Faculté des Sciences

Département de Géographie

PROJECTION DE VUES ORTHOGRAPHIQUES À PARTIR DE CHAMPS DE RADIANCE NEURONAUX (NERF) : VERS UNE DÉMOCRATISATION DES RELEVÉS ARCHITECTURAUX DE FAÇADES.

Mémoire présenté par : **Cyril WILKIN**

pour l'obtention du titre de

Master en sciences géographiques, orientation géomatique, à finalité spécialisée en geodata-expert

Année académique	:	**2024-2025**
Date de défense	:	**Septembre 2025**
Président de jury	:	Pr **René WARNANT**
Promoteur	:	Pr **Pierre HALLOT**
Jury de lecture	:	Pr **Roland BILLEN**
		Pr **Pierre HALLOT**
		Pr **René WARNANT**

REMERCIEMENTS

Tout d'abord, je tiens à remercier mon promoteur M. Hallot pour m'avoir suggéré un sujet de mémoire qui m'a autant passionné. Ses retours et ses conseils avisés n'ont pas manqué d'ouvrir de nouvelles perspectives dans mon esprit à chacune de nos réunions.

Je remercie également mes lecteurs M. Billen et M. Warnant pour l'intérêt porté à mon travail et leurs interventions dans les moments d'incertitude de celui-ci.

Un immense merci à Anicée Le Lay, Axel Jamar, Cadwal Borremans, Caroline Villance et Jean-Marc Wilkin pour leur disponibilité et leurs commentaires avisés lors de leurs nombreuses relectures. Leurs points de vue ayant réellement étayés mon travail.

Ce travail n'aurait pas été possible sans l'éternel et indéboulonnable soutien de mes amis Antoine Alexandre, Antoine Dricot, Nathan Vanesse et Tom Denis.

Il n'aurait pas été possible non plus sans la tendre et chaleureuse force de mes parents.

RÉSUMÉ

Les relevés architecturaux de façades s'appuient actuellement sur des méthodes traditionnelles couteuses en matériel et en temps ou nécessitant une expertise avancée.

Ce mémoire explore l'utilisation des Neural Radiance Fields (NeRF), une technique récente et prometteuse de modélisation 3D permettant de reconstruire des scènes réelles, y compris des textures complexes. L'étude se concentre sur leur application aux relevés architecturaux.

L'objectif principal est de déterminer si les NeRF peuvent rendre les relevés plus accessibles, y compris pour des utilisateurs novices. Deux hypothèses guident la recherche : la possibilité de produire une vue orthographique à partir d'un NeRF et l'accessibilité d'un workflow simplifié à un public non spécialiste.

Cette étude démontre qu'il est effectivement possible de produire des vues orthographiques par projection via les NeRF d'une part et de proposer un workflow accessible à un public de personnes non-initiées pour un coût raisonnable d'autre part. Un logiciel push button et une application web ont été créés. Ce travail s'est notamment appuyé sur la combinaison d'un capteur LiDAR d'iPhone et sur le framework Nerfstudio.

Ces résultats laissent entrevoir de belles avancées pour le futur, en particulier dans les domaines de l'archéologie et de l'architecture en rendant les relevés de haute qualité plus efficaces en ressources et largement accessibles.

ABSTACT

Architectural facade surveys currently rely on traditional methods that are either costly in terms of equipment and time or require advanced expertise.

This master's project investigates the use of Neural Radiance Fields (NeRF), a recent and promising 3D modeling technique capable of reconstructing real-world scenes, including complex textures, with a focus on their application to architectural surveys.

The main objective is to determine whether NeRF can make surveys more accessible, including for novice users. The research is guided by two hypotheses: first, that NeRF can be used to generate orthographic views, and second, that it can support a simplified workflow for a non-specialized audience.

The study demonstrates that orthographic views can indeed be produced via NeRF and that a workflow accessible to lay users is feasible at a reasonable cost. To this end, a push-button software and a web application were developed, leveraging an iPhone LiDAR sensor and the Nerfstudio framework.

These results suggest significant potential for future applications, particularly in the fields of archaeology and architecture, by making high-quality surveys more efficient and broadly accessible.

TABLE DES MATIÈRES

INTRODUCTION

Très récemment, l'introduction du concept de champ de radiances neuronal (NeRF) par Mildenhall et al. (2020) a véritablement ouvert un nouveau champ de recherche dans le domaine de la représentation 3D. Les applications sont nombreuses et variées, en particulier pour l'architecture et la conservation du patrimoine. Les NeRF représentent une possible nouvelle démarche de relevé architectural grâce à la production de représentations 3D détaillées, photoréalistes et rapides même pour des scènes complexes. En quelques années, de nombreuses études ont investigué la méthode en tant que potentiel outil de documentation et de communication. Les résultats sont de plus en plus pertinents à l'instar des techniques classiques de relevé.

Ce mémoire s'inscrit dans ce mouvement. En plus de nécessiter l'intervention d'un opérateur formé et expert, le relevé architectural est une démarche très souvent couteuse en temps et en matériel. Les résultats prometteurs des NeRF combinés aux nombreuses applications apparues dans un contexte de dynamisme offrent une possible solution pour démocratiser la production de relevés.

Afin d'explorer cette possibilité, le travail se concentre sur un type de représentation 2D issu des relevés : les projections de vues orthographiques. Ce support, souvent dérivé de nuages de points est un excellent outil de communication et de mesure. Les objectifs du mémoire sont d'évaluer si leur production est possible à partir de ce nouveau concept de NeRF et si les avantages rapportés permettent de rendre l'ensemble accessible à un public élargi.

Le travail s'articule sur trois expériences successives et dont les résultats peuvent être comparés. Les trois expériences partagent les mêmes données de référence et visent toutes l'obtention d'une vue orthographique par projection. Au départ, un workflow complet et complexe pour experts est obtenu. Il est ensuite simplifié pour un public intermédiaire grâce à l'utilisation d'un logiciel à interface simplifiée. Pour terminer, il est adapté en une application web accessible à un large public.

Les implémentations et les workflows ainsi obtenus devraient permettre de déterminer si la technique des NeRF est pertinente pour la projection de vues orthographiques. Cette approche s'inscrit dans la perspective d'une future démocratisation et d'une diffusion des relevés architecturaux auprès d'un public peu formé.

CHAPITRE I ÉTAT DE L'ART

Afin de poser les bases de cette étude et d'en comprendre les objectifs, il est nécessaire de passer en revue la littérature concernant le domaine. Pour commencer, ce chapitre propose un rappel des techniques et méthodes actuellement disponibles pour effectuer un relevé. Une deuxième partie, malgré le ton généraliste de l'état de l'art, est plus technique pour expliquer en profondeur le fonctionnement des NeRF. Cette technologie étant très récente et évoluant rapidement, le reste du chapitre retrace leurs applications en comparaison des techniques classiques et les axes de recherches actuels.

I-1 DÉFINITION D'UN RELEVÉ ARCHITECTURAL DE FAÇADE

La définition d'un relevé architectural de façade n'est pas si simple à établir. Dans le cadre de ce mémoire, nous le considérerons comme un cas spécifique de levé topographique. Ce dernier pouvant être défini comme suit : « Le levé consiste à se rendre sur le terrain, l'analyser puis, à l'aide d'instruments et de méthodes adaptées aux problèmes posés, enregistrer les observations qu'elles soient du type graphique, manuscrite ou numérique. » (Billen, 2020). Dans un contexte architectural, nous considérons que : « Le relevé extérieur d'ouvrage architectural est le moyen de créer une représentation du bâtiment dans son état de conservation. » (Alby, 2006). Pour résumer, ce mémoire adopte la définition suivante du relevé architectural de façade: *Un ensemble de techniques permettant d'acquérir mais aussi de stocker et de diffuser la représentation d'un bâtiment déjà existant, en particulier ses façades*.

I-2 UTILITÉ ET APPLICATIONS DES RELEVÉS

Le processus de relevé est très largement utilisé et dessert plusieurs secteurs d'activité(Boje et al., 2023) parmi lesquels le bâtiment, l'architecture, la gestion des risques et leur prévention ou encore la conservation du patrimoine (Costantino et al., 2023).

Comme défini précédemment, l'acquisition d'un relevé sert de base pour l'étude des bâtiments existants. L'article de Banfi et al. (2022) montre comment la modélisation obtenue après un relevé peut servir pendant tout le cycle de vie d'un bâtiment. L'article présente toute la méthodologie, de l'acquisition à la visualisation dans une API (Application Programing Interface) par l'utilisateur final (l'habitant, les experts et les différents corps de métiers). Pour résumer, le relevé permet d'établir un état des lieux puis d'effectuer un suivi pendant et après un projet.

En architecture, le relevé a de nombreuses applications. En effet, l'information géométrique peut se combiner avec d'autres données. Par exemple, en la combinant avec une analyse des matériaux et des simulations physiques, elle permet de mettre en évidence des zones endommagées que ce soit pour la prévention ou la réparation (Russo et al., 2019). Il est aussi possible d'évaluer la consommation énergétique, la qualité de l'air en intérieur ou l'ensoleillement (gain solaire) (Boje et al., 2023), qui sont très souvent des critères importants pour les clients. Combiné à une imagerie thermique, le relevé permet de mettre en évidence les jonctions qui sont souvent problématiques en termes d'isolation puis d'évaluer l'efficacité d'une rénovation (Banfi et al., 2022).

En conservation du patrimoine, la fabrication d'un jumeau numérique (DT) permet de créer une documentation exhaustive sur de nombreux aspects vitaux pour la préservation de bâtiments anciens. Selon Borkowski et Kubrat, (2024), ces aspects sont: la géométrie du bâtiment (dimensions et volumes), le style architectural dans lequel le bâtiment analysé a été conçu, les éléments structurels, les techniques de construction caractéristiques, les types et les juxtapositions des matériaux, le degré de dégradation des façades du bâtiment et d'autres dommages structurels éventuels. Ils permettent de répertorier les interventions effectuées sur les façades, de détailler l'étendue de l'intervention ou encore de simuler l'impact des facteurs externes sur le bâtiment et les risques encourus au cours du temps.

Cette documentation peut également être utilisée à des fins touristiques et éducationnelles grâce à la réalité augmentée (Kong, 2024; Zachos et Anagnostopoulos, n.d.).

I-3 APERÇU DES TECHNIQUES ACTUELLES

Puisque le mémoire se concentre sur l'utilisation d'une nouvelle technique dans le cadre des relevés architecturaux, elle est souvent comparée avec les techniques (dites classiques) en utilisation actuellement. En conséquence, cette section retrace rapidement les techniques classiques en mettant en évidence les points importants à discuter sans entrer dans le détail de chacune.

I-3.1 APPROCHE À DEUX DIMENSIONS OU À TROIS DIMENSIONS ?

De nos jours, les techniques sont nombreuses et varient fortement. Mais elles présentent des avantages différents et se complètent souvent. Au départ manuelles, les techniques de représentation 2D ont, de nos jours, évolué vers des modèles 3D.

Voici un bref historique (Alby, 2006) : *« La forme de représentation d'un relevé a évolué depuis le plan vers le modèle tridimensionnel numérique. Les plans sont utilisés comme extrait du modèle dont la fonction est de consigner le relevé sous la forme la plus représentative qu'est le modèle numérique. Cette migration est liée aux progrès de l'informatique mais aussi des techniques d'acquisitions*. ». En d'autres mots, la réalisation de plans en 2D est loin d'être obsolète mais constitue un produit supplémentaire après la modélisation.

Comme nous le verrons dans ce mémoire, obtenir un plan 2D, à partir d'un modèle 3D constitue une bonne approche pour avoir un aperçu de l'efficacité d'une technique.

I-3.2 DÉROULEMENT / MÉTHODOLOGIE GÉNÉRALE

Bien que différentes, les techniques suivent généralement le déroulé suivant (Alby, 2006; Boje et al., 2023; Borkowski et Kubrat, 2024):

- Phase d'acquisition : sur le terrain, un opérateur vient capturer de l'information géographique (manuellement ou avec une station totale, un LiDAR, un appareil photographique, …).
- Phase de prétraitement : l'information obtenue est traitée pour devenir utilisable (par exemple : géoréférencement, géocodage, nettoyage et fusion des nuages de points).
- Phase de modélisation (et d'export) : une représentation ou un modèle est produit (reconstruction) et stocké dans un format utilisable.
- Phase d'exploitation : le modèle est utilisé par l'utilisateur selon ses besoins.

I-3.3 MÉTHODES CLASSIQUES

Les méthodes classiques ont fortement varié au fur et à mesure des avancées technologiques (Alby, 2006). Les méthodes classiques de reconstruction 3D dans le cadre des relevés se séparent en deux types de catégories : les méthodes passives comme la photogrammétrie et les méthodes actives comme les mesures à la station totale et le LiDAR. Quatre méthodes classiques couramment utilisées sont décrites ci-dessous.

Méthode Manuelle

Les premières méthodes d'acquisitions établies consistaient en une mesure directe des points d'intérêt par l'opérateur (avec des instruments basiques comme le mètre ruban). La méthode parait simple mais afin d'assurer la validité de la géométrie, elle nécessite une grande rigueur d'exécution pour sélectionner les bonnes côtes et pour garantir la bonne tenue du carnet de terrain (Magri-Djenane et al., 2011).

Station totale

Cette première méthode manuelle manque d'efficacité et de précision, ce que corrige une autre méthode manuelle active consistant à effectuer des mesures précises d'angles et de distances à l'aide d'une station totale dans un levé topographique. « Un tachéomètre moderne équipé d'un distancemètre électronique et permettant de stocker numériquement les mesures effectuées s'appelle une station totale. Les stations totales ont connu de nombreuses évolutions (logiciel de calcul intégré, antenne GNSS intégrée, fonctionnement motorisé (one-man), …) » (Billen, 2020).

Les points relevés par la station peuvent aisément être exportés dans un logiciel de dessin par ordinateur (CAD). De plus, lorsque l'utilisateur a correctement mis en place la géocodification (un code étant assigné à une mesure en fonction de la nature du point), il est souvent possible que les plans et modélisations soient automatiquement réalisés. Cette méthode présente certains défauts : en plus d'être relativement lente, elle demande du matériel couteux et encombrant et a nécessairement besoin d'opérateurs formés à sa réalisation et à son traitement. Malgré ces contraintes, elle représente un moyen fiable, précis et éprouvé d'effectuer un relevé.

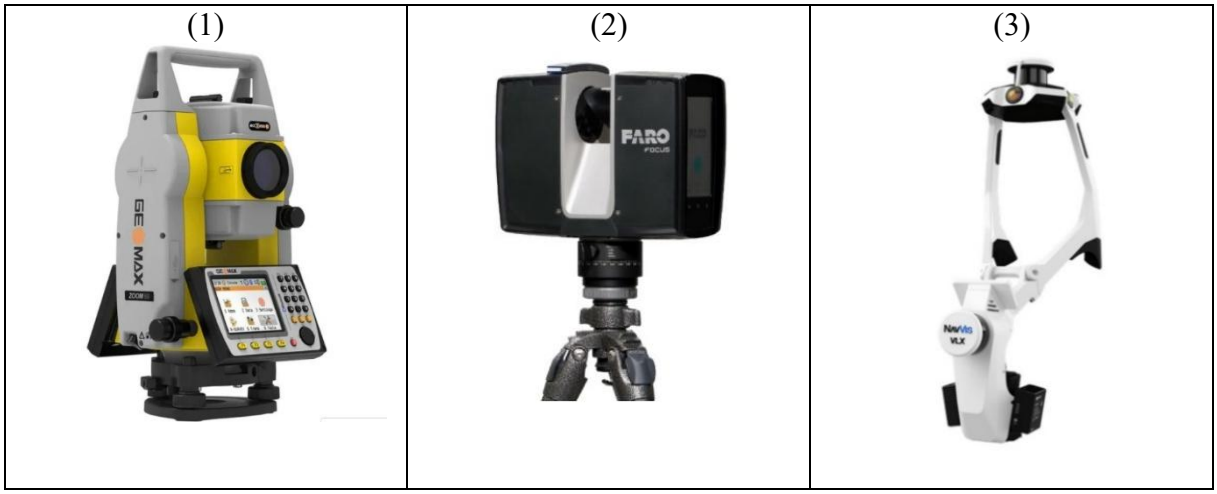


Figure 1 : Capteurs utilisés dans les techniques classiques : (1) Station totale GeoMax Zoom50 (My survey direct, 2025), (2) TLS Faro Focus premium ((Borkowski & Kubrat, 2024), (3) MLS NavVIS VLX 3 (Borkowski & Kubrat, 2024)**.**

LiDAR

Le LiDAR (« Light Detection And Ranging ») » utilise un laser afin de directement et activement mesurer des distances. Le principe est de scanner la scène d'intérêt en émettant des impulsions de lumière très rapidement qui vont venir frapper les surfaces. Le temps de retour (FOV) de l'impulsion rétropropagée de la surface vers le récepteur du LiDAR peut être combiné avec la vitesse lumière pour calculer la distance. Le calcul de distance est également possible en mesurant un déphasage sur un rayon constant. Grâce aux calculs, un point 3D est généré au niveau de la surface, le point peut être colorisé en intégrant une photographie en plus durant le scan. (Artec3D,2025).

De manière générale, le LiDAR est très précis. Cependant, des sources d'erreurs peuvent introduire une variation de quelques millimètres à quelques décimètres. Les sources d'erreurs sont multiples (mauvais étalonnage, mauvaises conditions environnementales (météorologie), bruit de portée, réflexivité de la surface, du temps passé depuis le dernier réétalonnage ou du bruit de portée. (Artec3D,2025).

Les LiDAR sont souvent classés en fonction du support qui les embarquent : des avions pour les LiDAR aéroportés, un support fixe pour les LiDAR stationnaires (« Terrestrial Lidar System » ou TLS) ou un support en mouvement pour les LiDAR mobiles (« Mobile Lidar System » ou MLS). Les TLS sont lents et induisent beaucoup de restrictions d'utilisation mais en contrepartie, ils proposent une excellente précision. Ils sont souvent utilisés comme vérité terrain pour la comparaison des autres techniques. Les principales restrictions du TLS impliquent la répétitivité de longues mises en station pour couvrir l'intégralité de la scène et le préplacement de cibles pour assurer la précision de l'alignement des nuages générés. Les TLS sont donc précis mais lents. Les MLS permettent de réduire cette répétition en effectuant des scans continus lors du parcours de la scène. Les LiDAR aéroportés par avion ou hélicoptère permettent de scanner le sol sur des larges bandes et assurent une couverture sur des kilomètres de scènes mais la résolution des scans est significativement réduite en fonction de la hauteur de vol. (Artec3D,2025).

Récemment, le LiDAR a été embarqué dans des smartphones, notamment Apple, en utilisant un type particulier de capteur miniature le SSL (« solid-state LiDAR »). Apple ne fournit pas la documentation précise de la technologie permettant au capteur d'effectuer le balayage sans posséder de pièce mobile. (Paukkonen, 2023)

Photogrammétrie digitale

La photogrammétrie est une technique passive de reconstruction 3D basée sur l'acquisition et l'analyse d'image. Les méthodes Structure-from-Motion (SfM) et Multi-View-Stereo (MVS) sont largement utilisées et considérées solides pour la reconstruction de modèles 3D (Croce et al., 2024). Elle se base sur le principe de la stéréoscopie à partir du recouvrement d'image 2D (triangulation des positions). Elle est souvent complémentaire avec les méthodes de relevés actives (LidAR, station totale). En effet, la photogrammétrie digitale présente une précision plus basse que le LiDAR mais propose une bien meilleure qualité de couleurs. Comme la technique d'acquisition est plus rapide, la photogrammétrie est souvent utilisée dans le contexte de scènes complexes ou lorsque la finalité se concentre prioritairement sur l'aspect visuel plutôt que sur la précision (comme souvent en conservation du patrimoine). (Salagean-Mohora et al., 2023; Borkowski & Kubrat, 2024)

Les supports de photogrammétrie sont nombreux (appareil photo, scanner, aéroporté, drone) et permettent de facilement générer des jeux d'images. Généralement la reconstruction implique l'alignement des images (triangulation des paramètres des caméras) à partir d'un nuage de points (PCD) éparse puis la densification du PCD. Elle peut aussi impliquer une recalibration des caméras et une géoréférenciation (passage dans un système de coordonnées de référence).

I-4 SUPPORTS DE DONNÉES

Les données acquises par les techniques classiques sont stockées sous différents supports de données. Il existe de nombreux logiciels comme MicMAC, RealityCapture ou Agisoft Metashape qui permettent la reconstruction des données 3D et leur stockage sous la forme d'un nuage de points (PCD), de maillages polygonaux (MESH) et de maillages texturés. Des images ortho rectifiées et des cartes bidimensionnelles peuvent même être dérivées d'un modèle photogrammétrique et correctement géoréférencées grâce à leur combinaison avec des méthodes topographiques traditionnelles de levé (Croce et al., 2024). Le LiDAR et la station totale permettent directement de créer un nuage de points à partir de leurs mesures.

L'étude de Villa (2017) donne un aperçu graphique des supports de données utilisés lors d'un relevé architectural dans le domaine de la conservation du patrimoine. La Figure 2 présente brillamment le passage d'un support de données à l'autre lors d'une reconstruction par photogrammétrie. La Figure 3 présente un des principaux dérivés des PCD, la projection orthographique représentant un relevé de façade. Elle est à l'échelle et permet d'effectuer des mesures en plus de fournir un support visuel conservant les dimensions (Gill, 2004).

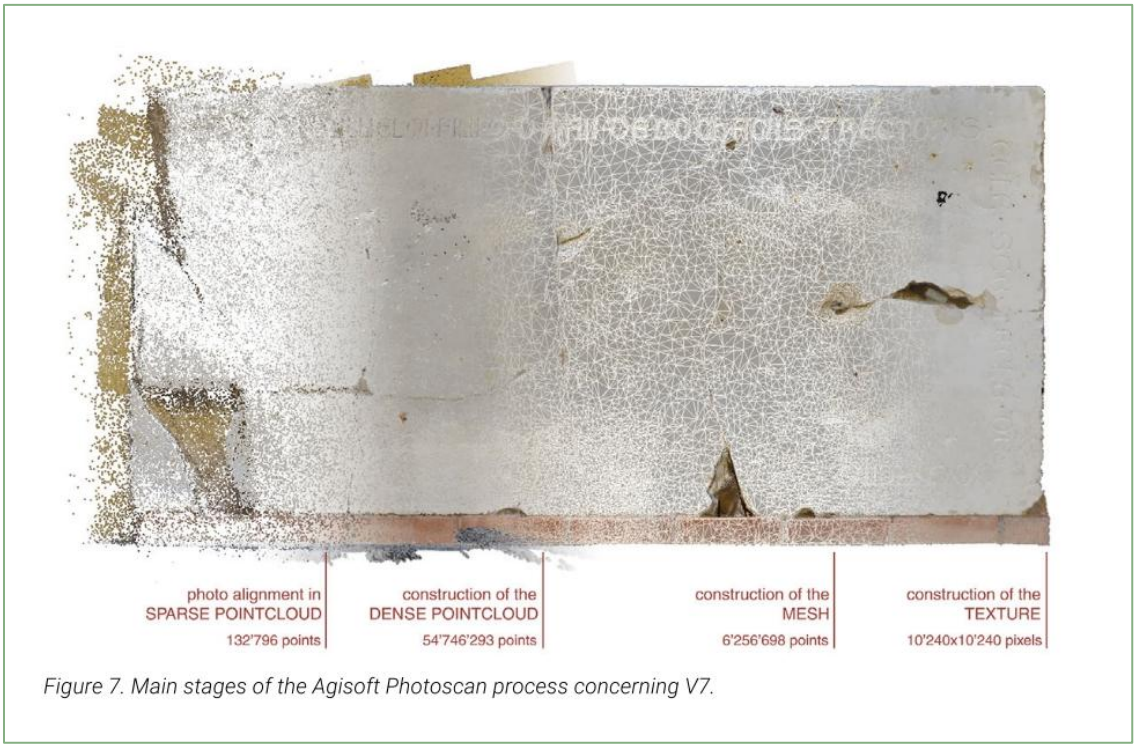


Figure 2 : Visualisation de toutes les étapes de reconstruction 3D par photogrammétrie d'une tombe (Villa et al., 2017)

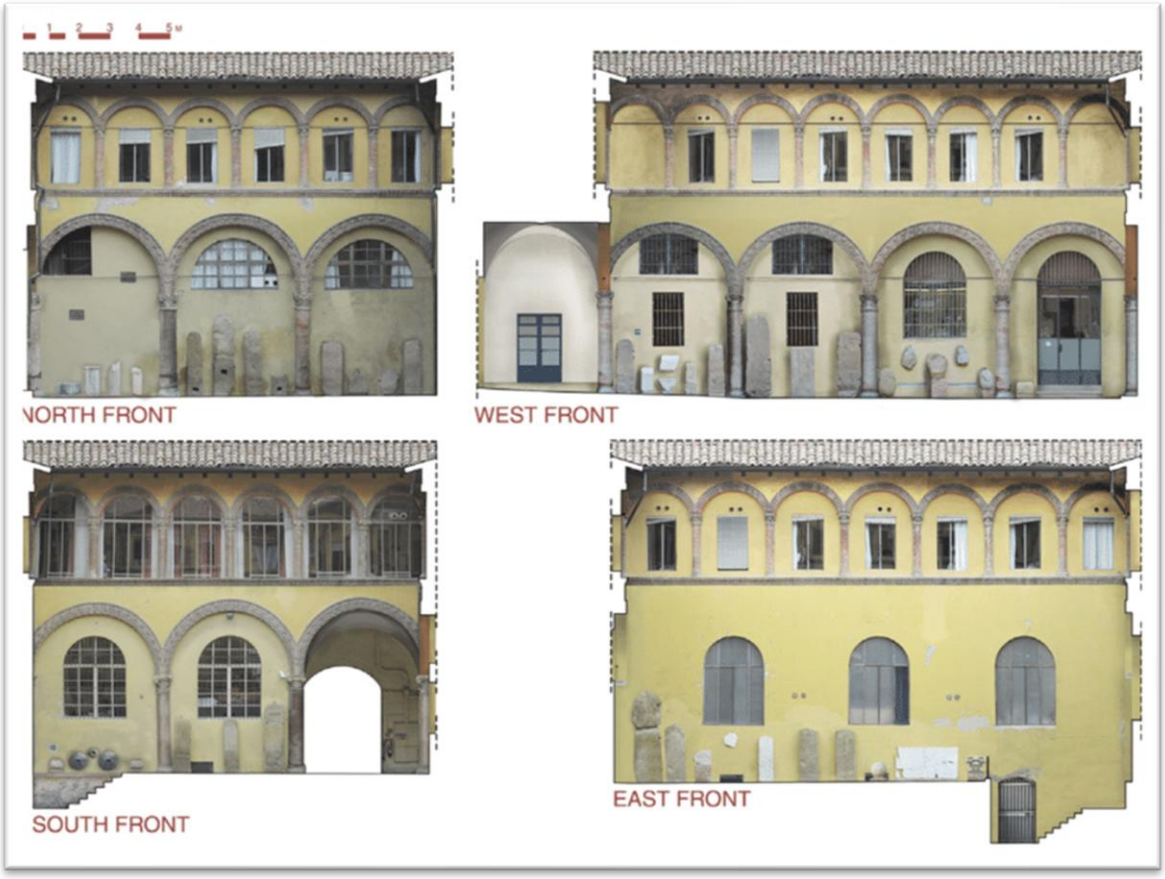


Figure 3 : Les 4 relevés d'élévation des façades du Palazzo Ancarano. (Villa et al., 2017)

I-5 NEURAL RADIANCE FIELDS (NERF)

Avant de comparer la nouvelle technique des NeRF avec les techniques classiques, il convient d'établir un contexte et une base théorique solide. Dans les sections qui suivent, les principes de bases permettant de comprendre leurs utilités et leurs perspectives sont d'abord expliqués. Ensuite, des améliorations du modèle original sont présentées, ainsi que les produits pouvant être élaborés à partir d'un NeRF, notamment ceux qui sont utiles pour les relevés.

Pour donner un rapide contexte: les NeRF ont commencés à gagner en popularité de par leur capacité à créer des reconstructions 3D dans des environnements réels. La recherche a été rapide et pousse le domaine depuis son introduction en 2020. Les améliorations apportées dans ces domaines ont attiré l'intérêt d'une large variété de disciplines, à la fois académiques et industrielles comme la robotique, la cartographie, la simulation, … (Tancik et al., 2023). En particulier dans la conservation du patrimoine où la recherche a été prompte à intégrer la technique (Vacca, 2023; Vandenabeele et al., 2023; Croce et al., 2024).

I-5.1 PRINCIPE DU NEURAL RADIANCE FIELDS

Le principe au cœur des champs de radiance neuronaux (en anglais, Neural Radiance Fields ou NeRF) représente le point le plus technique de ce mémoire. Cette section vulgarise l'article et le guide de Mildenhall et al. (2020) décrivant les mécaniques derrière le NeRF. L'objectif est de décrire celles-ci afin de comprendre tous les éléments pouvant être nécessaires dans la suite du mémoire. La section compile également les travaux des créateurs de Nerfstudio (Tancik et al., 2023) et du cours de Lalonde (2025). Le but est de simplifier au maximum l'approche sans en entacher la rigueur mathématique.

Le but du NeRF est de représenter une scène réelle et statique à l'aide d'une fonction continue définie dans l'espace, dont la forme est apprise en entraînant un réseau de neurones profond et entièrement connecté (ou perceptron multicouche, Multi-Layer Perceptron) (Mildenhall, 2020).

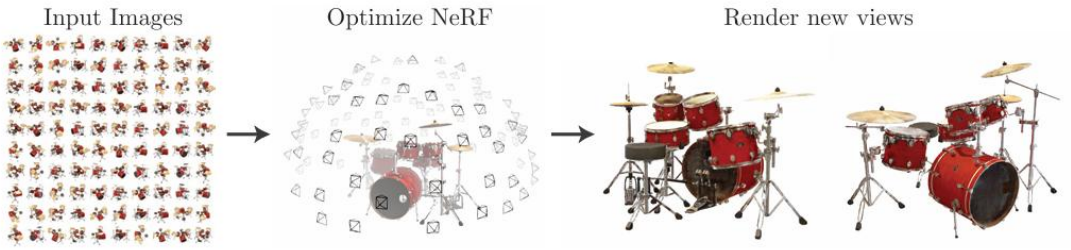


Figure 4 : Représentation générale du principe de la représentation NeRF (Mildenhall,2020).

Les paramètres de cette fonction sont estimés à partir d'images 2D capturant la scène réelle. Les paramètres intrinsèques et extrinsèques de chaque image (caméra ou frame, selon le contexte) doivent être fournis afin de servir d'entrées au réseau. La sortie de la fonction représente la scène en termes de radiance et de densité volumétrique. À chaque itération, le réseau évalue un échantillon de pixels tirés des images de référence. La fonction est régressée en minimisant, à chaque itération d'apprentissage, l'erreur photométrique entre les pixels d'une image de référence et ceux d'une vue synthétisée par rendu volumétrique. L'entrainement de la fonction permet de synthétiser de nouvelles vues à n'importe quelle position de la scène et selon n'importe quelle orientation de caméra (voir Figure 4). (Mildenhall, 2020; Mildenhall et al., 2020; Tancik et al., 2023; Zhou et al., 2024)

I-5.2 REPRÉSENTATION SOUS FORME DE CHAMP DE RADIANCE NEURONAL

Commençons par décrire concrètement cette fonction définissant le champ de radiance.
Le champ de radiance est défini comme étant la fonction vectorielle à 5 dimensions :

$\boldsymbol{F_\Omega}: (\boldsymbol{x}, \boldsymbol{d}) \rightarrowtail (\boldsymbol{c}, \boldsymbol{\sigma})$	(1)

Elle prend en entrée $\boldsymbol{x}$, la position en coordonnées (x, y, z) et $\boldsymbol{d}$, une direction de vision (θ, ϕ). La sortie est $\mathbf{c}$, la couleur émise (R, G, B) et $\boldsymbol{\sigma}$, la densité volumétrique (densité d'absorption à chaque point de l'espace 3D) (voir Figure 5).

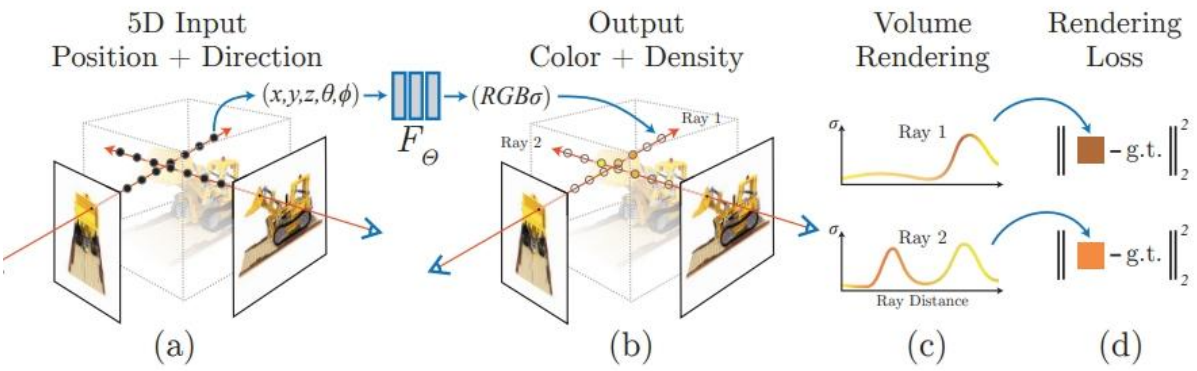


Figure 5 : Processus de représentation de scène par champ de radiance neuronal et procédure de rendu différentiable (Mildenhall, 2020).

Plus concrètement, pour obtenir les entrées de la fonction, des rayons partant de la caméra et traversant la scène sont utilisés pour générer un ensemble échantillonné de points 3D. La position de ces points et leurs directions de visualisation 2D sont utilisées comme entrées dans le réseau neuronal pour produire en sortie des couleurs et des densités. *« Densité volumétrique et couleur dépendante de la vue à tout emplacement continu »* (Mildenhall, 2020).

I-5.3 RENDU VOLUMÉTRIQUE.

À chaque itération, le réseau de neurones représente de mieux en mieux la fonction. Pour effectuer l'apprentissage, deux composantes essentielles sont nécessaires: une méthode d'évaluation (fonction de perte) et une méthode de rétropropagation (descente de gradient sur les poids). Pour l'évaluation, les pixels estimés par synthèse sont comparés aux pixels réels afin de minimiser les erreurs. Les ajustements des paramètres du réseau sont ensuite rétropropagés grâce à la fonction de perte. Pour permettre ces deux étapes, un rendu volumétrique différentiable est mis en place, autorisant la rétropropagation des gradients à travers le rendu.

Mathématiquement, Mildenhall (2020) utilise une méthode de rendu volumétrique classique: il interprète la densité volumétrique $\boldsymbol{\sigma(x)}$ comme la probabilité différentielle qu'un rayon traversant la scène soit interrompu par une particule (infinitésimale) à la position **x.**
De plus, il définit le rayon qui traverse la scène comme:

r(t)=o+td.	(2)

Où le centre de la caméra est défini par **o**, la direction de visée par le vecteur directionnel **d** et **t** est la distance le long du rayon depuis le centre de la caméra jusqu'à une position donnée.
Dès lors, on peut théoriquement estimer le rendu du rayon entre deux bornes $\boldsymbol{t_n}$ et $\boldsymbol{t_f}$ (near and far) pour prédire la couleur $\boldsymbol{C(r)}$ du rayon $\boldsymbol{r(t)}$ en résolvant l'intégrale:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma\big(r(t)\big)c(r(t), d)dt$$	(3)

Avec :

$$T(t) = e^{\left(-\int_{t_n}^{t} \sigma(r(s))ds\right)}$$	(4)

La fonction **T(t)** représente la transmittance accumulée le long du rayon de $\boldsymbol{t_n}$ à $\boldsymbol{t}$, c'est-à-dire la probabilité que le rayon voyage sans heurter de particule.

Le calcul de l'intégrale doit être réalisé par approximation numérique. Mildenhall (2020) utilise la quadrature de Max avec un échantillonnage stratifié (segments fixes mais points aléatoires dans le segment) pour éviter d'impacter la résolution globale.

On obtient la formule:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))c_i \quad (5)$$

Avec la transmittance,

$$T_i = e^{\left(-\sum_{j=1}^{i-1} \sigma_j \delta_i\right)} \quad (6)$$

Et l'inter-distance avec le point précédent,

$$\delta_i = t_{i+1} - t_i \quad (7)$$

Le terme $(1 - \exp(-\sigma_i \delta_i))$ correspond en réalité à α_i (un terme utilisé dans la technique de alpha compositing pour le rendu d'image numérique et qui correspond à l'opacité), ainsi la formule de l'approximation de l'intégrale devient finalement:

$$\hat{C}(r) = \sum_{i=1}^{N} T_i \alpha_i c_i \quad (8)$$

Dans la pratique, on calcule la transmittance plutôt sous la forme d'un produit (inverse de l'opacité des points précédents) (Lalonde, 2025):

$$Ti = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (9)$$

Le rendu volumétrique du rayon peut soit servir à générer une nouvelle vue, soit être utilisé pour optimiser le modèle puisqu'il est différentiable.

Par ailleurs, il est intéressant de remarquer que seule la couleur est affectée par la direction de vue ***d*** c'est-à-dire que la couleur est une radiance puisqu'elle est émise dans une direction.

D'ailleurs, c'est cette intégration partielle (la densité volumétrique n'étant pas fonction de la direction) qui confère au NeRF sa capacité a traiter les surfaces non-lambertiennes. Le propre de ces surfaces est de renvoyer la lumière différemment selon la direction d'observation. Puisque les techniques classiques n'intègrent pas la direction dans leurs prédictions, elles ne peuvent pas différentier si un changement de couleur vient d'une erreur de prédiction ou de la surface. De son côté, le NeRF peut identifier un changement de couleur en intégrant la densité volumétrique par rapport à la couleur et en fonction de la direction. La Figure 6 contient une illustration comparant la vérité du terrain et différents modèles de représentation, on voit que la plaque métallique réfléchissante du train du lego n'est pas du tout rendue dans le modèle qui n'intègre pas la composante directionnelle.

I-5.4 OPTIMISATION D'UN NERF

Grâce aux modèles de rendu volumétrique différentiable, il est désormais possible d'entraîner un MLP pour optimiser les paramètres de la fonction définie comme champ de radiance

neuronal. Cependant, il reste à définir une fonction de perte permettant de faire converger l'optimisation au fil des itérations. Le rôle de cette fonction est de comparer (photométrique de pixel à pixel) les couleurs prédites par le modèle avec les couleurs de référence issues des images réelles. La perte ainsi calculée peut être rétropropagée à travers le processus de rendu, ce qui permet d'ajuster les poids (c'est-à-dire les paramètres de la fonction) du MLP par descente de gradient. Mildenhall et al. (2020) proposent l'implémentation suivante pour l'entraînement: à chaque itération, un échantillon aléatoire de rayons (batch) est sélectionné parmi l'ensemble des pixels des images capturées. En fin d'itération, la fonction de perte calcule simplement la somme des erreurs quadratiques entre la couleur estimée et la couleur de référence pour chaque rayon de l'échantillon tel que:

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \left|\left| \widehat{C_{predicte}}(r) - C_{ref}(r) \right|\right|_2^2 \quad (10)$$

Où $\mathcal{R}$ représente l'ensemble de l'échantillon de rayons.

La perte est ensuite rétropropagée à travers le réseau grâce à un optimiseur (Adam dans ce cas) qui ajuste les poids et contrôle la vitesse de mise à jour (learning rate). Cela permet de réguler le comportement de l'apprentissage. Après un certain nombre d'itérations, les changements des paramètres deviennent de plus en plus faibles; on dit alors que le modèle converge vers un minimum de la fonction de perte, c'est-à-dire que les prédictions du réseau deviennent stables et proches des valeurs réelles. Les paramètres de la fonction décrivant les champs de radiance sont optimisés. La Figure 5 synthétise toutes les étapes décrites ci-dessus.

I-5.5 PREMIÈRES AMÉLIORATIONS DU NERF.

Mildenhall et al. (2020) explicitent deux améliorations qui ont été directement ajoutées aux premiers NeRF.

Tout d'abord, le « positional encoding » qui permet d'injecter de la variation dans les coordonnés brutes d'entrée et augmenter fortement leur dimension. L'effet recherché est de permettre au MLP de mieux rendre les variations de détails de la scène (les variations haute fréquence). La Figure 6 donne un aperçu concret de l'impact de cette amélioration sur le modèle final.

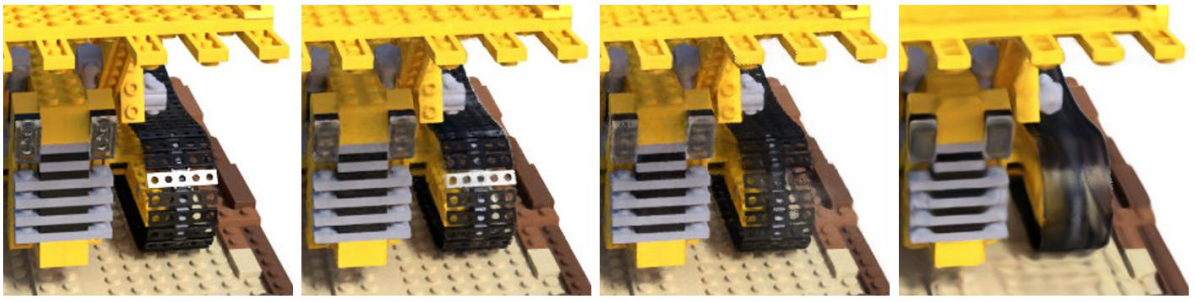


Figure 6 : Comparaison de différents modèles de rendu volumétrique avec la vérité terrain (Mildenhall, 2020)

Ensuite, l'utilisation d'un échantillonnage hiérarchique permet d'optimiser les performances du modèle. L'idée est que les scènes comportent de nombreuses zones vides (translucides) ou occultées. Effectuer trop d'échantillons dans ces zones est un gaspillage de calculs. Pour éviter ce problème, ce type d'échantillonnage a pour but de rediriger les ressources vers les zones dont

la contribution au rendu est significative. Pour cela, il faut optimiser deux réseaux à la place d'un seul : un premier réseau grossier et un second plus fin. Le grossier produit une estimation initiale de la densité le long des rayons pour guider l'échantillonnage plus fin vers les zones significatives. Mathématiquement les localisations des points échantillonnés sont différentes entre les deux modèles comme l'optimisation des poids. Cependant, leur optimisation doit être conduite en simultané pour continuer à guider le fin, dès lors que la fonction de perte a été adaptée pour mesurer l'erreur quadratique des deux réseaux :

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \left[\left\| \widehat{C_c}(r) - C(r) \right\|_2^2 + \left\| \widehat{C_f}(r) - C(\mathrm{r}) \right\|_2^2 \right] \quad (11)$$

Avec $\widehat{C_c}(r)$, la prédiction du volume grossier, $\widehat{C_f}(r)$, la prédiction du volume fin et $C(r)$ la vérité terrain (image de référence). À noter que lorsqu'on utilise le rendu volumétrique pour la visualisation des résultats, on ne rend évidement que le modèle fin.

I-5.6 RÉSULTATS EXPLOITABLES DES NERF

Le champ de radiance neuronal optimisé représente la géométrie de la scène de manière implicite. C'est-à-dire que, contrairement aux géométries explicites, telles que les nuages de points, les voxels et les maillages triangulaires accessibles en parcourant tous les éléments de l'espace de stockage, la géométrie implicite nécessite la sélection de coordonnées spatiales comme entrée pour les points d'échantillonnage. Par conséquent, la manipulation de ces résultats est fortement différente des techniques classiques puisqu'elle doit d'abord passer par un rendu volumétrique ou un tracé de rayon avant de produire un résultat. Néanmoins, cela comporte de nombreux avantages en termes de coût en calculs et d'optimisation. (Zhou et al., 2024)

Par ailleurs, il est tout à fait possible de dériver le NeRF dans les supports de données classiques vus précédemment (cf.I-4). Le nuage de points peut être obtenu en traçant des rayons dans une grille voxel d'interpolation puis en ajoutant les points ayant une densité suffisante. Et de façon traditionnelle, un meshage peut être produit à partir du PCD (voir Figure 7).

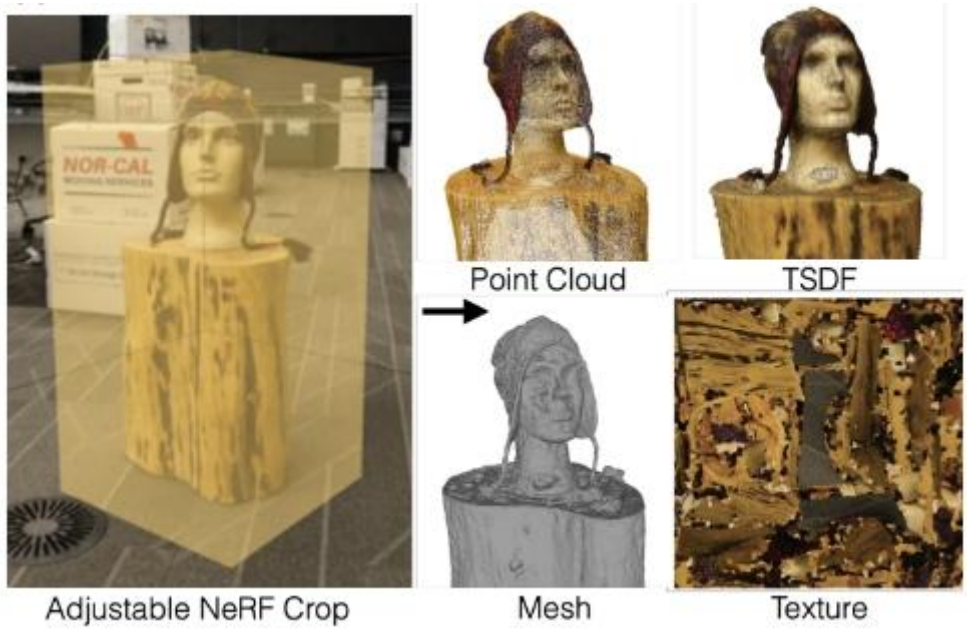


Figure 7 : Support de données classique obtenu à partir d'un NeRF (Tancik et al., 2023).

Les résultats le plus impressionnants restent le rendu de nouvelle vue synthétique partout dans l'espace de la scène. Les NeRF permettent de produire des images de haute qualité même pour des surfaces complexes. Si l'utilisateur fournit un trajet dans la scène, on peut rendre les images successives tout le long du trajet. Associé avec un frame rate et une direction de vue, il est possible de réaliser des vidéos de la scène. Il n'est pas possible de rendre compte de la haute précision des vidéos générées à partir d'un NeRF dans un rapport. Le lecteur est invité a observer ce genre de vidéo dans les documentations Nerfstudio, le site de Mildenhall ou en ouvrant la vidéo « ChapelleLupicin.mp4 » qui est disponible dans le dépôt de ce mémoire. Ce genre d'export a particulièrement intéressé la conservation du patrimoine et les architectes notamment comme outils de communication en combinaison avec la réalité augmentée. (Zachos & Anagnostopoulos, 2024).

I-5.7 VARIÉTÉ DES ALGORITHMES DE NERF

De très nombreux travaux ont suivi l'introduction des NeRF. La plupart s'orientent vers le développement et l'optimisation de la technique originale. D'autres explorent les nombreuses applications qui ont été rendues possibles. Pour rester concis, « seulement » cinq modèles de NeRF seront présentés car ils présentent un intérêt dans le cadre du mémoire. Nous présenterons également le Framework Nerfstudio, le premier grand pas en avant en termes d'accessibilité de la technique.

Mip-NeRF est un modèle établi par Barron & al. (2021) pour mieux gérer les images floues ou les détails à différentes échelles. Au lieu de traiter chaque rayon comme un simple point, Mip-NeRF représente chaque rayon comme un cône (« cone frustrum ») intégrant une zone de l'espace. Cela permet de prévenir l'aliasing et d'avoir un rendu plus net quand les pixels correspondent à des zones de la scène qui couvrent plusieurs surfaces.

Instant Nerf est un modèle établi par Müller (2022). Un travail prononcé sur la table de hachage, l'utilisation d'un encodage multi-résolution et la compaction du réseau neuronal ont permis une très forte réduction du temps d'entrainement en passant de quelques jours à des rendus quasi-instantanés.

Semantic-nerf est un modèle établi par Zhi (2021) qui permet d'introduire de l'information semantique dans le NeRF et donc de faciliter la segmentation de plans et de formes par propagation.

LeRF (Language Embedded Radiance Fields) est un modèle établi par Kerr (2023) qui permet d'interroger le rendu du NeRF à partir de requêtes en langage commun. La manipulation de la scène étant largement facilitée pour des utilisateurs non-initiés.

Un apport très significatif à la recherche a été apporté avec l'arrivé du framework Nerfstudio et de son modèle Nerfacto. Nerfstudio est un projet qui vise à harmoniser la recherche en fournissant un cadre de développement général. Il vient avec une interface de commande (CLI) et un viewer en temps réel. La combinaisons des deux permettant de superviser l'entièreté de la procédure concernant la production de NeRF et le développement de nouveaux modèles. Le Pipeline de traitement Nerfstudio permet d'utiliser des sources de données variées et de produire des exports dans les principaux format 3D (voir Figure 8) (Tancik et al., 2023).

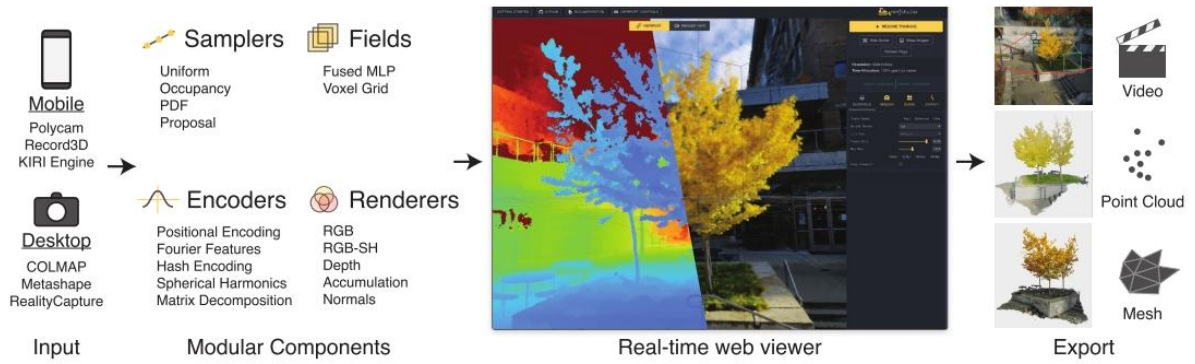


Figure 8 : Aperçu du Framework complet de Nerfstudio (de l'entrée à la sortie) (Tancik et al., 2023).

L'autre principal apport de Tancik (2023) vient du modèle Nerfacto qui combine les résultats de mip-nerf et de instant-ngp pour obtenir un excellent compromis entre vitesse et précision. Il permet de générer des rendus de qualité dans un très grand nombre de cas (Remondino et al., 2023).

I-6 COMPARAISON ENTRE NERF ET LES AUTRES TECHNIQUES.

Cet état de l'art a d'abord établi l'importance des relevés architecturaux de façades et exposé les principales techniques utilisées actuellement. A suivi la présentation d'une potentielle nouvelle technique concourante, les NeRF. Les applications de cette technique ont déjà commencé à être investiguées surtout dans le domaine de l'architecture et de la conservation du patrimoine. Généralement l'objectif est de comparer les différentes techniques entre elles pour définir leur utilité en fonction des cas d'application.

Certains résultats intéressants ont été soulignés au niveau de la portabilité des techniques NeRF. Différentes études ont déjà montré une différence flagrante de précision entre les reconstructions directes par TLS et la photogrammétrie à partir d'applications mobiles comme Pix4d et scanniverse. Dans la même idée, L'étude de Farhat (2017) montre que générer des MESH directement à partir d'applications de scan telles polycam ou pix4Dcatch offre bien moins de précision que par neuralangelo (un modèle puissant de NeRF). L'étude, insiste sur le fait que ce n'est pas le cas pour les nuages de points.

Concernant la comparaison pure des techniques, les études de Croce (2024) et de Remondino (2023) sont pionnières sur le sujet. Comme souvent, le TLS est utilisé comme la vérité terrain servant de base à la comparaison de techniques moins précises (ici les NeRF et la photogrammétrie). Croce souligne de nombreuses conclusions. Tout d'abord le workflow du NeRF est très proche de celui de la photogrammétrie classique (voir Figure 9). En effet, les données en entrée sont sensiblement les mêmes, des images en deux dimensions se superposant. Lorsque les images n'ont pas de coordonnées, il est nécessaire de procéder à un alignement comme en photogrammétrie classique pour retrouver les coordonnées et l'orientation des caméras (paramètres extrinsèques).

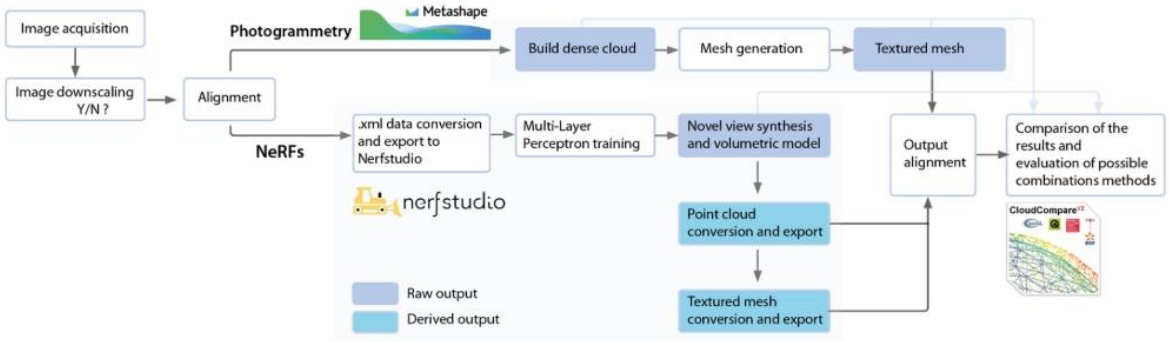
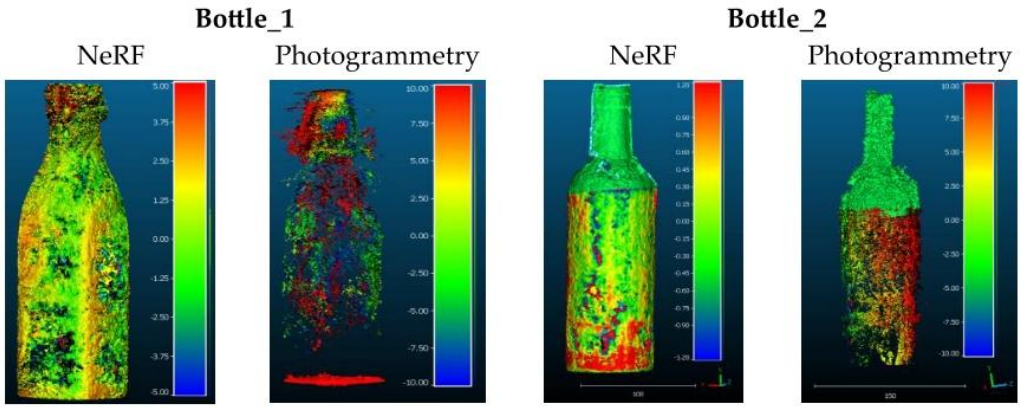


Figure 4. Overview of the proposed methodology.

Figure 9 : Comparaison du workflow des NeRF et de la photogrammétrie.(Croce et al., 2024).

Ensuite, il y a quand même des différences fondamentales: par rapport à la photogrammétrie, le NeRF permet de mieux représenter l'apparence des matériaux selon les points de vues de l'utilisateur, notamment pour les surfaces non-lambertiennes (transparentes ou réfléchissantes) (Croce et al., 2024). L'exemple d'une bouteille en verre chez Remondino (2023) est particulièrement parlant (voir Color-coded cloud-to-cloud comparisons for both Instant-NGP and Photogrammetry on the two transparent objects [unit:mm]

Figure 10).



Color-coded cloud-to-cloud comparisons for both Instant-NGP and Photogrammetry on the two transparent objects [unit:mm]

Figure 10 : Comparaison de la photogrammétrie et du NeRF pour l'acquisition de bouteilles (Remondino et al., 2023)

La représentation 3D du NeRF est un modèle volumétrique continu tandis que le nuage dense de points est discret. Cela a pour conséquence de rendre le système NeRF beaucoup plus fermé que la photogrammétrie et de rendre les conversions de format et les exports beaucoup plus complexes. Le nombre d'images et leurs résolutions est un facteur limitant pour les deux techniques : la photogrammétrie perd en efficacité avec la diminution de la qualité et du nombre d'images alors que c'est l'inverse pour le NeRF qui gagne en temps de traitement. A contrario, le NeRF présente des difficultés à très haute résolution. La photogrammétrie reste plus précise dans la plupart des cas et surtout celui des larges surfaces mattes.

Au-delà de ces comparaisons qualitatives, plusieurs études récentes se sont attachées à quantifier la précision réelle des NeRF. Remondino et al. (2023) montrent que, dans des conditions idéales sur des objets bien texturés, la photogrammétrie reste légèrement plus performante avec des écarts quadratiques moyens inférieurs au dixième de millimètre, tandis que les NeRF se situent autour de 0,1 à 0,2 mm. Cependant, dès que les surfaces deviennent complexes, comme le verre ou le métal réfléchissant, les NeRF surpassent la photogrammétrie en réduisant les erreurs à un ordre millimétrique, quand cette dernière atteint plusieurs millimètres. Croce et al. (2024), dans un contexte patrimonial appliqué, confirment cette tendance : sur une statue en marbre, un lutrin en bronze et une tour médiévale, les NeRF atteignent une précision généralement décimétrique, voire millimétrique dans des conditions favorables, et conservent une robustesse intéressante même avec moins d'images ou des résolutions plus faibles. La photogrammétrie, si elle reste la référence pour les détails architecturaux fins, perd nettement en qualité dans ces conditions.

La comparaison met en avant la rapidité et la capacité des NeRF à représenter des scènes restreintes avec des matériaux complexes. Elle montre également que la précision des NeRF est la plupart du temps légèrement en dessous de celle de la photogrammétrie (et donc du LiDAR). Les études concluent cependant que la technique serait intéressante si on la combinait avec d'autres, notamment pour la rapidité du visuel et la texturisation. (Croce et al., 2024; Remondino et al., 2023)

CHAPITRE II OBJECTIFS ET HYPOTHÈSES DE RECHERCHE

Le chapitre I met en évidence l'importance et l'utilité des relevés de façades en architecture. Il souligne aussi l'arrivée des NeRF comme une nouvelle technologie prometteuse dans la reconstruction 3D. Si les NeRF permettent de produire des PCD (Point Cloud Data) comparables à ceux produits en photogrammétrie (même avec des inputs de moindre qualité), alors il devient intéressant d'étudier les nouvelles possibilités offertes par ceux-ci (Croce et al., 2024). Certes le NeRF est une technologie particulièrement récente et encore en cours de développement mais la littérature semble indiquer qu'elle produit déjà des résultats exploitables (cf. I-5). Si sa précision est clairement une faiblesse, elle est contrebalancée par le temps de traitement fortement réduit, la reconstruction de textures habituellement complexes (réfléchissante, homogène ou lisse) et le peu de matériel nécessaire à sa mise en place.
Dans l'état actuel de développement des NeRF, la balance de leurs avantages et inconvénients les destine à la reconstruction dans des contextes où l'acquisition des données est restreinte, rendant les autres techniques trop couteuses ou difficiles à mettre en place (cf. I-5). Les interventions rapides lors de situations d'urgence (contrainte de temps), le nombre restreint d'images (contrainte de résolution) (Croce et al., 2024), le suivi au jour le jour (contrainte de résolution temporelle) et surtout le relevé effectué par un opérateur peu formé (contrainte de qualité d'acquisition) sont des contextes pratiques d'utilisations sur lesquels se concentre ce mémoire. Le dernier contexte étant particulièrement intéressant puisque qu'il revient à rendre accessible la reconstruction 3D et les relevés à un public bien plus large dans des contextes plus variés.

La création d'un workflow complet de relevés par vue orthographique qu'il soit simplement complémentaire aux techniques classiques ou réellement rendu accessible à un public de non-initiés, n'a pas encore été tentée. Ce défi constitue l'objectif de ce mémoire.

En d'autres termes, ce travail cherche à répondre à la question suivante:

Les avantages procurés par un NeRF permettent-ils de rendre plus accessibles les relevés de façades architecturales, surtout pour des utilisateurs non-initiés ?

Afin d'y répondre, nous pouvons poser deux hypothèses:

- Hypothèse 1 (H1): Il est possible d'obtenir des vues orthographiques d'une façade à partir de champs de radiances neuronaux.
- Hypothèse 2 (H2): La méthodologie (workflow) permettant d'obtenir des vues orthographiques est accessible à un public non-initié aux techniques de pointe et peut leur proposer une solution alternative facile et rapide.

Discuter de la validité de ces hypothèses devrait permettre d'établir un workflow convainquant et surtout d'évaluer s'il est applicable et pertinent, que ce soit par des experts du domaine ou des professionnels non-initiés.

CHAPITRE III MÉTHODOLOGIE

Ce troisième chapitre expose la méthodologie mise en place pour répondre aux différentes hypothèses de recherche. Elle est schématisée par la Figure 11 et détaillée ci-dessous.

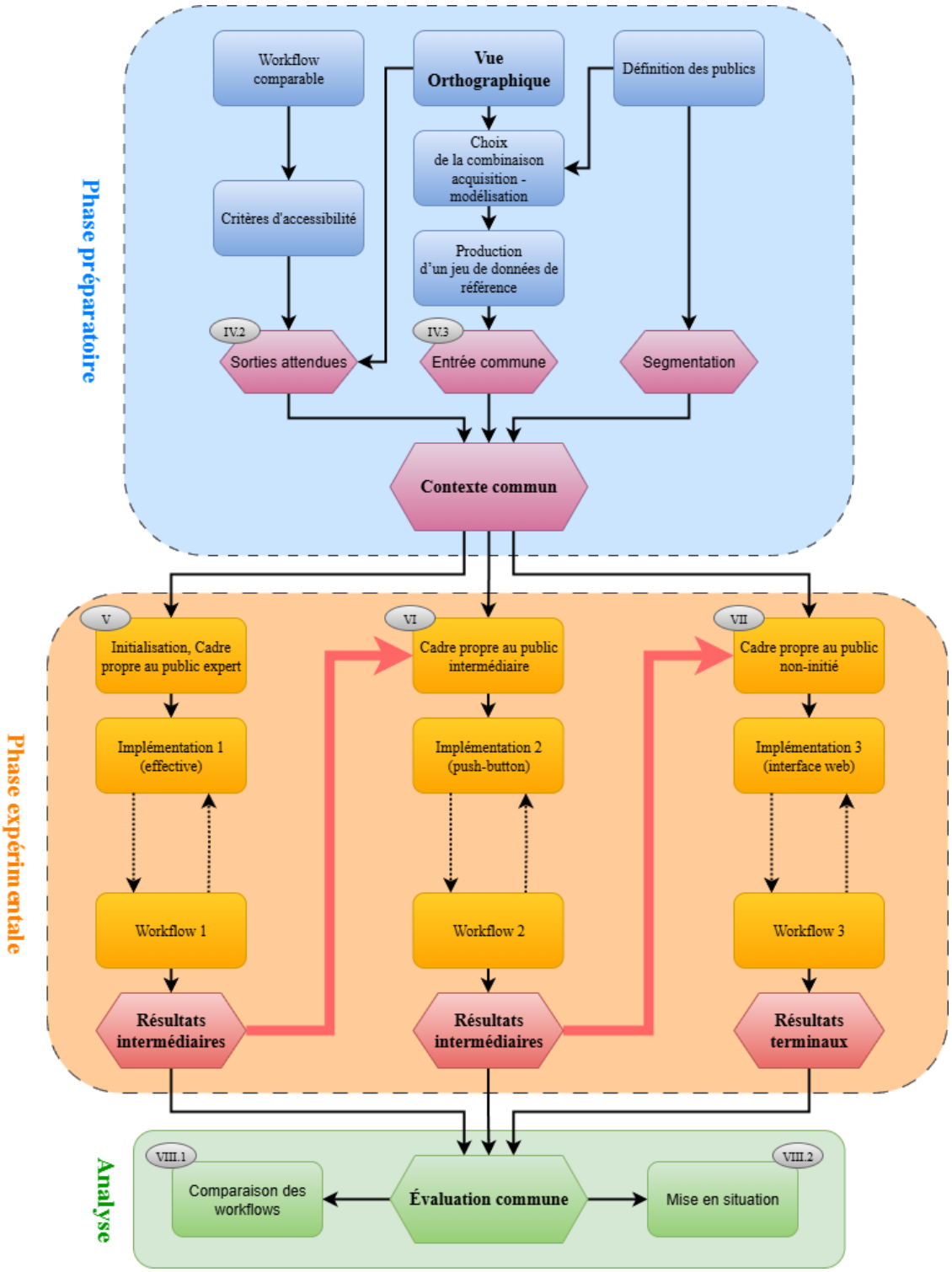


Figure 11 : Diagramme illustrant la méthodologie générale.

L'approche consiste d'abord à élaborer une version complète et potentiellement complexe d'un workflow intégrant des NeRF, puis à le simplifier progressivement. Pour éviter de perdre des résultats intermédiaires potentiellement utiles à un public plus averti, la mise en œuvre n'est pas linéaire: elle est segmentée en trois expériences successives et dont les résultats sont comparables, chacune visant un public différent, du plus expert au non initié. Chaque expérience part des mêmes données et objectif, puis fait l'objet d'une mise en œuvre (implémentation), d'une mise en situation et d'une analyse quantitative et qualitative pour discuter les deux hypothèses.

Il est important de préciser que les expériences sont successives pour conserver l'idée de simplification progressive. Chaque expérience se base sur les résultats de la précédente avant de les simplifier pour un public de plus en plus large. Les deux premiers groupes d'utilisateurs (ou publics) sont capables de valoriser les résultats intermédiaires grâce à leur niveau plus élevé de connaissances théoriques ou techniques. Une méthodologie linéaire constituée d'une seule expérience avec des simplifications en chaine n'aurait pas permis de conserver une quantité non négligeable de résultats intermédiaires significatifs.

En outre, les expériences ont un apport individuel, en parallèle les unes des autres. Cependant, elles s'inscrivent dans un contexte commun d'expérimentation leur permettant d'être comparées. Elle partent des mêmes conditions initiales, l'acquisition d'un jeu de données de référence et elles visent le même objectif, la production d'un workflow permettant la génération d'une vue orthographique. En d'autres termes, les expériences essayent toutes de valider la première hypothèse mais leur implémentation et donc leur workflow pour y arriver différent. La validation de la deuxième hypothèse implique celle de la première.

En premier, la phase préparatoire (cf. Chapitre IV) décrit les étapes de mise en place nécessaires au bon déroulement de la phase expérimentale.

- Un contexte de départ commun est défini, il se présente sous la forme d'un jeu de données de référence (entrée commune). Il est utilisé dans toutes les expériences et va permettre l'évaluation commune des résultats.
- Les choix de méthodes d'acquisition et la production du jeu de données sont pensés en amont pour la production de vues orthographiques par tous les publics.
- Une base comparative est mise en place. Elle est constituée de la vue orthographique, du jeu de données de référence et des workflows qui ont mené à la production de cette vue. Les workflows sont comparés sur base de leur accessibilité et une liste de critères est établie.
- Des publics sont définis, ce qui permet de segmenter les expériences en délimitant leur cadres et leur objectifs.

Ensuite, la phase expérimentale constitue le corps de l'étude. Elle est découpée en trois expériences suivant toutes le même schéma: à partir d'un cadre propre et du jeu de données de référence, une implémentation répondant aux besoins est mise en place (avec des ajustements si nécessaire) et les résultats sont produits.

- L'expérience 1 (cf. Chapitre V) sert d'initialisation. Elle vise un public d'experts et une implémentation effective produisant une vue orthographique.
- L'expérience 2 (cf. Chapitre VI) prend en compte les résultats intermédiaires de l'expérience 1 pour ajuster son cadre propre en identifiant des axes de simplification. Elle vise un public intermédiaire et une implémentation d'interface simplifiée de type « push-button ».

- L’expérience 3 (cf. Chapitre VII) prend en compte les résultats intermédiaires de l’expérience 2 (et donc de l’expérience 1), ici aussi, pour ajuster son cadre propre avec de nouveaux éléments de simplification. Elle vise un public large de non-initiés et une implémentation web.

Les résultats de l’expérience 3 sont dits terminaux et viendront s’ajouter aux résultats des deux autres expériences lors d’une évaluation commune.

En troisième et dernière partie, l’analyse (cf. Chapitre VIII) reprend les résultats des expériences et leurs produits pour les comparer et évaluer leur praticité (sur base des critères d’accessibilité établis) (cf. VIII-1). Elle comporte également une mise en situation en vue d’étayer les workflows finaux dans des cas plus complexes (cf. VIII-2). La mise en situation permet de discuter l’hypothèse 1 et la comparaison de discuter l’hypothèse 2.

En conclusion du mémoire, l’analyse est discutée (cf. Chapitre IX) pour répondre à la question de recherche et ouvrir sur des perspectives de développements futurs.

CHAPITRE IV PRÉPARATION ET SEGMENTATION

Ce quatrième chapitre passe en revue tout ce qui a été préparé en amont de la phase expérimentale. La préparation doit être rigoureuse et les limites de l'environnement strictes pour que les résultats des trois expériences puissent être comparés entre eux. Un contexte commun est donc défini et assure que les différentes expériences partent du même point de départ (entrée commune) et arrivent au même point d'arrivée (sorties attendues). Dans ce but, la phase préparatoire comprend la production d'un jeu de données de référence à partir d'une acquisition pensée pour la production de vues orthographiques et la définition des sorties attendues dont notamment des workflows comparables sur base de critères d'accessibilité. Pour permettre la segmentation des expériences, les publics sont également définis.

La définition du public large de non-initiés et de ses besoins influence également le choix de la méthode d'acquisition. De ce fait, la définition des publics est présentée dès la première section du chapitre.

IV-1 DÉFINITION DES PUBLICS

La production de relevés de façades est un domaine se situant à la croisée des chemins entre l'informatique, la géomatique, l'architecture et la conservation du patrimoine. Dès lors, il est difficile d'identifier tous les publics pouvant recourir à cette technique. Il est beaucoup plus simple de les classer sur base de leurs niveaux de connaissances.

Le premier public regroupe les utilisateurs ayant le plus haut niveau de connaissances théoriques et techniques. Ce public comprend les mécanismes de la technique et est capable de les mettre en œuvre à chaque étape. Ceci nécessite également de connaitre les bonnes pratiques d'acquisition mais aussi d'avoir de solides bases en informatique et en algorithmique pour appréhender et optimiser les paramètres liés au NeRF. Ces utilisateurs ont aussi une bonne connaissance des logiciels SIG (Système d'Information Géographique) permettant d'exploiter les nuages de points. Ce groupe d'utilisateur inclut les géomètres, les ingénieurs, les architectes et archéologues spécialisés dans la reconstruction 3D, … Le public répondra à l'appellation « public d'experts ».

Le second public regroupe les utilisateurs ayant un bon niveau de connaissances mais uniquement théoriques. Ce public comprend les mécanismes de la technique mais n'a pas forcément la formation en informatique ou en SIG nécessaire à sa mise en œuvre. Ce groupe représente les architectes et archéologues et les opérateurs du bâtiment initiés aux bases du NeRF et des SIG. Le public représente un entre deux avant un public plus large. Le public répondra à l'appellation « public intermédiaire ».

Le dernier public regroupe les utilisateurs ayant peu ou pas de connaissances théoriques et techniques. Ce public a simplement besoin des résultats et voit uniquement les avantages et inconvénients d'une nouvelle technique émergente en comparaison avec les techniques qu'il utilise couramment, le NeRF étant un outil supplémentaire. Ce public est composé des

architectes et archéologues non-spécialistes, de différents corps de métiers du bâtiment ou de tout autre intervenant, … Le public répondra à l'appellation « public de non-initiés ».

IV-2 CHOIX DE LA SORTIE COMMUNE ATTENDUE

Si les expériences sont différentes, il est impératif d'avoir des éléments pour les comparer. Un élément fixe : une vue orthographique d'une façade, et un élément variable : le workflow pour l'obtenir. La vue orthographique représente un produit fini, exploitable, tangible et de qualité constante et le workflow sert d'étalon pour évaluer les différences d'accessibilité des implémentations. Ces dernières étant chacune liée à un public en fonction de sa facilité d'accès. Cette section définit les différents éléments qui représentent les sorties attendues et les résultats validant une implémentation et permettant de terminer une expérience.

IV-2.1 VUE ORTHOGRAPHIQUE DE FAÇADE

Comme vu précédemment (cf. I-4), la vue orthographique est un produit très courant pouvant être obtenus à partir d'un relevé de façade. Celle-ci a été choisie comme produit final car son obtention permet de parcourir toutes les étapes depuis l'acquisition, en passant par la modélisation, l'export et les post traitements (cf. I-3.2). Elle devrait donner un aperçu du workflow nécessaire, ses avantages et ses inconvénients la distinguent des techniques couramment utilisées. Pour donner une définition stricte et rigoureuse d'une vue orthographique, nous nous référons à celle donnée par le dictionnaire SIG de ESRI (2025): « En analyse 3D, perspective qui permet d'afficher des données dans une scène en tant que plan bidimensionnel vu du dessus. La perspective ne pouvant pas être rapprochée dans une vue orthographique, l'échelle est constante sur la totalité de l'affichage. ».

D'un point de vue plus pratique, une vue orthographique est un cas spécifique de perspective. La perspective étant une représentation à deux dimensions d'un sujet à trois dimensions. Cela fait partie d'un langage structuré visant à communiquer de l'information sur le sujet. La manière d'effectuer la projection (mise en perspective) communique l'information différemment (vue oblique, à un ou deux points de perspectives, …). Dans le cas de la vue orthographique, elle les dimensions sont conservée en utilisant une ligne de projection perpendiculaire au plan de projection, une face de l'objet étant parallèle à ce plan. Les lignes de fuite deviennent parallèles comme si l'observateur était placé à une distance infinie de l'objet. (Gill, 2004).

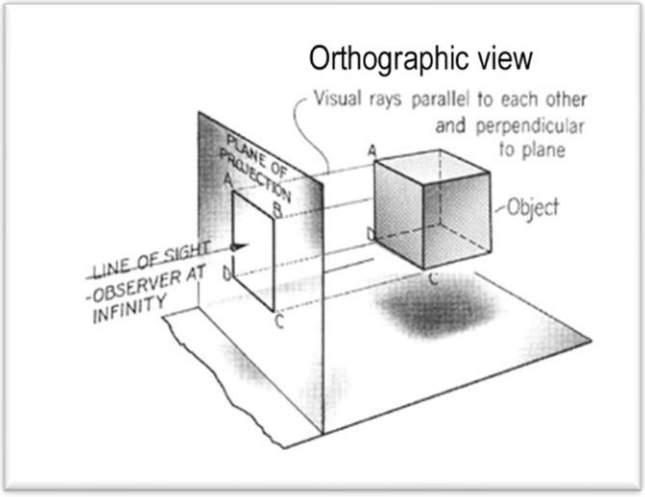


Figure 12 : Schéma utilisé en dessin technique décrivant la projection d'une vue orthographique (Gill, 2004)

Dans le cadre de ce mémoire, la projection orthographique est définie de façon générique comme la transformation d'un modèle ou d'un objet tridimensionnel en une représentation bidimensionnelle qui conserve les mesures de distances et de dimensions.

La théorie mathématique et les bibliothèques informatiques étant variées, les techniques pratiques sont explicitées pour chaque workflow dans la partie résultats.

Enfin, pour voir les produits concrets attendus, nous pouvons reprendre l'exemple de l'étude archéologique du Palazzo Ancarano (Villa et al., 2017) (cf. I-4). La Figure 3 présente des vues orthographiques relevant les 4 façades du bâtiment d'intérêt sont présentées. L'image est dans un format compressible et portable (.png) et dispose d'une échelle visible permettant de reporter des mesures sur un dessin technique.

IV-2.2 WORKFLOWS

Les workflows représentent le centre d'intérêt principal de ce mémoire. Ce sont ces éléments qui peuvent être comparés et ainsi permettre d'effectuer l'analyse des résultats (validant ou non les hypothèses de recherche).

IV-2.2.1 Terminologie

Il existe une certaine confusion autour des termes workflow, méthodologie et framework. Dans le cadre de ce mémoire, un workflow est considéré comme, une suite d'étapes à suivre (dans l'ordre) afin d'effectuer correctement un travail donné. Le mot « méthodologie » est utilisé dans son sens scientifique comme une description des procédés et méthodes scientifiques mis en œuvre par un chercheur pour répondre à ses hypothèses et déduire sa conclusion. Il aurait également pu être utilisé pour décrire la suite d'étapes à suivre mais cela aurait probablement introduit une confusion avec la sémantique de rédaction scientifique déjà utilisée. Le framework est assez proche mais désigne plutôt un cadre général avec des outils et des règles pour effectuer la tâche Il n'est pas forcément séquentiel. Il sert de base au développement de workflows. En soi, il correspond plus à ce que la méthodologie appelle «contexte commun» sans le cas particulier des sites d'études.

Le terme « workflow » est donc conservé pour insister sur la notion séquentielle. Il faut noter qu'il est choisi de garder le terme original venant de l'anglais « workflow » plutôt que sa traduction française « flux de travail » qui est beaucoup moins utilisée dans le monde professionnel.

IV-2.2.2 Définition

Un workflow est un processus répétable consistant en une série de tâches qui doivent être complétées en suivant une séquence spécifique. Il permet d'assurer que le processus soit correctement effectué à chaque fois. Cette approche accorde l'opportunité d'éliminer les tâches redondantes, de réduire le coût opérationnel et de répondre rapidement aux problèmes. (Lucidchart, 2025)

Le workflow peut s'écrire sous forme de document ou de guide textuel mais il est souvent plus facile de le comprendre quand il est résumé par un diagramme visuel. Dans l'exemple de l'étude du Palazzo Ancarano, un workflow graphique a été effectué pour synthétiser les grandes étapes en une image (voir Figure 13). S'il est plus facile de se représenter un workflow graphiquement, les comparer entre eux l'est beaucoup moins.

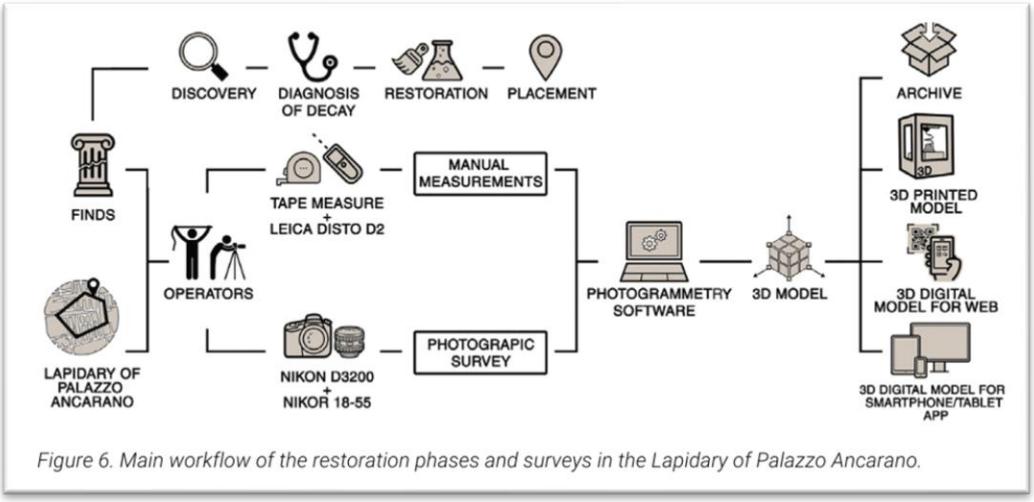


Figure 13 : Workflow graphique d'une étude archéologique passant par la photogrammétrie (Villa et al., 2017)

IV-2.2.3 Critères d'accessibilité

Pour pouvoir considérer les workflows comme des résultats d'expérience, ils doivent être comparables objectivement. La question de recherche portant en partie sur l'accessibilité au public de non-initiés, des critères de comparaison ont été établis pour évaluer cette accessibilité. Chaque étape du workflow sera évaluée en fonction de ces différents critères. Le nombre d'étapes étant accessibles servant de base de comparaison entre les expériences. Voici la liste des critères retenus:

- Compréhension des NeRF: est-ce que la réalisation de l'étape dépend de la compréhension des principes menant au NeRF. L'incompréhension pouvant induire des erreurs, voir rendre totalement inaccessible une étape.
- Compréhension des systèmes d'information géographique (SIG ou GIS): est-ce que la réalisation de l'étape dépend de la compréhension des systèmes SIG 3D ou de la reconstruction 3D ?
- Paramétrisation: est-ce que l'étape nécessite d'être paramétrée ? La liberté de paramétrisation peut avoir des avantages mais peut aussi perdre l'utilisateur et induire des erreurs. Une étape non préconfigurée pouvant limiter fortement l'accessibilité.
- Possibilité de fausses manipulations: est-ce que l'étape peut induire des erreurs à la suite de fausse(s) manipulation(s) venant de l'utilisateur ? Est-ce que l'étape possède des sécurités pour empêcher l'utilisateur de commettre de fausse(s) manipulation(s) ?
- Feedback: est-ce que la manipulation effectuée durant l'étape renvoie une notification textuelle ou visuelle, avant ou après sa réalisation ? Les feedbacks permettent à l'utilisateur de prévisualiser les résultats et évitent les erreurs. Les feedbacks sont souvent de bons guides visuels également.

Les critères d'accessibilité sont inversement proportionnels, c'est-à-dire que plus l'étape rencontre de critères, moins elle est accessible. Par ailleurs, on ajoutera également aux critères d'accessibilité le temps de réalisation de chaque étape et le temps total puisque c'est un critère assez simple mais très intéressant à évaluer.

Les workflows rejoignent leurs produits finaux (la vue orthographique) dans la partie analyse.

IV-3 CHOIX DE L'ENTRÉE COMMUNE

Cette section justifie les choix faits pour produire l'entrée qui sera commune à toutes les implémentations de la phase expérimentale. Se restreindre à une seule entrée signifie se restreindre à une seule technique d'acquisition. Chaque étape d'acquisition pourrait être personnalisée en fonction des capacités des publics cibles mais cela impliquerait des variations en termes de précision sur les vues orthographiques. De plus, avoir des acquisitions plus qualitatives (meilleure précision, géoréférencée, …) aurait empiété sur les techniques classiques comme la photogrammétrie et comme nous l'avons vu, le mémoire cherche surtout l'utilité des NeRF dans de nouvelles situations où les techniques classiques sont moins adaptées. Pour toutes ces raisons, l'entrée restera fixe pour tous les publics. Ceci implique que le mode d'acquisition soit adapté au public de non-initiés. Les choix de techniques d'acquisition, de référentiel, le nombre d'étapes et la production du jeu de référence sont donc fortement limités puisque contraints à s'adapter au public le moins connaisseur.

En outre, l'entrée commune est réfléchie pour permettre la production de vues orthographiques en passant par une modélisation par NeRF. Ceci implique plusieurs choix et conditions sur le référentiel mais aussi sur la technique d'acquisition et son couplage avec la modélisation par NeRF. Ces choix forment un cadre d'acquisition permettant de préparer la première campagne de données (matériel, site d'études). Le site principal peut ensuite être relevé pour produire un jeu de données de référence commun à toute la phase expérimentale. Les sites secondaires servent durant la mise en situation après la phase expérimentale. Au final, l'entrée commune est constituée d'un jeu de données et du workflow pour l'obtenir.

IV-3.1 CHOIX DU RÉFÉRENTIEL.

Lors d'un relevé, l'opérateur choisit toujours (consciemment ou non) de se placer dans un référentiel. C'est un choix important qui doit prendre en compte les limites de notre acquisition et du public.

Avant d'aller plus loin, il est important d'également effectuer un rappel de terminologie. Deux grands types de référentiels sont définis. En premier lieu, le référentiel relatif qui est un système de coordonnées propres à une représentation. Il rend compte seulement de la géométrie des objets les uns par rapport aux autres à l'intérieur de la scène. Ensuite, le référentiel absolu qui utilise un système de coordonnées se basant sur une unité métrique. Il permet de rendre les mesures reproductibles et comparables entre les expériences. Les systèmes de coordonnées de référence (SCR) sont un cas particulier de référentiel absolu qui s'appuient sur un ensemble de références et de lois mathématiques pour placer les coordonnées de façon cohérentes par rapport à la terre. Passer d'un référentiel relatif à un référentiel absolu s'appelle une mise à l'échelle tandis qu'aligner un référentiel absolu local pour coïncider avec un SCR s'appelle le géoréférencement (référentiel terrestre).

Les termes étant posés, il est plus facile de discuter des choix effectués dans la gestion des référentiels. De manière générale, il est possible d'effectuer des reconstructions 3D en restant uniquement dans un référentiel relatif. Cela est suffisant pour beaucoup d'applications (Visualisation, réalité virtuelle, impression 3D,…). Cependant en architecture ou en topographie, la mise à l'échelle est considérée comme primordiale. Dans le cas présent, une vue orthographique sans échelle ne permet d'effectuer aucune mesure de distance ou de surface, seulement de comparer l'agencement des éléments de la façade entre eux. Le passage d'un référentiel relatif à un référentiel absolu constituera donc un objectif minimal.

Les techniques ne sont pas toutes égales pour effectuer une mise à l'échelle. Le LiDAR et la station totale, par essence, effectuent des mesures de distances absolues et donc produisent des nuages très précis et déjà à l'échelle. À l'opposé, la photogrammétrie ne peut injecter l'information d'échelle qu'indirectement et a posteriori, soit en renseignant les coordonnées des caméras (enregistrées par un drone par exemple) avant la reconstruction, soit en retrouvant les paramètres extrinsèques des caméras pour procéder à leur alignement. L'échelle étant alors injectée avant la création du nuage dense.

En ce qui concerne le géoréférencement, peu importe la technique d'acquisition il nécessite de lier le référentiel absolu local à un SCR. Cela passe souvent par la prise d'une mesure GNSS ou la connexion a des points connus. Malheureusement, ces prises de mesures nécessitent une bonne précision, et l'utilisation de méthodes trop lourdes en temps, en matériel et en connaissances pour être correctement réalisées par le public de non-initiés.

Finalement, et pour respecter un cadre commun incluant tous les publics, l'acquisition sera effectuée dans un référentiel absolu local. Ceci permettant d'effectuer des mesures dans les vues orthographiques sans devoir mettre en place un géoréférencement. Encore une fois, la réflexion suivante peut être faite : s'il est possible de mettre en place un géoréférencement, il est aussi possible de mettre en place une technique plus précise que le NeRF (à noter quand même le cas des surfaces lisses et réfléchissantes).

IV-3.2 COMBINAISON ACQUISITION/MODÉLISATION.

La méthode d'acquisition est foncièrement liée aux méthodes de modélisation et aux applications qui lui succèderont. De façon contre-intuitive, pour choisir la méthode d'acquisition, il faut d'abord choisir une méthode de modélisation. Même si la technique est jeune il existe déjà une offre variable pour la modélisation par NeRF. On citera principalement Nerfstudio, Luma AI capture et PyTorch/PyTorch3D.

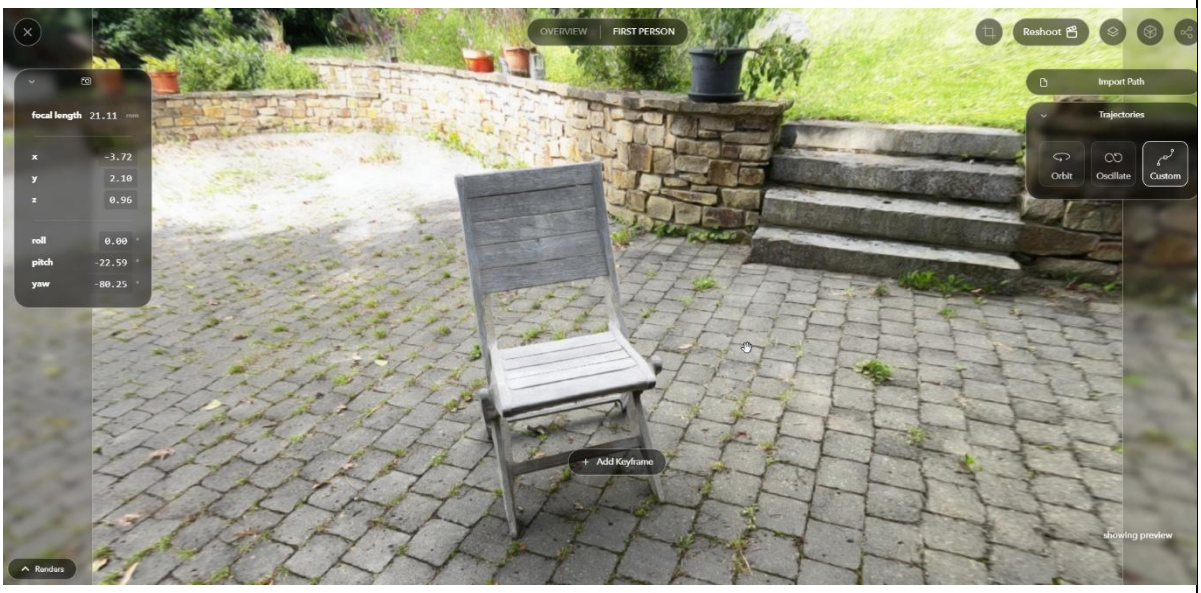
IV-3.2.1 Choix de la méthode de modélisation

PyTorch est une librairie Python permettant de réaliser, de créer et d'entrainer des réseaux, c'est donc l'option la plus « bas niveau » des trois. Elle permet une liberté maximale à condition de construire absolument tous les composants depuis la mise en forme des données jusqu'à l'export des résultats en passant par le réseau, l'entrainement, la visualisation, les post traitements, etc. Il serait donc tout à fait possible d'utiliser cette librairie pour la modélisation mais, recréer un algorithme NeRF performant puis le comparer avec d'autres existants nécessite une quantité extensive de temps. De plus, cela équivaudrait à s'enfoncer dans le backend alors que la question de recherche s'oriente plus vers du frontend. C'est pourquoi il a rapidement été choisi de se concentrer sur les deux autres techniques plus « haut niveau » qui intègrent déjà des pipelines solides, faisant au passage gagner beaucoup de temps.

Deux captures/modélisations ont été réalisées pour évaluer la facilité de prise en main de la modélisation entre Luma AI et Nerfstudio. Une chaise de jardin a été scannée puis reconstruite. La combinaison de la modélisation et de l'acquisition est faite avec Luma AI capture/Luma AI et Polycam/Nerfstudio. Les deux captures sont réalisées avec le même appareil (iPhone 13 pro).

La Figure 14 reprend les résultats visuels dans l'éditeur de Luma AI (1) et Nerfstudio (2).

1.



2.

Figure 14 : Visualisation de la modélisation dans l'éditeur de (1) Luma AI et (2) Nerfstudio (custom).

Nerfstudio est de plus haut niveau que PyTorch mais nécessite quand même une certaine connaissance en informatique pour son installation et la gestion par ligne de commande (CLI). Son interface comprend un grand nombre d'options pour la création de rendu vidéo, de PCD, de MESH. Nerfstudio n'est pas lié à une application de capture en particulier et accepte plusieurs sources de données.

Luma AI est plus simple d'utilisation (plus haut niveau, interface fluide et très simplifiée, visuel à chaque étape) et a sa propre application de capture. Après la capture, Luma AI produit le modèle dans son cloud et vous renvoie les résultats en indiquant une notification après les traitements. L'utilisateur peut uniquement personnaliser la trajectoire du rendu et télécharger le PCD, les MESH, ou la vidéo 360.

Finalement, c'est Nerfstudio qui a été retenu. En effet, si Luma AI semble plus indiqué pour un public de non-initiés, il cloisonne beaucoup trop son entrée et sa sortie. Il ne permet pas de

mettre automatiquement l'acquisition à l'échelle et nécessite obligatoirement de passer par une application tierce (comme CloudCompare) pour mettre à l'échelle mais aussi pour produire la vue orthographique. Nerfstudio laisse une totale liberté sur le choix de son entrée et sa sortie est directement enregistrée dans l'ordinateur de l'utilisateur.

Garder Nerfstudio comme pipeline de modélisation 3D par NeRF nécessitera donc de lui fournir des données à l'échelle et de traiter le PC qu'il produira en sortie pour projeter la vue orthographique. On commence déjà à voir que la solution idéale pour un public de non-initiés serait un hybride entre les deux : la simplicité de Luma AI et un pipeline Nerfstudio spécialisé pour la production de vues orthographiques.

IV-3.2.2 Choix de la méthode d'acquisition

Maintenant que la technique de modélisation est fixée, on peut enfin définir la méthode d'acquisition. Le choix de l'appareil de capture en lui-même mais aussi sa praticité; le format d'export des données et leurs compatibilités avec Nerfstudio sont des éléments qui conditionnent le choix de la méthode d'acquisition générale.

Une multitude d'appareils de captures sont disponibles pour capturer des données (appareil photo, smartphone, drone, scanner,…). Ce sont principalement leurs capacités à produite des données qui limitent le choix; les données devant permettre la mise à l'échelle et être compatibles avec Nerfstudio.

Même s'il est possible de créer son propre parseur de données (et élargir les sources), Nerfstudio intègre déjà des parseurs pour beaucoup de sources de données utilisées industriellement (voir Tableau 1).

Tableau 1 : Liste des sources d'entrées préconfigurées de Nerfstudio (Nerfstudio,2025)

Data	Capture Device	Requirements
Images	Any	COLMAP
Video	Any	COLMAP
360 Data	Any	COLMAP
Polycam	IOS with LiDAR	Polycam App
KIRI Engine	IOS or Android	KIRI Engine App
Record3D	IOS with LiDAR	Record3D app
Spectacular AI	IOS, OAK, others	App / sai-cli
Metashape	Any	Metashape
RealityCapture	Any	RealityCapture
ODM	Any	ODM
Aria	Aria glasses	Project Aria

Les trois premières sources sont les plus simples: un jeu d'image, une vidéo ou un aperçu 360°. Ils peuvent être acquis à partir de n'importe quel appareil de capture. Leur traitement est automatique dans Nerfstudio (nécessite une installation de Colmap accessible à Nerfstudio). Cela implique qu'on ne passe pas directement par Colmap pour l'alignement et que la mise à l'échelle ne soit pas accessible. Il n'est donc pas possible d'utiliser ces formats, le choix doit se porter sur des approches plus élaborées.

Metashape, RealityCapture, OpenDroneMap et Kiri engine sont des logiciels spécialisés en 3D/SIG. Il prennent généralement des séries d'images ou des vidéos en entrée, pouvant être capturées à partir de n'importe quel appareil de capture. Avant l'alignement des caméras, ils permettent d'effectuer une mise à l'échelle très précise (et même un géoréférencement) et donc constituent une source de données exploitables. Leur utilisation implique cependant l'ajout d'un logiciel intermédiaire dans le pipeline et donc de complexifier le workflow pour les utilisateurs. Ces logiciels apportent une solution intermédiaire par défaut.

Aria n'as pas été envisagé car nécessitant un appareil de capture trop spécifique.

Il reste finalement les trois applications mobiles 3D: Record 3D, Polycam et Spectacular AI. Seul Polycam est concentré sur l'acquisition de PCD, en plus d'avoir une mise à l'échelle directe en profitant du LiDAR d'Apple. Ce dernier pouvant être ajusté par des simples clics dans l'interface de l'application avant l'export. Record3D, quant à lui, est beaucoup plus focalisé sur la gestion des vidéos 3D et n'offre pas la possibilité de mise à l'échelle, revenant au même niveau que les logiciels spécialisés précédents. Spectacular AI est assez prometteur, se concentrant sur le suivi des caméras à l'aide de l'IMU (pour le déplacement en casque VR et le pilotage de drone), il donne exactement les données à l'échelle nécessaire à Nerfstudio. Il n'est cependant disponible que sous la forme de kit de développement et son utilisation est donc trop complexe pour les non-initiés.

Ainsi, c'est Polycam qui offre la solution la plus intuitive et la plus accessible, l'application réduit les intermédiaires en combinant capture et mise à l'échelle au sein d'une même interface simplifiée pour les publics non-initiés. Le principal défaut de l'application étant qu'il restreint les appareils de capture aux smartphones, (notamment ceux d'Apple, qui permettent de profiter de la précision du LiDAR). Les smartphones Apple restant largement accessibles au grand public et ergonomiques, cette limite est relativement mineure.

Pour conclure cette section, la méthode d'acquisition la plus propice utilise l'application Polycam sur IOS (LiDAR + photogrammétrie) permettant une combinaison facile avec la modélisation NeRF sous Nerfstudio. Cette combinaison sera commune aux trois expériences et s'adapte à tous les publics.

IV-3.3 MATÉRIELS ET LOGICIELS GÉNÉRIQUES

Le cadre d'acquisition/modélisation étant établi, il est désormais possible de lister le matériel requis lors du relevé du jeu de données de référence. L'inventaire sert aussi pour les sites secondaires, la phase expérimentale et l'analyse des résultats.

IV-3.3.1 Appareil de capture

La compatibilité LiDAR de Polycam se fait uniquement sur les appareils mobiles d'Apple. Le LiDAR Apple n'est disponible que sur les modèles récents (à partir de l'iPad pro 2020 (mars 2020) et de l'iPhone 12 pro (octobre 2020).

(1)

(2)

(3)

Figure 15 : Matériel d'acquisition : (1) Face arrière de l'iphone 13 pro (Apple, 2025), (2) Matétriel général d'acquisition (drone, casque, radiocommande, batteries de rechanges, iphone 13 pro, perche retractable, mètre ruban et niveau à bulles), (3) Drone dji fpv avec l'iphone sur monture lors d'un vol d'essai.

- Durant ce mémoire, les acquisitions sont réalisées à l'aide de l'iPhone 13 pro. Nous pouvons voir sa face arrière à la Figure 15.1, son LiDAR est notamment situé sous les 3 caméras en bas à droite (cercle noir plus petit). Entre autres spécifications, on notera : la dimension (146,7mm x 71.5mm x 7,65mm), le poids (204 g), le stockage (256GB), la ram (6GB) le processeur (Apple A15 Bionic (6-cœurs CPU, 5-cœurs GPU, Neural Engine 16-cœurs)), le système de caméras à trois objectifs (Grand-angle (f/1.5), ultra-grand-angle (f.1.8) et le téléobjectif (f/2.8)). En termes de senseurs, il est équipé d'un gyroscope 3-axes, un accéléromètre, un capteur de proximité, un capteur de luminosité et bien évidement d'un LiDAR. La combinaison des caméras, du LiDAR et des capteurs de mouvement (IMU) permet à Polycam d'opérer à son plein potentiel. Apple est très discret sur les spécifications de son LiDAR mais retenons en particulier sa portée restreinte à 5m.

En Belgique, en 2025, on peut trouver cet iPhone dans une brochette de prix allant de 400 à 600 euros selon son état.

IV-3.3.2 Support d'acquisition

L'acquisition du jeu de données de référence se fait à la main, c'est-à-dire que l'utilisateur filme manuellement en portant l'iPhone autour de la façade d'intérêt. C'est évidemment le support le plus accessible à tous les publics. Cependant pour la section mise en situation, il est intéressant de compléter ce support avec des alternatives plus complexes.

Tout d'abord, la perche qui augmente la portée de l'utilisateur (permettant de voir certains toits ou encadrement assez bas) (voir Figure 15, .2). Dans ce cas-ci, la perche fait 3m de haut et est rétractable, elle permet d'atteindre des lignes d'acquisition à 4m si l'utilisateur la tient à bout de bras.

Ensuite, il a été possible de monter l'iPhone sur un drone qui était à disposition. Le drone est un dji fpv. Il est sorti en 2021 et pèse 795g pour une autonomie de 10 à 20 minutes. L'intérêt d'utiliser ce drone est qu'il est conçu pour le vol à la première personne (fpv) de ce fait, ses moteurs sont très puissants (par rapport à son poids) et n'ont pas de difficulté à soulever la masse supplémentaire de l'iPhone. Il est tout de même équipé d'un module GNSS et permet donc la stabilisation, le RTH et la localisation. On peut le voir avec sa manette et son casque VR à la figure 15.2 et en vol avec l'iPhone en .3.

Ces deux options ne sont pas du tout indispensables mais sont deux pistes intéressantes. De son côté, la perche est un simple ajout pouvant augmenter la portée des utilisateurs. L'utilisation du drone, quant à elle, serait intéressante dans les cas d'interventions en zone sinistrée où l'accès au bâtiment est trop dangereux. La question étant de savoir si l'acquisition peut facilement être conduite avec ces supports, s'ils rajoutent trop de contraintes ou en enlèvent.

Néanmoins, le mémoire se concentre principalement sur le workflow après l'acquisition (l'acquisition devant rester la plus simple possible par rapport au public des non-initiés) donc les essais avec des supports alternatifs sortent un peu du cadre du mémoire. Afin de ne pas surcharger le corps du mémoire les résultats des campagnes annexes sont présentes dans l' annexe 1.

IV-3.3.3 Ordinateur de calcul

Pour terminer la partie hardware, voici les spécifications de l'ordinateur utilisé pour les calculs. L'ordinateur personnel de l'auteur a été utilisé, ce qui signifie que de simples ordinateurs non spécialisés peuvent être utilisés (dans le cadre de la recherche d'implémentation au moins).

Faire tourner Nerfstudio reste exigeant sur le hardware. Le coût prohibitif en calculs nécessite des performances correctes surtout en termes de GPU. En effet, Nerfstudio a été développé en utilisant spécifiquement les cartes graphiques Nvidia puisqu'il se base sur tiny-cuda-nn, une bibliothèque open-source développée par Nvidia eux-mêmes et qui permet de créer des réseaux neuronaux très rapides en les optimisant pour les cartes Nvidia. L'utilisation d'autres cartes ou du CPU est possible mais rend le processus extrêmement lent ce qui fait perdre beaucoup d'intérêt par rapport à la photogrammétrie.

Le Tableau 2 reprend les spécifications de l'ordinateur utilisé (comme outil d'implémentations et comme serveur).

Tableau 2: Spécifications du système utilisé pour les calculs

Fabriquant	HP
Modèle	OMEN 25L Desktop GT12-1xxx
CPU	Intel(R) Core(TM) i7-11700F @ 2.50GHz (16 processeurs logiques, 8 processeurs physiques)
Système d'exploitation	Windows 11 (64 bits)
GPU	NVIDIA GeForce RTX 3070 (Nom du pilote DirectX : nvldumd.dll, Version du pilote et DirectX : 32.0.15.7216, Date du pilote : 26/1/2025)
RAM	16 Go
SSD	512 Go
Disque dur	1000 Go

À noter que l'ordinateur n'est pas neuf et est en service depuis le 26 octobre 2022. Il est difficile de rendre compte de son état d'usure mais on peut au moins indiquer son temps d'activité (temps où l'ordinateur est allumé) qui est de 9 340 114 secondes (2594,5 heures).

IV-3.3.4 Matériel software orienté 3D

Les deux sections qui suivent passent brièvement en revue les softwares qui vont être utilisés dans les implémentations. Une simple description (pour mettre à niveaux tous les lecteurs) est donnée comme introduction mais les détails seront explicités en fonction de leur utilité au sein de chaque implémentation.

On peut classer les logiciels et les bibliothèques en deux catégories : ceux orientés 3D/SIG et ceux purement dédiés aux développements. Toujours dans un soucis d'accessibilité tous les logiciels sont disponibles gratuitement et souvent open-source. La seule exception est Polycam qui, en versions gratuites, est bridées dans le volume de requête (130 images par acquisition). La limitation peut être levées par le biais d'un abonnement de 17,99 euros pour Polycam. La fonction principale du logiciel reste complètement accessible en version gratuite. De ce fait, les workflows ne seront pas impactés si on possède ou non la versions payante. La mise à jour payante représente uniquement du confort et du volume de traitement supplémentaire.

- Polycam: comme on l'a déjà évoqué, Polycam est une application mobile multi-plateforme. Elle permet la capture automatisée de plans, d'objets, d'espaces et d'images 360 en 3 dimensions en combinant la photogrammétrie et le LiDAR (s'il est disponible). Elle permet également d'exporter les résultats et de les rendre accessibles sur le web par url. De nombreuses entreprises et institutions l'utilisent déjà (Polycam, 2025) et elle est souvent citée dans la littérature. Cette application est essentielle (puisque la capture et l'alignement passent par elle) et sera présente dans chaque workflow. La version utilisée durant ce mémoire est la version 5.0.6 (f59fa916).
- CloudCompare: logiciel 3D spécialisé dans le traitement des PCD et des MESH. Il a originellement été conçu pour effectuer des comparaisons entre PCD denses ou entre PCD et MESH. Il a ensuite été étendu pour être plus générique et comprend notamment l'alignement, le rééchantillonnage, la gestion des champs scalaires, le calcul de statistiques, la gestion des capteurs, la segmentation, …) (CloudCompare, 2025). Ce logiciel est assez important car il permet notamment de générer des vues orthographiques nativement en plus de permettre de visualiser facilement des PCD et des MESH.

IV-3.3.5 Matériel software orienté développement

Une grosse partie du travail d'implémentation réside dans le développement d'une série de scripts, la modification de scripts existants et la recherche de bibliothèque et solutions. Cette approche est très complexe et multidisciplinaire, la bonne exécution de celle-ci dans un temps raisonnable nécessite souvent plusieurs logiciels pour la gestion de l'environnement de programmation et des bibliothèques ou pour la recherche de solutions.

- Miniconda : Miniconda est une version gratuite et miniature de la distribution Anaconda. Elle inclut de base seulement Conda et Python. Anaconda étant une distribution spécialise dans la science des données avec Python et R. Elle permet de gérer les environnements, les commandes et d'installer automatiquement de nombreux packages (Anaconda, 2025). Elle est nécessaire pour le fonctionnement de Nerfstudio mais est aussi très utile pour l'installation d'autres packages. Elle permet aussi de gérer

des environnements, ce qui permet de faire des versions différentes pour chaque implémentation tout en restant sur le même ordinateur.

- GitHub : plateforme collaborative de développement pour construire et partager des codes, des bibliothèques et des logiciels. Elle est extrêmement connue et entrepose énormément d'éléments dont certains seront retrouvés dans nos implémentations. Les extensions de GitHub permettent de coder et de télécharger du code très facilement. A noter que Nerfstudio est stocké sur cette plateforme et nécessite celle-ci pour son téléchargement et ses mises à jour, il était obligatoire de le signaler.

Une partie des logiciels utilisés intègre de l'intelligence artificielle. Au vu de ce que cela signifie dans le cadre d'un mémoire au niveau de l'intégrité et du plagiat, il est important de signaler leurs utilisations mais aussi de bien exposer les limites établies. L'annexe 2 reprend toute la discussion qui démontre un usage réfléchi, critique et transparent de ces outils.

IV-3.4 SITES D'ÉTUDE

Cette section décrit le site d'étude choisi pour la production du jeu de données de référence en service durant toute la phase expérimentale. Des sites secondaires ont également été sélectionnés pour servir durant la mise en situation.

Tous les sites sont situés en Belgique dans la province de Namur et plus précisément aux alentours de Lustin et d'Ivoy. Les sites sont proches de quelques kilomètres les uns des autres par praticité, leurs localisations précises (en WGS84) sont reprises à la Figure 16.

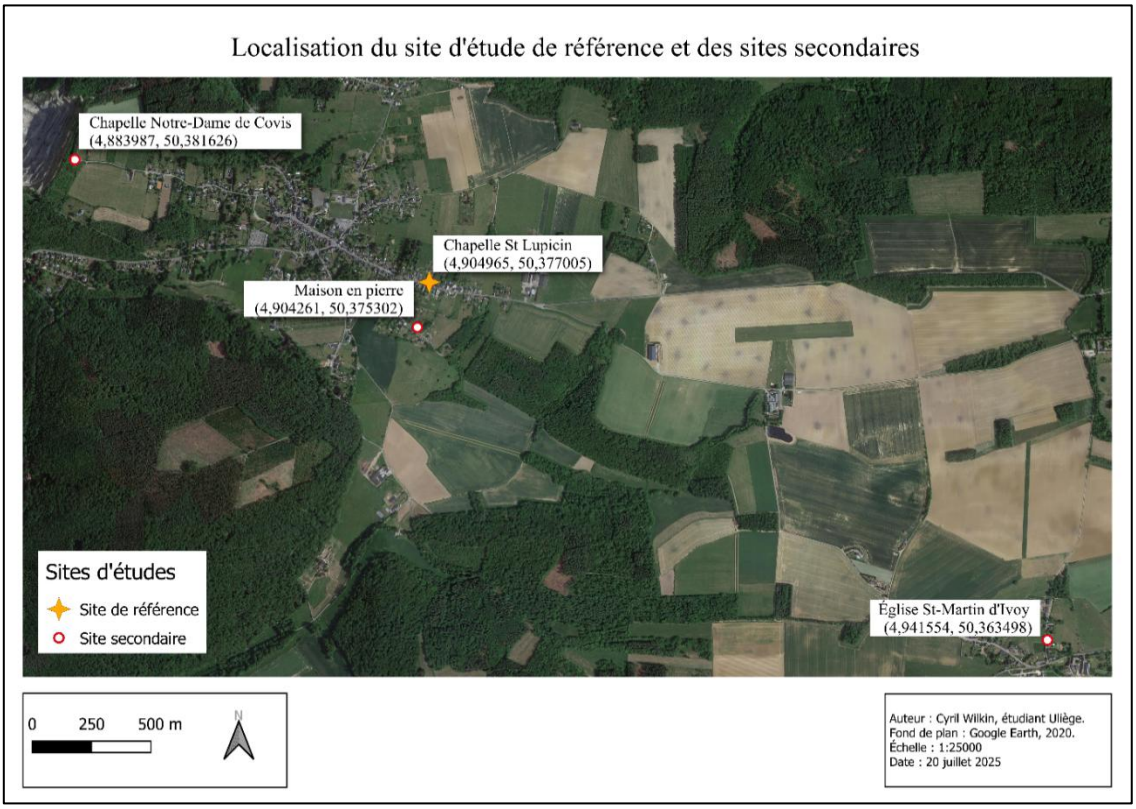


Figure 16 : Cartographie des différents sites d'étude

IV-3.4.1 Site principal : chapelle St-Lupicin de Lustin

Les caractéristiques recherchées pour le site principal doivent faire ressortir les avantages des NeRF par rapport aux techniques de modélisations classiques. De plus, ce site doit permettre un relevé très simple pour être représentatif de ceux effectués par des non-initiés.

1. 

2. 

Figure 17 : Site principal : façade avant (1) et façade arrière (2) de la chapelle St-Lupicin de Lustin

Le site choisi est la chapelle St-Lupicin située à Lustin dans la rue des 4 Arbres. C'est une petite chapelle de 7m de haut pour 5m de large (voir Figure 17). Deux grands axes ont mené au choix de ce site.

Premièrement, le site assure au jeu de données de référence une bonne précision et une scène complète sans occlusion. En effet, la chapelle est située au centre de la place, ce qui permet de se mouvoir librement durant le relevé mais surtout d'effectuer un tour complet de l'objet d'intérêt. De plus, la chapelle est suffisamment petite pour effectuer une seule acquisition, réduisant ainsi le risque de déviation. Pour finir, elle est très simple (6 façades planes et une toiture) ce qui ne devrait pas générer d'occlusions.

Deuxièmement, le site possède des particularités mettant en avant certains avantages des NeRF. Les façades arrières sont uniformément blanches, n'ont pas de fenêtre et ne sont pas encadrées par des pierres de tailles comme les façades avant. De plus, la chapelle n'est pas carrée mais hexagonale ce qui réduit la netteté des angles entre les façades (le toit aussi). Ces deux particularités rendent la chapelle très lisse, ce qui aurait tendance à entraver fortement la détection de points correspondants si on utilisait une approche photogrammétrique.

Dans une moindre mesure, le site permet d'observer le comportement des NeRF sur les surfaces transparentes et sa capacité à différencier les matériaux de différentes tailles (tuile, pierre de taille, briquette). La façade avant est relativement décorée et servira de façade de référence pour la projection de vue orthographique. La vue des autres façades évaluera la projection suivant des angles différents et pas forcément parallèles à un axe (forme hexagonale).

IV-3.4.2 Sites secondaires : église Saint Martin d'Ivoy, chapelle Notre-Dame de Covis et maison en pierre

Les sites secondaires sont sélectionnés pour simuler des scénarios plus complexes de capture et de modélisation. Leurs relevés sont analysés après la phase expérimentale en tant que mise en situation et permettent d'enrichir les discussions. Un aperçu des trois sites est disponible à la Figure 18.

Figure 18 : Sites secondaires : Aperçu de l'église St-Martin d'Ivoy (1), de la maison en pierre (2) et de la chapelle Notre-Dame de Covis (3)

L'église St-Martin d'Ivoy est située à Ivoy (Maillen, section A, Parcelle 553). C'est une ancienne église paroissiale de style roman et classique. (SPW, 2018). Les nombreuses reconstructions sont particulièrement visibles sur la façade sud qui comporte un grand nombre d'agencements et de types de pierres différents. Le site est sélectionné pour plusieurs caractéristiques : l'église est entourée d'un cimetière carré très éparse permettant de s'écarter de la façade sud de 20m; le cimetière n'est clôturé que par un bas muret en pierre, ce qui permet d'effectuer un tour complet du bâtiment; la plus grande longueur du bâtiment est de 25m, de la porte d'entrée à l'arrière de la nef; la diversité des matériaux et leur intérêt archéologique. Ces différentes caractéristiques permettent d'effectuer plusieurs expériences: un relevé de grande envergure avec un tour complet, 5 relevés à des distances différentes de la façade (2m, 3m, 4m, 5m, 10m), 2 relevés utilisant la perche et le drone. Chaque relevé est destiné à être évalué en praticité, au visuel (avec les différentes reconstructions) .

La chapelle Notre-Dame de Covis est située à Lustin au fond de la rue Covis. Cette chapelle a été entièrement reconstruite en 2019 (Nostalgie Lustinoise, 2021). Son nouveaux style moderne est particulièrement intéressant pour faire ressortir les possibilités des NeRF. En effet, l'édifice est constitué de tôles métalliques, de tuiles réfléchissantes et surtout ses façades avant et arrière sont totalement en verre. De plus, le jardin l'entourant limite à certains endroits le recul possible. Le relevé effectué permet d'évaluer l'efficacité des NeRF pour modéliser les matériaux réfléchissants et transparents et aussi la forme complexe de la chapelle.

Enfin la maison en pierre située à Lustin. Elle a pour particularité d'avoir une façade sud très verticale (pignon de 11 m pour 6m de large) et d'être sur un terrain privé. Elle permet de comparer un relevé effectué à partir du sol et un relevé réalisé au drone. Le site étant privé, les relevés peuvent être faits sans autorisation. Le relevé au sol permet également d'évaluer la qualité des NeRF quand une grande partie de l'objet d'intérêt n'est pas couvert par le LiDAR (le LiDAR Apple à une portée limitée à 5m).

IV-3.5 ACQUISITION DU JEU DE DONNÉES DE RÉFÉRENCE

Cette première acquisition délivre à la phase expérimentale un jeu de données simple et représentatif, ainsi qu'un début du workflow qui restera identique jusqu'à la mise en situation. Il est donc important de bien décrire chaque étape. Cette section donne également un bon aperçu du fonctionnement de l'application Polycam et de son intérêt.

IV-3.5.1 Acquisition de référence avec Polycam

L'acquisition par Polycam reste très similaire à celles pouvant être retrouvées dans d'autres applications de scan 3D mais c'est surtout dans son traitement qu'elle trouve son intérêt. Polycam a été conçu pour générer des maillages 3D à partir d'une combinaison de photogrammétrie et de LiDAR. Lors du traitement, il utilise les données du LiDAR pour retrouver les paramètres intrinsèques et extrinsèques des caméras, de plus il corrige automatiquement les images et génère les depth map correspondantes. Ces données brutes sont ensuite utilisées pour modéliser un MESH texturé. Or, ces données brutes peuvent être utilisées pour de nombreuses autres applications (notamment les NeRF, les reconstructions d'intérieur, …) (polyform, 2025). Polycam permet donc d'exporter les données brutes et ce sont elles qui constituent les jeux de données exploitables par Nerfstudio (Chaque image du scan possédant des métadonnées sur les caméras).

La suite de la section présente l'acquisition de référence et les étapes suivies dans Polycam pour générer le jeu de données de référence.

L'acquisition de référence (Chapelle St-Lupicin, Lustin) s'est déroulée le 6 avril 2025 à 7h21 du matin. Le temps était très légèrement couvert mais stable et assurant une bonne luminosité. L'acquisition a pris place le matin pour éviter les ombres portées de l'après-midi (la place est entourée de bâtiments) et pour profiter d'une heure où la place était vide de tout véhicule, laissant ainsi une liberté de mouvements maximale. La Figure 17 donne des images directement tirée de l'acquisition et donne un bon aperçu des conditions environnementales de l'acquisition.

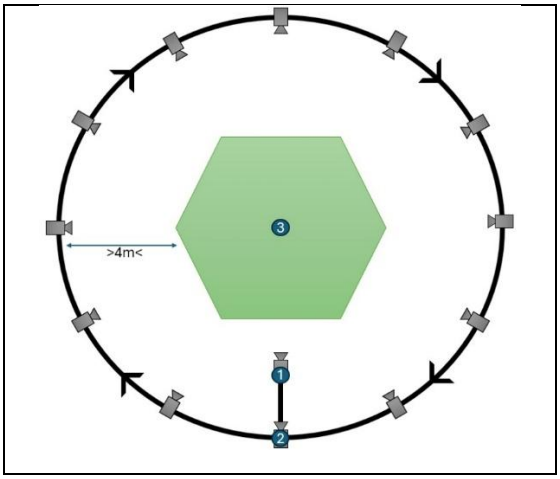


Figure 19 : Représentation du trajet de l'acquisition de référence

Pour rester le plus simple possible un seul tour de la chapelle a été effectué en manuel. Comme le montre la Figure 19, le tour commence par un dézoome de la porte d'entrée (de 1 vers 2), effectue un trajet circulaire de centre 3, puis revient en 2. L'acquisition se termine par un zoom de 2 en 1 pour revenir au point de départ. Durant tout le circuit, l'iPhone a été tenu verticalement et orienté pour rester perpendiculaire aux façades. De surcroit, l'opérateur s'est maintenu à une distance de 4m des façades et a tenu l'iPhone à hauteur de poitrine, ceci assurant de toujours capturer la chapelle de sa base à la croix du toit. L'outil de scan libre de Polycam garantit que le nombre de photos soit suffisant et que la vitesse de parcours ne soit pas trop élevée. En plus de la triangulation en temps réel comme indicateur visuel, l'application envoie des notifications si la capture tremble trop ou si la vitesse de parcours augmente trop. L'acquisition a duré 11 minutes seulement.

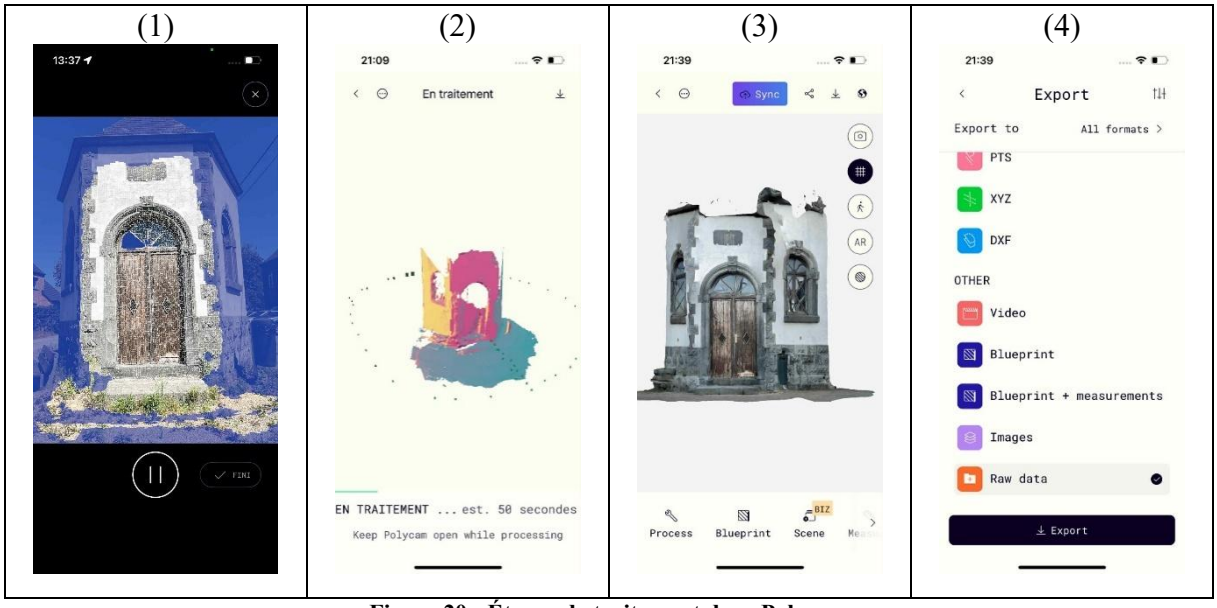


Figure 20 : Étapes de traitement dans Polycam

Voici les différentes étapes effectuées dans Polycam durant l'acquisition et jusqu'à l'export des données brutes:

1) Activation du mode développeur. Ceci est nécessaire car en mode normal, l'export des données brutes est caché.
2) Création d'une nouvelle capture en mode LiDAR. Polycam propose de nombreux modes de capture différents comme le mode plan, le mode objet ou espace mais seul le mode LiDAR permet de scanner librement une scène sans protocole de base.
3) Acquisition en suivant bien les notifications à l'écran. Exemple à la Figure 20-1.
4) Traitement des données brutes en mode espace. Générer le MESH reste obligatoire car la calibration des données brutes fait partie du pipeline de traitement. Les traitements peuvent être personnalisés pour produire différentes qualités de MESH mais cela n'a pas d'impact sur les données. Le mode par défaut « espace » est donc amplement suffisant. La Figure 20-2 montre Polycam en train de réaliser le traitement de la chapelle, la heatmap est visible ainsi que chaque prise de vue. Dans l'acquisition de référence le traitement prend 50 secondes.
5) Visualisation du MESH. Il peut être visualisé en mode perspective ou orthographique, avec ou sans texture. Analyser le MESH permet une première validation de l'acquisition car il représente les endroits frappés par le LiDAR. Dans le cas de l'acquisition de

référence (voir Figure 20-3), le MESH généré montre bien que le LiDAR a frappé toutes les façades. Il apparait aussi que le MESH a des problèmes de modélisation le long des bordures et des axes mais les surfaces des façades sont correctement modélisées.

6) Synchronisation du projet en ligne (pour faciliter l'export à venir). Durée : 2 minutes.
7) Exportation des données brutes. L'icône des exports offre un large choix de produits, « raw data » est sélectionné (voir Figure 20-4).
8) Envoi des données par « mail drop » ou téléchargement depuis le cloud. Cette étape est assez longue et fastidieuse, impliquant de passer par des boites mails extérieures et de passer d'iOS à Windows. Durée de 5 à 20 minutes selon la taille du jeu de données mais aussi la disponibilité des serveurs d'Apple. Dans l'acquisition de référence, les données brutes ont été envoyées par mail drop pour être téléchargées sur l'ordinateur de calcul. Les données ont été disponibles sur l'ordinateur de calcul au bout d'environ 7 minutes.

En tout est pour tout, l'acquisition de référence a pris 20 minutes. Elle a capturé 56 images, et a généré un MESH de 54 500 vertex pesant 6.2 Mo. Le jeu de données brutes exporté pèse 20,6 mo.

Ces 8 étapes peuvent être compilées en 3 sections : d'abord la capture de la scène en mode LiDAR, ensuite, la génération du MESH et la calibration des caméras et enfin, l'export des données brutes. Ceci constitue la base de tous les workflows qui suivront en phase expérimentale. La capture pourra être discutée en fonction des résultats de la mise en situation.

IV-3.5.2 Jeu de données de référence

D'après la documentation Polycam (2025), les données brutes sont contenues dans un dossier « Keyframe », qui contient pour chaque frame capturé :

- Un fichier au format JSON (sous-dossier « camera ») contenant les dimensions de l'image, le timestamp, un score de flou (blur_score), les paramètres intrinsèques de caméra (la distance focale et le centre optique), les paramètres extrinsèques de caméra (l'orientation est reprise depuis sa matrice de transformation), la vélocité angulaire (mesure la vitesse de rotation pour le suivi de mouvement) et la profondeur centrale (souvent mesurée en mètres).
- Une « Confidence map » (sous-dossier « confidence »), une image représentant la fiabilité de chaque pixel (avec un dégradé de gris).
- Une « Depth map » (sous-dossier « depth ») , une images représentant la distance des pixels depuis le capteur LiDAR (toujours en dégradé de gris).
- L'image RGB en .jpg pour la texturisation (sous-dossier « image »).

Chaque frame possède un numéro d'identification permettant d'associer les données contenues dans les 4 sous-dossiers. Le jeu de données contient également des métadonnées générales : les métadonnées du MESH, les coordonnées et l'altitude de la capture ainsi qu'une vidéo et une vignette de prévisualisation. Ce qui constitue le jeu de données de référence et qui sera utilisé durant la phase expérimentale (l'entrée commune) sont les 56 images capturées lors de l'acquisition de référence, ainsi que les 56 fichiers JSON contenant les paramètres de caméra associés.

IV-4 SÉGMENTATION DE LA PHASE EXPÉRIMENTALE

Ce chapitre IV se termine par la segmentation de la phase expérimentale. L'entrée commune et les sorties attendues étant définies, un contexte commun encadrant les futures expériences est créé (voir schéma à la Figure 21). Il convient désormais de définir la portée de chaque expérience. En effet, comme cela est expliqué dans la méthodologie (cf. Chapitre III), la phase exploratoire est segmentée, afin de produire des résultats pour chaque public indépendamment les uns des autres. Ainsi, cette approche de segmentation offre la possibilité de ne pas perdre les résultats intermédiaires pouvant être utiles pour les publics expert ou intermédiaires.

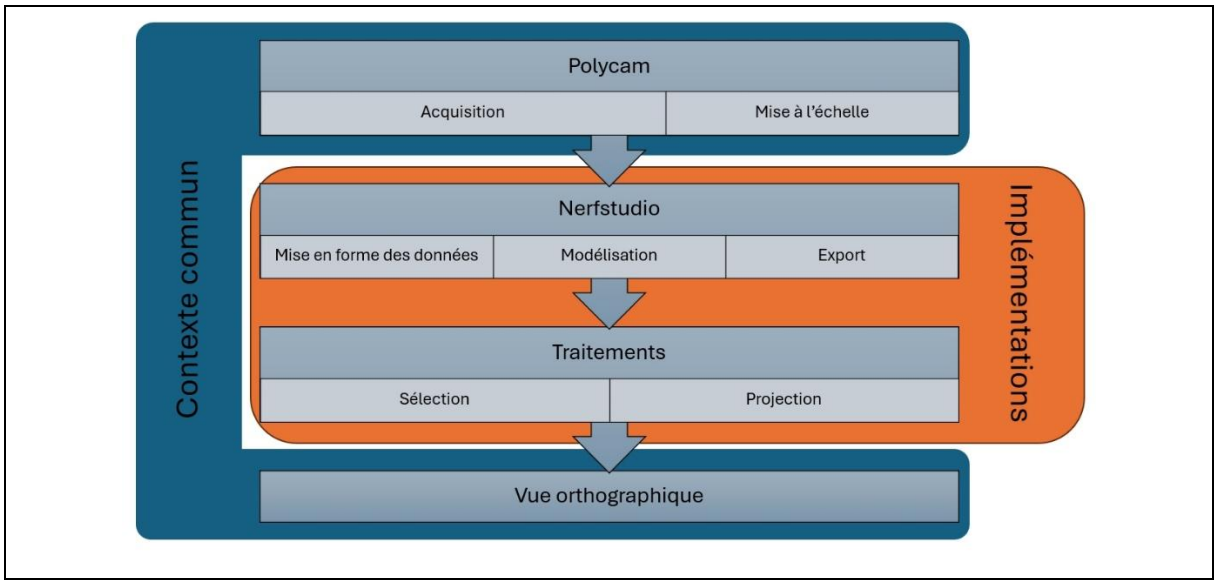


Figure 21 : Diagramme représentant les limites du contexte commun et des expériences

Concrètement, chaque expérience se voit attribuer un public spécifique tel que défini précédemment (cf. IV-1). La première implémentation produit un workflow qui s'adresse à un public d'experts, la seconde à un public intermédiaire et la troisième à un public de non-initiés.

Chaque expérience se voit donner une direction générale et un idéal de développement qui correspond aux limitations et exigences de son public. L'implémentation de la première expérience vise avant tout à permettre la production effective d'une vue. Tandis que les implémentations des expériences 2 et 3 visent plutôt des applications modèles à atteindre. Spécifiquement:

- La première implémentation doit uniquement produire un workflow avec l'objectif d'obtenir la vue orthographique. Elle n'a pas de limitation puisque le public d'initiés peut utiliser tous les outils. Elle doit cependant être rigoureuse car les deux autres implémentations s'appuieront sur celle-ci. La direction générale à respecter dégage un workflow de traitement similaire à ceux de géomatique, une suite d'étapes qui transforme une donnée en un produit à l'aide de plusieurs logiciels.
- La deuxième implémentation possède des limitations techniques en informatique notamment. L'utilisation de lignes de commandes et les installations lourdes sont donc proscrites. La paramétrisation des NeRF est restreinte aux grandes options. Le public maitrise néanmoins l'utilisation des logiciels génériques (comme CloudCompare) pour la visualisation ou la gestion des PCD. Dans cette configuration, une application de bureau accessible hors-ligne dans laquelle l'utilisateur est guidé pour suivre le workflow de traitement (connaissance théorique) semble indiquée pour répondre aux limitations.

En effet, dans un tel outil numérique, les opérations sont lancées par des boutons (application « push-button »). L'utilisateur n'a donc pas besoin de savoir rentrer une ligne de commande ou d'instruction python mais il conserve une grande liberté valorisée par sa connaissance théorique. RealityCapture, MetaShape, Pix4D,… sont des exemples de suites logicielles de photogrammétrie construites sur ce modèle.

- La troisième implémentation est fortement limitée. L'utilisateur doit rencontrer le moins d'étapes possibles entre l'acquisition et le produit. Le NeRF doit déjà être paramétré, la gestion des fichiers et la production des vues doivent être automatisées. La seule opération laissée à l'utilisateur est l'acquisition et la personnalisation de la vue orthographique. L'éditeur de Luma Ai et son application de capture proposent un bon exemple de workflow ultra-simplifié et intuitif. Celui-ci dirige l'utilisateur directement vers le produit attendu. Le site est accessible à toutes les plateformes via le web et prend en charge tous les calculs lourds des NeRF.

Le Tableau 3 synthétise la segmentation et aide à se représenter les capacités et les objectifs de chaque implémentation.

Tableau 3 : Récapitulatif des limitations et objectifs des implémentations

Workflow	Connaissances théoriques	Connaissances pratiques	Lignes de codes	Ajuster des paramètres/ CloudCompare	Faire l'acquisition	Modèle/ objectif	Hypothèse
1	Oui	Oui	Oui	Oui	Oui	Obtenir une vue orthographique	H1
2	Oui	Non	Non	Oui	Oui	AutoCAD, Agisoft Metashape	H2
3	Non	Non	Non	Non	Oui	Editeur Luma AI	H2

Comme cela est observable, les workflows sont graduellement de plus en plus accessibles. Dans le même temps, le niveau de connaissance requis se réduit progressivement. En partant de la première expérience qui adresse la première hypothèse (obtenir la vue orthographique 2D), les deux autres expériences sont développées ; celles-ci voient leurs workflows simplifiés pour adresser la seconde hypothèse. Le contexte commun permet de garder comparables les résultats des différentes expériences.

CHAPITRE V PREMIÈRE EXPÉRIENCE : WORKFLOW À DESTINATION D'UN PUBLIC D'EXPERTS

La première expérience suit la méthodologie établie au Chapitre III. Au départ du contexte commun, on dégage un cadre propre à l'expérience menant à une implémentation qui répond aux problématiques de ce cadre. Grâce à cette implémentation, il est possible de dégager un workflow et des vues orthographiques pouvant être comparées et analysées. Le chapitre passe en revue le cadre, l'implémentation, détaille en profondeur le workflow puis présente ses produits (sorties attendues). Les conclusions intermédiaires de l'expérience sont consignées dans la dernière partie du chapitre et seront utiles pour les expériences suivantes.

V-1 CADRE PROPRE À L'EXPÉRIENCE ET PRÉSENTATION

Dans cette première expérience, le tout premier workflow est établi. Il faut garder en tête qu'il sert de base pour les expériences suivantes. Dès lors, l'implémentation mise en place se doit d'utiliser des technologies qui peuvent être optimisées en terme d'accessibilité.

Dans cette optique, seulement deux logiciels de traitement sont utilisés : Nerfstudio, qui entraîne le modèle NeRF et qui exporte les résultats sous forme d'un PCD, puis CloudCompare, qui projette orthogonalement le nuage sur un plan bidimensionnel en générant une image (.png). Les principes régissant les NeRF sont largement expliqués au Chapitre I, toute la force de Nerfstudio est de proposer une solution « tout en un » pour appliquer ces principes en quelques commandes dans une interface par ligne de commandes (CLI).

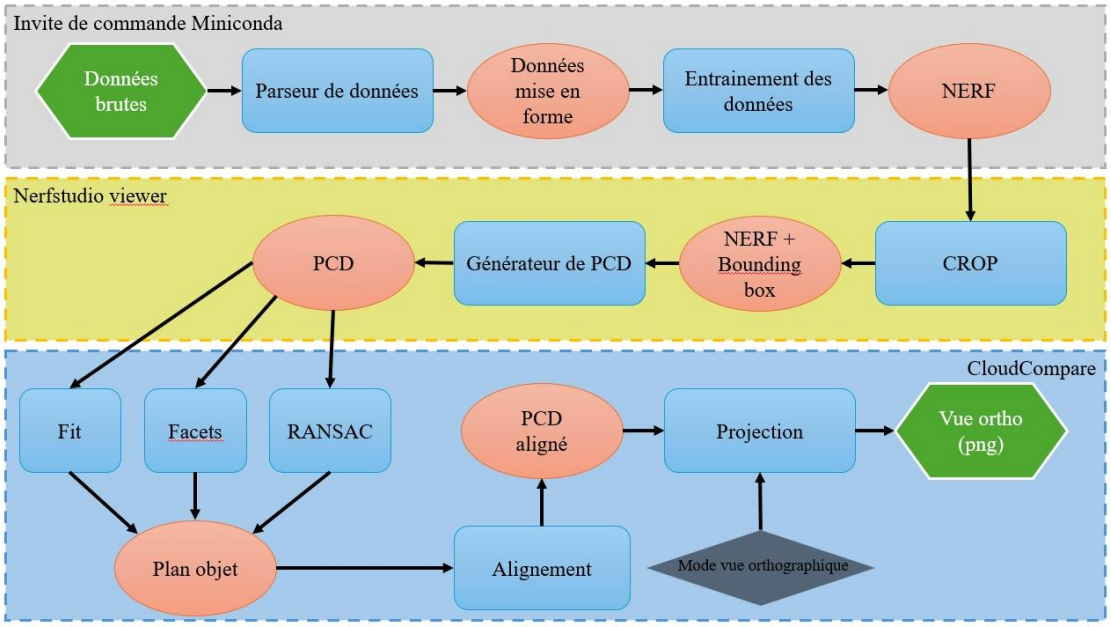


Figure 22 : Résumé graphique du workflow après l'implémentation de la première expérience.

Le diagramme de la Figure 22 présente un résumé graphique du workflow possible grâce à l'implémentation développée dans cette expérience. Les données, brutes au départ, sont préparées et puis, à partir d'une invite de commande, permettent d'entrainer le modèle NeRF. Ce dernier permet à son tour de générer un PCD dans le Nerfstudio viewer. Ce PCD est ensuite importé dans CloudCompare, ou l'objet peut être aligné pour sa projection. CloudCompare permet d'effectuer un rendu représentant une projection orthographique du PCD en suivant un de ses plans.

V-2 IMPLÉMENTATION

Durant cette expérience, l'implémentation se base uniquement sur les installations de logiciels.

CloudCompare est un projet open source et peut être téléchargé à partir du site de son créateur Daniel Girardeau-Montaut (2025): https://www.cloudcompare.org/release/. Le logiciel vient avec son installateur et il suffit de suivre les instructions pour obtenir une version opérationnelle de CloudCompare.

Le site de Nerfstudio et son dépôt GitHub fournissent une large documentation pour l'installation de l'application. Cependant, les auteurs indiquent eux-mêmes que l'installation reste particulièrement fragile, notamment sur Windows. Les installations sont notamment sensibles à la casse suite aux mises à jour des packages et des éléments non-contrôlés par l'installation (Nerfstudio documentation, 2022). Dans les faits, beaucoup d'erreurs et de conflits de versions peuvent survenir pendant et après l'installation. Ce qui explique les multiples possibilités d'installations (git, pixi, docker image,…) qui essayent d'encapsuler un environnement stable. Le Forum du GitHub est très actif pour essayer de résoudre ces problèmes. Dans le cadre de ce mémoire, l'installation de Nerfstudio est considérée comme étant obligatoirement effectuée par un initié théorique et pratique et elle doit être considérée comme du cas par cas.

L'installation réalisée est présentée à titre d'exemple. Elle a été faite sur l'ordinateur de calcul (présenté en IV-3.3.3) qui est doté d'une carte graphique NVIDIA GeForce RTX 3070 et utilise le système d'exploitation Windows 11. Voici les différentes étapes qui ont mené à la création d'un environnement permettant le bon fonctionnement de Nerfstudio :

1) Pré-installations : tous les prérequis pour obtenir l'environnement nécessaire à l'installation de Nerfstudio.
    a. Installation de git: la commande interne qui permet de rapidement télécharger des dépôts provenant de GitHub.
    b. Installation de Visual Studio 2022: la version de bureau contient un kit de développement en C++ (C++ Build Tools) nécessaire au bon fonctionnement de CUDA. La documentation de Nerfstudio indique qu'il doit être activé mais ce n'est pas toujours nécessaire en fonction de l'installation initiale.
    c. Installation de Miniconda 3: le gestionnaire d'environnement venant avec l'installation de Python. C'est un élément essentiel de l'installation car il cloisonne Nerfstudio dans un environnement spécial créé exclusivement pour son fonctionnement. L'isolation de l'installation permet d'éviter beaucoup de conflits avec toutes autres applications.
    d. Installation de CUDA version 11.8: le NVIDIA CUDA Toolkit fournit un environnement pour créer des applications à haute performance sur des systèmes

permettant l'accélération GPU (NVIDIA, 2025). Le toolkit CUDA n'est pas toujours installé par défaut sur les ordinateurs. Par contre, il peut être automatiquement mis à jour. La version avec laquelle a été développé Nerfstudio et qui doit être installée est la version 11.8 (pas une autre, attention aux doublons et au conflits). Si l'installation du toolkit ne le fait pas, il peut être utile de mettre les drivers de la carte graphique NVIDIA à jour, en première approche.

e. Désinstallation de Python, PyTorch, funtorch, tiny-cuda-nn : si aucun autre projet ne les utilise, il est préférable de désinstaller toutes les versions précédentes de ses 4 bibliothèques dans l'environnement général. L'installation peut fonctionner sans cette étape mais elle s'expose a des problèmes de compatibilité de versions. Ces erreurs sont parfois difficilement repérables et/ou corrigeables (exemple durant la deuxième expérience).
f. Installation de Python 3.8 : Nerfstudio ne fonctionne pas en dessous de la version 3.8. Il peut fonctionner avec des versions plus à jour mais, encore une fois, cela risque d'ajouter de l'instabilité.
g. Création d'un environnements dans Conda : dans une fenêtre PowerShell lancée par Miniconda, il faut lancer la commande *« conda create --name nerfstudio -y python=3.8 »*
h. Il est recommandé de nommer l'environnement « nerfstudio » pour continuer à suivre les instructions de la documentation mais le nom peut être personnalisé. Dans le cadre de ce mémoire, un environnement est créé par implémentation (avec des noms différenciables).

2) Installation des dépendances : en plus de Nerfstudio, il faut charger certaines dépendances dans l'environnement.
   a. Activation de l'environnement: dans une fenêtre PowerShell lancée par Miniconda, activer l'environnement avec la commande *« conda activate Nerfstudio »*. Le nom avant l'invite de commande doit passer de *(base)* à *(Nerfstudio)*
   b. Installation de PyTorch 2.1.2 pour CUDA 11.8: le guide recommande d'utiliser la commande *« pip install torch==2.1.2+cu118 torchvision==0.16.2+cu118 --extra-index-url https://download.pytorch.org/whl/cu118 ».* Contrairement au guide automatique nous déconseillons d'installer CUDA avec la commande Conda avant PyTorch mais de le faire manuellement (cette fausse manipulation ayant fait échouer plusieurs tentatives d'installation).
   c. Installation de tiny-cuda-nn : un petit framework autoportant et très optimisé pour l'entrainement et l'interrogation de réseaux de neurones. Nerfstudio s'en sert pour optimiser et éclairer ses NeRF. C'est à cette extension que Nerfstudio doit la rapidité de ses entrainements de modèles. Le guide recommande d'utiliser la commande passant par torch : *« pip install git+https://github.com/NVlabs/tiny-cuda-nn/#subdirectory=bindings/torch ».*

3) Installation de Nerfstudio : l'environnement est enfin prêt à accueillir la bibliothèque de Nerfstudio. L'installation peut se faire depuis le gestionnaire de paquets Python pip *(« pip install nerfstudio »*) ou depuis le dépôt GitHub *(« git clone https://github.com/nerfstudio-project/nerfstudio.git »*). L'installation depuis GitHub a été préférée car plus à jour. C'est optionnel mais il est également possible de mettre à

jour la “complétion de Tab” (avec ns-install-cli) et de télécharger des modules de développement propres à Nerfstudio (pip install -e .[dev], pip install -e .[docs]). Ces 3 modules optionnels ont été principalement utilisés durant les expériences 2 et 3. L’installation correcte de Nerfstudio peut être vérifiée si la commande « ns-train –help » (une demande d’information de la commande d’entrainement) renvoie son texte sans erreur.

L’installation nécessite donc de nombreuses étapes qui doivent être respectées et effectuées dans l’ordre pour produire une installation fonctionnelle. L’installation reste fragile car si un des éléments est mis à jour par une quelconque source extérieure, il peut briser en cascade le reste de l’installation. De plus, la casse n’est pas toujours visible directement et peut se révéler tardivement en fonction de l’utilisation spécifique d’une fonction. L’environnement a été brisé des dizaines de fois surtout pendant les tests d’algorithmes et lors des deuxième et troisième expériences en installant des extensions Python.

Le temps d’installation est relativement long notamment à cause des dépendances. Python, PyTorch et Nerfstudio demandent l’installation de plusieurs centaines de dépendances (369 rien que pour Python). L’installation initiale de NVIDIA CUDA est également assez longue. Le temps total pour une installation propre pouvant aller de 30 minutes à 2h. Sachant que la désinstallation prend également du temps (15 à 30 minutes), il est aisé de comprendre en quoi briser un environnement peut être chronophage.

Pour résumer, l’installation de Nerfstudio apparait assez instable. Le développement autour de Nerfstudio est assez risqué si on ne cloisonne pas correctement ses environnements. L’installation par des non-initiés en informatique n’est clairement pas envisageable. Toutefois, il est important de rappeler que c’est un projet open-source de recherche académique mis en place il y a seulement 3 ans et qu’il est toujours en cours de développement.

V-3 WORKFLOW TEXTUEL DÉTAILLÉ

Passée l’étape complexe de l’installation, Nerfstudio peut être opéré à partir de l’invite de commande Miniconda3 qui, à l’opposé de l’installation, est très stable. De plus, une aide intégrée pour chaque commande est proposée. Le workflow qui est présenté dans cette section est largement détaillé avec une énumération de chaque étape, des illustrations, des justifications des choix de méthodes et des recommandations. Son résumé graphique a déjà été présenté à la Figure 22.

V-3.1 MISE EN FORME DES DONNÉES BRUTES

Comme déjà évoqué dans la préparation (cf. IV-3.2), la combinaison modélisation-acquisition a été choisie pour justement faciliter cette étape de mise en forme des données brutes. Le parseur de données étant déjà installé et prêt à recevoir les données de Polycam.

La commande générique est *« ns-process-data polycam --data {chemin_vers_données_ZIP} --output-dir {chemin_vers_dossier_sortie} »*. ns-process-data appelle le parseur, sa configuration est paramétrée pour recevoir des données Polycam. Les données brutes sont fournies au format zip (pas besoin de décompresser l’export venant de Polycam) et la commande demande finalement le dossier dans lequel les données parsées (mises en forme) doivent être stockées.

L’utilisateur doit prêter attention au chemin qu’il choisit pour renseigner le jeu de données et le dossier de sortie. Les chemins peuvent être absolus tant que le fichier de données et le répertoire

de sortie sont sur le même disque que l'environnement de travail de Nerfstudio. Les chemins peuvent aussi être relatifs mais cela nécessite de déplacer l'environnement de travail vers le dossier de données brutes (avec un simple changement de dossier ou cd). Le dossier de sortie est créé ou doit être accessible relativement depuis celui stockant les données. L'approche par chemins absolus semble plus indiquée pour éviter les erreurs (oublis des utilisateurs).

Les étapes de la mise en forme sont donc :

1) lancer le PowerShell Miniconda et activer l'environnement Nerfstudio;
2) déplacement (cd) vers le dossier contenant les données brutes;
3) lancer le parseur avec ns-process-data.

Dans le cas du jeu de données de référence, les données traitées sont organisées dans un dossier « processed » contenant les données du zip décompressées (données brutes) et un fichier transform.json. Ce dernier rassemble les 56 json contenant les paramètres de caméra en un seul endroit et calcule pour chacun la matrice de transformation résultante. Le dossier contient également quatre sous-dossiers contenant chacun les 56 images à 4 niveaux de résolution différents (downscales successifs par 2, 4, 8). Le downscaling est utile pour faire de l'optimisation dans certains pipelines d'entrainement (convergence plus rapide, résolution parallèle,…).

A la fin de cette étape, le dossier « processed » contient les données prêtes pour l'entrainement du modèle NeRF par Nerfstudio.

V-3.2 MODÉLISATION

C'est à cette étape que le workflow présenté diverge totalement des méthodes classiques de photogrammétrie. L'entrainement d'un modèle neuronal pour générer une représentation volumétrique de la scène est présenté dans cette section.

V-3.2.1 Choix de la méthode NeRF

L'état de l'art a déjà montré qu'il existait de nombreuses méthodes NeRF (cf I-5.7). Nerfstudio inclut par défaut 6 méthodes: le NeRF originel de Mildenhall et al. (2020) Instant-NGP, Mip-Nerf , TensoRF et surtout Splatfacto et Nerfacto qui sont les méthodes développées spécifiquement par Nerfstudio. Il contient aussi un support pour intégrer 16 autres méthodes venant de développeurs tiers.

Vu les problèmes rencontrés durant l'installation, seules les méthodes déjà incluses ont été considérées pour éviter de devoir rajouter des composants de sources extérieures. Cependant, les méthodes tierces sont très intéressantes et mériteraient plus d'investigations pour de futurs travaux (streaming, sémantique, nettoyage, éditeur,…).

Dans les méthodes inclues, Mip nerf et Instant-NGP ont déjà été discuté dans l'états de l'art. TensoRF représente son champ de radiance avec des tenseurs ce qui permet de fortement réduire l'empreinte mémoire. Splatfacto est la méthode conçue par Nerfstudio pour utiliser les Gaussian splatting comme entrainement du modèle. Ces différentes méthodes présentent chacune leur avantage mais ne correspondent pas à l'utilisation du NeRF voulue dans cette étude. Splatfacto est hors sujet puisqu'il utilise des géométries déjà connues. Nerfstudio recommande de ne pas utiliser la méthode originale NeRF car elle est sous-efficiente dans la plupart des cas. Tandis que Mip-nerf et Instant-NGP, grâce à leurs avantages, sont repris dans

Nerfacto. Pour ces différentes raisons la méthode utilisée est celle développée par Nerfstudio: Nerfacto. Elle reprend les avantages des autres et évite les risques de briser l'environnement.

Nerfacto est une méthode développée pour fonctionner avec des données réelles statiques, comme des façades de bâtiments par exemple. La méthode est largement discutée par Remondino (2023).

V-3.2.2 Paramètres de la commande d'entrainement

Pour lancer l'entrainement avec la méthode Nerfacto, il suffit d'utiliser un seule commande ns-train. Il faut cependant prêter attention aux paramètres de la commande. La simple commande *« ns-train nerfacto --data {chemin_vers_dossier_traité} »* permet de lancer l'entrainement avec tous les paramètres par défaut. Mais elle peut être largement personnalisée. La commande *« ns-train nerfacto –help »* indique qu'il existe de nombreuses autres options. Nerfacto, compte 147 options par défaut, notamment pour gérer le pipeline, la configuration hardware, la configuration du viewer, la gestion des données, l'optimisation, etc.

Les options principales qui ont été utilisées durant l'expérience sont listées ci-dessous.

1) L'aide : --*help* (affiche l'aide)

2) La gestion des fichiers : --*data* (pour fournir le dossier de données traitées) ,--*output-dir* (le dossier de sortie), --*relative-model-dir* (pour enregistrer le modèle en chemin relatif), --*load-dir* (pour charger un modèle réentraîné) et --*load-config* (pour charger un fichier .yaml (ceux-ci sont utilisés comme stockage des modèles temporaires ou terminés)).

3) La gestion de l'entrainement: --*method-name* (pour choisir sa méthode), --*timestamp* (pour effectuer un suivi temporel), --*vis* (pour sélectionner un viewer extérieur), --*start-paused* (pour mettre en pause ou redémarrer un entrainement).

4) La paramétrisation de l'entrainement: --*max-num-iterations* (nombre d'étapes (30000 par défaut)), --*steps-per-save* (nombre d'étapes entre les sauvegardes (2000 par défaut)), --steps-*per-eval-batch* (nombre d'étapes avant l'échantillon d'un batch de rayons (500 par défaut)), --*steps-per-eval-image* (nombre d'étapes avant l'évaluation d'une image (500 par défaut) et --*steps-per-eval-all-images* (nombre d'étapes avant l'évaluation de toutes les images (25000 par défaut).

5) La gestion du viewer: --*viewer.quit-on-train-completion* (quitter le viewer après l'entrainement (par défaut il reste ouvert continuellement), --*viewer.num-rays-per-chunk* (le nombre de rayons par chunk à rendre dans le viewer), --*viewer.websocket-port* (le port sur lequel le serveur du viewer est ouvert)

Dans le workflow général, on retiendra en priorité: l'option –data qui est la seule obligatoire. Il est recommandé de tout de même configurer le –*output-dir* car sinon il enregistre les résultats (.yaml et export) dans un dossier source de l'installation Nerfstudio, c'est-à-dire qu'il sera perdu au milieu des dépendances de Miniconda.

Pour jouer avec les performances du modèle, les options gérant les actions entre les étapes sont intéressantes. Il faut faire attention à toujours rester sous le –*max-num-iterations*. Mais de manière générale les paramètres par défaut des options sont correctement calibrés pour un grand nombre d'opérations.

Dans le cas du jeu de données de référence, la commande utilisée est la suivante:

« ns-train nerfacto --data {chemin_vers_dossier_traité} --output-dir {chemin_vers_dossier_sortie} --max-num-iterations 10000 --steps-per-eval-all-images 8000 --viewer.quit-on-train-completion False --viewer.websocket-port 7007 »

L'entrainement a été réduit à 10000 étapes (avec une évaluation de toutes les images à 8000) car la taille de la scène est assez restreinte et que le modèle converge bien avant les 30000 étapes par défaut. Le chemin de sortie a été configuré.

L'optimisation a été effectué durant 10000 itérations et a duré 5 minutes et 55 secondes et s'est terminée par l'ouverture du serveur soutenant le viewer Nerfstudio. La Figure 23 présente le viewer avec un rendu du jeu de données de référence généré par Nerfacto.

Figure 23 : Viewer Nerfstudio (viser) affichant un rendu du jeu de données de référence.

V-3.3 EXPORT DU NUAGE DE POINTS

Le support de données choisi pour faire la transition entre Nerfstudio et CloudCompare est le nuage de points. D'une part, il représente un produit de relevés très utilisé (coupe, visualisation,…) et d'autre part, parce que les MESH produits par Nerfstudio sont difficilement paramétrables (5 options, uniquement reconstruction de Poisson). De plus, la présence d'artefacts avec la méthode Nerfacto affecte fortement la qualité du MESH.

V-3.3.1 Ajustement de la bounding box (crop)

Avant l'export du nuage, il est obligatoire d'ajuster la bounding box du nuage avec l'outil crop de Nerfstudio. L'activation du crop se fait dans l'onglet « Control » du viewer de base. Dans le dossier *« crop viewport »* (dernier dossier) se trouve la boite *« enable viewport »* permettant de visualiser le crop par défaut. Une fois coché, un tableau d'ajustement apparait et permet de paramétrer le crop en fonction des axes locaux du viewer. Le nuage est manipulé en ajustant la taille, la position et la rotation du crop en fonction des trois axes X,Y,Z.

L'ajustement de la bounding box est très important car il donne déjà un alignement (a priori) au nuage de points. Le centre et l'orientation de la bounding box sont directement matérialisés dans la scène. Pour faciliter l'ajustement, il est recommandé de commencer par positionner le centre de la box au centre de la façade ou du bâtiment. Ensuite, les axes doivent être orientés avec les paramètres de rotations pour que l'axe X soit perpendiculaire à la façade et que le plan

ZY lui soit parallèle. Finalement, l'étendue est ajustée avec les paramètres d'échelles. La Figure 24 présente le jeu de données de référence après l'ajustement du crop. On remarque l'outil *« crop viewport »* activé en bas à droite, le système de coordonnées de la bounding box au centre de la chapelle et le système de coordonnées local de Nerfstudio en bas à gauche.

Figure 24 : Jeu de données de référence après ajustement et rognage dans le viewer de Nerfstudio

Il est à noter que cette étape n'est pas du tout ergonomique. Manipuler une bounding box en paramétrant une matrice numérique est très lent. De plus, le viewer est obligé de générer un rendu à chaque petit ajustement, l'utilisateur est donc à chaque fois obligé d'attendre quelques secondes pour observer le résultat. Il serait bien plus facile de manipuler une box si elle était matérialisée dans la scène et qu'elle pouvait être ajustée en faisant glisser les coins (comme on retrouve généralement dans les logiciels SIG et Luma AI).

V-3.3.2 Génération du PCD

Le viewer de Nerfstudio ne peut pas directement générer un PCD. Il comporte cependant un outil permettant de personnaliser une ligne de commande Nerfstudio. La commande peut être ensuite copiée/collée dans un PowerShell Miniconda pour réellement générer le PCD.

L'outil se trouve dans l'onglet « export » du viewer, dans le dossier « point cloud ». Avant de personnaliser la commande, deux options doivent obligatoirement être cochées : *« use crop »*, pour générer le PCD uniquement dans la bounding box et *« save in world frame »* pour conserver la mise à l'échelle de Polycam. Ensuite, l'utilisateur peut personnaliser le nombre du point du nuage, le filtre des outliers (retirer les points isolés), la méthode de calcul des normales et le dossier de sortie où sera stocké le PCD. Il est à noter que le nombre de points est personnalisable et permet de gérer la densité du PCD pour une même bounding box. Pour la mise en forme des données, il est recommandé d'utiliser un chemin absolu pour le dossier de sortie afin d'éviter les mêmes problèmes de chemins relatifs que ceux rencontrés à l'installation de Nerfstudio (cf.V-3.1).

Pour l'exemple, voici la commande qui a été générée pour le jeu de données de référence:
« *ns-export pointcloud --load-config {chemin_absolu_vers_modele\config.yml} --output-dir {chemin_absolu_vers_dossier_sortie} --num-points 1000000 --remove-outliers True --normal-method open3d --save-world-frame True --obb_center -0.2599999905 -0.0099999998*

0.0799999982 --obb_rotation 0.0000000000 0.0000000000 0.0000000000 --obb_scale 0.7200000286 0.7500000000 1.6900000572 »

Le nuage de points lors de l'export est donc d'un million de points à l'échelle réelle. Sa génération par l'installation Nerfstudio a duré 2 minutes 28 secondes. Il est à noter que l'ajustement de la bounding box (en comprenant son fonctionnement) a pris une quinzaine de minutes.

V-3.4 PROJECTION DANS CLOUDCOMPARE.

La projection est réalisée à l'aide de CloudCompare. Les étapes doivent reproduire la définition d'une vue orthographique (cf. IV-2.1). L'observateur est considéré comme placé à l'infini pour permettre aux lignes de projection de devenir parallèles et donc de conserver les dimensions de l'objet. En pratique cela signifie que le plan de projection (la future image) doit être aligné parallèlement à un des plans de l'objet.

V-3.4.1 Méthode d'extraction du plan de l'objet.

Trouver un plan dans un nuage de points est un large sujet et de nombreuses approches existent. CloudCompare propose plusieurs outils remplissant cette fonction :

- Fit to plane : utilise une analyse en composante principale (PCA) sur tous les points pour déterminer la normale d'un seul plan à partir des vecteurs propres et des valeurs propres (cloudcompare forum, 2010)
- qRANSACSD : un plugin configurant l'algorithme RANSAC (random sample consensus) appliqué pour la détection de forme (Shape détection). Le principe est de trouver les formes (shape detection) incluant le plus grand nombre de points cohérents à une distance seuil (consensus). Il commence toujours par échantillonner de manière aléatoire (random sample) des primitives de forme 3D. Il peut être paramétré pour détecter spécifiquement des plans (à partir de 3 points aléatoires). (CloudCompare, 2022). Il est à noter que, comme l'échantillonnage est aléatoire, les résultats sont non-reproductibles (CloudCompare, 2010).
- qFacets : un plugin permettant l'extraction de facettes (plan fracturé). Le principe est de découper le nuage en très petits plans (par récursion avec des arbres, ou systématiquement avec des octrees) puis de les fusionner en plans plus grands sur base de leur planéarité (distance et angle maximum ou propagation par fast marching). Le but étant d'obtenir des facettes recouvrant tout le nuage de points (CloudCompare,2016).

La documentation de CloudCompare s'étend largement sur l'utilisation et le fonctionnement de ces différentes méthodes et leur code est accessible en open-source. La Figure 25 présente un aperçu des trois méthodes sur le jeu de données de référence.

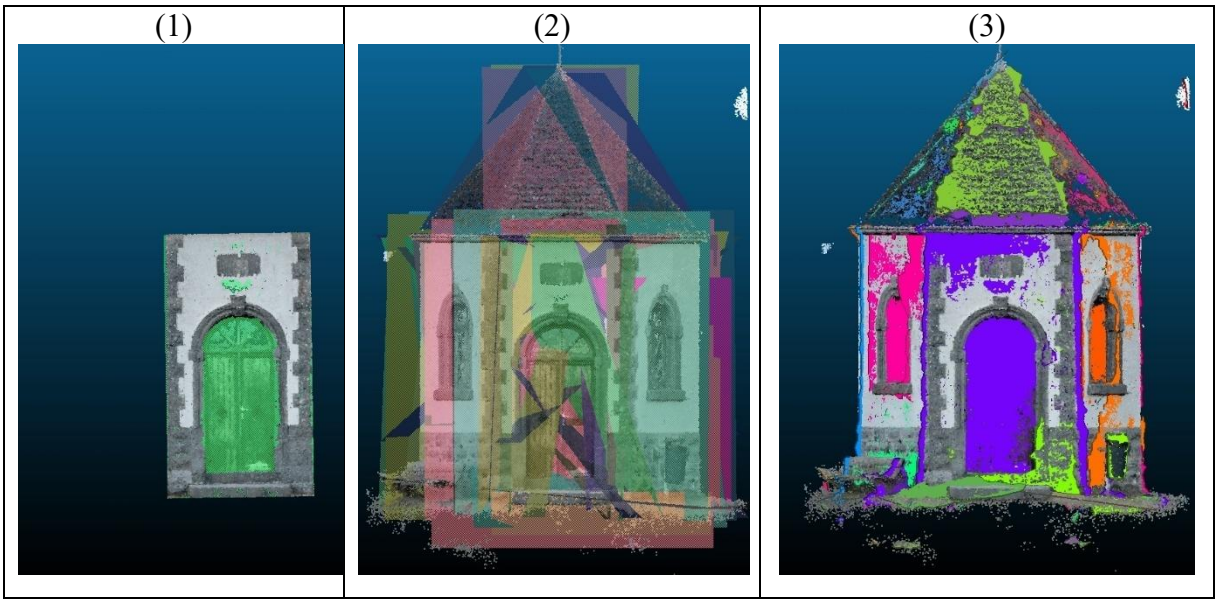


Figure 25 : Détection de plans avec une PCA (1), qRANSACSD en mode plan (2) et qFacets en mode fast marching (3)

« Fit to plane » se différencie fortement des deux autres méthodes car il détecte statistiquement un seul plan. Dès lors si le nuage n'a pas de plan de préférence (comme dans le cas de la chapelle hexagonale), le plan sera créé au barycentre de l'objet. Le résultat est donc plus fiable mais nécessite de découper et nettoyer le PCD au préalable pour appliquer Fit to plane sur la façade d'intérêt. Il est à noter que le découpage peut être effectué avec le crop de Nerfstudio comme avec l'outil de segmentation de CloudCompare. Le temps de calcul de Fit to plane est quasiment instantané si on ne compte pas le temps de nettoyage.

De leurs côtés, qRANSACSD et qFacets permettent de détecter des plans sur tout l'objet et donc de travailler avec des objets plus complets. Les deux méthodes présentent tout de même des différences. qFacets agglomère les plans ensemble donc les facettes générées apparaissent « collées » au nuage. qRANSACSD place des plans à partir des barycentres et des points cohérents et les englobe ensuite, les plans peuvent donc dépasser du PCD. Concernant le temps de calcul du jeu de données de référence, qRANSACSD est relativement rapide (4 minutes pour le jeu de données de référence) et qFacets est plus lent (9 minutes pour la méthode des kdtree), voir extrêmement lent (57 minutes pour la méthode des fast marching). Les temps de calcul sont fortement liés à la paramétrisation initiale des méthodes et du PCD analysé. Ces temps de références sont plus indicatifs que représentatifs. Une analyse de complexité des différents algorithmes serait nécessaire pour établir leur efficacité réelle. La paramétrisation de qRANSACSD semble tout de même plus aisée car le nombre de points cohérents minimum et la distance seuil sont des valeurs qu'on se représente plus facilement physiquement.

Aucune des trois méthodes n'est optimale dans tous les cas de figure. C'est à l'utilisateur de choisir la méthode correspondant à son cas. Fit to plane est quasiment instantané et précis si le PCD représente une façade seule, déjà bien nettoyée et tenante d'un seul plan. qFacets permet de détecter plusieurs plans et sera plus visuel dans le cas d'un objet complexe acquis avec un tour complet. Il ne détecte que des façades planes mais elles sont facilement sélectionnables si l'utilisateur veut réaliser plusieurs vues. qRANSACSD permet également de détecter plusieurs plans voir même d'autres primitives (comme des cylindres ou des cônes), il est généralement plus rapide que qFacets mais la superposition des plans rend leur sélection moins visuelle. Il est donc plus indiqué pour des scènes de tours complets incluant des détails complexes et pas

forcément planaires. Son temps de calcul réduit est également plus indiqué pour les scènes plus larges.

Dès lors, on préfère Fit to plane pour les PCD unis planaires, qFacets pour les PCD uniquement multi-planaires, qRANSACSD pour les PCD complexes et larges.

V-3.4.2 Alignement et Projection

Le plan de projection correspond au plan de la caméra de CloudCompare. Pour obtenir, une projection il faut donc aligner le plan de l'objet extrait avec la caméra CloudCompare et ensuite effectuer un rendu en désactivant la perspective (utilisateur placé à l'infini). Voici les étapes à suivre pour obtenir une projection de qualité :

1) Pré-traitement: Chargement du PCD dans CloudCompare et nettoyage des artefacts grossiers s'il y en a avec l'outil de découpe. Le but est surtout de faciliter le travail des méthodes d'extraction de plan. Par ailleurs, il est conseillé de travailler sur un clone pour éviter les mauvaises manipulations sur le nuage original.
2) Extraction du plan objet: En fonction du PCD et avec l'aide des méthodes vues précédemment.
3) Alignement: Le plan objet extrait est sélectionné dans l'arborescence. L'option *« Align camera»* du menu contextuel est uniquement disponible pour les plans. Elle permet d'aligner le plan de projection (la caméra) sur le plan de l'objet. Après l'alignement, il est obligatoire que la caméra ne soit plus modifiée (on ne peut plus bouger dans la scène). Pour éviter de faire des mauvaises manipulations, il est possible de sauvegarder la vue à l'aide de l'outil *« save viewport as object »* dans l'onglet « *display* » (ctrl-v).
4) (Optionnel) Découpage et deuxième nettoyage plus fin : La façade d'intérêt peut être redécoupée s'il est souhaité de conserver uniquement certains détails ou, par exemple, pour retirer une toiture ou des façades adjacentes. Il est également possible de nettoyer les derniers artefacts restants.
5) Élimination des points cachés : Le plugin « *Hidden point removal* » permet d'éliminer tous les points qui ne sont pas directement visibles depuis l'angle de vue de la caméra et selon un certain seuil. Cela permet d'éliminer les points en arrière pouvant contribuer à la couleur et fausser l'aperçu réel de la façade. Après cette opération, il est conseillé d'ajuster la taille des points et/ou la couleur du fond.
6) Mode perspective : Le mode perspective doit être désactivé, Cloudcompare rend alors la scène comme si sa caméra était placée à l'infini. Dans l'option « *set current view mode* » on sélectionne « *Orthographic projection* ».
7) Projection : CloudCompare génère un rendu de la vue actuelle. Comme le plan de l'objet est aligné et que le mode perspective est désactivé, il réalise de facto une projection orthographique de la façade. Dans l'onglet « *display* », on sélectionne « *render to file* ». L'outil doit être paramétrisé correctement pour conserver l'échelle : le zoom doit rester à 1, *« don't scale features »* doit être désactivé pour afficher les points à la taille visible, *« render overlay item »* doit être coché pour afficher la barre d'échelle. La Figure 26 présente la fenêtre de projection correctement paramétrée. Il est à noter que l'aspect du rendu peut être modifié avant la projection de celui-ci en jouant avec la taille des points, les shaders, le fond de plan, etc.

Dans le cas où la scène est multi-planaire ou complexe, l'utilisateur peut répéter le workflow pour toutes les façades d'intérêt. Il suffit de sélectionner un autre plan à l'étape de l'alignement (3) puis de créer un clone avant de passer aux étapes suivantes. La multiplication des clones évite de perdre des points aux étapes 4 et 5.

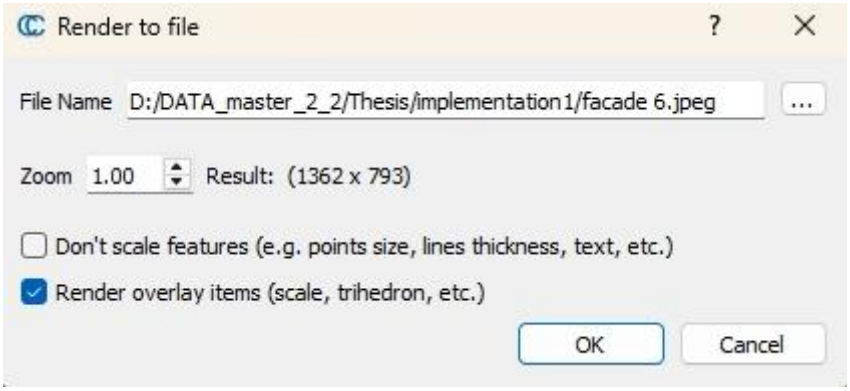


Figure 26 : Fenêtre de rendu de CloudCompare paramétrée pour la projection

Les vues orthographiques générées pour le jeu de données de référence sont présentées dans la section suivante (Figure 27 et Figure 28).

V-4 SORTIES ATTENDUES : WORKFLOW ET VUES FINALES

Cette section présente les sorties obtenues en appliquant le workflow au jeu de données de référence. Les vues orthographiques représentent une vue générale et une extraction des six façades de la chapelle. Chacune des vues est enregistrée au format .jpeg. L'archive fournie adjacente au mémoire conserve les sept vues orthographiques générées, ainsi que les sept nuages de points (.ply) dont elles sont extraites. Le workflow est quantifié, sur base des critères d'accessibilité, pour être rendu comparable avec ceux des autres expériences.

V-4.1 VUE ORTHOGRAPHIQUE DU JEU DE DONNÉES DE RÉFÉRENCE

Figure 27 : Vue orthographique générale de la chapelle (sortie de la première implémentation)

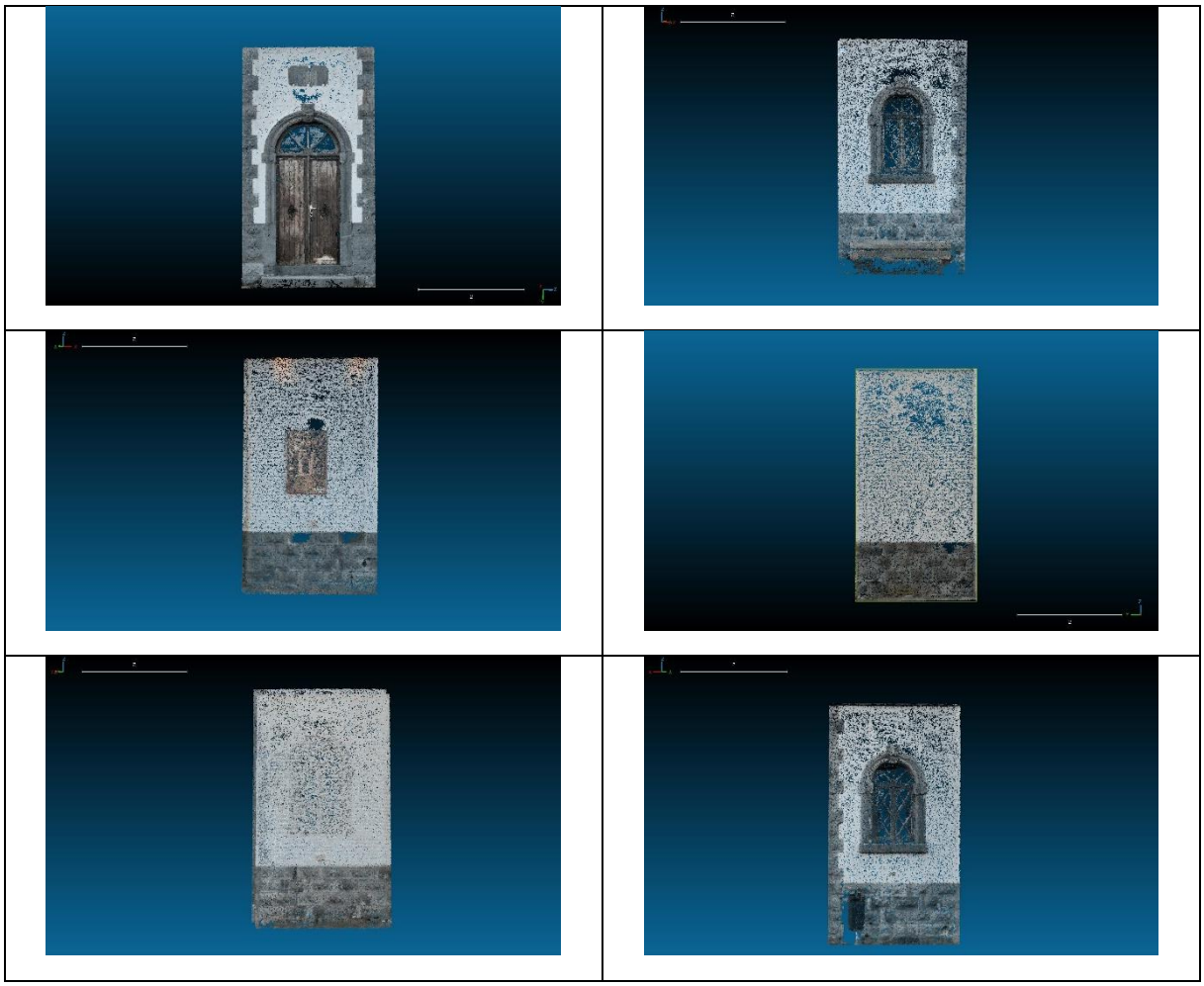


Figure 28 : Vues orthographiques isolant les six façades (dans le sens horaire en partant de la façade avant sud).

V-4.2 ANALYSE DU WORKFLOW TEXTUEL

Tableau 4 : Résumé et analyse du premier workflow.

Numéro d' étape	Intitulé	Nécessite une paramétrisation	Fausse manipulation possible	Feedback en cours de procédure	Feedback après la procédure	Comprendre les NeRF	Comprendre les reconstructions	Temps de réalisation (minutes)
1	Lancer le PowerShell Miniconda et activer l'environnement Nerfstudio.	×	×	×	√	×	×	1
2	Déplacement vers le dossier de données brutes	×	×	×	√	×	×	1
3	Créer la commande ns-process-data	√	√	×	×	×	×	5
4	Lancer le parser avec ns-process-data	×	×	√	√	×	×	1 à 5
5	Créer la commande ns-train	√	√	×	×	√	×	5
6	Lancer l'entrainement sur les données prétraitées avec ns-train	×	×	√	√	√	×	5 à 120

7	Ouvrir le viewer de Nerfstudio	×	√	×	√	×	×	1
8	Ajustement de la bounding box dans le viewer	√	√	√	√	×	√	15
9	Créer la commande ns-export avec l'outil du viewer Nerfstudio	√	√	×	×	√	√	2
10	Génération du PCD dans le PowerShell	√	×	√	√	×	×	3 à 10
11	Chargement du PCD dans CloudCompare	√	√	×	√	×	√	1
12	Nettoyage des artefacts de reconstruction	×	√	√	√	×	√	10
13	Extraction du plan objet	√	√	×	√	×	√	1 à 60
14	Alignement du plan objet	×	×	×	√	×	×	1
15	Découpage de la façade et dernier nettoyage	×	√	√	√	×	√	10
16	Elimination des points cachés	√	√	×	√	×	√	2
17	Passage en mode projection orthographique	×	×	×	√	×	×	1
18	Projection et export de la vue orthographique.	√	√	×	√	×	√	2
Total		9	11	6	15	3	8	77 à 252

NON : ×; OUI : √

V-5 RÉSULTATS INTERMÉDIAIRES

Les résultats de cette expérience seront comparés dans la partie résultats et serviront de base pour le cadre propre de la deuxième expérience (simplification progressive).

Cette expérience a démontré qu'il est possible d'obtenir une vue orthographique d'une façade en passant par une reconstruction à partir d'un NeRF. Le workflow dégagé comporte 18 étapes. En suivant celui-ci, un utilisateur expert du domaine peut produire des vues orthographiques (et des PCD) représentant des façades en 77 à 252 minutes selon les paramètres qu'il utilise. Dans le cas du jeu de données de références le temps de réalisation total est d'environ 90 minutes.

Le workflow a l'avantage de proposer de nombreux feedbacks visuels (6/18 pendant et 15/18 après). Cela est dû à la grande liberté de paramétrisation (9/18) qui est offerte en utilisant CloudCompare et en réalisant par soi-même les commandes Nerfstudio. Cependant cette liberté a un coût : plus de la moitié des étapes peuvent induire une mauvaise manipulation (11/18) car elles nécessitent souvent de comprendre les principes derrière le NeRF (3/18) et/ou les reconstructions 3D (8/18).

L'implémentation a montré que même s'il n'y a que deux logiciels à installer, la procédure d'installation de Nerfstudio est très instable et doit être vue au cas par cas. L'installation par un utilisateur nécessite donc qu'il possède une certaine connaissance en informatique (ligne de

commande, GitHub, gestion des fichiers, navigation dans l'invite de commande, notion de chemin absolu et abstrait).

À l'issue de la première expérience, plusieurs axes de simplification peuvent être dégagés pour la prochaine expérience. Premièrement, la réduction du nombre d'étapes induisant une possibilité de mauvaise manipulation est obligatoire même si cela implique une perte de liberté. Deuxièmement, le workflow est trop bas niveaux. Or, le public intermédiaire a une connaissance théorique des NeRF et des reconstructions 3D mais pas forcément de leur mise en œuvre. L'expérience deux doit donc s'éloigner du bas niveau en évitant les manipulations dans l'invite de commande, la gestion des fichiers et la manipulation précise de PCD. Troisièmement, il faut trouver une méthode permettant de distribuer l'installation de Nerfstudio pour contourner les difficultés liées à son installation.

En suivant ces axes de simplification, l'expérience deux s'oriente donc effectivement vers une interface utilisateurs de type « push button ». Ce qui signifie que l'utilisateur suit un workflow, en le comprenant, grâce à une succession de boutons. Il ne devrait plus être obligé d'exécuter lui-même les procédés derrière les étapes et il évitera un certain nombre d'erreurs car seuls les paramètres principaux sont disponibles et sont déjà pré-configurés.

CHAPITRE VI DEUXIÈME EXPÉRIENCE : WORKFLOW À DESTINATION D'UN PUBLIC INTERMÉDIAIRE

VI-1 CADRE PROPRE À L'EXPÉRIENCE ET PRÉSENTATION

Dans cette seconde expérience, un second workflow est établi. Il constitue une simplification du workflow issu de la première expérience. En accord avec la segmentation établie durant la phase préparatoire (cf. IV-4), le public intermédiaire a les connaissances pour opérer une interface de type « push-button ». Les étapes de production sont réalisables simplement en activant une série de boutons réunis dans une seule interface. Puisque le public comprend le fonctionnement théorique des NeRF et des SIG 3D, les étapes à suivre peuvent rester relativement proches du workflow 1 mais, leur exécution doit être simplifiée (plus haut-niveau) et la paramétrisation réduite à son essentiel. Surtout, le nombre d'étapes pouvant induire de fausses manipulations doit être réduit significativement.

Un résumé graphique du workflow issus de l'implémentation de la seconde expérience est présenté à la Figure 29.

Contrairement à celui de la première expérience, le workflow de la deuxième expérience est totalement linéaire. L'entièreté des opérations liées au Nerf (en orange) et à la projection (en vert) sont réalisées dans la même interface à l'exception de l'ajustement de la bounding box (en violet) qui se fait dans le viewer de Nerfstudio et qui constitue le seul apport de vérité terrain de l'utilisateur. Toutes les autres étapes pouvant impliquer des choix menant à une mauvaise manipulation ont été compactées pour suivre un workflow fluide et robuste comme présenté.

Techniquement, l'interface permet d'opérer les CLI de Nerfstudio et des processus similaires à ceux faits dans CloudCompare. L'utilisateur n'a plus à créer de commande et peut simplement ouvrir les données brutes avec une fenêtre de son explorateur de fichiers. Les autres commandes Nerfstudio sont directement préconfigurées avec ses informations et des paramètres par défaut. Pour la partie projection, l'interface ne conserve que les processus nécessaires identifiés dans le premier workflow. L'interface intègre des feedbacks à l'aide d'une fenêtre visuelle pour la projection et un terminal pour la partie NeRF. Des notifications d'erreurs et des checks permettent de bloquer l'utilisateur s'il est en train de faire une fausse manipulation.

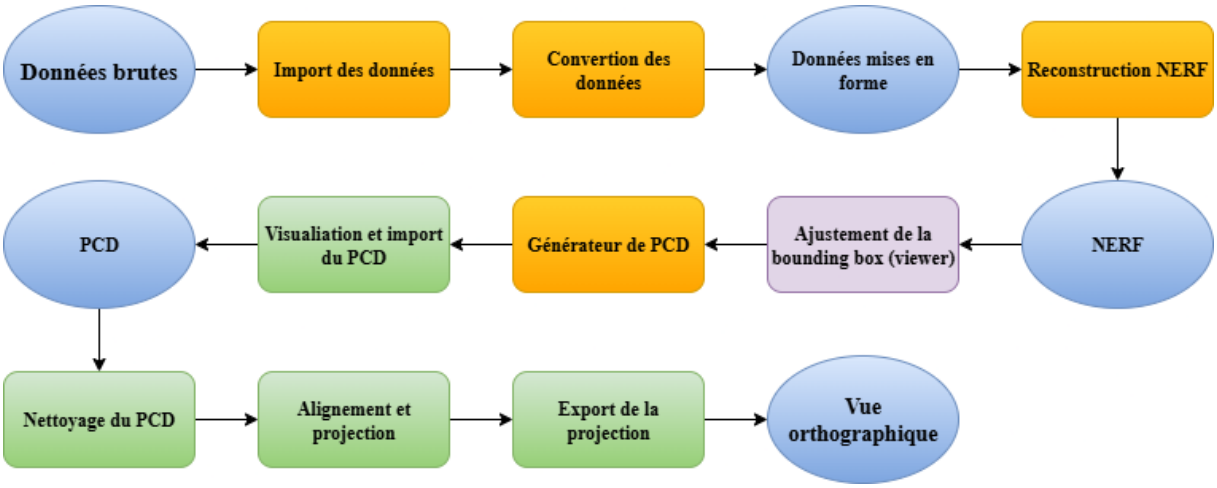


Figure 29 : Résumé graphique du workflow issus de l'implémentation de la seconde expérience.

VI-2 IMPLÉMENTATION

Cette section présente l'implémentation de l'expérience 2. Une version portable de cette implémentation est distribuée avec l'archive annexe du mémoire et porte le nom de « NSPCAssistant » pour « Nerfstudio et Polycam assistant ».

VI-2.1 INSTALLATION

L'implémentation continue de se baser sur Nerfstudio pour les traitements des NeRF mais la gestion de ses commandes et la projection sont réalisées dans l'interface utilisateur qui vient par-dessus. Pour cette implémentation, il est donc toujours nécessaire d'installer Nerfstudio et de paramétrer un environnement Miniconda capable de l'accueillir comme durant la première expérience (cf. V-2). L'avantage est que l'environnement Miniconda créé est tout à fait capable de permettre également l'installation des composants de l'interface. Le désavantage c'est que l'utilisateur doit effectuer la difficile installation, qu'il doit la demander à un expert ou que l'installation doit être rendue accessible en cloud entre plusieurs utilisateurs. Réaliser une encapsulation ou un package contenant Nerfstudio et ses dépendances n'est pas possible car il doit être relié à la carte graphique NVIDIA pour exécuter tiny-cuda-nn. À moins que la carte graphique et le hardware soient identiques, le portage pur et simple de toute l'application n'est pas possible. Il serait théoriquement possible de créer un installateur en préconfigurant plusieurs cartes graphiques mais c'est un travail beaucoup plus conséquent. Cette dépendance à l'installation de Nerfstudio est un des problèmes majeurs qui n'a pas pu être résolu au cours de l'expérience 2.

VI-2.1.1 Gestion des dépendances Python

L'installation comporte donc les étapes vues lors de la première expérience (cf. V-2). À cela s'ajoute l'installation des dépendances nécessaires à l'interface. L'interface est entièrement en Python 3.12.4 ce qui remplit les critères de Nerfstudio (minimum version 3.8). À noter que pour éviter la casse, il est recommandé de ne pas installer les dépendances de l'interface dans le même environnement que celui de Nerfstudio. Il est plus sécurisé de créer un environnement Conda spécifique à l'installation manuelle des dépendances ou de laisser le script launch_assitant.py (voir l'archive) installer automatiquement les dépendances requises dans l'environnement de base de l'ordinateur. Voici une liste présentant rapidement les dépendances requises :

1) Dépendances liées à la création de l'interface:
    a. *tkinter*: la dépendance principale qui permet de structurer et de créer l'interface, de lui ajouter toutes ses fonctionnalités responsives (widget) comme les menus, les boutons, le champ texte, les pop-up, etc. Il existe de nombreuses librairies permettant la création d'interfaces mais tkinter a le principal avantage d'être directement inclus avec Python.
    b. *ttkbootstrap*: une dépendance qui permet d'éditer le style visuel de tkinter et de lui donner un aspect plus moderne dans le style de bootstrap (javascript pour le web).
2) Dépendances liées au système (inclus avec Python):
    a. *Subprocess*: une dépendance fondamentale permettant de lancer des commandes externes à Python dans le système. Cette commande permet à l'utilisateur de lancer le CLI Nerfstudio
    b. *Threading:* une dépendance pour la gestion des processus asynchrones. Elle permet de lancer les CLI Nerfstudio dans des terminaux qui ne sont pas synchronisés avec l'interface pour éviter que celle-ci reste figée en attendant la fin des calculs NeRF pouvant être très long, voir même continu.
    c. *Datetime:* une dépendance pour intégrer un horodatage dans les feedbacks et le terminal Nerfstudio.
    d. *os:* une dépendance basique pour la gestion de fichiers dans le système. Principalement utilisée pour la navigation dans l'arborescence des fichiers.
    e. *Webbrowser*: une dépendance permettant le dialogue avec un navigateur web et qui permet notamment de lancer le viewer de Nerfstudio.
3) Dépendances liées aux calculs et à la visualisation:
    a. *Matplotlib*: une dépendance très utilisée pour la génération de graphiques (plot). Les graphiques peuvent être obtenus à partir de données 3D (PCD) ou 2D (projection). La dépendance donne les principaux outils de visualisation.
    b. *Numpy*: une dépendance également très utilisée pour la gestion des tableaux numériques (array). Elle permet de créer des tableaux mais aussi d'en extraire des statistiques. La plupart des calculs de matrices de rotation et de manipulation des PCD est effectuée grâce à numpy. Son indexation permettant la manipulation efficace de PCD lourds comportant des millions de points.
    c. *scikit-learn:* la dépendance open-source Scikit-learn est spécialisé dans la science des données et l'apprentissage par ordinateur. Elle contient de nombreux outils qui se basent et interagissent avec numpy et matplotlib (Sickit-learn,2025). Elle est principalement utilisée pour effectuer les calculs d'alignement avec une PCA pour la détection du plan objet et les proches voisins pour le nettoyage des points aberrants (outliers).
4) Dépendances liées au PCD:
    a. *pyntcloud*: une dépendance permettant l'import et l'export des PCD. Elle permet notamment de séparer couleur et coordonnées lors de la lecture.
    b. *pandas*: une dépendance qui propose une gestion des tableaux comme numpy. Il permet d'organiser les données sous forme de dataframe. Pyntcloud n'a été conçue que pour enregistrer ou lire à partir des dataframe et ne peut les traduire au format numpy utilisable pour les calculs. Dès lors pandas est obligatoire pour effectuer une conversion vers numpy.

Contrairement à l'installation de Nerfstudio, l'installation des dépendances peut facilement être automatisée. Le script « *launch_assitant.py* » et « *test_package.py* » contenue dans l'archive, permettent de tester la présence des dépendances et le cas échéant de lancer une installation à la volée. Dès lors, un nouvel utilisateur doit installer Nerfstudio puis simplement lancer l'interface pour obtenir une installation complète. Cependant, il peut choisir d'installer manuellement les dépendances si son installation de Nerfstudio est trop fragile.

VI-2.2 ARBORESCENCE DES SCRIPTS ET STRUCTURE DE L'INTERFACE.

L'archive contient tous les scripts Python et les fichiers nécessaires à son portage. L'arborescence reste simple avec une seule branche. Elle contient d'abord « *NSPCAssistantv8.py* », le script principal qui permet de construire l'interface, de la styliser, d'appeler toutes les fonctions et la définition des méthodes liées à la gestion des fichiers. En plus des dépendances requises, le script appelle deux autres scripts Python contenant les opérations liées à la projection (« *NSPC_2Dtool.py* ») et les opérations liées au NeRF (« *NSPC_3Dtool.py* »). Le découpage en 3 scripts permettant de travailler en parallèle sur ces 3 axes sans risque de casse. L'archive contient également des fichiers de support pour le portage : un « *READme.md* » classique pour mettre l'archive en contexte, un « *requirement.txt* » qui contient la liste des dépendances Python requises, le « *test_package.py* » pour vérifier qu'un environnement est conforme et le « *launch_assitant.py* » qui a pour but de lancer « *NSPCAssitantv8.py* » si l'environnement contient bien les dépendances requises (ou les installe si ce n'est pas le cas). Enfin l'archive contient le batch windows « *NSPCAssistant.bat* » qui permet par un simple double clic de lancer « *launch_assitant.py* » et donc de démarrer l'interface.

Au final, l'arborescence des scripts permet à un utilisateur ayant réalisé l'installation de lancer l'interface par un simple double clic comme n'importe quelle application Windows.

VI-2.3 PRÉSENTATION DE L'INTERFACE

Cette section présente la version finale de l'interface qui a été implémentée dans cette expérience. La Figure 30 donne un aperçu général de l'interface tandis que les Figure 31 et 32 effectuent un zoom sur les boutons des chaines de traitements principales. Il est difficile de se représenter tous les détails et de se rendre compte de l'interactivité avec de simples illustrations, c'est pourquoi le lecteur est invité à lancer et expérimenter l'application fournie avec l'archive (même sans l'installation Nerfstudio). Les PCD des façades sont fournis et peuvent permettre de tester les traitements de projection.

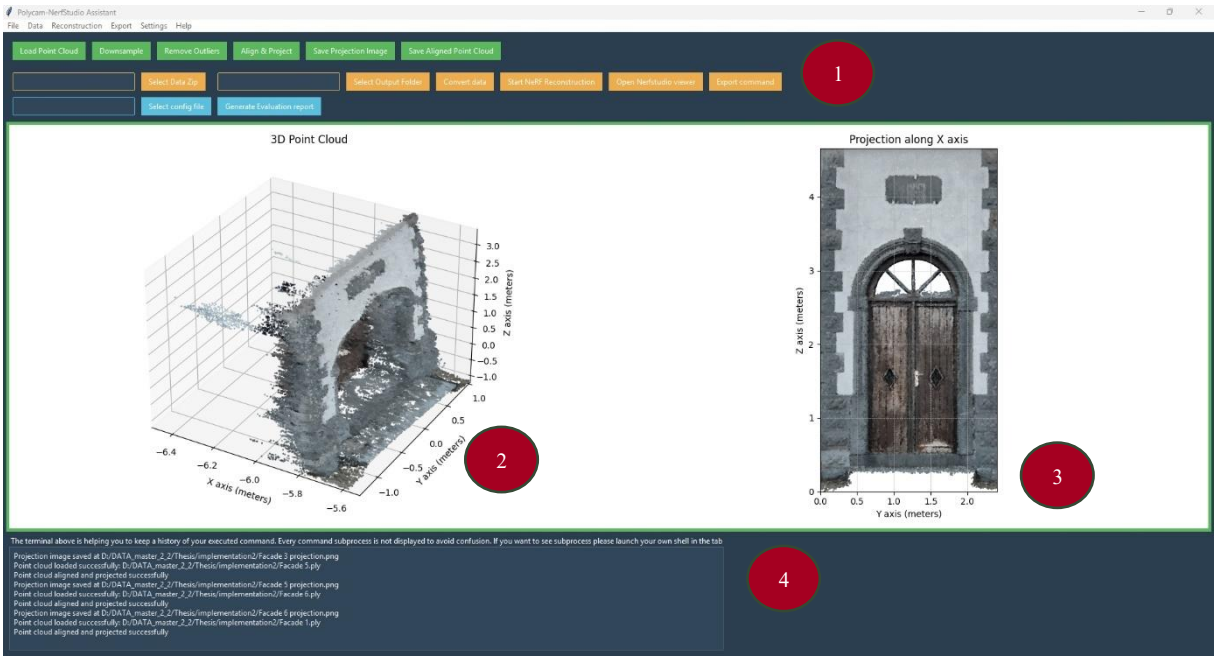


Figure 30 : Aperçu général de la version finale de l'interface NSPCAssistant implémenté lors de la deuxième expérience

L'interface est contenue dans une fenêtre tkinter Windows classique qui peut être déplacée, fermée et redimensionnée. La Figure 30 permet d'observer les principaux composants visuels de l'interface.

En (1) : les chaines de boutons qui permettent de suivre le workflow et d'effectuer les traitements. La chaine de la partie projection (anciennement la partie qui était effectuée sur CloudCompare) est en vert. Tandis que la chaine de la partie NeRF est en orange. La chaine bleue, est une chaine permettant de générer un rapport d'évaluation rendant compte de la qualité des NeRF. Cela permet à l'utilisateur de quantifier la reconstruction des NeRF, la chaine sera donc surtout utilisée dans la partie mise en situation.

En (2) (partie gauche de la fenêtre de visualisation principale) : un graphique 3D (matplotlib) permettant d'observer le PCD issu du NeRF puis un aperçu des traitements qu'il subit.

En (3) (partie droite de la fenêtre de visualisation principale) : un graphique 2D (matplotlib) qui fournit un aperçu de la projection avant son enregistrement.

En (4) : un terminal de feedback qui renseigne l'utilisateur des opérations effectuées ou en cours. La fonction feedback peut être appelée par n'importe quel traitement pour écrire (souvent avec un horodatage) ce que l'interface est en train de réaliser.

Les onglets situés au-dessus des chaines de boutons, contiennent des fonctions plus avancées, en cours de développement ou des paramètres. Notamment l'onglet file > Nerfstudio command prompt qui permet d'ouvrir une console Windows avec l'environnement Nerfstudio. Ce dernier est déjà chargé et prêt à recevoir des commandes personnalisées. Dans l'onglet settings, on trouve miniconda3_path qui permet d'indiquer la localisation de l'installation Nerfstudio de l'utilisateur. Les autres fonctions des onglets comprennent le workflow des boutons et des fonctions en cours de développement.

VI-2.4 NOTE SUR LES VERSIONS PRÉCÉDENTES

La réalisation de l'implémentation a beaucoup évolué au fil des retours des utilisateurs et du promoteur du mémoire. Pour garder la section la plus concise possible, seule la version finale est présentée mais elle garde toutes les fonctions de ses versions précédentes. Les versions précédentes étaient surchargées de boutons pour faire face à plusieurs cas de figure. Suite aux différents retours, seuls les boutons essentiels au workflow par défaut ont été conservés. Les boutons et options qui n'ont pas été conservés sont toujours disponibles dans l'implémentation finale mais sont regroupés dans les menus supérieurs. L'expérience étant déjà concluante, certaines options en cours de développement n'ont pas été achevées, ce qui explique la présence de certaines fonctions vides (placeholder) dans les menus mais qui pourraient être mises à jour si le développement reprenait.

VI-3 WORKFLOW TEXTUEL DÉTAILLÉ

Les scripts Python de l'interface font plus d'un millier de lignes, il est beaucoup plus simple et intuitif d'expliquer uniquement les calculs et fonctions principales au fil du workflow en suivant les chaines de boutons plutôt que de revoir le code ligne par ligne et d'ensuite en tirer le workflow. Cette section est donc à la fois une explication du workflow à suivre et une justification théorique et pratique des fonctions effectuées à chaque étape.

VI-3.1 CHAINES DE TRAITEMENTS NERF

La première chaine de traitement (Figure 31) permet de remplacer les étapes qui étaient précédemment effectuées dans l'invite de commande. Elle permet d'utiliser le CLI de Nerfstudio à partir de simples boutons et de ne plus devoir paramétrer soi-même les commandes avant de les lancer dans l'invite. Ce changement permet de gagner énormément de temps et également de réduire les possibilités de mauvaises manipulations. En somme, l'utilisateur n'écrit plus de commandes par lui-même.

Figure 31 : Boutons de la chaine de traitement NeRF dans l'interface finale

VI-3.1.1 Sélection des données brutes et du dossier de sortie :

Une des principales difficultés et perte de temps liées au CLI de Nerfstudio est la gestion des chemins d'entrée et de sortie. En plus de devoir faire attention entre chemins absolus et relatifs, chaque commande demande d'aller chercher le chemin des données dans les gestionnaires de fichiers. Ce qui alourdit la création de commandes déjà assez longue. Les deux boutons *« select data zip »* et *« select output folder »* invoquent chacun une fonction qui utilise *« filedialog »*, une bibliothèque native de tkinter permettant d'ouvrir le gestionnaire de fichiers et de sélectionner en un clic le fichier désiré. Le chemin est alors enregistré dans une variable de l'interface et pourra être utilisé dans la création de commande.

« Select data zip » permet d'enregistrer le chemin vers les données brutes et *« select output folder »* le chemin vers le dossier de sortie. Il est à noter que pour le premier, seul un chemin vers un fichier avec une extension .zip est autorisée, ce qui ajoute déjà un détrompeur pour l'utilisateur.

L'opération qui prenait quelques minutes auparavant ne prend que quelques clics en plus d'éviter les erreurs.

VI-3.1.2 Mise en forme des données (convert data)

Le bouton *« convert data »* permet de configurer et lancer la *commande « ns-process-data »* pour la mise en forme des données brutes.

Cette étape intègre une des mécaniques principales de l'interface: l'utilisation du CLI Nerfstudio comme sous-processus. La bibliothèque *« subprocess »* native de Python permet de générer des sous-processus pour exécuter des commandes externes. Un sous-processus peut notamment exécuter une commande qui ouvre une nouvelle fenêtre Miniconda 3 dans laquelle l'environnement Nerfstudio est actif. Il suffit alors de faire suivre une commande Nerfstudio pour qu'elle soit exécutée. Les traitements pouvant être assez longs (voire continus avec le viewer), *« subprocess »* est géré de façon asynchrone. Il est exécuté dans un thread séparé (de la bibliothèque native *« threading »*) pour permettre à l'interface de ne pas se figer en attendant la fin du processus.

Dans le cas du bouton « *convert data* », voici la commande qui est lancée en sous processus:

« command = fr'{self.miniconda_path}\Scripts\activate.bat && conda activate nerfstudio && ns-process-data polycam --data "{data_zip_file}" --output-dir "{output_folder}" && exit' »

Le sous processus commence par lancer le batch natif de Miniconda pour lancer une fenêtre, il active l'environnement nommé Nerfstudio puis il lance la mise en forme des données avec les chemins absolus fournis par l'utilisateur. Quand la mise en forme est terminée, la commande se termine et ferme la fenêtre qui avait été ouverte par Miniconda.

Concrètement lors du clic, l'utilisateur voit une invite de commande s'ouvrir et exécuter automatiquement la commande « *ns-process-data »*. La fenêtre se ferme quand le traitement est terminé. Il est à noter que la commande ne se lance pas si l'utilisateur n'a pas rempli les champs. Un feedback est envoyé au terminal pour notifier la réussite de l'opération.

En terme d'accessibilité, l'étape prend toujours de 1 à 5 minutes (2 minutes pour le jeu de données de référence) mais l'étape de création de la commande Nerfstudio a été supprimée, ce qui fait gagner environ 5 minutes et évite les fausses manipulations.

VI-3.1.3 Reconstruction NeRF,

Le bouton *« Start NeRF reconstruction »* se base sur le même principe que l'étape précédente mais envoie une commande *« ns-train »*. Voici une commande type qui est envoyée:

« command = fr'{self.miniconda_path}\Scripts\activate.bat && conda activate nerfstudio && ns-train nerfacto --data "{output_folder}" && exit' »

La commande utilise automatiquement le dossier de sortie dans lequel les données ont été mises en forme comme données d'entrainement. Il est aussi à remarquer que la commande est celle par défaut et n'a pas été paramétrisée comme durant la première expérience. Ne pouvant pas s'adapter au cas par cas, il a été choisi de conserver la commande la plus basique et donc de faire confiance au paramètre choisi par les créateurs de Nerfstudio pour s'adapter à la plus grande variété de données.

Comme précédemment, le temps de réalisation de l'étape reste théoriquement similaire à celui de la première expérience (de 5 minutes à 2 heures selon les jeux de données) mais

l'étape de création de la commande a été supprimée. Cependant en pratique, les utilisateurs observeront des traitements plus longs car le nombre d'étapes par défaut de Nerfstudio est de 30 000 et qu'on ne peut le configurer (contrairement à la première expérience avec 10 000 étapes). Par exemple, dans le cas du jeu de données de référence, le temps de calcul est passé de 6 à 22 minutes.

À la fin de l'entrainement, Nerfstudio ouvre le serveur du viewer sur le port local 7007, la fenêtre ne se ferme donc pas et le sous processus continue en arrière-plan. La fenêtre ne se ferme que si l'utilisateur la quitte manuellement en utilisant le raccourci CTRL + C.

VI-3.1.4 Ouverture du viewer et ajustement de la bounding box.

Le bouton « Open Nerfstudio Viewer » permet d'ouvrir une page du navigateur web à l'adresse locale 7007 grâce à la bibliothèque « web browser ». Une fois le viewer ouvert, les mêmes étapes qu'à l'expérience 1 peuvent être réalisées en ajustant une bounding box à l'objet d'intérêt puis en créant la commande permettant de générer le PCD associé (cf. V-3.3).

Comme l'analyse des résultats le montrera, l'interface a des difficultés à projeter des scènes complexes ou multi planaires donc la box doit être ajustée pour extraire un seul plan de la façade. Sinon les étapes et leur temps de réalisation sont similaires (15 minutes).

VI-3.1.5 Génération du PCD

Le bouton « Export command » ouvre une fenêtre contextuelle qui contient un champ textuel. L'utilisateur n'a plus qu'à entrer la commande d'export générée par le viewer de Nerfstudio à l'étape précédente. La commande est alors lancée en sous processus comme ns-train ou ns-process-data. Le temps de réalisation est identique à celui de l'expérience 1 pour un même jeu de données (2 minutes dans le cas du jeu de donnée de référence).

VI-3.2 CHAINE DE TRAITEMENT POUR LA PROJECTION

La deuxième chaine de traitement (Figure 32) suit les étapes de projection précédemment réalisées dans CloudCompare. L'intérêt est de concentrer toutes les opérations dans une seule et même interface ainsi de réduire la liberté de l'utilisateur et donc d'éviter les fausses manipulations. Les fonctions principales *(« Load Point Cloud »* et *« Align & Project »*) ne comportent plus d'option de paramétrisation, ainsi les résultats produits à partir d'un même PCD sont identiques pour tous les utilisateurs.

Figure 32 : Boutons de la chaine de traitement pour la projection dans l'interface finale

VI-3.2.1 Chargement et visualisation du PCD

Le bouton *«Load Point Cloud »* permet de charger un PCD (celui produit par Nerfstudio) dans l'interface. Techniquement, l'interface demande à l'utilisateur le chemin vers le PCD grâce au module *« filedialog »* de tkinter (support des extensions .pcd et .ply), puis le nuage est importé en tant que variable en utilisant la dépendance *« PyntCloud »*, qui permet de lire la plupart des formats de nuages de points et de les convertir en dataframe pandas (coordonnées + couleurs). Toutes les fonctions suivantes utilisent des arrays (tableaux) numpy car leur structure homogène est plus compacte et donc plus rapide pour les calculs. C'est pourquoi, le dataframe est directement décomposé en deux tableaux numpy, un pour la couleur et un pour les coordonnées. Les deux tableaux sont utilisés pour générer un visuel

(plot) grâce à la dépendance « *matplotlib* ». Le graphique est en trois dimensions avec des axes et une grille, l'échelle des axes n'est pas recentrée pour permettre à l'utilisateur de se rendre compte de la position du centre de la bounding box. Un feedback est envoyé dans le terminal quand le PCD est chargé et que le plot est affiché.

Concrètement, l'utilisateur doit simplement sélectionner le PCD dans la fenêtre de gestionnaire de fichier qui s'ouvre quand il clique sur le bouton, ensuite le plot affiche le PCD et l'utilisateur reçoit une notification pour signifier que le nuage est bien chargé (ou non).

VI-3.2.2 Option de nettoyage des nuages.

Deux options sont disponibles pour nettoyer le PCD, *« Downsample »* et *« remove outliers ».* Ce sont des options qui ne sont pas obligatoires pour obtenir une projection mais qui permettent de rendre le nuage plus léger (accélération des calculs et des rendus) ou plus cohérent (suppression des points aberrants).

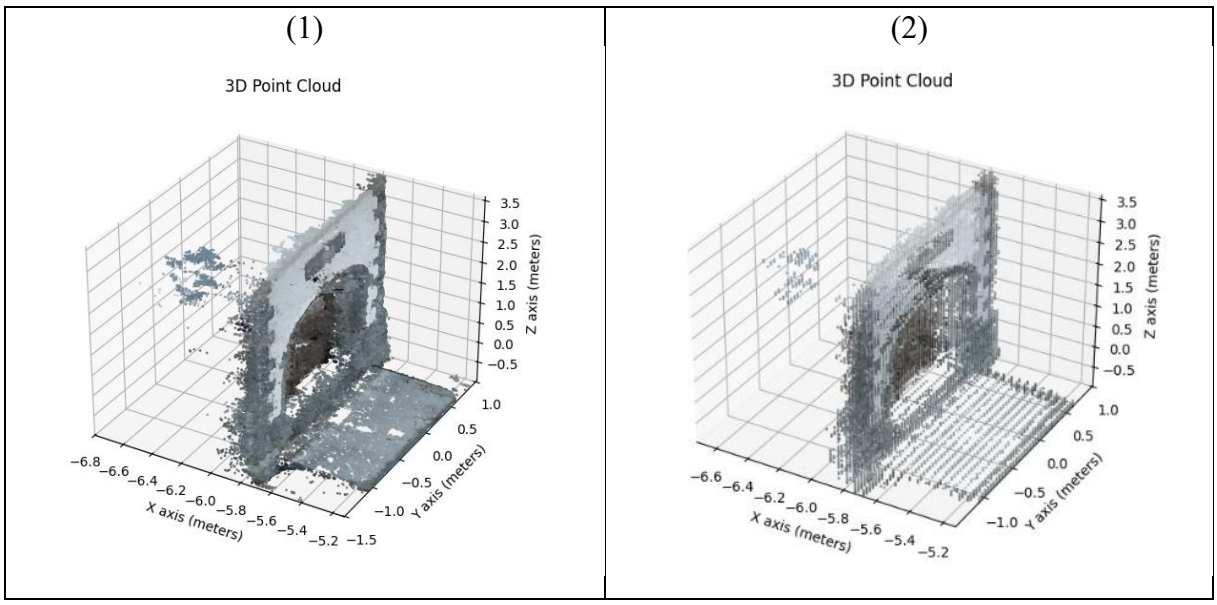


Figure 33 : Graphique de visualisation du nuage de points de la façade principale, avant (1) et après (2) nettoyage automatique (nettoyage volontairement grossier pour l'illustration).

Premièrement, « *downsample* » permet d'effectuer un sous-échantillonnage sur base d'une grille de voxels qui a pour but de réduire le nombre de points du nuage (et donc sa densité). L'implémentation est réalisée à l'aide de numpy et se base sur celle de la documentation de Point cloud library (PCL, 2025). À partir du point minimal du PCD, on divise l'espace en voxels d'une certaine taille. Ensuite, les voxels contenant au moins un point sont répertoriés. Seul un point situé au centre de chaque voxel répertorié est conservé pour former un sous échantillon du nuage original et créer un nouveau nuage.

Deuxièmement, « remove outliers » permet de supprimer les points aberrants sur base de la densité locale. Si le point n'est pas situé dans une zone assez dense (outliers), il n'est pas considéré comme cohérent avec le reste du nuage et est donc supprimé. L'implémentation est réalisée à l'aide de numpy et Scikit-leran, et se base sur celle de la documentation de Point cloud library (PCL, 2025). Techniquement, le procédé utilise un algorithme de détection des plus proches voisins (Le « NearestNeigbors() » de Sickit-learn) pour déterminer les n voisins que possède chaque point, n étant un nombre de voisins défini par l'utilisateur. Ensuite la distance moyenne entre un point et ses voisins est calculée. Grace à une moyenne de toutes

ces distances et d'un écart type défini par l'utilisateur, un seuil est établi et permet de séparer les points cohérents (inliers) et aberrants (outliers).

Les deux options ne sont pas obligatoires, c'est pourquoi une légère paramétrisation est permise à l'utilisateur: la taille des voxels définit pour le sous-échantillonnage (l'importance de la réduction), le nombre de voisins et l'écart type pour la suppression de points aberrants (dureté du seuil).

VI-3.2.3 Alignement et Projection

Les 5 dernières étapes du workflow précédent sont effectuées en une seule fois grâce au bouton « Align & Project ». La série de traitements reste similaire (extraction du plan objet, alignement puis projection) mais l'utilisateur n'a plus besoin d'effectuer aucune opération manuellement. Cela permet d'éviter une importante source de mauvaises manipulations et ne demande aucune connaissance technique, mais cela a le désavantage d'être plus exigeant en termes d'entrée. Dans CloudCompare, l'utilisateur pouvait choisir la méthode d'extraction adaptée à la scène et puis validait le plan objet sélectionné (apportant une certaine connaissance de la vérité terrain). Dans cette implémentation, seul le cas de figure d'une scène uni planaire a été pris en charge puisque les scènes multiplanaires ou complexes nécessitent une paramétrisation et une sélection du plan.

Extraction du plan objet

Seul le cas uniplanaire est pris en compte, l'implémentation utilise donc une PCA similaire à celle de l'opération *« fit to plane ».* Scikit définit la PCA comme une technique linéaire de réduction de dimension, elle est utilisée pour décomposer des jeux de données en une série de composantes qui permettent d'expliquer la majeure partie de la variance (Scikit-learn, 2025).

Le plan est ici représenté uniquement par sa normale, L'implémentation se base donc sur l'estimation des normales de la Point Cloud Library (PCL, 2025) avec numpy pour les calculs et Scikit pour la PCA. Pour réaliser l'estimation, le nuage est d'abord centré au centroïde, puis la matrice de covariance et une PCA (extraction en 3 composantes) sont calculées. La direction de la normale est donnée par le vecteur propre qui a la plus petite valeur propre. En effet, les dimensions du plan varient beaucoup plus en largeur et en longueur mais beaucoup moins en profondeur, sélectionner la plus petite valeur propre signifie sélectionner la dimension qui varie le moins, donc la profondeur (perpendiculaire au plan).

Alignement et projection

Pour réaliser la projection, le but est d'aligner la normale du plan objet avec l'axe X de référence. Grace à cet alignement, il est possible de tracer un graphique seulement en projetant les points à partir de leurs coordonnées en Z et Y puisque les X sont alignés. Le but est donc de calculer une matrice de rotation pour passer du référentiel local du plan extrait dans le référentiel orthonormé. Pour ce faire on utilise numpy, on commence par reconstruire le système référentiel local à partir de la normale extraite précédemment qui devient l'axe X: normalisation de X (*np.linalg.norm*) puis, projection de Y sur le plan perpendiculaire (*np.array* et *np.dot*) à X et création de Z par produit vectoriel de X et Y pour obtenir un système droit (*np.cross*). La matrice de rotation est composée des directions des trois axes (*np.stack*) qui ont été transposés. La fonction « *dot »* de numpy permet d'appliquer la transposée au nuage original et de créer un nouveau nuage ou la normale du plan objet est alignée avec l'axe des X, le plan formé par les axes ZY étant parallèle au plan objet.

La projection est effectuée par matplotlib, simplement en créant un graphique bidimensionnel des points du nouveau nuage à partir des coordonnées des points en Z et Y uniquement. Des exemples avec le jeu de données de référence sont présentés dans la section suivante (VI-4).

L'implémentation de l'alignement et de la projection est solide mais elle a pour conséquence que la vérité terrain injectée précédemment par l'utilisateur lors de la sélection du bon plan, ne peut plus se faire que lors de l'ajustement de la bounding box. De ce fait, il n'est plus possible de traiter des scènes complexes ou multiplanaires car le cadre en 3 axes représentant un plan essaye de s'adapter à toute la scène et donc de convenir à tous les plans en même temps. (comme le fit to plane qui ne fonctionne que sur un plan prédécoupé).

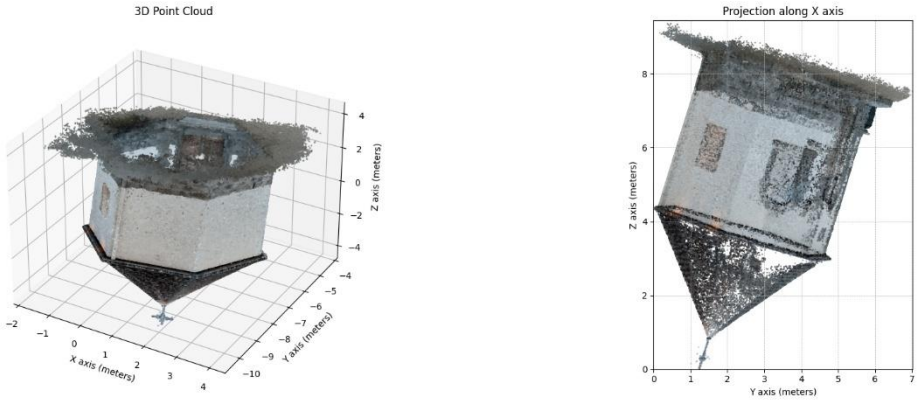


Figure 34 : Tentative d'alignement sur une scène multiplanaire menant à une projection erronée.

VI-3.2.4 Sauvegarde des résultats

Les bouton *« save projection image »* et *« save aligned point cloud »* permettent d'enregistrer les résultats. Le nuage projeté peut être exporté en réalisant l'opération inverse de l'import avec Pyntcloud. Le graphique de la projection peut être exporté en utilisant la fonction *savefig* de matplotlib. Un feedback est envoyé au terminal et à l'utilisateur.

VI-4 SORTIES ATTENDUES : WORKFLOW ET VUES FINALES

Comme pour la première expérience, cette section présente les sorties obtenues en appliquant le workflow au jeu de données de référence. Les vues orthographiques représentent l'extraction des 6 façades de la chapelle. De même, le workflow est quantifié, sur base des mêmes critères d'accessibilité que durant la première expérience. Les résultats sont aussi disponibles dans l'archive. Les vues sont enregistrées au format .png dans le sous dossier implementation2. L'implémentation proprement dite se retrouve dans le sous dossier « NSPCAssistant_portage ».

VI-4.1 VUES ORTHOGRAPHIQUES DU JEU DE DONNÉES DE RÉFÉRENCE

Figure 35 : Expérience 2 : Vues orthographiques isolant les 6 façades (dans le sens horaire en partant de la façade avant sud).

VI-4.2 ANALYSE DU WORKFLOW TEXTUEL

Tableau 5 : Résumé et analyse du deuxième workflow.

Numéro d'étapes	Intitulé	Nécessite une paramétrisation	Fausse manipulation possible	Feedback en cours de procédure	Feedback après la procédure	Comprendre les NeRF	Comprendre les SIG 3D	Temps de réalisation (minutes)
1	Sélection des données brutes	×	×	∨	∨	×	×	1
2	Sélection du dossier de sortie	×	×	∨	∨	×	×	1
3	Conversion des données brutes	×	×	×	∨	×	×	1 à 5
4	Lancer l'entrainement sur les données prétraitées avec ns-train	×	×	∨	∨	∨	×	5 à 120
5	Ouvrir le viewer de Nerfstudio	×	×	∨	∨	×	×	1
6	Ajustement de la bounding box dans le viewer	∨	∨	∨	×	×	∨	15
7	Créer la commande ns-export avec l'outil du viewer Nerfstudio	∨	∨	×	×	∨	∨	2
8	Lancer le générateur de PCD	×	×	∨	∨	×	×	3 à 10
9	Charger le PCD généré dans l'interface	×	×	∨	∨	×	×	1
10	Nettoyage automatique du PCD	∨	∨	×	∨	×	∨	3 à 10
11	Alignement automatique et Projection du PCD	×	×	×	∨	×	×	1 à 5
12	Export de la vue orthographique.	∨	×	×	∨	×	×	1 à 5
Total		4	3	7	10	2	3	35 à 176

NON : ×; OUI : ∨

VI-5 RÉSULTATS INTERMÉDIAIRES

VI-5.1 COMPARAISON AVEC LA PREMIÈRE EXPÉRIENCE.

La comparaison des analyses des workflows textuels entre la première et la deuxième expérience permet de démontrer que les axes de simplification ont été rencontrés.

En effet, le nombre d'étapes a été réduit (de 18 à 12) mais surtout la possibilité d'effectuer des mauvaises manipulations a été réduite de 11/18 à 3/12 comme attendu. Les trois étapes qui peuvent induire de fausses manipulations sont l'ajustement de la bounding box, la paramétrisation de la commande de génération du PCD et le nettoyage des points. Ce sont d'ailleurs les trois seules étapes avec l'enregistrement de la vue orthographique qui nécessitent une paramétrisation.

Concernant les feedbacks, l'interface est un peu moins bonne que son homologue CloudCompare pour les feedbacks en cours de procédure (7/12) car les opérations sont réalisées en arrière-plan et le résultat est affiché après (impliquant parfois des freezes). En

revanche, elle est excellente en feedback post procédure (10/12) grâce au terminal qui affiche également des feedbacks pour les opérations liées au NeRF. Les deux seules étapes non notifiées, sont encore une fois les deux étapes réalisées dans le viewer.

L'interface nécessite également moins de connaissances théoriques, surtout concernant les SIG 3D qui passe de 8 étapes sur 18 impliquant la compréhension des SIG 3D à 3 étapes sur 12. La compréhension des NeRF étant assez similaire (de 3/18 à 2/12).

Le temps de réalisation théorique a également été fortement réduit passant d'une fourchette de 77 à 252 minutes, à une fourchette de 35 à 176 minutes. Le temps de réalisation pour le jeu de données de référence étant passé de 90 à 50 minutes.

Au niveau qualitatif, les vues orthographiques sont bien plus lisibles grâce à la graduation des axes et le fond blanc. Malheureusement, il n'est plus possible de rendre la projection de tour complet (scènes multiplanaires ou complexes). De plus, l'implémentation souffre d'une erreur déjà présente avec CloudCompare (mais qui pourrait être ajustée) avec l'inversion de normale. Lors de l'estimation de la normale, il peut arriver que sa direction soit inversée avec pour conséquence de projeter le plan à l'envers par rapport au système d'axes (voir façades 2 et 5, Figure 35). C'est un problème purement esthétique qui n'empêche pas la lecture et les mesures.

On remarque par ailleurs que le découpage à l'aide de la bounding box est très sensible et une mauvaise paramétrisation peut se répercuter sur les résultats. Si on rajoute à cela la nature continue des NeRF pouvant prêter à confusion visuellement (entre la réalité d'un mur solide et le rendu qui est une accumulation), il est assez difficile de générer des PCD corrects. Par exemple, l'aspect parcellaire de la projection des façades 5 et 6 est dû à l'extraction d'un PCD trop fin ou mal orienté.

VI-5.2 AXE DE SIMPLIFICATION POUR LA TROISIÈME EXPÉRIENCE

En analysant ces changements, il est constaté que ce sont surtout les étapes qui sont effectuées dans le viewer Nerfstudio qui sont les plus susceptibles d'introduire des erreurs; ceci étant corrélé avec le fait qu'elles nécessitent la compréhension à la fois du NeRF et des SIG 3D. Cette étape est pourtant nécessaire pour que l'utilisateur puisse injecter la quantité minimale de vérités terrains (la connaissance du terrain qu'il a acquise). Un des principaux axes de simplification pour la troisième expérience est donc de permettre à l'utilisateur d'utiliser sa connaissance du terrain sans induire de fausse manipulation, ou du moins qu'elle soit facilement réversible.

Également, malgré une grande portabilité l'implémentation nécessite tout de même l'accès à un environnement Miniconda avec Nerfstudio configuré. Ce qui signifie qu'un utilisateur du public de non-initiés qui n'a pas accès à un expert du domaine pour l'installation de Nerfstudio ne pourra pas techniquement utiliser l'interface. Dès lors, l'axe de simplification sera de contourner ce problème en rendant l'accès à l'installation de Nerfstudio accessible.

En conséquence, les axes de simplification orientent l'expérience 3 vers la production d'une application en ligne avec une nouvelle technique de projection et n'impliquant plus la manipulation de bounding box. Une application en ligne permet d'installer Nerfstudio sur un serveur dédié que l'utilisateur interroge (inconsciemment) pour générer son modèle NeRF.

CHAPITRE VII TROISIÈME EXPÉRIENCE : WORKFLOW À DESTINATION DES NON-INITIÉS

VII-1 CADRE PROPRE À L'EXPÉRIENCE ET PRÉSENTATION

Dans cette troisième expérience, un dernier workflow est établi. Il constitue la simplification au plus haut niveau du workflow issu de la première expérience. En accord avec la segmentation établie durant la phase préparatoire (cf. IV-4), le public de non-initiés n'a que peu ou pas de connaissance en NeRF ou en SIG 3D. Par conséquent, son accès à cette nouvelle technologie ne peut se faire qu'à distance dans une application web qui délocalise les traitements et l'implémentation. L'interface utilisateur (frontend) doit être responsive et demander un minimum d'étapes, avec un maximum de feedbacks et des sécurités pour éviter les mauvaises manipulations. La paramétrisation doit être réduite aux choix esthétiques. Les résultats intermédiaires de l'expérience 2 indiquent également que la méthode de projection et d'alignement doit être revue pour que la connaissance du terrain de l'utilisateur puisse être injectée sans risque.

VII-2 IMPLÉMENTATION

Cette section présente l'implémentation réalisée lors de l'expérience 3. L'Archive, annexe au mémoire, contient un dossier « NSFacadeTooL » qui inclut les principaux scripts de l'application (Serveurs, frontend et viewer personalisé).

VII-2.1 INSTALLATION ET LANCEMENT

Lancer l'application nécessite toujours de réaliser l'installation de Nerfstudio sur la machine hôte. Mais dorénavant, cette opération fait partie du backend de l'application et comme l'utilisateur non-initié n'interagit qu'avec le frontend, il n'est plus tributaire d'une installation Nerfstudio. C'est une des méthodes les plus efficaces pour rendre Nerfstudio distribuable et accessible. C'est d'ailleurs pour cette raison que les créateurs de Nerfstudio ont intégré de nombreux outils permettant la mise en ligne du viewer. Des outils qui, de facto, ralentissaient les workflows des expériences 1 et 2 (lancement d'un serveur accueillant le viewer), mais qui rendent l'expérience 3 envisageable.

Une fois qu'un environnement pouvant exécuter le CLI de Nerfstudio est présent, il suffit de lancer les serveurs à partir de l'arborescence de scripts. Il faut également remplacer le viewer classique de Nerfstudio par le viewer personnalisé, simplement en écrasant le viewer.py original dans l'installation (cette dernière se trouve normalement dans les dépendances Conda typiquement à « C:\ProgramData\miniconda3\envs\nerfstudio\Lib\site-packages\nerfstudio\viewer) ».

Pour lancer les serveurs en local il faut accomplir trois tâches.

1) Lancer le frontend: dans le dossier frontend, lancer l'invite de commande (en tapant « *cmd* » dans la barre de navigation) puis entrer la commande « npm start » dans l'invite.
2) Lancer le backend principal: lancer une PowerShell miniconda 3, naviguer vers le dossier *« backend »*, entrer les commandes *« set PYTHONUTF8=1 »* pour éviter les erreurs de caractère et « conda activate nerfstudio » pour lancer le serveur dans un environnement supportant le CLI Nerfstudio. Enfin, lancer le serveur avec « node server.js ».
3) Lancer le serveur d'entrainements: même procédure que pour le backend principal mais il faut lancer le serveur dans le dossier « *training-server »*.

VII-2.2 ARCHITECTURE DU SERVEUR

L'architecture de l'application a largement évolué au fil de l'implémentation. Sa version finale se base sur 2 serveurs backend et une interface frontend. Effectivement, le backend est séparé entre un serveur principal permettant de répondre au frontend pour l'enregistrement des utilisateurs, la gestion de leurs données et l'exécution du viewer Nerfstudio, et un serveur spécial pour réaliser la mise en forme des données et l'entrainement. La séparation a plusieurs justifications. Premièrement, scinder les serveurs permet de découpler les traitements NeRF, assez lourds et exigeants en GPU, du support de l'interface devant rester responsive. Deuxièmement, lors d'une hypothétique mise en ligne, un serveur pourra être dédié GPU. Les services mettant à disposition des machines de calcul (google, amazon, …) sont généralement assez couteux (tarifs horaires pour les GPU et mensuel pour les CPU). C'est pourquoi si l'application devait être mise à disposition du public, il serait préférable de regrouper tous les traitements de la journée, ou de ne l'activer que lorsqu'une quantité de demandes suffisante est reçue. Les traitements plus légers (backend principal) peuvent être réalisés avec une machine moins couteuse en CPU. Quoiqu'il en soit, le découplage des serveurs est une optimisation amenant de la fluidité et une potentielle rentabilité à l'application. Il n'a pas été possible d'effectuer la mise en ligne de l'application mais l'implémentation a été pensée pour intégrer cette problématique dans des développements futurs.

Un diagramme de l'architecture est présenté à la Figure 36. Le frontend (en vert) n'est constitué que de l'interface utilisateur et du viewer de Nerfstudio. L'utilisateur charge les données brutes et recevra les vues orthographiques. Le backend est plus complexe et s'articule autour d'une base de données utilisateurs. Les serveurs principaux et le serveur d'entrainement ont accès à la base de données et peuvent écrire dans celle-ci. Les 2 serveurs peuvent interagir avec le CLI Nerfstudio. Le serveur principal lance le viewer avec *ns-viewer* et le serveur d'entrainement lance celui-ci et la mise en forme des données avec ns-train et ns-process-data. La communication entre le frontend et le backend se fait avec le protocole http classique GET/POST.

L'application est construite en *javascript* et en *React* pour s'aligner avec les langages et frameworks utilisés par le viewer Nerfstudio et ainsi, garder une cohérence entre les deux composants principaux du frontend.

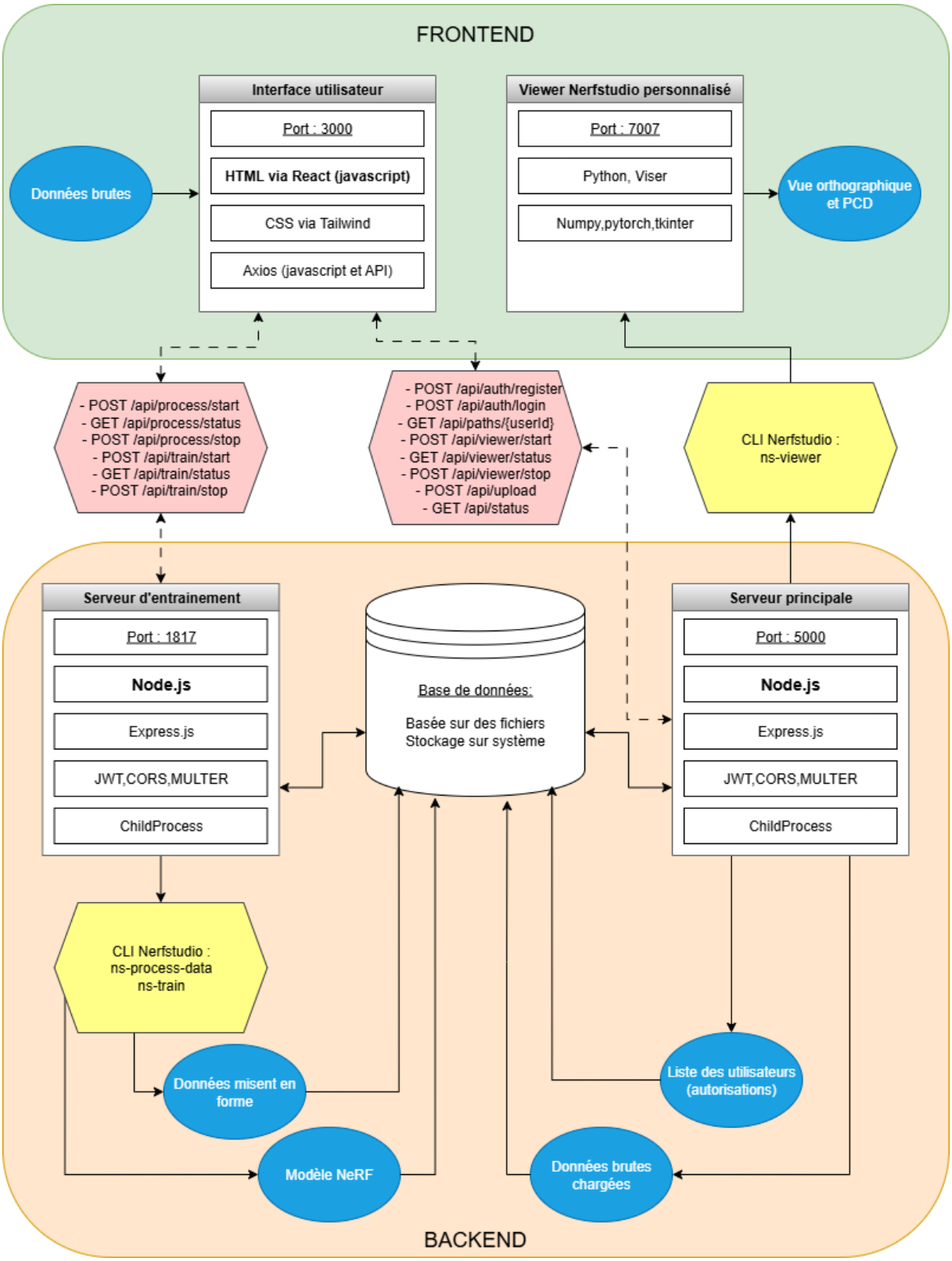


Figure 36 : Diagramme de l'architecture complète de l'application web implémentée lors de la troisième expérience. Les données sont en bleu, le CLI Nerfstudio (miniconda3) est en jaune et les API (communication) en rouge.

VII-2.3 COMMUNICATION

L'application est construite pour permettre à l'interface utilisateur d'employer une communication de type HTTP/REST asynchrone pour interagir avec les deux serveurs du

backend. Les deux composants principaux de la communication sont *Axios* pour la création et l'envoi des requêtes et *Express.js*, un framework serveur qui écoute et répond aux requêtes. *Express* facilite grandement la création de points d'entrée (endpoint) côté serveur. Il fournit également des outils pour une gestion automatique, notamment des « middleware », les logiciels intermédiaires de communication et de sécurité comme *JWT* (token d'authentification), *Cors* (pour des requêtes avec des origines croisées entre les 3 serveurs et la BD) et *MULTER* (qui permet l'upload et le transfert de fichiers volumineux comme le zip de données brutes).

Il y a 14 points d'entrée, 8 pour le serveur principal et 6 pour le serveur d'entrainement. Le serveur principal peut recevoir les requêtes liées à l'utilisateur (Enregistrement d'un nouvel utilisateur, connexion d'un utilisateur, stockage des données, synchronisation des données et localisation du dossier de stockage des données utilisateur) et celles liées au viewer (Start, stop, statut actuel). Le serveur d'entrainement peut recevoir les requêtes liées à la mise en forme des données de ns-process-data (Start, stop, statut actuel) et de l'entrainement avec ns-train (Start, stop, statut actuel). Voici un flux typique amenant à l'ouverture du viewer:

1) *POST 5000/api/upload*, l'interface utilisateur envoie les données brutes à stocker dans le serveur principal
2) *GET 5000/api/paths/${user.id}*, l'interface utilisateur demande au serveur principal le chemin vers les données que l'utilisateur vient de stocker.
3) *POST 1817/api/process/start*, l'interface utilisateur demande au serveur d'entrainement de mettre en forme les données avec la commande *ns-process-data* qu'il crée à partir du chemin récolté à l'étape précédente. Le serveur d'entrainement lance un Child process pour exécuter la commande dans le CLI de Nerfstudio. Nerfstudio enregistre les résultats directement dans la BD puisqu'elle est stockée localement. Le CLI se ferme automatiquement quand l'opération est terminée.
4) *POST 1817/api/train/start*, la même opération que la précédente mais avec *ns-train* pour l'entrainement du modèle NeRF. Il est à noter que la commande est spécifiquement configurée pour ne pas lancer de viewer à la fin de l'entrainement comme pour la configuration par défaut.
5) *POST 5000/api/viewer/start*, l'interface demande au serveur principal d'ouvrir un viewer pour le dernier modèle NeRF que l'utilisateur actuel a généré.

VII-2.4 BASE DE DONNÉES

La base de données utilisée dans l'implémentation est de type « file-based NoSQL ». C'est un format très simple et rapide qui s'appuie sur une arborescence de fichier mais qui n'intègre pas de structure SQL, l'information est stockée à la fois sous la forme de fichiers JSON (données à propos des utilisateurs) et de fichiers binaires (données utilisées par les utilisateurs). Les fichiers système sont stockés localement sur le même système que les serveurs de telle sorte que l'accès leur soit garanti à tous les deux. L'arborescence de la base de données est structurée à partir de dossiers imbriqués. Le dossier général contient le fichier « *users.json* » dans lequel sont stockés la liste des utilisateurs et leurs informations (ID, mail, mot de passe) puis un sous-dossier *« user_data »*. À chaque fois qu'un nouvel utilisateur est créé, un sous-sous-dossier est créé dans user_data sur base de son ID. Chaque dossier utilisateur comporte encore 4 dossiers : « *uplaods »* pour les fichiers zip de données brutes, « *processed »* pour stocker les données mises en forme, « *training »* pour stocker les données d'entrainement (checkpoints) et « *outputs »* pour stocker le modèle NeRF (config.yml).

L’utilisation d’une telle base de données représente un avantage pour le développement et la rapidité de l’application en général. Il est cependant important de noter que si l’application devait être mise en ligne et rendue accessible au public, il faudrait retravailler la BD (et donc les requêtes) pour la construire sur une base SQL qui est bien plus propice à gérer un grand nombre d’utilisateurs et de requêtes simultanées.

VII-2.5 PRÉSENTATION DU FRONTEND

Le frontend est constitué de l’interface utilisateur (Figure 37) et du viewer Nerfstudio personnalisé (Figure 38). L’utilisateur non-initié opère l’application à distance à partir de ces deux composantes et n’a plus à posséder de machines puissantes ni à comprendre en profondeur le processus, ni à interagir avec un CLI.

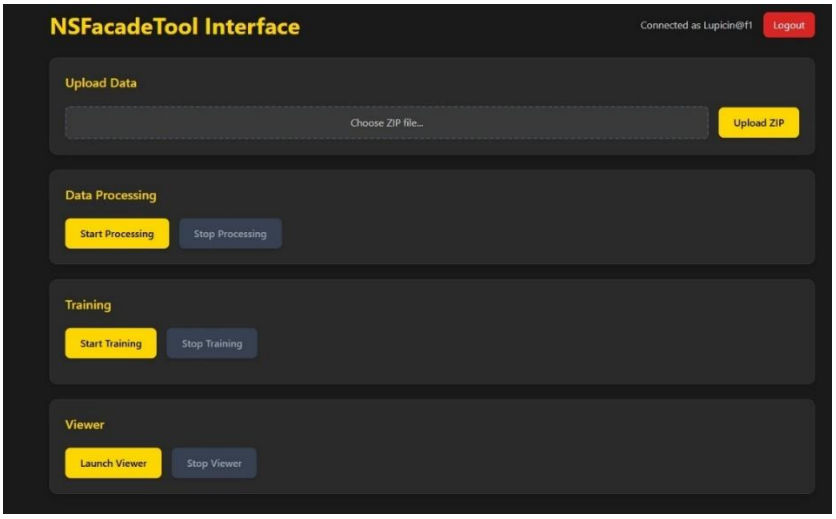


Figure 37 : Aperçu de l'interface utilisateur implémentée lors de l'expérience 3.

L’utilisateur peut se connecter à ses sessions précédentes, il peut charger les données qu’il a acquises avec Polycam, les mettre en forme et lancer l’entrainement du modèle NeRF. Une fois l’entrainement terminé, il peut lancer le viewer Nerfstudio personnalisé dans une nouvelle page web. Pour effectuer toutes ces opérations il n’a interagi qu’avec 4 boutons sans paramétrisation.

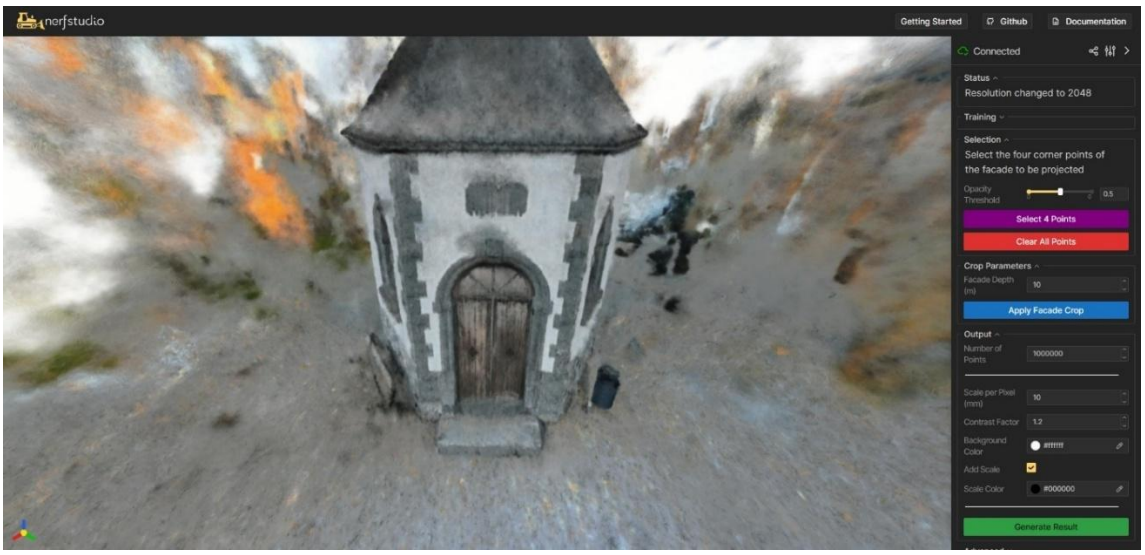


Figure 38 : Aperçu du viewer personnalisé lors de l'expérience 3

Le viewer personnalisé est très similaire à l'original de Nerfstudio. Il conserve la structure générale, la fenêtre de visualisation et les opérateurs sous la forme de dossier déroulant à droite. En revanche, de nouveaux dossiers d'opérateur ont été ajoutés pour intégrer le workflow de production de vues orthographiques. Les anciens opérateurs étant complexes mais pouvant être utiles à un public d'experts sont regroupés dans le dossier *« advanced »*.

L'introduction de ce nouveau viewer est particulièrement intéressante car celui-ci permet de compacter à la fois les étapes d'ajustement de la bounding box et les étapes de projection en une interface unique, opérable avec seulement 3 boutons et quelques options pré-paramétrées. Cela a été rendu possible par une approche inédite se basant sur les propriétés des NeRF et des rayons qui sera explicitée dans la section suivante.

Pour résumer, grâce à ses deux composantes, le frontend permet à l'utilisateur de transformer les données brutes en vue orthographique et en PCD en seulement quelques boutons, sans installation et avec très peu de connaissances théoriques et pratiques.

VII-3 WORKFLOW TEXTUEL DÉTAILLÉ

De façon similaire à l'expérience 2, l'implémentation de l'application fait plusieurs milliers de lignes de codes. L'approche simple et intuitive qui est d'expliquer uniquement les calculs et fonctions principales au fil du workflow plutôt que de revoir le code ligne par ligne a été conservée dans cette section.

VII-3.1 INTERFACE UTILISATEUR

L'interface utilisateur remplit plusieurs fonctions. Tout d'abord, c'est le premier contact de l'utilisateur quand il charge la page web. Ensuite, elle comprend tout le système de compte utilisateur avec la connexion et l'enregistrement. Enfin, elle adapte la chaine de traitement de l'expérience 2 pour interagir avec le CLI de Nerfstudio. Accessoirement, elle permet aussi de passer à la deuxième interface de projection, c'est-à-dire le viewer Nerfstudio.

VII-3.1.1 Connexion/enregistrement

L'application prend en charge la multiplicité des utilisateurs. Un dossier est créé dans la base de données à chaque enregistrement d'un nouvel d'utilisateur. Lorsqu'un utilisateur se connecte, il suffit à l'interface utilisateur de demander le chemin vers son dossier pour que l'utilisateur puisse reprendre les tâches qu'il était en train d'effectuer. Il peut notamment lancer un entrainement et reprendre le workflow plus tard quand les calculs ont été effectués. Lors du chargement de l'application, l'utilisateur arrive sur une page de connexion ou il peut saisir son adresse e-mail et son mot de passe; s'il est nouveau il peut s'inscrire. Une fois connecté, une page d'accueil avec un message de bienvenue et de la documentation est fournie à l'utilisateur pour le familiariser avec l'application. La Figure 39 donne un aperçu des pages de connexion et d'accueil.

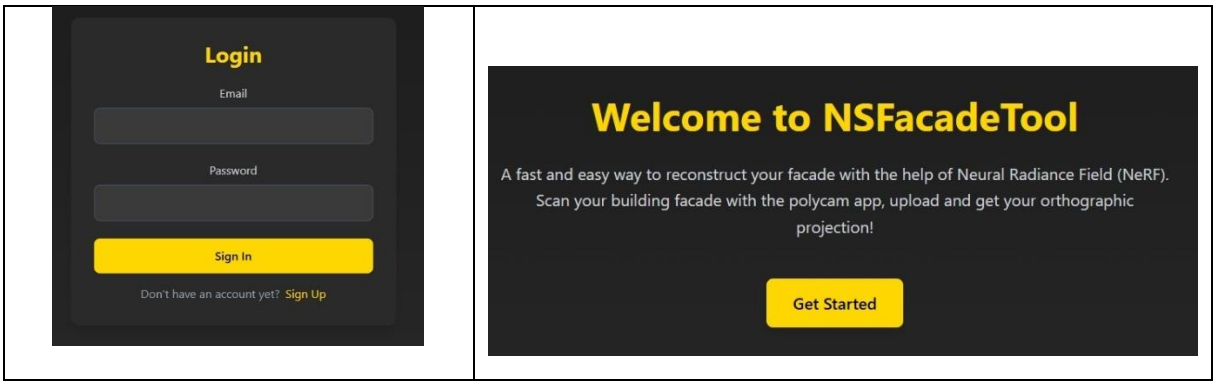


Figure 39 : Les modules de la page de connexion et de la page d'acceuil.

VII-3.1.2 Chargement des données

Le premier cadre de la page principale permet à l'utilisateur de charger les données brutes dans la base de données. L'utilisateur commence par sélectionner son fichier avec le cadre pointillé puis il peut le charger sur le serveur avec le bouton « upload ZIP ». Un feedback est affiché sous le cadre en cas d'échec ou de réussite.

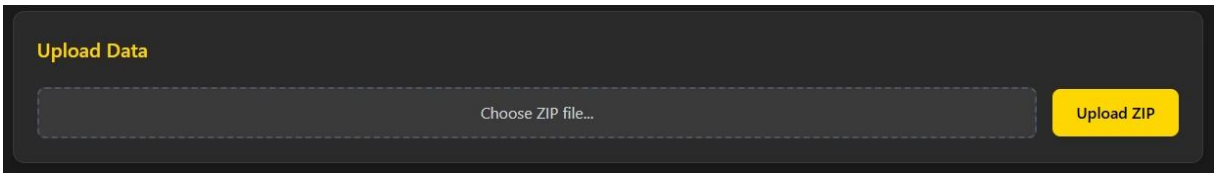


Figure 40 : Cadre de chargement des données brutes.

Techniquement, le fichier zip choisi par l'utilisateur est d'abord stocké dans une variable temporaire de React (file) puis est attaché à un objet prévu pour l'envoi (formdata). Grâce à la requête POST 5000/api/upload, l'objet peut être envoyé via http vers le backend. La requête peut être réceptionnée au point d'entrée du serveur principal après avoir été contrôlée par les middlewares de vérification (JWT pour les tokens d'identité utilisateur, Multer pour la réception et la continuité d'un stockage cohérent pour chaque utilisateur). Les middlewares vérifient si l'utilisateur existe et est bien connecté avant d'autoriser le stockage du fichier dans le dossier utilisateur de la base de données. Si l'utilisateur n'existe pas le fichier est refusé ou un utilisateur peut être créé. Dans une utilisation normale, le dossier existe puisque l'inscription des nouveaux utilisateurs crée automatiquement un dossier pour ceux-ci.

VII-3.1.3 Mise en forme et entrainement

La mise en forme des données et l'entrainement sont similaires à ceux des implémentations précédentes. Les seules différences sont la méthode d'accès et d'interaction avec le CLI Nerfstudio et la présence directement dans l'interface des résultats du CLI.

Comme le montre la Figure 41, les cadres permettant de lancer les traitements sont fortement simplifiés. Il y a seulement cinq boutons: le lancement et l'arrêt pour la mise en forme et pour l'entrainement et un bouton pour observer la convergence de l'entrainement dans le viewer Nerfstudio. Lorsque l'un des deux traitements est lancé, un terminal s'affiche sous les boutons et renvoie les impressions (les « prints ») du CLI en quasi-temps réel. Le terminal de l'entrainement est annoté avec la signification des impressions, notamment le nombre d'étapes et donc le pourcentage avant la fin de l'entrainement.

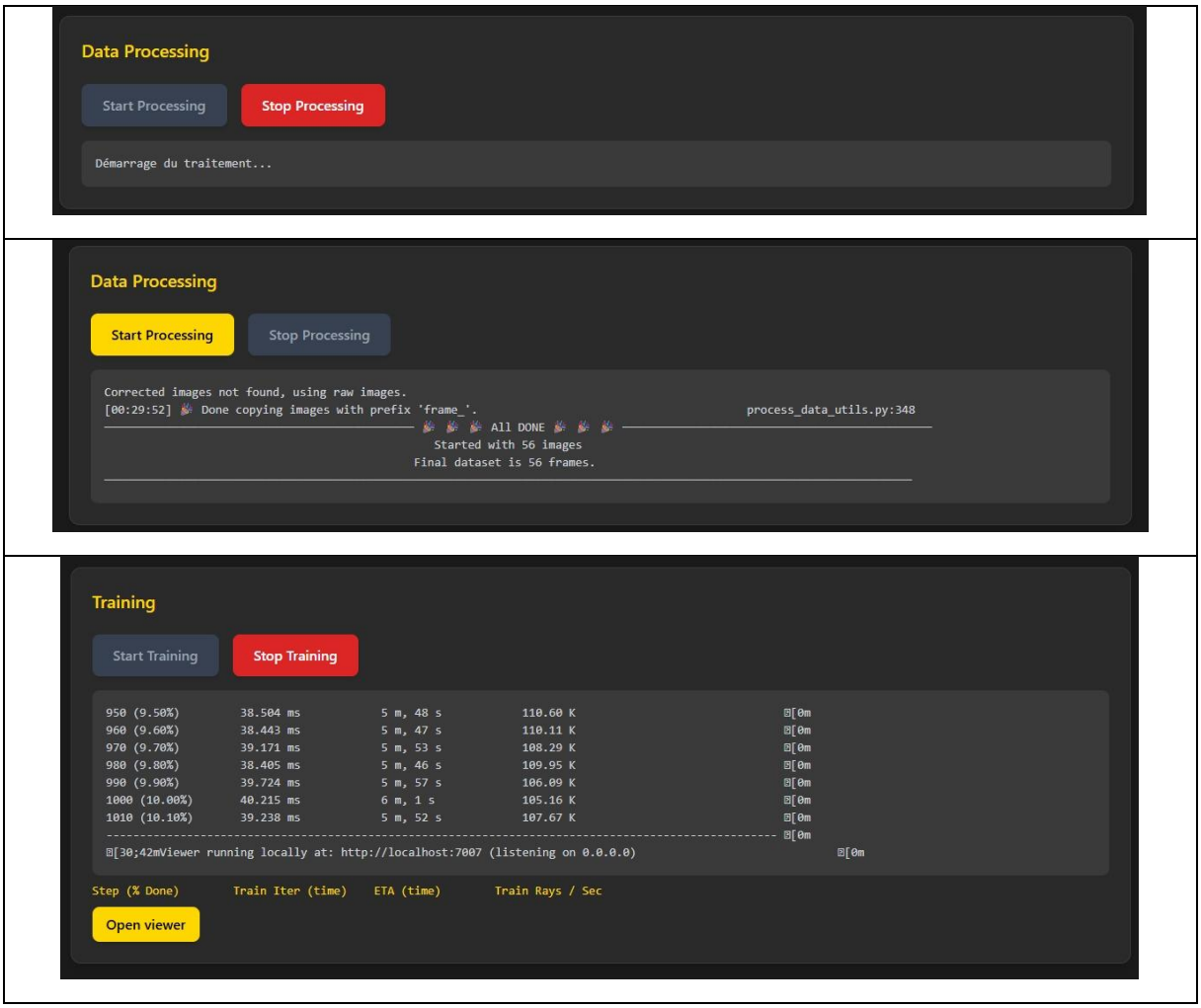


Figure 41 : Aperçu des cadres de mise en forme et d'entrainement selon plusieurs configurations : (1) en cours de mise en forme, (2) résultats de la mise en forme, (3) en cours d'entrainement

Techniquement, l'interface utilisateur demande au serveur d'entrainement de démarrer un sous processus en utilisant la fonction *« spawn »* de la bibliothèque « child process ». De la même manière que pour les expériences précédentes, le serveur d'entrainement a besoin de plusieurs informations pour construire la commande Nerfstudio, notamment les chemins vers le dossier de l'utilisateur. La structure identique pour chaque utilisateur permet de simplement spécifier le chemin général vers son dossier. Lorsque le serveur d'entrainement connait le chemin vers le dossier principal de l'utilisateur dans *« user_data »*, il peut créer par concaténation les chemins vers les sous-dossiers secondaires (/uploads pour les données brutes, /processed pour les données mises en forme et /outputs pour le modèle NeRF). Les commandes peuvent être remplies avec les nouveaux chemins et puis être lancées dans un sous-processus. Le serveur étant initialisé dans un environnement Conda, apprêté Nerfstudio, les sous processus ont la possibilité d'utiliser le CLI Nerfstudio.

Pour obtenir le chemin vers le dossier général, le frontend effectue pour commencer la requête *«GET 5000/api/path/{user.id} »* qui permet de récupérer le chemin pour l'utilisateur en cours, puis il lance une nouvelle requête vers le serveur d'entrainement contenant le chemin (« *POST 1817/api/process/start »* ou « *POST 1817/api/train/start »*). Les commandes stop et status permettent d'arrêter le sous-processus en cours ou d'observer les dernières impressions de la console en cours (l'interface rafraichissant toutes les secondes les terminaux de feedback).

VII-3.1.4 Ouverture du viewer

L'ouverture du viewer fonctionne sur le même principe de sous-processus et de requête que celui utilisé pour trouver le chemin du modèle à afficher. La seule différence est que la console est lancée par le backend principal et que donc seul l'ID de l'utilisateur est nécessaire puisque la BD est attenante à ce serveur dans notre implémentation. Le modèle ayant été calculé par l'entrainement, la configuration est totalement contenue dans le fichier .yml situé dans le sous dossier « processed ». Le serveur sélectionne le dernier fichier .yml créé pour ajouter son chemin dans la commande et lancer « ns-viewer ». Il est à noter que le serveur vérifie si un viewer n'occupe pas déjà le port 7007 et renvoie une erreur si le port est déjà occupé.

Une fois le viewer lancé, un lien est proposé à l'utilisateur pour qu'il passe dans l'interface suivante.

VII-3.2 VIEWER PERSONNALISÉ

En suivant les axes de simplification, la directive est d'intégrer les opérations de projections directement dans le viewer. Il en découle l'élaboration d'un viewer personnalisé qui est le principal apport technique de cette implémentation. L'approche est inédite pour la combinaison NeRF-projection mais se base sur la même méthodologie que le générateur de PCD.

VII-3.2.1 Principe théorique

L'approche de cette implémentation nécessite de comprendre en profondeur les principes du viewer et des NeRF. Voici le développement théorique qui sous-tend l'approche.

Contexte

Le viewer de Nerfstudio n'est pas un viewer conventionnel comme CloudCompare ou blender. La scène n'est pas un objet 3D pré-rendu avec une caméra qui tourne autour de l'objet comme pour les PCD et les MESH. Pourtant, à première vue, le viewer utilise effectivement un référentiel local qui lui est propre pour déplacer une caméra virtuelle dans la scène. Mais dans les faits, l'espace 3D dans le viewer est en quelque sorte « vide » et ne contient que cette caméra et son référentiel. Cela est dû au fait que la représentation 3D du modèle NeRF n'est pas du tout chargée dans la scène (et elle ne peut tout simplement pas car en réalité, elle n'a pas de signification physique). Le principe fondamental qui a popularisé les NeRF et qui permet de contourner cette restriction est donc de représenter la scène 3D à partir de vues synthétiques 2D générées par rendu volumétrique. Puisque la vue peut être générée en n'importe quelle position et selon n'importe quelle direction, une vue synthétique peut être générée partout dans la scène. Concrètement, le viewer affiche des vues 2D en fonction de sa caméra et pas d'une scène 3D pré-rendue. Or pour projeter un objet en 3D selon un système d'axes comme lors des expériences précédentes, il faut matérialiser le dit objet et son plan de projection dans la scène. Le but de l'approche est de réussir à matérialiser les points principaux dans la scène du viewer du NeRF. Précédemment, ce problème était contourné en générant un PCD complet puis en injectant une signification dans le nuage grâce à une segmentation des façades.

Principe

L'approche utilise deux types d'informations pour injecter cette signification sémantique dans la scène du viewer: la connaissance du terrain de l'utilisateur (capable de différencier les surfaces solides et les surfaces transparentes) et la densité sortie du modèle. Et ce, dans le but de placer un point qui a un sens dans le référentiel local du viewer. Dans ce cas-ci, on cherche à placer un point au niveau de la façade du bâtiment sur une surface solide.

Les coordonnées 3D du point sont reconstruites à partir de l'équation du rayon $\boldsymbol{r(t) = o + td}$ (cf. équation (2), I-5.3). L'origine o et la direction d sont définies par l'utilisateur à partir de la vue synthétique en cours de visualisation. L'origine o correspond à la position 3D actuelle de la caméra du viewer. En cliquant sur un pixel de l'image 2D qui correspond à la surface du bâtiment, l'utilisateur détermine la direction d pointant vers la surface solide. Grâce à ces deux premières composantes, le viewer peut créer un rayon qui traverse toute la scène et il peut évaluer l'opacité à intervalle régulier. En fonction d'un seuil d'opacité défini par l'utilisateur, le tracé du rayon est interrompu à l'endroit où la surface devient opaque (c'est-à-dire à l'endroit où elle devient solide dans la réalité). La distance entre l'origine et le point qui devient opaque donne la dernière composante, la profondeur t. Grâce aux trois composantes, il est possible de calculer la position d'un point significatif (sur une surface solide) dans la scène 3D du viewer.

Ce processus est répété plusieurs fois pour déterminer le plan de l'objet nécessaire à la projection orthographique. La bounding box est ajustée à ce plan et un petit PCD encadrant le plan est généré pour texturer la vue orthographique. Le PCD est aligné sur la façade et projeté sur le plan objet en suivant sa normale.

Sélection par densité volumétrique

Lorsque le viewer génère un rayon à partir du modèle, il s'étend à l'infini en traversant la scène sans distinction. De par son échantillonnage stratifié en N points et son bornage, le modèle permet d'estimer l'opacité locale α_i de chaque point du rayon à partir de la densité volumétrique:

$$\alpha_i = 1 - e^{-\sigma_i \delta_i} \quad (12)$$

Avec σ_i, la densité volumétrique au point i et δ_i l'inter distance entre les points.

Pour retrouver l'opacité finale du rayon, il faut accumuler l'opacité selon la transmittance à travers les précédents segments. On peut réaliser cela avec une approche par somme pondérée telle que:

$$\widehat{\alpha_{final}}(r) = \sum_{i=1}^{N} T_i \, \alpha_i \quad (13)$$

Avec la transmittance du segment (idem équation (9)I-5.3),

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (14)$$

On obtient une équation similaire à celle de la couleur estimée (cf. équation (8), I-5.3) mais celle-ci représente uniquement la densité. Afin de simplifier les calculs, la formule standard utilise l'équivalent où l'opacité est l'inverse de la transmittance cumulée, tel que:

$$\widehat{\alpha_{final}}(r) = 1 - T_N = 1 - \prod_{i=1}^{N} (1 - \alpha i) \quad (15)$$

Une fois que l'opacité a été prédite pour chaque point; le rayon est parcouru depuis son origine et le premier point qui dépasse le seuil d'opacité α_{seuil} est sélectionné**.**

La Figure 42 permet d'illustrer graphiquement ce principe: les surfaces solides sont plus denses volumétriquement et forment des pics, l'opacité accumule ces pics. Quand on dépasse le seuil, on considère que tous les points suivants sont opaques. Ce qui signifie que le rayon a rencontré une surface solide, la surface que l'utilisateur a « pointé ». La profondeur à laquelle le rayon devient opaque est dépendante des propriétés de l'objet 3D. Par exemple, pour la caméra 2 qui produit un rayon parallèle mais à une position différente, on voit que la profondeur est plus élevée parce que la façade est plus lointaine.

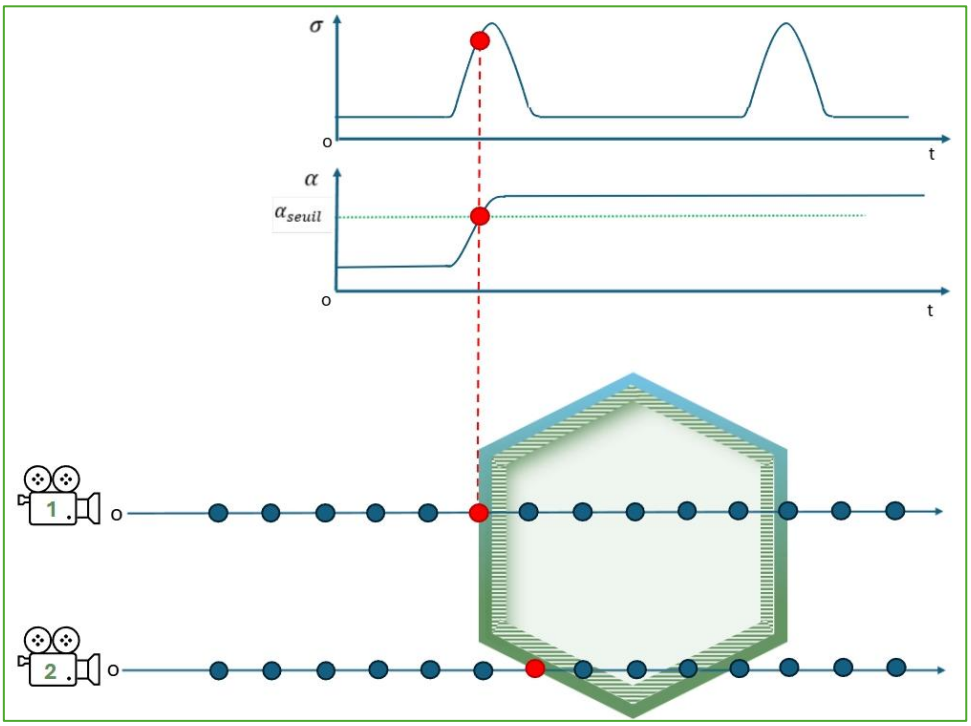


Figure 42 : Illustration du principe général de détection de surface opaque

Il est à noter que la sélection par densité volumétrique du NeRF, n'est pas comparable avec la sélection par densité surfacique classique en reconstruction 3D. Ici la densité est une fonction continue dans l'espace qui permet de représenter la transmittance d'un point à une position donnée et dans une direction donnée (cf. équation (1), I-5.2) et pas simplement une concentration de points.

Les deux sections précédentes ont établi les fondamentaux théoriques, il est désormais possible d'expliquer techniquement le workflow détaillé.

VII-3.2.2 Sélection des 4 coins de la façade

La première étape dans le viewer personnalisé consiste en la sélection de points sur la façade correspondant à ses quatre coins.

Cela permet à la fois de déterminer le plan sur le principe de la sélection par densité volumétrique, de construire le plan à partir d'une primitive de trois points et de le dimensionner grâce au 4ème. C'est également beaucoup plus visuel pour l'utilisateur qui « encadre » la façade qu'il a intuitivement détectée. Par ailleurs, il est intéressant de noter que cette approche permet notamment d'éviter la sélection de « blobs », des artefacts de reconstruction qui sont souvent

problématiques lors de la génération automatique d'un PCD. Lors de la sélection de points, l'utilisateur se place (encore une fois intuitivement) dans une configuration où le rayon ne passe pas par un blob (une tache blanche bloquant sa vue).

Le dossier « selection » contient un seul paramètre : le seuil d'opacité. Il est laissé au choix de l'utilisateur pour qu'il puisse jouer avec la profondeur avec laquelle s'enfonce le rayon (son clic). Le seuil est réglé à 0.5 par défaut pour apparaitre relativement rapidement à la surface de la façade.

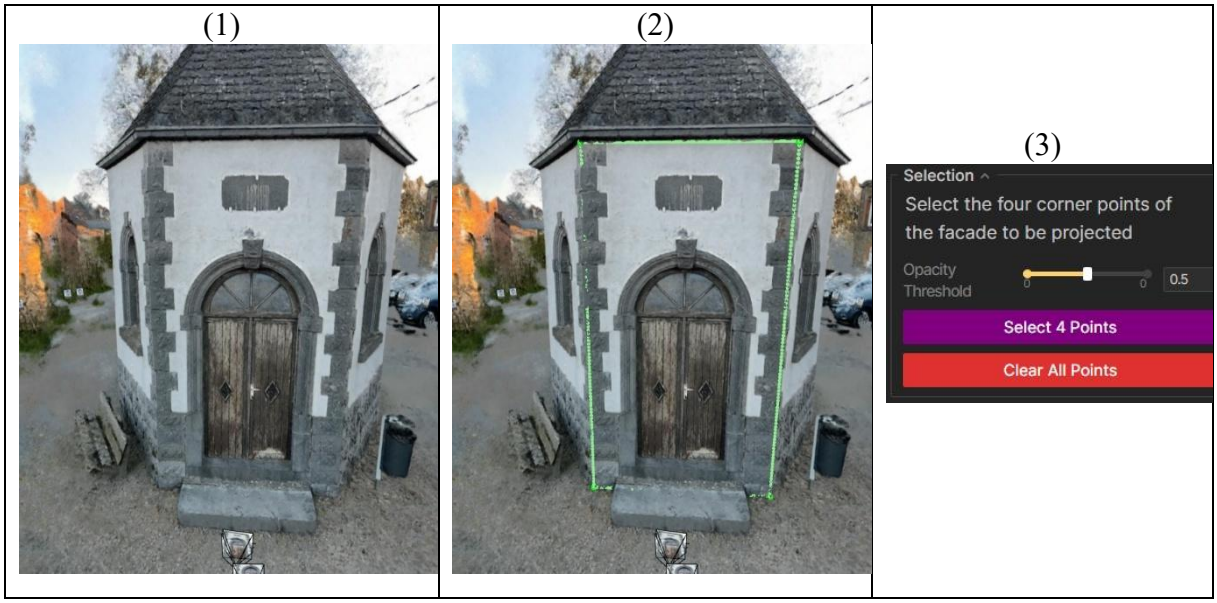


Figure 43 : Procédure de sélection des 4 coins de la façade : (1) Scène avant la sélection, (2) scène avec le cadre sélectionné et (3) interface utilisateur de la sélection

Techniquement, l'opération se base sur le principe de sélection par densité volumétrique mais l'adapte pour le framework Nerfstudio. En effet, Nerfstudio peut effectuer un rendu volumétrique et donc calculer la densité et l'opacité, mais il peut seulement l'obtenir à travers sa structure de données appelée « RayBundle » pour générer tous les rayons d'une image (pixel area). Pour simplifier la génération, il a été beaucoup plus simple de configurer un RayBundle ultra simplifié qui ne trace qu'un seul rayon fictif à partir du pixel pointé par l'utilisateur. Normalement, le modèle Nerfacto peut directement calculer la profondeur mais s'il échoue en cas de valeur aberrante, un tracé de rayon (ray marching) manuel a été implémenté en créant 256 points (base de rayon) disposés à intervalle régulier le long du rayon.

Une fois que la profondeur est calculée, un point est créé à partir de sa position o et de la direction d. Les points sont dans le référentiel local de référence et ne sont pas visibles dans l'interface, le viewer crée des markers à la position des points (mise à l'échelle du référentiel local du viewer)

VII-3.2.3 Ajustement de la bounding box (rognage)

Dans les expériences précédentes, en plus de nécessiter une certaine connaissance des PCD, l'ajustement de la bounding box était une étape très chronophage mais nécessaire pour isoler les parties de façades d'intérêt. Dans cette implémentation, cette opération a pu être automatisée car les points de la façade sont déjà segmentés. Il suffit donc d'ajuster la position (centre du plan), la dimension (à partir du calcul des vecteurs du plan) et d'aligner (à partir d'une matrice de rotation) pour paramétrer automatiquement la bounding box.

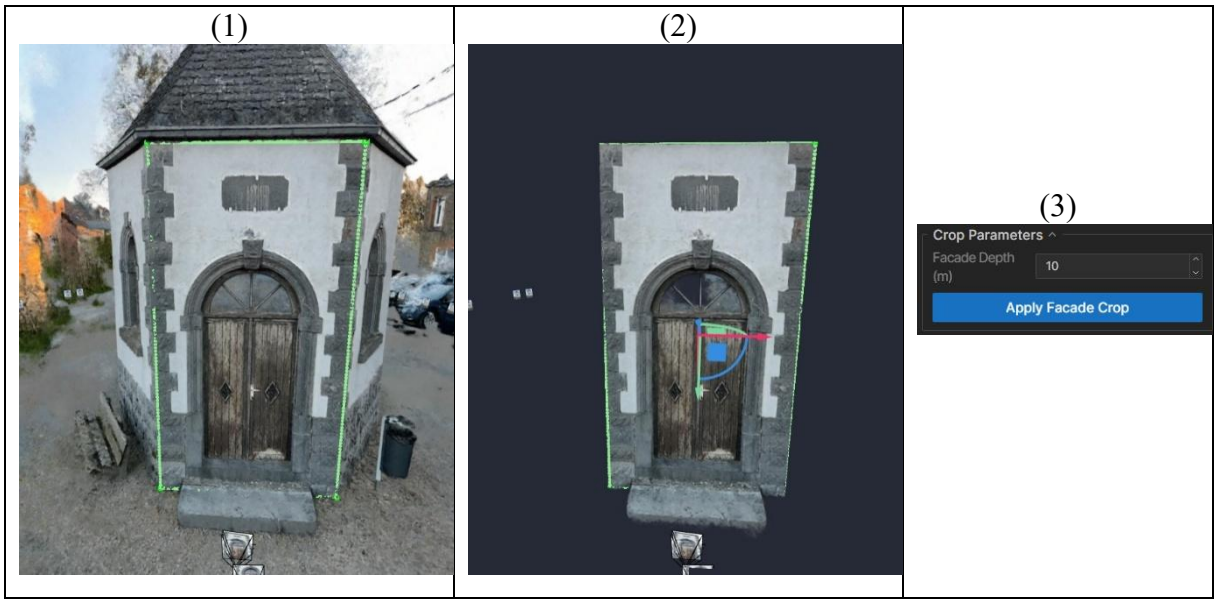


Figure 44 : Procédure d'ajustement de la bounding box : (1) Scène après la sélection sans crop, (2) scène sélectionnée et rognée et (3) interface utilisateur du crop

L'interface utilisateur ne comporte également qu'un paramètre: la profondeur intrinsèque de la façade. Techniquement, cela permet d'ajuster la taille de la normale du plan et donc d'élargir ou non la bounding box dans le seul axe qui n'est pas dimensionné par les points sélectionnés. En pratique, cela permet d'intégrer les reculs et reliefs en retrait des façades complexes.

VII-3.2.4 Génération des résultats

Le dossier *« output »* comprend une série de paramètres de personnalisation des résultats et permet de générer tous les résultats en un bouton. Il permet de réunir l'étape de génération du PCD et toutes les étapes de projection en une seule opération.

Le visuel de la projection est obtenu par la projection d'un PCD dense sur le plan objet sélectionné (qui de ce fait est confondu avec le plan de projection). Comme pour l'expérience 2, la génération du PCD est réalisée à l'aide d'une commande auto-configurée qui est lancée dans un sous-processus *« subprocess »*. Les seuls paramètres de la commande variables sont le chemin du dossier de sortie et le nombre de points du nuage *(« number of points »*). Le chemin du fichier de sortie est récupéré avec une fenêtre de navigation dans le gestionnaire de fichier (*filedialog* de *tkinter*) qui se lance avant de générer les résultats et le nombre de points est configuré par l'utilisateur.

Ensuite, le nuage généré est chargé dans le viewer à l'aide d'un module *« open3D »* et tous ses points sont alignés et centrés sur le plan objet pour être prêts à sa projection. Un nuage des quatre points de la façade est aussi enregistré à côté du PCD principal. À ce stade l'utilisateur possède déjà un PCD dense de la façade (orienté et mesurable) sauvegardé au format .ply.

La projection intègre les paramètres utilisateur pour créer une image en 2D et au format .png. La grille de pixels est déterminée en divisant les dimensions (largeur et hauteur) du plan (avec une marge de 20%) par l'échelle par pixel *(« scale per pixel »)* fournie par l'utilisateur. Ensuite les points sont projetés sur la grille, chaque pixel comptabilise les points qui lui sont projetés, puis sa couleur est obtenue par la moyenne des couleurs des points comptabilisés.

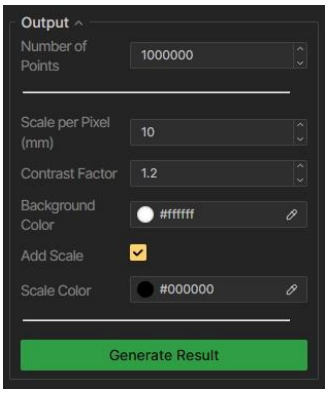


Figure 45 : Interface utilisateur pour générer les vues orthographiques.

Les autres paramètres du dossier sont esthétiques : le choix du fond de plan («Background color »), un filtre de contraste («Contrast factor »), l'ajout d'une échelle (barre et numérique) et sa couleur (« scale color »). D'ailleurs le nombre de points et l'échelle par pixel sont à considérer comme des options esthétiques également puisqu'ils contrôlent la résolution de l'image finale.

En pratique l'utilisateur configure l'aspect de sa projection puis clique sur le bouton « generate result ». Un explorateur de fichier s'ouvre pour choisir le dossier de sauvegarde des résultats puis la projection et son nuage sont générés. La progression des calculs peut être suivie dans le dossier « status » qui fait office de terminal de feedback. Il faut noter que toutes les opérations précédentes sont également notifiées par des feedbacks dans ce terminal.

VII-4 SORTIES ATTENDUES : WORKFLOW ET VUES FINALES

Comme pour les autres expériences, cette section présente les sorties obtenues en appliquant le workflow au jeu de données de référence. Les vues orthographiques représentent l'extraction des six façades de la chapelle. De même, le workflow est toujours quantifié sur base des critères d'accessibilité. Les résultats sont aussi disponibles dans l'archive ; les vues sont enregistrées au format .png dans le sous dossier « implementation3 ». L'implémentation est trop complexe pour être directement transférée mais les principaux scripts sont disponibles dans le sous dossier « NSFacadeTooL », notamment viewer.py (viewer personnalisé), app.js et ses fichiers de configuration annexe (le frontend en react), les deux scripts java des serveurs (.js du backend principal et du serveur d'entrainement).

Figure 46 : Vues orthographiques isolant les 6 façades (dans le sens horaire en partant de la façade avant sud).

VII-4.2 ANALYSE DU WORKFLOW TEXTUEL

Tableau 6 : Résumé et analyse du troisième workflow

Numéro d'étapes	Intitulé	Nécessite une paramétrisation	Fausse manipulation possible	Feedback en cours de procédure	Feedback après la procédure	Comprendre les NeRF	Comprendre le SIG 3D	Temps de réalisation (minutes)
1	Connexion	✗	✗	✓	✓	✗	✗	1
2	Sélection des données brutes	✗	✗	✓	✓	✗	✗	1
3	Conversion des données brutes	✗	✗	✓	✓	✗	✗	1 à 5
4	Lancer l'entrainement	✗	✗	✓	✓	✗	✗	5 à 30
5	Ouvrir le viewer de Nerfstudio	✗	✗	✓	✗	✗	✗	1
6	Sélection du plan de la façade	✓	✓	✓	✓	✗	✗	5
7	Ajustement de la profondeur de la façade et rognage	✓	✓	✓	✓	✗	✗	1
8	Personnalisation de la projection	✓	✓	✓	✓	✗	✓	1 à 5
9	Choix du dossier de sortie	✗	✗	✓	✓	✗	✗	1
10	Projection et export	✗	✗	✓	✓	✗	✗	1 à 15
Total		3	3	10	9	0	1	18 à 65

VII-5 RÉSULTATS TERMINAUX

Au terme de cette expérience 3, un workflow a été établi en se basant sur l'implémentation d'une application web à deux serveurs. Il répond aux axes de simplification des expériences précédentes, c'est-à-dire éviter (ou minimiser) toute mauvaise manipulation, réunir toutes les opérations pour fluidifier le workflow et réduire le niveau de connaissance nécessaire à celui du public de personnes non-initiées. Le workflow est finalement réduit à 10 étapes et aucune ne nécessite la connaissance préalable des NeRF ou des SIG 3D. Seules la personnalisation de la projection ou la compréhension de la résolution demandent une certaine connaissance basique.

Les 10 étapes possèdent des feedbacks avant et après (sauf pour l'ouverture du viewer mais l'opération est quasi instantanée). Les trois étapes qui nécessitent une paramétrisation sont la sélection de la façade, le rognage et la personnalisation de la projection. Les trois étapes sont cependant préconfigurées et les possibles fausses manipulations qui en découlent sont soit esthétiques, soit facilement réversibles.

Les principaux points d'amélioration de l'application sont d'une part, l'ajout de barres de progression pour l'entrainement et la génération du nuage (qui n'ont pas pu être implémentées à cause du passage en sous-processus) et d'autre part, un tutoriel avec des fenêtres contextuelles pour la première utilisation (qui n'a pas pu être implémentée à cause de l'imbrication du CLI dans un sous-processus). La comparaison de ces résultats avec ceux des autres expériences est abordée dans le chapitre suivant (voir VIII-1.2).

CHAPITRE VIII ANALYSE DES RÉSULTATS

VIII-1 COMPARAISONS DES IMPLÉMENTATIONS

Le respect du contexte commun permet aux expériences de produire des résultats qui peuvent être comparés.

VIII-1.1 COMPARAISON QUALITATIVE DES VUES ORTHOGRAPHIQUES

La Figure 47 permet de comparer trois vues orthographiques obtenues pour chaque expérience en suivant leur workflow attitré. Elles sont toutes les trois été réalisées avec le même nombre d'étapes d'entrainement (30 000) sur le même jeu de données. Le PCD de projection a toujours été généré avec 1 000 000 de points. La précision des angles et la netteté a augmenté au fur et à mesure des expériences. Cela est dû au fait que l'ajustement de la bounding box a été de plus en plus précis au fur et à mesure des expériences.

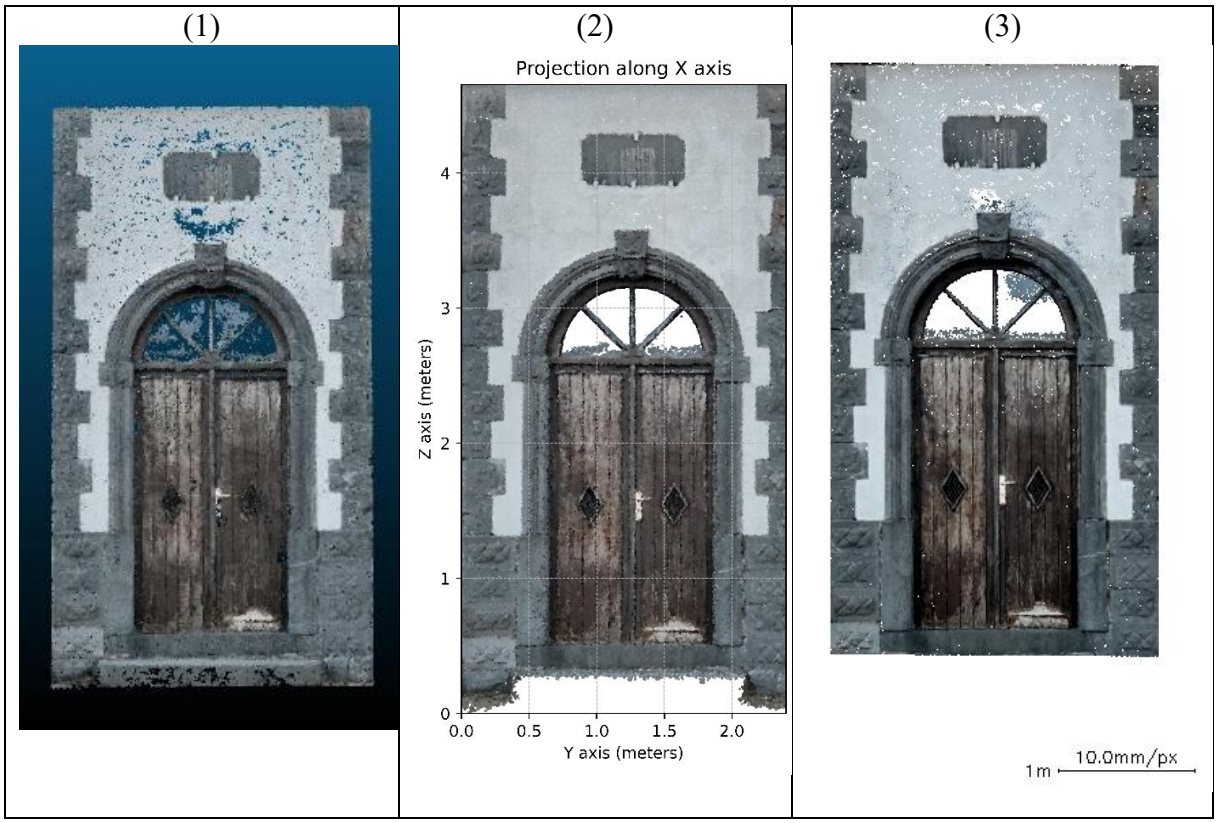


Figure 47 : Comparaison des vues orthographiques de la première façade du jeu de données de référence générée lors des 3 expériences (ordre croissant).

Pour la première expérience, le PCD général est généré sur toute la scène, puis est nettoyé et ensuite découpé pour obtenir la façade. Cela permet de rapidement découper les scènes complexes ou multiplanaires mais a pour défaut de réduire le nombre de point par PCD de projection (celui de la façade 1 compte 322 705 points). Techniquement, cet impact peut être réduit en augmentant le nombre de points du PCD général ou en ajustant plus précisément la bounding box.

Effectivement, l'expérience 2 ajuste la bounding box autour d'une façade (comme elle ne peut traiter que des nuages uniplanaires) et le nuage de projection compte bien 1 000 000 de points. Par conséquent, la résolution de la projection est plus fine et le nuage de point ne possède presque plus de lacunes, contrairement à la projection de la première expérience.

Lors de la troisième l'expérience, le plan objet est construit avant la génération du PCD, ce qui permet d'ajuster la bounding box au plus près du plan. Au final, le PCD de projection compte toujours 1 million de points mais ils sont concentrés dans une zone plus restreinte. Ceci explique en partie, pourquoi la projection de l'expérience 3 apparait plus nette. L'explication réside aussi dans le format d'export et la gestion de l'échelle. En effet, dans la deuxième expérience, la projection est calculée en fonction de l'échelle par matplotlib alors que dans la troisième expérience, une échelle est définie en fonction d'une résolution cible (ici 1 cm/pixel). Au final, la projection apparait plus nette mais fait apparaitre plus de lacunes que dans l'expérience 2 (mais moins que dans l'expérience 1) à cause de sa bounding box plus fine qui n'a pas cumulé des détails plus lointains.

D'un point de vue esthétique, la projection de l'expérience 2 possède plus d'informations grâce à matplotlib qui permet de facilement générer une grille, des axes, les titres. Les projections des expériences 1 et 3 sont plus simples et ne possèdent qu'une barre d'échelle. Cependant, ces aspects pourraient facilement être implémentés afin de répondre aux standards professionnels si le développement du projet continuait.

Pour conclure, en dehors de ces considérations esthétiques et de la résolution, les trois expériences ont réussi à produire des vues orthographiques à partir d'un modèle NeRF. Elles valident donc toutes les trois la première hypothèse de la question de recherche.

VIII-1.2 COMPARAISON QUANTITATIVE DE L'ACCESSIBILITÉ DES WORKFLOWS

Les workflows peuvent être comparés sur base des critères d'évaluation définis durant la préparation (cf. IV-2.2.3). Cette base comparative est une bonne approche pour remettre en perspective chaque workflow avec son public. Une large partie du tableau a déjà été évaluée lors de l'analyse des résultats intermédiaires de chaque expérience; celle-ci était axée sur le développement des expériences et la simplification du workflow.(cf.V-5, VI-5 et VII-5). L'analyse ci-dessous se concentre sur l'ensemble des résultats dans une vision plus globale.

Tableau 7 : Tableau comparatif des trois workflows en fonction du nombre d'étapes

Numéro	Workflow 1	Workflow 2	Workflow 3
Nombre d'étapes total	**18**	**12**	**10**
Nécessite une paramétrisation	9 (50%)	4 (33%)	3 (30%)
Fausse manipulation possible	11 (61%)	3 (25%)	3 (30%)
Feedback en cours de procédure	6 (33%)	7 (58%)	10 (100%)
Feedback après la procédure	15 (83%)	10 (83%)	9 (90%)
Comprendre les NeRF	3 (17%)	2 (17%)	0 (0%)
Comprendre les SIG 3D	8 (44%)	3 (25%)	1 (10%)
Niveaux d'exigence (NeRF + SIG 3D)	**30.5%**	**21%**	**5%**
Temps de réalisation (minutes)	**77 à 252**	**35 à 176**	**18 à 65**

Les workflows ont suivi la logique de simplification qui se traduit directement par la réduction du nombre total d'étapes par workflow.

La simplification se traduit également dans la forte réduction des possibilités de fausses manipulations (de 61% à 30%) induites par l'utilisateur ou une mauvaise paramétrisation. Cela étant en partie lié à l'augmentation (proportionnellement) du nombre de feedbacks et de détrompeurs, en cours et après, les procédures effectuées durant les étapes.

Mais surtout, les workflows ont réduit significativement le niveau de compréhension nécessaire en accord avec leur public cible. En cumulant les étapes où l'utilisateur doit faire intervenir sa connaissance des NeRF ou des SIG 3D, un aperçu de l'exigence des workflows en termes d'expertise est calculé : on obtient 30,5 % pour le public d'experts, 21 % pour le public intermédiaire et finalement 5 % pour le public de non-initiés.

Le temps d'opération est également fortement réduit en passant au minimum de 77 minutes à 18 minutes. Comme l'acquisition des données brutes est identique (environ 15 minutes), on peut estimer que le temps total minimum d'opération pour obtenir une vue orthographique (et donc un PCD) est d'environ 90 minutes pour le premier workflow, 50 minutes pour le second et seulement 35 minutes pour le troisième.

Rappelons que Croce et al. (2024) et Remondino et al. (2023) ont établi que la précision des NeRF varie généralement entre le niveau décimétrique et millimétrique selon les conditions d'acquisition et la nature des surfaces. Dans le cadre de nos expérimentations, de rapides mesures de contrôle ont confirmé cette tendance, en mettant en évidence des écarts de l'ordre du centimètre par rapport aux références, ce qui correspond à une précision suffisante pour des usages architecturaux standards.

Pour résumer, avec un temps de réalisation de 35 minutes et un niveau d'exigence de 5%, on peut clairement établir que le workflow défini à l'expérience 3 grâce aux expériences précédentes permet de rendre accessible la projection de vue orthographique (et de PCD) à partir de NeRF à un public large d'utilisateurs non-initiés et donc de démocratiser les relevés architecturaux.

VIII-2 MISE EN SITUATION

La mise en situation sert principalement à ouvrir à la discussion les résultats obtenus lors de la phase expérimentale. Elle permet d'explorer les résultats des workflows dans des situations plus complexes que le jeu de données de référence, ce dans le but d'en percevoir les potentielles limites et les points d'améliorations futures.

De nombreuses acquisitions ont été effectuées sur les sites secondaires présentés à la section IV-3.4.2. Celles-ci ont été utilisées pour générer une vue orthographique en suivant le workflow 3, puis le 2 et le 1 si un échec ou des résultats insuffisants étaient observés avec le précédent workflow. Un rapport d'évaluation a été généré dans Nerfstudio pour quantifier la qualité de la représentation 3D. Ces résultats sont disponibles dans l'Annexe 1 du document. Pour simplifier la lecture et rester concis, cette section analyse seulement les principaux résultats utiles et qui amènent une discussion.

VIII-2.1 SCÈNE LARGE : ÉGLISE D'IVOY

L'acquisition de la façade large de l'église d'Ivoy a permis de mettre en évidence deux éléments : le problème de la scalabilité de la puissance de calcul et les préférences en termes de relevé.

En ce qui concerne la scalabilité, c'est-à-dire l'extensibilité de l'application en puissance de calcul au niveau graphique (VRAM), l'acquisition d'un jeu de données bien plus large (± 300 frames) a été possible mais le calcul d'un PCD d'un million de points se heurte à la limite de mémoire. Or sans PCD, les étapes de projections ne peuvent pas être effectuées. En réduisant le nombre de points à un nuage de seulement 10 000 points, il a été possible de réaliser une projection en utilisant le workflow 1. Elle est cependant très (trop) lacunaire (Figure 48-1).

Au niveau de la méthode d'acquisition, les acquisitions ont montré que le parcours et le support influencent fortement la qualité des résultats et la vitesse de convergence du modèle. Au final, c'est l'acquisition qui utilise des aller-retours à 10m de distance du mur mais avec un changement d'angle vertical qui présente les meilleurs résultats. La qualité diminue au fur et à mesure que la distance avec le mur diminue. La qualité des acquisitions qui ne change pas l'angle mais la distance au mur est donc inférieure (Figure 48-1 et -2). L'acquisition avec la perche et le drone n'ont pas d'impact sur la qualité des résultats mais permettent une meilleure couverture du mur (Figure 48-4).

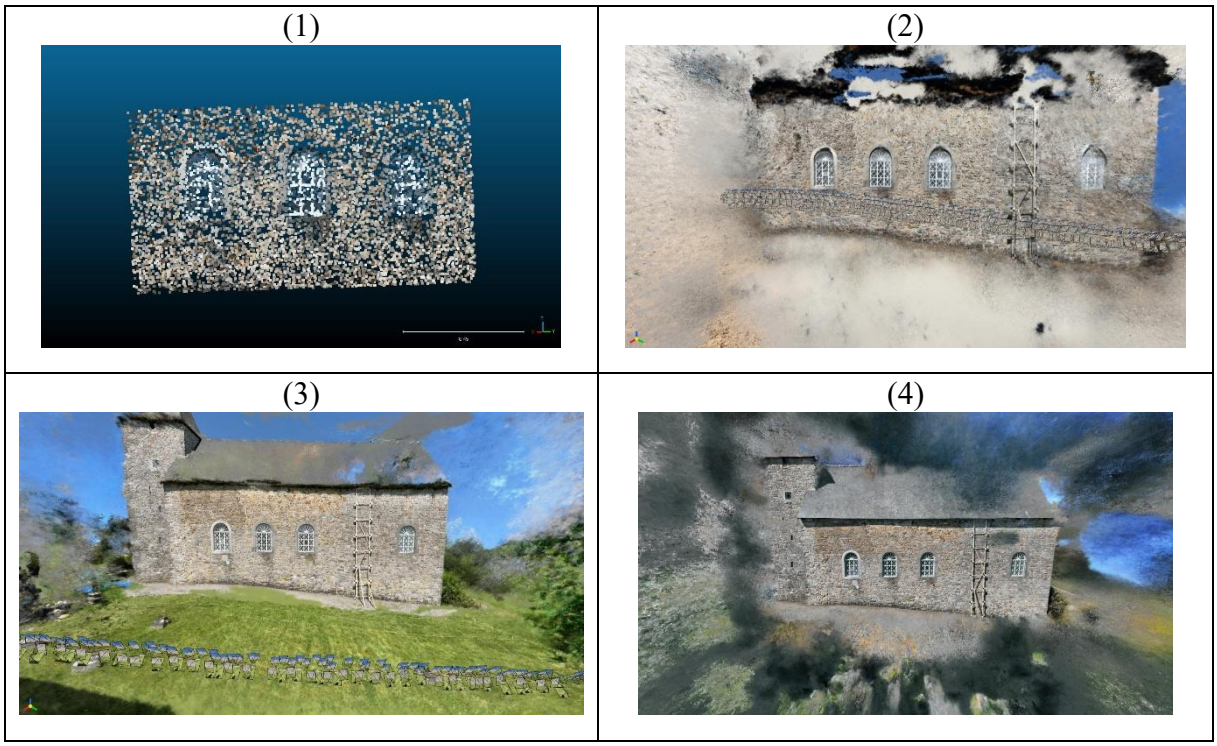


Figure 48 : Acquisition d'une scène large : (1) limite de génération du PCD, (2) Rendu d'un changement d'angle à 2m de distance de la façade (mauvaise qualité), (3) Rendu d'un changement d'angle à 10m de distance de la façade (scène complète), (4) Rendu du balayage horizontal par drone à 4m de distance de la façade (rendu précis du toit).

Il est à noter que la restriction à 30 000 étapes imposée dans le workflow 2 et 3 ne permet pas d'arriver à la convergence du modèle du NeRF. La différence de qualité des différentes acquisitions est visible mais la qualité globale est fortement réduite à cause de cette restriction. Le tour complet de l'église n'a même pas pu générer de modèle à cause de la limitation en mémoire également.

VIII-2.2 SCÈNE VERTICALE : MAISON EN PIERRE

La scène verticale permet de discuter l'intérêt d'utiliser un drone comme support d'acquisition dans le cas ou certaines parties de la scène sont inaccessibles. La façade de la maison en pierre a été scannée depuis le sol et depuis le drone qui a effectué un balayage vertical à environ 4m de distance de la façade.

En termes d'acquisition, il a été possible de générer des résultats similaires avec le workflow 3 pour le sol et le drone, ce qui confirme qu'il est possible d'utiliser ce support. Ce cas de figure soulève un problème intéressant: effectivement les images sont similaires mais la répartition de la densité et des lacunes est totalement différente. En effet, avec l'acquisition au niveau du sol, la densité diminue graduellement avec la hauteur du bâtiment tandis que pour le drone la densité diminue mais par bande verticale (voir Figure 49). Les bandes verticales moins denses correspondent aux balayages qui ont été effectués trop rapidement. Les limites sont donc complétement différentes: une limite de hauteur pour le sol (et donc une limite dans le choix de scène) et une limite de matériel pour le drone (présence ou non d'un régulateur de vitesse).

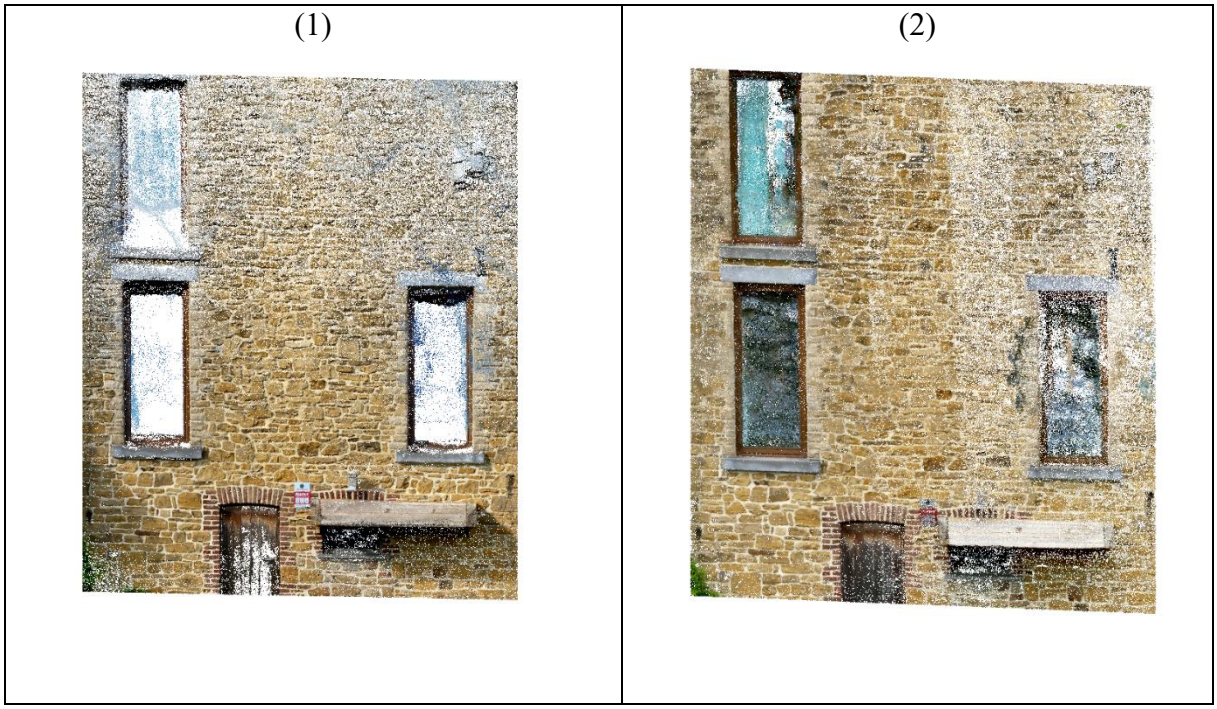


Figure 49 : Comparaison de la projection d'une façade verticale en fonction d'une acquisition au sol (1) et au drone (2)

VIII-2.3 SCÈNE COMPLEXE AVEC MATÉRIAUX NON-LAMBERTIENS : CHAPELLE NOTRE-DAME DE COVIS.

La chapelle Notre-Dame de Covis est une scène très complexe avec des angles dans les façades, un recouvrement en métal et des façades avant et arrière vitrées. La technique des NeRF a été popularisée spécialement pour sa capacité à rendre ce type de scène (grâce à l'intégration de la composante directionnelle). Les résultats de l'acquisition ont été obtenus en appliquant le premier workflow (le seul à pouvoir rendre une scène complexe) et sont extrêmement encourageants (Figure 50). En plus d'être précis, les rendus volumétriques représentent parfaitement la scène, y compris l'intérieur de la chapelle visible à travers les verrières (autel et chaise). La projection permet facilement de discerner le maillage de tôles de recouvrement et la délimitation de la porte en verre. D'ailleurs, le verre n'a pas obstrué la génération du PCD et n'est pas représenté dans le nuage de points.

Un point intéressant à noter est que la structure non planaire de la chapelle rend la sélection de points de l'expérience 3 particulièrement ardue puisqu'elle a été pensée pour capturer un plan objet rectangle avec 4 points à ses coins. Le workflow 2 ne pouvant traiter que les scènes uniplanaires, la vue orthographique n'a pu être produite qu'avec le workflow 1.

(1)

(2)

Figure 50 : Vue de la chapelle Notre-Dame de Covis : (1) Représentation volumétrique et (2) Nuage de points dans CloudCompare.

Les différentes acquisitions secondaires ont permis une comparaison avec le jeu de données de références. La conclusion est que la qualité de l'acquisition de référence est largement meilleure que toutes les autres, ce qui indique que la redondance des données (pour un même nombre d'étape d'optimisation) est néfaste pour la qualité des données. L'acquisition du jeu de données ayant effectué un seul trajet à 4m de la façade en faisant varier l'angle horizontal sur très peu de frame.

CHAPITRE IX DISCUSSION ET PERSPECTIVES

Grace à l'analyse des résultats effectuée au chapitre précédent, il est désormais possible de discuter l'ensemble du travail et de répondre à la question de recherche en fonction des hypothèses posées à partir de celle-ci (cf. Chapitre II). Des perspectives de développement futur à partir des principaux constats discutés sont également proposées.

IX-1 DISCUSSIONS DES RÉSULTATS ET DE LA QUESTION DE RECHERCHE.

Le but du travail était de répondre à la question de recherche :

Les avantages procurés par un NeRF permettent-ils de rendre plus accessibles les relevés de façades architecturales, surtout pour des utilisateurs non-initiés ?

Les résultats des expériences ont mis en évidence la possibilité d'obtenir des vues orthographiques d'une façade à partir de champs de radiances neuronaux. La troisième expérience en particulier a démontré que c'était possible tout en proposant une solution rapide et facile qui reste accessible à un public d'utilisateurs non-initiés aux techniques de pointe. Ainsi, au terme du travail, l'hypothèse 1 et l'hypothèse 2 ont pu être validées.

Pour répondre à la question de recherche, le travail parcourt l'intégralité des opérations d'un relevé classique depuis l'acquisition jusqu'à la production d'une représentation 2D mesurable mais en passant par une représentation 3D à l'aide d'un NeRF. Il a permis de mettre en évidence que même si le NeRF reste actuellement moins précis que la photogrammétrie(Croce et al., 2024), il permet une représentation 3D rapide des scènes réelles parfois complexes. Cette représentation ne nécessite que peu de matériel car l'acquisition peut être effectuée avec un smartphone commun et les traitements peuvent être délocalisés sur le web. Les interfaces utilisateurs peuvent être fortement simplifiées pour permettre l'accès à la création de relevés sans la compréhension du NeRF ou des SIG 3D comme l'a démontré l'expérience 3 et son workflow.

Cependant, la simplification à un prix. En effet, l'obtention de la vue orthographique et l'optimisation du modèle d'un NeRF nécessite une exécution précise et une bonne paramétrisation des différents traitements comme l'ont démontré les expériences 1 et 2. La simplification pour un public peu ou pas formé à la gestion de ces paramètres implique donc de restreindre le workflow en choisissant arbitrairement des paramètres pour pré-configurer chaque étape du workflow. Le workflow final proposé dans l'expérience 3 est donc accessible mais ne peut traiter que les cas par défaut. La mise en situation a permis d'identifier une série de cas particuliers qui sortent du cadre envisagé par le workflow 3 comme les scènes trop larges pour terminer l'optimisation du NeRF et les géométries complexes. Les cas particuliers doivent être traités avec le premier workflow (effectif) qui laisse un degré total de liberté mais qui n'est donc pas accessible à un public non-initié.

Les résultats du travail sont donc des workflows complémentaires avec une application web qui traitent une grande partie des cas de figure et qui est opérable par le public non-initié et un workflow générique pouvant traiter les cas particuliers sous réserve d'une compréhension plus poussée des NeRF.

IX-2 PERSPECTIVE DE DÉVELOPPEMENT

Le travail offre un grand nombre de possibilités de développement et de nouvelles perspectives pour la production de relevés architecturaux.

Un retour auto-critique des implémentations réalisées lors des expériences permet de dégager certains enseignements. Les trois expériences ont certes permis de prouver la possibilité de produire des projections mais elles sont très loin des standards de l'industrie en termes d'optimisation et de conception. De nombreuses fonctions n'ont pas pu être implémentées et l'ergonomie générale est encore largement perfectible. Le portage web public multi-support avec des serveurs associés à des machines puissantes pour les calculs du NeRF, la conception d'une BD SQL avec des nouvelles requête et les possibilités de mise à l'échelle (scalabilité) pour la gestion d'un grand nombre d'utilisateurs sont trois principaux axes directs pour continuer le développement.

Dans la même idée, le travail s'est plus largement concentré sur les traitements et la modélisation en dépit de l'acquisition puisque le public non-initié n'est pas sensé pouvoir réaliser une acquisition parfaite. Malgré une rapide mise en situation, il reste de nombreuses zones d'ombre dans la détermination d'une acquisition idéale. Notamment, à partir du tour simple avec une variation angulaire importante. Comme autre axe de développement autour de l'acquisition, il faudrait déterminer la distance idéale en fonction de la taille de l'objet, déterminer les besoins minimaux en termes de redondance des données, évaluer précisément la précision du NeRF en comparaison directe avec la photogrammétrie et le LiDAR et évaluer la possibilité de géoréférencement manuel ou automatique dans un SCR et sa précision.

Concernant l'optimisation des NeRF eux-mêmes, le champ d'amélioration est très large au vu de la dynamique actuelle de la recherche. L'optimisation de l'application passera forcément par une implémentation se basant uniquement sur pytorch ou d'autres langages encore plus bas niveaux. Notamment pour les cas qui nécessitent de sortir du cadre de Nerfstudio. L'intégration de modèles sémantiques comme Semantic-NeRF ((Zhi et al., 2021)) pour la détection de plan objet permettrait de passer au-dessus de l'étape de sélection et permettrait de gérer des façades complexes. Dans la même idée, l'utilisation des LERF ((Kerr et al., 2023)) et de modèles pré-entrainés qui permettent d'interroger le NeRF sur base du langage courant faciliterait encore plus les interactions avec les utilisateurs non-initiés.

Voici quelques idées concrètes d'applications pouvant intégrer les résultats de ce travail :

- Une application mobile IOS qui permettrait de fusionner l'application d'acquisition avec l'interface de modélisation et de projection. Les traitements étant réalisés après un upload en ligne. En quelque sorte, la fusion entre Polycam et LumaAi, mais pour le relevé de façade mise à l'échelle. L'implémentation d'une telle application constituerait un excellent axe de développement pour une potentielle quatrième expérience.

- Un logiciel tout en un pour le relevé de façade par drone (embarqué ou non). L'acquisition idéale serait préconfigurée en fonction d'un volume donné et d'une surface à scanner. Ensuite un planificateur de vol calculerait la vitesse de déplacement du drone ainsi que l'orientation de sa caméra pour optimiser la variation d'angle verticale et horizontale entre les prises de vues, éviter le flou de mouvement et assurer un recouvrement correct. La projection de la façade et la génération du PCD/MESH étant automatisée après le vol.
- Un logiciel ou une application mobile de suivi temporel des chantiers, des sites culturels endommagés ou des bâtiments sinistrés. L'outil permettrait de recapturer une vue à partir de points de référence constants ou d'une vérité terrain. La projection permettant de quotidiennement cartographier l'évolution des textures du bâtiment d'intérêts. Une combinaison avec les techniques classiques est facilement envisageable.
- La combinaison avec un Lidar plus puissant que celui d'Apple pour représenter précisément les objets à géométrie complexe comportant de nombreuses surfaces non-lambertiennes (avec de grandes et multiples verrières, construites avec des matériaux modernes).

Quoiqu'il en soit, les perspectives de développement (qu'elles découlent directement de ce travail ou projetées dans des applications futures) sont larges, pluridisciplinaires et ouvertes à des publics variés.

CHAPITRE X CONCLUSION

Les domaines de l'architecture et de la conservation du patrimoine recherchent activement une solution pouvant faciliter la démarche très utilisée de relevé de façade, notamment en réduisant son coût en temps, en matériel et en opérateur expert. Ce mémoire s'inscrit dans le récent mouvement des NeRF ouvert par Mildenhall et al., (2020) qui, par sa nouvelle approche, répond directement à cette demande.

Cette étude propose une méthodologie efficace pour évaluer si les avantages de cette nouvelle méthode de représentation 3D à partir d'une modélisation par NeRF est pertinente pour la problématique du relevé de façade.

Pour concrétiser la demande des architectes et archéologues, l'objectif de ce travail a été la création d'un workflow complet de relevés architecturaux par vue orthographique puis la simplification de celui-ci vers un public plus large de non-initiés.

Une série d'expériences successives a permis d'effectivement produire les différents workflows attendus pour répondre aux dits objectifs. Elles ont également démontré que la production de projections orthographiques est possible grâce à la combinaison d'une acquisition LiDAR Polycam et d'une modélisation sous Nerfstudio; confirmant que la méthode permet de rapidement représenter des scènes complexes multiplanaires incorporant des matériaux réfléchissants grâce à la nature directionnelle du NeRF (un avantage propre).

De plus, une grande partie des traitements peuvent être automatisés pour faciliter l'accessibilité d'un large public varié et pouvant s'étendre au-delà du domaine de l'architecture et de la conservation.

À cette fin, une interface simplifiée avec des chaines de boutons et une application web utilisant une méthode inédite de sélection par densité volumétrique et la connaissance intuitive de l'utilisateur pour paramétrer sa propre projection ont été créées. L'implémentation de ces deux applications a pu être réalisée et peut être facilement déclinable pour rencontrer les exigences du public professionnel. Le relevé architectural qui en découlera sera rapide, ergonomique et ne nécessitera pas ou peu de connaissance théorique.

Les résultats sont très encourageants même si le développement doit continuer pour prendre en compte toute la diversité des cas possibles d'acquisition et de public. En particulier, la scalabilité des scènes de grande envergure mériterait d'être investiguée plus avant.

Les perspectives pour le futur sont multiples et prometteuses telles une application mobile IOS, un logiciel tout-en-un pour le relevé par drone entre autres.

En conclusion, ce mémoire a démontré qu'il était possible d'effectuer un relevé de façade en utilisant la technologie NeRF, et ce pour tout type d'utilisateur. Les résultats obtenus laissent entrevoir de belles avancées pour le futur, en particulier dans les domaines de l'archéologie et de l'architecture en rendant possible des relevés de façades de haute qualité, plus économes en ressources et accessibles à tout public.

CHAPITRE XI BIBLIOGRAPHIE

Alby, E. (2006). *Élaboration d'une méthodologie de relevé d'objets architecturaux Contribution basée sur la combinaison de techniques d'acquisition*. Docteur, Nancy, Université de Nancy I Henri Poincaré, inédit, 304p. https://www.researchgate.net/publication/32228350.

Anaconda. (s.d.). *Anaconda Documentation.* Consulté le 20 juillet 2025. https://www.anaconda.com/docs/getting-started/miniconda/mainv

Apple. (2025). *IPhone Support*. Consulté le 20 juillet 2025. https://www.apple.com/by/iphone-13-pro/

Artec 3D. (2025). *What is LiDAR?* Consulté le 20 juillet 2025. https://www.artec3d.com/fr/learning-center/what-is-lidar

Banfi, F., Brumana, R., Salvalai, G. & Previtali, M. (2022). Digital Twin and Cloud BIM-XR Platform Development: From Scan-to-BIM-to-DT Process to a 4D Multi-User Live App to Improve Building Comfort, Efficiency and Costs. *Energies*, *15*(12). https://doi.org/10.3390/en15124497.

Barron, J.T., Mildenhall, B., Tancik, M. Hedeman, P.,Martin-Brualla, R. & Srinivasan, P.P. (2021) Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. UC Berkley.

Billen, R. (2020). Lever: Topographie partim A. Notes de cours. *ULiège*.

Boje, C., Mack, N., Kubicki, S., Lopez Vidal, A., Casado Sánchez, C., Brassier, P. & Dugué, A. (2023). Digital Twin systems for building façade elements testing. *Proceedings of the European Conference on Computing in Construction*, Heraklion, 10 juillet 2023. https://doi.org/10.35490/EC3.2023.240.

Borkowski, A. S. & Kubrat, A. (2024). Integration of Laser Scanning, Digital Photogrammetry and BIM Technology: A Review and Case Studies. *Eng*, *5*(4), 2395–2409. https://doi.org/10.3390/eng5040125.

Cloud Compare. (s.d.). *3D point cloud and mesh processing software Open Source Project*. Consulté le 20 juillet 2025. https://www.danielgm.net/cc/

CloudCompare. (2024). *Downloads.* Consulté le 20 juillet 2025. https://www.cloudcompare.org/release/

CloudCompare. (2016). *Facets (plugin)*. Consulté le 20 juillet 2025. https://www.cloudcompare.org/doc/wiki/index.php?title=Facets_%28plugin%29

CloudCompare. (2015). *Forum: Tools>fit>Plane and RANSAC*. Consulté le 20 juillet 2025. https://www.danielgm.net/cc/forum/viewtopic.php?t=1201

CloudCompare. (2022). *RANSAC Shape Detection (plugin)*. Consulté le 20 juillet. 2025https://www.cloudcompare.org/doc/wiki/index.php/RANSAC_Shape_Detection_(plugin)

Cursor. (2025). *Welcome. Learn abour Cursor and how to get started.* Consulté le 20 juillet 2025. https://docs.cursor.com/welcome.

Costantino, D., Pepe, M. & Restuccia, A. G. (2023). Scan-to-HBIM for conservation and preservation of Cultural Heritage building: the case study of San Nicola in Montedoro church (Italy). *Applied Geomatics*, *15*, 607–621. https://doi.org/10.1007/s12518-021-00359-2/Published.

Croce, V., Billi, D., Caroti, G., Piemonte, A., De Luca, L. & Véron, P. (2024). Comparative Assessment of Neural Radiance Fields and Photogrammetry in Digital Heritage: Impact of Varying Image Conditions on 3D Reconstruction. *Remote Sensing*, *16*(2). https://doi.org/10.3390/rs16020301.

ESRI. (s.d.). *Support Technique, Dictionnaire SIG. Vue Orthographique.* Consulté le 20 juillet 2025. https://support.esri.com/fr-fr/gis-dictionary/orthographic-view

Farhat, A., Sattar, A., Visalli, A. & Jones, B. (2017). *The Accuracy of 3D Mesh Generation of Neuralangelo vs. Photogrammetry and LiDAR Technology*. https://doi.org/10.1109/ACCESS.2022.Doi.

Gill, C. (2004). Visualizing continuity between 2D and 3D graphic representations. *International Engineering and Product design Education Conference*. Delft, 2 septembre 2004. https://www.researchgate.net/publication/238661630.

Github (2025) PolyCam/polyform. https://github.com/PolyCam/polyform Consulté le 20 juillet 2025

Kerr, J., Kim, C. M., Goldberg, K., Kanazawa, A. & Tancik, M. (2023). *LERF: Language Embedded Radiance Fields.* http://arxiv.org/abs/2303.09553.

Kong, X. (2024). Monitoring Time-Varying Changes of Historic Structures Through Photogrammetry-Driven Digital Twinning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48(2–2024)*, 181–186. https://doi.org/10.5194/isprs-archives-XLVIII-2-2024-181-2024.

Lalonde, J.F. (2025). *Module de Cours. Introduction à la photograhie algorithmique*. Université de Laval. Consulté le 20 juillet 2025. GIF-4105/7105 Photographie Algorithmique H25

Lucidchart (2025) What is a workflow? Benefits and examples of repeatable processes. https://www.lucidchart.com/blog/what-is-workflow Consulté le 20 juillet 2025

Magri-Djenane, S., Madhoui, M. & Belarbi, S. (2011). Cours : Technique du relevé architectural. *Université Mohamed Khider Biskra*, 1–17. Cours : TECHNIQUE DU RELEVE ARCHITECTURAL.

Mildenhall, B. J. (2020). *Neural Scene Representations for View Synthesis*. Doctor of Philosophy, Berkeley, University of California, inédit, 77. EECS-2020-223.pdf.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. & Ng, R. (2020). *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.* Springer, Cham. *https://doi.org/10.1007/978-3-030-58452-8_24*. http://arxiv.org/abs/2003.08934. https://doi.org/10.1145/3503250

Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG), 41(4), 1-15. https://doi.org/10.1145/3528223.3530127.

My Survey Direct (2025) *GeoMax Zoom50 Reflectorless Total Station.* Consulté le 17 juillet 2025. https://www.mysurveyingdirect.com/collections/geomax-total-stations/products/geomax-zoom50-reflectorless-total-station

Nerfstudio. (2022). *Installation.* Consulté le 20 juillet 2025. https://docs.nerf.studio/quickstart/installation.html

Nostalgie Lustinoise. (2021). La chapelle Notre Dame de Covis. Consulté le 20 juillet 2025. https://essai.campingledouaire.be/la-chapelle-notre-dame-de-covis/

NVIDIA Développer. (2025). *CUDA Toolkit*. Consulté le 20 juillet 2025. https://developer.nvidia.com/cuda-toolkit

Paukkonen, N. (2023). Towards a Mobile 3D Documentation Solution. Video-Based Photogrammetry and iPhone 12 Pro as Fieldwork Documentation Tools. *Journal of Computer Applications in Archaeology*, 6(1), 143–154. https://doi.org/10.5334/jcaa.135.

Point Cloud Library. (1.15.0-dev). *Downsampling a PointCloud using a VoxelGrid filter*. Consulté le 20 juillet 2025. https://pointclouds.org/documentation/tutorials/voxel_grid.html

Point Cloud Library. (1.15.0-dev). *Estimating Surface Normals in a PointCloud*. Consulté le 20 juillet 2025 https://pointclouds.org/documentation/tutorials/normal_estimation.html

Point Cloud Library. (1.15.0-dev). *Removing outliers using a StatisticalOutlierRemoval filter*. Consulté le 20 juillet 2025. https://pointclouds.org/documentation/tutorials/statistical_outlier.html

PolyCam. (s.d.). *How to Extract Raw Data and What Is Included*. Consulté le 20 juillet 2025. https://learn.poly.cam/hc/en-us/articles/38276871185044-How-to-Extract-Raw-Data-and-What-Is-Included

Remondino, F., Karami, A., Yan, Z., Mazzacca, G., Rigon, S. & Qin, R. (2023). A Critical Analysis of NeRF-Based 3D Reconstruction. *Remote Sensing*, 15(14). https://doi.org/10.3390/rs15143585.

Russo, M., Carnevali, L., Russo, V., Savastano, D. & Taddia, Y. (2019). Modeling and deterioration mapping of façades in historical urban context by close-range ultra-

lightweight UAVs photogrammetry. *International Journal of Architectural Heritage*, 13(4), 549–568. https://doi.org/10.1080/15583058.2018.1440030.

Scikit learn. (2025). 2.*5. Decomposing signals in components (matrix factorization problems.* Consulté le 20 juillet 2025. https://scikit-learn.org/stable/modules/decomposition.html#pca

SPW. (2018). *Inventaire du Patrimoine Culturel Immobilier : église paroissiale (Eglise Saint-Martin)*. Consulté le 17 juillet 2025. http://www.wallonie.be/patrimoine/ipic.

Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., Mcallister, D., Kerr, J. & Kanazawa, A. (2023). Nerfstudio: A Modular Framework for Neural Radiance Field Development. *Proceedings - SIGGRAPH 2023 Conference Papers*, 1–13. https://doi.org/10.1145/3588432.3591516.

Université de Liège. (2023). *Charte ULiège d'utilisation des intelligences artificielles génératives dans les travaux universitaires*. Consulté le 20 juillet 2025. https://www.student.uliege.be/cms/c_19230399/fr/faq-student-charte-uliege-d-utilisation-des-intelligences-artificielles-generatives-dans-les-travaux-universitaires.

Vacca, G. (2023) 3D Survey with Apple LiDAR Sensor—Test and Assessment for Architectural and Cultural Heritage. *Heritage* 2023, 6, 1476–1501. https://doi.org/10.3390/heritage6020080

Vandenabeele, L., Hacki, M. & Pfister, M. (2023) Crowd-sourced surveying for building archaeology: The potential of structure from motion (SFM) and neural radiance fields (NERF). *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives,* 1599-1605

Villa, V., Pignatale, T. & Tramentozzi, I. (2017). The lapidary of Palazzo Ancarano. Digital technologies and 3D printing to improve knowledge of the stele collection. *Studies in Digital Heritage*, *1*(2), 769–784. https://doi.org/10.14434/sdh.v1i2.23252.

Zachos, A. & Anagnostopoulos, C.-N. (2024). Using Terrestrial Laser Scanning, Unmanned Aerial Vehicles and Mixed Reality Methodologies for Digital Survey, 3D Modelling and Historical Recreation of Religious Heritage Monuments. *ACM Journal on Computing and Cultural Heritage*, 17(4), 1-23. https://doi.org/10.1145/3679021.

Zhi, S., Laidlow, T., Leutenegger, S. & Davison, A. J. (2021). *In-Place Scene Labelling and Understanding with Implicit Scene Representation*. http://arxiv.org/abs/2103.15875.

Zhou, L., Wu, G., Zuo, Y., Chen, X. & Hu, H. (2024). A Comprehensive Review of Vision-Based 3D Reconstruction Methods. *Sensors,* 24(7). https://doi.org/10.3390/s24072314.

ANNEXE 1 : CAMPAGNE DE DONNÉES DES SITES SECONDAIRES

Pour évaluer la qualité d'une reconstruction, un rapport est généré (sous format .json) avec *ns-eval* dans Nerfstudio. Le rapport évalue trois indices de qualité des images produites et deux indices de performance :

1) Le PSNR (Peak Signal-to-Noise Ratio) qui quantifie le bruit (la différence entre l'image estimée et la vérité terrain). Il se mesure en décibel, des hautes valeurs indiquent une bonne ressemblance.
2) Le SSIM (Structural Similarity Index) qui identifie la similarité de structures (entre l'image estimée et la vérité terrain). Sur une échelle normalisée, proche de 1 veut dire structure similaire.
3) LPIPS (Learned Perceptual Image Patch Similarity) : un indice basé sur un réseau de neurones similaire à la vision assistée par ordinateur (imite la vision humaine). À l'inverse, sur une échelle normalisée, proche de 1 veut dire que l'image est perçue différemment.
4) NRPS (Number of Rays Per Second) : le nombre de rayons générés par seconde par le modèle, plus le modèle en trace, plus il est rapide à converger.
5) FPS (Frames Per Second) : la vitesse de rendu volumétrique.

JEU DE DONNÉES DE RÉFÉRENCE

Les indices du jeu de données de référence pour comparaison.

Nombre de frame	PSNR	SSIM	LPIPS	NRPS	FPS
56	**17,75(+-2.45)**	**0.45(+-0.12)**	**0.34(+-0.05)**	545081	**0.74**

FAÇADE LARGE DE L'ÉGLISE D'IVOY.

Ont été testés: les allers et retours manuels à distance fixe du mur (2, 3, 4, 5 et 10m) avec un changement d'angle vertical entre chaque trajet, des aller-retours avec une distance qui augmente progressivement mais avec un angle fixe (manuel, perche et drone) et une acquisition au drone avec un balayage horizontal à 4m de distance du murs).

Les distances fixes du mur étant matérialisées par des cordes tendues au sol.

Figure : Acquisition manuelle à l'église d'Ivoy par aller-retours à une distance fixe (4m du mur, angle vertical 20°). On peut voir les guides au sol et la perche.

Type	Nombre de frame	PSNR	SSIM	LPIPS	NRPS	FPS
Angles verticaux à 2m	301	13,21(+-0.68)	0.15(+-0.9)	0.66(+-0.12)	575270	0.78
Angles verticaux à 3m	327	13.29(+-1.01)	0.16(+-0.13)	0.56(+-0.13)	566781	0.77
Angles verticaux à 4m	321	13.35(+-0.85)	0.20(+-0.16)	0.48(+-0.16)	578700	0.78
Angles verticaux à 5m	299	13.27 (+-1.02)	0.18 (+-0.17)	0.48 (+-0.18)	554751	0.75
Angles verticaux à 10m	**258**	**15,74 (+- 1.2)**	**0.33 (+-0.21)**	**0.39 (+-0.24)**	**537548**	**0.73**
Recul parallèle	348	13,24 (+-0.66)	0.09(+-0.03)	0.63 (+-0.08)	191179	0.26
Perche parallèles	330	13.80(+-0.87)	0.13(+-0.08)	0.60(+-0.07)	567710	0.77
Drone à balayage horizontale	394	13,06(+-1.10)	0.12(+-0.06)	0.63(+-0.08)	567321	0.77
Drone parallèle	399	12.75(+-0.99)	0.17(+-0.10)	0.48(+-0.14)	**577539**	0.78

FAÇADE VERTICALE : MAISON EN PIERRE

Type	Nombre de frame	PSNR	SSIM	LPIPS	NRPS	FPS
Sol	109	15,07(+-1.03)	0.26(+-0.08)	**0.38(+-0.11)**	568372	0.77
Drone	260	14.34(+-1.14)	0.23(+-0.09)	0.62(+-0.11)	570856	0.78

MATÉRIAUX NON-LAMBERTIENS : CHAPELLE RUE COVIS

Nombre de frame	PSNR	SSIM	LPIPS	NRPS	FPS
559	14.78(+-1.63)	**0.41(+-0.15)**	0.63(+-0.12)	567222	0.77

ANNEXE 2 : UTILISATION DE L'IA

Tout a été fait pour strictement rester dans le respect de la Charte ULiège d'utilisation des intelligences artificielles génératives dans les travaux universitaires : « L'ULiège soutient et développe chez ses étudiants et étudiantes des usages réfléchis, responsables, critiques et transparents de l'IA, pour tirer le meilleur parti de ce puissant auxiliaire. » (ULiège, 2023).

ChatGPT : L'intelligence artificielle générative est désormais bien connue et son usage encadré par la charte. ChatGPT n'a pas été utilisé pour rédiger ce mémoire ou des parties de ce mémoire, il n'a pas non plus aidé à le structurer (usage réfléchi). Il a principalement aidé dans la recherche documentaire et celle des bibliothèques solutions. Les sources sont demandées ET consultées dans chaque requête qui a été faite (usage critique). De plus, l'usage de ChatGPT pour la recherche documentaire s'est faite après la recherche sur ULiège Library. Son utilisation est signifiée dans le mémoire (usage transparent). Il est à noter que, au départ, ChatGPT canevas était utilisé pour le débogage des codes mais cette utilisation a été rapidement arrêtée au profit de Cursor car elle faisait trop d'erreurs par manque de contexte et les sources étaient difficilement identifiables.

Cursor : Cursor est un environnement de développement (éditeur de code) sorti en 2023. De façon similaire à GitHub copilote ou le canevas de ChatGPT, Cursor utilise des LLM (large langage model), des IA génératives textuelles pour l'aide à l'écriture de codes. Il permet entre autre de proposer des solutions, des approches, de faire du débogage, d'analyser un code pour le comprendre ou l'optimiser, d'accélérer l'écriture,… Étant directement chargé dans l'environnement de travail, il a une vue sur l'ensemble des fichiers d'un projet. Il comprend et aide à comprendre le contexte de l'ensemble du projet contrairement aux IA génératives classiques ne voyant le code que partiellement.

Cursor intègre plusieurs outils spécifiques : la tabulation ou prédication multiligne (doit être validée par l'utilisateur avec tab), le mode agent (en fonction de la requête, l'IA peut lire et modifier le code à travers tous les fichiers du projet et ce de façon autonome), l'édition sélective (on peut pointer à l'IA les lignes à modifier et voir les changements qu'elle effectue, cela empêche la dispersion), le mode chat (interface de conversation avec historique et checkpoint qui ne peut pas appliquer de modification autonome), les règles (on peut définir des règles d'utilisation, des conventions spécifiques au projet, des préférences générales), la mémoire (Cursor stocke le contexte, les préférences et les décisions ayant été faites précédemment et qui peuvent être appelés à tout moment), un MCP (un « model contexte Protocol » pour intégrer des outils, base de données et documentation externe), l'indexation (permettant la recherche à travers tous les fichiers) et les modèles multiples (Cursor tire avantage de plusieurs Ia génératives : chatgpt4o – 4.1, claude 3.5-4 sonnet, o3, gemini 2.5 pro) (Cursor, 2025).

Ce sont ces nombreux outils qui permettent d'augmenter massivement la productivité tout en assurant un respect de la charte.

Concernant l'usage réfléchi, Cursor peut être limité pour que chaque suggestion et modification ne s'applique que si l'utilisateur les valide individuellement (mode chat, tabulation et règles). De plus les règles permettent de limiter la recherche uniquement dans une documentation fournie au préalable (donc avec des auteurs connus et vérifiés, sans ajouter de recherche web). Le mode chat permet d'analyser les modifications et les sourcer si besoin. Pour rester dans un

usage réfléchi, le mode agent ne sera pas activé (n'impliquant plus forcément la compréhension de l'utilisateur).

Concernant l'usage critique, le mode chat, la tabulation et l'édition sélective permettent un contrôle de chaque modification, l'utilisateur peut toujours interroger le chat et discuter avant d'appliquer une modification. Il arrive souvent de refuser une modification qui est redondante ou erronée. On peut indiquer des solutions précédemment trouvées ou externes pour éviter que l'IA n'essaye d'appliquer en boucle la même erreur (ce qui était très courant avec ChatGPT). Pour assurer l'usage critique, une analyse des codes ajoutés est effectuée à chaque version d'une implémentation. Durant l'analyse, toutes les articulations vitales (ou les ajouts par rapport à la version précédente) sont pointées et on interroge le mode chat à propos des sources et exemples d'utilisation de ces morceaux de codes. De même lors du débogage, plusieurs solutions sont recherchées.

Pour finir avec l'usage transparent, l'utilisation de Cursor est largement signifiée dans le mémoire (cette section) et en commentaires des différents scripts fournis. Beaucoup de modifications, le débogage et le nettoyage final du code ont été réalisés avec Cursor (donc des IA), il est donc attendu d'en retrouver les traces dans les scripts.

Pour conclure, l'usage des IA dans ce mémoire est récurrent mais ne s'est jamais fait au détriment de la compréhension de l'auteur (c'est même l'inverse) et toujours dans le respect des sources en évitant activement le plagiat.

ANNEXE 3 : LOG DU TERMINAL DE L'EXPERIENCE 2

Voici les feedbacks obtenus dans le terminal de l'expérience 2. Ils ont été générés lors d'une session type pour l'export des projections des six façades du jeu de données de référence.

Data zip file selected: D:/DATA_master_2_2/Thesis/avril6 chapelle stlu 1tour/6_4_2025.zip

Output folder selected: D:/DATA_master_2_2/Thesis/implementation2

Starting data converter at 2025-08-11 03:38:16

Command: C:\ProgramData\miniconda3\Scripts\activate.bat && conda activate nerfstudio && ns-process-data polycam --data "D:/DATA_master_2_2/Thesis/avril6 chapelle stlu 1tour/6_4_2025.zip" --output-dir "D:/DATA_master_2_2/Thesis/implementation2" && exit

Thread launched at 2025-08-11 03:38:16

Starting NeRF reconstruction (nerfacto) at 2025-08-11 03:38:39

Command: C:\ProgramData\miniconda3\Scripts\activate.bat && conda activate nerfstudio && ns-train nerfacto --data "D:/DATA_master_2_2/Thesis/implementation2" && exit

Thread launched at 2025-08-11 03:38:39

Command: ns-export pointcloud --load-config outputs\implementation2\nerfacto\2025-08-11_033850\config.yml --output-dir D:\DATA_master_2_2\Thesis\implementation2 --num-points 1000000 --remove-outliers True --normal-method open3d --save-world-frame True --obb_center 0.0000000000 -0.0099999998 0.0700000003 --obb_rotation 0.0000000000 0.0000000000 0.0000000000 --obb_scale 0.1000000015 0.2700000107 0.5799999833

Thread launched at 2025-08-11 04:08:37

Launched command: ns-export pointcloud --load-config outputs\implementation2\nerfacto\2025-08-11_033850\config.yml --output-dir D:\DATA_master_2_2\Thesis\implementation2 --num-points 1000000 --remove-outliers True --normal-method open3d --save-world-frame True --obb_center 0.0000000000 -0.0099999998 0.0700000003 --obb_rotation 0.0000000000 0.0000000000 0.0000000000 --obb_scale 0.1000000015 0.2700000107 0.5799999833

Command: C:\ProgramData\miniconda3\Scripts\activate.bat && conda activate nerfstudio

Thread launched at 2025-08-11 04:09:27

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 1.ply

Point cloud aligned and projected successfully

Outliers removed with nb_neighbors=20, std_ratio=1.0

Point cloud aligned and projected successfully

Point cloud downsampled with voxel size: 0.05

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 1 projection.png

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 2.ply

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 2 projection.png

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 3.ply

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 3 projection.png

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 4.ply

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 4 projection.png

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 5.ply

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 5 projection.png

Point cloud loaded successfully: D:/DATA_master_2_2/Thesis/implementation2/Facade 6.ply

Point cloud aligned and projected successfully

Projection image saved at D:/DATA_master_2_2/Thesis/implementation2/Facade 6 projection.png

LISTE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

Nom	Acronyme
2 Dimensions, 3 Dimensions	*2D, 3D*
Application Programming Interface	*API*
Building Information Model	*BIM*
Banques de Données	*BD*
Computer-Aided Design	*CAD*
Command line interface	*CLI*
Computer Processing Unit	*CPU*
Digital Twin (jumeau numérique)	*DT*
Environmental Systems Research Institute	*ESRI*
Global Navigation Satellite System	*GNSS*
Graphic Processing Unit	*GPU*
Inertial Measurement Unit	*IMU*
Intelligence Artificielle	*IA*
Identification Number	*ID*
IPhone Operating System	*IOS*
Light Detection and Ranging	*LiDAR*
Maillage polygonal	*MESH*
Multi Layer Perceptron	*MLP*
Neural Radiance Field	*NeRF*
Analyse en composantes principales	*PCA*
Real Time Kinematic	*RTH (RTK)*
Système de Coordonnées de Référence	*SCR*
Système d'Information Géographique	*SIG*
Structured Query Language	*SQL*
Point Cloud Data	*PCD*
Terrestrial Laser Scanning	*TLS*
Virtual Reality	*VR*