

Ecole Nationale des Ponts et Chaussées



Département d'Ingénierie Mathématique et Informatique

Approches avancées de vision par ordinateur pour la reconstruction 3D et la détection d'objets au plafond

Membres du groupe :

AKHYAR FATIMA-ZAHRAE
ARCHIDI DIAE
LIAB MERYEM

Professeur encadrant :

LEPETIT VINCENT

2024 / 2025

Table de contenu

1 Introduction	4
1.1 Contexte général	4
1.2 Problématique et objectifs	4
1.3 Intérêt et applications potentielles	5
2 Approche de la reconstruction 3D	5
3 Analyse visuelle et préparation des données	6
3.1 Détection de la géométrie dominante : extraction des plans	6
3.2 Modélisation par homographie	7
3.2.1 Modélisation des poses caméra en reconstruction 3D	7
3.2.2 Projection du plan image dans l'espace 3D sur le plan du plafond	7
3.2.3 Définition d'un repère 2D dans le plan du plafond	8
3.2.4 Détermination d'un rectangle englobant pour le plafond, dans le plan du plafond	9
3.2.5 Calcul de l'homographie entre l'image de départ et l'image cible	9
3.3 Fusion d'images : techniques de stitching	10
4 Segmentation automatisée : intégration du modèle Segment Anything	10
5 Résultats expérimentaux	11
6 Ouvertures et perspectives de recherche	13
7 Conclusion	13
8 Annexes	14
A Présentation de MAST3R	14
B Méthode RANSAC	15
C Homographie	16
D Segment Anything Model (SAM)	18
9 Bibliographie	19

Tout le travail réalisé dans le cadre de ce projet (Code / résultats) est disponible dans ce répertoire :

https://github.com/MeryemLiab/Projet_Departement_IMI

Nous tenons à remercier tout particulièrement M. LEPETIT Vincent, notre encadrant académique du projet, pour sa disponibilité, ses conseils avisés et son accompagnement tout au long de ce travail. Ses orientations ont été déterminantes pour la bonne conduite de ce projet.

Abstract

This research project, conducted in collaboration with Saint-Gobain, aims to develop an automated pipeline to identify, localize, and assess the reusability of modular ceiling tiles in office buildings, using photographs captured on site. Aligned with circular economy principles, the proposed method seeks to distinguish reusable components from damaged or obsolete ones to optimize material recovery. The pipeline integrates 3D reconstruction, geometric plane estimation, homographic projection, image rectification, and segmentation. We first employed COLMAP and MAST3R to generate a dense point cloud and recover camera poses. A robust ceiling plane was then estimated via RANSAC, enabling the rectification and stitching of perspective images into a top-down view. Segmentation was initially performed using the Segment Anything Model (SAM). To move toward a fully autonomous system, we initiated manual annotation with makesense.ai and plan to fine-tune a pretrained instance segmentation model on our domain-specific dataset. Preliminary results are promising and demonstrate the feasibility of applying vision-based methods to facilitate sustainable construction practices.



1 Introduction

1.1 Contexte général

Ce projet, réalisé en collaboration avec Saint-Gobain, s'inscrit dans une démarche d'économie circulaire visant à favoriser le réemploi des matériaux de construction plutôt que leur élimination, afin de réduire l'empreinte environnementale des rénovations. L'objectif principal a été de développer une méthode automatisée pour identifier, localiser et évaluer l'état de conservation des éléments présents sur les plafonds et cloisons dans des environnements tertiaires comme les bureaux.

Les plafonds peuvent contenir une diversité de composants techniques : luminaires, détecteurs de fumée, diffuseurs d'air, sprinklers, trappes d'accès, etc. L'enjeu est de repérer automatiquement les éléments encore réutilisables, en bon état, et compatibles avec une nouvelle configuration, tout en excluant ceux qui sont endommagés, obsolètes ou inadaptés. Une attention particulière a été portée à la capacité des outils développés à non seulement détecter ces objets, mais aussi à les comptabiliser automatiquement, en vue de faciliter l'inventaire sur chantier.

Pour répondre à ces défis, le projet a mobilisé des outils numériques exploitant des images captées sur site, afin d'évaluer de manière fiable l'état des composants des plafonds. Cette approche technologique permet de soutenir des pratiques de rénovation plus durables, en facilitant l'identification, le comptage et la valorisation des éléments existants dans une logique de réutilisation maîtrisée.

1.2 Problématique et objectifs

L'étude débute par un constat technique selon lequel il est difficile de capturer l'ensemble d'un plafond dans une seule image exploitable pour la détection automatique de ses différents éléments. En effet, les caméras standards possèdent un champ de vision trop limité pour couvrir la totalité du plafond en une seule prise, ce qui impose la capture de multiples images sous différents angles. Pour le cas des caméras à 360°, bien qu'offrant une vue plus large, elles introduisent des distorsions géométriques et des étirements de pixels, rendant les images peu adaptées à la détection.



FIGURE 1 – Difficultés de capture d'un plafond en une seule image. Un redressement géométrique est nécessaire pour corriger la perspective avant d'effectuer une segmentation fiable des éléments structurels.

Ce projet vise à concevoir une pipeline capable de générer automatiquement une vue redressée et orthonormée du plafond à partir d'images prises sous différents angles, en vue d'une segmentation automatique de ses éléments (fig :2). Deux axes majeurs structurent cette démarche :

- Redressement des images à perspective : Cette étape consiste à corriger les déformations induites par la perspective et à assembler les images résultantes afin de générer une vue redressée de l'ensemble du plafond. La méthode proposée s'appuie sur une approche de reconstruction 3D. Celle-ci permet de représenter les différentes images dans un référentiel commun, de modéliser le plan du plafond, et de projeter les vues partielles sur ce plan. Une matrice d'homographie, encodant la relation géométrique entre les images sources et l'image cible, est ensuite calculée pour produire une vue finale redressée, à la fois cohérente et exploitable.

- Segmentation de la vue orthorectifiée : Une fois la vue redressée obtenue, une segmentation automatique est appliquée afin de détecter les différents éléments présents dans le plafond. Cette phase démontre la faisabilité de la détection sur une surface planaire redressée, en l'occurrence le plafond, malgré la présence de structures répétitives telles que les dalles, les luminaires, les détecteurs ou autres composants architecturaux. L'objectif est de valider que la transformation géométrique appliquée en amont permet une segmentation fiable et exploitable.

Ce projet propose ainsi une solution, combinant des techniques de représentation et de géométrie 3D, de redressement d'images et de segmentation, en vue d'une analyse automatisée des plafonds à partir d'images multi-perspectives.

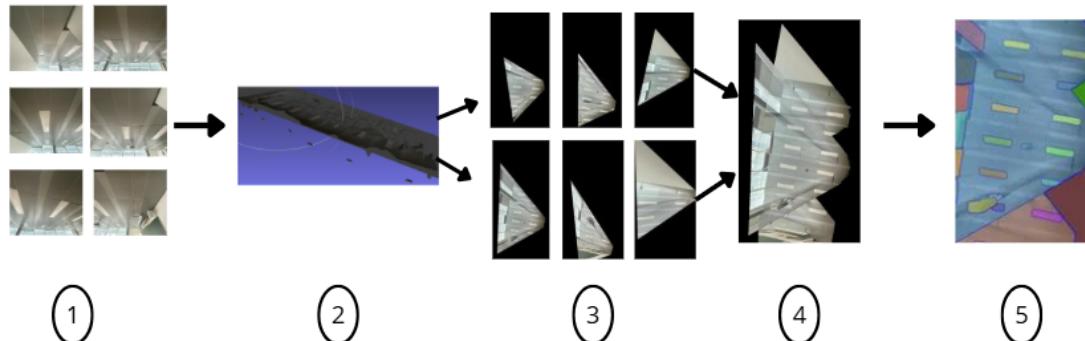


FIGURE 2 – Pipeline de la méthode proposée : (1) images en perspective, (2) reconstruction 3D avec **Mast3r** et détection du plan du plafond, (3) redressement des parties valides de chaque image en perspective, (4) assemblage et *stitching* des images redressées, (5) détection des éléments structurels du plafond avec **SAM**.

1.3 Intérêt et applications potentielles

Ce projet vise à permettre une évaluation fiable de l'état des matériaux à partir d'images prises in situ, en amont des opérations de rénovation. Cette démarche présente un double intérêt : elle favorise le réemploi des matériaux encore utilisables dans une logique de sobriété, et elle simplifie le travail des professionnels en facilitant le repérage préalable. Le plafond a été choisi comme cas d'étude en raison de ses nombreuses contraintes : surface en hauteur difficilement accessible, encombrée d'éléments techniques variés (luminaires, capteurs, ventilation), et souvent photographiée sous des angles peu favorables. Ces difficultés en font un excellent terrain pour tester une méthode automatisée de reconnaissance et d'inventaire. Malgré ces obstacles, les résultats sont prometteurs : ils montrent que des images prises dans des conditions réalistes permettent une estimation fiable de ce qui peut être conservé. Cela ouvre des perspectives concrètes, notamment pour mieux planifier la dépose, limiter les déchets, et anticiper les besoins en approvisionnement.

Cette approche est par ailleurs transposable à d'autres éléments visibles dans les bâtiments, comme les murs ou certains meubles intégrés. Elle pourrait ainsi être mobilisée dans de nombreuses situations de rénovation, d'aménagement ou de maintenance. À plus long terme, la méthode pourrait s'inscrire dans des processus plus larges de gestion du bâti : suivi de chantier, contrôle qualité ou conservation d'un état des lieux structuré à des fins de pilotage.

2 Approche de la reconstruction 3D

Dans le cadre de ce projet, la reconstruction 3D a été utilisée afin de générer une représentation spatiale fidèle des plafonds à partir d'un ensemble d'images capturées sous différents angles de vue. L'objectif principal était de produire un modèle 3D exploitable pour des analyses ultérieures.

Nous avons d'abord utilisé **COLMAP**, un outil open source de Structure-from-Motion (SfM) et de Multi-View Stereo (MVS), largement reconnu pour sa robustesse et sa précision. Il nous a permis de reconstruire un nuage de points dense à partir des images sous différents angles. Le processus inclut plusieurs étapes : la détection et la mise en correspondance des points clés (features), l'estimation des paramètres intrinsèques et extrinsèques des caméras, puis la reconstruction du nuage de points 3D. Cependant, bien que COLMAP offre d'excellentes performances dans un cadre général, nous avons constaté certaines limitations liées à nos conditions spécifiques de capture (ex. : faible texture sur les surfaces, éclairage non homogène, etc.).

Pour ces raisons, nous avons expérimenté l'outil **MAST3R** [Annexe A] (Multi-view Attention-guided Sparse-To-dense Reconstruction), qui repose sur des techniques d'apprentissage profond, notamment une attention guidée

permettant d'exploiter plus efficacement les redondances et les relations entre les vues. MAST3R s'est avéré plus adapté à nos besoins, notamment en raison de sa capacité supérieure à reconstruire des surfaces planes peu texturées, une caractéristique fréquente dans les images de plafonds. Le modèle reconstruit est ensuite exporté au format .glb, un format compact et optimisé pour la visualisation interactive. Cette représentation constitue une base solide pour les étapes suivantes de notre pipeline.

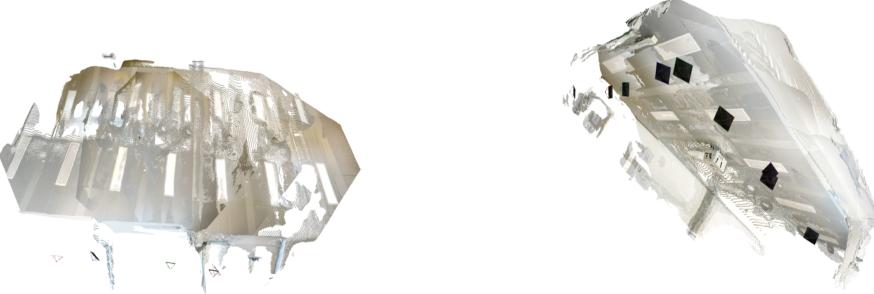


FIGURE 3 – Reconstruction 3D du plafond d'une salle obtenue avec *Mast3r*, illustrant à la fois le nuage de points décrivant la géométrie tridimensionnelle et les poses des caméras.

3 Analyse visuelle et préparation des données

3.1 Détection de la géométrie dominante : extraction des plans

À l'issue de la reconstruction 3D, nous avons procédé à l'identification du plan du plafond dans le nuage de points généré. Cette étape constitue une phase essentielle de notre pipeline, en fournissant un repère géométrique de référence pour la suite des traitements, notamment l'alignement de la scène, la segmentation des objets fixés au plafond et la projection 2D. Le fichier .glb issu de MAST3R contient un ensemble de géométries représentant les points reconstruits. En utilisant la bibliothèque trimesh, on extrait l'ensemble des coordonnées des points appartenant aux différents sous-ensembles du maillage. L'objectif initial était de collecter l'intégralité des points disponibles afin de disposer d'une base de données suffisamment riche pour l'estimation du plan du plafond.

Pour cela, on a utilisé un ajustement de type RANSAC (Random Sample Consensus). Le modèle considéré ici est celui d'un plan affine dans l'espace tridimensionnel, représenté par son vecteur normal et une constante d'offset. L'algorithme itératif RANSAC sélectionne aléatoirement des sous-ensembles de points et ajuste un plan à chaque itération, en évaluant à chaque fois le nombre d'inliers, c'est-à-dire les points dont la distance orthogonale au plan est inférieure à un seuil prédéfini. Le plan ayant obtenu le plus grand consensus est alors retenu comme estimation finale du plan du plafond. Le résultat de cette étape est double : d'une part, l'obtention explicite de l'équation du plan du plafond, utilisée comme référentiel dans les étapes ultérieures ; d'autre part, l'identification d'un sous-ensemble de points appartenant effectivement au plafond, sur lesquels pourront s'appuyer des traitements géométriques plus précis.

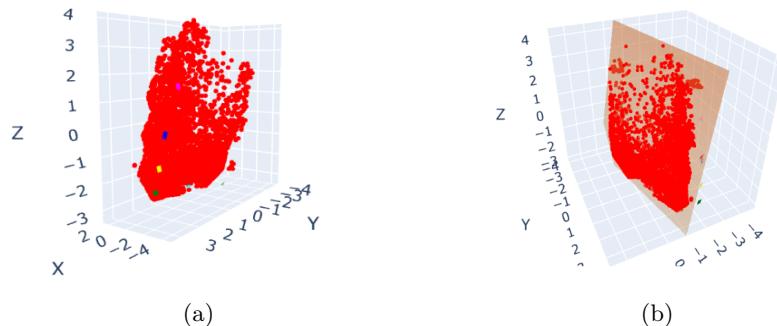


FIGURE 4 – Visualisation 3D de la scène reconstituée. (a) Nuage de points brut et poses des caméras estimées. (b) Même scène avec en plus le plan du plafond extrait automatiquement par RANSAC. Celui-ci identifie une surface plane dominante à partir des points 3D, en sélectionnant un sous-ensemble d'inliers cohérents avec un modèle plan.

3.2 Modélisation par homographie

Afin de générer une image unique représentant l'intégralité du plafond à partir des images en perspective utilisées en reconstruction 3D, nous avons développé une méthode permettant de calculer l'homographie entre chaque image en perspective et l'image cible orthorectifiée souhaitée. Chacune de ces images de départ contribue à la formation d'une partie de l'image finale complète, dans l'optique d'assembler, lors d'une étape ultérieure, les différentes sections redressées après application de l'homographie.

3.2.1 Modélisation des poses caméra en reconstruction 3D

Le calcul de l'homographie s'appuie sur les coordonnées des géométries 3D issues de la reconstruction. Le modèle MAST3R encode les poses des caméras en combinant deux géométries distinctes :

- Une géométrie rectangulaire à quatre points, notés A , B , C et D , représentant les coins de l'image dans l'espace 3D.
- Une géométrie à quatorze points, où l'on a identifié le sixième point, noté O , comme étant le centre optique de la caméra.

Ainsi, chaque pose de caméra est modélisée sous la forme d'une pyramide à base quadrilatère $ABCD$ et sommet O , illustrant le champ de vision de la caméra dans l'espace. (figure 5)

3.2.2 Projection du plan image dans l'espace 3D sur le plan du plafond

Après avoir déterminé les coins A , B , C et D appartenant au plan image dans l'espace 3D, ceux-ci sont projetés sur le plan du plafond en calculant l'intersection entre la droite reliant chaque coin au centre optique O et le plan du plafond. Le quadrilatère formé par ces points d'intersection représente la projection spatiale sur le plan du plafond de la portion à redresser de l'image d'origine.

Cependant, pour certaines poses de caméra, notamment en ce qui concerne les points inférieurs C et D , les rayons OC et OD ne sont pas orientés vers le plafond. Dans ces cas, l'intersection se produit derrière la caméra, ce qui rend la rétroposition géométriquement incohérente.

Pour résoudre ce problème, nous cherchons des points intermédiaires $C' \in [BC]$ et $D' \in [AD]$ tels que les rayons $O\vec{C}'$ et $O\vec{D}'$ soient orientés vers le plafond.

Le plan du plafond est défini par l'équation cartésienne :

$$\pi : ax + by + cz + d = 0 \quad (1)$$

où le vecteur $\vec{n} = (a, b, c)$ est la normale au plan.

On impose que \vec{n} pointe vers la caméra, ce qui se traduit par la condition : $\vec{n} \cdot O\vec{M} < 0$, où M est le centre géométrique du rectangle $ABCD$. Dans le cas du point C , pour garantir que le rayon $O\vec{C}'$ est orienté vers le plafond, on cherche un point C' tel que :

$$\cos(\theta) = \frac{\vec{v} \cdot \vec{n}}{\|\vec{v}\|} < 0, \quad \text{où } \vec{v} = O\vec{C}'.$$

Le point C' est alors interpolé sur le segment $[C, B]$ par : $C' = (1 - u) \cdot C + u \cdot B$, avec $u \in [0, 1]$.

Si $\cos(\theta) < 0$, cela signifie que le rayon forme un angle obtus avec la normale, donc qu'il pointe dans la direction opposée à \vec{n} . C'est exactement ce que l'on recherche, puisque \vec{n} pointe vers la caméra, et le rayon doit pointer vers le plafond. Enfin, afin de contrôler précisément l'angle d'incidence tout en conservant un maximum d'image, nous imposons :

$$\boxed{\cos(\theta) = -\varepsilon, \quad \varepsilon > 0.}$$

Cette contrainte permet de garantir une inclinaison minimale du rayon vers le plafond, assurant à la fois la validité géométrique et l'optimisation du champ de vision utile.

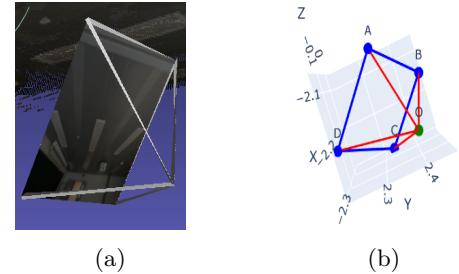


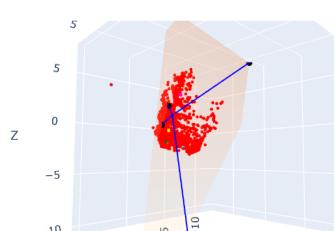
FIGURE 5 – Poses caméra : (a) Modélisation des poses caméra en 3D avec MAST3R, (b) Représentation géométrique d'une pose via une pyramide (base ABCD, sommet O).

Une formule similaire est utilisée pour déterminer le point D' sur le segment $[A, D]$, garantissant également que le rayon OD' pointe vers le plafond avec une inclinaison conforme à la contrainte $\cos(\theta) = -\varepsilon$. Ces nouveaux points C' et D' remplacent les coins originaux C et D dans la construction du quadrilatère projeté, assurant ainsi une projection géométriquement valide vers le plafond. Ce processus permet donc de corriger localement les limites du champ de vision dans les cas où certaines parties de l'image ne sont pas directement projectables, tout en conservant une zone utile maximale de l'image originale, sans compromettre la fidélité du redressement final.

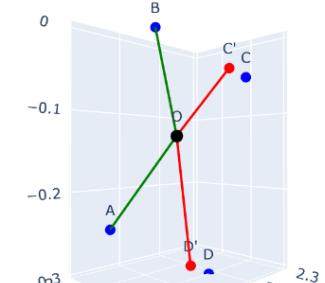
De ce fait, la portion de l'image cernée par les points 3D A, B, C' et D' correspond à correspond à la région de l'image dans le plan 2D définie par les points de coordonnées en pixel : $A_{img}(0, 0)$, $B_{img}(W, 0)$, $C'_{img}(0, (1-u)H)$ et $D'_{img}(W, (1-u)H)$, où W désigne la largeur de l'image en pixels et H sa hauteur. Ainsi, pour chacune des images en perspective, le redressement s'applique spécifiquement au quadrilatère $A_{img}B_{img}C'_{img}D'_{img}$, correspondant à la partie projectable de l'image vers le plan du plafond.



(a)



(b)



(c)

FIGURE 6 – a) Image source à projeter; b) Projection du plan image sur le plafond; c) Représentation des points projetés dans la scène 3D.

3.2.3 Définition d'un repère 2D dans le plan du plafond

Une fois le plan du plafond estimé, il est nécessaire de définir un repère 2D orthonormé dans ce plan afin d'y exprimer les coordonnées des points d'intersection entre les rayons reliant les coins A, B, C' et D' au centre optique O et le plan du plafond. Pour ce faire, on commence par déterminer le centre du nuage de points correspondant au plafond, noté $O_{plafond}$, en calculant le centre de son enveloppe rectangulaire :

$$O_{plafond} = \left(\frac{x_{\min} + x_{\max}}{2}, \frac{y_{\min} + y_{\max}}{2}, \frac{z_{\min} + z_{\max}}{2} \right).$$

Les bornes $x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}$ correspondent aux extrémités de l'enveloppe rectangulaire du nuage de points représentant le plafond.

Ce point est ensuite projeté orthogonalement sur le plan du plafond, défini par l'équation (1). La projection orthogonale d'un point M sur ce plan est donnée par : $\text{Proj}_\pi(M) = M + (-d - \vec{n} \cdot M) \vec{n}$.

La projection de $O_{plafond}$ sur le plan du plafond donne le point $O'_{plafond}$, qui est choisi comme origine du repère 2D. Deux points supplémentaires, notés P_x et P_y , sont sélectionnés sur les bords de l'enveloppe rectangulaire pour définir des directions initiales selon les axes X et Y. Ces points sont également projetés sur le plan du plafond :

$$P_x = \text{Proj}_\pi(M_x), \quad P_y = \text{Proj}_\pi(M_y), \quad \text{où } M_x \text{ et } M_y \text{ sont définis par :}$$

$$M_x = \left(x_{\max}, \frac{y_{\min} + y_{\max}}{2}, \frac{z_{\min} + z_{\max}}{2} \right), \quad M_y = \left(\frac{x_{\min} + x_{\max}}{2}, y_{\max}, \frac{z_{\min} + z_{\max}}{2} \right).$$

Le vecteur directeur de l'axe x du repère est alors :

$$\vec{u}_x = \frac{\vec{O'_{plafond}P_x}}{\|\vec{O'_{plafond}P_x}\|},$$

tandis que le vecteur \vec{v} reliant O'_{plafond} à P_y est orthonormalisé par rapport à \vec{u}_x pour obtenir \vec{u}_y :

$$\vec{v} = O'_{\text{plafond}} \vec{P}_y, \quad \vec{u}_y = \frac{\vec{v} - (\vec{v} \cdot \vec{u}_x)\vec{u}_x}{\|\vec{v} - (\vec{v} \cdot \vec{u}_x)\vec{u}_x\|}.$$

On obtient ainsi une base orthonormée (\vec{u}_x, \vec{u}_y) dans le plan du plafond, centrée en O'_{plafond} , servant à exprimer les coordonnées projetées en 2D des différentes intersections.

3.2.4 Détermination d'un rectangle englobant pour le plafond, dans le plan du plafond

Avec ces deux vecteurs orthonormés, on peut calculer pour chaque intersection notée $I(x, y, z)$ ses coordonnées $I_{2D}(x_{2D}, y_{2D})$ dans le repère 2D grâce au produit scalaire entre $O'_{\text{plafond}} \vec{I}$ et les vecteurs du repère :

$$x_{2D} = O'_{\text{plafond}} \vec{I} \cdot \vec{u}_x, \quad y_{2D} = O'_{\text{plafond}} \vec{I} \cdot \vec{u}_y$$

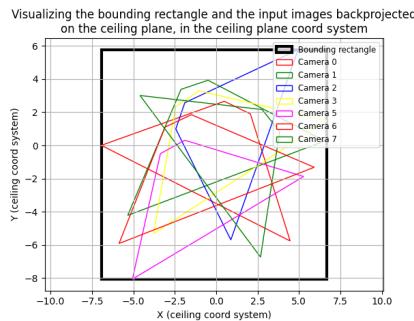


FIGURE 7 – Rectangle englobant et des images projetées sur le plan du plafond, dans le système de coordonnées du plafond

En parcourant l'ensemble des points projetés dans ce repère 2D, notés $I_{2D}^A, I_{2D}^B, I_{2D}^{C'} et $I_{2D}^{D'}$ pour chaque pose de caméra, on extrait les valeurs minimales et maximales des coordonnées projetées :$

$$x_{\min} = \min\{x_{2D}\}, \quad x_{\max} = \max\{x_{2D}\},$$

$$y_{\min} = \min\{y_{2D}\}, \quad y_{\max} = \max\{y_{2D}\},$$

ce qui permet de délimiter un rectangle englobant toutes les intersections au plafond, et donc l'ensemble des quadrilatères projetés depuis les images initiales. (figure 7)

Connaissant ensuite les dimensions de l'image à générer, notées W_{output} et H_{output} , on applique une transformation affine pour convertir chaque point I_{2D} en coordonnées pixels $I_{\text{img}}(x_{\text{img}}, y_{\text{img}})$:

$$x_{\text{img}} = \frac{(x_{2D} - x_{\min}) * W_{\text{output}}}{(x_{\max} - x_{\min})}, \quad y_{\text{img}} = \frac{(y_{2D} - y_{\min}) * H_{\text{output}}}{(y_{\max} - y_{\min})}$$

Cette opération permet de replacer, dans une image finale 2D, les portions projetées et redressées de chaque vue en perspective initiale.

3.2.5 Calcul de l'homographie entre l'image de départ et l'image cible

La dernière étape consiste à calculer la matrice d'homographie H pour chaque pose de caméra, en associant les coins d'une image source notés $A_{\text{img}}(0, 0)$, $B_{\text{img}}(W, 0)$, $C'_{\text{img}}(0, (1-u)H)$ et $D'_{\text{img}}(W, (1-u)H)$, où W désigne la largeur de l'image initiale en pixels et H sa hauteur, à leurs équivalents dans l'image finale 2D $I_{\text{img}}^A, I_{\text{img}}^B, I_{\text{img}}^{C'}$ et $I_{\text{img}}^{D'}$. Ces derniers sont obtenus à partir de la projection 3D des rayons sur le plan du plafond, puis de la conversion en coordonnées pixels des coordonnées 2D des intersections en 3D exprimés dans le repère du plan via la transformation affine décrite précédemment. En pratique, une fois H calculée, on applique son inverse H^{-1} pour, pixel par pixel, remapper chaque point de l'image finale vers l'image source. Ce remappage permet d'extraire la valeur du pixel de l'image source à l'aide d'une interpolation bilinéaire, uniquement si le point projeté se trouve à l'intérieur du polygone délimité par les coins $A_{\text{img}}, B_{\text{img}}, C'_{\text{img}}$ et D'_{img} .

Cette opération est répétée pour chaque image. On obtient ainsi N images redressées (avec N le nombre de poses de caméras), prêtes à être assemblées dans une image globale cohérente, représentant la totalité du plafond dans un espace 2D rectifié.

Chaque image redressée correspond à une partie du plafond. L'image finale se présente donc sous la forme d'un patchwork, dans lequel sont incluses successivement les différentes images corrigées.

Une première approche pour réaliser cet assemblage consiste à générer l'image résultante en calculant la moyenne des intensités des pixels des N images redressées. Cependant, cette méthode s'est avérée peu pertinente, car les positions des poses de caméra estimées par Mast3r lors de la reconstruction 3D ne sont pas parfaites, entraînant des décalages entre les images redressées qui ne se recouvrent pas correctement.

Pour pallier ces imprécisions, une étape supplémentaire d'alignement a été ajoutée, basée sur l'estimation d'une homographie entre les images à assembler. La méthode utilisée repose sur la détection automatique de points d'intérêt, suivie de l'estimation d'une transformation géométrique, homographie, entre paires d'images redressées issues de différentes poses de caméra. On commence par la détection de points d'intérêt caractéristiques dans chaque image, puis le calcul de descripteurs locaux associés à ces points, qui sont ensuite comparés entre les deux images afin d'établir des correspondances. La détection de points d'intérêt est délicate en raison de la structure répétitive du plafond (dalles, luminaires identiques, etc.). Toutefois, cette tâche est facilitée par le redressement préalable qui supprime la perspective, ainsi que par la présence dans chaque image d'éléments non plafonniers, tels que les murs ou les fenêtres, qui servent de repères robustes pour les algorithmes de détection.

Concrètement, la détection et la description des points d'intérêt sont effectuées à l'aide de l'algorithme AKAZE d'OpenCV et les correspondances entre descripteurs sont ensuite recherchées via un appariement par brute force avec recherche des deux meilleurs voisins kNN . Grâce à ces correspondances, il devient possible d'identifier les similarités entre les différentes parties des images (figure 8, (a)). L'estimation de l'homographie est réalisée à l'aide de l'algorithme RANSAC, utilisé précédemment pour l'estimation du plan du plafond, qui permet de rejeter les correspondances aberrantes c'est à dire les outliers et de ne conserver que celles compatibles avec une transformation géométrique cohérente. La matrice d'homographie ainsi obtenue sert à transformer l'une des images pour l'aligner précisément avec l'autre. Une fois alignées, ces images sont fusionnées afin de générer une vue globale. L'image résultante est ensuite initialisée avec l'une des deux images, puis complétée par les pixels non couverts à l'aide de la seconde, afin de générer une vue globale. (figure 8, (b))

4 Segmentation automatisée : intégration du modèle Segment Anything

Une fois le processus de stitching achevé et l'image redressée obtenue, nous avons entrepris une phase de segmentation d'image afin d'isoler les différents éléments structuraux présents sur le plafond. Cette étape permet de simplifier l'analyse de la scène en regroupant les pixels en régions homogènes correspondant à des objets ou structures significatifs. Pour cette tâche, nous avons utilisé le modèle Segment Anything (SAM) [Annexe D], développé par Meta AI et conçu pour généraliser à une grande variété d'objets sans nécessiter de réentraînement spécifique. Dans notre cas, il a été appliqué à l'image redressée du plafond, qui présente une vue en perspective corrigée des dalles et des objets suspendus comme les projecteurs ou les sprinklers.

L'efficacité de la segmentation peut être influencée par plusieurs paramètres importants, tels que la densité des

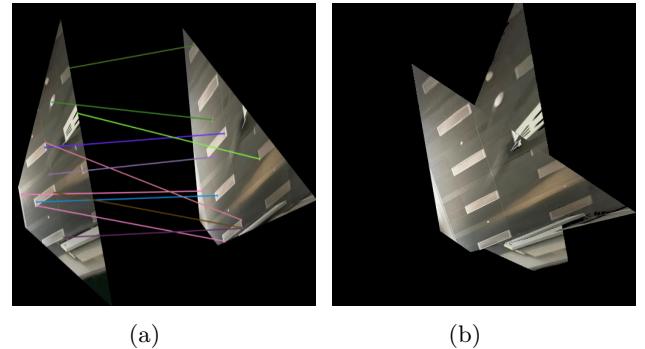


FIGURE 8 – (a) Visualisation des correspondances détectées par AKAZE. (b) Résultat du stitching des images avec l'homographie obtenue par RANSAC.

Bien que la structure du plafond soit fortement répétitive, rendant la détection des correspondances plus difficile, le stitching est possible grâce au redressement obtenu par la méthode MASt3R, ce qui n'était pas possible avant dans le cas des images en perspectives.

points prompts générés automatiquement et le seuil de confiance associé aux masques. En ajustant ces paramètres, il est possible d'optimiser la qualité des résultats, notamment dans les zones où les contours des objets sont bien définis.

Les résultats obtenus sont illustrés dans les figures suivantes.

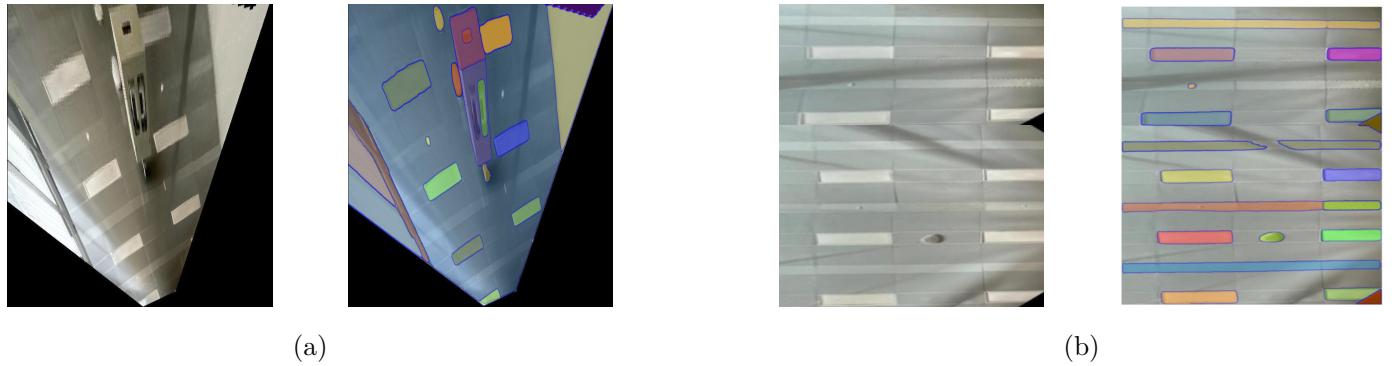


FIGURE 9 – (a) À gauche, image redressée du plafond avant segmentation ; à droite, résultat de la segmentation automatique obtenue avec Segment Anything. (b) segmentation appliquée sur une autre portion du plafond. Dans les deux cas, la détection des luminaires, des détecteurs de fumée et des projecteurs confirme la robustesse du modèle pour une lecture automatisée des structures techniques.

Les résultats obtenus montrent que le modèle Segment Anything parvient à identifier efficacement plusieurs objets d'intérêt présents dans la scène, tels que certaines dalles du plafond, le projecteur central ainsi que les sprinklers. Cette capacité à détecter des structures géométriques régulières témoigne de la pertinence de l'approche dans un contexte architectural intérieur. Toutefois, des limites apparaissent dans les zones où les transitions visuelles entre éléments sont peu marquées. En particulier, lorsque la séparation entre deux dalles adjacentes est peu contrastée ou partiellement obstruée, le modèle tend à fusionner ces régions, rendant la segmentation moins précise.

5 Résultats expérimentaux

On présente dans cette section un exemple de plafond redressé. Ce dernier est situé dans une salle de cours (à l'école des Ponts) présentant une structure relativement répétitive, constituée de dalles rectangulaires disposées de manière régulière, avec d'autres éléments architecturaux (luminaires ou projecteur). Six vues ont été capturées depuis des angles variés, permettant un recouvrement suffisant pour la reconstruction 3D. On rapporte dans les figures 11, 12, 13 les résultats obtenus lors des traitements intermédiaires sur ces images sources en fig.10 :

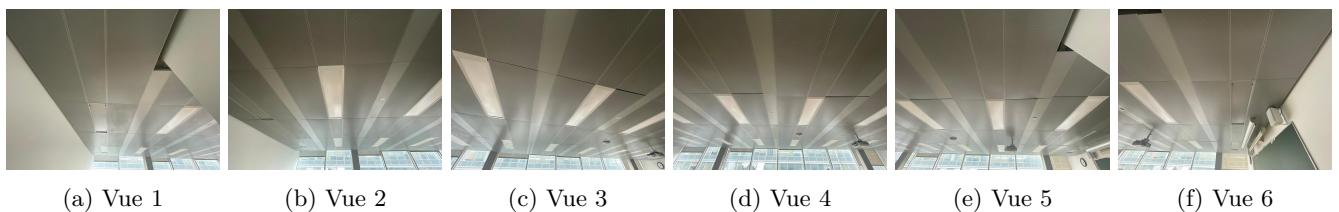


FIGURE 10 – Images en perspective prises sous différents angles de vue.

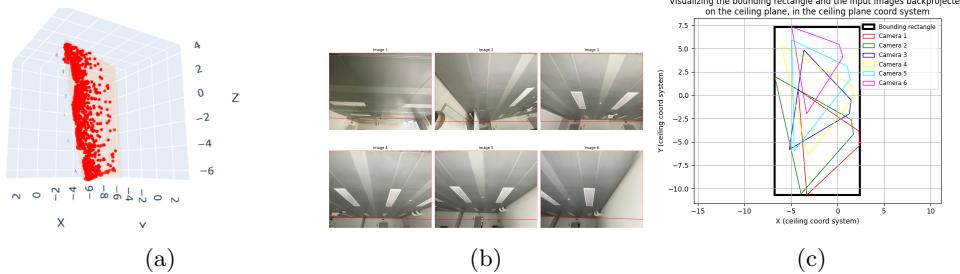


FIGURE 11 – Traitements intermédiaires sur les images capturées : a) reconstruction 3D et visualisation du plan du plafond, b) portion de chaque image en perspective à projeter sur le plan du plafond pour obtenir une projection valide, c) visualisation du rectangle englobant et des images projetées sur le plan du plafond, dans le système de coordonnées du plafond.

$$\begin{aligned} H_1 &= \begin{bmatrix} 1.6225e-3 & -9.3633e-1 & 1.0000e3 \\ 1.1315e-1 & -8.7539e-1 & 9.3492e2 \\ 9.2091e-6 & -8.5344e-4 & 1.0000e0 \end{bmatrix} & H_2 &= \begin{bmatrix} -2.4634e-2 & -8.3012e-1 & 9.2623e2 \\ 9.7121e-2 & -9.5416e-1 & 1.0658e3 \\ -6.9598e-6 & -8.1695e-4 & 1.0000e0 \end{bmatrix} & H_3 &= \begin{bmatrix} 1.5046e-2 & -7.9741e-1 & 8.7548e2 \\ 1.0228e-1 & -1.1050e0 & 1.3019e3 \\ 1.1015e-6 & -8.0744e-4 & 1.0000e0 \end{bmatrix} \\ H_4 &= \begin{bmatrix} -2.6279e-3 & -8.5525e-1 & 9.2699e3 \\ 1.3384e-1 & -1.3340e0 & 1.4744e3 \\ 1.7960e-5 & -8.4574e-4 & 1.0000e0 \end{bmatrix} & H_5 &= \begin{bmatrix} 4.2554e-3 & -8.0503e-1 & 8.8523e2 \\ 1.5608e-1 & -1.5036e0 & 1.7017e3 \\ 3.0732e-5 & -8.3282e-4 & 1.0000e0 \end{bmatrix} & H_6 &= \begin{bmatrix} 1.7587e-2 & -7.4143e-1 & 8.0143e2 \\ 2.2104e-1 & -1.7659e0 & 1.9656e3 \\ 6.0555e-5 & -8.5195e-4 & 1.0000e0 \end{bmatrix} \end{aligned}$$

FIGURE 12 – Matrices d’homographie estimées pour les images 1 à 6.

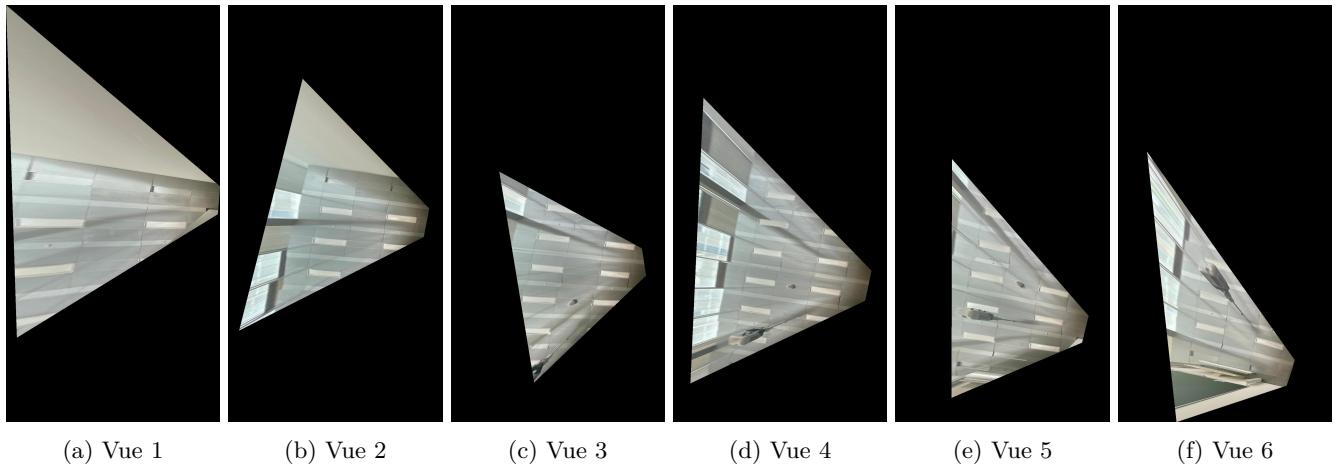


FIGURE 13 – Images redressées correspondant aux différentes vues du plafond.

L’application du pipeline aboutit à l’image finale suivante, support de la segmentation avec SAM : (fig 14)

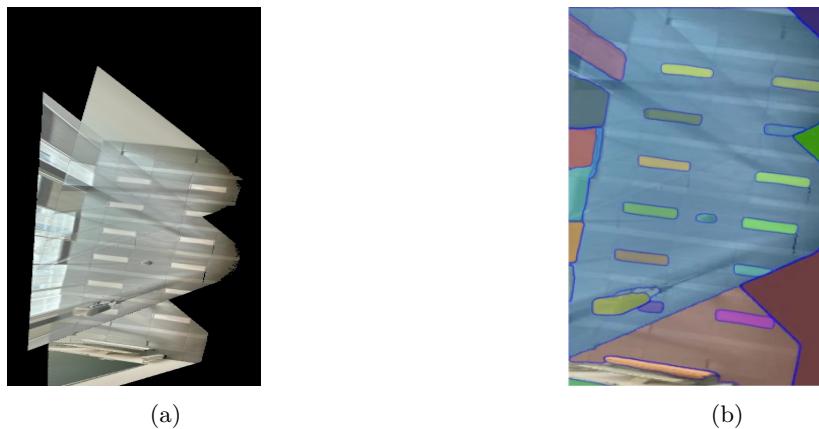


FIGURE 14 – Résultats : (a) assemblage des images redressées, (b) détection des éléments structurels du plafond.

Les résultats obtenus démontrent que la méthode développée permet d’atteindre les objectifs fixés au début du projet. En effet, grâce au redressement géométrique des images prises sous différents angles, il est désormais possible

de générer une vue exploitable du plafond, sur laquelle la détection automatique des éléments techniques peut être réalisée de manière cohérente.

6 Ouvertures et perspectives de recherche

À ce stade du projet, notre approche s'appuie sur l'utilisation du modèle **Segment Anything** pour l'étape de segmentation. Bien que cette solution présente une grande flexibilité et permette des résultats satisfaisants sans nécessiter d'entraînement spécifique, elle ne constitue qu'une première étape vers l'objectif final du projet. En effet, à terme, l'idéal serait de disposer d'un modèle capable de prédire directement les objets d'intérêt dans les images, sans intervention manuelle ni étape intermédiaire de segmentation assistée. Dans cette perspective, une piste naturelle d'évolution du projet consiste à développer un système capable de détecter et segmenter automatiquement les objets d'intérêt, sans action manuelle préalable. Pour cela, nous avons initié la création d'un jeu de données annoté manuellement, première étape indispensable pour entraîner un modèle supervisé.

L'annotation a été réalisée à l'aide de la plate-forme libre **makesense.ai**, qui permet d'importer des images, d'y dessiner des polygones ou des boîtes englobantes, et de leur associer des étiquettes. Nous avons choisi une annotation précise par polygones, car elle permet d'épouser fidèlement les contours des objets, ce qui est particulièrement important pour la qualité de l'apprentissage supervisé en segmentation.

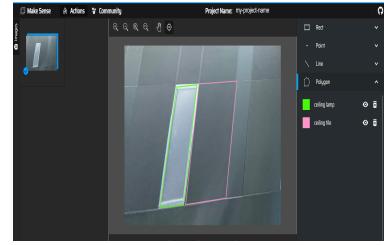


FIGURE 15 – Annotation réalisée sur des dalles de plafond à l'aide de makesense.ai. Chaque zone a été annotée afin d'identifier visuellement les composants structuraux.

Toutefois, en raison du volume limité d'exemples disponibles à ce stade et du coût élevé associé à l'annotation pixel-par-pixel, il n'est pas envisageable d'entraîner un modèle performant depuis zéro. Cela justifie le recours à une approche par **fine-tuning**, qui permet de capitaliser sur les représentations déjà apprises par un modèle pré-entraîné, tout en l'adaptant à la spécificité de notre jeu de données et de nos classes d'objets. Cela consiste à réentraîner partiellement un modèle pré-entraîné sur une nouvelle tâche ou un nouveau domaine, tout en conservant l'essentiel de ses connaissances acquises lors d'un entraînement initial. Concrètement, on reprend les poids du modèle initial et on les ajuste légèrement à l'aide d'un jeu de données spécifique au nouveau problème. Cette méthode permet de réduire considérablement le besoin en données annotées, de bénéficier d'une convergence plus rapide lors de l'entraînement et d'améliorer les performances par rapport à un entraînement "from scratch", surtout si le jeu de données est de taille limitée.

7 Conclusion

Ce travail a démontré la faisabilité et la pertinence d'une approche intégrée combinant reconstruction 3D, estimation géométrique robuste et segmentation pour l'analyse automatique des plafonds dans le cadre de projets de réemploi des objets. En mobilisant des outils avancés en vision par ordinateur, nous avons conçu une pipeline capable de générer, à partir d'images prises sur le terrain, une vue redressée du plafond et une segmentation des composants structuraux. L'approche proposée repose sur une modélisation rigoureuse du plan du plafond, obtenue à partir d'un nuage de points dense, et sur la projection géométriquement cohérente des images vers une vue orthorectifiée. Cette étape est cruciale pour garantir une segmentation fiable malgré la complexité visuelle des scènes.

Les résultats expérimentaux obtenus dans un environnement réel confirment la robustesse du pipeline et la qualité des résultats, en particulier pour la détection d'objets répétitifs tels que les dalles, les luminaires ou les grilles techniques. À terme, ce travail ouvre des perspectives concrètes pour l'automatisation du diagnostic de réemploi dans les bâtiments, dans une logique de circularité des matériaux ou des objets. Il préfigure également des applications plus larges en gestion patrimoniale et maintenance prédictive.

8 Annexes

A Présentation de MAST3R

mast3r (*Matching and Stereo 3D Reconstruction*) est une méthode récente de reconstruction 3D métrique, introduite en 2024 par Wang et al., qui repose sur une extension du framework **dust3r**. Elle a pour objectif de produire, à partir d'un ensemble non calibré d'images RGB, un nuage de points dense accompagné de correspondances locales précises entre pixels, sans nécessiter de poses de caméras initiales.

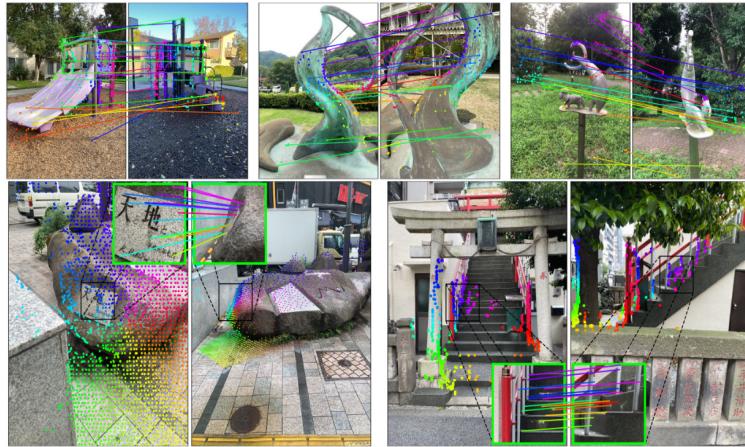


FIGURE 16 – Caption

Contrairement aux pipelines classiques de Structure-from-Motion (SfM) et Multi-View Stereo (MVS), mast3r adopte une approche **directement end-to-end**, sans passer par une géométrie explicite ou des appariements intermédiaires de features clés. Le modèle intègre dans son architecture un mécanisme d'**attention inter-vues** pour agréger efficacement l'information visuelle et géométrique.

L'architecture de mast3r peut être schématisée en trois modules principaux :

- **Architecture de MASt3R**

L'architecture de MASt3R repose sur un double pipeline symétrique, où chaque image d'entrée passe par le même encodeur visuel basé sur un ViT (Vision Transformer), partageant les poids entre les deux branches. Voici les principales étapes du traitement :

1. **Encodage ViT** : Chaque image est encodée par un ViT encoder pour produire une représentation spatiale compacte H^1, H^2 des deux vues, contenant des informations visuelles et structurelles à différents niveaux de résolution.
2. **Attention croisée dans le Transformer Decoder** : les représentations H^1 et H^2 sont injectées dans un décodeur Transformer avec attention croisée entre les deux vues, permettant de faire émerger des correspondances visuelles cohérentes dans l'espace image. Ce module favorise une mise en correspondance fine entre les deux images, sans passer par un mécanisme explicite de coût stéréo.
3. **Heads spécialisées** :
 - head **3D** (Head_{3D}) produit une *pointmap* $X^{i,j}$, qui donne pour chaque pixel son point correspondant dans l'espace 3D ;
 - head de **confiance** $C^{i,j}$ prédit une probabilité d'appariement pour chaque point ;
 - head **descripteur** (Head_{desc}) fournit une carte de caractéristiques locales $F^{i,j}$ (features denses de taille $H \times W \times d$).
4. **Appariement final** : les cartes de descripteurs sont comparées par un algorithme de plus proche voisin rapide (Fast NN), qui permet :

- un **matching géométrique** via les pointmaps (3D) ;
- un **matching fondé sur les descripteurs** $F^{i,j}$.

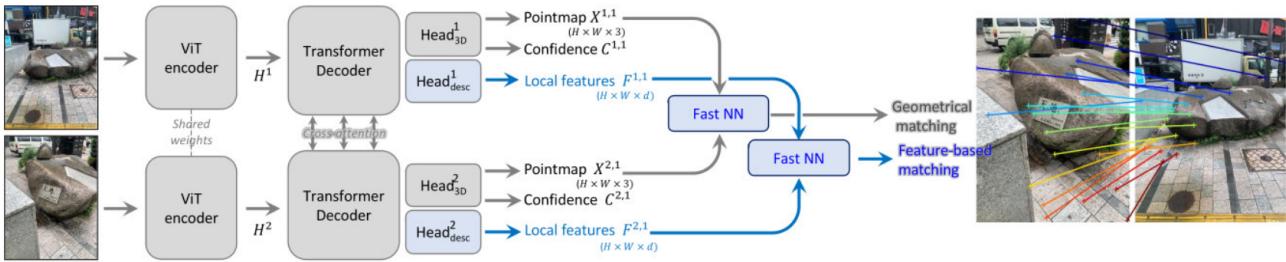


FIGURE 17 – Architecture de MASt3R : encodeur ViT partagé, attention croisée, prédiction de pointmaps 3D et descripteurs locaux.

Le modèle est entraîné sur des tâches de correspondance et de reconstruction 3D simultanément, avec des pertes combinant :

- une **perte photométrique** entre les pixels reprojetés ;
- une **perte géométrique** sur la précision des profondeurs ;
- une **perte de correspondance dense** (matching supervision).

*Code disponible dans le répertoire du labo naver : <https://github.com/naver/mast3r> .

B Méthode RANSAC

L'algorithme RANSAC (pour *Random Sample Consensus*) est une méthode itérative développée par Fischler et Bolles en 1981, visant à estimer les paramètres d'un modèle géométrique à partir de données massivement bruitées ou corrompues. Contrairement aux méthodes classiques d'ajustement (telles que les moindres carrés), RANSAC est robuste aux outliers, c'est-à-dire aux points ne correspondant pas au modèle recherché.

Le principe est le suivant : à chaque itération, un petit échantillon de points est sélectionné aléatoirement pour estimer un modèle (par exemple une droite, un plan ou une homographie). Ce modèle est ensuite évalué en mesurant combien de points du jeu de données total sont compatibles avec lui — on les appelle les *inliers*. Après un nombre fixe d'itérations, le modèle générant le plus grand nombre d'inliers est conservé comme solution optimale.

RANSAC est particulièrement utilisé en vision par ordinateur et en traitement 3D pour détecter des structures géométriques simples dans des données complexes, comme des plans dans un nuage de points, des lignes dans une image ou des correspondances d'homographies entre vues.

Sa robustesse en fait une méthode de référence pour le pré-traitement de données avant des étapes de segmentation ou d'optimisation plus fines.

L'algorithme générique de RANSAC est le suivant :

Algorithm 1 RANSAC (Random Sample Consensus)

Require: `data` ▷ Ensemble d'observations

`modele` ▷ Modèle ajustable aux données

`n` ▷ Nombre minimal de points nécessaires pour ajuster le modèle

`k` ▷ Nombre maximal d'itérations

`t` ▷ Seuil pour déterminer si un point est un inlier

`d` ▷ Nombre minimal d'inliers pour valider un modèle

Ensure: `meilleur_modèle, meilleur_ensemble_points, meilleure_erreur`

```

meilleur_modèle ← aucun
meilleur_ensemble_points ← aucun
meilleure_erreur ← ∞
for i = 1 to k do
    points_aléatoires ← n points choisis aléatoirement dans data
    modèle_possible ← ajustement du modèle sur points_aléatoires
    ensemble_points ← points_aléatoires
    for all point p dans data non inclus dans points_aléatoires do
        if l'erreur de p par rapport à modèle_possible < t then
            ajouter p à ensemble_points
        end if
    end for
    if taille de ensemble_points > d then
        modèle_possible ← réajustement sur ensemble_points
        erreur ← erreur globale du modèle sur ensemble_points
        if erreur < meilleure_erreur then
            meilleur_modèle ← modèle_possible
            meilleur_ensemble_points ← ensemble_points
            meilleure_erreur ← erreur
        end if
    end if
end for
return meilleur_modèle, meilleur_ensemble_points, meilleure_erreur

```

C Homographie

Une **homographie** est une transformation projective entre deux plans. Elle est représentée par une matrice $H \in \mathbb{R}^{3 \times 3}$, définie à un facteur d'échelle près. Cette transformation agit sur des points en coordonnées homogènes :

$$x' = Hx$$

où x et x' sont des vecteurs homogènes représentant respectivement un point dans l'image source et dans l'image cible.

Formulation du problème

Chaque correspondance de points $x_i \leftrightarrow x'_i$ fournit **deux équations linéaires indépendantes** sur les coefficients de H . Étant donné que H comporte 9 coefficients mais est défini à un facteur d'échelle près, il possède **8 degrés de liberté**. Par conséquent, **au moins 4 correspondances** sont nécessaires pour estimer H .

L'équation fondamentale s'écrit :

$$\lambda \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Cette relation peut être linéarisée en un système homogène.

Mise sous forme matricielle

Soit h le vecteur colonne des 9 éléments de la matrice H , concaténés ligne par ligne. Pour chaque correspondance $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$, on peut écrire :

$$A_i h = 0$$

où $A_i \in \mathbb{R}^{2 \times 9}$ est une matrice construite à partir des coordonnées du point x_i . En général :

$$A_i = \begin{bmatrix} 0 & 0 & 0 & -x_i & -y_i & -1 & y'_i x_i & y'_i y_i & y'_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix}$$

En empilant les A_i pour $i = 1, \dots, n$, on obtient une matrice $A \in \mathbb{R}^{2n \times 9}$, et on cherche à résoudre :

$$Ah = 0$$

Résolution par la méthode DLT (Direct Linear Transformation)

Lorsque le système est exact (cas sans bruit avec 4 correspondances), il admet une solution non triviale. En pratique, pour des données bruitées (plus de 4 correspondances), on cherche une solution au sens des moindres carrés sous contrainte :

$$\min \|Ah\| \quad \text{sous la contrainte} \quad \|h\| = 1$$

Cette contrainte évite la solution triviale $h = 0$, et permet de sélectionner un vecteur unitaire.

Décomposition en valeurs singulières (SVD)

La **décomposition en valeurs singulières** (SVD) permet d'écrire la matrice A sous la forme :

$$A = U \Sigma V^T$$

où :

- $U \in \mathbb{R}^{2n \times 2n}$ et $V \in \mathbb{R}^{9 \times 9}$ sont des matrices orthogonales,
- $\Sigma \in \mathbb{R}^{2n \times 9}$ est une matrice diagonale contenant les **valeurs singulières** de A , triées par ordre décroissant.

La solution h correspond au **vecteur propre associé à la plus petite valeur singulière**, c'est-à-dire la dernière colonne de V .

Algorithme DLT

Algorithm 2 Estimation de l'homographie par la méthode DLT

Require: Ensemble de correspondances $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$

Ensure: Matrice d'homographie $H \in \mathbb{R}^{3 \times 3}$

```

1: for chaque correspondance  $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$  do
2:     Construire la matrice  $A_i$ 
3: end for
4: Empiler les  $A_i$  pour obtenir  $A \in \mathbb{R}^{2n \times 9}$ 
5: Calculer la décomposition SVD de  $A$  :  $A = U\Sigma V^T$ 
6: Extraire  $h$  comme la dernière colonne de  $V$ 
7: Réorganiser  $h$  en une matrice  $H \in \mathbb{R}^{3 \times 3}$ 
return  $H$ 

```

D Segment Anything Model (SAM)

Le modèle Segment Anything (SAM) a été proposé par Meta AI dans le but de concevoir un système de segmentation d'images capable de s'adapter à une grande variété de tâches sans entraînement spécifique. Contrairement aux approches classiques centrées sur des classes d'objets fixes ou des scénarios bien définis, SAM repose sur une approche dite promptable : l'utilisateur guide le modèle en lui fournissant un indice (un point, une boîte, un masque ou une description textuelle) indiquant ce qu'il souhaite segmenter. Ce paradigme permet à SAM de généraliser efficacement à de nouveaux contextes, tout en offrant des performances en temps quasi réel.

Architecture du modèle

Comme le montre la Figure 18, l'architecture de SAM repose sur trois composantes principales :

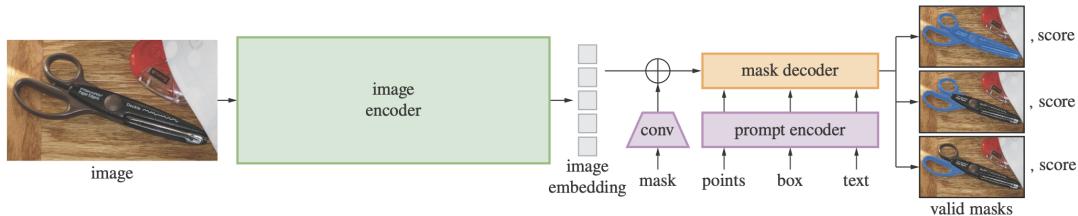


FIGURE 18 – Segment Anything Model (SAM) : vue générale.

- Un encodeur d'image : basé sur un ViT (Vision Transformer) pré-entraîné via MAE, il extrait une *embedding* riche et unique de l'image. Cet encodeur est coûteux, mais il est appelé une seule fois par image, permettant une réutilisation amortie dans des scénarios interactifs.
- Un encodeur de prompts : selon le type de prompt (points, boîtes, masques ou texte), différentes techniques d'encodage sont utilisées :
 - Points et boîtes : encodés via des embeddings positionnels.
 - Texte : encodé via un encodeur CLIP.
 - Masques : traités par convolution.

Ces embeddings sont ensuite intégrés à ceux de l'image.

- Un décodeur léger de masques : inspiré d'architectures Transformers modifiées, il combine les embeddings de l'image et des prompts pour prédire un ou plusieurs masques. Ce décodeur fonctionne en temps réel (environ 50 ms dans un navigateur), ce qui rend le modèle utilisable de manière interactive.

Ce modèle représente une avancée majeure dans le domaine de la segmentation d'images. L'objectif est de créer un modèle fondation pour la segmentation, à l'instar de ce que CLIP ou GPT représentent pour le langage naturel.

SAM se distingue par sa capacité à produire des masques d'objets à partir de requêtes appelées prompts (sous forme de points, de boîtes, de masques ou même de texte), avec une grande généralisation en *zero-shot* sur des tâches et distributions d'images jamais vues.

Grâce à son architecture, SAM atteint une excellente performance en *zero-shot* sur plus de 23 jeux de données variés, rivalisant voire surpassant des modèles spécialisés entraînés de manière supervisée. Il peut être utilisé pour :

- la génération de propositions d'objets,
- la segmentation d'instances ou de catégories,
- l'annotation automatique de jeux de données,
- la segmentation à partir de texte libre (dans une certaine mesure).

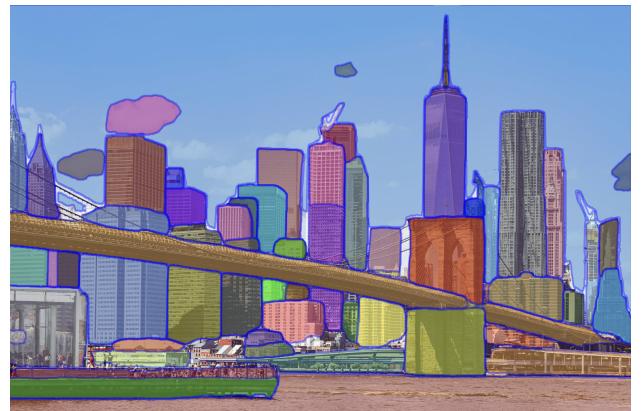


FIGURE 19 – Exemple de segmentation par SAM à partir d'un point. À gauche : image d'entrée avec prompt. À droite : masque généré.

En complément de la Figure 18, les images de la Figure 19 illustrent un exemple typique d'application de SAM : à partir d'un simple point d'intérêt fourni sur une image, le modèle génère automatiquement un masque précis correspondant à l'objet visé. Ce processus ne nécessite aucun entraînement spécifique sur cette image ou sur une classe cible, démontrant ainsi la puissance du paradigme promptable de SAM.

9 Bibliographie

1. <https://arxiv.org/abs/2406.09756>
2. <https://colmap.github.io/>
3. <https://github.com/naver/mast3r>
4. <https://www.sciencedirect.com/science/article/pii/S2772941925000584>
5. https://perso.ensta-paris.fr/~manzaner/Cours/R0B313/R0B313_Vision3D01_2019.pdf
6. https://perso.ensta-paris.fr/~manzaner/Cours/R0B317/TP1_Homographie.pdf
7. <https://segment-anything.com/>