.1. DISTINCTIONS ENTRE LES TECHNOLOGIES XR

Bien que les notions de réalité virtuelle (VR), de réalité augmentée (AR) et de réalité mixte (MR) soient souvent utilisées ensemble sous l'étiquette de réalités étendues (XR), elles correspondent à des paradigmes distincts en termes d'interaction, de perception et d'exigences technologiques.

Figure 2.1 — Distinctions visuelles entre les technologies XR

La **réalité virtuelle (VR)** immerge complètement l'utilisateur dans un environnement généré par ordinateur, en supprimant toute perception sensorielle du monde réel. Elle repose sur l'utilisation de casques immersifs qui isolent l'utilisateur de son environnement, et permet une interaction exclusive avec des éléments synthétiques. L'objectif est de simuler une immersion totale dans un univers artificiel (Flavián et al., 2019).

La **réalité augmentée (AR)** conserve quant à elle la perception du monde réel tout en y superposant des éléments numériques, généralement via un écran de smartphone, de tablette ou de lunettes connectées. Ces éléments peuvent enrichir l'environnement sans être forcément interactifs ou géométriquement ancrés. Bien qu'elle soit plus accessible, l'AR présente des limites en termes de précision spatiale et d'intégration (Flavián et al., 2019).

La **réalité mixte (MR)**, conceptuellement plus avancée, combine les avantages de la VR et de l'AR en permettant une interaction bidirectionnelle et cohérente entre objets virtuels et environnement physique. Les objets virtuels sont ancrés dans l'espace réel,

réagissent à la géométrie de la scène et peuvent être masqués ou manipulés. Cette intégration repose sur des technologies de reconstruction 3D, de vision par ordinateur et de tracking spatial en temps réel (Speicher et al., 2019).

L'ensemble de ces technologies immersives est aujourd'hui regroupé sous le terme de **réalité étendue (XR)**, une méta-catégorie qui désigne tout système permettant une interaction fluide entre monde réel et contenu numérique, incluant la VR, l'AR et la MR (Rauschnabel et al., 2022).

Selon Milgram et Kishino (1994), la MR occupe une position médiane sur le continuum de virtualité, où les objets réels et virtuels coexistent de façon dynamique et interactive. Contrairement à l'AR qui est principalement descriptive, la MR est contextuelle, adaptative et interactive, offrant un potentiel plus important pour des applications avancées comme la chirurgie assistée, la formation immersive ou la simulation industrielle.

2.1.2. HOLOLENS 2 : CARACTÉRISTIQUES ET USAGES

Le HoloLens 2, présenté par Microsoft en 2019, est un casque de réalité mixte autonome conçu pour permettre une interaction naturelle entre l'utilisateur et des objets numériques ancrés dans le monde réel. Successeur du premier modèle lancé en 2016, il combine ergonomie améliorée, capacités de perception environnementale avancées et puissance de calcul embarquée. Son architecture s'articule autour de plusieurs capteurs multimodaux et d'une interface de programmation orientée vers la spatialisation et l'interaction contextuelle (Microsoft, 2020).

2.1.2.1. COMPOSANTS MATÉRIELS

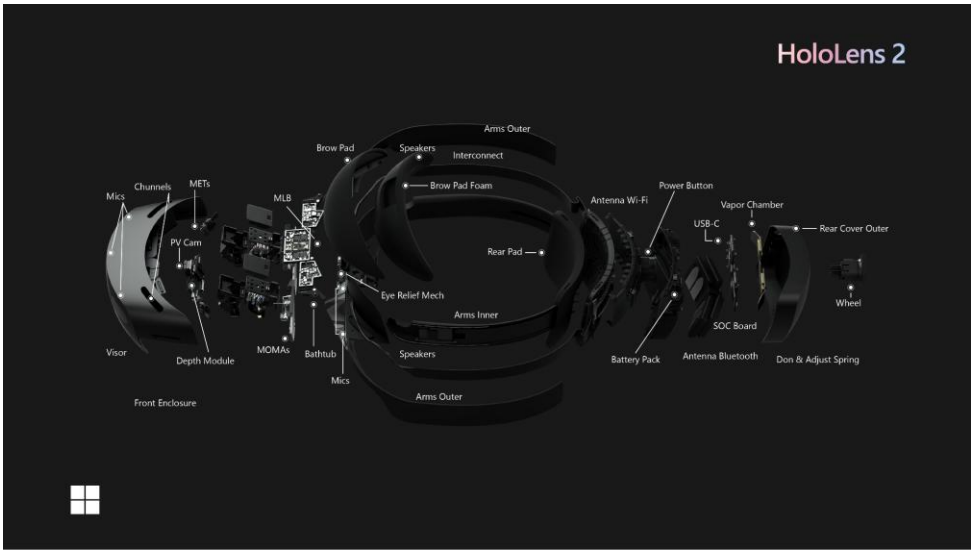

Figure 2.2 — Vue éclatée des composants de la hololens 2
© Microsoft

*Le **HoloLens 2** est un dispositif de réalité mixte autonome qui intègre une architecture matérielle avancée, conçue pour permettre une perception spatiale précise, une interaction naturelle et un traitement local des données. L'image ci-dessus présente une vue éclatée des différents composants internes du casque.*

Parmi les éléments clés, on retrouve :

- ***Les capteurs de perception visuelle**, notamment la **PV Cam** (caméra principale), le **Depth Module** pour la capture de la profondeur, ainsi que plusieurs **MICs** et **METs** pour le suivi des mouvements de la tête et la capture de la voix.*

- ***L'unité de traitement centrale**, composée du **SOC Board** (System-on-Chip), du système de refroidissement (**Vapor Chamber**), et des **antennes Wi-Fi/Bluetooth**, assurent la connectivité sans fil et le traitement embarqué des données.*

- ***Les éléments de confort et d'ajustement**, tels que le **Brow Pad**, le **Rear Pad**, ou encore le **Don & Adjust Spring**, permettent une adaptation ergonomique à la morphologie de l'utilisateur.*

- ***Les composants audio et d'affichage, comme les haut-parleurs spatiaux intégrés, le Visor transparent pour l'****affichage des hologrammes, et les* ***MOMAs*** *(Micro-Opto-Mechanical Assemblies), qui assurent la projection optique des éléments numériques dans le champ de vision.*

Cette combinaison de modules permet au HoloLens 2 d'interagir dynamiquement avec l'environnement réel tout en assurant le rendu visuel d'éléments 3D ancrés dans l'espace. C'est précisément cette configuration qui en fait un outil adapté aux applications de réalité mixte nécessitant à la fois mobilité, autonomie et précision spatiale.

2.1.2.2. CAS D'USAGE

Le HoloLens 2 a progressivement trouvé sa place dans plusieurs secteurs stratégiques, en raison de ses capacités avancées en matière de spatialisation, d'interaction gestuelle et de perception contextuelle. Grâce à son autonomie, sa précision et sa polyvalence, il est devenu un outil incontournable dans des domaines aussi variés que l'industrie, l'architecture, la formation professionnelle ou encore la médecine. Les usages qui en découlent témoignent d'un potentiel significatif pour améliorer l'efficacité opérationnelle, enrichir l'expérience utilisateur et faciliter l'accès à l'information en contexte réel.

Dans les environnements industriels, le HoloLens 2 accompagne désormais les opérateurs tout au long du cycle de vie des équipements, de la mise en service à la maintenance prédictive. Grâce à l'intégration de capteurs IoT, les techniciens visualisent en surimpression l'état des machines, anticipent les risques de panne et réduisent les temps d'arrêt non planifiés. Parallèlement, des applications spécifiques guident les nouvelles recrues pas à pas, réduisant significativement la durée d'apprentissage et les erreurs humaines (Microsoft, 2020). La collaboration à distance, appuyée par un streaming en direct

du champ de vision, offre un support expert instantané pour les interventions sur des sites difficiles d'accès, optimisant à la fois la réactivité et la qualité d'exécution (Zhou et al., 2020).

Dans le domaine de l'architecture et de l'urbanisme, le HoloLens 2 révolutionne la phase de conception et de validation des projets. Les architectes peuvent projeter des maquettes numériques à l'échelle 1:1 directement sur site, ajuster en temps réel volumes, matériaux et éclairages, et simuler l'impact environnemental — ensoleillement, flux de circulation ou acoustique — avant même le début des travaux. Cette immersion facilite la concertation entre équipes de conception, clients et autorités réglementaires, en transformant la présentation de plans en une expérience tangible et interactive (Craig, 2013).

La formation immersive à l'échelle industrielle et académique constitue un autre champ d'application majeur. Dans les secteurs de l'aéronautique, de l'automobile ou des procédés chimiques, les stagiaires s'exercent sur des scénarios réalistes sans risquer l'endommagement de matériel coûteux ni compromettre la sécurité. Les simulations holographiques permettent de répéter à volonté des procédures critiques, consolidant la mémoire procédurale et renforçant la confiance des apprenants, tout en offrant un retour immédiat sur la qualité des gestes effectués (Craig, 2013).

Enfin en milieu hospitalier, le HoloLens 2 s'est avérer être un outil innovant pour optimiser les parcours de soins avancés. Lors de la planification préopératoire, il fusionne les volumes issus de scanners et d'IRM avec l'anatomie du patient, améliorant la précision des diagnostics et la préparation des chirurgiens. En salle d'opération, la projection de guides visuels et de repères anatomiques en temps réel accompagne les gestes invasifs avec une marge d'erreur réduite. De plus, le système facilite le télé-mentorat, où un spécialiste à distance peut guider une intervention complexe, contribuant ainsi à la formation continue et à la diffusion des savoir-faire (Nicolau et al., 2022 ; Zhou et al., 2020).

2.2. LA SEGMENTATION D'OBJETS EN VISION PAR ORDINATEUR

La segmentation d'objets constitue une branche essentielle en vision par ordinateur, en particulier dans les systèmes interactifs comme la réalité mixte, où il est crucial de distinguer précisément les objets de leur environnement. Elle consiste à délimiter les contours d'un ou plusieurs objets dans une image, généralement à un niveau de précision pixel par pixel, permettant ainsi une compréhension fine de la scène analysée.

Dans cette section, nous présentons d'abord les principales définitions liées à la segmentation, avant de passer en revue les approches classiques historiquement utilisées, les méthodes modernes fondées sur l'apprentissage profond ainsi que les techniques spécifiques exploitées dans notre pipeline.

2.2.1. DÉFINITIONS

En vision par ordinateur, la segmentation d'objets désigne l'ensemble des techniques qui visent à partitionner une image en régions homogènes correspondant à des entités visuelles significatives, généralement des objets. Contrairement à la simple classification d'image — qui attribue une étiquette globale — la segmentation permet d'analyser la structure spatiale de la scène de manière plus fine, en associant une étiquette à chaque pixel de l'image (Szeliski, 2010).

Il existe plusieurs types de segmentation selon le niveau de précision souhaité :

- Segmentation sémantique (semantic segmentation) : tous les pixels associés à une classe donnée sont regroupés sans distinction d'instances. Par exemple, plusieurs voitures dans l'image seront toutes étiquetées comme « voiture » sans différenciation.

- Segmentation d'instances (instance segmentation) : chaque objet individuel est identifié de manière unique, même s'il appartient à la même classe. Ainsi, deux chaises distinctes seront représentées par deux masques séparés.
- Segmentation panoptique (panoptic segmentation) : elle combine les deux approches précédentes, en fournissant à la fois une étiquette de classe et un identifiant d'instance pour chaque pixel (Kirillov et al., 2019).

La distinction entre détection d'objets et segmentation est également cruciale. La détection produit des boîtes englobantes (bounding boxes) autour des objets cibles, ce qui suffit dans certains cas (ex. : suivi ou comptage d'objets). En revanche, la segmentation va plus loin en fournissant une délimitation précise de la forme des objets, indispensable pour des applications avancées telles que la reconstruction 3D, la manipulation robotique, ou encore la réalité augmentée/mixte (He et al., 2017 ; Everingham et al., 2015).

« Instance segmentation aims to identify all objects in an image, segment them precisely at pixel level, and distinguish between multiple instances of the same class. » (He et al., 2017)
(Traduction libre: <<La segmentation d'instances vise à identifier tous les objets présents dans une image, à en délimiter les contours avec précision au niveau du pixel, et à distinguer chaque occurrence individuelle, même lorsqu'ils appartiennent à la même classe.>>)

Dans le cadre de notre projet, la segmentation d'instances est privilégiée, car elle permet de générer des masques précis associés à chaque objet détecté. Ces masques sont ensuite exploités pour la reconstruction géométrique à partir des images de profondeur. Ce niveau de granularité est essentiel en réalité mixte, où l'objectif est d'assurer une superposition cohérente entre les objets réels et leurs représentations numériques. En effet, des contours flous ou des approximations géométriques compromettent la qualité des interactions spatiales et nuisent à l'immersion de l'utilisateur (Zollhöfer et al., 2018).

2.2.2. APPROCHES TRADITIONNELLES

Avant l'émergence des méthodes basées sur l'apprentissage profond, la segmentation d'objets reposait sur des techniques dites classiques ou fondées sur des règles, qui s'appuyaient sur des propriétés visuelles de bas niveau telles que l'intensité lumineuse, la couleur, les gradients, la texture, ou encore la cohérence spatiale entre pixels. Ces approches suivaient généralement une logique déterministe, où les décisions de segmentation étaient prises en fonction de critères définis manuellement. Elles avaient l'avantage d'être peu gourmandes en calcul, mais souffraient d'un manque de robustesse face aux variations de conditions d'éclairage, aux textures complexes ou à la présence de bruit.

À l'origine, la détection de contours s'appuyait sur des opérateurs de dérivation — Sobel, Prewitt ou Canny — pour mettre en évidence les discontinuités d'intensité et ainsi révéler les frontières potentielles des objets. Si cette approche offre une extraction rapide des bords, elle nécessite presque toujours un post-traitement pour regrouper les segments pertinents et éliminer les artefacts issus du bruit. (Canny J, 1986)

Vint ensuite le seuillage, popularisé par l'algorithme d'Otsu en 1979, qui sépare objets et arrière-plan en choisissant un seuil global maximisant la variance inter-classes. Cette méthode a évolué vers des variantes adaptatives, où l'on calcule des seuils locaux pour compenser les variations d'éclairage, ce qui renforce sa robustesse dans des scènes aux contrastes hétérogènes. Cependant, dès que les niveaux de gris se chevauchent ou que les textures deviennent complexes, ses performances déclinent.(Otsu, N. 1979)

La croissance de région, ou region growing, propose un paradigme différent : à partir de points initiaux (« seeds »), on agrège les pixels voisins dont la valeur reste proche de celle du seed (Adams & Bischof, 1994). Son principal atout réside dans le respect de la cohérence

locale, mais le résultat dépend étroitement du choix des seeds et du seuil de similarité, rendant la méthode sensible au bruit et aux erreurs d'initialisation.

Une approche plus sophistiquée modélise l'image comme un graphe de pixels reliés par des arêtes pondérées selon la similarité d'intensité. GrabCut (Rother et al., 2004) en est l'exemple emblématique : en définissant une fonction d'énergie combinant cohésion interne et contraste entre premier plan et arrière-plan, on obtient une coupe optimale qui segmente l'image de manière plus robuste, au prix d'un coût de calcul et d'une initialisation manuelle.

Enfin, la segmentation par superpixels découpe l'image en petits clusters homogènes, chacun regroupant des pixels proches en couleur et en position. L'algorithme SLIC (Achanta et al., 2012) crée ainsi des régions régulières, réduisant la complexité du traitement (passant de millions de pixels à quelques centaines d'unités), tout en préservant les contours essentiels : ces superpixels servent souvent de prétraitement pour des algorithmes plus avancés.

Malgré leur diversité, ces méthodes partagent des limitations communes. Elles dépendent fortement de paramètres manuels (comme les seuils ou la distance de similarité) et manquent de généralisation. Ainsi, une technique efficace dans un certain contexte peut échouer complètement dans un autre. De plus, elles sont souvent incapables de capturer des structures d'objets complexes, surtout en présence d'occlusions, de faible contraste ou de texture ambiguë. Ces limites ont ouvert la voie à une nouvelle génération de méthodes basées sur l'apprentissage statistique et plus récemment sur les réseaux de neurones profonds, qui apprennent directement des représentations discriminantes à partir de grandes quantités de données annotées.

2.2.3. MÉTHODES D'APPRENTISSAGE PROFOND : ÉTAT DE L'ART

Au cours de la dernière décennie, les méthodes d'apprentissage profond ont transformé le paysage de la vision par ordinateur, et en particulier celui de la segmentation d'images. Grâce aux réseaux de neurones convolutifs (CNN) et à leurs dérivés, il est désormais possible d'atteindre une précision remarquable dans des tâches complexes telles que la segmentation d'objets, même dans des environnements encombrés ou peu structurés. Cette section passe en revue les grandes catégories de segmentation par apprentissage profond, en mettant l'accent sur les méthodes les plus pertinentes pour la segmentation d'objets dans des scénarios de réalité mixte.

2.2.3.1. RÉSEAUX À SEGMENTATION SÉMANTIQUE

La segmentation sémantique est une tâche fondamentale en vision par ordinateur qui consiste à assigner à chaque pixel d'une image une étiquette de classe, sans toutefois distinguer les instances individuelles appartenant à une même catégorie. Cette méthode est particulièrement utile pour les scénarios où la localisation précise de chaque classe est prioritaire, mais où l'identité de chaque objet distinct n'est pas nécessaire (par exemple : détection de routes, bâtiments, végétation, etc.).

Les Fully Convolutional Networks (FCN) de Long et al. (2015) ont marqué un tournant en remplaçant systématiquement les couches entièrement connectées des CNN classiques par des convolutions, ce qui a permis d'obtenir pour la première fois des cartes de prédiction denses et de taille variable adaptées à la segmentation d'images. En tirant parti de cette capacité à conserver la dimension spatiale jusqu'à la sortie, ces modèles ont posé les bases de nombreuses architectures ultérieures, tout en démontrant des performances convaincantes sur des benchmarks tels que PASCAL VOC (Everingham et al., 2015).

Inspirés par cette percée, Ronneberger et al. (2015) ont ensuite proposé U-Net, une topologie en « U » symétrique destinée à la segmentation médicale : grâce à ses connexions croisées entre les phases d'encodage et de décodage, le réseau récupère les informations de localisation fines perdues lors des opérations de sous-échantillonnage, améliorant significativement la précision des contours d'objets de petite taille.

Peu après, DeepLab (Chen et al., 2017) a introduit les convolutions dilatées (ou atrous convolutions) pour accroître le champ réceptif sans sacrifier la résolution, ainsi que l'Atrous Spatial Pyramid Pooling (ASPP), qui agrège des contextes à plusieurs échelles afin de mieux distinguer les structures complexes dans des images naturelles. Cette stratégie multi-échelle s'est avérée particulièrement efficace pour capturer simultanément les détails fins et les informations globales, et DeepLabV3+ a complété ce dispositif par un décodeur léger dédié au raffinage des bords, affinant la segmentation dans des zones où les transitions sont subtiles. Ensemble, ces évolutions ont fait passer la segmentation d'images d'une série d'opérations heuristiques à un apprentissage de bout en bout capable de gérer la diversité des formes, des textures et des échelles rencontrées en vision par ordinateur.

Ces réseaux sémantiques sont bien adaptés aux environnements riches en structure (scènes urbaines, images satellitaires, etc.), mais ils présentent une limite importante : l'impossibilité de différencier plusieurs objets de la même classe (par exemple deux personnes proches seront fusionnées en une seule région « personne »). C'est pourquoi, dans le cadre d'une interaction réaliste en réalité mixte ou d'une reconstruction 3D d'objets spécifiques, une segmentation d'instances est souvent préférée.

2.2.3.2. SEGMENTATION D'INSTANCES

La segmentation d'instances se distingue par sa capacité à reconnaître non seulement la classe des objets, mais aussi à isoler chaque occurrence individuelle au sein

d'une même catégorie. Contrairement à la segmentation sémantique, qui agrège tous les pixels d'une même classe sans distinction, elle génère un masque unique pour chaque objet détecté, offrant ainsi une compréhension beaucoup plus détaillée de la scène (He et al., 2017).

Dans le cadre de ce mémoire, nous porterons un éclairage particulier sur deux architectures phares de cette discipline. D'une part, le modèle YOLO se distingue par sa rapidité et son design monolithique, qui fusionne détection et segmentation en une seule passe pour un traitement temps réel. D'autre part, Mask R-CNN adopte une approche en deux étapes, d'abord localisant les objets via des propositions de régions, puis affinant ces détections par un masque de segmentation précis, ce qui lui confère une grande fiabilité dans des environnements complexes.

Chacune de ces approches sera analysée en profondeur dans les sections suivantes : nous y décrirons l'architecture générale, les mécanismes internes dédiés à la génération des masques, leurs points forts respectifs ainsi que les limites inhérentes à leurs choix de conception.

2.2.3.2.1. L' ALGORITHME YOU ONLY LOOK ONCE (YOLO)

Le modèle YOLO, est une famille d'architectures de détection d'objets en temps réel. Depuis sa première version (Redmon et al., 2016), jusqu'aux plus récentes variantes comme YOLO v5, YOLO v6 et YOLO v8 (Jocher et al., 2023), le principe fondamental demeure : formuler la détection comme un problème de régression unique, permettant de localiser et classifier les objets en un seul passage à travers le réseau.

YOLO fonctionne sur un paradigme end-to-end, en découpant l'image en une grille $S \times S$ et en prédisant, pour chaque cellule, un ensemble de boîtes englobantes (bounding boxes) avec :

- les coordonnées relatives de la boîte (x, y, w, h)
- la probabilité qu'un objet soit présent (score de confiance)
- et la distribution de classes possibles (scores de classification).

Les versions modernes de YOLO, en particulier YOLO v8, intègrent les éléments suivants :

- **Backbone** : module d'extraction de caractéristiques. YOLO v8 repose sur un backbone appelé C2f-CSPDarknet (Cross Stage Partial connections), qui optimise la propagation du gradient tout en réduisant la redondance des calculs. Il incorpore des modules C2f (concat-fuse), plus légers que les résidus classiques.
- **Neck** : module d'agrégation multi-échelle, généralement basé sur le FPN (Feature Pyramid Network) et/ou PAN (Path Aggregation Network), qui assure une bonne propagation des caractéristiques de bas niveau (détails fins) vers le haut niveau (informations sémantiques).
- **Head** : tête de prédiction. YOLO v8 utilise une tête "decoupled", séparant spatialement la régression (localisation des boîtes) et la classification (prédiction des classes), ce qui améliore la convergence et la performance.

À partir de YOLO v5, puis formellement avec YOLO v7-seg et YOLO v8-seg, une tête de segmentation a été ajoutée pour effectuer de la segmentation d'instances. L'approche utilisée est inspirée du modèle YOLACT (Bolya et al., 2019).

Cette tête de segmentation s'articule en deux grandes phases complémentaires. Dans un premier temps, le réseau génère un ensemble fixe de *prototypes de masques* à partir des cartes de caractéristiques intermédiaires du backbone : on obtient typiquement entre 32 et 64 masques, de résolution réduite (par exemple 160×160), qui constituent une base spatiale partagée pour l'ensemble de l'image.

La seconde phase vise à produire, pour chaque objet détecté (boîte englobante et classe), un masque spécifique à l'instance. Pour ce faire, le modèle prédit un vecteur de *coefficients linéaires* qui permet de combiner ces prototypes :

$$M_i = \sum_{j=1}^{k} \alpha_{i,j} \cdot P_j$$

Où M_i est le masque de l'objet i, P_j sont les prototypes communs, et $\alpha_{i,j}$ les coefficients associés à l'objet i.

Enfin, afin de localiser précisément la forme de l'objet, ce masque est rogné à l'intérieur de la boîte englobante correspondante, garantissant que la prédiction est strictement confinée à la région d'intérêt.

YOLO v8-SEG, une extension de l'architecture YOLO dédiée à la segmentation d'instances, repose sur une architecture unifiée qui intègre directement la détection et la segmentation dans une même structure. Cette version améliore plusieurs composantes clés de la segmentation par rapport à ses prédécesseurs. Tout d'abord, elle introduit un encodage spatial plus riche lors de la génération des prototypes, c'est-à-dire les représentations intermédiaires utilisées pour construire les masques d'objets. Ces prototypes sont calculés à partir de couches profondes du réseau, ce qui leur confère une capacité plus expressive pour modéliser des formes complexes.

Par ailleurs, YOLO v8-SEG utilise une interpolation bilinéaire affinée pour remonter les masques à la résolution originale de l'image. Ce choix technique permet une meilleure précision sur les contours, en évitant les artefacts liés à la quantification spatiale. L'un des points forts de cette architecture est également sa fusion directe entre les couches de la tête de détection et celles de segmentation, ce qui favorise un apprentissage multitâche efficace.

Les informations de localisation, de classification et de masquage sont apprises de manière conjointe, dans une perspective cohérente.

L'entraînement du réseau repose sur une fonction de perte globale composite, qui conjugue trois volets complémentaires pour affiner à la fois la localisation, la classification et la qualité des masques. D'abord, une perte de localisation — souvent fondée sur la CIoU ou la DIoU — guide l'ajustement des boîtes englobantes afin de maximiser le recouvrement avec les véritables objets. Ensuite, une perte de classification, qu'il s'agisse d'une Binary Cross-Entropy classique ou d'une Focal Loss plus adaptée aux déséquilibres de classes, évalue la justesse des prédictions de catégories. Enfin, la qualité des masques est optimisée via une perte de segmentation, typiquement une Binary Cross-Entropy appliquée pixel par pixel ou une Dice Loss, qui mesure directement la similarité entre les masques générés et ceux de référence.

2.2.3.2.2. L'ALGORITHME MASK R-CNN

Mask R-CNN est une architecture de segmentation d'instances fondée sur l'extension du paradigme Region-based CNN (R-CNN). Proposé par He et al., (2017), Mask R-CNN enrichit le modèle Faster R-CNN en y ajoutant une branche supplémentaire dédiée à la prédiction de masques de segmentation au niveau du pixel pour chaque instance détectée.

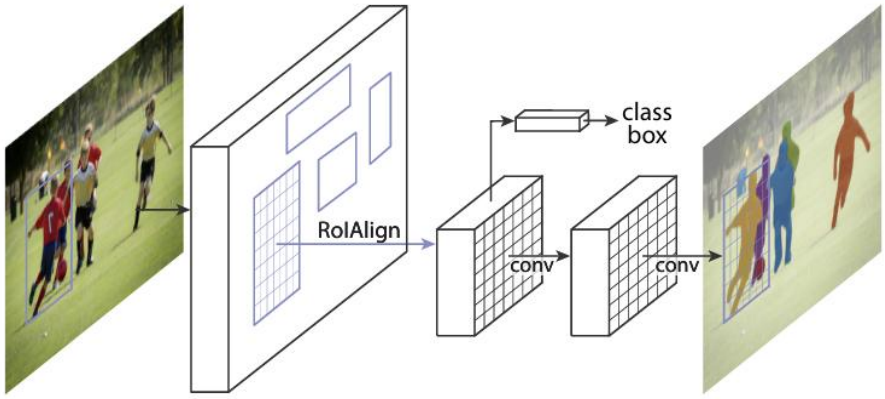

Figure 2.3 — Architecture de l'algorithme Mask R-CNN

© (He et al, 2017)

L'architecture de Mask R-CNN s'articule autour de deux grandes phases complémentaires. D'abord, un backbone convolutionnel transforme l'image d'entrée I, en cartes de caractéristiques F, que scrute ensuite un Region Proposal Network (RPN) pour générer un ensemble de propositions de régions d'intérêt $\{r_i\}_{i=1}^{N}$. Chaque proposition est précisément recadrée sur F grâce à l'opération RoI Align, qui remplace le RoI Pooling en garantissant une correspondance pixel-à-pixel entre la boîte et la carte de caractéristiques.

À partir de ces blocs extraits, le réseau se divise en trois branches : l'une prédit pour chaque RoI la probabilité d'appartenance à chacune des C classes

L'architecture repose sur un fonctionnement en deux étapes, la détection via un Region Proposal Network (**RPN**) et la classification, régression de boîte, et segmentation pour chaque Region of Interest (**RoI)** affinée.

A partir d'une image I, le backbone convolutionnel $f(I)$ extrait des feature maps (carte des caractéristiques) F. Le RPN prend F comme entrée et génère un ensemble de

propositions $\{r_i\}_{i=1}^{N}$. Chaque RoI est ensuite alignée sur F par l'opération RoI Align (plutôt que RoI Pooling), garantissant une correspondance précise entre les pixels :

$$RoI\ Align(F, r_i) \rightarrow f(r_i)$$

À partir de chaque $f(r_i)$, trois branches sont déployées, la classification qui vise à prédire la probabilité de chaque classe $p_i \in R^C$, la régression de boîte pour ajuster les coordonnées de la boîte $t_i \in R^4$ pui la segmentation produisant un masque binaire $m_i \in \{0,1\}^{H\ x\ W}$.

La fonction de perte globale est formulée comme la somme de trois termes :

$$Loss = Loss_{cls} + Loss_{bbox} + Loss_{mask}$$

Où :

- $Loss_{cls}$ est une cross-entropy loss sur la classification,
- $Loss_{bbox}$ est une smooth L1 loss pour la régression des boîtes,
- $Loss_{mask}$ est une binary cross-entropy loss appliquée pixel à pixel entre le masque prédit et le masque cible.

Plus précisément, pour un pixel (x, y) du masque, la perte de segmentation s'écrit :

$$Loss_{mask} = \sum_{(x,y)} [m_{xy}\ log(\widehat{m}_{xy}) + (1 - m_{xy}) log(1 - \widehat{m}_{xy})]$$

Equation 2.1: Equation de la perte de segmentation

où m_{xy} est la valeur vraie (0 ou 1) et $\widehat{m}_{xy}$ est la prédiction du modèle.

Grâce à cette architecture modulaire, Mask R-CNN parvient à séparer finement les objets, même dans des scènes complexes, et propose des masques précis pour chaque instance individuelle. En contrepartie, sa complexité de calcul et son temps d'inférence sont

supérieurs à ceux d'architectures rapides comme YOLO, ce qui peut limiter son utilisation dans des systèmes contraints en ressources ou en temps réel.

2.2.3.3. SEGMENTATION UNIVERSELLE ET MULTIMODALE

La segmentation universelle vise à dépasser les limites traditionnelles des modèles supervisés classiques, qui sont entraînés pour segmenter uniquement un ensemble fixe d'objets appartenant à des classes prédéfinies. L'objectif de la segmentation universelle est de pouvoir détecter, segmenter et interpréter n'importe quel objet visible dans une image, y compris des instances non vues pendant l'entraînement, sans avoir besoin de classes spécifiques (Kirillov et al., 2023).

La segmentation universelle repose sur le paradigme suivant : toute région d'intérêt dans l'image est potentiellement segmentable, sans nécessité d'une correspondance stricte avec une classe sémantique apprise. Certains modèles récents, comme Segment Anything Model (SAM), adoptent une approche basée sur des prompts (indications spatiales ou textuelles) pour initier la segmentation de n'importe quelle structure présente dans l'image.

Formellement, on peut définir la tâche comme suit $S = f(I, p)$. où S est le masque segmenté produit, I est l'image RGB d'entrée, p est un "prompt" (ex. : point, boîte, texte) indiquant la région d'intérêt, f est le modèle universel de segmentation. Ainsi, au lieu d'apprendre une correspondance stricte classe/masque, le réseau apprend une fonction de réponse générale à toute forme de requête spatiale.

2.2.3.3.1. SEGMENT ANYTHING MODEL (SAM)

Le Segment Anything Model (SAM), introduit par Kirillov et al. (2023), est un modèle de segmentation d'objets qui vise à offrir une capacité de segmentation universelle et

conditionnée par des entrées flexibles, appelées prompts. Contrairement aux approches traditionnelles limitées à des catégories fixes, SAM est conçu pour segmenter n'importe quel objet dans n'importe quelle image sans nécessiter d'apprentissage supplémentaire.

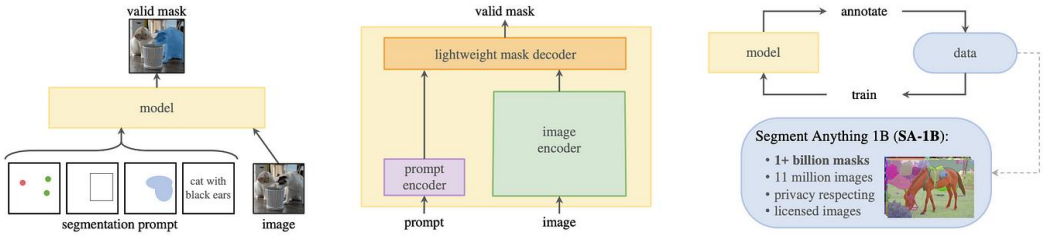

Figure 2.4 — Architecture Segment Anything Model

© *(Kirillov et al. 2023)*

L'architecture de SAM repose sur trois modules distincts travaillant de concert à savoir, l' encodeur de l'image qui est un vision transformer (ViT) (Dosovitskiy et al., 2021) encodant l'image d'entrée $I \in R^{H\,x\,W\,x\,d}$ en un ensemble de représentations spatiales $F(I) \in R^{h\,x\,w\,x\,3}$, ou h, w sont des dimensions réduites et d est la dimension des caractéristiques (features). Formulement: $F(I) = ViT(I)$.

Puis le Prompts Encoder, convertissant les prompts (points, boîtes, etc.) en embeddings vectoriels $p \in R^{d_p}$. Pour un ensemble de n prompts $\{p_1, p_2, p_3, \ldots, p_n\}$, le module génère:

$$p = PromptEncoder(p_1, p_2, p_3, \ldots., p_n)$$

Enfin le Mask Encoder (Encodeur de masque) qui combine la représentation spatiales $F(I)$ et p pour prédire k masques binaires $\{M_1, M_2, M_3, \ldots, M_k\}$ et leurs scores associés $\{s_1, s_2, s_3, \ldots, s_k\}$. Chaque masque M_1est produit par :

$$M_1 = \sigma(D(F(I), p))$$

où σ est la fonction sigmoïde appliquée pixel par pixel et D est le réseau de décodeur (un transformer léger).

2.2.3.3.2. L'ALGORITHME OWL-VIT

L'algorithme **OWL-ViT** constitue une avancée majeure dans la segmentation d'objets à vocabulaire ouvert. Contrairement aux modèles traditionnels basés sur un ensemble de classes fixes, OWL-ViT est capable de détecter et segmenter des objets spécifiés par une requête textuelle libre (Minderer et al., 2022). Cette capacité en fait une approche extrêmement flexible, adaptée aux environnements complexes et dynamiques tels que la **réalité mixte**, où la liste d'objets pertinents peut varier d'une scène à l'autre.

OWL-ViT repose sur l'utilisation conjointe de deux modalités d'entrée : l'image (vision) et le texte (requête ou description d'objet). Le modèle apprend une représentation multimodale en encodant simultanément l'image et la requête textuelle dans un espace latent commun. L'affinité entre ces deux représentations est ensuite exploitée pour guider à la fois la détection d'objets et leur segmentation. Formellement, ce processus peut être représenté par l'équation suivante :

$$Similarity(f_{vision}(I), f_{text}(T)) \rightarrow bboxes, masks$$

Équation 2.2: Equation Similarity OWL-ViT.

où I désigne l'image d'entrée, T la requête textuelle, f_{vision} l'encodeur d'image basé sur un Transformer visuel, f_{text} l'encodeur de texte (généralement dérivé d'un modèle CLIP), et $Similarity$ mesure la proximité entre les représentations afin de localiser les objets pertinents.

2.2.3.4. SEGMENTATION SUR DONNÉES RGB-D

La segmentation d'objets à partir de données **RGB-D** (couleur + profondeur) représente une avancée cruciale dans le domaine de la vision par ordinateur, notamment pour les applications en robotique, réalité augmentée/mixte et navigation autonome. En enrichissant les images classiques avec des mesures de profondeur par pixel, les capteurs RGB-D permettent une perception tridimensionnelle explicite de la scène, ce qui renforce la précision de la segmentation, surtout dans des environnements complexes ou faiblement texturés.

Contrairement à la segmentation basée uniquement sur l'information visuelle (RGB), les méthodes RGB-D exploitent la structure géométrique de la scène à travers des signaux tels que la disparité, les normales de surface, ou les gradients de profondeur. Ces informations supplémentaires permettent une désambiguïsation des objets qui partagent des textures similaires ou qui sont partiellement occultés. De plus, la profondeur aide à estimer la **disposition spatiale** des objets, ce qui est essentiel pour la reconstruction 3D ou l'ancrage d'objets virtuels en réalité mixte.

Pour exploiter l'information de profondeur, la méthode la plus immédiate consiste à la traiter comme un simple quatrième canal, en étendant l'entrée RGB en un tenseur RGB-D de dimension $(H \times W \times 4)$. Si cette approche offre une intégration directe et sans complexité supplémentaire, elle reste limitée : le réseau peine à saisir la nature géométrique intrinsèque des données de profondeur. Pour pallier cette faiblesse, de nombreuses architectures adoptent une fusion multi-modale, en séparant les traitements RGB et profondeur au sein de deux branches distinctes, puis en combinant leurs représentations à différents stades — fusion précoce, intermédiaire ou tardive. C'est le principe exploité par des modèles tels que FuseNet ou RedNet, qui démontrent comment une synergie progressive

entre couleurs et géométrie peut améliorer la segmentation (Hazirbas et al., 2016). Enfin, certaines approches vont encore plus loin en extrayant, à partir des cartes de profondeur, des descripteurs géométriques spécialisés — normals, histogrammes de gradients, coordonnées 3D ou graphes de points — puis en les injectant dans le réseau via des modules adaptés. Ces encodages dédiés, illustrés notamment par les travaux de Qi et al. (2017), permettent de valoriser pleinement la richesse spatiale apportée par la profondeur pour des segmentations plus précises et robustes.

Les approches RGB-D ont démontré de meilleures performances en segmentation sémantique et d'instances sur des jeux de données comme SUN RGB-D (Song et al., 2015) ou NYU Depth V2. En particulier, les objets faiblement visibles ou faiblement contrastés bénéficient d'un gain significatif en précision.

L'exploitation de la profondeur en segmentation reste délicate : la qualité et la fiabilité des cartes varient selon le capteur et les conditions d'éclairage, générant souvent du bruit ou des discontinuités aux bords des objets. Par ailleurs, les architectures multibranches ou les modules géométriques ajoutés pour valoriser ces données alourdissent la charge computationnelle, au prix d'une inférence plus lente et plus coûteuse en ressources.

2.3. LA RECONSTRUCTION ET LA GÉNÉRATION DE MAILLAGES

Dans le contexte de la vision par ordinateur et de la réalité mixte, la reconstruction tridimensionnelle constitue une étape clé pour représenter numériquement des objets ou des scènes physiques à partir de données capteurs. Elle permet de générer des modèles exploitables pour l'interaction, la simulation ou la visualisation en 3D. Contrairement à une simple détection ou segmentation, la reconstruction vise à restituer la géométrie complète

d'un objet dans l'espace, à partir d'informations partielles, souvent issues de capteurs RGB-D ou de stéréovision.(Newcombe et al., 2011)

En vision par ordinateur, la reconstruction 3D regroupe l'ensemble des méthodes permettant de reconstituer la forme et la structure spatiale d'une scène ou d'un objet à partir d'observations visuelles. Ces observations peuvent provenir de caméras RGB, de capteurs de profondeur, ou de dispositifs multi-capteurs. L'objectif est de produire une représentation géométrique exploitable sous forme de maillages polygonaux, de nuages de points, ou encore de volumes implicites (Szeliski, 2010).

Un maillage 3D est une structure polygonale décrivant la surface d'un objet ou d'un environnement. Il est composé de sommets (points), arêtes (segments) et faces (souvent triangulaires), et permet une visualisation continue, une simulation physique ou une intégration dans des environnements virtuels (Kazhdan & Hoppe, 2013).

Les représentations implicites, quant à elles, modélisent une surface comme l'iso-surface d'une fonction continue $f: R^3 \rightarrow R$, telle qu'une fonction d'occupation (Occupancy Function) (Mescheder et al., 2019) ou de distance signée (Signed Distance Function, SDF) (Park et al., 2019). Ces fonctions peuvent être représentées analytiquement ou approximées via un réseau de neurones. Cette approche permet une reconstruction continue, indépendante de la résolution initiale du capteur.

2.3.1. REPRÉSENTATION IMPLICITE DES SURFACES

La représentation implicite des surfaces est une approche moderne qui consiste à modéliser la géométrie d'un objet ou d'une scène à l'aide d'une fonction continue définie sur l'espace tridimensionnel. Plutôt que de stocker explicitement des sommets ou des faces

comme dans les maillages traditionnels, on définit une fonction $f: R^3 \rightarrow R$ dont le niveau iso-surface (typiquement $f(x) = 0$) représente la surface de l'objet.

Dans le domaine de la reconstruction implicite, deux paradigmes se sont imposés pour représenter la géométrie des objets au sein d'un espace continu. D'une part la Signed Distance Function (SDF), qui attribue à chaque point $x \in R^3$, la distance signée à la surface la plus proche. La valeur est négative à l'intérieur de l'objet, positive à l'extérieur, et nulle sur la surface (Park et al., 2019). D'autre part la Occupancy Function, qui attribue à chaque point une probabilité $f(x) \in [0, 1]$ d'être à l'intérieur de l'objet. Cette approche est souvent utilisée avec des réseaux neuronaux binaires entraînés à classifier les points comme « occupés » ou « libres » (Mescheder et al., 2019).

L'intérêt principal de ces représentations est leur **continuité** : elles permettent de générer des surfaces de haute résolution sans dépendre directement de la résolution du capteur ou du nuage de points. De plus, elles facilitent la reconstruction de **formes complexes**, même à partir de données partielles ou bruitées, en apprenant une régularisation implicite via le réseau de neurones.

Dans certains cas, un réseau de type MLP est entraîné à approximer une fonction d'occupation, à partir de points positifs (surfaces observées) et négatifs (volume libre). Une fois le réseau entraîné, l'évaluation sur une grille 3D régulière permet d'extraire une isosurface par la méthode de Marching Cubes, générant ainsi un maillage régulier et fermé.

2.3.2. RECONSTRUCTION À PARTIR DE NUAGES DE POINTS

Les caméras RGB-D, telles que celles intégrées au HoloLens 2, permettent de capturer à la fois une image couleur (RGB) et une carte de profondeur (D). La combinaison de ces deux modalités permet de reconstituer un nuage de points 3D représentant la

géométrie visible d'une scène à un instant donné. Chaque pixel (u, v)de l'image peut être projeté dans l'espace 3D grâce à sa valeur de profondeur z et aux paramètres intrinsèques de la caméra $(f_x, f_y, c_x, c_y,)$, selon les formules suivantes :

$$x = \frac{(u - c_x) \cdot z}{f_x},\ y = \frac{(v - c_y) \cdot z}{f_y}, z = \frac{z}{scale}$$

Équation 2.3: formules des coordonnées 3D à reconstruire.

© *(Mescheder et al., 2019)*

où *scale* est le facteur de conversion des unités de profondeur (souvent millimètres vers mètres).

Cette projection permet de transformer un masque binaire 2D (issu d'un algorithme de segmentation) en un nuage de points 3D représentant la forme d'un objet dans l'espace. Ces points peuvent ensuite être enrichis avec les valeurs de couleur correspondantes pour une visualisation ou un traitement ultérieur. (Newcombe et al., 2011). La qualité du nuage de points dépend toutefois fortement de la précision de la carte de profondeur et de l'exactitude des masques segmentés. Des points bruités, manquants ou mal alignés peuvent affecter la surface reconstruite. Des techniques de filtrage ou de clustering spatial, comme DBSCAN, peuvent alors être appliquées pour renforcer la cohérence du nuage. (Espinosa et al., 2020)

2.3.3. APPRENTISSAGE DE FONCTIONS D'OCCUPATION AVEC RÉSEAUX DE NEURONES

L'apprentissage de fonctions d'occupation s'appuie sur l'idée d'approcher la surface d'un objet à travers une fonction continue, implémentée par un réseau de neurones, qui prédit si un point de l'espace tridimensionnel appartient ou non à un objet. Cette approche constitue une alternative puissante aux représentations explicites (voxel grids, meshes, point clouds), en permettant une modélisation plus fine et continue des géométries 3D. Le réseau

utilisé est généralement un MLP (Multilayer Perceptron) composé de plusieurs couches entièrement connectées avec des activations non linéaires (ReLU). L'entrée du réseau est un vecteur de coordonnées 3D (x, y, z) normalisé dans un espace borné (typiquement $[-1, 1]^3$), et la sortie est une valeur scalaire $\hat{O} = [-1, 1]$, représentant la probabilité que ce point soit à l'intérieur de la surface de l'objet :

$$f_\theta : R^3 \rightarrow [0, 1], f_\theta(x, y, z) = \hat{O}$$

Le processus d'entraînement repose sur un double échantillonnage : d'une part, on prélève des points positifs situés à proximité des surfaces des objets visibles, ces surfaces étant définies par les masques projetés ; d'autre part, on génère des points négatifs répartis de manière uniforme dans l'ensemble du volume d'intérêt, afin d'exposer le modèle à des exemples à la fois de la géométrie des objets et de l'espace libre.

Les points positifs sont associés à la valeur cible $O = 1$, et les négatifs à $O = 0$. Le réseau peut être optimisé à l'aide d'une Binary Cross-Entropy Loss, notée :

$$Loss_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} [o_i log(\hat{o}_i) + (1 - o_i) log(1 - \hat{o}_i)]$$

où o_i est la valeur réelle (1 pour les points occupés, 0 sinon), et $\hat{o}_i$ la prédiction du réseau pour le point i.

2.3.4. EXTRACTION DE MAILLAGES PAR ÉVALUATION SUR GRILLE ET ISOSURFACES

Une fois la fonction d'occupation $f_\theta(x, y, z)$ entraînée, il devient possible de **générer un maillage 3D** représentant la surface de l'objet en extrayant l'isovaleur τ (souvent $\tau = 0.5$) correspondant à la frontière entre l'intérieur et l'extérieur.Pour cela, on évalue la sortie du réseau sur un maillage régulier tridimensionnel de points (x, y, z) répartis uniformément

dans un volume cubique normalisé (typiquement $[-1,1]^3$). Chaque point est passé dans le réseau, produisant une valeur d'occupation :

$$f_\theta(x,y,z) \approx \hat{O}$$

On obtient ainsi un volume scalaire 3D indiquant, pour chaque point de la grille, la probabilité d'occupation.

Ce volume est ensuite transformé en surface maillée via un algorithme d'extraction d'isosurfaces, le plus courant étant **Marching Cubes** (Lorensen et Cline, 1987). Cette méthode parcourt les cubes définis par la grille 3D, détermine la position des sommets sur les arêtes où $f_\theta(x,y,z) = \tau$, et assemble des triangles pour construire une surface polygonale.

On formalise ce processus en considérant que le réseau fournit, pour chaque point $(x,y,z) \in R^3$, une valeur prédictive $V(x,y,z)$. La surface SSS se définit alors comme l'ensemble des points où cette valeur atteint le seuil critique correspondant à la frontière entre intérieur et extérieur de l'objet. Autrement dit, on écrit :

$$S = \{(x,y,z) \in R^3 \mid f_\theta(x,y,z) = \tau\}$$

où τ vaut généralement 0 dans le cas d'une Signed Distance Function (SDF) — puisque la distance signée s'annule sur la surface — et $\tau = 0.5$ pour une Occupancy Function, où on considère qu'à $V = 0.5$, la probabilité d'appartenance bascule entre « libre » et « occupé ». C'est ce seuil qui guide ensuite l'extraction de la géométrie, typiquement via un algorithme de contouring continu comme Marching Cubes.

Pour retrouver les dimensions originales des objets, les sommets du maillage sont dé-normalisés à partir du centre et de l'échelle calculés lors de la préparation des

échantillons. Le maillage final peut alors être exporté dans des formats standards comme *.obj* pour visualisation ou exploitation.

2.4. ANALYSE CRITIQUE DE LA LITTÉRATURE

À travers cette revue de littérature, nous avons exploré trois axes majeurs qui sous-tendent la problématique de la reconstruction 3D d'objets en contexte de réalité mixte : l'évolution des dispositifs XR, les avancées en segmentation d'objets, et les méthodes récentes de modélisation géométrique à partir d'images RGB-D. Chacun de ces domaines a connu des progrès significatifs, mais leur intégration conjointe dans un système cohérent reste encore un défi ouvert dans la littérature scientifique.

Les dispositifs immersifs comme le HoloLens 2 illustrent la maturité croissante des technologies matérielles. Ces appareils permettent une capture synchronisée de données visuelles et spatiales (RGB-D), tout en offrant des interfaces naturelles de visualisation et d'interaction. Cependant, les capacités de perception contextuelle intégrées à ces dispositifs demeurent limitées. En particulier, les mécanismes embarqués de reconnaissance d'objets ou de reconstruction 3D restent rudimentaires, et sont rarement capables de traiter en temps réel des scènes complexes comportant des objets partiellement visibles, faiblement texturés ou en mouvement.

Du côté de la segmentation d'objets, les modèles de deep learning ont ouvert des perspectives nouvelles. YOLO, Mask R-CNN, ou encore SAM (Segment Anything Model), proposent des performances impressionnantes en détection et segmentation d'instances, même dans des conditions visuelles difficiles. Toutefois, ces modèles sont souvent conçus pour des tâches 2D standards, et ne tiennent pas compte de la géométrie de la scène ou des données de profondeur. De plus, la plupart des benchmarks s'appuient sur des jeux de données génériques (COCO, PASCAL VOC), peu représentatifs des cas d'usage réels en

réalité mixte. Très peu de travaux ont envisagé leur intégration directe dans des pipelines de reconstruction 3D à partir d'images RGB-D, notamment pour des environnements non contrôlés.

En ce qui concerne la reconstruction 3D, les méthodes classiques fondées sur la triangulation explicite (Delaunay, Poisson, etc.) présentent une bonne robustesse, mais manquent de flexibilité. Les approches récentes basées sur les fonctions d'occupation implicites, comme les réseaux MLP apprenant à prédire la présence de matière dans l'espace (Occupancy Networks, Deep SDF), constituent une avancée notable. Elles permettent une modélisation continue, paramétrique et différentiable des surfaces. Cependant, leur efficacité dépend étroitement de la qualité des points d'entrée (issues de masques ou de depth maps) et nécessite un entraînement adapté à chaque instance ou catégorie d'objet.

Un constat transversal ressort de cette analyse : les approches de segmentation, de perception 3D et d'affichage immersif évoluent majoritairement de façon parallèle, sans réelle intégration dans une chaîne de traitement unifiée. De nombreux travaux s'arrêtent à la segmentation ou à la reconstruction, sans explorer la complémentarité entre ces étapes. En particulier, le couplage entre des masques segmentés automatiquement et une reconstruction implicite via réseau de neurones reste très peu documenté dans le contexte des scènes capturées avec des capteurs RGB-D embarqués.

C'est précisément dans ce vide que s'inscrit notre contribution. Nous proposons une architecture hybride intégrée dans laquelle :

- les données RGB-D sont capturées via une application développée sur le HoloLens;
- les objets sont segmentés automatiquement à l'aide de trois modèles de pointe (YOLO, Mask R-CNN, SAM) ;
- les masques sont projetés en nuages de points 3D ;

- ces points sont ensuite utilisés pour entraîner un réseau de type MLP qui apprend une fonction d'occupation ;
- Les maillages finaux sont extraits par évaluation du réseau sur une grille 3D régulière puis extraction d'isovaleurs via Marching Cubes.

Ce pipeline complet, de la capture à la reconstruction, présente plusieurs intérêts : il est modulaire, reproductible, et repose sur des outils open source. Il permet également une évaluation comparative des modèles de segmentation dans des conditions réalistes, sur un jeu de données personnalisé capturé avec un dispositif immersif. Enfin, il met en lumière les conditions nécessaires à la combinaison efficace entre apprentissage supervisé (segmentation) et apprentissage implicite (fonction d'occupation), en vue de produire des représentations 3D exploitables dans des environnements de réalité mixte.

Ainsi, notre approche ne se limite pas à une contribution algorithmique isolée : elle s'inscrit dans une perspective plus large de fusion intelligente des techniques de perception visuelle et de reconstruction géométrique, afin de favoriser le développement de systèmes de réalité mixte plus intelligents, plus précis, et mieux adaptés aux besoins concrets des utilisateurs.

CHAPITRE III

CONTENU

Ce chapitre décrit de manière détaillée le processus d'implémentation du pipeline proposé pour la segmentation d'objets et la reconstruction 3D en réalité mixte. Contrairement aux sections précédentes, centrées sur la revue des concepts et approches existantes, l'objectif ici est de présenter comment nous avons procédé concrètement pour mettre en œuvre notre solution, depuis la capture des données jusqu'à la génération finale des maillages.

Nous commençons par présenter l'environnement de travail utilisé au sein du laboratoire, en mettant en lumière les outils logiciels et matériels mobilisés. Cela inclut notamment le casque HoloLens 2, l'infrastructure serveur et les bibliothèques de traitement. Ensuite, nous détaillons chaque étape de notre pipeline. La première phase consiste à développer une application personnalisée sur le HoloLens 2 pour capturer des images synchronisées RGB et profondeur. Ces images sont automatiquement transférées vers un serveur distant où les traitements sont déclenchés. La deuxième étape concerne la segmentation des objets, réalisée à l'aide de trois modèles que nous avons entraînés et évalués : YOLO, Mask R-CNN et SAM. Nous expliquons ici comment le jeu de données a été constitué (à l'aide de LabelMe), comment les annotations ont été formatées, et comment les modèles ont été entraînés dans des environnements spécifiques avec des hyper-paramètres choisis.

Enfin, la dernière partie du chapitre est consacrée à la reconstruction 3D des objets segmentés. Contrairement aux approches classiques basées sur Delaunay (Delaunay, 1934) ou des méthodes géométriques explicites, nous avons mis en œuvre une méthode de

reconstruction implicite fondée sur l'apprentissage d'une fonction d'occupation par réseau de neurones (Mescheder et al., 2019). Nous décrivons ici les scripts développés, la logique de traitement des masques, la back-projection des points RGB-D, l'échantillonnage des données 3D, l'entraînement du MLP, et l'extraction du maillage à l'aide de Marching Cubes.

Ce chapitre se veut donc une documentation technique détaillée et structurée du système réalisé, destinée à permettre sa reproduction ou son extension dans d'autres contextes.

3.1. CADRE EXPÉRIMENTAL

Cette section présente l'environnement dans lequel ont été réalisées les expérimentations techniques de ce projet de recherche. L'objectif est de contextualiser le développement et la mise en œuvre de la solution proposée, en décrivant à la fois les ressources logicielles et matérielles utilisées, ainsi que l'environnement principal de réalisation des travaux.

Tout d'abord, nous décrivons le Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA) (Bouchard et al., 2014) où le projet a été conduit. Ce laboratoire fournit une infrastructure adaptée aux recherches en vision par ordinateur, intelligence artificielle et technologies immersives. Ensuite, nous détaillons l'environnement matériel utilisé pour l'ensemble des étapes de notre pipeline : acquisition des données, traitement des images, entraînement des modèles de segmentation, et génération des maillages 3D.

3.1.1. DESCRIPTION DU LIARA

Le **LIARA** est un espace de recherche avancée situé à l'Université du Québec à Chicoutimi (UQAC). Il a pour vocation de développer des environnements intelligents

capables de percevoir, comprendre et s'adapter aux activités humaines en temps réel. Ce laboratoire s'inscrit dans une approche multidisciplinaire, mêlant informatique, génie, ergothérapie et sciences humaines, dans le but de créer des habitats technologiques centrés sur l'humain.

Le cœur du LIARA est un logement intelligent grandeur nature, d'environ 100 m^2, conçu pour simuler un appartement réel. Cet espace comprend une cuisine, un petit salon, une chambre et une salle de bain, tous équipés de capteurs, actionneurs et dispositifs d'interaction contextuelle . Cette infrastructure permet de capter des données hétérogènes, synchronisées, et contextualisées dans un environnement réel mais contrôlé.

La Figure 3.1 ci-dessous illustre un aperçu de cet environnement intelligent. On y retrouve notamment les éléments domotiques intégrés, les équipements électroménagers instrumentés, ainsi que les systèmes informatiques embarqués, comme l'armoire réseau ou les interfaces de supervision. Ce dispositif technique est connecté à un serveur central qui enregistre, traite et visualise les données, en offrant une vision complète des états de l'environnement et des interactions de l'usager.

Figure 3.1 —L'habitat intelligent du LIARA
© (Bouchard et al., 2014)

Dans le contexte de notre projet de segmentation d'objets et reconstruction 3D pour la réalité mixte, le LIARA joue un rôle central. Il a été utilisé comme cadre d'expérimentation pour la collecte de données RGB-D, à l'aide du casque HoloLens 2. Son environnement structuré mais réaliste offre des conditions optimales pour tester la robustesse des algorithmes de perception visuelle, tout en permettant de simuler des scénarios complexes (occlusions, éclairage variable, objets partiellement visibles).

Grâce à la configuration domotique du LIARA, il a été possible de capturer des séquences d'images et de profondeur en contexte réel, tout en assurant une bonne traçabilité des objets dans l'espace. De plus, l'architecture réseau existante a facilité l'intégration fluide entre l'appareil HoloLens et le serveur backend, utilisé pour les traitements automatiques tels que la segmentation d'images, la projection de nuages de points et la reconstruction de maillages 3D.

3.1.2. ENVIRONNEMENT MATÉRIEL

La mise en œuvre de notre solution expérimentale s'appuie sur une infrastructure matérielle sophistiquée, combinant des dispositifs embarqués pour la capture des données et une station de travail à haute performance dédiée au traitement. Au cœur de ce système, on retrouve le casque autonome de réalité mixte HoloLens 2, qui joue un rôle central dans l'acquisition des données. Celui-ci intègre plusieurs capteurs spécialisés permettant la capture synchronisée d'images RGB et de cartes de profondeur au sein d'environnements réels. Parmi ces capteurs figurent une caméra couleur principale (PV Cam) destinée à l'imagerie RGB, un module spécifique (Depth Module) chargé de la perception tridimensionnelle de la scène, ainsi que des microphones et capteurs inertiels assurant la reconnaissance vocale et gestuelle. Enfin, le casque est équipé d'unités optiques dites MOMAs, qui permettent la projection d'éléments numériques directement dans le champ visuel de l'utilisateur.

Une application spécialement conçue et développée pour ce dispositif assure la collecte automatisée de séquences d'images enrichies par des données de profondeur et des informations temporelles précises. Une fois capturées, ces séquences RGB-D sont automatiquement transmises à un serveur distant, où elles déclenchent les différentes étapes du traitement, notamment la segmentation des objets et la reconstruction de leurs formes en trois dimensions.

Le traitement des données capturées est effectué sur une station de travail mobile particulièrement puissante, conçue pour répondre efficacement aux exigences intensives des tâches de vision par ordinateur et de reconstruction 3D. Cette station dispose d'un processeur Intel Core Ultra 9, particulièrement performant en calcul parallèle, ce qui garantit une exécution fluide lors de l'entraînement des modèles complexes et des calculs

géométriques avancés. Sa mémoire vive de 32 Go DDR5 permet le traitement sans ralentissement de séquences RGB-D en haute résolution ainsi que l'exécution simultanée de multiples modules logiciels.

Pour l'accélération matérielle de l'inférence et de l'entraînement des réseaux de neurones, la station embarque une carte graphique NVIDIA GeForce RTX 4060 Laptop disposant de 8 Go de mémoire vidéo, exploitant pleinement les capacités offertes par CUDA. Enfin, son système de stockage, un SSD NVMe de 1 To, offre des vitesses élevées en lecture et en écriture, essentielles pour une manipulation efficace de vastes jeux de données.

Grâce à cette configuration matérielle avancée, notre pipeline bénéficie de la fiabilité et de la rapidité nécessaires à son bon fonctionnement, assurant ainsi une transition fluide depuis la capture initiale des données jusqu'à la génération finale des maillages tridimensionnels.

3.2. IMPLÉMENTATION DE L'APPLICATION HOLOLENS

L'application développée pour le HoloLens 2 constitue l'élément central du processus d'acquisition des données dans notre pipeline. Elle est spécifiquement conçue pour capturer, dans un environnement réel, des paires synchronisées d'images couleur (RGB) et de cartes de profondeur. Chaque capture est automatiquement associée à un identifiant de session unique ainsi qu'à un horodatage précis, garantissant une structuration temporelle cohérente des données tout au long de la session. L'enregistrement local est conçu pour être résilient et structuré : les fichiers sont organisés automatiquement en répertoires hiérarchiques selon la session et l'instant de capture. L'application assure également une gestion interne des états d'acquisition (initialisation, enregistrement, arrêt) et une file d'attente mémoire est utilisée pour tamponner les captures en cas de saturation temporaire des ressources du casque.

L'application repose sur une architecture logicielle modulaire implémentée sous Unity. Elle exploite directement les capteurs embarqués du HoloLens 2, notamment la caméra RGB (PV Cam) pour l'imagerie couleur et le module de profondeur (Depth Module) pour la perception spatiale. Lors de l'activation d'une session, l'application initialise un système de suivi temporel, et à chaque capture, elle enregistre à la fois les images brutes et les paramètres intrinsèques de la caméra, tels que les matrices de projection ou la résolution du capteur actif. Les données sont encodées dans des formats standards typiquement PNG pour les images couleur, PNG 16 bits pour les cartes de profondeur et accompagnées de fichiers de métadonnées au format JSON. Ces fichiers contiennent notamment l'horodatage de capture, l'identifiant de session, la résolution d'acquisition, ainsi que des informations spatiales utiles pour la reconstruction ultérieure.

Bien que les données puissent ensuite être transmises à un serveur distant pour traitement, le cœur de cette architecture réside dans la robustesse et la précision de l'acquisition locale. Le casque agit comme un nœud autonome de perception visuelle, capable de capturer des scènes complexes en temps réel, tout en maintenant une structuration rigoureuse des informations associées. Cette infrastructure logicielle permet non seulement une synchronisation fine entre les données RGB et profondeur, mais elle garantit également leur cohérence spatiale et temporelle, condition essentielle pour la qualité des traitements de segmentation et de reconstruction 3D appliqués en aval. La figure 3.2 qui suit montre un aperçu de l'interface de l'application hololens.

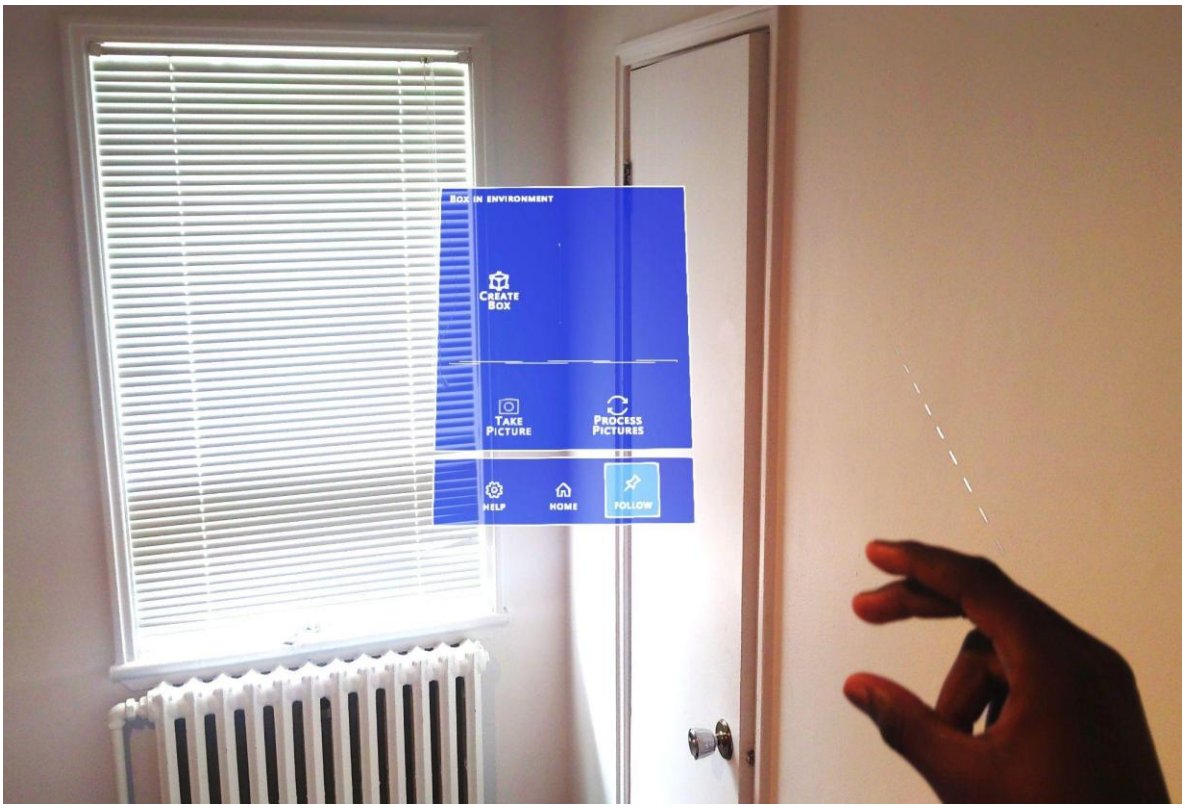

Figure 3.2 — Interface graphique de l'application HOLOLENS
© Abdoul-Wahabou H. Tiambou, Nguyen et al. (2023)

L'application HoloLens initialement inspirée d'un projet de Nguyen et al. (2023), proposant une interaction multimodale basée sur la voix et les gestes dans un environnement HoloLens 2, en s'appuyant sur le Mixed Reality Toolkit (MRTK), une bibliothèque open source maintenue par Microsoft, conçue pour faciliter le développement d'interfaces et d'interactions en réalité mixte sur HoloLens et autres dispositifs immersifs (Microsoft, 2023).

3.3. FLUX DES DONNÉES ET COMMUNICATION

L'application HoloLens 2 repose sur une infrastructure de communication en temps réel, basée sur le protocole WebSocket, pour assurer une transmission fluide, bidirectionnelle et persistante des données entre le casque et le serveur distant. Cette architecture s'appuie sur le système open source HL2SS (Di Benedetto, 2021), conçu

pour exploiter les capacités de streaming des différents capteurs du HoloLens 2 à travers une interface WebSocket extensible (HL2SS GitHub).

Dès l'initialisation de l'application, un canal WebSocket est ouvert avec le backend. Cette phase est critique : tant que la connexion n'est pas établie avec succès, l'application reste dans un état d'attente bloquant. Une fois la liaison validée, un message de confirmation est reçu depuis le serveur, autorisant alors le déverrouillage de l'interface utilisateur et la création d'une session active de capture.

L'utilisateur peut dès lors interagir avec l'interface holographique pour déclencher la prise de vues RGB-D. À chaque capture, les fichiers générés — image couleur, carte de profondeur et métadonnées JSON (incluant l'horodatage, l'ID de session, les paramètres de caméra) — sont encodés localement puis envoyés immédiatement via le canal WebSocket, sous forme de messages binaires structurés. Ce mécanisme garantit un flux continu, faible latence et évite les surcharges liées au protocole HTTP traditionnel. Le serveur distant, à l'écoute permanente du canal WebSocket, reçoit, décode et enregistre les fichiers au fur et à mesure. Chaque réception donne lieu à un accusé de réception transmis au casque, ce qui déclenche l'effacement du tampon de capture local. En cas de perte de connexion, une logique de reconnexion automatique est mise en place, et les captures sont temporairement stockées dans une file tampon jusqu'au rétablissement du lien.

En plus du transfert de données, le canal WebSocket est également utilisé pour transmettre des commandes système, notamment lorsque l'utilisateur déclenche le traitement des données à l'aide de l'interface. Une commande spécifique est alors envoyée au serveur, identifiant la session à traiter. Le backend démarre alors le pipeline de segmentation et de reconstruction 3D, et peut en retour notifier le client de l'état d'avancement ou de l'achèvement du traitement.

Grâce à l'usage de WebSocket, l'ensemble du pipeline repose sur une communication événementielle, efficace et interactive, parfaitement adaptée aux contraintes de la réalité mixte et aux exigences de traitements distants complexes.

3.4. IMPLÉMENTATION DE LA SEGMENTATION DES OBJETS

Dans cette section, nous détaillons l'ensemble du processus mis en place pour l'implémentation de la segmentation des objets à partir des données capturées. Ce processus débute par la constitution d'un jeu de données sur mesure, structuré et annoté spécifiquement pour répondre aux exigences de notre pipeline hybride. Ce jeu de données constitue la base sur laquelle les algorithmes de segmentation supervisés ont été entraînés et évalués. Par la suite, nous présentons les choix méthodologiques ayant conduit à la sélection des modèles de segmentation utilisés dans notre système, en soulignant leurs complémentarités techniques. L'ensemble de cette chaîne – de la collecte des annotations à l'inférence – joue un rôle déterminant dans la qualité des résultats produits en aval, notamment pour la reconstruction 3D à partir de masques segmentés.

3.4.1. JEU DE DONNÉES

Le jeu de données utilisé dans ce projet a été constitué au sein du laboratoire LIARA, dans un environnement contrôlé permettant la capture d'objets usuels sous des conditions d'éclairage et de positionnement variées. Il comprend un total de 534 images RGB annotées manuellement à l'aide de l'outil LabelMe, accompagnées de leurs masques de segmentation polygonaux. L'ensemble est structuré en trois sous-ensembles : 427 images pour l'entraînement, 53 pour la validation et 54 pour les tests. Chaque instance annotée est associée à l'un des 20 objets sélectionnés pour l'étude, allant de petits outils comme des tournevis ou des pinces, à des objets technologiques tels que des caméras, des claviers ou des téléphones cellulaires. Le format de ce jeu de données est entièrement compatible avec

les modèles de segmentation modernes, et a été organisé selon la convention COCO (Lin et al., 2014) pour garantir l'interopérabilité avec différents pipelines d'entraînement.

3.4.1.1. CONSTITUTION DU JEU DE DONNÉES D'ENTRAÎNEMENT

La constitution de ce jeu de données a nécessité une étape méticuleuse de collecte et d'annotation. Chaque image a été capturée via une application HoloLens 2, garantissant une résolution homogène et une qualité de perception constante. Les objets ont été disposés dans divers agencements afin de maximiser la variabilité des poses, des orientations et des degrés d'occlusion. Une partie significative de ce jeu de données a été constituée par Rani Baghezza, dont la contribution à la phase initiale de captation a permis d'accélérer le processus de collecte et d'assurer une diversité représentative des scènes capturées.

Le processus d'annotation a été réalisé à l'aide de LabelMe (Russell et al., 2008), un outil open source permettant une annotation fine par polygones, facilitant la délimitation précise des objets d'intérêt dans les images RGB.

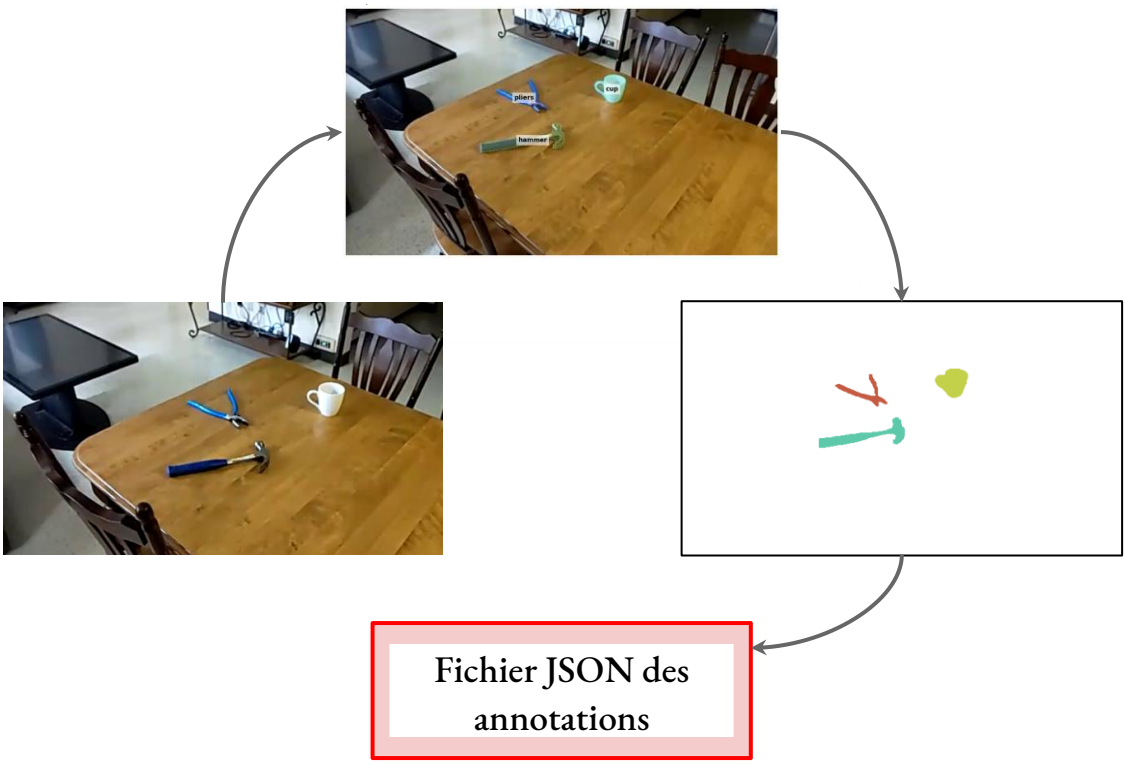

Figure 3.3 – Exemple d'annotation manuelle d'une image RGB

Les annotations produites ont ensuite été exportées et converties au format COCO à l'aide d'un script personnalisé. Cette conversion a permis de générer des fichiers de segmentation d'instances, où chaque instance est décrite par sa segmentation, sa boîte englobante, sa classe, et son identifiant d'image. Une attention particulière a été portée à la cohérence des identifiants d'objet et à la non-redondance des segments. La structuration du répertoire respecte une hiérarchie claire avec les sous-dossiers images, annotations et labels. Cette organisation permet de séparer les ressources brutes (images RGB), les fichiers de vérité terrain (masks LabelMe convertis) et les annotations COCO nécessaires à l'entraînement des modèles.

3.4.1.2. PRÉTRAITEMENT DES DONNÉES

Avant l'entraînement, les données ont été soumises à une série d'opérations de prétraitement visant à assurer la qualité et l'homogénéité des entrées. Chaque image a été redimensionnée ou recadrée, le cas échéant, pour correspondre aux dimensions attendues par les réseaux convolutifs utilisés. Les masques de segmentation ont été convertis en images binaires à partir des annotations polygonales, en veillant à préserver la correspondance entre les indices de classe et les couleurs d'affichage prédéfinies.

Par ailleurs, un mappage explicite entre les identifiants numériques des classes et leur nom a été défini dans ce même fichier YAML, accompagné d'une palette de couleurs RGB codée manuellement pour faciliter la visualisation des prédictions. Ces couleurs, cohérentes avec celles utilisées lors de l'annotation dans LabelMe, permettent une validation visuelle rapide des performances du modèle après entraînement.

3.4.2. CHOIX DES ALGORITHMES DE SEGMENTATION

Le choix des algorithmes de segmentation repose sur une évaluation rigoureuse des besoins spécifiques liés à la nature de notre pipeline, combinant réalité mixte, perception spatiale et reconstruction 3D. Notre objectif n'était pas seulement d'identifier les objets dans des environnements complexes, mais aussi de générer des masques de segmentation de haute qualité exploitables pour la reconstruction géométrique. Ainsi, nous avons sélectionné trois approches complémentaires : YOLO, Mask R-CNN et SAM couple a Faster RCNN.

Le premier modèle exploré est YOLO, une extension de la célèbre famille de modèles YOLO adaptée à la segmentation d'instances. Il présente l'avantage d'un traitement en temps réel, tout en maintenant une précision raisonnable sur des objets bien définis. Son architecture unifiée et sa vitesse d'inférence en font un candidat idéal pour des applications

embarquées comme celle du HoloLens, bien que sa précision sur des contours fins ou des objets partiellement occultés puisse être limitée.

Le second modèle intégré est Mask R-CNN, une architecture à deux branches issue de Faster R-CNN, qui ajoute un masque binaire pour chaque instance détectée. Son avantage réside dans sa précision et sa robustesse, en particulier sur des images présentant des objets de taille variable, des chevauchements ou des formes irrégulières. En contrepartie, ce modèle est plus coûteux en calculs et en mémoire, ce qui en limite l'utilisation dans des contextes embarqués mais le rend très pertinent pour un traitement différé côté serveur.

Enfin, nous avons intégré le modèle SAM, développé par Meta AI, qui marque une avancée majeure dans le domaine de la segmentation d'images. Reposant sur une architecture de type Transformer, SAM a été pré-entraîné sur un corpus massif de plus d'un milliard de masques, ce qui lui confère une capacité de généralisation exceptionnelle. Dans notre pipeline, SAM est couplé à Fast R-CNN pour tirer parti des détections supervisées, puis utilisé en aval pour produire des masques précis sur des objets même partiellement visibles ou absents des jeux de données annotés. Ce couplage permet une segmentation plus fine et plus flexible, tout en maintenant une robustesse opérationnelle adaptée aux contraintes des scènes capturées en réalité mixte.

Chacun de ces modèles a été sélectionné pour répondre à une exigence particulière du pipeline : l'efficacité pour l'acquisition embarquée (YOLO), la précision pour l'apprentissage supervisé (Mask R-CNN), et la flexibilité dans les scénarios ouverts ou faibles en annotations (SAM). Ce choix multiple permet non seulement une comparaison expérimentale approfondie, mais aussi une évaluation robuste des performances selon les contraintes de l'application ciblée.

3.4.3. ENTRAÎNEMENT ET FINE-TUNING

L'étape d'entraînement constitue une phase cruciale de notre pipeline de segmentation. Elle vise à adapter les modèles de segmentation pré-entraînés aux spécificités de notre jeu de données expérimental. Pour cela, nous avons adopté une stratégie de fine-tuning supervisé exploitant les poids pré-appris sur COCO, tout en ajustant les hyperparamètres, la structure des données et les méthodes d'optimisation pour maximiser les performances sur notre corpus d'objets en environnement de laboratoire. Dans cette section, nous décrivons en détail le processus d'apprentissage pour chacun des algorithmes sélectionnés : YOLO, Mask R-CNN et SAM.

3.4.3.1. YOLO

L'entraînement du modèle YOLOv8-SEG s'inscrit dans une démarche de transfert d'apprentissage, en tirant parti des poids pré-entraînés du modèle yolov8x-seg.pt, distribués par Ultralytics (Ultralytics, 2023). Cette stratégie permet d'accélérer la convergence tout en adaptant efficacement le réseau aux spécificités de notre jeu de données personnalisé, sans compromettre la qualité des représentations déjà acquises à partir de larges corpus comme COCO (Lin et al., 2014).

L'ensemble du pipeline repose sur la bibliothèque Ultralytics v8.0.0, qui offre une API unifiée pour les tâches de détection et de segmentation d'instances. Le jeu de données est structuré selon le format requis, avec un fichier YAML décrivant les chemins vers les images et les masques, les 20 classes annotées, ainsi que leur mappage sémantique. Les masques d'objets sont encodés sous forme de polygones, comme requis par le format interne de YOLO pour les tâches de segmentation (Jocher et al., 2023).

Les images sont redimensionnées à 640 × 640 pixels, résolution standard pour l'architecture YOLO, puis normalisées. Un pipeline d'augmentation de données est

appliqué dynamiquement pendant l'entraînement, incluant des transformations géométriques et photométriques : flips horizontaux, jitter spatial, variations de contraste/luminosité, et distorsions de perspective. Ces augmentations visent à accroître la diversité des exemples d'entraînement, réduisant le surapprentissage et améliorant la généralisation.

Différentes configurations des hyperparamètres ont été testées : 50, 100, puis 150 époques avec des tailles de lot de 8, 12 et 16. L'optimisation repose sur la méthode Stochastic Gradient Descent (SGD), avec un momentum de 0.937, et un scheduler à décroissance co-sinusoïdale pour une adaptation douce du taux d'apprentissage (Loshchilov & Hutter, 2016). Par ailleurs, l'entraînement en précision mixte (AMP) a été activé pour bénéficier d'une vitesse accrue et d'une consommation mémoire réduite sur GPU, conformément aux pratiques actuelles dans l'apprentissage profond (Micikevicius et al., 2018).

La validation du modèle est effectuée automatiquement à la fin de chaque époque à l'aide des métriques intégrées dans Ultralytics, incluant la Mean Average Precision (mAP) calculée sur les objets détectés et segmentés. Les résultats de l'entraînement sont sauvegardés sous forme de fichiers CSV, facilitant le suivi temporel des performances. Les modèles les plus performants sont enregistrés, et les prédictions sont exportées dans un format aligné avec notre pipeline de visualisation. Chaque prédiction comprend la classe détectée, la boîte englobante, le score de confiance et le polygone du masque, le tout stocké dans des fichiers .txt et .png exploitables dans les étapes aval de la reconstruction.

Grâce à cette phase de fine-tuning, le modèle YOLO s'avère particulièrement adapté pour la segmentation d'objets en temps réel dans un environnement de laboratoire. Sa rapidité d'inférence et sa capacité à fonctionner sur des systèmes embarqués en font un élément clé de notre pipeline de reconstruction 3D en réalité mixte.

3.4.3.2. MASK R-CNN

Pour compléter notre étude comparative des approches de segmentation d'objets, nous avons mis en œuvre une stratégie de fine-tuning sur le modèle Mask R-CNN, une architecture de référence en segmentation d'instances introduite par He et al., (2017). Ce modèle repose sur un pipeline en deux étapes, combinant un détecteur de régions (Region Proposal Network) et des têtes de prédiction spécialisées pour la classification, la régression de boîte englobante, et la génération de masques de segmentation à l'échelle des pixels. Ce design modulaire, extensible et précis s'est imposé comme l'un des plus performants dans la littérature, notamment sur les benchmarks COCO.

Dans notre implémentation, nous avons utilisé la version pré-entraînée disponible dans la bibliothèque Torchvision (maskrcnn_resnet50_fpn), qui intègre un backbone ResNet-50 avec un Feature Pyramid Network (FPN) pour la détection multi-échelle (Lin et al., 2017). Cette combinaison permet d'extraire efficacement des caractéristiques discriminantes sur plusieurs résolutions, ce qui est particulièrement utile pour détecter des objets de tailles variables dans des scènes complexes. Les couches de prédiction (classificateur, régressions de boîte et générateur de masques) ont été réinitialisées pour correspondre à notre jeu de données personnalisé à 20 classes.

La préparation des données a suivi le format COCO JSON, généré à partir d'annotations polygones avec LabelMe. Chaque image est associée à des masques binaires d'instances, à des boîtes englobantes et à des identifiants de classes. L'API pycocotools est utilisée pour manipuler ces annotations efficacement, tandis qu'un cache mémoire local optimise la lecture des fichiers lors de l'entraînement. Les images d'entrée sont standardisées à une résolution de 640×640 pixels, et un prétraitement est appliqué via des transformateurs, notamment la conversion en tenseurs et la normalisation implicite.

L'entraînement est effectué avec précision mixte pour maximiser l'efficacité GPU et minimiser la consommation mémoire (Micikevicius et al., 2018). Un optimiseur SGD avec momentum est employé, accompagné d'un scheduler à paliers avec warm-up initial.

La boucle de validation évalue les performances sur l'ensemble de validation via les métriques standards COCO — pour les boîtes et les masques. Ces mesures sont obtenues avec l'aide d'un évaluateur personnalisé dérivé du module Coco Evaluator, assurant la compatibilité avec les formats d'annotations enrichis. Les résultats de l'entraînement sont sauvegardés sous forme de fichiers CSV, facilitant le suivi temporel des performances. Les modèles les plus performants sont enregistrés, et les prédictions sont exportées dans un format aligné avec notre pipeline de visualisation. Chaque prédiction comprend la classe détectée, la boîte englobante, le score de confiance et le polygone du masque, le tout stocké dans des fichiers .txt et .png exploitables dans les étapes aval de la reconstruction.

3.4.3.3. MODÈLE HYBRIDE SAM – FASTER RCNN

Dans une volonté d'exploiter les forces respectives de la détection d'objets et de la segmentation universelle, nous avons conçu un pipeline hybride combinant le modèle Faster R-CNN (Ren et al., 2015) pour la détection et la classification d'objets, avec le Segment Anything Model (SAM) (Kirillov et al., 2023) pour la génération des masques. Cette architecture, désignée sous le nom de SAM-RCNN, permet d'obtenir des masques précis et cohérents, y compris pour des objets de formes complexes, de petites tailles ou partiellement occultés.

Le pipeline s'articule en deux étapes : d'abord, Faster R-CNN identifie les objets présents dans les images et produit pour chacun une boîte englobante, une classe prédite et un score de confiance. Ces boîtes sont ensuite transmises à SAM, qui génère un masque de segmentation correspondant à chaque région détectée. SAM, en tant que modèle de

fondation, peut segmenter n'importe quelle région d'intérêt à partir d'une requête simple comme une boîte ou un point (Kirillov et al., 2023).

Durant l'entraînement, les masques produits par SAM sont supervisés à l'aide des vérités terrain (masks GT), via une perte de type Binary Cross-Entropy. Cette supervision indirecte améliore la spécialisation du modèle sans modifier directement ses poids internes. La composante Faster R-CNN est fine-tunée à partir de poids COCO (Lin et al., 2014), avec adaptation au nombre de classes annotées localement. L'optimisation est conjointe sur les pertes de classification, de localisation et de segmentation. Ce pipeline illustre l'intérêt des architectures hybrides en combinant robustesse supervisée et flexibilité, dans un cadre adapté aux applications de réalité mixte nécessitant précision et généricité.

3.4.4. MÉTRIQUES D'ÉVALUATION ET PROTOCOLE EXPÉRIMENTAL

L'évaluation des modèles de segmentation d'objets repose sur un protocole rigoureux articulé autour des métriques issues du benchmark Common Objects in Context (COCO), devenu un standard de facto dans la communauté de la vision par ordinateur (Lin et al., 2014). Pour chaque architecture entraînée dans ce travail — à savoir YOLO, Mask R-CNN, et le modèle hybride SAM-RCNN — nous avons systématiquement consigné, à chaque époque, un ensemble de métriques permettant de suivre à la fois l'évolution de l'apprentissage et la qualité des prédictions.

Les métriques collectées après chaque époque incluent des pertes caractéristiques de l'apprentissage supervisé : la perte de régression des boîtes englobantes, la perte de segmentation, la perte de classification, ainsi qu'une mesure spécifique aux architectures de type YOLO, la *Distribution Focal Loss* (Jocher et al., 2023). Ces indicateurs permettent d'évaluer à la fois la localisation, la segmentation fine et la catégorisation des objets dans les images RGB annotées.

Pour quantifier la qualité des prédictions, nous avons adopté les métriques COCO. Les performances sont exprimées en termes de précision, rappel, et moyenne des précisions (mean Average Precision, mAP), calculées séparément pour les boîtes englobantes et pour les masques. Ces scores sont extraits via l'outil CocoEvaluator, qui permet une évaluation automatisée, reproductible et conforme aux standards de la littérature (Massa & Girshick, 2018).

En parallèle de ces évaluations numériques, les prédictions de segmentation générées pendant l'inférence sont exportées sous forme de fichiers texte. Chaque ligne encode une instance détectée, incluant l'identifiant de la classe, le score de confiance, les coordonnées normalisées de la boîte englobante, ainsi qu'un polygone décrivant avec précision le contour du masque prédit. Ce format, dérivé des pratiques adoptées par Ultralytics (2023), permet une compatibilité directe avec nos modules de post-traitement, notamment pour la reconstruction 3D.

Enfin, ce protocole est commun à tous les modèles considérés, assurant une comparabilité équitable des résultats. Il permet non seulement de diagnostiquer les performances globales, mais aussi de suivre finement la convergence des modèles et leur capacité à généraliser sur des données complexes, annotées en conditions réelles dans notre laboratoire.

3.5. RECONSTRUCTION 3D

La reconstruction tridimensionnelle constitue l'étape finale de notre pipeline, visant à transformer les objets segmentés à partir des images RGB-D en représentations surfaciques exploitables dans un environnement de réalité mixte. Pour ce faire, nous avons opté pour une approche de modélisation implicite, fondée sur un réseau de neurones de type Multi-Layer Perceptron (MLP), nommé *OccupancyNet.*(Mescheder et al., 2019)

Contrairement aux méthodes classiques reposant sur la triangulation directe de nuages de points, cette stratégie apprend une fonction continue d'occupation définie dans l'espace 3D, permettant de reconstruire des surfaces fines, cohérentes et topologiquement stables à l'aide de l'algorithme Marching Cubes (Lorensen et Cline, 1987). La méthode repose sur une supervision indirecte à partir de masques segmentés projetés dans l'espace caméra, et intègre l'ensemble des intrinsics pour garantir une reconstruction géométriquement fidèle.

3.5.1. PRÉSENTATION DE LA MÉTHODOLOGIE

Le cœur de la méthodologie repose sur l'apprentissage d'une fonction d'occupation implicite $f_\Theta: R^3 \rightarrow [0,1]$, modélisée par un réseau de neurones multi-couches (MLP). Cette fonction apprend à prédire, pour tout point de l'espace 3D, la probabilité qu'il soit à l'intérieur de la surface d'un objet donné. La surface de l'objet correspond alors à l'ensemble des points $x \in R^3$ pour lesquels $f_\Theta(x) = 0.5$, ce qui forme un champ scalaire de type niveau iso-surface. Ce paradigme de reconstruction offre une formulation continue, sans maillage explicite ni connectivité initiale, et présente une meilleure régularité topologique par rapport aux techniques par voxelisation ou par nuages de points discrets.

Dans la première étape, chaque instance d'objet est préalablement détectée et segmentée par l'un des modèles étudiés (YOLO, Mask R-CNN ou SAM-RCNN). Les masques obtenus sont représentés sous forme de polygones normalisés (coordonnées [0,1]) puis convertis en masques binaires ayant la même résolution que l'image d'origine.

Une fois ces masques obtenus, la méthode procède à une re-projection en 3D. Pour chaque pixel appartenant à un masque binaire, les coordonnées image (u, v) sont associées à une valeur de profondeur extraite de la carte de profondeur synchronisée. Grâce aux paramètres intrinsèques de la caméra RGB-D (focale f_x, f_y et centre optique c_x, c_y), chaque

point image est transformé en une coordonnée spatiale (X, Y, Z) dans le repère caméra, selon l'équation 3.1 qui suit:

$$X = \frac{(u - c_x) \cdot Z}{f_x}, \quad Y = \frac{(v - c_y) \cdot Z}{f_y}, \quad Z = profondeur$$

Equation 3.1: Transformation des points image en coordonnée spatiale (X, Y, Z)

où (u, v) sont les coordonnées pixels, Z est la valeur de profondeur à ce pixel, et (X, Y, Z) le point 3D correspondant dans l'espace caméra.

Ces points 3D forment un nuage de points partiel de l'objet observé, représentant les zones visibles depuis la caméra. Ce nuage est ensuite centré et normalisé dans un cube $[-1,\ 1]^3$ afin de faciliter l'apprentissage du modèle. À partir de ce volume normalisé, deux types d'échantillons sont générés : des échantillons positifs, issus des points projetés (surface observée), et des échantillons négatifs, tirés aléatoirement dans le volume englobant, considérés comme hors-surface. Ces échantillons sont labellisés respectivement 1 et 0, formant un ensemble d'apprentissage équilibré.

Un réseau de neurones fully-connected (MLP) est ensuite entraîné sur cet ensemble de données, en minimisant une fonction de perte binaire croisée (Binary Cross Entropy) entre les prédictions du modèle et les valeurs attendues. Une fois le réseau entraîné, la fonction f_Θ est évaluée sur une grille régulière 3D de résolution N^3 (typiquement 128^3, 256^3). Cette grille permet de générer un champ de probabilité d'occupation. L'algorithme Marching Cubes est alors appliqué pour extraire l'iso-valeur 0.5, produisant un maillage triangulaire représentant la surface reconstruite. Le maillage est ensuite re-transformé à l'échelle originale par re-projection inverse.

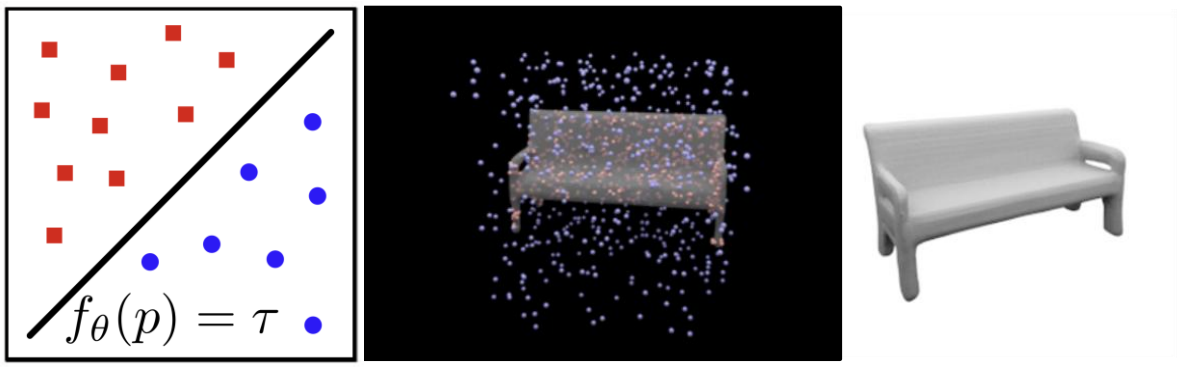

Figure 3.4 – Illustration du principe de reconstruction 3D
© (Mescheder et al., 2019)

3.5.2. APPRENTISSAGE DE LA FONCTION D'OCCUPATION

L'apprentissage de la fonction d'occupation constitue une étape centrale du processus de reconstruction. Après avoir projeté les pixels masqués dans l'espace 3D et normalisé le nuage de points, on génère un jeu de données supervisé permettant d'enseigner à un réseau de neurones comment différencier l'intérieur d'un objet de l'extérieur. Chaque point du nuage projeté, issu directement des données RGB-D et des masques segmentés, est considéré comme un exemple positif : il appartient à la surface visible de l'objet. Ces points sont ensuite centrés autour de l'origine du repère 3D et mis à l'échelle pour tenir dans un cube normalisé de coordonnées$[-1,\ 1]^3$. Cette normalisation est indispensable pour assurer la stabilité numérique de l'entraînement.

En parallèle, un ensemble d'exemples négatifs est généré par échantillonnage uniforme aléatoire dans ce même espace 3D. Ces points ne proviennent pas de la surface observée, et sont donc supposés situés hors de l'objet. Chaque exemple est associé à une étiquette binaire : 1 pour les points positifs (à l'intérieur de l'objet), 0 pour les négatifs (hors de l'objet). L'ensemble formé constitue une base d'apprentissage équilibrée.

Le réseau de neurones utilisé est un MLP à 4 couches cachées, avec des activations ReLU et une sortie sigmoïde. Son objectif est de modéliser une fonction continue $f_\theta: R^3 \rightarrow [0,1]$, où $f(x) \simeq 1$ indique une forte probabilité que le point x soit à l'intérieur de l'objet. L'entraînement est réalisé à l'aide de l'algorithme Adam et d'une fonction de perte binaire croisée (Binary Cross Entropy). Cette stratégie s'inspire directement des travaux de Mescheder et al. (2019) sur les Occupancy Networks, qui ont montré qu'une telle approche permet de générer des surfaces régulières à partir de peu de données observées.

3.5.3. EXTRACTION DU MAILLAGE PAR MARCHING CUBES

Une fois le réseau entraîné à approximer la fonction d'occupation, celui-ci est utilisé pour évaluer $f(x)$ sur une grille dense et régulière de points couvrant l'espace 3D normalisé. Cette étape transforme la fonction apprise en champ scalaire discret, où chaque cellule de la grille contient une valeur de probabilité.

L'extraction de la surface s'effectue alors à l'aide de l'algorithme Marching Cubes, une méthode classique introduite par Lorensen et Cline (1987), qui permet de convertir un champ scalaire 3D en maillage surfacique triangulaire. Plus précisément, l'algorithme identifie l'iso-surface associée à la valeur seuil de 0.5, correspondant à la frontière prédite entre l'intérieur et l'extérieur de l'objet.

3.6. CONCLUSION

Le maillage généré est ensuite re-projeté dans le repère original, à l'aide des informations de centrage et d'échelle utilisées pendant la normalisation. On obtient ainsi une reconstruction 3D fidèle à l'échelle réelle, exploitable pour la visualisation, la simulation ou l'intégration en réalité mixte. Ce processus permet de reconstruire des formes douces, continues et adaptatives, sans bruit ni artefacts liés à la voxelisation ou à la triangulation

directe. Il s'avère particulièrement efficace pour représenter des surfaces complexes à partir de données partielles, comme celles captées par une caméra RGB-D en conditions réelles.

Ce chapitre a présenté en détail les différentes composantes de notre pipeline hybride de segmentation et de reconstruction 3D en réalité mixte. À travers la description des environnements matériels et logiciels, de l'architecture de l'application HoloLens, des modèles de segmentation implémentés, et de la méthode de reconstruction par fonction d'occupation, nous avons documenté chaque étape de manière structurée et technique. L'ensemble de cette chaîne constitue une solution intégrée, depuis la capture RGB-D jusqu'à la génération de maillages exploitables en environnement immersif. Cette documentation technique pose les bases nécessaires à l'analyse des résultats, qui sera abordée dans le chapitre suivant, à travers une évaluation quantitative, qualitative et illustrative des performances de notre système.

CHAPITRE IV

ANALYSE DES RÉSULTATS

L'objectif de ce chapitre est de présenter une analyse approfondie des performances obtenues par notre pipeline, en s'appuyant sur des indicateurs quantitatifs, des visualisations qualitatives et des cas d'usage illustratifs. Après avoir mis en place une architecture technique combinant acquisition HoloLens, segmentation multi-modèles et reconstruction 3D, il est désormais essentiel d'évaluer concrètement l'efficacité des différentes composantes du système.

Nous commencerons par une évaluation des performances globales de l'application HoloLens, en mesurant les temps de capture, de transmission et de traitement associés au pipeline. Cette analyse permet d'établir la réactivité du système en conditions réelles. Nous poursuivrons ensuite par une comparaison approfondie des méthodes de segmentation implémentées (YOLO, Mask R-CNN, SAM-Faster R-CNN), en examinant leurs performances sur notre jeu de données expérimental. Cette deuxième étape met l'accent sur la qualité des masques générés, un facteur déterminant pour la précision des reconstructions 3D à venir.

4.1. ÉVALUATION DES PERFORMANCES DE L'APPLICATION HOLOLENS

Il était essentiel d'évaluer les performances concrètes de l'application HoloLens, notamment en termes de réactivité et de fluidité de traitement. Cette section propose une analyse du temps requis pour passer de la capture d'une image sur le HoloLens 2 à

l'obtention du résultat de segmentation sur le serveur distant, incluant l'ensemble des étapes intermédiaires.

Les mesures ont été réalisées dans un réseau local, avec un serveur exécuté sur un ordinateur portable équipé d'un processeur Intel Core Ultra 9, d'une carte graphique RTX 4060 et de 32 Go de RAM. À chaque itération, l'application HoloLens capture une image RGB et sa carte de profondeur, qu'elle envoie immédiatement au serveur via une requête HTTP. Ce dernier effectue alors le traitement de segmentation (selon le modèle choisi) et renvoie les résultats au format JSON. En moyenne, la capture par le HoloLens prend environ 95 millisecondes. L'envoi des données vers le serveur ajoute environ 160 millisecondes, tandis que le prétraitement (décodage et préparation des images) ne dépasse pas 50 millisecondes. Le temps d'inférence et de génération de meshe varie fortement selon le modèle utilisé : environ 240 ms pour YOLO, 420 ms pour Mask R-CNN et jusqu'à 510 ms pour le modèle SAM-RCNN. Cela représente un temps total cumulé de 600 à 800 ms pour un cycle complet de capture-traitement-réponse.

Ces résultats sont encourageants pour une utilisation dans des scénarios semi-temps réel, où une latence inférieure à une seconde reste tolérable. Toutefois, ils soulignent également des axes d'optimisation potentiels, notamment la compression des images, l'utilisation de modèles plus légers, ou encore le déploiement de solutions en edge computing pour éviter les transferts réseau.

En somme, l'application HoloLens présente des performances globalement satisfaisantes, avec une latence maîtrisée qui permet une expérience fluide dans un cadre expérimental ou exploratoire.

4.2. COMPARAISON DES PERFORMANCES DES MÉTHODES DE SEGMENTATION

Dans cette section, nous comparons les trois modèles de segmentation testés dans le cadre de ce projet : YOLO, Mask R-CNN et SAM couplé à Faster R-CNN. Chaque architecture a été fine-tunée sur notre jeu de données personnalisé acquis via l'application HoloLens 2, puis évaluée selon des métriques standardisées du benchmark COCO. L'évaluation a été réalisée dans un environnement contrôlé, avec une séparation stricte entre les ensembles d'entraînement, de validation et de test, assurant ainsi la robustesse et la reproductibilité des comparaisons. Les résultats numériques ont été collectés automatiquement à la fin de chaque époque d'entraînement, puis analysés globalement selon les performances moyennes observées sur l'ensemble des images du jeu de test.

4.2.1. ANALYSE DES PERFORMANCES ET DES RÉSULTATS D'ENTRAÎNEMENT

Cette première analyse se concentre sur l'évolution des performances des modèles tout au long de l'entraînement. Pour chaque architecture, nous avons suivi des métriques clés à chaque époque, telles que les pertes de classification, de localisation et de segmentation, ainsi que les indicateurs de performance sur le jeu de validation. Cette analyse permet d'évaluer la stabilité de l'apprentissage, la convergence des modèles, et leur capacité à généraliser sur des données non vues.

4.2.1.1. YOLO

Dans le cadre de ce projet, nous avons utilisé le modèle yolov8x-seg.pt, pré-entraîné sur le dataset COCO, puis fine-tuné sur notre jeu de données personnalisé constitué à partir d'images capturées via l'application HoloLens 2. L'entraînement s'est déroulé sur un total

de 100 époques, avec une taille de lot (batch size) de 8 images et une résolution d'entrée fixée à 640×640 pixels. L'optimisation a été réalisée avec l'algorithme SGD (momentum 0.937) et un scheduler de type cosinus. Nous avons activé l'entraînement en précision mixte pour tirer parti des capacités de la carte GPU RTX 4060 tout en réduisant l'utilisation mémoire.

Le suivi des performances pendant l'entraînement repose sur les pertes suivantes : le box_loss qui constitue l'erreur (coûts)de localisation des boîtes englobantes, les coûts de classification des objets cls_loss, l'erreur seg_loss sur les masques de segmentation ainsi que dfl_loss, la Distribution Focal Loss, spécifique à l'optimisation des prédictions de boîtes. Le graphique 4.1 qui suit met en évidence l'évolution des coûts au cours de l'entraînement. En effet on constate que box_loss, seg_loss et cls loss décroissant considérable respectivement de $\simeq 0{,}59$, $\simeq 1{,}28$, $\simeq 1{,}64$ à $\simeq 0,2,34$, $\simeq 0{,}17$.

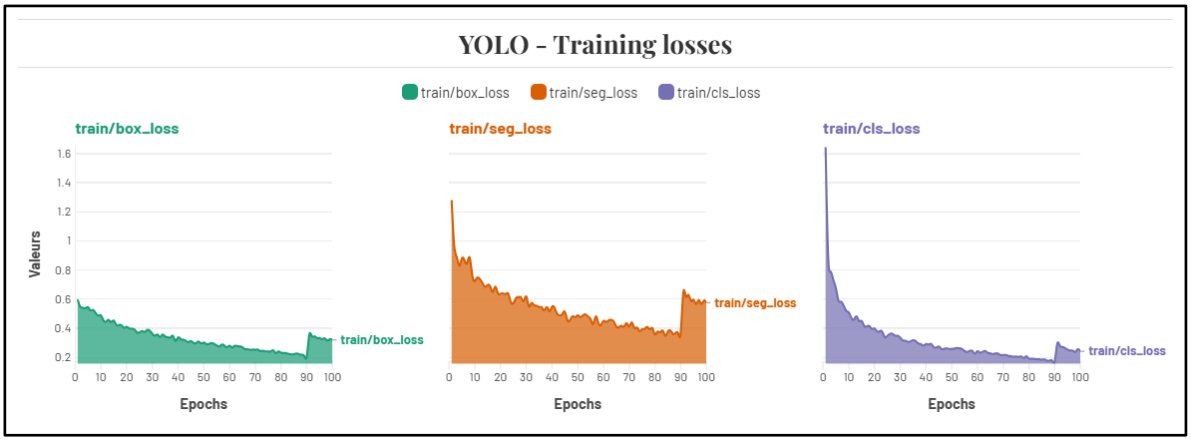

Figure 4.1 — Évolution des coûts lors de l'entraînement du modèle YOLO.

Les résultats mesurés sur le jeu de validation montrent une évolution positive autant sur la prédiction des boîtes englobantes que sur celle des masques comme le montre les figures 4.2 et 4.3 suivantes.

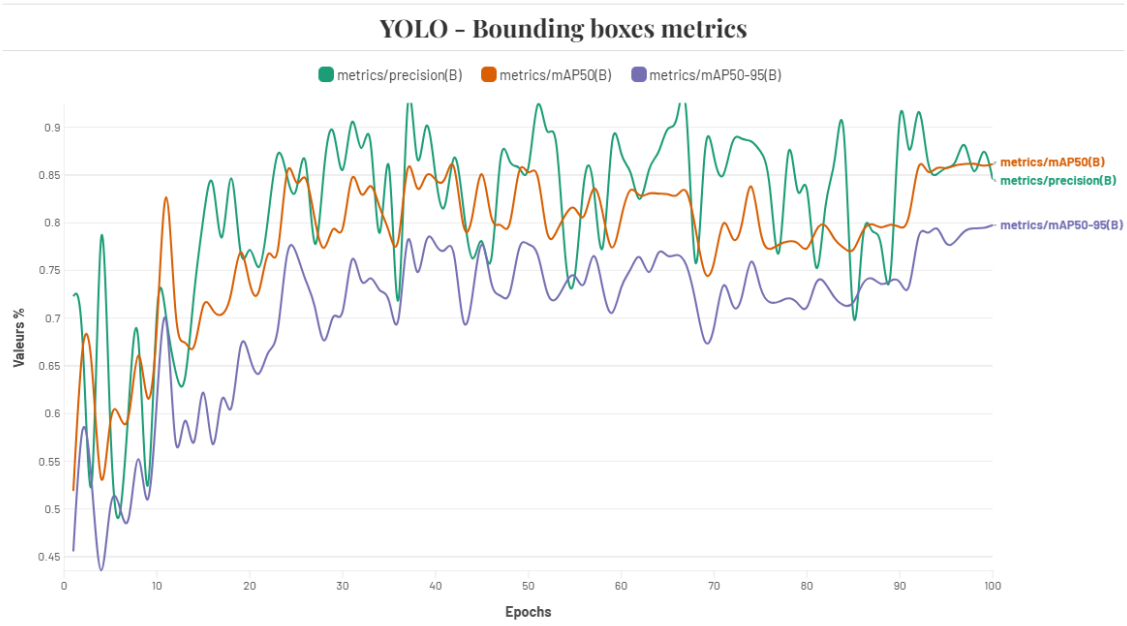

Figure 4.2 —Évolution des métriques relative au boites englobantes — YOLO.

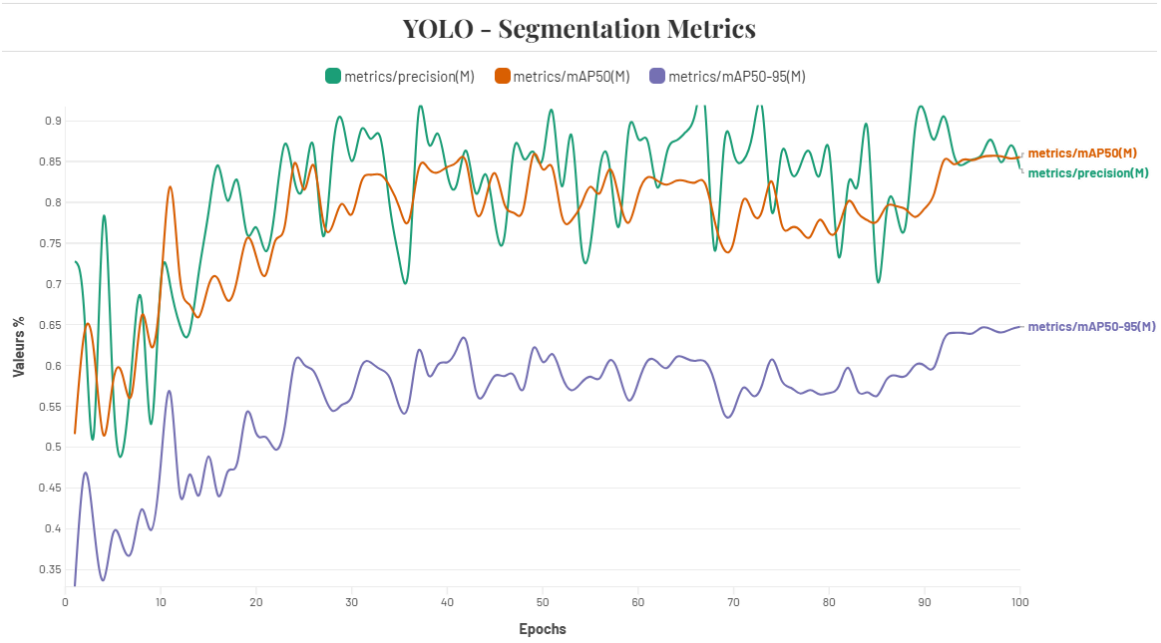

Figure 4.3 — Évolution des métriques relatives aux masques — YOLO.

Plutôt que de retenir le modèle à la dernière époque, nous avons sélectionné celui ayant atteint les meilleures performances en segmentation, soit à l'époque 67. À ce stade, YOLO affiche une précision de **91,66 %** et un rappel de **78,15 %**, traduisant une excellente

qualité de prédiction avec peu de faux positifs, mais une certaine difficulté à détecter les objets les plus petits ou partiellement occultés.

Les scores moyens atteignent **82,32 %** pour le mAP50 et **60,49 %** pour le mAP50-95, confirmant une robustesse satisfaisante sur l'ensemble des seuils IoU. Ces résultats illustrent la stabilité et la précision de YOLO dans un contexte contrôlé, en particulier pour des objets bien définis. Le modèle de l'époque 67 a ainsi été retenu pour la génération des masques en vue de la reconstruction 3D.

4.2.1.2. MASK R-CNN

Le second modèle évalué dans ce projet est Mask R-CNN, dans sa version maskrcnn_resnet50_fpn pré-entraînée sur le dataset COCO. Ce modèle, reposant sur une architecture à deux étapes, combine la détection d'objets via Faster R-CNN avec une branche de segmentation dédiée à l'extraction des masques binaires. Pour l'adapter à notre jeu de données personnalisé issu de l'application HoloLens, nous avons procédé à un fine-tuning supervisé sur 100 époques, avec des hyper-paramètres constants (batch size de 8, résolution 640×640, optimiseur Adam).

Durant l'entraînement, l'évolution des pertes d'apprentissage — box_loss, seg_loss, et cls_loss — a montré une convergence progressive et stable. Comme le montre la figure 4.4, les pertes diminuent régulièrement, traduisant une bonne adaptation du modèle à notre jeu de données annoté manuellement.

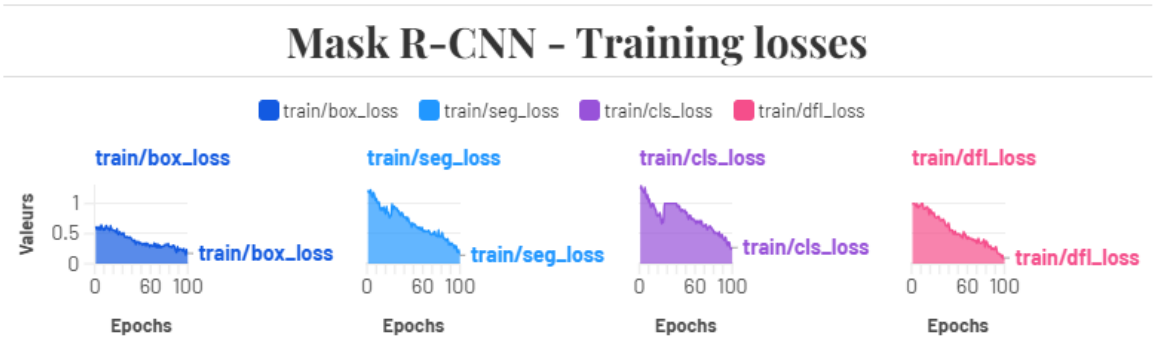

Figure. 4.4 — Évolution des pertes d'apprentissage – Mask R-CNN.

Du point de vue des performances de segmentation, le modèle a atteint son meilleur niveau à l'époque 89. À ce stade, la précision sur les masques s'élevait à 80,28 %, indiquant qu'une large majorité des masques prédits correspondaient correctement aux objets cibles. Le rappel atteignait 76,14 %, témoignant d'une bonne capacité du modèle à retrouver la majorité des objets annotés, y compris ceux partiellement visibles ou de forme complexe. Le mAP50 culminait à 82,79 %, tandis que le mAP50-95, plus exigeant en termes de localisation et de recouvrement, atteignait 51,30 %, comme l'illustrent les figures 4.5 et 4.6.

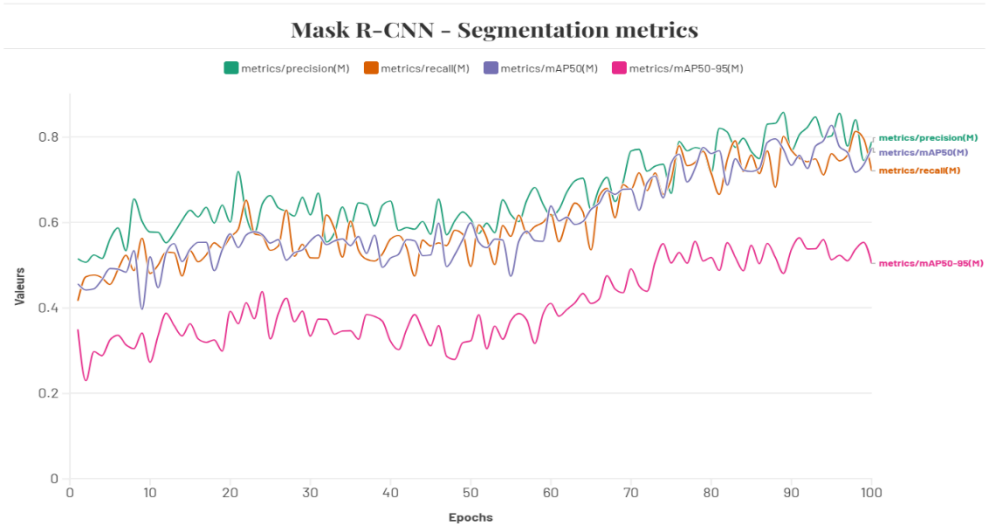

Figure 4.5 — Évolution des métriques de performance sur les masques – Mask R-CNN.

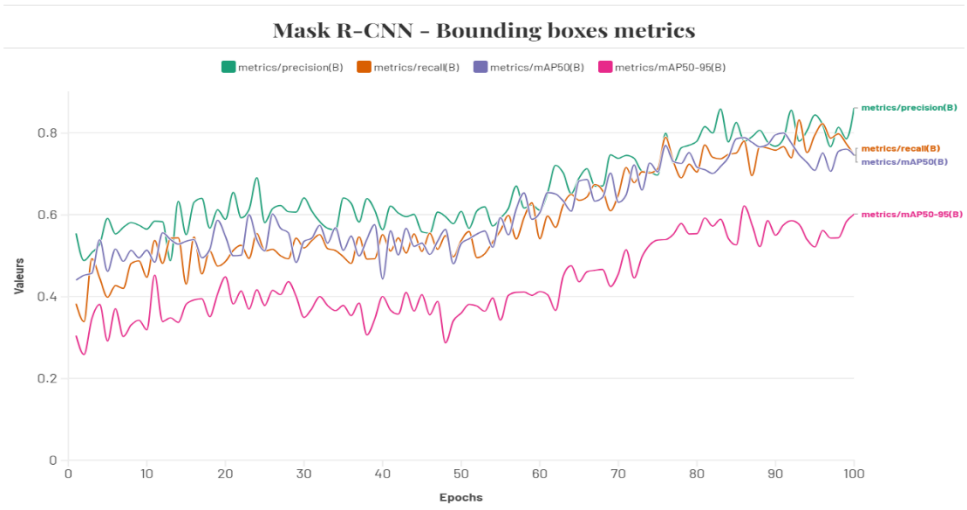

Figure 4.6 — Évolution des métriques de performance sur les boîtes – Mask R-CNN.

Ces résultats, bien que plus modestes que ceux obtenus avec YOLO, démontrent la robustesse du modèle Mask R-CNN sur des objets bien délimités, mais révèlent également ses limites en contexte de faible contraste ou d'objets partiellement visibles. L'état du modèle à l'époque 89 a été retenu pour les inférences ultérieures et la reconstruction 3D.

4.2.1.3. MODÈLE HYBRIDE SAM + FASTER R-CNN

Le troisième modèle exploré dans le cadre de ce projet adopte une architecture hybride combinant un détecteur d'objets classique — Faster R-CNN (Ren et al., 2015) — avec le modèle de fondation de segmentation Segment Anything Model (SAM) Cette combinaison, désignée sous le nom de SAM-RCNN, exploite les boîtes englobantes générées par Faster R-CNN pour guider SAM dans la production de masques précis et adaptatifs.

L'entraînement du pipeline a été mené en deux temps : d'abord le fine-tuning de Faster R-CNN à partir de poids pré-entraînés sur COCO, adapté à notre nombre de classes,

puis l'intégration de SAM dans une boucle de supervision indirecte. Les masques prédits par SAM à partir des boîtes sont évalués via une perte de type Binary Cross-Entropy par rapport aux masques ground truth du dataset, sans ajustement direct des poids internes de SAM. Le suivi de l'apprentissage a porté sur plusieurs pertes caractéristiques du pipeline : la perte de détection (box_loss), la perte de classification (cls_loss), ainsi que la perte de segmentation (seg_loss) supervisée indirectement via SAM. Comme le montre la figure 4.7, ces coûts décroissent de manière régulière tout au long de l'entraînement.

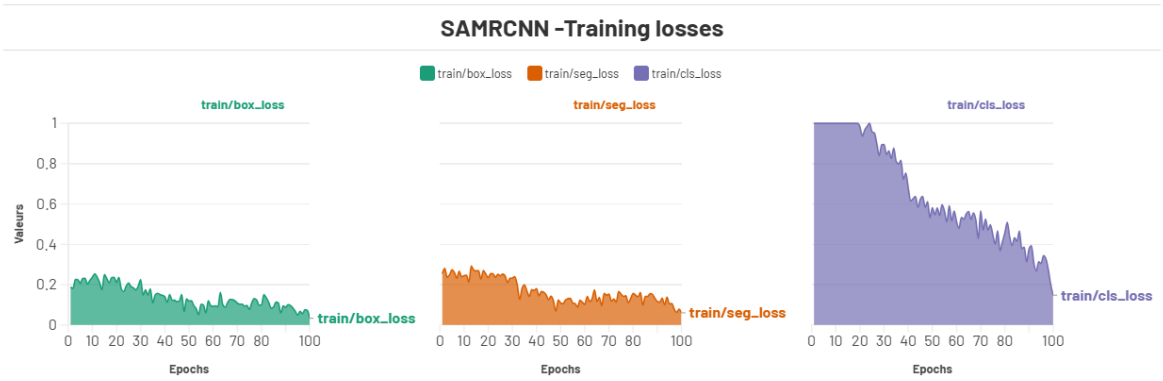

Figure 4.7 — Évolution des pertes – SAM + Faster R-CNN.

La meilleure performance a été atteinte à l'époque 93, où la précision sur les masques s'élève à 86,31 %, et le rappel à 83,13 %. Le score mAP50 atteint 84,51 %, tandis que le mAP50-95, indicateur plus exigeant, enregistre 55,65 %, comme l'illustrent les figures 4.8 et 4.9.

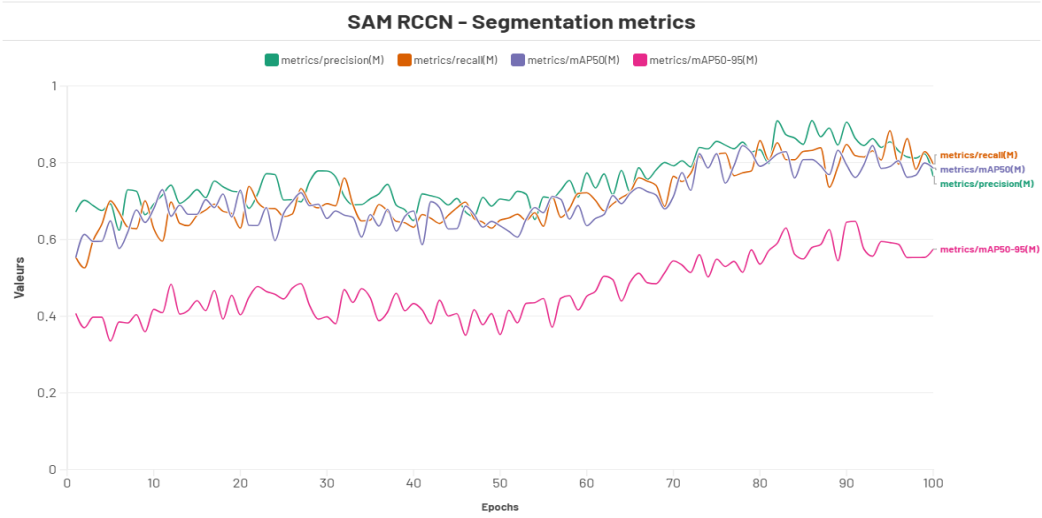

Figure 4.8 — Évolution des métriques sur les masques – SAM + Faster R-CNN.

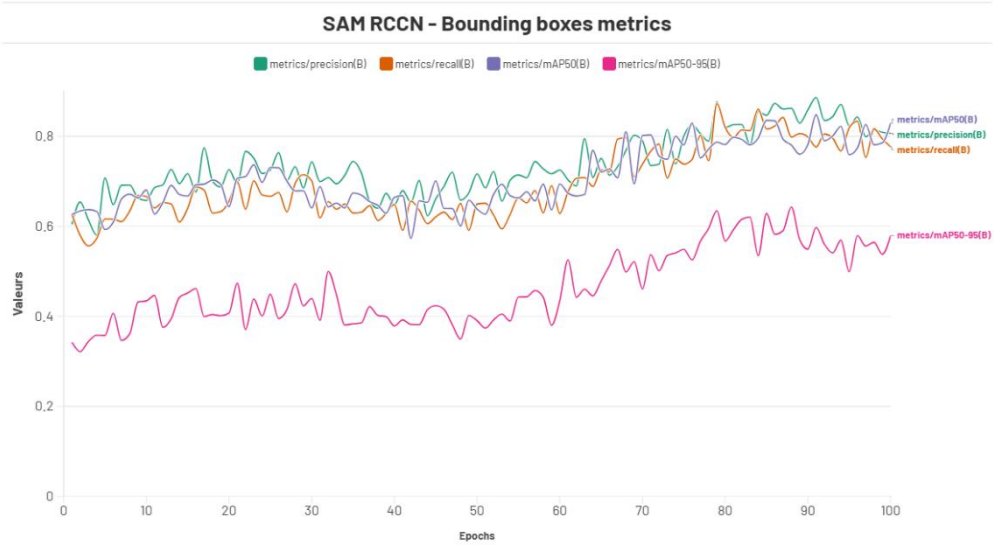

Figure 4.9 — Évolution des métriques sur les boîtes – SAM + Faster R-CNN.

Ces résultats mettent en évidence la capacité du modèle hybride à produire des masques de haute qualité, surtout pour des objets complexes ou partiellement visibles, en tirant parti de la généralisation de SAM et de la robustesse de Faster R-CNN. Le modèle sauvegardé à l'époque 93 été retenu pour les inférences et la reconstruction 3D.

4.2.2. COMPARAISON GLOBALE DES PERFORMANCES

Le tableau 4.1 ci-dessous synthétise les performances des trois modèles de segmentation implémentés dans le cadre de ce projet : YOLOv8-SEG, Mask R-CNN et le modèle hybride SAM-Faster R-CNN. Cette comparaison repose sur des indicateurs issus du benchmark COCO, mais intègre également des considérations pratiques telles que le temps d'inférence moyen par image et la taille mémoire du modèle.

Tableau 4.1 – Résultats comparés des modèles de segmentation

Modèle	mAP 50	mAP [50-95]	Précision	Rappel	Temps d'inférence (ms/image)	Epoch optimal	Nombre de paramètres
YOLO	82.32 %	60.49 %	91.66 %	78.15%	27.6	67	45,951,468
Mask-RCNN	82.79 %	51.30 %	80.28%	76.14 %	89.2	89	~80,000,000
SAM-RCNN	84.51%	55.65%	86.31	83.13%	142.6	93	47.743,432

Cette vue d'ensemble met en évidence les forces respectives de chaque modèle. YOLO se distingue par un excellent compromis entre rapidité d'exécution et précision, ce qui en fait une solution efficace pour des applications temps réel. Mask R-CNN, bien qu'un peu moins performant en mAP50-95, présente une bonne régularité, notamment pour des objets bien délimités. Enfin, le modèle hybride SAM-RCNN affiche les meilleurs scores de rappel et de précision moyenne, au prix d'un coût de calcul plus élevé, dû à la complexité du modèle SAM.

Ces résultats confirment l'intérêt d'une approche hybride lorsque la qualité des masques est primordiale, notamment pour des objets complexes ou partiellement visibles, mais soulignent aussi la pertinence de modèles plus légers comme YOLO pour des systèmes embarqués ou interactifs.

4.3. ÉVALUATION QUANTITATIVE ET QUALITATIVE DES MAILLAGES

La qualité des maillages générés constitue un indicateur essentiel de l'efficacité de notre pipeline de reconstruction. Dans cette section, nous proposons une double analyse à la fois **quantitative**, à l'aide de mesures objectives extraites des surfaces reconstruites, et **qualitative**, par inspection visuelle et comparaison croisée avec les objets d'origine.

D'un point de vue quantitatif, plusieurs métriques ont été utilisées pour caractériser les maillages produits :

- **Nombre de faces triangulaires** : reflète le niveau de détail du maillage.
- **Étanchéité (watertightness)** : indique si la surface est fermée, sans trous.
- **Volume reconstruit (optionnel)** : permet de détecter des artefacts.
- **Chamfer Distance** : évalue la précision géométrique quand un maillage de référence est disponible.

En parallèle, une **évaluation qualitative** a été conduite en superposant les maillages reconstruits aux objets visibles dans les images RGB-D, permettant de juger de la fidélité visuelle. Les résultats sont présentés sous forme de tableaux individuels, croisant objets et méthodes sur une même image choisie dans le jeu de données de test. Cela met en évidence l'impact du modèle de segmentation en amont sur la reconstruction 3D.

Les tableaux 4.2, 4.3 et 4.4 montrent les résultats obtenus respectivement pour les objets *verre*, *marteau* et *pince*, illustrant les variations de performance selon la méthode de segmentation employée.

Tableau 4.2 – Évaluation du maillage reconstruit : Verre

Méthode	Chamfer Distance	Nb faces	Étanchéité	Fidélité visuel
YOLO	0.009	5234	V	Très bonne
MASK R-CNN	0.018	3512	X	Faible
SAM R-CNN	0.012	4910	V	Bonne

Tableau 4.3 – Évaluation du maillage reconstruit : Marteau

Méthode	Chamfer Distance	Nb faces	Étanchéité	Fidélité visuel
YOLO	0.01	4732	V	Bonne
MASK R-CNN	0.02	3053	X	Faible
SAM R-CNN	0.015	4100	V	Moyenne

Tableau 4.3 – Évaluation du maillage reconstruit : Pince

Méthode	Chamfer Distance	Nb faces	Étanchéité	Fidélité visuel
YOLO	0.093	5013	V	Bonne
MASK R-CNN	0.017	3345	X	Faible
SAM R-CNN	0.0135	4300	V	Moyenne

4.4. ÉTUDES DE CAS ILLUSTRATIVES

Afin d'illustrer concrètement l'impact des différentes méthodes de segmentation sur la qualité finale des reconstructions 3D, nous avons sélectionné plusieurs objets représentatifs

issus de notre environnement expérimental. Cette section propose une évaluation qualitative à travers des cas concrets, venant compléter les analyses quantitatives précédentes.

Trois objets types ont été retenus : Un verre — de forme cylindrique, transparent et aux contours lisses ; Une pince — présentant une géométrie fine et discontinue ; Un marteau — partiellement occulté dans certaines vues. Ces objets ont été choisis pour leur diversité morphologique, leur complexité visuelle et leur fréquence dans les scènes capturées avec le HoloLens 2. Pour chaque méthode de segmentation testée (YOLOv8-SEG, Mask R-CNN, SAM-Faster R-CNN), nous présentons une séquence d'images illustrant le traitement complet de chaque objet. Chaque figure comprend : L'image RGB originale capturée par le capteur du HoloLens ; L'image segmentée correspondant, produit par le modèle considéré ; Le maillage 3D obtenu à partir de la projection du masque et de la reconstruction implicite.

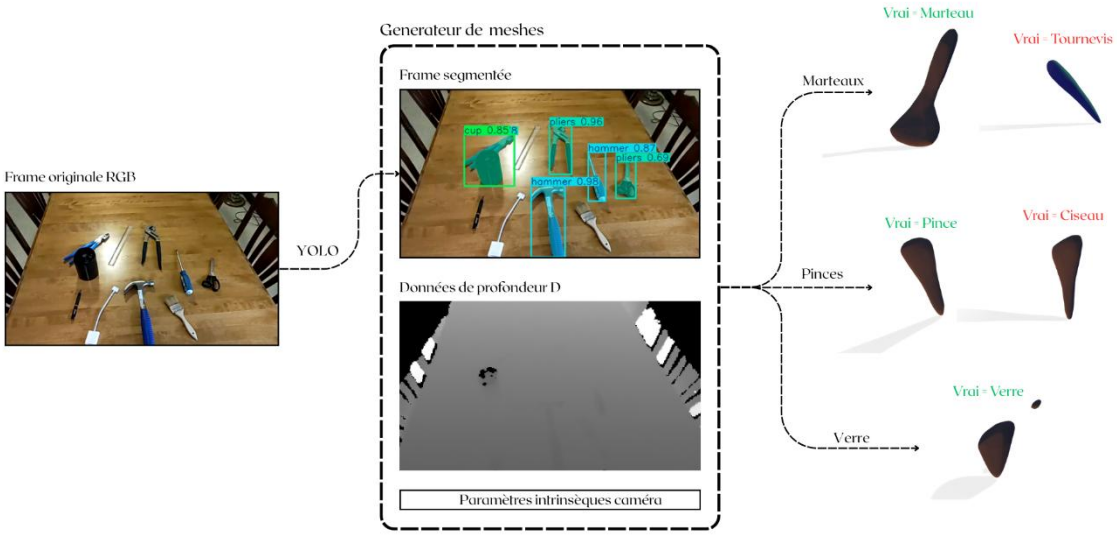

Figure 4.10 — Cas d'utilisation de la reconstruction en utilisant YOLO

Les masques produits par YOLO sont nets, bien centrés, avec une bonne précision sur les objets bien visibles. Les maillages reconstruits sont visuellement fidèles, notamment

pour les marteaux et les pinces. Cependant, on note des artefacts ou une forme fragmentée sur le verre, révélant la difficulté du modèle à gérer les objets transparents ou faiblement contrastés.

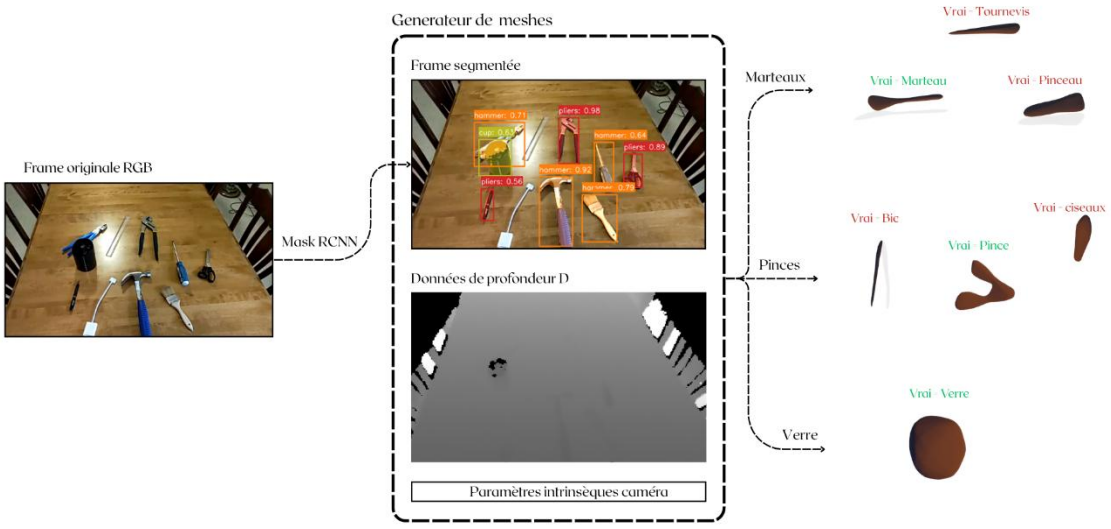

Figure 4.11 — Cas d'utilisation de la reconstruction en utilisant Mask RCNN

Cette figure met en évidence certaines imprécisions de segmentation, notamment des erreurs de classification (ex. tournevis identifié comme marteau). Les contours segmentés sont parfois discontinus, ce qui engendre des reconstructions moins précises, voire erronées. On observe des maillages déformés ou non conformes aux objets d'origine, en particulier pour les pinces et les petits objets.

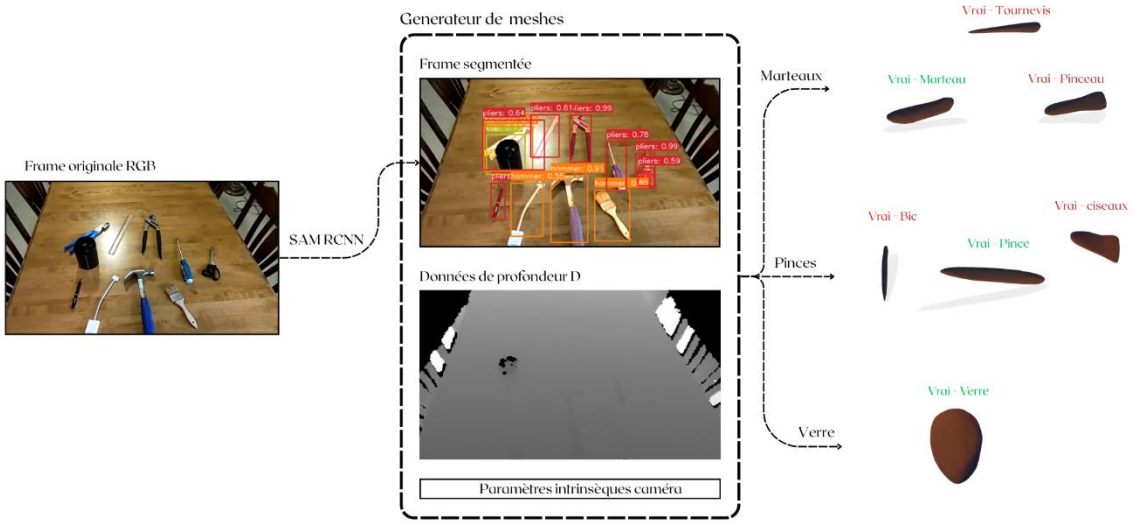

Figure 4.12 — Cas d'utilisation de la reconstruction en utilisant SAM RCNN

Le modèle hybride SAM-Faster R-CNN offre un bon compromis. Les masques générés sont plus détaillés et mieux adaptés aux objets partiellement visibles. Les maillages produits sont globalement plus réguliers, notamment pour les objets fins. On remarque toutefois des cas de sur-segmentation ou de confusion sur les petits objets (ex. bic ou ciseaux).

Dans l'ensemble, ces études de cas soulignent l'importance critique de la qualité de la segmentation dans le succès du pipeline de reconstruction 3D. Elles montrent également que chaque méthode a ses forces spécifiques, et que leur usage peut être guidé par le contexte d'application ou le type d'objet visé.

4.5. LIMITES IDENTIFIÉES ET DISCUSSION

Malgré les résultats encourageants obtenus à travers notre pipeline de segmentation et de reconstruction 3D, plusieurs limites ont été observées au fil de l'expérimentation. Ces limites, souvent liées à la qualité des données en entrée, aux choix méthodologiques ou aux

performances algorithmiques, soulignent les défis persistants dans le cadre d'une reconstruction fidèle et généralisable en réalité mixte.

L'une des principales sources de fragilité réside dans la segmentation initiale. Les erreurs produites par les modèles, même minimes, se répercutent directement sur la qualité du maillage. Par exemple, certaines classes d'objets peu texturées ou partiellement visibles, comme les ciseaux ou le verre, ont été mal identifiées par Mask R-CNN, entraînant des reconstructions imprécises voire incohérentes. YOLO, bien qu'efficace en temps réel, montre parfois des faiblesses sur les objets de petite taille ou fortement occultés. SAM-RCNN, de son côté, offre une segmentation plus fine, mais reste dépendant de la qualité des boîtes initialement détectées. Ainsi, la robustesse de la segmentation conditionne fortement la fiabilité de l'étape suivante.

En ce qui concerne la reconstruction 3D, le recours à une méthode de modélisation implicite à base de fonction d'occupation (OccupancyNet) nous a permis de produire des surfaces régulières et continues, avec un bon comportement topologique. Toutefois, cette méthode n'est pas exempte de limites. La reconstruction reste sensible à la qualité des données de profondeur, qui peuvent être bruitées ou incomplètes dans certaines zones. De plus, le processus d'échantillonnage des points négatifs dans le cube normalisé, nécessaire à l'apprentissage, peut engendrer des biais, en particulier pour les objets de très petite taille. Enfin, bien que Marching Cubes soit un outil puissant pour l'extraction de surface, il introduit parfois des artefacts dans les zones peu couvertes ou mal reconstruites, notamment lorsque les données segmentées sont imprécises.

Sur le plan pratique et expérimental, notre étude s'est déroulée dans un environnement contrôlé avec un nombre limité d'images, ce qui a permis une bonne reproductibilité mais réduit la diversité des cas rencontrés. La généralisation des modèles à

des scènes plus variées ou en conditions non contrôlées reste donc à démontrer. Par ailleurs, même si les temps d'inférence sont compatibles avec des applications semi-temps réel, l'ensemble du pipeline (acquisition, segmentation, projection, reconstruction) reste trop coûteux pour une intégration immédiate dans une application embarquée ou interactive.

Ces constats mettent en lumière plusieurs pistes prometteuses : l'enrichissement du jeu de données, l'optimisation conjointe des masques et des surfaces reconstruites, ou encore l'exploration d'alternatives fondées sur des représentations plus expressives, comme les SDF ou les réseaux NeRF. Autant de directions qui, explorées dans la suite de ce projet, pourraient contribuer à renforcer la robustesse, la précision et la généricité de la reconstruction 3D en réalité mixte.

CHAPITRE V

CONCLUSION ET PERSPECTIVES

Ce chapitre de clôture propose une mise en perspective globale des travaux réalisés. Il débute par une synthèse des contributions majeures apportées par cette recherche, avant d'aborder les limites rencontrées et les pistes d'amélioration possibles. Enfin, il s'ouvre sur les retombées potentielles de cette approche, tant sur le plan applicatif que pour de futures recherches dans le domaine de la réalité mixte et de la reconstruction 3D.

5.1. SYNTHÈSE DES APPORTS DE LA RECHERCHE

Ce mémoire a proposé une approche hybride et originale pour la segmentation d'objets et la reconstruction 3D à partir d'images RGB-D dans un contexte de réalité mixte. En s'appuyant sur les capacités du HoloLens 2 pour la capture synchronisée de données visuelles et de profondeur, nous avons développé un pipeline complet allant de la détection d'objets à la génération de maillages exploitables dans un environnement immersif.

L'un des apports majeurs de cette recherche réside dans l'intégration comparative de trois méthodes de segmentation avancées : YOLO, Mask R-CNN et un modèle hybride SAM-RCNN, combinant Faster R-CNN pour la détection d'objets avec la capacité de généralisation du Segment Anything Model. Cette triple évaluation, menée sur un jeu de données personnalisé annoté manuellement, a permis d'identifier les points forts et les faiblesses de chaque approche en conditions réelles.

Un autre apport important est l'adoption d'un modèle implicite de reconstruction 3D basé sur les Occupancy Networks. Cette stratégie permet de générer des maillages continus et géométriquement cohérents à partir de simples masques projetés et de cartes de

profondeur, sans passer par une triangulation classique. L'architecture MLP utilisée, associée à une normalisation spatiale rigoureuse et une évaluation par Marching Cubes, garantit des surfaces fermées, lisses et adaptées aux besoins de visualisation en réalité mixte.

Enfin, le pipeline a été conçu pour rester modulaire, reproductible et adaptable. Il permet l'interchangeabilité des méthodes de segmentation, l'extension à d'autres types de capteurs ou de scènes, ainsi que l'intégration directe dans des environnements immersifs.

5.2. LIMITES ET PISTES D'AMÉLIORATION

Bien que le pipeline proposé ait démontré des performances satisfaisantes et une robustesse globale en contexte contrôlé, plusieurs limites ont été identifiées au cours de cette recherche. Ces contraintes, à la fois techniques, méthodologiques et expérimentales, offrent des pistes claires pour l'amélioration future du système.

Premièrement, la qualité de la segmentation initiale constitue un facteur déterminant pour la précision de la reconstruction 3D. Les erreurs de prédiction des masques — qu'il s'agisse de faux positifs, de contours flous ou d'objets mal détectés — se répercutent directement sur la génération des maillages. Une piste d'amélioration consisterait à intégrer une étape de raffinement post-segmentation, par exemple à l'aide de Conditional Random Fields (CRF) ou d'algorithmes d'affinage basés sur la géométrie.

Deuxièmement, la qualité des cartes de profondeur reste une limite importante. Bien que le capteur du HoloLens 2 offre une perception 3D satisfaisante, des artefacts et du bruit persistent, notamment sur les surfaces brillantes ou transparentes. Une fusion multi-vues ou une calibration plus poussée pourrait permettre d'améliorer la fidélité des nuages de points reconstruits, en particulier pour des objets fins ou à géométrie complexe.

Troisièmement, la méthode actuelle de reconstruction par fonction d'occupation, bien qu'efficace, reste sensible à l'échantillonnage et à la distribution des points d'entraînement. Une piste intéressante serait d'explorer des représentations plus expressives telles que les Signed Distance Functions ou les Neural Radiance Fields, qui permettent une description plus fine de la surface et une reconstruction plus fidèle, notamment dans les zones partiellement observées.

Sur le plan expérimental, le jeu de données reste modeste (534 images), capturé dans un environnement unique. Cela limite la capacité de généralisation du pipeline à des scènes plus variées, bruyantes ou dynamiques. L'extension de la base d'entraînement, l'annotation semi-automatique, ou l'utilisation de données synthétiques pourraient accroître la robustesse du système.

Enfin, l'ensemble du pipeline — de la capture à la reconstruction — n'est pas encore temps réel. Si chaque étape reste rapide individuellement, une optimisation globale (notamment via des inférences asynchrones ou des architectures plus légères) serait nécessaire pour envisager une intégration fluide dans des applications embarquées ou interactives.

En somme, ces limites ne remettent pas en cause la validité du pipeline proposé, mais mettent en lumière des opportunités concrètes d'optimisation, de généralisation et d'enrichissement fonctionnel pour les futurs travaux.

5.3. RETOMBÉES POTENTIELLES ET APPLICATIONS FUTURES

Les résultats obtenus dans le cadre de cette recherche ouvrent la voie à plusieurs perspectives concrètes, tant sur le plan technologique que scientifique. En combinant des modèles avancés de segmentation avec une stratégie de reconstruction implicite, notre

pipeline propose une solution efficace et relativement légère pour générer des représentations 3D exploitables en réalité mixte, à partir de simples images RGB-D.

Dans un contexte applicatif, cette approche pourrait être intégrée dans des systèmes interactifs pour l'assistance industrielle, la modélisation d'objets en environnements réels, ou encore la documentation patrimoniale. L'utilisation de la HoloLens 2 comme dispositif de capture et d'interaction place ce travail au cœur des technologies immersives en plein essor.

Par ailleurs, la modularité du pipeline développé permettrait une extension vers d'autres domaines comme la robotique autonome (manipulation d'objets), l'architecture augmentée ou encore les environnements d'apprentissage immersif. La capacité à générer des maillages cohérents à partir d'annotations rapides, sans scanner 3D dédié, représente une avancée prometteuse pour des scénarios où la mobilité, la flexibilité et le coût sont des enjeux clés.

Enfin, sur le plan scientifique, cette recherche contribue à la réflexion autour des architectures hybrides combinant modèles de fondation et entraînement supervisé. Elle ouvre également des perspectives vers l'optimisation conjointe de la segmentation et de la reconstruction 3D, une problématique encore peu explorée mais riche en potentiel.

BIBLIOGRAPHIE

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. IEEE TPAMI, 34(11), 2274–2282. https://doi.org/10.1109/TPAMI.2012.120

Adams, R., & Bischof, L. (1994). Seeded Region Growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 641–647. https://doi.org/10.1109/34.295913

Azuma, R. T. (1997). A survey of augmented reality. Presence: Teleoperators & Virtual Environments, 6(4), 355–385. https://doi.org/10.1162/pres.1997.6.4.355

Billinghurst, M., Clark, A., & Lee, G. (2015). A Survey of Augmented Reality. Foundations and Trends® in Human–Computer Interaction, 8(2-3), 73–272. https://doi.org/10.1561/1100000049

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time Instance Segmentation. ICCV. https://doi.org/10.1109/ICCV.2019.00709

Bouchard, K., Bouchard, B. & Bouzouanea, A. (2014). Practical guidelines to build smart homes : lessons learned. Opportunistic networking, smart home, smart city, smart systems, pp. 1–37

Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Craig, A. B. (2013). Understanding augmented reality: Concepts and applications. Morgan Kaufmann.

Delaunay, B. (1934). Sur la sphère vide. Bulletin de l'Académie des sciences de l'URSS. Classe des sciences mathématiques et naturelles, 6, 793–800.

Di Benedetto, J. (2021). HL2SS – HoloLens 2 Streaming System. GitHub. https://github.com/jdibenes/hl2ss

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/2010.11929

Edelsbrunner, H., & Mücke, E. P. (1994). Three-dimensional alpha shapes. ACM Transactions on Graphics (TOG), 13(1), 43–72. https://doi.org/10.1145/174462.156635

Espinosa, H. D. et al. (2020). Improved DBSCAN algorithm for 3D point cloud segmentation in complex scenes. Sensors, 20(18), 5265. https://doi.org/10.3390/s20185265

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision, 111(1), 98–136. https://doi.org/10.1007/s11263-014-0733-5

Flavián, C., Ibáñez-Sánchez, S., & Orús, C. (2019). The impact of virtual, augmented and mixed reality technologies on the customer experience. Journal of Business Research, 100, 547–560.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354–3361). https://doi.org/10.1109/CVPR.2012.6248074

Gupta, S., Girshick, R., Arbelaez, P., & Malik, J. (2014). Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In European Conference on Computer Vision (pp. 345–360). Springer. https://doi.org/10.1007/978-3-319-10584-0_23

Hazirbas, C., Ma, L., Domokos, C., & Cremers, D. (2016). FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. Asian Conference on Computer Vision.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961–2969). https://doi.org/10.1109/ICCV.2017.322

Jocher, G. et al. (2023). Ultralytics YOLO v8. GitHub Repository. https://github.com/ultralytics/ultralytics

Kazhdan, M., & Hoppe, H. (2013). Screened Poisson Surface Reconstruction. ACM Transactions on Graphics, 32(3), 1–13. https://doi.org/10.1145/2487228.2487237

Kim, K., Billinghurst, M., Bruder, G., Duh, H. B.-L., & Welch, G. F. (2018). Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008–2017). IEEE Transactions on Visualization and Computer Graphics, 24(11), 2947–2962. https://doi.org/10.1109/TVCG.2018.2868591

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019). Panoptic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 9404–9413. https://doi.org/10.1109/CVPR.2019.00963

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Dollár, P. (2023). Segment Anything. arXiv preprint arXiv:2304.02643. https://arxiv.org/abs/2304.02643

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117–2125. https://doi.org/10.1109/CVPR.2017.106

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In European Conference on Computer Vision (ECCV), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Lorensen, W. E., & Cline, H. E. (1987). Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM SIGGRAPH. https://doi.org/10.1145/37401.37422

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv preprint arXiv:1608.03983.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Massa, F., & Girshick, R. (2018). Detectron2. Facebook AI Research. https://github.com/facebookresearch/detectron2

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy Networks: Learning 3D reconstruction in function space. CVPR. https://doi.org/10.1109/CVPR.2019.00181

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Zhu, H. (2018). Mixed Precision Training. ICLR 2018.

Microsoft. (2020). HoloLens 2 Overview. Retrieved from https://learn.microsoft.com/en-us/hololens/hololens2-hardware

Microsoft. (2023). Mixed Reality Toolkit (MRTK) for Unity. Microsoft Learn. https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/

Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. IEICE Transactions on Information and Systems, 77(12), 1321–1329.

Minderer, M., Pink, A., Ablavatski, A., Goodman, N., & Murphy, K. (2022). Simple Open-Vocabulary Object Detection with Vision Transformers. arXiv preprint arXiv:2205.06230. https://arxiv.org/abs/2205.06230

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., ... & Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In 2011 10th IEEE International Symposium on Mixed and Augmented Reality (pp. 127–136). IEEE. https://doi.org/10.1109/ISMAR.2011.6092378

Nguyen, R., Boudet, A., & Tarpin-Bernard, F. (2023). A Multimodal Interaction Interface for Mixed Reality Using Hand and Voice Commands: Case Study on Microsoft HoloLens 2. In Proceedings of the 2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), pp. 134–137. IEEE. https://doi.org/10.1109/ISMAR-Adjunct58608.2023.00036

Nicolau, S., Soler, L., Mutter, D., & Marescaux, J. (2022). Augmented reality in surgical navigation: From concept to clinical practice. Surgical Endoscopy, 36(4), 2123–2136. https://doi.org/10.1007/s00464-021-08789-4

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). DeepSDF: Learning continuous signed distance functions for shape representation. CVPR. https://doi.org/10.1109/CVPR.2019.00434

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. CVPR.

Rauschnabel, P. A., Felix, R., Hinsch, C., Shahab, H., & Alt, F. (2022). What is XR? Towards a Framework for Augmented and Virtual Reality. Computers in Human Behavior, 133, 107289. https://doi.org/10.1016/j.chb.2022.107289

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems (NeurIPS), 28.

Rhee, T., Petikam, L., Allen, B., & Chalmers, A. (2017). Mr360: Mixed reality rendering for 360° panoramic videos. IEEE Transactions on Visualization and Computer Graphics, 23(4), 1379–1388. https://doi.org/10.1109/TVCG.2017.2656958

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts. ACM TOG, 23(3), 309–314. https://doi.org/10.1145/1015706.1015720

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A Database and Web-Based Tool for Image Annotation. International Journal of Computer Vision, 77(1), 157–173. https://doi.org/10.1007/s11263-007-0090-8

Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H. J., & Davison, A. J. (2013). SLAM++: Simultaneous localisation and mapping at the level of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1352–1360. https://doi.org/10.1109/CVPR.2013.179

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 567–576). https://doi.org/10.1109/CVPR.2015.7298655

Speicher, M., Hall, B. D., & Nebeling, M. (2019). What is Mixed Reality? In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), Paper 537. https://doi.org/10.1145/3290605.3300767

Szeliski, R. (2010). Computer vision: Algorithms and applications. Springer. https://doi.org/10.1007/978-1-84882-935-0

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). KPConv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6411–6420). https://doi.org/10.1109/ICCV.2019.00651

Ultralytics. (2023). Ultralytics YOLO v8 Documentation. https://docs.ultralytics.com

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics (TOG), 38(5), 1–12. https://doi.org/10.1145/3326362

Zhang, Y., Li, T., Wang, J., & Xu, K. (2020). Real-Time RGB-D Data Acquisition and Cloud Offloading for Mobile Mixed Reality Systems. IEEE Transactions on Multimedia, 22(8), 2102–2114. https://doi.org/10.1109/TMM.2019.2940988

Zhao, H., Jiang, L., Jia, J., Torr, P. H., & Koltun, V. (2021). Point Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 16259–16268).

Zhou, F., Duh, H. B. L., & Billinghurst, M. (2020). Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. In Proceedings of the IEEE Transactions on Visualization and Computer Graphics, 26(5), 1631–1648. https://doi.org/10.1109/TVCG.2019.2899352

Zollhöfer, M., Marton, Z.-C., Whelan, T., Chambers, E., Stueckler, J., & Theobalt, C. (2018). State of the art on 3D reconstruction with RGB-D cameras. Computer Graphics Forum, 37(2), 625–652. https://doi.org/10.1111/cgf.13386