

THÈSE PRÉSENTÉE  
POUR OBTENIR LE GRADE DE  
**DOCTEUR**  
**DE L'UNIVERSITÉ DE BORDEAUX**

ECOLE DOCTORALE SCIENCES PHYSIQUES ET DE L'INGENIEUR

Par **Matthieu VILAIN**

***Mécanisme d'attention en apprentissage profond  
pour la mise en correspondance d'images***

Sous la direction de : **Yannick BERTHOUMIEU**

Encadrant : **Guillaume BOURMAUD**

Encadrant : **Rémi GIRAUD**

Soutenue le 16 décembre 2024

Membres du jury :

M. Hazem WANNOUS	Professeur	Institut Mines-Télécom Lille	Rapporteur
M. Renaud PETERI	Maître de conférences, HDR	Université de La Rochelle	Rapporteur
M. Christian GERMAIN	Professeur	Université de Bordeaux	Président
M. Yannick BERTHOUMIEU	Professeur	Université de Bordeaux	Directeur
M. Guillaume BOURMAUD	Maître de conférences	Université de Bordeaux	Invité
M. Rémi GIRAUD	Maître de conférences	Université de Bordeaux	Invité



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Problématique et objectif de la thèse . . . . .	10
1.2	Applications de la mise en correspondance d'images . . . . .	12
1.3	Organisation du manuscrit . . . . .	15
<b>2</b>	<b>Mise en correspondance d'images et mécanisme d'attention</b>	<b>17</b>
2.1	Introduction . . . . .	19
2.2	Mise en correspondance d'images . . . . .	20
2.2.1	Introduction . . . . .	20
2.2.2	Paradigmes de mise en correspondance . . . . .	25
2.2.3	Discussion . . . . .	31
2.3	Avènement de l'attention dans la vision par ordinateur . . . . .	32
2.3.1	Le mécanisme d'attention . . . . .	33
2.3.2	Limitations et variantes . . . . .	36
2.3.3	L'architecture Transformer . . . . .	38
2.3.4	L'attention en vision par ordinateur . . . . .	39
2.4	Relation entre l'attention et la mise en correspondance . . . . .	41
2.4.1	L'attention comme opération de communication entre les descripteurs . . . . .	41
2.4.2	Analyse de l'attention pour la mise en correspondance . . . . .	43
2.5	Conclusion . . . . .	51
<b>3</b>	<b>De la mise en correspondance de points d'intérêt aux méthodes semi-denses</b>	<b>53</b>
3.1	Introduction . . . . .	55
3.2	État de l'art . . . . .	56
3.2.1	Le paradigme épars (S2S) : SuperGlue . . . . .	56
3.2.2	Passage au semi-dense sans détecteur (SDF) : LoFTR . . . . .	58
3.2.3	Changement de paradigme, besoin d'évaluation équitable . . . . .	59
3.3	Notre approche . . . . .	60
3.3.1	Matching sur demande . . . . .	60
3.3.2	Architecture SAM . . . . .	62
3.3.3	Attention structurée . . . . .	64
3.3.4	Entraînement . . . . .	67
3.4	Expériences . . . . .	67
3.4.1	Estimation de pose de caméra, homographie et séquence d'images . . . . .	68
3.4.2	Étude d'ablation . . . . .	71
3.4.3	Étude de l'attention structurée . . . . .	71
3.4.4	Évolution de SAM . . . . .	72
3.5	Conclusion . . . . .	78

<b>4</b>	<b>Vers une mise en correspondance d'images dense</b>	<b>79</b>
4.1	Introduction . . . . .	81
4.2	État de l'art . . . . .	82
4.2.1	L'approche régressive de DKM . . . . .	82
4.2.2	Classification en cascade avec CasMTR . . . . .	84
4.2.3	Discussion . . . . .	85
4.3	Matching dense par recherche en faisceaux . . . . .	86
4.3.1	Problème de multimodalité . . . . .	87
4.3.2	Prédiction de la covisibilité . . . . .	93
4.3.3	Notre architecture BEAMER . . . . .	97
4.3.4	Étape d'entraînement . . . . .	99
4.4	Expériences . . . . .	102
4.4.1	Analyse de l'architecture . . . . .	102
4.4.2	Estimation de pose de caméra et précision de correspondance . . . . .	108
4.5	Conclusion . . . . .	118
<b>5</b>	<b>Conclusion</b>	<b>121</b>
5.1	Contributions et discussion . . . . .	122
5.2	Perspectives de travaux futurs . . . . .	123

## Mécanisme d'attention en apprentissage profond pour la mise en correspondance d'images

**Résumé :** La mise en correspondance d'images est un problème fondamental de vision par ordinateur visant à établir des correspondances 2D entre deux images présentant un recouvrement partiel. Ce problème peut être particulièrement complexe en raison des changements de perspective, des variations de luminosité, ou encore d'occultations. Récemment, ces difficultés ont été en partie surmontées grâce aux réseaux de neurones profonds utilisant un mécanisme d'attention ("Transformer").

Dans un premier temps, nous présentons le contexte général de la mise en correspondance (métriques d'évaluation, bases de données, paradigmes, etc.), ainsi que le mécanisme d'attention. Nous montrons que l'introduction d'un tel mécanisme est particulièrement adaptée pour modéliser les relations complexes entre deux images, permettant ainsi de pallier les limites des réseaux siamois.

Nous nous intéressons ensuite aux approches dites semi-denses. Ces méthodes de l'état de l'art utilisent un réseau avec mécanisme d'attention mais leurs performances sont presque exclusivement évaluées au travers de métriques d'estimation de pose relative entre les deux images. Ainsi, dans cette deuxième partie nous cherchons à comprendre le lien entre la capacité de ces méthodes à établir des correspondances et la qualité de la pose estimée.

Finalement, nous abordons le problème de la mise en correspondance dense, où l'objectif est d'établir une correspondance pour chaque pixel des images. L'utilisation d'un mécanisme d'attention au niveau pixelique est un défi au vu de sa complexité calculatoire. Nous proposons de l'inclure dans une méthode hiérarchique de recherche en faisceau permettant ainsi au réseau de bénéficier d'un mécanisme d'attention pixelique et d'une complexité calculatoire raisonnable.

**Mots-clés :** Mise en correspondance, Mécanisme d'attention, Apprentissage profond, Vision par ordinateur

---

### Attention mechanism in deep learning for image matching

**Abstract:** Image matching is a fundamental problem in computer vision that aims to establish 2D correspondences between two images with partial overlap. This problem can become particularly challenging due to perspective changes, variations in lighting, or occlusions. Recently, these difficulties have been partially overcome by deep neural networks using attention mechanisms (Transformers).

First, we present the general context of image matching, discussing evaluation metrics, available datasets, and different paradigms, while introducing the attention mechanism. We show that integrating this mechanism is particularly well-suited for modeling complex relationships between two images, thus overcoming some of the limitations of Siamese networks.

We then focus on semi-dense approaches. These state-of-the-art methods, which also use networks with attention mechanisms, are mostly evaluated based on relative pose estimation metrics between the images. In this section, we explore the link between these methods' ability to establish correspondences and the accuracy of the estimated pose.

Finally, we address the problem of dense matching, where the goal is to establish correspondences for every image pixel. Using an attention mechanism at the pixel level is a significant challenge due to its computational complexity. We propose to incorporate this mechanism into a hierarchical beam search method, allowing the network to benefit from pixel-wise attention while maintaining reasonable computational complexity.

**Keywords:** Matching, Attention mechanism, Deep learning, Computer vision

---



# Liste des acronymes

<b>AA</b>	Auto-Attention ( <i>self-attention</i> )
<b>AC</b>	Attention-Croisée ( <i>cross-attention</i> )
<b>AGNN</b>	Réseau de neurones attentionnel en graphe ( <i>Attentional Graph Neural Network</i> )
<b>AR</b>	Réalité augmentée ( <i>Augmented Reality</i> )
<b>AUC</b>	Aire sous la courbe ( <i>Area Under the Curve</i> )
<b>CE</b>	Entropie croisée ( <i>Cross Entropy</i> )
<b>CNN</b>	Réseau de neurones convolutif ( <i>Convolutional Neural Network</i> )
<b>DIM</b>	Mise en correspondance dense ( <i>Dense Image Matching</i> )
<b>ELU</b>	Fonction unité linéaire exponentielle ( <i>Exponential Linear Unit</i> )
<b>FPN</b>	Réseau de neurones convolutif produisant des cartes de caractéristiques multi-échelles ( <i>Feature Pyramid Network</i> )
<b>GPU</b>	Unité de traitement graphique ( <i>Graphics Processing Unit</i> )
<b>GT</b>	Vérité terrain ( <i>Ground Truth</i> )
<b>k-NN</b>	Méthode des k plus proches voisins ( <i>k-Nearest Neighbors</i> )
<b>LV</b>	Vecteur latent ( <i>Latent Vector</i> )
<b>MA</b>	Précision de mise en correspondance ( <i>Matching Accuracy</i> )
<b>MAD</b>	Distance moyenne d'attention ( <i>Mean Attention Distance</i> )
<b>MCD</b>	Distance moyenne au correspondant ( <i>Mean Correspondance Distance</i> )
<b>MLP</b>	Perceptron multi-couches ( <i>Multi Layer Perceptron</i> )
<b>MNN</b>	Méthode des plus proches voisins mutuels ( <i>Mutual Nearest Neighbour</i> )
<b>NLP</b>	Traitement du langage naturel ( <i>Natural Language Processing</i> )
<b>PE</b>	Encodage positionnel ( <i>Positional Encoding</i> )
<b>ReLU</b>	Fonction unité linéaire rectifiée ( <i>Rectified Linear Unit</i> )
<b>RNN</b>	Réseau de neurones récurrent ( <i>Recurrent Neural Network</i> )
<b>S2S</b>	Éparse-à-Éparse ( <i>Sparse-to-Sparse</i> )
<b>SDF</b>	Semi-dense sans détecteur ( <i>Semi-dense Detector Free</i> )
<b>SfM</b>	Structure acquise à partir d'un mouvement ( <i>Structure from Motion</i> )
<b>SLAM</b>	Localisation et cartographie simultanées ( <i>Simultaneous Localization And Mapping</i> )





# **Chapitre 1**

## **Introduction**

## 1.1 Problématique et objectif de la thèse

Cette thèse a pour objectif de développer des approches fondées sur l'apprentissage automatique pour la mise en correspondance d'images, en mettant un accent particulier sur l'intégration du mécanisme d'attention afin de renforcer la robustesse et la précision des solutions proposées.

La mise en correspondance d'images, ou *image matching*, est un concept fondamental en vision par ordinateur. Elle consiste à identifier des correspondances entre deux images représentant une même scène 3D (Figure 1.1). L'établissement d'un ensemble de paires de positions 2D entre des caractéristiques similaires dans différentes images sert de base à de nombreuses tâches de vision par ordinateur, notamment la localisation visuelle et l'estimation de pose de caméra, l'acquisition de structure à partir du mouvement (*structure from motion* ou SfM), la cartographie et localisation simultanées (*simultaneous localization and mapping* ou SLAM), l'estimation du flux optique, la récupération d'images, la fusion d'images, et bien plus encore.

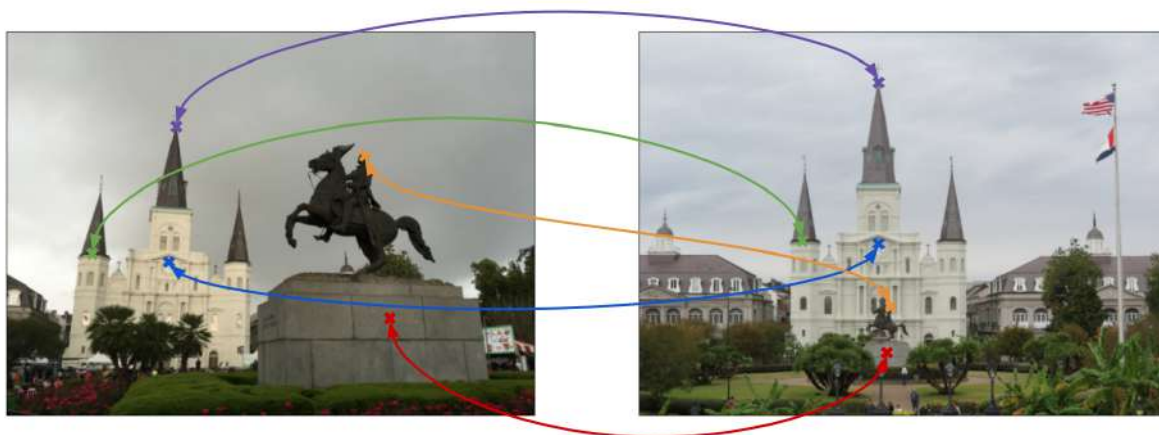


FIGURE 1.1 – **Mise en correspondance d'images.** Étant donné une paire d'images partiellement covisibles, l'objectif de la mise en correspondance d'images est de trouver un ensemble de correspondances 2D à 2D entre les deux images.

La mise en correspondance est un problème particulièrement difficile en raison de la grande variété de perturbations visuelles qui peuvent survenir lors de la capture d'images en conditions réelles. Deux images capturant une même scène peuvent être prises sous des points de vue totalement différents, introduisant des changements d'échelle (Figure 1.2b) ou de perspective (Figure 1.2c), nécessitant ainsi le développement de méthodes capables d'inférer une certaine compréhension de la structure 3D de la scène. Ces différents points de vue peuvent également entraîner un faible recouvrement visuel entre les images, et la nature 3D de la scène observée introduit naturellement des occultations (Figure 1.2g) qui peuvent partiellement ou entièrement masquer une partie de la scène. En conséquence, la résolution du problème de mise en correspondance est intrinsèquement limitée par les informations disponibles dans la paire d'images. Les réglages du dispositif de capture, tels que la focale (Figure 1.2d), la mise au point (Figure 1.2f) ou la colorimétrie (Figure 1.2i), peuvent également altérer l'apparence des objets de la scène, rendant difficile la mise en correspondance. La nature dynamique du monde introduit aussi fréquemment des perturbations entre les deux images. L'heure du jour ou de la nuit, la saison, la météo ou les conditions de luminosité (Figure 1.2e) altèrent également l'apparence

des objets de la scène. Enfin, l'objet observé par nos images peut contenir des structures répétitives locales (Figure 1.2h), où des régions apparaissent comme similaires sans correspondre pour autant. Ce problème typique de la mise en correspondance nécessite le développement de méthodes capables de lever l'ambiguïté entre ces motifs répétitifs en prenant en compte l'information globale des images pour établir des correspondances correctes. En pratique, une paire d'images naturelles contient fréquemment une combinaison de plusieurs des perturbations citées précédemment. Ces défis nécessitent une compréhension approfondie de la structure 3D sous-jacente de la scène, et la résolution de ces difficultés reste un problème ouvert dans la communauté de la vision par ordinateur.



(a) Image de référence



(b) Changement d'échelle



(c) Changement de perspective



(d) Changement de focale



(e) Changement d'illumination



(f) Changement de focus



(g) Occultation



(h) Structures répétitives



(i) Changement colorimétrique

FIGURE 1.2 – **Visualisation des différentes perturbations pouvant compliquer la mise en correspondance d'images.** La grande diversité des perturbations qui peuvent survenir au sein d'une paire d'images nécessite le développement de méthodes invariantes, discriminatives et capables d'inférer une certaine compréhension de la structure 3D de la scène afin de trouver des correspondances.

Cette thèse se concentrera sur les techniques d'apprentissage automatique pour la mise en correspondance d'images, qui se sont imposées comme des solutions idéales face aux défis inhérents à cette tâche. L'apprentissage automatique, en particulier l'apprentissage profond et les réseaux de neurones convolutifs, offre une capacité à généraliser et à apprendre des représentations robustes face aux variations visuelles complexes telles que les changements d'échelle, de perspective, d'illumination et les occultations. Ces techniques permettent aux modèles d'extraire des caractéristiques riches et discriminantes, essentielles pour établir des correspondances fiables, même lorsque les images présentent des perturbations significatives.

Notre intérêt se portera particulièrement sur le mécanisme d'attention [Vaswani et al., 2017], initialement introduit dans le domaine du traitement automatique du langage naturel (NLP) et qui a révolutionné la manière de traiter les séquences de mots en permettant aux architectures dites *Transformer* de construire des relations complexes entre les éléments d'une séquence. En vision par ordinateur, l'attention a gagné en popularité grâce à sa capacité à permettre aux modèles d'apprentissage profond de se concentrer sur des régions spécifiques des images. Cependant, le coût calculatoire quadratique de l'opération d'attention pose un défi particulier lorsqu'il s'agit de traiter des images. En effet, le mécanisme d'attention nécessite de calculer une matrice d'attention qui capture les relations entre tous les pixels d'une image, ce qui entraîne une complexité en temps et en mémoire proportionnelle au carré du nombre de pixels. Cela devient rapidement prohibitif, surtout pour des images de haute résolution, et il est souvent nécessaire de découper l'image en plusieurs patches pour réduire la complexité de l'opération. Néanmoins, ces dernières années, la communauté de mise en correspondance d'images a porté un intérêt particulier au mécanisme d'attention. Sa capacité à créer de la communication intra et inter-images permet de générer des représentations encore plus discriminatives, englobant le contexte global des deux images, et permet de lever les ambiguïtés des structures répétitives ou des autres perturbations mentionnées précédemment. Toutefois, l'intégration du mécanisme d'attention reste confrontée au défi de son coût calculatoire élevé. Contrairement à des tâches comme la reconnaissance d'objets, où le découpage en patches permet de réduire le coût de l'attention, la mise en correspondance d'images nécessite une grande précision pour trouver des correspondances au niveau pixelique. Le défi de cette thèse reposera donc sur le développement de méthodes incorporant le mécanisme d'attention pour trouver des correspondances précises pour la mise en correspondance d'images.

## 1.2 Applications de la mise en correspondance d'images

Dans cette thèse, nous ne nous limitons pas à un cadre applicatif spécifique, mais nous nous intéressons à la problématique générale de la mise en correspondance d'images et à son utilisation pour l'estimation de pose de caméra, qui sera abordée en détail dans le chapitre 2. Toutefois, la mise en correspondance d'images trouve des applications dans un large éventail de domaines concrets, apportant des solutions à des problématiques variées, telles que la reconstruction de structures 3D, l'intégration d'objets virtuels en réalité augmentée, la navigation précise des robots autonomes, la recherche d'images similaires, et la fusion d'images pour améliorer la qualité visuelle. Ces applications démontrent l'importance de la mise en correspondance d'images en tant qu'outil polyvalent et indispensable dans divers secteurs, comme illustré dans la Figure 1.3.

**Reconstruction 3D.** (Figure 1.3c) La mise en correspondance d'images joue un rôle fondamental dans la reconstruction 3D, qui consiste à reconstruire la structure tridimensionnelle d'une scène à partir de multiples images prises sous des angles différents. En établissant des correspondances précises entre les images, il est possible de déterminer les positions spatiales

des points-clés dans l'espace 3D, permettant ainsi de reconstruire des modèles géométriques détaillés de la scène. Cette approche est au cœur de l'acquisition de structure à partir du mouvement (Structure from Motion, ou SfM), qui permet de générer des reconstructions 3D à partir d'images non calibrées [Özyeşil et al., 2017]. SfM combine la détection des points d'intérêt, l'appariement entre les images, et l'estimation des paramètres de caméra pour estimer la géométrie de la scène. Cette technique est largement utilisée dans des domaines tels que l'architecture [Zhou et al., 2012], la modélisation d'environnements [Iglhaut et al., 2019], l'archéologie numérique [Anderson, 2020], et le cinéma [Liu et al., 2022]. Récemment, des algorithmes de SfM ont été utilisés après l'incendie de Notre-Dame de Paris pour reconstruire un modèle 3D précis de l'intérieur de la cathédrale.

**Réalité augmentée.** (Figure 1.3a) La réalité augmentée (AR) repose fortement sur la mise en correspondance d'images pour intégrer des objets virtuels dans un environnement réel. Pour garantir une expérience immersive, il est crucial de localiser et suivre précisément les caractéristiques de l'environnement en temps réel. En utilisant la mise en correspondance d'images, les solutions d'AR peuvent détecter les surfaces et les points de repère nécessaires pour placer de manière cohérente des objets virtuels [Song and Li, 2021], ce qui permet une interaction fluide entre les éléments virtuels et le monde réel.

**Robotique et navigation autonome.** (Figure 1.3b) Dans le domaine de la robotique et de la navigation autonome, la mise en correspondance d'images est essentielle pour permettre aux robots et véhicules de comprendre leur environnement [Yadav and Singh, 2016]. En comparant les images capturées en temps réel avec des images de référence, les systèmes peuvent estimer leur position, détecter des obstacles et planifier des trajets sûrs. La mise en correspondance d'images est au cœur des systèmes de cartographie et localisation simultanées (SLAM) [Placed et al., 2023], qui permettent aux robots d'explorer et de naviguer dans des environnements inconnus.

**Récupération d'images.** (Figure 1.3d) La récupération d'images consiste à retrouver, parmi une base de données, des images similaires à une image donnée en entrée. La mise en correspondance d'images permet de comparer efficacement les caractéristiques de l'image requête avec celles de la base de données pour identifier les correspondances les plus proches [Wang et al., 2021]. Cette technique est utilisée dans des applications telles que la recherche d'images par contenu, l'identification de doublons, ou la détection de plagiat visuel. Par exemple, en imagerie médicale, la récupération d'images permet de retrouver des cas similaires à partir d'une image de référence [Rahman et al., 2007], facilitant ainsi le diagnostic médical et la prise de décision clinique.

**Fusion d'images.** (Figure 1.3e) La fusion d'images consiste à combiner plusieurs images d'une même scène pour en créer une représentation plus complète ou de meilleure qualité. La mise en correspondance d'images est une étape préalable essentielle pour aligner les images de manière précise, en identifiant les points correspondants [Solsona et al., 2017]. Une fois alignées, les images peuvent être fusionnées pour améliorer la résolution, augmenter la gamme dynamique, ou réduire le bruit. Cela est particulièrement utile en imagerie médicale [James and Dasarathy, 2014], en astrophotographie [Sun, 2014], et dans d'autres contextes où la qualité de l'image est cruciale.





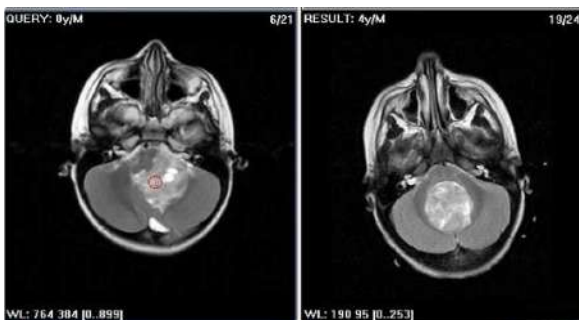
(a) Réalité augmentée (source : [Apple, 2024])



(b) Conduite autonome (source : [Waymo, 2019])



(c) Reconstruction 3D [Schönberger and Frahm, 2016]



(d) Récupération d'images [Huang et al., 2005]



(e) Fusion d'images [Jacquet, 2014]

**FIGURE 1.3 – Exemple d'applications pratiques de la mise en correspondance d'images.** Ces applications démontrent l'importance de la mise en correspondance d'images en tant qu'outil polyvalent dans divers secteurs nécessitant une compréhension d'une scène 3D et/ou la création de liens entre plusieurs images.

## 1.3 Organisation du manuscrit

Dans le Chapitre 2, nous présentons le contexte général de la mise en correspondance (métriques d'évaluation, bases de données, paradigmes, etc.), ainsi que le mécanisme d'attention. Nous montrons que l'introduction d'un tel mécanisme est particulièrement adaptée pour modéliser les relations complexes entre deux images, permettant ainsi de pallier les limites des réseaux convolutifs siamois.

Dans le Chapitre 3, nous nous intéressons aux approches dites semi-denses. Ces méthodes de l'état de l'art utilisent un réseau avec mécanisme d'attention, mais leurs performances sont presque exclusivement évaluées au travers de métriques d'estimation de pose relative entre les deux images. Ainsi, dans ce deuxième chapitre, nous cherchons à comprendre le lien entre la capacité de ces méthodes à établir des correspondances et la qualité de la pose estimée.

Finalement, dans le Chapitre 4, nous abordons le problème de la mise en correspondance dense, où l'objectif est d'établir une correspondance pour chaque pixel non occulté. L'utilisation d'un mécanisme d'attention au niveau pixelique est un défi au vu de sa complexité calculatoire. Nous proposons de l'inclure dans une méthode hiérarchique de recherche en faisceaux, permettant ainsi au réseau de bénéficier d'un mécanisme d'attention pixelique et d'une complexité calculatoire raisonnable.

### Contributions :

- Dans le Chapitre 2, nous proposons une analyse du mécanisme d'attention dans le cadre spécifique de la mise en correspondance d'images.
- Dans le Chapitre 3, nous introduisons un nouveau paradigme de mise en correspondance *sur demande* permettant d'étudier le lien entre la précision de la mise en correspondance et les performances d'estimation de pose de caméra. Nous proposons également une nouvelle architecture, basée sur notre nouvelle attention structurée, démontrant des performances comparables ou supérieures aux méthodes de l'état de l'art. Ces contributions ont fait l'objet d'une publication dans la conférence internationale VISAPP [Vilain et al., 2024].
- Dans le Chapitre 4, nous proposons la formulation et une étude de la problématique de multimodalité dans le cadre de la mise en correspondance dense. Nous proposons également une nouvelle architecture ainsi qu'une formulation de l'attention basée sur la recherche en faisceaux. Nous avons aussi développé une nouvelle manière de prédire la covisibilité entre deux images pour échantillonner efficacement des correspondances. Ces travaux feront l'objet d'un prochain article.





## **Chapitre 2**

### **Mise en correspondance d'images et mécanisme d'attention**

## Table des matières

2.1	Introduction . . . . .	19
2.2	Mise en correspondance d'images . . . . .	20
2.2.1	Introduction . . . . .	20
2.2.1.1	Formalisation . . . . .	20
2.2.1.2	Métriques d'évaluation . . . . .	22
2.2.1.3	Bases de données . . . . .	23
2.2.2	Paradigmes de mise en correspondance . . . . .	25
2.2.2.1	Matching éparse . . . . .	26
2.2.2.2	Matching semi-dense . . . . .	29
2.2.2.3	Matching dense . . . . .	30
2.2.3	Discussion . . . . .	31
2.3	Avènement de l'attention dans la vision par ordinateur . . . . .	32
2.3.1	Le mécanisme d'attention . . . . .	33
2.3.1.1	Formulation . . . . .	33
2.3.1.2	Encodage positionnel . . . . .	35
2.3.2	Limitations et variantes . . . . .	36
2.3.3	L'architecture Transformer . . . . .	38
2.3.4	L'attention en vision par ordinateur . . . . .	39
2.3.4.1	Origines et problématiques . . . . .	39
2.3.4.2	Architecture Transformer pour les images . . . . .	40
2.4	Relation entre l'attention et la mise en correspondance . . . . .	41
2.4.1	L'attention comme opération de communication entre les descripteurs . . . . .	41
2.4.2	Analyse de l'attention pour la mise en correspondance . . . . .	43
2.4.2.1	L'attention comme opération de communication . . . . .	45
2.4.2.2	Ablation de l'attention . . . . .	50
2.5	Conclusion . . . . .	51

## 2.1 Introduction

La mise en correspondance, ou *matching*, d'images est un pilier central de la vision par ordinateur, jouant un rôle clé dans des domaines variés tels que la reconstruction 3D, la navigation autonome, la réalité augmentée, ou encore l'estimation de mouvement. Cette tâche consiste à déterminer quels points ou régions dans une image correspondent aux mêmes points ou régions dans une autre image afin d'inférer des informations sur la scène observée, telles que la structure 3D ou la position relative des caméras. Toutefois, cette mise en correspondance de paires d'images présente des défis majeurs en raison des variations importantes de conditions visuelles, qu'il s'agisse de changements d'éclairage, de perspective, de texture ou de géométrie entre les images à comparer.

Historiquement, trois grands paradigmes de mise en correspondance ont été proposés. Le premier est celui du paradigme classique, appelé aussi éparsé ou *Sparse-to-Sparse*, dans lequel des points d'intérêt distincts sont détectés dans chaque image, puis décrits par des descripteurs locaux avant d'être mis en correspondance. Cette approche a produit des résultats satisfaisants mais est limitée aux régions texturées des images. Le second paradigme, semi-dense, s'efforce d'élargir la couverture en cherchant à établir des correspondances dans des régions plus vastes de l'image tout en restant computationnellement abordable. Enfin, le paradigme dense pousse encore plus loin la couverture en tentant d'associer chaque pixel d'une image à un pixel correspondant dans l'autre image. Bien que très prometteur, le *matching* dense reste coûteux en termes de calcul et pose encore des défis techniques importants.

Depuis 2017, l'introduction du mécanisme d'attention dans le domaine du traitement du langage naturel (NLP) a bouleversé la manière dont les modèles apprennent et utilisent les relations entre les éléments d'une séquence. L'attention, et plus particulièrement l'auto-attention, permet à un modèle de pondérer l'importance des différents éléments d'une séquence par rapport à un élément donné, et de modéliser des relations complexes à longue portée au sein des données. Ce mécanisme, qui a démontré des performances spectaculaires dans les tâches de traduction automatique et de génération de texte, est au cœur des architectures *Transformer*, comme les célèbres modèles BERT et GPT. L'attention a révolutionné le traitement de l'information séquentielle en offrant une approche flexible, capable de capturer des dépendances globales de manière efficace.

La vision par ordinateur a rapidement exploré le potentiel de ce mécanisme pour des tâches comme la classification d'images, la segmentation, et plus récemment, la mise en correspondance d'images. Bien que les images ne soient pas des séquences, les pixels et les caractéristiques locales peuvent être interprétés comme des entités à relier, analogues aux mots dans une phrase. Le mécanisme d'attention permet de pondérer l'importance relative des caractéristiques extraites de différentes parties d'une image ou entre plusieurs images, renforçant ainsi les connexions pertinentes et filtrant les informations non utiles. Cela marque une rupture avec les méthodes traditionnelles de mise en correspondance, qui reposaient sur la proximité locale des caractéristiques.

Dans le cadre de la mise en correspondance d'images, l'attention présente plusieurs avantages notables. Premièrement, elle permet d'établir des relations globales au-delà des simples similarités locales, ce qui est crucial pour résoudre des correspondances dans des scènes complexes ou répétitives. Cela permet également de trouver des correspondances dans des régions non texturées ou faiblement discriminatives. Enfin, contrairement aux architectures siamoises qui traitent indépendamment les deux images de la paire, l'attention permet une communication entre les représentations des deux images, élargissant ainsi le contexte pour mieux traiter des changements de points de vue importants ou des structures répétitives ambiguës. Ces capacités

sont particulièrement intéressantes pour des applications exigeantes comme la reconstruction 3D ou l'estimation de pose de caméra.

Dans cette partie, nous commencerons par une présentation générale des paradigmes de mise en correspondance d'images, en détaillant les approches classiques basées sur le matching de points d'intérêt, ainsi que les méthodes plus récentes de correspondance semi-dense et dense. Ensuite, nous introduirons le mécanisme d'attention, issu initialement du traitement du langage naturel, en expliquant ses fondements et son intégration progressive dans le domaine de la vision par ordinateur. Nous examinerons ensuite comment ce mécanisme révolutionne la mise en correspondance d'images, en facilitant l'établissement de relations globales et en améliorant la robustesse des correspondances. Nous aborderons différentes approches de matching modernes intégrant l'attention et conclurons par une analyse détaillée de l'utilisation de l'attention dans le contexte spécifique de la mise en correspondance d'images.

## 2.2 Mise en correspondance d'images

### 2.2.1 Introduction

Dans le cadre du traitement d'images, les tâches de mise en correspondance de caractéristiques locales visent à établir des correspondances précises entre des descripteurs extraits de différentes images. Ces descripteurs peuvent inclure divers éléments tels que des points d'intérêt, des régions distinctives ou des segments linéaires. La capacité à identifier et apparier efficacement ces caractéristiques similaires entre les images est essentielle pour de nombreuses applications en vision par ordinateur. Parmi ces applications figurent la fusion d'images [Tang et al., 2022a, Cao et al., 2022, Sun et al., 2023], la localisation visuelle [Sattler et al., 2012, Sattler et al., 2017], la reconstruction 3D basée sur le mouvement (Structure from Motion, SfM) [Agarwal et al., 2011, Heinly et al., 2015, Cadena et al., 2016], la localisation et la cartographie simultanées (SLAM) [Mur-Artal and Tardos, 2017, Zhao et al., 2019], l'estimation du flux optique [Liu et al., 2011, Weinzaepfel et al., 2013], ainsi que la recherche et la récupération d'images [Radencovic et al., 2019, Cao et al., 2020]. Toutefois, les transformations d'échelle, les différences de points de vue, les variations d'éclairage, la répétition de motifs et les diversités de texture peuvent générer des divergences significatives entre les images, compliquant l'établissement de correspondances fiables. Ainsi, garantir la précision et la fiabilité des correspondances locales reste un défi majeur en vision par ordinateur. Dans ce qui suit, nous formaliserons la tâche de mise en correspondance afin de déterminer les clés d'une correspondance précise, puis nous présenterons les critères d'évaluation des méthodes de matching.

#### 2.2.1.1 Formalisation

Nous travaillons avec un ensemble d'images  $I = \{I_i\}_{i=1\dots N}$  capturant une même scène 3D notée  $\Psi$ , avec  $I_i \in \mathbb{R}^{H_i \times W_i \times 3}$  où  $H_i$  est la hauteur de l'image  $i$  et  $W_i$  la largeur. Nous voulons trouver des correspondances entre une image source  $I_S \in I$  et une image cible (ou *image target*)  $I_T \in I$  possédant un recouvrement. Ces  $K$  correspondances sont un ensemble de paires de positions 2D  $C = \{(\mathbf{p}_{S,i}, \mathbf{p}_{T,i})\}_{i=1\dots K}$  où  $\mathbf{p}_{S,i}$  et  $\mathbf{p}_{T,i}$  sont covisibles, c'est-à-dire qu'ils sont respectivement les projections dans  $I_S$  et  $I_T$  d'un même point 3D  $\mathbf{P}_i \in \Psi$  (schématisé dans la Figure 2.1).

Trouver des correspondances  $\{(\mathbf{p}_{S,i}, \mathbf{p}_{T,i})\}_{i=1\dots K}$  dans l'ensemble total de paires de positions possibles  $\{(\mathbf{p}_{S,i}, \mathbf{p}_{T,j})\}_{i=1\dots H_S \times W_S, j=1\dots H_T \times W_T}$  uniquement à partir de l'information RGB contenue dans les pixels de  $I_S$  et  $I_T$  est impossible, car la simple intensité des pixels n'est pas

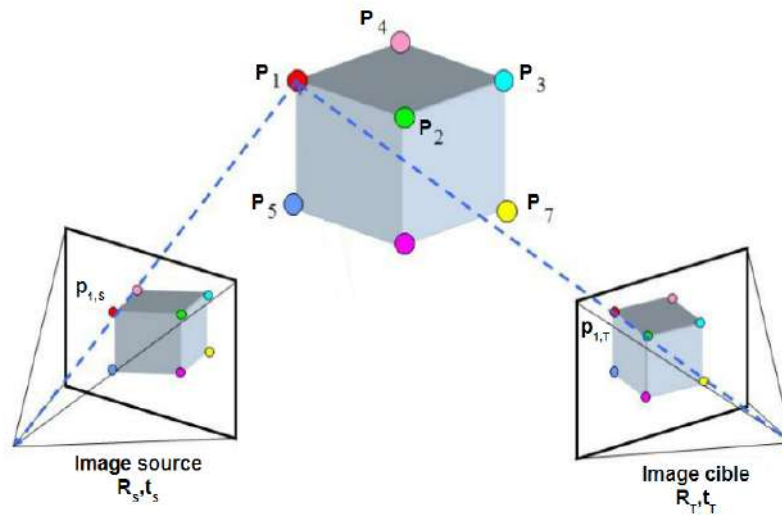


FIGURE 2.1 – **Illustration de la mise en correspondance.** Deux caméras capturent deux vues différentes d'une même scène 3D. L'objectif du matching est de trouver un ensemble de correspondances, c'est-à-dire des points de l'image source et de l'image cible qui correspondent aux mêmes points 3D de la scène observée. *Schéma adapté de [Agarwal et al., 2010].*

suffisamment discriminante pour les distinguer les uns des autres. Nous allons donc construire des représentations de haut niveau, des descripteurs,  $\mathbf{h}_{S,i}$  et  $\mathbf{h}_{T,i} \in \mathbb{R}^D$  pour les positions  $\mathbf{p}_{S,i}$  et  $\mathbf{p}_{T,j}$ . L'objectif de ces représentations est d'intégrer de l'information contextuelle de l'image pour faciliter la création de correspondances entre ces positions. Différentes stratégies peuvent être utilisées pour créer ces représentations, certaines seront décrites dans les sections suivantes. Néanmoins, pour un matching robuste et précis, nous souhaitons que les descripteurs suivent deux propriétés [Hassaballah et al., 2019] :

- **Discriminativité.** Pour deux positions  $\mathbf{p}_{S,i}$  et  $\mathbf{p}_{T,j}$  ne correspondant pas au même point 3D de  $\mathcal{S}$ , nous voulons que leurs descripteurs  $\mathbf{h}_{S,i}$  et  $\mathbf{h}_{T,j}$  soient différents, c'est-à-dire que la distance  $d(\mathbf{h}_{S,i}, \mathbf{h}_{T,j})$  soit grande. A contrario, s'ils correspondent au même point 3D, nous souhaitons que la distance  $d(\mathbf{h}_{S,i}, \mathbf{h}_{T,j})$  soit proche de 0. Cela peut représenter un véritable défi dans le cas de structures répétitives dans une image. La description devra intégrer le contexte de l'image pour que, même si des positions se ressemblent visuellement, leurs représentations soient différentes.
- **Invariance.** Les descriptions de deux positions  $\mathbf{p}_{S,i}$  et  $\mathbf{p}_{T,j}$  correspondant au même point 3D de  $\mathcal{S}$  doivent être robustes aux perturbations visuelles et garder une distance  $d(\mathbf{h}_{S,i}, \mathbf{h}_{T,j})$  proche de 0. Cette propriété peut être difficile à préserver, particulièrement dans les cas de grands changements de perspective ou de colorimétrie (ex : images jour/-nuit).

Considérer l'ensemble total de paires de positions possibles  $\{(\mathbf{p}_{S,i}, \mathbf{p}_{T,j})\}_{i=1 \dots H_S \times W_S, j=1 \dots H_T \times W_T}$  comme ensemble de recherche n'est également pas réaliste. Pour des images de taille  $640 \times 640$  pixels, on se retrouve avec environ  $10^{11}$  correspondances candidates. Il existe différentes méthodes pour réduire cet espace de recherche, que nous couvrirons dans les sections suivantes. Mais à nouveau, pour une sélection efficace de certaines positions afin de réduire l'espace de recherche, nous souhaitons conserver certaines propriétés [Tuytelaars and Mikolajczyk, 2008] :

- **Consistance.** Si l'on sélectionne la position  $\mathbf{p}_{S,i}$  correspondant au point 3D  $\mathbf{P}_i$  dans l'image source, alors on veut que  $\mathbf{p}_{T,i}$ , la projection de  $\mathbf{P}_i$  dans l'image cible, soit également sélectionnée. Ce critère pour la sélection de points est crucial pour les méthodes de mise en correspondance se basant sur des points d'intérêt, car un détecteur non consistant empêcherait toute correspondance précise. En pratique, une consistance parfaite est très difficile à atteindre.
- **Précision.** Bien sûr, nous voulons que les positions  $\mathbf{p}_{S,i}$  et  $\mathbf{p}_{T,i}$  soient précises et correspondent bien au même point 3D et non à des points voisins. Dans certaines paires d'images avec de grands changements de points de vue, une petite erreur dans l'espace image peut entraîner une grande erreur dans l'espace 3D et avoir un impact important si l'on souhaite par la suite estimer la position relative des caméras.

### 2.2.1.2 Métriques d'évaluation

Pour développer des méthodes de mise en correspondance qui seront utilisées pour des tâches comme la reconstruction 3D, il est nécessaire de mesurer leur capacité à établir des correspondances fiables et de construire des métriques capturant toutes les problématiques liées à la mise en correspondance.

Considérons un ensemble de  $K$  paires d'images de test  $\{(\mathbf{I}_{S_k}, \mathbf{I}_{T_k})\}_{k=1 \dots K}$  pour lesquelles on possède la vérité terrain sous la forme d'un ensemble de paires de points 2D  $\{(\mathbf{p}_{S_k,i}^{GT}, \mathbf{p}_{T_k,i}^{GT})\}_{i=1 \dots L}$ . Nous souhaiterions évaluer la capacité d'une méthode  $\mathcal{M}$  à prédire des positions estimées  $\hat{\mathbf{p}}_{T_k,i}$  dans la cible des points  $\mathbf{p}_{S_k,i}^{GT}$  de la source. Nous pouvons, dans un premier temps, nous intéresser à la précision de ces correspondances en calculant leur *précision*. La **précision de mise en correspondance** (*Matching Accuracy* ou MA) [Truong et al., 2020], c'est-à-dire la moyenne sur l'ensemble des images du ratio de correspondances correctes à différents seuils d'erreur en pixels ( $\eta$ ), peut être définie comme suit :

$$\text{MA}(\eta) = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{i=1}^{L_k} \left[ \left\| \hat{\mathbf{p}}_{T_k,i} - \mathbf{p}_{T_k,i}^{GT} \right\|_2 < \eta \right]}{L_k}, \quad (2.1)$$

où  $K$  est le nombre de paires d'images de test,  $L_k$  est le nombre de correspondances avec une vérité terrain dans la paire d'images  $\#k$ , et  $[\cdot]$  est le crochet d'Iverson. Cette métrique offre une bonne capacité d'analyse car elle permet de distinguer la précision d'une méthode (score de MA élevé pour des  $\eta$  faibles) et la robustesse (score de MA proche de 1 pour des  $\eta$  plus élevés). Cependant, beaucoup de méthodes n'estiment pas la totalité des correspondances disponibles dans la vérité terrain, mais uniquement un sous-ensemble. Dans ce cas, la *précision* ne sera calculée que sur ce sous-ensemble, et on peut rapporter le ratio du nombre de correspondances estimées sur le nombre de correspondances totales. On peut alors se demander s'il est préférable d'avoir une méthode avec peu de correspondances estimées mais très précises, ou beaucoup de correspondances estimées mais moins précises ?

Une métrique très largement utilisée qui contourne ce problème est l'**estimation de pose relative de caméra**. Ici, nous allons nous servir des correspondances estimées par  $\mathcal{M}$  pour trouver la rotation et la translation entre les caméras qui ont servi à capturer  $\mathbf{I}_{S_k}$  et  $\mathbf{I}_{T_k}$ . Les matrices estimées de rotation  $\hat{\mathbf{R}}$  et de translation  $\hat{\mathbf{t}}$  sont obtenues à partir des correspondances estimées par  $\mathcal{M}$  en utilisant l'algorithme RANSAC [Fischler and Bolles, 1981] combiné à l'algorithme des 5 points [Nistér, 2004]. L'erreur par rapport aux matrices de vérité terrain ( $\mathbf{R}_{GT}, \mathbf{t}_{GT}$ ) est ensuite exprimée sous la forme d'une erreur angulaire  $\theta_{total}$  :

$$\begin{aligned}
\theta_{\mathbf{t}} &= \cos^{-1} \left( \frac{\hat{\mathbf{t}} \cdot \mathbf{t}_{GT}}{\|\hat{\mathbf{t}}\| \|\mathbf{t}_{GT}\|} \right), \\
\theta_{\mathbf{R}} &= \cos^{-1} \left( \frac{\text{trace}(\hat{\mathbf{R}} \mathbf{R}_{GT}^T) - 1}{2} \right), \\
\theta_{total} &= \max(\theta_{\mathbf{t}}, \theta_{\mathbf{R}}),
\end{aligned} \tag{2.2}$$

On rapporte généralement cette erreur pour nos  $K$  paires de l'ensemble de test en utilisant l'aire sous la courbe (AUC) à différents seuils  $\tau_j$ , ce qui correspond au pourcentage de paires pour lesquelles l'erreur angulaire totale  $\theta_{total}$  est inférieure au seuil  $\tau_j$  :

$$AUC(\tau_j) = \frac{1}{K} \sum_{i=1}^K I(\theta_{total,i} \leq \tau_j), \tag{2.3}$$

où  $I(\cdot)$  est la fonction indicatrice qui vaut 1 si l'erreur est inférieure au seuil et 0 sinon. Évaluer une méthode à travers l'estimation de pose de caméra permet de se rapprocher du cadre applicatif de la mise en correspondance. Elle part du postulat que ce qui importe réellement, c'est la précision de l'estimation de la pose de la caméra, car c'est ce qui va permettre de reconstruire correctement la scène ou de positionner correctement la caméra dans l'espace. Cependant, certains facteurs comme le nombre de correspondances utilisées, leurs répartitions spatiales ou leurs précisions peuvent avoir un grand impact sur l'estimation de pose, mais ne sont pas quantifiés par la métrique.

Les métriques de **précision** et d'**estimation de pose de caméra** sont complémentaires car elles évaluent deux aspects différents mais interdépendants de la qualité des correspondances estimées par une méthode  $\mathcal{M}$ . La précision de matching mesure la proportion de correspondances correctes entre deux images, ce qui est essentiel pour assurer une bonne correspondance locale et est très proche de la tâche réelle effectuée par  $\mathcal{M}$ . Cependant, elle ne tient pas compte de l'impact géométrique global de ces correspondances sur des tâches applicatives comme la reconstruction 3D ou la localisation. En revanche, les métriques d'estimation de pose, telles que l'erreur angulaire, évaluent comment ces correspondances affectent la précision de la position et de l'orientation relatives des caméras, en prenant en compte la robustesse aux outliers et l'importance de la répartition géométrique des correspondances dans l'image. Cependant, si l'estimation de pose de caméra prend en compte la robustesse aux outliers et la répartition des correspondances, elle ne quantifie pas leur impact à travers le score d'AUC final de  $\mathcal{M}$ . Ensemble, ces métriques fournissent une évaluation complémentaire de la qualité des correspondances pour des applications géométriques en vision par ordinateur.

Notons également qu'il est difficile de construire des correspondances de vérité terrain précises pour le matching d'images. Cela nécessite de connaître avec exactitude la position 3D des points dans la scène et leurs projections correspondantes dans chaque image, ce qui implique une calibration précise des caméras et une connaissance parfaite de la géométrie de la scène. L'utilisation de capteurs comme la Kinect de Microsoft, de scanners laser 3D ou d'outils comme COLMAP [Schönberger and Frahm, 2016] permet de connaître la géométrie de la scène, mais intègre différents bruits qui peuvent avoir un impact sur nos métriques d'évaluation.

### 2.2.1.3 Bases de données

Dans cette section, nous présentons les bases de données (*datasets*) les plus utilisées pour évaluer la mise en correspondance d'images. La Figure 2.2 montre des exemples de paires d'images pour chacun de ces datasets.





FIGURE 2.2 – Exemples de paires d'images pour différentes bases de données.

**MegaDepth** [Li and Snavely, 2018] est un dataset d'estimation de pose relative. Il se compose d'environ un million d'images provenant de 196 scènes extérieures différentes, chacune avec des poses et des cartes de profondeur connues. Ces cartes de profondeur sont générées à partir de la reconstruction éparse des scènes et d'un calcul stéréo multi-vues à l'aide de COLMAP [Schönberger and Frahm, 2016, Schönberger et al., 2016]. Sa taille, ses changements extrêmes de point de vue, les motifs répétitifs dans ses images et sa diversité en font un bon dataset d'entraînement. Cependant, ses cartes de profondeur ne sont pas complètes et parfois uniquement disponibles pour l'élément principal de l'image. 1500 paires extraites de deux scènes, représentant la basilique Saint-Pierre de Rome et la Porte de Brandebourg à Berlin, forment un ensemble de test appelé MegaDepth-1500, qui est couramment utilisé pour évaluer les méthodes de mise en correspondance.

**HPatches** [Balntas et al., 2017] est un dataset de matching et d'estimation d'homographie, similaire à l'estimation de pose pour des scènes planaires. Il se compose de 116 scènes caractérisées par des variations de point de vue (59 scènes) et de luminosité (57 scènes). Chaque scène est composée de 5 paires d'images avec une complexité croissante, illustrant des scénarios divers aussi bien en extérieur qu'en intérieur. HPatches est très largement utilisé, mais ses matrices d'homographie sont peu précises pour certaines scènes composées de plusieurs plans.

**ScanNet** [Dai et al., 2017] est un dataset en intérieur à grande échelle conçu pour l'estimation de pose relative. Il est composé d'ensembles d'entraînement, de validation et de test bien définis, comprenant environ 230 millions de paires d'images issues de 1613 scènes. Ce jeu de données inclut des images et des cartes de profondeur, et contient davantage de régions avec peu de texture que MegaDepth. Les cartes de profondeur sont capturées par une Microsoft Kinect et



présentent un niveau de bruit assez élevé.

**Aachen Day-Night** [Zhang et al., 2020] est un dataset composé de 4328 images de jour et 98 images de nuit, utilisé pour des tâches de localisation. La présence de variations importantes d'illumination et de point de vue en fait un jeu de données d'évaluation intéressant, mais il est exclusivement composé d'images d'Aix-la-Chapelle, ce qui peut entraîner un manque de diversité.

**InLoc** [Taira et al., 2018] est un dataset en intérieur comprenant 9972 images avec des cartes de profondeur issues de scanners 3D. 329 images RGB de ce jeu sont utilisées pour tester la performance des algorithmes de localisation visuelle en intérieur.

**ETH3D** [Schöps et al., 2019] est un benchmark complet pour les algorithmes de stéréo multi-vues. Ce jeu de données englobe une vaste gamme de scènes, tant en intérieur qu'en extérieur, capturées à l'aide de caméras DSLR haute résolution et de vidéos stéréo. ETH3D offre divers protocoles d'évaluation adaptés à la stéréo multi-vues haute résolution, à la stéréo multi-vues basse résolution sur des séquences vidéo, ainsi qu'à la stéréo à deux vues.

## 2.2.2 Paradigmes de mise en correspondance

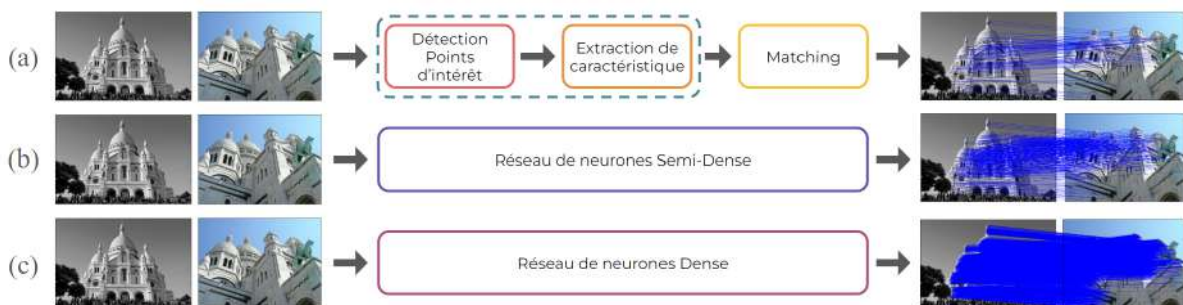


FIGURE 2.3 – **Présentation des principaux paradigmes de matching.** (a) **Paradigme épars** (ou **Sparse-to-Sparse**) où la tâche de mise en correspondance est divisée en trois étapes : détection des points d'intérêt, description de ces points et mise en correspondance. (b) **Paradigme semi-dense**, qui fusionne ces trois étapes en un seul réseau. (c) **Paradigme dense** qui vise à établir des correspondances pour tous les pixels. À droite, des exemples de correspondances produites par chaque paradigme.

Le matching d'image est un domaine de recherche qui a suscité l'intérêt depuis plusieurs décennies en raison de son large spectre d'applications en vision par ordinateur. Dès ses débuts, de nombreux défis ont émergé, notamment ceux liés à la robustesse des algorithmes face aux variations de conditions telles que l'illumination, la perspective ou encore les déformations géométriques. L'évolution rapide des technologies de traitement d'image et des capacités de calcul a permis de traiter progressivement ces problématiques, mais l'application du matching d'images reste complexe et soumise à des contraintes spécifiques. Cette diversité de besoins a conduit à l'émergence de plusieurs paradigmes dans l'approche du matching d'images, chacun ayant ses forces et ses limites en fonction des exigences applicatives.

Parmi ces paradigmes, on retrouve l'approche classique basée sur le **matching épars** de points d'intérêt, qui repose sur la correspondance de caractéristiques locales entre les images. Ensuite, nous trouvons le **matching semi-dense**, qui considère des régions plus vastes de

l'image tout en gardant une certaine parcimonie dans le nombre de points évalués. Enfin, le **matching dense**, quant à lui, vise à établir une correspondance pour chaque pixel, offrant ainsi une précision maximale au prix d'une plus grande complexité. Ces paradigmes sont illustrés dans la Figure 2.3.

Ces paradigmes constituent la base des techniques modernes de matching d'image et seront détaillés dans les sections suivantes. Nous introduirons les méthodes principales associées à chaque approche, en explorant leurs principes fondamentaux ainsi que les algorithmes emblématiques qui les illustrent. Nous omettons ici volontairement les méthodes utilisant le mécanisme d'attention, car elles seront abordées plus tard dans la section 2.4.

### 2.2.2.1 Matching épars

Le paradigme de matching épars, ou *Sparse-to-Sparse* (S2S), représente l'un des pipelines historiques les plus utilisés pour le matching d'images. Il repose sur un enchaînement de trois étapes distinctes mais complémentaires : (1) la détection des points d'intérêt dans une image, (2) l'extraction de descripteurs locaux pour caractériser ces points, et enfin (3) la recherche de correspondances entre les descripteurs des différentes images.

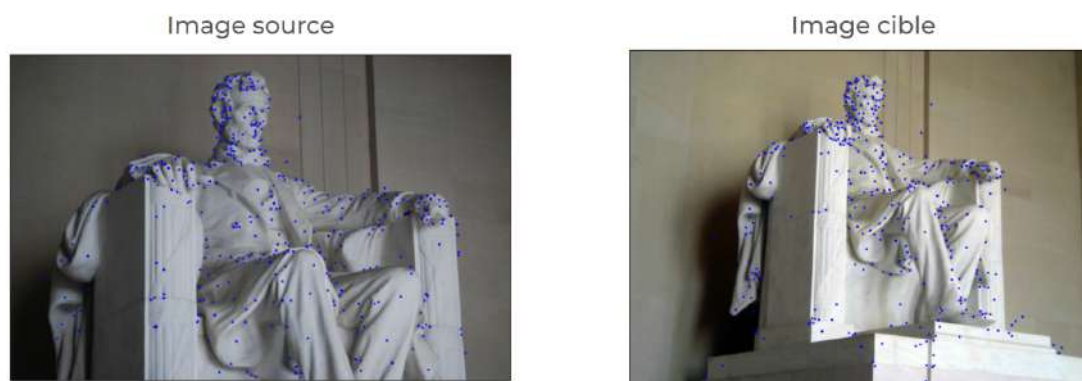


FIGURE 2.4 – **Points considérés par le paradigme de matching épars.** Des points d'intérêt sont détectés dans l'image source et l'image cible, puis décrits à l'aide de leur information locale, et enfin des correspondances sont trouvées entre ces deux ensembles de points. On remarque que dans les régions uniformes, comme les murs, il est difficile de trouver des points d'intérêt et donc des correspondances.

**La détection de points d'intérêt** est une étape cruciale du pipeline de matching d'images, car elle influence directement la qualité des correspondances entre les images. Une bonne détection doit être à la fois consistante (ou répétable) et précise. Cela signifie que les mêmes points d'intérêt doivent être détectés sous différentes conditions (changement de perspective, rotation, etc.), tout en étant suffisamment distinctifs pour être différenciés des autres points dans l'image.

Un exemple classique est le **détecteur de coins de Harris** [Harris and Stephens, 1988]. Ce détecteur fonctionne en calculant les gradients locaux des images à l'aide d'un noyau gaussien pour identifier les points ayant des variations importantes dans plusieurs directions, comme les coins. Ce détecteur est particulièrement répétable, grâce à son invariance à la translation 2D et à la rotation dans le plan. Cependant, il présente des limitations face à des perturbations visuelles plus complexes, telles que les changements d'échelle, de point de vue ou encore d'éclairage. Pour répondre à ces besoins de robustesse, des méthodes plus sophistiquées, comme **SIFT** (*Scale-Invariant Feature Transform*) [Lowe, 1999], ont été développées et restent largement

utilisées à ce jour. SIFT repose sur un calcul de la différence de Gaussienne appliqué à des images multi-échelles, ainsi que sur le calcul d'orientation principale, effectué à partir d'un histogramme local de gradients. Cela rend ces détecteurs particulièrement robustes aux rotations et aux changements d'échelle. SIFT, pouvant être trop lent pour certains cadres applicatifs, a vu l'apparition d'autres détecteurs, comme SURF (*Speeded-Up Robust Features*) [Bay et al., 2006] ou FAST (*Features from Accelerated Segment Test*) [Rosten and Drummond, 2006], spécifiquement conçus pour des applications en temps réel, où la vitesse de traitement est primordiale.

**L'extraction de descripteurs locaux** a pour but de créer des représentations discriminatives, c'est-à-dire capables de différencier des points d'intérêt uniques et invariants face à diverses transformations visuelles, telles que la rotation, le changement d'échelle ou les variations d'éclairage.

L'algorithme **SIFT** propose également une stratégie pour décrire les points d'intérêt qu'il détecte. Il repose sur un ensemble d'histogrammes d'orientations extraits d'une fenêtre locale autour de chaque point d'intérêt détecté. Ce processus de construction des descripteurs permet de capturer la distribution des gradients d'intensité autour du point clé, fournissant ainsi une représentation riche et robuste des caractéristiques locales. La structure de ces histogrammes rend le descripteur SIFT non seulement discriminatif, mais aussi invariant aux rotations et aux changements d'échelle. Un autre descripteur notable est **BRIEF** (*Binary Robust Independent Elementary Features*) [Calonder et al., 2010], conçu pour être à la fois très rapide à calculer et hautement discriminatif. Après avoir appliqué un lissage gaussien sur l'image afin de réduire le bruit, BRIEF procède en comparant les valeurs d'intensité entre le pixel du point d'intérêt et ceux d'autres points d'intérêt voisins. Ces comparaisons de valeurs sont ensuite encodées sous forme d'un vecteur binaire, facilitant ainsi des correspondances rapides entre les images. Cependant, il souffre d'une mauvaise robustesse face aux rotations, car les comparaisons d'intensité ne prennent pas en compte l'orientation du point d'intérêt. **ORB** (*Oriented FAST and Rotated BRIEF*) [Rublee et al., 2011] reprend le principe de BRIEF, mais améliore sa robustesse aux rotations en intégrant l'estimation de l'orientation du point d'intérêt fournie par le détecteur FAST.

**Détection et description jointes.** Avec l'émergence de l'apprentissage profond, et particulièrement des réseaux de neurones convolutifs (CNN), une nouvelle approche a vu le jour, cherchant à fusionner les étapes de détection et de description des points d'intérêt dans les images. Ce changement marque un tournant majeur, passant des méthodes traditionnelles basées sur des caractéristiques manuellement définies à des techniques basées sur l'apprentissage. Les CNN se sont révélés extrêmement efficaces pour fournir des descripteurs invariants aux transformations telles que la rotation, l'échelle et les variations d'éclairage. Leur utilisation dans une architecture siamoise, où les mêmes poids sont utilisés pour traiter en tandem deux vecteurs d'entrée différents afin de calculer des vecteurs de sortie comparables, s'est montrée particulièrement efficace pour la mise en correspondance. Cette approche a entraîné un gain significatif en robustesse par rapport aux méthodes traditionnelles, qui reposaient sur des heuristiques manuelles.

L'une des premières tentatives pour combiner détection et description dans un modèle d'apprentissage profond est le modèle **LIFT** (*Learned Invariant Feature Transform*) [Yi et al., 2016]. LIFT utilise trois réseaux CNN distincts : le premier est un détecteur qui prend en entrée un patch d'image et produit des cartes de score pour identifier les points d'intérêt ; le deuxième est un estimateur d'orientation qui calcule l'orientation principale du patch ; et le troisième est un descripteur qui génère les caractéristiques locales. Ces trois réseaux sont entraînés de manière non supervisée en utilisant une fonction de coût contrastive, garantissant ainsi la cohérence

et la pertinence des points détectés et décrits. Contrairement à LIFT, qui travaille sur des patches d'images, **Superpoint** [DeTone et al., 2018] traite l'image entière en une seule fois. Il utilise un encodeur CNN siamois, suivi de deux décodeurs distincts : l'un pour la détection des points d'intérêt et l'autre pour la description. Cette approche permet à Superpoint de détecter un ensemble beaucoup plus riche de points d'intérêt, et de le faire de manière répétable dans diverses conditions. Superpoint illustre ainsi parfaitement l'avantage des architectures d'apprentissage basées sur les CNN pour les tâches conjointes de détection et de description, offrant une solution plus efficace et robuste que les approches traditionnelles.

**L'étape de matching** a pour objectif de trouver les correspondances entre les descripteurs extraits dans l'image source  $I_S$  et l'image cible  $I_T$ . Cette étape est étroitement liée aux phases précédentes de détection et de description, car la qualité des correspondances dépend fortement de la précision et de la robustesse des descripteurs générés.

La méthode la plus simple pour identifier la correspondance d'un point d'intérêt dans l'image source est de chercher son plus proche voisin (*Nearest Neighbour*, NN) parmi les points d'intérêt de l'image cible, en se basant sur la distance entre les descripteurs. Cette approche consiste à parcourir de manière exhaustive tous les descripteurs issus de  $I_S$  et à sélectionner celui dont la distance avec le descripteur issu de  $I_T$  est la plus petite. Cependant, cette méthode peut être coûteuse en termes de calcul lorsque le nombre de points d'intérêt est important. On lui préfère alors la version des k-plus proches voisins (k-NN) [Fix and Hodges, 1989]. Par la suite, une vérification des plus proches voisins mutuels (*mutual nearest neighbour*) [Chidananda Gowda and Krishna, 1978] peut être appliquée pour filtrer les correspondances incorrectes. Cette étape permet de supprimer les outliers non covisibles, augmentant ainsi la fiabilité des correspondances. Des stratégies basées sur l'apprentissage peuvent également être employées pour le matching [Yi et al., 2018, Zhang et al., 2019]. Dans ces approches, un réseau de neurones multicouches (MLP) peut être entraîné pour prédire des scores de confiance sur les correspondances. Plutôt que de simplement se baser sur la distance entre des descripteurs, le MLP peut apprendre à reconnaître des correspondances plausibles en fonction des caractéristiques des descripteurs dans les images source et cible.

Avec le temps, le paradigme de matching éparsé (S2S) a prouvé qu'il était à la fois fiable et rapide. Grâce aux nombreuses améliorations, il est devenu une méthode populaire pour le matching d'images. Cependant, malgré ses succès, certaines limitations persistent. Tout d'abord, la mise en correspondance ne peut se faire qu'entre les points d'intérêt trouvés par le détecteur, ce qui restreint souvent les correspondances aux régions texturées des images. Cela peut poser problème dans des environnements comme les scènes d'intérieur, où de grandes portions de l'image peuvent être uniformes. Dans de tels scénarios, les méthodes S2S échouent à trouver des correspondances fiables, car les détecteurs peinent à identifier des points d'intérêt dans des zones non texturées. De plus, le nombre de points d'intérêt détectés est souvent limité, ce qui signifie que le S2S n'exploite qu'une petite fraction des images. Cela conduit à une perte d'informations potentiellement importantes, surtout lorsque des régions cruciales de l'image ne sont pas prises en compte. Enfin, les descripteurs locaux utilisés dans le paradigme S2S sont souvent sensibles aux variations d'éclairage ou de perspective. Les changements de conditions visuelles entre les images source et cible peuvent réduire considérablement la qualité des correspondances, rendant ces méthodes moins robustes dans des environnements visuellement complexes. Ces limitations soulignent la nécessité de méthodes plus avancées, capables de détecter des points d'intérêt de manière plus exhaustive et plus robuste face aux variations visuelles.



### 2.2.2.2 Matching semi-dense

Le paradigme semi-dense s'est développé pour surmonter les limitations du paradigme Detection-Description-Matching. Il vise à densifier les correspondances tout en maintenant un compromis entre couverture et coût computationnel. Dans ce paradigme, les correspondances sont trouvées pour tous les points de l'image source, mais à une résolution plus basse. Une fois ramenées à la résolution d'origine, cela revient à considérer les points situés sur une grille régulière dans l'image source, d'où le terme "semi-dense". Contrairement aux méthodes éparées, les correspondances ne se limitent plus aux seules régions texturées, et le nombre de points pris en compte est constant et souvent bien plus élevé que dans les approches basées sur des détecteurs de points d'intérêt. Cette approche permet d'améliorer significativement la couverture des correspondances et leur robustesse dans des environnements complexes ou faiblement texturés.

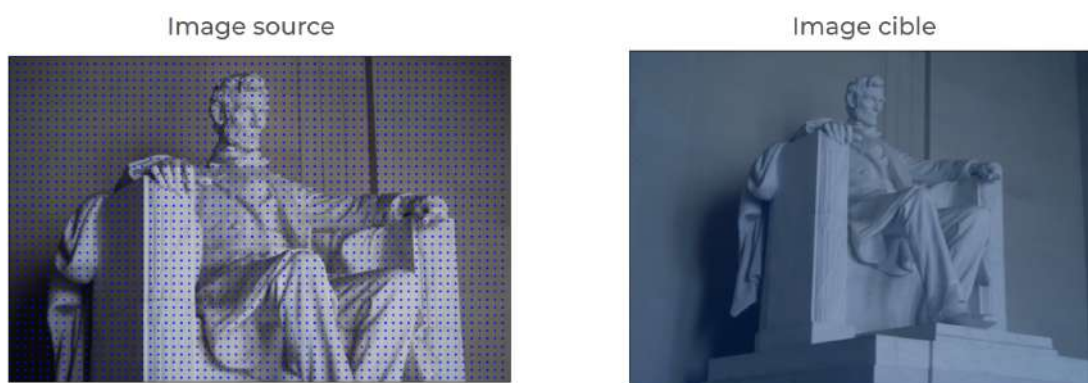


FIGURE 2.5 – **Points considérés par le paradigme de matching semi-dense.** Contrairement au paradigme éparse, le matching semi-dense ne repose pas sur la détection de points d'intérêt. Les points considérés sont répartis de manière homogène dans la source et peuvent trouver leur correspondant à n'importe quelle position de la cible après le raffinement des correspondances grossières. Cependant, la grille de points considérés dans la source manque une grande partie des points, qui, pour certains, peuvent être très informatifs et auraient pu être utiles pour la mise en correspondance.

**NCNet** (*Neighbourhood Consensus Networks*) [Rocco et al., 2018] se distingue des méthodes traditionnelles en ne s'appuyant pas sur la détection explicite de points d'intérêts, mais plutôt sur les relations de voisinage spatial dans un espace de correspondances. Après l'extraction de cartes de descripteurs via un CNN siamois, NCNet calcule la similarité cosinus entre tous les descripteurs des images source et cible, créant un espace de corrélation 4D. Ensuite, un mécanisme de consensus spatial (convolutions 4D) est utilisé sur le volume de corrélation pour évaluer la cohérence locale dans les régions de l'image. Cela signifie que seules les correspondances cohérentes au sein de voisinages locaux sont sélectionnées, tandis que les incohérentes sont filtrées via un mécanisme *Soft Mutual Nearest Neighbour*. Le réseau est entraîné de manière faiblement supervisée avec des paires d'images positives et négatives. Les correspondances les plus prometteuses sont finalement extraites en appliquant des softmax sur le volume de corrélation. Grâce à cette approche, NCNet est particulièrement efficace pour trouver des correspondances même dans les régions non-texturées, où les méthodes traditionnelles échouent souvent. La construction du volume de corrélation 4D et son traitement avec des convolutions est très coûteux en mémoire, et bien que l'on puisse utiliser des convolutions sous-maillées (*Submanifold Sparse Convolutions*) [Rocco et al., 2020a] pour réduire les calculs, il est difficile de construire un réseau NCNet très profond. **DualRC-Net** (*Dual-Resolution Correspondence*

*Networks*) aborde le problème de l'estimation des correspondances en adoptant une approche grossière puis fine (*coarse-to-fine*). DualRC-Net utilise un *Feature Pyramid Network* (FPN) siamois pour extraire des cartes de descripteurs à deux niveaux de résolution : grossière et fine. Dans un premier temps, des correspondances globales sont établies à une résolution grossière, similaire à l'approche de NCNet, en calculant un volume de corrélation 4D. Par la suite, un volume de corrélation à pleine résolution est généré et les prédictions des deux réseaux (grossière et fine) sont combinées pour produire des correspondances finales à la résolution d'origine. Ces deux étapes, entraînées conjointement, permettent de capturer les correspondances globales rapidement et de les affiner avec précision, tout en maintenant une gestion efficace des calculs. Certaines méthodes comme **Patch2Pix** [Zhou et al., 2021] proposent d'affiner les correspondances au niveau des pixels en utilisant des contraintes épipolaires. Les contraintes épipolaires sont des relations géométriques fondamentales entre deux images prises à partir de points de vue différents, qui décrivent comment les points de l'image source se projettent dans l'image cible en fonction de la position relative des caméras. Ces contraintes épipolaires permettent de réduire l'espace de recherche à pleine résolution. Cette approche semi-dense est utilisée par de nombreuses méthodes intégrant le mécanisme d'attention, que nous développerons dans la section 2.4.

Le paradigme semi-dense offre l'avantage de ne plus nécessiter de détecteur de points d'intérêt, élargissant ainsi l'espace de recherche des correspondances à l'ensemble de l'image. Le matching s'effectue à basse résolution, entre les régions des images, et la sélection des correspondances est réalisée a posteriori dans l'espace de corrélation. Grâce à cette approche, toutes les régions de l'image, y compris les zones non texturées, peuvent désormais participer aux correspondances, améliorant ainsi la couverture globale de l'image. Ce paradigme a permis une nette amélioration des résultats d'estimation de pose de caméra en comparaison aux méthodes S2S. Cependant, une limitation demeure : le matching à basse résolution extrait généralement une seule correspondance par région, alors qu'il serait possible d'en extraire davantage pour capturer des détails plus fins.

### 2.2.2.3 Matching dense

Le matching dense a récemment regagné en popularité grâce à l'augmentation des capacités de calcul modernes. Contrairement aux techniques semi-denses, qui se limitent à certaines régions de l'image, le matching dense vise à établir des correspondances pour chaque pixel de l'image source, offrant ainsi une précision accrue et une répartition homogène des correspondances sur l'ensemble de l'image. Cette approche permet d'obtenir des résultats plus détaillés et plus complets, en particulier dans des scènes complexes où les méthodes semi-denses pourraient manquer de finesse.

**DGC-Net** (*Dense Geometric Correspondence Network*) [Melekhov et al., 2019] utilise une pyramide de caractéristiques avec une corrélation globale à plusieurs résolutions pour capturer des correspondances, même en présence de transformations complexes. Il affine les correspondances via un mécanisme de "*warping layer*" qui ajuste progressivement les alignements à chaque niveau de la pyramide. **GLU-Net** (*Global-Local Universal Network*) [Truong et al., 2020], de son côté, combine des informations globales et locales en utilisant une approche pyramidale similaire. Une première étape de correspondance globale est suivie d'un raffinement local pour ajuster les détails fins, capturant ainsi les structures globales puis les précisions locales. **PDC-Net** (*Probabilistic Dense Correspondence Network*) [Truong et al., 2021a], en revanche, se distingue par une approche probabiliste qui génère une distribution de probabilités pour chaque correspondance potentielle et estime la confiance de chaque correspondance,



FIGURE 2.6 – **Points considérés par le paradigme de matching dense.** Ici, on cherche pour tous les points de l'image source une correspondance à n'importe quelle position de l'image cible. Cela permet de capturer à la fois les points d'intérêts et les points dans les régions uniformes. Cependant, le nombre de points à considérer étant nettement plus élevé que pour les méthodes semi-denses, l'empreinte mémoire et le temps de calcul des méthodes denses s'avèrent plus importants.

gérant ainsi mieux les zones ambiguës ou faiblement texturées. Cette estimation d'incertitude est essentielle pour renforcer la fiabilité des correspondances et permettre une meilleure sélection des correspondances qui seront utilisées pour l'estimation de pose de caméra. Malgré l'estimation d'incertitude de PDC-Net, ces trois méthodes n'arrivent pas à égaler les performances d'estimation de pose des meilleures méthodes semi-denses utilisant de l'attention. Très récemment, **DKM** [Edstedt et al., 2023] a surpassé les méthodes semi-denses en estimation de pose grâce au paradigme de matching dense. DKM utilise des descripteurs kernelisés, ressemblant à une opération d'attention, afin de mieux capturer les relations non linéaires entre les pixels. D'abord, des caractéristiques locales sont extraites des deux images à l'aide d'un réseau convolutif profond. Ensuite, ces caractéristiques sont transformées à l'aide de noyaux (kernels) non linéaires, permettant de capturer des interactions plus complexes entre les pixels que les noyaux de convolution. Une carte de correspondances denses est ensuite générée en comparant ces descripteurs kernelisés pour chaque pixel dans les deux images. Combinée à un raffinement convolutif et une estimation de la covisibilité entre les images, DKM se montre particulièrement performant pour des transformations géométriques complexes.

Le matching dense présente plusieurs avantages significatifs, notamment sa capacité à fournir des correspondances pixel-par-pixel entre deux images, capturant ainsi l'intégralité des informations d'une scène, ce qui est essentiel pour certaines applications comme la reconstruction 3D. Les performances de DKM en estimation de pose suggèrent que le paradigme dense est prometteur pour la mise en correspondance d'images. Cependant, le principal défi de ce paradigme réside dans son coût computationnel élevé, en particulier lorsque les images sont de haute résolution, ce qui entraîne une augmentation exponentielle du nombre de correspondances à traiter.

### 2.2.3 Discussion

Chaque paradigme de matching apporte des avantages spécifiques adaptés à différentes contraintes et applications. Le paradigme éparse, basé sur la détection de points d'intérêt, est particulièrement prisé pour sa légèreté et sa rapidité. Grâce à des détecteurs robustes, ce paradigme est idéal pour les systèmes embarqués ou les applications en temps réel, offrant un bon compromis entre précision et coût computationnel, notamment dans les scènes extérieures texturées. Il présente cependant des limites dans les environnements à faible texture ou avec des

changements de perspective importants.

Le matching semi-dense offre un équilibre entre précision et performance. Ces méthodes sont capables de trouver des correspondances non seulement dans les régions texturées, mais aussi dans des zones uniformes, là où les méthodes S2S échouent souvent. Cela permet un meilleur conditionnement pour des problématiques d'estimation de pose ou de reconstruction. Il peut être rapide si le cadre applicatif permet l'utilisation d'un processeur graphique (GPU).

Enfin, le matching dense offre la couverture la plus complète en fournissant des correspondances pour chaque pixel, ce qui permet d'exploiter pleinement toutes les informations disponibles dans les images. DKM montre que cette approche peut être particulièrement robuste face aux grands changements de perspective et de géométrie, ce qui la rend idéale pour des tâches complexes de reconstruction 3D de haute précision. Cependant, le matching dense est extrêmement coûteux en calcul, rendant son utilisation peu pratique dans les systèmes de navigation embarqués ou les applications en temps réel nécessitant une faible latence.

## 2.3 Avènement de l'attention dans la vision par ordinateur

Le mécanisme d'attention dans le domaine du traitement du langage naturel (NLP) a révolutionné la manière dont les relations complexes entre différents éléments d'une séquence peuvent être modélisées. Introduit par les travaux sur l'architecture Transformer [Vaswani et al., 2017], le mécanisme d'attention a permis de surmonter les limitations des architectures basées sur les réseaux récurrents (*Recurrent Neural Network* ou RNN), en offrant une approche capable de capturer efficacement les dépendances à longue distance. L'idée principale derrière l'attention est de permettre à un modèle d'identifier les relations les plus pertinentes entre différents éléments d'une séquence, en pondérant leur importance relative, au lieu de traiter les informations de manière linéaire ou hiérarchique, comme dans les RNN ou les CNN.

L'un des aspects clés du mécanisme d'attention est la notion d'auto-attention (*self-attention*), où chaque élément d'une séquence est mis en relation avec tous les autres éléments de cette même séquence. Cette capacité à traiter les dépendances globales sans contrainte séquentielle a montré des améliorations spectaculaires dans des tâches comme la traduction automatique, la génération de texte et la compréhension de documents. L'utilisation de l'attention a donné naissance à des modèles puissants comme BERT et GPT, qui dominent aujourd'hui le domaine du NLP.

Suite à ces succès, le mécanisme d'attention a rapidement été adapté à la vision par ordinateur, notamment pour des tâches complexes telles que la classification d'images, la segmentation, et également la mise en correspondance d'images. L'idée de modéliser des relations globales, indépendamment de la proximité spatiale des caractéristiques locales, a permis d'améliorer significativement les performances des modèles dans la capture des relations non locales, cruciales pour comprendre des scènes visuelles complexes. La suite de ce chapitre présentera en détail le fonctionnement du mécanisme d'attention et expliquera comment il a été adapté aux besoins spécifiques de la vision par ordinateur, ouvrant ainsi son application à la mise en correspondance d'images.



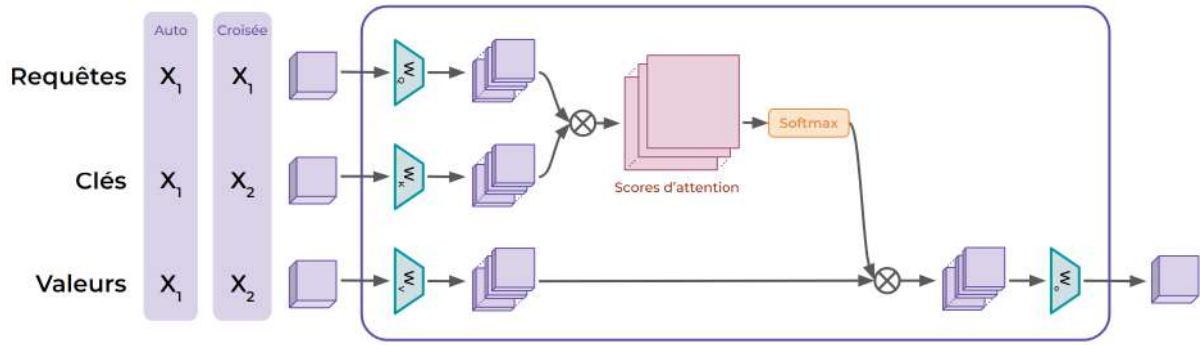


FIGURE 2.7 – **Schéma de fonctionnement du mécanisme d'attention.** Des requêtes, clés et valeurs sont créées à partir de différentes entrées en fonction qu'il s'agisse d'une auto-attention ou d'une attention-croisée. L'attention peut être vue comme une manière de reconstruire dynamiquement les requêtes avec l'information des valeurs, en fonction de la similarité entre les requêtes et les clés.

## 2.3.1 Le mécanisme d'attention

### 2.3.1.1 Formulation

Une fonction d'attention peut être décrite comme mappant un ensemble de requêtes (*queries*) et un ensemble de paires clé-valeur (*keys-values*) à une sortie, où les requêtes  $\mathbf{q}$ , les clés  $\mathbf{k}$  et les valeurs  $\mathbf{v}$  sont tous des vecteurs :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_q}}\right)V, \quad (2.4)$$

Avec,  $Q = XW_q = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_m]^T$ ,  
 $K = XW_k = [\mathbf{k}_1 \quad \mathbf{k}_2 \quad \dots \quad \mathbf{k}_n]^T$ ,  
 $V = XW_v = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_n]^T$ ,

Ici,  $X$  est la séquence de vecteurs d'entrée de notre couche d'attention.  $W_q \in \mathbb{R}^{m \times d_q}$ ,  $W_k \in \mathbb{R}^{n \times d_q}$  et  $W_v \in \mathbb{R}^{n \times d_v}$  sont des matrices de projection apprises pour nos requêtes, clés et valeurs. Notons que le nombre de clés et de valeurs  $n$  doit être égal, mais le nombre de requêtes  $m$  peut varier. De même, la dimension des clés et des requêtes  $d_q$  doit correspondre, mais celle des valeurs  $d_v$  peut varier. En effet, l'Eq 4.4 correspond à une **auto-attention** où  $Q$ ,  $K$ , et  $V$  sont construits à partir de la même séquence  $X$ . Dans ce cas précis, nous avons donc  $m = n$  et  $d_q = d_v$ . Il est également possible de faire de l'attention entre deux séquences, appelée **attention-croisée**. Pour deux séquences  $X_1$  et  $X_2$ , les requêtes seront construites à partir de la première séquence,  $Q = W_q X_1$ , et les clés-valeurs à partir de la seconde,  $K = W_k X_2$ ,  $V = W_v X_2$ .

Décomposons l'opération d'attention pour en comprendre son fonctionnement :

$$\begin{aligned}
 S = \text{softmax} \left( \frac{QK^T}{\sqrt{d_q}} \right) &= \begin{pmatrix} \text{softmax} \left( \frac{1}{\sqrt{d_q}} \langle \mathbf{q}_1 \cdot \mathbf{k}_1, \mathbf{q}_1 \cdot \mathbf{k}_2, \dots, \mathbf{q}_1 \cdot \mathbf{k}_n \rangle \right) \\ \text{softmax} \left( \frac{1}{\sqrt{d_q}} \langle \mathbf{q}_2 \cdot \mathbf{k}_1, \mathbf{q}_2 \cdot \mathbf{k}_2, \dots, \mathbf{q}_2 \cdot \mathbf{k}_n \rangle \right) \\ \vdots \\ \text{softmax} \left( \frac{1}{\sqrt{d_q}} \langle \mathbf{q}_m \cdot \mathbf{k}_1, \mathbf{q}_m \cdot \mathbf{k}_2, \dots, \mathbf{q}_m \cdot \mathbf{k}_n \rangle \right) \end{pmatrix}, \\
 &= \begin{pmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \dots & s_{m,n} \end{pmatrix}, \tag{2.5}
 \end{aligned}$$

Cette matrice  $S$ , représentant le produit entre les requêtes et les clés, est appelée **carte d'attention**. Nous pouvons exprimer le produit scalaire entre une clé et une requête données en termes d'angle  $\theta$  entre eux :  $\mathbf{q}_i \cdot \mathbf{k}_j = |\mathbf{q}_i| |\mathbf{k}_j| \cos(\theta)$ . Par conséquent, en ignorant la magnitude pour le moment, plus l'angle de la clé  $\mathbf{k}_j$  et de la requête  $\mathbf{q}_i$  sont proches, plus leur représentation dans la carte d'attention est importante. L'amplitude du produit scalaire peut être affectée par la dimensionnalité  $d_q$  des vecteurs. La division par  $\sqrt{d_q}$  n'affecte pas la distribution globale des scores d'attention mais garantit simplement que les scores gardent une magnitude raisonnable et ne submergent pas les calculs ultérieurs du mécanisme d'attention. L'exponentiation par le *softmax* amplifie les valeurs de cosinus positives et diminue les valeurs négatives. Comme le *softmax* est appliqué sur les lignes de la carte d'attention, chaque ligne peut être vue comme une distribution de probabilités indiquant la ressemblance d'une requête  $\mathbf{q}_i$  par rapport aux valeurs de  $K$ . Ensuite, on multiplie la carte d'attention avec la matrice de valeurs :

$$\text{softmax} \left( \frac{QK^T}{\sqrt{d_q}} \right) V = \begin{pmatrix} \sum_{i=1}^n s_{1,i} \mathbf{v}_i \\ \sum_{i=1}^n s_{2,i} \mathbf{v}_i \\ \vdots \\ \sum_{i=1}^n s_{m,i} \mathbf{v}_i \end{pmatrix}, \tag{2.6}$$

Ce qu'il faut retenir ici, c'est que l'attention aboutit à une série de moyennes pondérées des lignes de  $V$ , où la pondération dépend de la relation entre les requêtes et les clés d'entrée. Chaque des  $m$  requêtes dans  $Q$  aboutit à une somme pondérée spécifique des vecteurs de valeurs. On peut maintenant comprendre les termes *requête-clé-valeur* (*query-key-value*) inspirés des bases de données : on cherche des clés en fonction de nos requêtes et on récupère les valeurs en fonction des clés trouvées.

Un autre composant essentiel proposé par [Vaswani et al., 2017] est la **Multi-Head Attention**. Au lieu d'effectuer une seule fonction d'attention, on projette notre séquence d'entrée  $h$  fois avec différentes projections linéaires, et nous effectuons  $h$  fonctions d'attention en parallèle.

$$\text{Multi-Head Attention}(X_1, X_2, X_3) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_o, \tag{2.7}$$

$$\text{avec, } \text{head}_i = \text{Attention}(X_1 W_{q,i}, X_2 W_{k,i}, X_3 W_{v,i}),$$

La fonction d'attention est calculée en parallèle à partir de chacun de ces ensembles d'entrées, et les résultats sont concaténés côte à côte dans une seule matrice. Le résultat final est obtenu via une transformation linéaire  $W_o$  de la matrice concaténée, afin de retrouver la même dimensionnalité que l'entrée. Les matrices de projection des requêtes, clés et valeurs étant différentes pour chaque head, les  $h$  cartes d'attention sont donc calculées dans des espaces distincts. Cela permet au modèle de traiter conjointement des informations provenant de différents sous-espaces à différentes positions. Par exemple, pour la requête  $\mathbf{q}_i$ , son produit scalaire avec  $\mathbf{k}_j$  peut être faible dans l'espace de la  $head_1$ , mais très élevé dans l'espace de la  $head_2$ .

Dans le mécanisme d'attention, les paramètres qui seront appris lors de l'entraînement sont les matrices de projection  $W_q$ ,  $W_k$ ,  $W_v$  et  $W_o$ . Une couche d'attention va donc apprendre à trouver les espaces optimaux dans lesquels projeter la séquence d'entrée afin de : effectuer des comparaisons efficaces ( $W_q$ ,  $W_k$ ), et organiser l'information de façon pertinente ( $W_v$ ). Elle va aussi apprendre à reprojeter l'information de tous ces espaces dans un espace commun ( $W_o$ ). De cette manière, l'auto-attention et l'attention-croisée nous permettent de relier efficacement et rapidement chaque élément d'une séquence à tous les éléments de la même séquence ou d'une autre séquence. La Figure 2.7 schématise le fonctionnement du mécanisme d'attention.

### 2.3.1.2 Encodage positionnel

L'ordre des mots dans une phrase ou l'organisation des pixels dans une image est une caractéristique fondamentale pour comprendre la sémantique de ces données. Les RNN peuvent capturer ces relations de position grâce à leur aspect séquentiel. Les CNN capturent également naturellement l'information spatiale et la position relative des éléments dans les données, grâce à la localité et au partage des poids des filtres de convolution. En revanche, l'attention est invariante à la permutation, c'est-à-dire que l'on peut changer l'ordre des requêtes, clés et valeurs sans affecter le résultat final. Pour pallier cela, il est nécessaire d'ajouter explicitement des informations positionnelles afin de modéliser correctement les relations entre les éléments d'une séquence. Plusieurs approches ont été développées pour intégrer cette information positionnelle dans l'opération d'attention, chacune avec ses propres avantages et limitations.

**L'encodage positionnel sinusoïdal** est la méthode proposée dans l'architecture Transformer originale [Vaswani et al., 2017]. Cette technique utilise des fonctions sinus et cosinus pour encoder les positions, de manière à ce que les valeurs d'encodage positionnel varient de manière régulière en fonction de la position ( $pos$ ) et de la dimension. Une fois créée, l'information de position est simplement additionnée aux descripteurs.

$$\begin{aligned} PE(pos, 2i) &= \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \\ PE(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right), \end{aligned} \tag{2.8}$$

Cette formulation permet de créer des encodages où les positions proches sont représentées par des vecteurs similaires, facilitant ainsi la modélisation des relations séquentielles. Un avantage clé de cette approche est qu'elle ne nécessite pas de paramètres appris, ce qui la rend particulièrement efficace pour généraliser à des séquences de longueur différente de celles rencontrées lors de l'entraînement.

**Encodage positionnel appris [Devlin et al., 2019, Radford et al., 2019].** Une autre approche consiste à apprendre les vecteurs d'encodage positionnel directement à partir des données. Chaque position dans la séquence est associée à un vecteur de dimension fixe, appris conjointement avec les autres paramètres du modèle. L'information de position ainsi créée est également additionnée aux features.

$$PE(pos) = \mathbb{W}_{pos}, \quad (2.9)$$

Cette méthode a l'avantage d'être plus flexible et de pouvoir capturer des relations positionnelles spécifiques aux données. Cependant, elle nécessite un nombre supplémentaire de paramètres et peut être moins robuste à la généralisation sur des séquences de longueur différente de celles vues lors de l'entraînement.

**Encodage positionnel relatif [Shaw et al., 2018, Su et al., 2023].** Au lieu d'ajouter l'information de position absolue aux features, on peut modifier la formulation de l'attention pour prendre en compte les relations positionnelles relatives entre les éléments de la séquence. Plutôt que d'encoder explicitement la position, cette approche encode les distances entre les positions des paires d'éléments de la séquence, permettant ainsi au modèle de généraliser plus efficacement à des séquences de longueur variable.

$$Attention(Q, K, V) = softmax \left( \frac{Q(K + r_{ij})^T}{\sqrt{d_q}} \right) V, \quad (2.10)$$

Où  $r_{ij}$  représente le vecteur de distance relatif entre les positions  $i$  et  $j$ . Cette méthode est particulièrement efficace pour les tâches nécessitant une prise en compte fine de la structure des données, comme dans le traitement des spectres audio [Huang et al., 2018, Z. et al., 2023, Qi et al., 2023] ou des séquences biologiques [Senior et al., 2020, Jumper et al., 2021].

### 2.3.2 Limitations et variantes

Couche	RNN	Conv.	Att. originale	Reformer	Linformer	Att. lineaire
Complexité	$O(n \cdot d^2)$	$O(c \cdot n \cdot d^2)$	$O(n^2 \cdot d)$	$O(n \log(n))$	$O(n \cdot k \cdot d)$	$O(n \cdot d^2)$

TABLE 2.1 – **Tableau comparatif de la complexité des différentes versions de l'opération d'attention.**  $n$  est la longueur de la séquence,  $d$  la dimension des vecteurs de la séquence.  $c$  est la taille du noyau de convolution et  $k$  est le facteur de réduction de dimension pour Linformer.

Si l'on décompose l'opération d'attention (Eq 4.4), on observe que le produit  $QK^T$ , qui calcule le produit scalaire de chaque paire requête-clé, nécessite  $O(n^2 \cdot d_q)$  opérations. L'application du *softmax* sur chaque ligne de la matrice de scores coûte  $O(n^2)$  et le produit matriciel avec  $V$  coûte  $O(n^2 \cdot d_v)$  opérations. La complexité totale de l'opération est donc  $O(n^2 \cdot d_q + n^2 + n^2 \cdot d_v)$ . En général, la complexité est dominée par le terme  $O(n^2)$ , rendant l'attention [Vaswani et al., 2017] quadratique par rapport à la longueur de la séquence.

Cette complexité quadratique est un problème majeur parce qu'elle empêche l'architecture Transformer de s'adapter efficacement à des séquences longues et rend son entraînement coûteux en mémoire et en temps (Tableau 2.1). C'est pourquoi des méthodes alternatives ont été développées pour pallier ces problèmes [Katharopoulos et al., 2020, Wang et al., 2020, Vyas et al., 2020, Kitaev et al., 2020, Schlag et al., 2021, Chen et al., 2021, Dao et al., 2022]. Nous allons en étudier deux d'entre elles.

**L'informel** [Wang et al., 2020] démontre que l'opération d'auto-attention peut être approché par une matrice de faible rang. Ils exploitent cette propriété pour proposer une nouvelle version (Eq 2.11) qui réduit la complexité de l'attention classique en projetant les clés et les valeurs dans un espace de dimension réduite  $k$  (où  $k \ll n$ ) :

$$Attention(Q, K, V) = softmax \left( \frac{Q(EK)^T}{\sqrt{d_q}} \right) (EV), \quad (2.11)$$

où  $E$  est une matrice de projection de dimension  $k \times n$ , avec  $k$  beaucoup plus petit que  $n$ . Cette fois-ci, en décomposant l'opération, on note que  $EK$  et  $EV$  nécessitent respectivement  $O(n \cdot k \cdot d_k)$  et  $O(n \cdot k \cdot d_v)$  opérations. Le produit  $Q(EK)^T$  a une complexité de  $O(n \cdot k \cdot d_k)$  et le produit final avec  $EV$  coûte  $O(n \cdot k \cdot d_v)$ . On obtient donc une complexité finale de  $O(n \cdot k \cdot d_k + n \cdot k \cdot d_v) \approx O(n \cdot k)$ , ce qui est linéaire en  $n$  et dépend également de  $k$ , mais avec une réduction significative par rapport à l'approche softmax classique.

**L'attention linéaire** proposée dans [Katharopoulos et al., 2020] retire le noyau exponentiel de l'attention et formule l'auto-attention comme un produit scalaire à noyau linéaire (Eq : 2.12). De cette manière, ils utilisent la propriété d'associativité du produit matriciel pour réduire drastiquement la complexité :

$$\begin{aligned} Attention(Q, K, V) &= \phi(QK^T)V, \\ &= (\phi(Q)\phi(K)^T)V, \\ &= \phi(Q)(\phi(K)^TV), \end{aligned} \quad (2.12)$$

où  $\phi(x)$  est une fonction d'activation linéaire (ReLU ou ELU).

De cette manière, le produit  $\phi(K)^TV$  a un coût de  $O(n \cdot d_q \cdot d_v)$  et le produit avec  $\phi(Q)$  coûte  $O(n \cdot d_q \cdot d_v)$ . La complexité finale est donc de l'ordre de  $O(n)$ , linéaire par rapport à la longueur de la séquence  $n$ . C'est une amélioration significative par rapport à la complexité quadratique de l'attention softmax.

Malgré le fait que ces deux versions de l'attention permettent de passer à une complexité linéaire, elles ont également d'importantes limitations. La projection linéaire dans un espace de dimension réduite dans Eq 2.11 peut entraîner une perte d'information, ce qui peut affecter la performance du modèle, surtout pour des tâches nécessitant une attention très précise sur des éléments spécifiques de la séquence. L'attention linéaire (Eq 2.12), de son côté, repose sur une approximation qui peut ne pas capturer toutes les interactions complexes entre les éléments d'une séquence, ce qui peut dégrader les performances sur des tâches où ces interactions sont critiques. En pratique, l'attention originale de [Vaswani et al., 2017] reste celle qui offre les meilleures performances sur une plus grande variété de tâches et reste la plus utilisée.

La FlashAttention [Dao et al., 2022] peut également être considérée comme une amélioration intéressante de l'attention, même s'il s'agit plus d'une optimisation d'implémentation. Elle est conçue pour être "IO-aware", c'est-à-dire qu'elle prend en compte les lectures et écritures entre les niveaux de mémoire du GPU. Les auteurs utilisent un mécanisme de "tiling" pour réduire le nombre de lectures/écritures mémoire, ce qui permet de diminuer la complexité d'accès à la mémoire tout en maintenant la précision de l'attention.

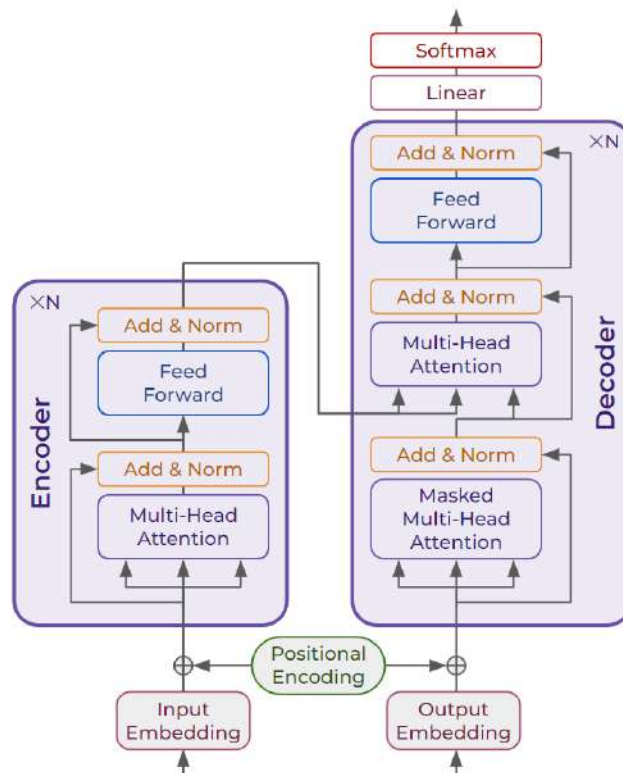


FIGURE 2.8 – **Architecture Transformer proposée dans [Vaswani et al., 2017].** Elle se compose d'une partie encodeur et d'une branche décodeur, utilisant toutes les deux le mécanisme d'attention. Une information de position est ajoutée aux entrées, car l'opération d'attention est invariante aux permutations.

### 2.3.3 L'architecture Transformer

En exploitant des mécanismes d'attention, l'architecture Transformer [Vaswani et al., 2017] a ouvert la voie à des modèles plus puissants et plus efficaces dans la capture des relations complexes au sein des séquences de texte. L'architecture Transformer (Figure 2.8) se compose de deux blocs principaux : l'encodeur (*encoder*) et le décodeur (*decoder*). Chacun de ces blocs est constitué de plusieurs couches successives identiques. L'encodeur, qui est l'élément central de notre discussion, est particulièrement important pour comprendre le fonctionnement interne de ce modèle.

Chaque encodeur est composé de deux sous-couches essentielles : un mécanisme d'**auto-attention multi-head** et un **perceptron multi-couches positionnel**. L'attention multi-head a été décrite en détail dans la section 2.3.1. Après la couche d'auto-attention, le résultat passe par un perceptron multi-couches, appliqué indépendamment à chaque position de la séquence. Ce réseau est composé de deux couches linéaires séparées par une activation ReLU. Cette étape est cruciale car elle permet au modèle de transformer les vecteurs de sortie en des représentations plus complexes et non linéaires, enrichissant ainsi la capacité du modèle à apprendre des relations complexes entre les éléments de la séquence d'entrée.

Enfin, deux mécanismes supplémentaires sont intégrés dans chaque sous-couche de l'encodeur : la **couche de normalisation** et les **connexions résiduelles**. La couche de normalisation est appliquée après le mécanisme de self-attention et après le réseau feed-forward pour stabiliser l'apprentissage et améliorer la convergence. Les connexions résiduelles, quant à elles, consistent à ajouter l'entrée d'origine de chaque sous-couche à sa sortie, ce qui aide à atténuer le problème de l'évanouissement du gradient et à assurer une meilleure propagation des



gradients pendant l'apprentissage.

Le décodeur, quant à lui, ressemble beaucoup à l'encodeur, mais ajoute une couche d'attention-croisée pour intégrer l'information de l'encodeur. Étant peu utilisé dans les applications de vision par ordinateur, nous ne détaillerons pas davantage son architecture.

### 2.3.4 L'attention en vision par ordinateur

Les succès spectaculaires du mécanisme d'attention dans le domaine du traitement du langage naturel (NLP) ont naturellement suscité l'intérêt des chercheurs en vision par ordinateur. Après avoir démontré sa capacité à capter des dépendances à longue portée et à améliorer considérablement les performances des modèles de langage, la question s'est posée de savoir si ces mêmes gains pouvaient être obtenus dans le domaine de la vision. Cependant, appliquer l'attention à des images présente des défis uniques. Contrairement aux séquences de mots dans le NLP, une image est une structure bidimensionnelle dense, composée de milliers voire de millions de pixels. Traiter une image comme une simple séquence de pixels n'est donc pas trivial. Cette section explore comment l'attention a été adaptée pour répondre à ces défis spécifiques et comment elle a transformé les approches classiques de la vision par ordinateur.

#### 2.3.4.1 Origines et problématiques

L'attention visuelle est un concept étudié depuis des décennies dans le cadre de la vision. Inspirée du système cognitif humain [Broadbent, 1958, Treisman, 1964, Treisman and Gelade, 1980], l'attention visuelle imite la capacité cognitive humaine à capturer des informations spécifiques, amplifiant les détails critiques pour se concentrer davantage sur les aspects essentiels des données. Ce concept a été repris par la communauté de vision par ordinateur [Itti and Koch, 2001] pour construire des systèmes capables d'extraire l'information importante d'une image afin de l'utiliser pour des tâches comme la détection d'objets, la segmentation d'images et la reconnaissance de scènes. Ils utilisent souvent le concept de saillance visuelle [Itti et al., 1998, Treue, 2003], qui fait référence aux caractéristiques d'une scène visuelle qui attirent automatiquement l'attention d'un observateur, pour guider l'attention. L'auto-attention en machine learning, que nous avons détaillée dans les sections précédentes, partage beaucoup de similarités avec ce concept. En effet, l'auto-attention de [Vaswani et al., 2017] appliquée aux pixels d'une image peut être vue comme une généralisation apprise de l'attention visuelle, où chaque pixel peut apprendre ses régions d'intérêt. Chaque ligne de nos cartes d'attention (Eq 2.5) peut alors être vue comme une carte de saillance, et avec plusieurs *heads*, chaque pixel peut extraire plusieurs cartes de saillance.

Cependant, alors que la motivation de l'attention visuelle en neuroscience est de filtrer les données perçues pour ne garder que l'information importante, réduisant ainsi la complexité de l'espace d'entrée, l'auto-attention utilisée en machine learning ne suit pas ce principe. En effet, chaque pixel peut déterminer ses propres régions d'intérêt, mais l'auto-attention n'a pas de mécanisme global pour identifier et se limiter aux régions importantes de l'image. Cela nous amène à la principale problématique de l'utilisation de l'attention de [Vaswani et al., 2017] en vision par ordinateur : sa complexité en  $O(n^2)$  (section 2.3.2). Alors que, dans le NLP, les séquences ont des longueurs pouvant aller jusqu'à des dizaines de milliers d'éléments, une image HD contient environ  $10^6$  pixels. Traiter une image comme une séquence de pixels et utiliser l'architecture Transformer est donc impraticable, car le calcul des cartes d'attention pourrait nécessiter plusieurs téraoctets de mémoire pour tout le réseau.

D'un autre côté, l'auto-attention en vision par ordinateur promet une capacité à capturer des dépendances globales entre différentes parties d'une image et à pondérer dynamiquement

l'importance des régions de l'image en fonction de la tâche. De nombreux travaux ont cherché à utiliser le mécanisme d'attention pour différentes tâches de vision tout en maintenant une complexité raisonnable [Hassanin et al., 2024]. La section suivante abordera le fonctionnement de certaines de ces architectures.

### 2.3.4.2 Architecture Transformer pour les images

L'incorporation du mécanisme d'attention dans la vision par ordinateur a initié une transformation majeure dans la manière dont les modèles traitent les images, permettant une capture plus riche et plus flexible des relations spatiales au sein des données visuelles. Cette révolution a été amorcée par l'introduction de modèles comme DETR (DEtection TRansformers) [Carion et al., 2020] et Vision Transformer (ViT) [Dosovitskiy et al., 2020] en 2020, qui ont démontré le potentiel des Transformers, initialement développés pour le traitement du langage naturel, dans le domaine de la vision. DETR a redéfini la détection d'objets en remplaçant les pipelines complexes traditionnels par un modèle end-to-end utilisant l'attention pour modéliser directement les relations entre les objets dans une image, simplifiant ainsi le processus tout en améliorant la précision. Parallèlement, ViT a adapté le mécanisme d'attention aux tâches de classification d'images en traitant les images comme des séquences de patches, et a montré que les Transformers pouvaient surpasser les performances des CNN traditionnels en exploitant efficacement les relations globales dans les images.

En 2021, l'évolution de ces concepts a conduit à l'émergence de modèles plus spécialisés et optimisés, comme le Swin Transformer [Liu et al., 2021] et le Twins Transformer [Chu et al., 2021], qui ont poussé plus loin les capacités des Transformers en vision par ordinateur. Le Swin Transformer a introduit une hiérarchie d'attention basée sur des fenêtres glissantes, permettant une gestion plus efficace des images à haute résolution et une meilleure performance sur des tâches variées comme la segmentation et la détection d'objets. De manière complémentaire, le Twins Transformer a combiné l'attention globale et locale dans une architecture parallèle, capturant à la fois les détails fins et les informations globales dans les images, améliorant ainsi les résultats sur les benchmarks de classification et de segmentation.

En parallèle, d'autres modèles comme Perceiver [Jaegle et al., 2021] et Perceiver IO [Jaegle et al., 2022] ont cherché à généraliser l'usage des Transformers au-delà des tâches de vision, en proposant une architecture flexible capable de traiter des données de formes et de tailles variées. Ces modèles ont été conçus pour être plus efficaces en termes de calcul, grâce à l'utilisation d'un espace latent, tout en conservant les avantages du mécanisme d'attention, ce qui les rend particulièrement adaptés aux applications nécessitant une grande échelle et une flexibilité des données. De même, LambdaNetworks [Bello, 2021], introduit en 2021, a offert une alternative aux Transformers en proposant une couche d'attention plus légère, appelée "*lambda layer*", qui capture les interactions globales avec une complexité computationnelle réduite, tout en maintenant des performances compétitives sur les tâches de vision.

L'année 2023 a vu l'apparition du SAM (Segment Anything Model) [Kirillov et al., 2023], un modèle de segmentation universel qui repousse les limites de la segmentation d'images en exploitant le mécanisme d'attention pour segmenter automatiquement n'importe quel objet, sans nécessiter d'entraînement spécifique pour chaque tâche. SAM représente un pas en avant vers la généralisation des modèles d'attention dans des applications pratiques de vision par ordinateur, illustrant le potentiel de ces modèles pour des tâches variées et complexes.

Enfin, des modèles hybrides comme le CvT (Convolutional Vision Transformer) [Wu et al., 2021a] ont cherché à combiner les forces des convolutions et des Transformers, en introduisant des couches convolutionnelles dans l'architecture Transformer pour traiter plus efficacement les images à haute résolution tout en conservant les bénéfices du mécanisme d'attention. Ces



développements montrent comment les modèles d'attention en vision par ordinateur continuent de se diversifier, améliorant continuellement les capacités de traitement des images et ouvrant de nouvelles perspectives pour les applications futures.

## 2.4 Relation entre l'attention et la mise en correspondance

### 2.4.1 L'attention comme opération de communication entre les descripteurs

Le mécanisme d'attention est devenu un outil central dans les architectures de réseaux neuronaux pour la mise en correspondance d'images, apportant une nouvelle dimension à cette tâche en améliorant la précision et la robustesse des correspondances. Le rôle de l'attention dans le matching repose sur sa capacité à pondérer les relations entre les éléments des images en fonction de leur importance, permettant ainsi de créer des correspondances plus fines et plus discriminantes. On exploite les deux types d'attention : l'auto-attention et l'attention-croisée, qui remplissent des rôles complémentaires pour améliorer la qualité des correspondances.

L'attention-croisée permet de créer une communication entre deux images en produisant des représentations qui intègrent l'information des deux vues. Concrètement, cela se traduit par l'utilisation des descripteurs issus de l'image source pour construire les requêtes ( $Q$ ) et des descripteurs issus de l'image cible pour construire les clés ( $K$ ). Le produit des matrices  $Q$  et  $K$  permet de quantifier la similarité entre les descripteurs de chaque image dans différents espaces, appris grâce à  $W_q$  et  $W_k$  (Eq 4.4). Les cartes d'attention pour chaque head peuvent être considérées comme différentes cartes de correspondances. Cette caractéristique de l'attention améliore la précision du matching, car les descripteurs sont enrichis par l'information provenant des deux images, et les correspondances ambiguës sont mieux résolues.

L'auto-attention, quant à elle, crée une communication au sein d'une même image, permettant aux descripteurs de prendre en compte les relations globales à travers toute la scène. Dans le cadre de la mise en correspondance, cela est particulièrement utile pour réduire les ambiguïtés entre des descripteurs similaires ou peu informatifs, en permettant à chaque descripteur de "voir" les autres descripteurs de l'image et de moduler son importance en conséquence. L'auto-attention peut ainsi renforcer la discrimination des descripteurs, rendant les correspondances avec l'autre image plus fiables et robustes. Cette capacité à améliorer la représentation interne des images est particulièrement précieuse dans les scènes complexes, où des zones peu texturées ou des motifs répétitifs peuvent entraîner des erreurs de correspondance.

L'utilisation combinée de l'attention-croisée et de l'auto-attention permet de créer des représentations très riches et expressives, où chaque descripteur bénéficie à la fois d'une connaissance de son propre contexte (auto-attention) et d'une compréhension directe de sa relation avec les descripteurs de l'autre image (attention-croisée). Ainsi, le mécanisme d'attention transforme la manière d'aborder le matching d'images en créant des descripteurs globaux, plutôt que locaux comme dans les CNN, tout en établissant une communication entre les deux images. Son utilisation améliore grandement la performance globale des méthodes de mise en correspondance d'images. Nous allons à présent détailler son intégration dans certaines méthodes.

**Attention entre descripteurs locaux.** Dans la section 2.2.2.1, nous avons vu que la mise en correspondance d'images peut être décomposée en trois étapes : la détection de points d'intérêt, la description locale de ces points et la mise en correspondance. **SuperGlue** [Sarlin et al., 2020] combine des approches classiques de détection et de description de points d'intérêt, comme

Superpoint [DeTone et al., 2018], et intègre un mécanisme d'attention pour améliorer la communication entre les descripteurs locaux. Après la détection initiale, un encodage positionnel est ajouté aux descripteurs pour tenir compte de leur emplacement dans l'image. Un graphe est ensuite construit pour modéliser les relations entre les points d'intérêt, et des blocs Transformers sont utilisés pour la communication intra-image via de l'auto-attention. Cette étape permet aux descripteurs d'échanger des informations au sein de la même image. Des blocs Transformer avec attention-croisée sont ensuite appliqués entre la source et la cible, puis entre la cible et la source, facilitant ainsi l'appariement des points d'intérêt. Enfin, des cartes de correspondances sont calculées pour déterminer les correspondances finales. Des améliorations, comme **LightGlue** [Lindenberger et al., 2023], optimisent cette approche en ajoutant un mécanisme de sélection progressive des correspondances matchables, permettant ainsi d'éliminer efficacement les outliers (correspondances incorrectes) et de réduire le coût computationnel, tout en maintenant des performances robustes. Cette approche permet de mieux gérer la complexité du calcul des correspondances dans des environnements variés, tout en restant efficace dans des contextes où les ressources sont limitées.

**Attention entre des cartes de descripteurs basse résolution.** Comme nous l'avons vu dans la section 2.2.2.2, le paradigme semi-dense nous permet de mettre en correspondance des représentations basse résolution des images, puis de raffiner ces prédictions. Dans ce paradigme, les descripteurs sont locaux, car leur description est restreinte au champ récepteur du réseau de convolution.

**LoFTR** (*Local Feature Transformer*) [Sun et al., 2021] reprend et améliore le concept introduit par SuperGlue pour le paradigme semi-dense, en utilisant des modules d'auto-attention et d'attention-croisée pour favoriser la communication à la fois intra-image et inter-image. Un CNN est utilisé pour extraire des descripteurs denses à basse résolution ( $\frac{1}{8}$  de la résolution d'origine) et à haute résolution ( $\frac{1}{2}$  de la résolution d'origine) pour les deux images. Comme dans SuperGlue, une succession d'auto-attention et d'attention-croisée est ensuite appliquée sur ces descripteurs. Cependant, le nombre de descripteurs est plus important que dans le paradigme S2S, les auteurs optent donc pour de l'attention linéaire [Katharopoulos et al., 2020] plutôt que pour de l'attention classique [Vaswani et al., 2017]. Une fois les descripteurs raffinés par les couches Transformer, les correspondances grossières sont extraites en calculant la carte de correspondances entre la source et la cible. Les matchs grossiers sont ensuite affinés par un module utilisant de l'attention sur des patches extraits autour des correspondances grossières. Grâce à l'utilisation de l'attention, LoFTR parvient à modéliser les relations entre les pixels de manière plus globale, améliorant la précision de son matching dans des régions non texturées et sa robustesse face à des transformations complexes. Cette flexibilité conduit à une amélioration significative des performances dans des tâches comme l'estimation de pose, faisant de LoFTR une méthode de référence dans le domaine.

De nombreuses autres méthodes basées sur LoFTR émergent, chacune se focalisant sur certaines problématiques. **QuadTree Attention** [Tang et al., 2022b] adapte l'attention pour traiter les images de manière hiérarchique en utilisant une structure en arbre, permettant une attention plus efficace en se concentrant d'abord sur les grandes régions, puis sur les sous-régions, ce qui réduit le coût computationnel. **ASpanFormer** [Chen et al., 2022], quant à lui, utilise un mécanisme d'attention à portée adaptative, ajustant dynamiquement la granularité des correspondances, toujours sans détecteur explicite de points-clés, ce qui lui permet de mieux capturer les relations locales et globales selon le contexte. **MatchFormer** [Wang et al., 2022] suit une stratégie de raffinement progressif, où les correspondances sont réévaluées et améliorées à plusieurs étapes. Cela permet de corriger les erreurs au fur et à mesure et de capturer des détails

fins dans les correspondances, là où LoFTR pourrait manquer de précision en raison de son approche plus directe. Enfin, **3DG-STFM** [Mao et al., 2022] utilise un modèle "*student-teacher*" guidé par des informations géométriques 3D, permettant de maintenir la cohérence géométrique tout en établissant des correspondances semi-denses, particulièrement adaptées aux tâches de reconstruction et de suivi 3D. Ces approches partagent l'objectif d'améliorer la qualité des correspondances tout en réduisant la complexité computationnelle, en s'appuyant sur l'attention pour optimiser le processus de matching.

**Matching dense.** Utiliser l'attention pour le matching dense est difficile, principalement en raison de la complexité computationnelle et des exigences en mémoire du mécanisme d'attention, qui augmentent de manière quadratique avec la taille des images. Établir des correspondances pour chaque pixel représente donc un défi, et peu de méthodes denses parviennent à intégrer de l'attention jusqu'à la résolution d'origine. On peut noter **PMatch** [Zhu and Liu, 2023], basé sur le matching grossier de LoFTR, qui combine une approche de modélisation d'image masquée avec une construction progressive des correspondances. PMatch apprend à reconstruire les informations manquantes (masquées) et utilise un mécanisme de raffinement local pour améliorer les régions mises en correspondance par LoFTR. Cependant, dans cette méthode, l'attention n'est utilisée qu'au niveau grossier. **COTR** (*CORrespondence TRansformer*) [Jiang et al., 2021], en revanche, utilise de l'attention même à pleine résolution en appliquant un raffinement multi-échelle par zoom. Lors de l'inférence, le modèle commence par trouver des correspondances approximatives à une échelle réduite, puis affine ces correspondances en "zoomant" sur des régions spécifiques des images. À chaque étape, des patches sont extraits autour des correspondances trouvées, et l'algorithme réapplique l'attention pour affiner davantage les correspondances à des résolutions plus élevées. Le problème est que les correspondances sont obtenues séquentiellement, ce qui rend l'algorithme très long pour obtenir toutes les correspondances (plusieurs minutes par paire d'images). **Eco-TR** [Tan et al., 2022a] propose de réduire ce coût calculatoire en sauvegardant en mémoire une partie du modèle commune aux différents zooms, mais cette méthode reste significativement plus lente que les méthodes semi-denses comme LoFTR.

L'utilisation du mécanisme d'attention dans les réseaux de mise en correspondance d'images a ouvert de nouvelles perspectives. En facilitant la communication entre les descripteurs des deux images à apparier, ce mécanisme permet de mieux comprendre le contexte global de la scène et d'identifier des correspondances pertinentes, même dans des régions uniformes ou peu texturées. Cela réduit significativement les erreurs causées par des variations d'échelle, de point de vue ou d'illumination.

## 2.4.2 Analyse de l'attention pour la mise en correspondance

Dans cette section, nous proposons une analyse du mécanisme d'attention dans le cadre particulier de son utilisation dans les réseaux de mise en correspondance. La motivation derrière cette étude est de mieux comprendre comment les mécanismes d'auto-attention (AA) et d'attention-croisée (AC) capturent et relient l'information entre deux images, et comment ils contribuent à établir des correspondances robustes.

**Cadre expérimental.** L'objectif ici n'est pas de concevoir une architecture avec des performances comparables aux meilleures méthodes de l'état de l'art, mais plutôt de permettre un cadre simple pour l'analyse de l'attention. Nous allons donc considérer deux architectures (illustrées dans la Figure 2.9) : une architecture siamoise entièrement convolutionnelle, ainsi qu'une

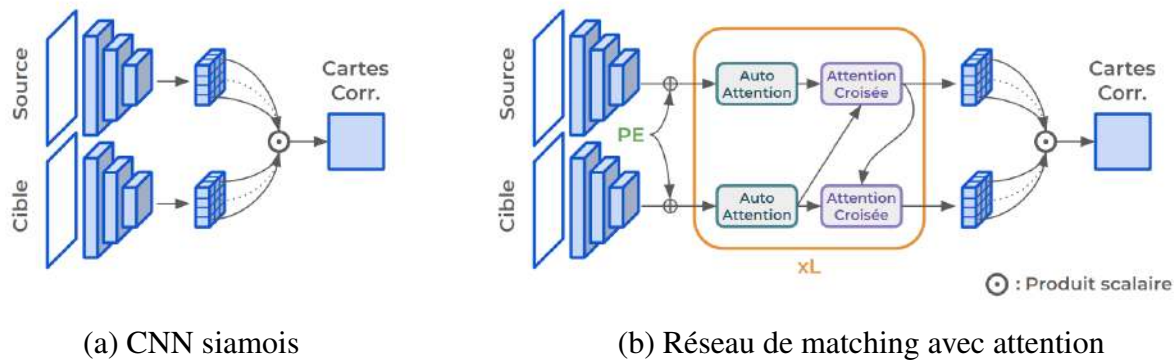


FIGURE 2.9 – Architectures utilisées pour notre étude de l'attention dans les réseaux de mise en correspondance. (a) Réseau siamois n'utilisant que des couches de convolution. (b) Même architecture avec l'ajout de couches d'auto-attention et d'attention-croisée.

version à laquelle on ajoute des couches d'attention. Le CNN utilisé est un ResNet-18 [He et al., 2015] construisant une pyramide de cartes de descripteurs multi-résolution, où la résolution la plus grossière est à  $\frac{1}{16}$ <sup>ème</sup> de la résolution d'origine. Comme nous cherchons uniquement à analyser le mécanisme d'attention, nous effectuons la mise en correspondance à ce niveau le plus grossier, sans chercher à affiner les prédictions par la suite. Pour l'architecture avec attention, nous utilisons le même ResNet-18, auquel on ajoute  $L$  couches de communication traitant les cartes de descripteurs grossières des images. Chacune de ces couches est composée de 4 blocs transformer (voir section 2.3.3) :

- Un bloc contenant une couche d'**auto-attention sur la source**.  $Q$ ,  $K$  et  $V$  sont alors des projections des descripteurs de la source, et les cartes d'attention  $S$  (voir équation 2.5) modélisent les relations entre les descripteurs de la source. L'objectif de cette couche est de permettre la communication entre les descripteurs de l'image source.
- Un bloc contenant une couche d'**auto-attention sur la cible**.  $Q$ ,  $K$  et  $V$  sont alors des projections des descripteurs de la cible, et les cartes d'attention  $S$  (voir équation 2.5) modélisent les relations entre les descripteurs de la cible. L'objectif de cette couche est de permettre la communication entre les descripteurs de l'image cible.
- Un bloc contenant une couche d'**attention-croisée entre la source et la cible**. Ici,  $Q$  est une projection de la source alors que  $K$  et  $V$  sont des projections de la cible. Les cartes d'attention modélisent donc, pour chaque descripteur de la source, sa relation avec tous les descripteurs de la cible. Les descripteurs de la source sont ensuite mis à jour en utilisant les poids d'attention pour pondérer l'information de la cible contenue dans  $V$  (voir équation 2.6). L'objectif de cette couche est donc de permettre la communication de la cible vers la source.
- Un bloc contenant une couche d'**attention-croisée entre la cible et la source**. Ici,  $Q$  est une projection de la cible alors que  $K$  et  $V$  sont des projections de la source. Les cartes d'attention modélisent donc, pour chaque descripteur de la cible, sa relation avec tous les descripteurs de la source. Les descripteurs de la cible sont ensuite mis à jour en utilisant les poids d'attention pour pondérer l'information de la source contenue dans  $V$  (voir équation 2.6). L'objectif de cette couche est donc de permettre la communication de la source vers la cible.

Par défaut, cette architecture est composée de  $L = 4$  couches de communication, et chaque opération d'attention est réalisée sur 8 heads. Toutes les variantes présentées ci-dessous sont entraînées pendant 24 heures sur MegaDepth [Li and Snavely, 2018] en utilisant une entropie

croisée sur les cartes de correspondances comme fonction de coût. Les images d'entraînement et d'évaluation sont redimensionnées pour que leur côté le plus grand soit égal à 640 pixels. Les résultats quantitatifs sont calculés sur l'ensemble d'évaluation MegaDepth-1500, composé de deux scènes non vues pendant l'entraînement. Ces résultats sont présentés sous la forme de courbes cumulatives de précision de matching à différents seuils d'erreur en pixels (voir section 2.2.1.2).

### 2.4.2.1 L'attention comme opération de communication

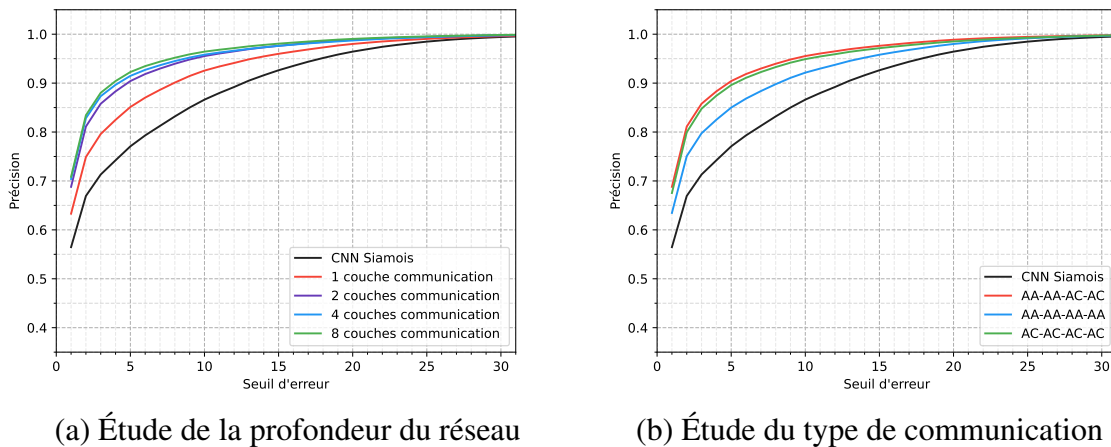


FIGURE 2.10 – **Étude de l'ajout d'opérations d'attention à un réseau siamois.** (a) Impact de la quantité de couches dédiées à la communication. (b) Types d'attention utilisés dans chaque couche de communication.

Dans un premier temps, nous nous intéressons à l'ajout de ces couches de communication composées d'auto-attention (AA) et d'attention-croisée (AC). Nous entraînons différents réseaux : un réseau siamois sans attention, donc sans communication entre la source et la cible, ainsi que quatre réseaux avec  $L = 1$ ,  $L = 2$ ,  $L = 4$  et  $L = 8$  couches de communication. Ces résultats sont présentés dans la Figure 2.10 (a). On peut y voir que même l'ajout d'une seule couche de communication, contenant 2 couches d'auto-attention et 2 couches d'attention-croisée, permet d'améliorer significativement les performances de mise en correspondance d'un réseau siamois. En effet, la taille du champ réceptif de notre ResNet-18 est de 235 pixels ; les descripteurs grossiers du réseau siamois ne peuvent donc pas encoder l'information globale de l'image et peuvent avoir des difficultés à distinguer des structures répétitives, localement similaires, mais distantes dans l'image. L'ajout d'une seule couche d'auto-attention permet d'étendre le champ réceptif du réseau à toute l'image, car chaque descripteur local produit par le ResNet-18 est mis à jour par une combinaison des informations de tous les descripteurs de l'image. De plus, le réseau siamois produit des descripteurs pour la source et pour la cible de manière indépendante, ce qui peut poser problème dans le cas de forts changements de perspective où le seul contexte d'une image ne permet pas de construire des descripteurs suffisamment représentatifs et discriminants. En ajoutant des couches d'attention-croisée, la construction des descripteurs peut prendre en compte le contexte global des deux images. Le réseau peut donc construire un descripteur pour une région de l'image source en fonction de l'information de l'image cible représentant la même scène 3D sous un point de vue différent. Créer cette interaction entre les deux images dans les réseaux de mise en correspondance est crucial, car elle permet au réseau de construire un *raisonnement* sur la structure 3D de la scène observée et de mieux gérer les



changements de points de vue importants ou les occultations partielles. Dans la Figure 2.10 (a), on peut voir que plus on ajoute de *couches de communication*, plus le réseau parvient à produire des correspondances précises. Cela vient certainement du fait qu'en ajoutant des couches de communication, le réseau dispose d'une plus grande capacité pour construire des représentations de la source et de la cible intégrant le contexte global de la paire d'images.

Les couches d'auto-attention permettent de créer une communication intra-image et les couches d'attention-croisée une communication inter-images, mais quelle information ces couches utilisent-elles dans le cadre de la mise en correspondance ? Commençons par les couches d'attention-croisée et la communication inter-images. Dans les couches où l'image source est utilisée pour construire les requêtes ( $Q$ ) et l'image cible pour construire les clés ( $K$ ), les différentes cartes d'attention représentent alors l'importance des différentes régions de la cible dans la mise à jour des descripteurs de la source. Dans la Figure 2.11, nous visualisons, pour un point de la source, ces cartes d'attention pour différentes heads et pour différentes *couches de communication* dans un réseau où  $L = 4$ .

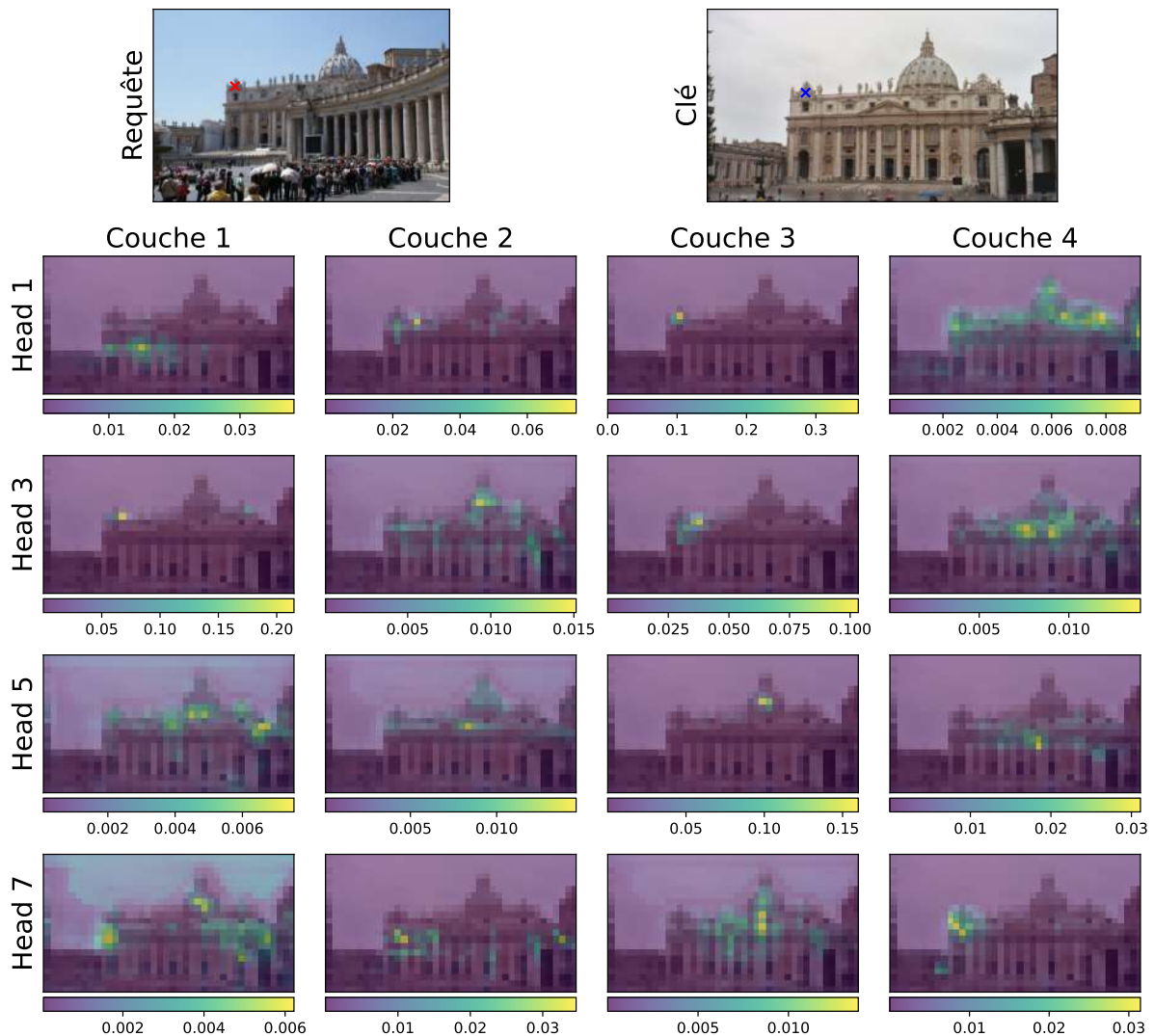


FIGURE 2.11 – Visualisation des cartes d'attention-croisée entre la source et la cible. La croix rouge (×) correspond au point requête pour lequel nous visualisons les cartes d'attention. Pour information, nous notons d'une croix bleue (×) son correspondant dans l'image cible.



La Figure 2.11 nous montre dans un premier temps que l'attention-croisée n'utilise pas uniquement l'information locale autour du correspondant, mais utilise le contexte général de l'image cible. On note également que chaque head semble se concentrer sur des régions bien spécifiques de l'image cible, et que même au sein d'une même couche, les heads offrent une grande diversité de régions considérées comme importantes. Par exemple, dans la Figure 2.11, la head 1 de la couche 2 et la head 7 de la couche 4 se concentrent sur la région autour du correspondant, alors que la head 7 de la couche 1 ou la head 1 de la couche 4 semblent se concentrer sur les structures répétitives causées par la symétrie du bâtiment. Dans la Figure 2.12, on observe les mêmes comportements dans les couches d'auto-attention. Chaque head capture un aspect spécifique de l'image, et la diversité des heads permet d'encoder le contexte général de l'image source.

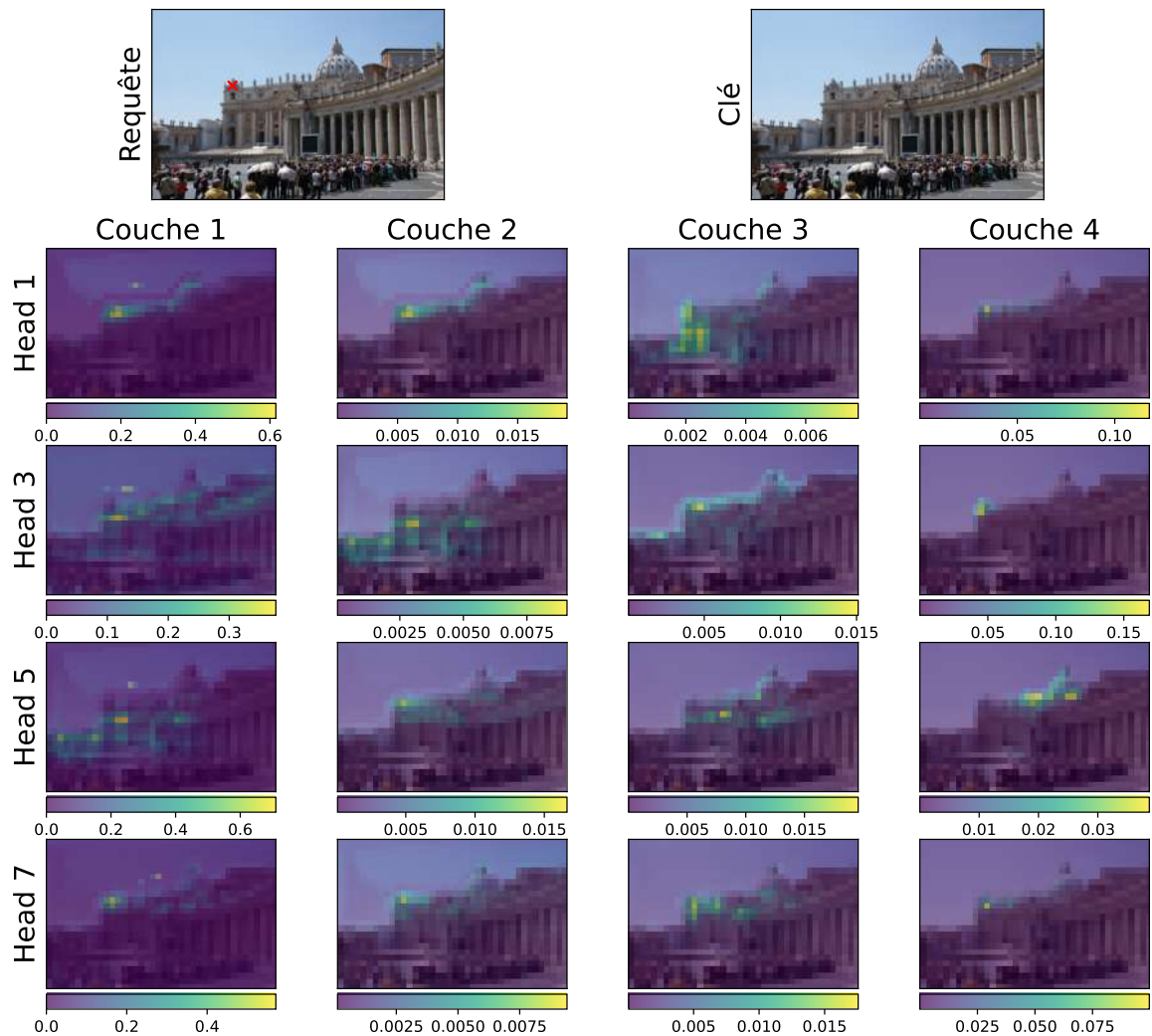


FIGURE 2.12 – **Visualisation des cartes d'auto-attention sur la source.** La croix rouge (×) correspond au point requête pour lequel nous visualisons les cartes d'attention.

Dans les visualisations des cartes d'attention, il est difficile de percevoir une différence entre les différentes couches du réseau. Pour évaluer de manière quantitative comment le mécanisme d'attention se répartit spatialement dans une image, les auteurs du *Vision Transformer* (ViT) [Dosovitskiy et al., 2020] proposent de calculer la distance moyenne d'attention (*Mean Attention Distance* ou MAD). Dans une couche d'auto-attention à  $H$  heads calculant des cartes d'attention  $\{S^h\}_{h=1\dots H}$ , la MAD mesure la moyenne des distances entre chaque pixel, pondérée par le poids d'attention :

$$\text{MAD} = \frac{1}{H} \sum_{h=1}^H \frac{\sum_{i=1}^N \sum_{j=1}^N S_{i,j}^h \cdot d_{i,j}}{\sum_{i=1}^N \sum_{j=1}^N S_{i,j}^h}, \quad (2.13)$$

où  $N$  est le nombre de pixels de l'image et  $d_{i,j}$  la distance entre les positions  $i$  et  $j$ . Une MAD faible signifie que l'attention est concentrée autour du point de référence, alors qu'une MAD élevée indique une attention plus dispersée dans l'image. Nous proposons de modifier cette métrique en calculant également la distance moyenne au correspondant (Mean Correspondent Distance ou MCD) dans les couches d'attention-croisée :

$$\text{MCD} = \frac{1}{H} \sum_{h=1}^H \frac{\sum_{i=1}^N \sum_{j=1}^M S_{i,j}^h \cdot d_{c_i,j}}{\sum_{i=1}^N \sum_{j=1}^M S_{i,j}^h}, \quad (2.14)$$

où  $N$  et  $M$  sont respectivement le nombre de pixels dans l'image requête et dans l'image clé, et  $d_{c_i,j}$  est la distance dans l'image entre le correspondant  $c_i$  du pixel requête  $i$  et le pixel  $j$ . Une MCD faible signifie que l'attention-croisée est concentrée autour de la correspondance, alors qu'une MCD élevée indique une attention plus dispersée dans l'image, capturant le contexte. Dans la Figure 2.13, nous calculons la MAD et la MCD des couches d'attention d'un réseau où  $L = 8$ . Les résultats sont moyennés sur 10 scènes d'évaluation de MegaDepth, et nous rapportons également l'écart type entre les différentes heads.

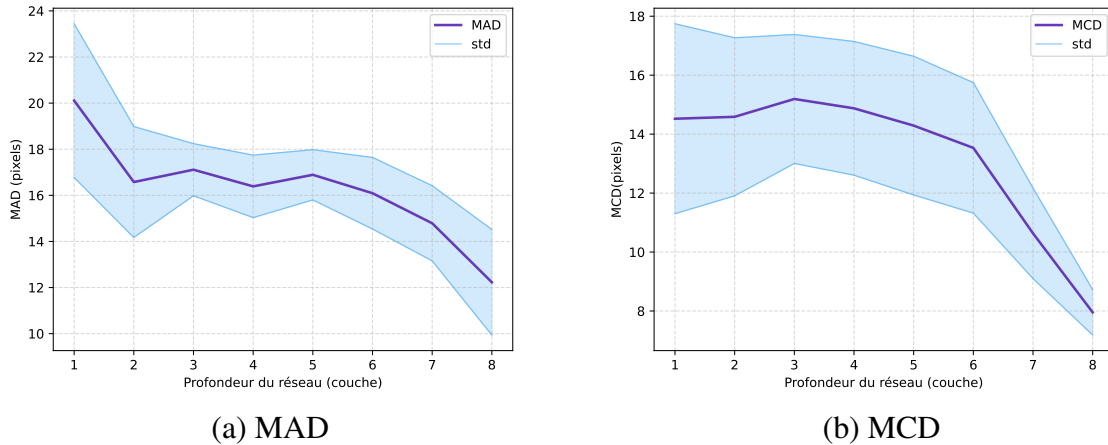


FIGURE 2.13 – (a) Distance moyenne d'attention (MAD) dans les cartes d'auto-attention. (b) Distance moyenne au correspondant (MCD) dans les cartes d'attention-croisée.

La Figure 2.13 montre que les attentions des premières couches de notre modèle de mise en correspondance n'ont pas la même dispersion que celles des dernières couches. Que ce soit pour l'auto-attention ou l'attention-croisée, les premières couches dispersent l'attention dans l'image, créant ainsi une communication entre des régions distantes et capturant le contexte général des images. La dispersion de l'attention diminue ensuite avec la profondeur du réseau, permettant une communication avec des régions locales et encodant ainsi les détails nécessaires pour une mise en correspondance précise. On peut également noter que la variance de la dispersion dans les couches d'attention-croisée varie significativement entre les couches.

En analysant les cartes d'attention et la dispersion de l'attention, on remarque que l'auto-attention et l'attention-croisée présentent des comportements similaires, mais leurs rôles dans un réseau de mise en correspondance sont bien distincts. Pour évaluer leur importance respective, nous comparons dans la Figure 2.10 (b) un modèle utilisant une combinaison d'auto-attention et d'attention-croisée avec des modèles utilisant uniquement de l'auto-attention et

uniquement de l'attention-croisée. On observe que l'attention-croisée est indispensable pour de bonnes performances de mise en correspondance. En effet, sans ces couches, le modèle construit des représentations indépendantes pour la source et la cible et ne peut donc pas intégrer le contexte global de la paire d'images. On note également que doubler l'attention-croisée sans utiliser d'auto-attention permet d'obtenir de bonnes performances de mise en correspondance. Cela peut s'expliquer par le fait que le CNN prend en charge en partie la communication intra-image de manière locale et que l'enchaînement d'un grand nombre de couches d'attention-croisée permet aussi une communication intra-image indirecte. Toutefois, les meilleures performances sont atteintes en combinant l'attention-croisée avec l'auto-attention.

Les analyses précédentes nous permettent de mieux comprendre le rôle et l'impact de l'attention dans la communication entre une paire d'images, mais quel effet a-t-elle sur nos cartes de correspondances ? Pour rappel, la carte de correspondance  $C_i$  pour une position  $i$  de l'image source mesure la similarité entre son descripteur  $\mathbf{h}_{S,i}$  et l'ensemble des descripteurs de l'image cible  $\{\mathbf{h}_{T,j}\}_{j=1\dots M}$  :

$$C_i = \text{softmax}(\mathbf{h}_{S,i} \cdot [\mathbf{h}_{T,1} \quad \mathbf{h}_{T,2} \quad \dots \quad \mathbf{h}_{T,M}]), \quad (2.15)$$

où la fonction softmax permet d'obtenir une distribution de probabilité. Le correspondant de la position  $i$  dans la source sera alors l'argmax de  $C_i$ . Pour visualiser l'impact de l'attention sur les cartes de correspondances, nous calculons ces dernières à différents niveaux du réseau : en utilisant les descripteurs issus du CNN, puis en utilisant les descripteurs après chaque couche de communication. Les cartes sont présentées dans la Figure 2.14. On peut y voir qu'après le CNN, la carte de correspondance est ambiguë, avec des valeurs élevées pour de nombreux points autour du correspondant. Après une seule couche de communication, l'attention permet de fortement diminuer l'ambiguïté dans cette région, et après les 4 couches, le descripteur du correspondant est presque le seul à avoir un produit scalaire élevé. Les opérations d'auto-attention et d'attention-croisée apportent une communication qui intègre l'information contextuelle des deux images, permettant aux descripteurs d'être discriminants les uns des autres et facilitant ainsi la mise en correspondance.

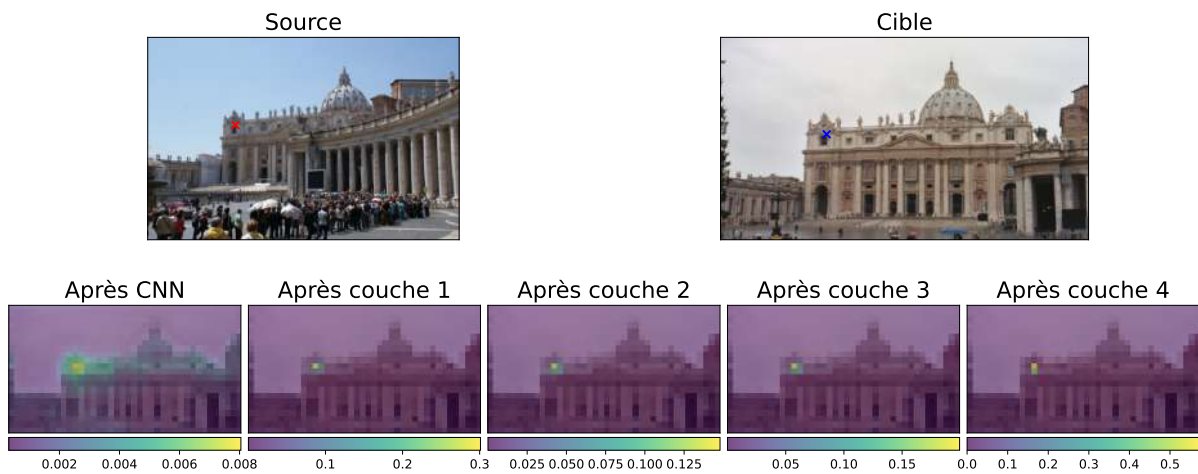
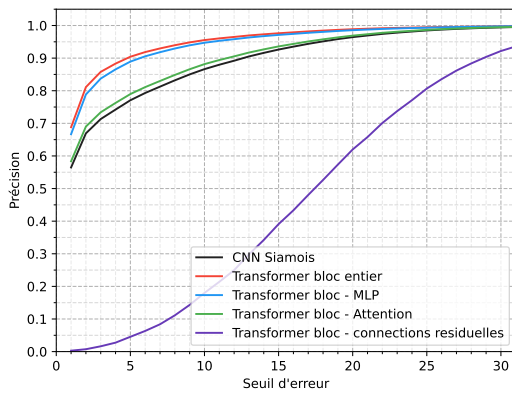


FIGURE 2.14 – **Visualisation des cartes de correspondances à différents niveaux du réseau.** La croix rouge (×) correspond au point requête pour lequel nous visualisons les cartes de correspondances. La croix bleue (×) est le correspondant de vérité terrain du point requête.

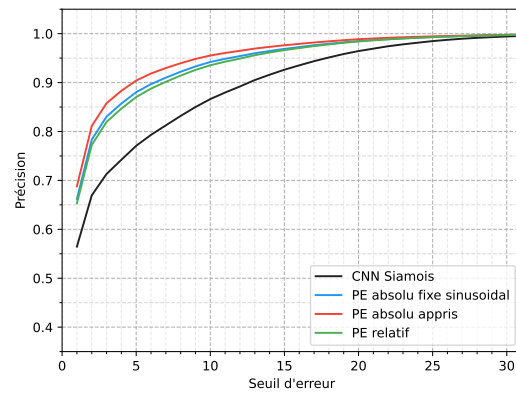
### 2.4.2.2 Ablation de l'attention

Nous terminons cette étude par deux analyses plus générales de l'architecture transformer, toujours dans le cadre de la mise en correspondance d'images.

Dans un premier temps, nous nous intéressons à la composition d'un bloc transformer, décrit dans la section 2.3.3. Nous entraînons différents réseaux en retirant soit le perceptron multi-couches (MLP), soit l'opération d'attention, soit les connexions résiduelles pour mesurer leur impact sur les performances de mise en correspondance. La Figure 2.15 (a) montre que retirer l'opération d'attention réduit les performances à celles d'un réseau siamois, car plus aucune communication n'est permise entre les deux images. En revanche, retirer le perceptron multi-couches diminue très peu les performances du réseau. Les connexions résiduelles, quant à elles, sont indispensables car, d'une part, elles permettent de conserver l'information présente avant le bloc transformer, et, d'autre part, elles permettent d'éviter le problème de l'atténuation du gradient lors de la rétropropagation.



(a) Ablation du bloc transformer



(b) Étude de l'encodage positionnel

FIGURE 2.15 – **Étude d'ablation de l'architecture transformer.** (a) Dans cette étude, nous retirons un à un les différents composants du bloc transformer pour analyser leur importance. (b) Dans cette étude, nous comparons différentes manières d'encoder la position dans le mécanisme d'attention (voir section 2.3.1.2).

Enfin, nous avons vu dans la section 2.3.1.2 que l'opération d'attention est invariante aux permutations et qu'elle nécessite l'utilisation d'un encodage de la position pour conserver les relations spatiales existantes entre les pixels. Dans la Figure 2.15 (b), nous comparons les performances de mise en correspondance de réseaux entraînés avec un encodage absolu fixe, un encodage absolu appris, et un encodage relatif de la position des pixels. Alors que l'encodage relatif se montre plus performant dans des tâches comme la classification d'images ou la détection d'objets [Wu et al., 2021b], il s'avère moins robuste pour la mise en correspondance d'images. Cela peut s'expliquer par le fait qu'il est pertinent pour l'opération d'auto-attention, mais que son calcul de différence de position dans une opération d'attention-croisée entre deux images est moins informatif que d'encoder directement la position absolue des pixels. On note également que l'apprentissage de l'encodage absolu de la position offre plus de flexibilité au réseau qu'un encodage fixe.

## 2.5 Conclusion

Ce chapitre a présenté une exploration de la mise en correspondance d'images et de l'avènement du mécanisme d'attention dans le domaine de la vision par ordinateur. La tâche de mise en correspondance, fondamentale dans de nombreuses applications comme la reconstruction 3D, la navigation autonome, ou la localisation visuelle, repose sur l'établissement de correspondances robustes entre des points caractéristiques dans deux images. Traditionnellement, cette tâche a été abordée à travers trois paradigmes principaux : le matching éparsé, semi-dense et dense, chacun apportant des solutions adaptées à différentes contraintes visuelles et applicatives. Cependant, ces approches traditionnelles, bien qu'efficaces, présentent des limites significatives, notamment face aux variations d'illumination, de perspective et de texture, ou encore dans les environnements visuellement complexes.

L'introduction du mécanisme d'attention a ouvert de nouvelles perspectives dans la manière d'aborder ces défis. Initialement conçu pour le traitement du langage naturel, ce mécanisme a démontré sa capacité à capter des relations complexes et à longue portée au sein des données séquentielles, dépassant ainsi les architectures traditionnelles basées sur des approches locales. Appliqué à la vision par ordinateur, le mécanisme d'attention, et plus particulièrement l'auto-attention et l'attention croisée, a montré son potentiel pour améliorer les performances des modèles de mise en correspondance d'images, notamment en renforçant les connexions entre descripteurs tout en filtrant les correspondances non pertinentes.

Dans ce contexte, l'attention permet de dépasser les limitations des méthodes basées sur des relations locales en introduisant une vue d'ensemble du contexte des images, facilitant ainsi la mise en correspondance dans des scènes où les approches classiques échouent. L'auto-attention renforce les relations globales entre les descripteurs au sein d'une même image, tandis que l'attention-croisée entre deux images permet de construire des représentations prenant en compte le point de vue des deux images. Ces mécanismes ont permis d'améliorer la précision et la robustesse des correspondances, en particulier dans des environnements où les méthodes traditionnelles sont mises à mal, comme les scènes faiblement texturées ou avec des répétitions de motifs.

L'étude approfondie de l'attention dans le cadre de la mise en correspondance d'images, objet de cette thèse, est donc d'un grand intérêt. Le potentiel de cette approche pour améliorer des tâches critiques comme la reconstruction 3D ou l'estimation de pose de caméra justifie une exploration poussée. En effet, les premiers travaux intégrant l'attention, tels que SuperGlue et LoFTR, ont démontré des performances significativement supérieures aux méthodes classiques. Ces résultats incitent à penser que l'attention pourrait devenir une composante centrale des futurs systèmes de mise en correspondance d'images.

Dans le chapitre suivant, nous proposerons une méthode permettant de passer du paradigme éparsé au paradigme semi-dense et chercherons à comprendre les facteurs importants du gain de performances des méthodes semi-dense en estimation de pose de caméra. Enfin, le dernier chapitre explorera notre approche pour intégrer le mécanisme d'attention tout au long du réseau pour de la mise en correspondance dense, offrant une perspective nouvelle sur l'utilisation de l'attention dans le traitement global des images.





## **Chapitre 3**

### **De la mise en correspondance de points d'intérêt aux méthodes semi-denses**

## Table des matières

3.1	Introduction . . . . .	55
3.2	État de l'art . . . . .	56
3.2.1	Le paradigme épars (S2S) : SuperGlue . . . . .	56
3.2.2	Passage au semi-dense sans détecteur (SDF) : LoFTR . . . . .	58
3.2.3	Changement de paradigme, besoin d'évaluation équitable . . . . .	59
3.3	Notre approche . . . . .	60
3.3.1	Matching sur demande . . . . .	60
	Définition. . . . .	61
	Métrique d'évaluation . . . . .	61
3.3.2	Architecture SAM . . . . .	62
	Motivations. . . . .	62
	Extraction des descripteurs. . . . .	62
	Étape de communication et espace latent. . . . .	62
	Étape de raffinement des prédictions. . . . .	64
	Détails de l'architecture. . . . .	64
3.3.3	Attention structurée . . . . .	64
	Motivations. . . . .	64
	Formalisation. . . . .	65
3.3.4	Entraînement . . . . .	67
3.4	Experiences . . . . .	67
3.4.1	Estimation de pose de caméra, homographie et séquence d'images . . . . .	68
	Implémentations. . . . .	68
	Positions de requête. . . . .	68
	3.4.1.1 Estimation de pose de caméra, évaluation sur MegaDepth. . . . .	68
	3.4.1.2 Estimation d'homographie, évaluation sur HPatches. . . . .	69
	3.4.1.3 Matching dans une séquence d'images, évaluation sur ETH3D. . . . .	70
	Discussion des résultats. . . . .	70
3.4.2	Étude d'ablation . . . . .	71
3.4.3	Étude de l'attention structurée . . . . .	71
3.4.4	Évolution de SAM . . . . .	72
3.5	Conclusion . . . . .	78

## 3.1 Introduction

Historiquement, la mise en correspondance d’images se réalise en trois étapes : la détection de points d’intérêt, leur description, puis un matching de ces descripteurs. L’étape de matching est donc réalisée de manière éparsée (*sparse-to-sparse* ou S2S), et repose fortement sur la qualité des étapes de détection et de description. Les correspondances sont limitées aux paires possibles parmi les ensembles de points détectés dans les deux images. Bien que les correspondances soient fiables grâce à la robustesse des détecteurs, elles sont restreintes aux régions de l’image où les points peuvent être efficacement détectés et décrits, généralement des régions texturées. Même dans ces régions, trouver une correspondance peut être imprécis en raison de la nature éparsée des correspondances candidates.

Pour surmonter les limitations de la mise en correspondance S2S, des stratégies de matching éparsée à dense (*sparse-to-dense*) [Germain et al., 2020], puis semi-dense [Rocco et al., 2020b] ont été proposées. Avec l’introduction de LoFTR [Sun et al., 2021] en 2021, cette dernière stratégie, combinée aux mécanismes d’attention, a montré des résultats prometteurs et des améliorations significatives dans les benchmarks d’estimation de pose de caméra, sans nécessiter de phase de détection de points d’intérêt. Les méthodes comme LoFTR réalisent une mise en correspondance dense à faible résolution, puis en raffinant ces prédictions grossières, ce qui conduit à des prédictions finales semi-denses. Par conséquent, ces méthodes sont classées comme des méthodes Semi-Denses sans Détecteur (*Semi-dense Detector Free* ou SDF).

Un problème inhérent à la tâche de matching d’images est la difficulté d’évaluer les méthodes. En pratique, cette question est souvent résolue en utilisant des tâches *proxy* telles que l’estimation relative de pose de caméra. Cependant, les méthodes SDF sont conçues pour produire des correspondances précises au niveau du pixel, et la relation entre leur capacité à établir des correspondances précises et la qualité de la pose estimée résultante a reçu peu d’attention.

Une hypothèse logique est que plus une méthode de mise en correspondance est précise, meilleure sera sa performance dans les benchmarks d’estimation de pose. Cependant, les méthodes SDF diffèrent fondamentalement du paradigme S2S. Elles établissent plus de correspondances, dans différentes régions, et la communication entre les descripteurs intègre davantage de contexte. Ces nouvelles caractéristiques masquent les ingrédients clés pour obtenir de bons résultats en estimation de pose et remettent en question notre hypothèse précédente selon laquelle seule la précision des correspondances compte. Dans ce chapitre, nous introduirons une méthode de mise en correspondance d’images utilisant de l’attention structurée, et basée sur un paradigme de mise en correspondance *sur demande*, qui permettra de rapprocher les paradigmes S2S et SDF pour comprendre les ingrédients des bonnes performances des méthodes semi-denses. SAM peut s’adapter pour se comporter comme une méthode S2S ou comme une méthode SDF, permettant ainsi des comparaisons équitables entre les deux. Notre objectif sera d’explorer l’hypothèse selon laquelle le matching dans les régions sans texture améliore l’estimation de la pose de la caméra. De plus, nous investiguons un nouveau mécanisme d’attention structurée qui maintient une représentation entièrement positionnelle tout au long du réseau, ce qui semble avantageux dans le contexte de la mise en correspondance d’images.

Dans un premier temps, nous analyserons deux méthodes S2S et SDF pour comprendre la différence entre les deux paradigmes et chercher les ingrédients de la réussite des méthodes SDF. Ensuite, nous présenterons notre approche, notre architecture, ainsi que notre module d’attention structurée. Puis, nous présenterons nos résultats pour les méthodes SDF les plus performantes et notre architecture.

## 3.2 État de l'art

Dans cette partie, nous allons voir la différence entre les méthodes S2S et SDF de l'état de l'art. Les différentes approches S2S sont décrites dans la Section 2.2.2.1, et la Section 2.2.2.2 est dédiée aux différentes approches SDF. Ici, nous nous concentrerons sur les détails de deux méthodes proches dans leur fonctionnement : SuperGlue [Sarlin et al., 2020] pour les S2S et LoFTR [Sun et al., 2021] pour les SDF.

Ces méthodes ont permis des améliorations significatives sur les benchmarks d'estimation de pose de caméra lors de leur parution. SuperGlue a introduit l'utilisation de l'attention pour faire communiquer les descripteurs de deux images, et LoFTR reprend ce mécanisme pour faire communiquer des représentations denses basse résolution des images. Bien que les deux paradigmes soient différents, leur similarité architecturale nous permet de poser des hypothèses solides sur les causes du gain de performance en estimation de pose qu'apporte LoFTR.

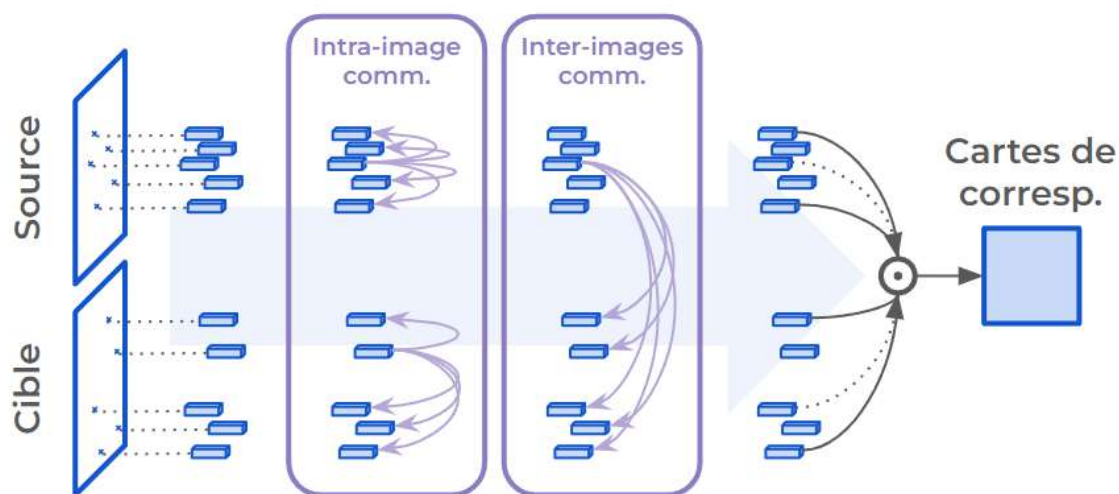
### 3.2.1 Le paradigme éparsé (S2S) : SuperGlue

Les méthodes S2S furent pendant longtemps les plus performantes pour la mise en correspondance d'images et sont encore très largement utilisées. Cependant, ces méthodes sont généralement vulnérables aux importants changements de conditions. Les descripteurs locaux peuvent échouer à capturer suffisamment d'information pour éliminer les ambiguïtés de correspondance lors d'importants changements d'éclairage, de point de vue ou pour des structures répétitives. Les méthodes S2S classiques construisent indépendamment les descripteurs des points d'intérêt, perdant l'information globale à l'intérieur des images. L'utilisation d'un réseau de convolution siamois pour la création des descripteurs permet de partiellement conserver cette information globale des images. Cependant, ces traitements restent indépendants pour les deux images, et ce manque de communication entre les descripteurs limite leur capacité à capturer des relations complexes entre les points d'intérêt.

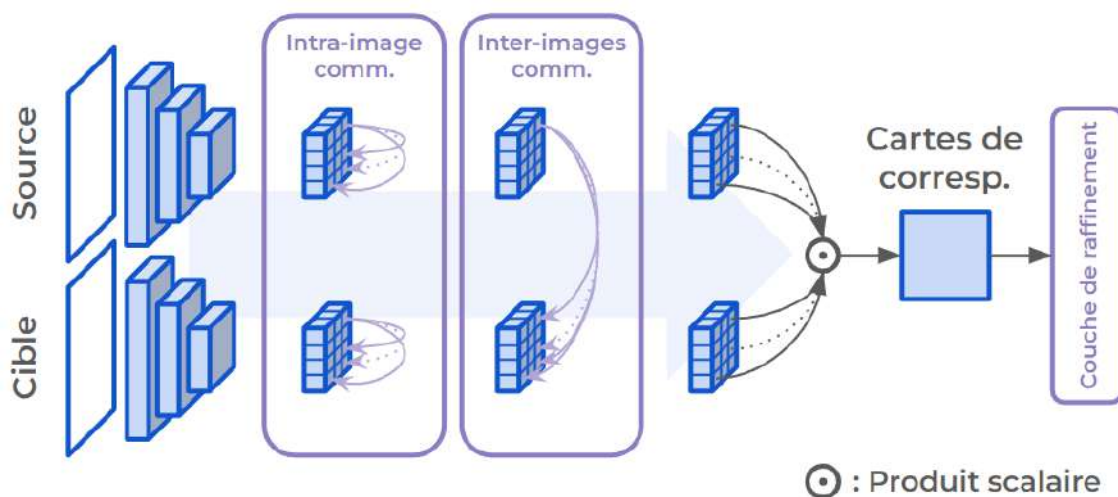
SuperGlue propose de surmonter ces limitations en apprenant conjointement à produire des descripteurs robustes, mais aussi à trouver les correspondances entre ces descripteurs à travers des cartes de correspondances éparées. Ils utilisent un réseau de neurones attentionnel en graphes (AGNN [Veličković et al., 2018]) pour faire communiquer tous les descripteurs et ainsi capturer les interactions contextuelles entre les points d'intérêt intra- et inter-images.

SuperGlue (schématisé dans la Figure 3.1 (a)) fonctionne en plusieurs étapes clés :

- **Extraction des points d'intérêt et des descripteurs.** On commence par extraire des points d'intérêt dans les images à l'aide de méthodes locales ou de modèles appris comme SuperPoint (SP) [DeTone et al., 2018]. Pour chaque point d'intérêt, un descripteur est créé à partir de l'information contenue dans l'image autour du point d'intérêt.
- **Encodage des relations contextuelles avec un AGNN.** Le modèle est composé d'une succession de couches permettant la communication entre les différents descripteurs. Un graphe est construit où chaque nœud correspond à un point d'intérêt. Des connexions sont créées entre les nœuds d'une même image pour capturer les relations intra-image et d'autres connexions sont ajoutées entre les nœuds des différentes images pour capturer les relations inter-images. Le message de communication dans le graphe et la mise à jour des descripteurs sont réalisés par des opérations d'auto-attention et d'attention-croisée.
- **Création des correspondances finales.** Une fois les descripteurs mis à jour, des cartes de correspondances sont créées en calculant le produit scalaire entre les descripteurs des points d'intérêt de l'image source et ceux de l'image cible. Les points d'intérêt non



(a) Méthode épars (S2S) : SuperGlue



(b) Méthode semi-dense sans détecteur (SDF) : LoFTR

FIGURE 3.1 – **Schémas comparatifs entre les méthodes S2S (a) et SDF (b).** Les méthodes S2S utilisant de l'attention, comme SuperGlue [Sarlin et al., 2020], commencent par détecter et décrire des points d'intérêt. Elles utilisent ensuite de l'attention pour créer de la communication entre les descripteurs. Enfin, les correspondances sont établies en calculant des cartes de correspondances. De leur côté, les méthodes SDF comme LoFTR [Sun et al., 2021] extraient des représentations denses à résolution grossière pour les deux images. De l'attention est ensuite utilisée pour créer de la communication entre les deux représentations denses. Enfin, des correspondances grossières sont établies en calculant des cartes de correspondances, puis ramenées à la résolution d'origine grâce à un module de raffinement.

covisibles sont filtrés en ajoutant un descripteur "poubelle" spécifique lors du calcul des cartes.

- **Optimisation.** SuperGlue est entraîné de manière supervisée en utilisant des paires d'images annotées, avec une entropie croisée comme fonction de coût qui favorise les correspondances correctes tout en pénalisant les mauvaises correspondances.

Cette méthode présente des avantages significatifs en matière de correspondance d'images, notamment sa capacité à intégrer des relations contextuelles globales grâce à l'utilisation d'auto-attention et d'attention-croisée. Cela permet à SuperGlue d'établir des correspondances robustes même dans des cas de forte ambiguïté. Cependant, SuperGlue dépend de la qualité des points d'intérêt extraits en amont, ce qui peut limiter ses performances dans des scénarios où les points d'intérêt sont rares, mal détectés ou mal décrits.

### 3.2.2 Passage au semi-dense sans détecteur (SDF) : LoFTR

LoFTR est une méthode de mise en correspondance d'images qui se distingue par le fait qu'elle n'utilise pas de détecteurs de points d'intérêt traditionnels comme SIFT, ORB ou SuperPoint. Au lieu de cela, LoFTR adopte une approche semi-dense où un réseau convolutif siamois est appris pour générer des cartes grossières de descripteurs et la mise en correspondance est apprise via des cartes de correspondances denses basse résolution. LoFTR reprend les motivations de SuperGlue et intègre des couches d'attention pour permettre une communication entre les cartes de descripteurs.

LoFTR (schématisé dans la Figure 3.1 (b)) fonctionne en plusieurs étapes similaires à SuperGlue mais permettant de ne plus utiliser de détecteur :

- **Extraction des cartes de descripteurs pour les images.** LoFTR commence par extraire des cartes de descripteurs à partir des deux images en utilisant un CNN siamois léger (ResNet-18 [He et al., 2015]). Contrairement à SuperGlue qui se concentre sur les points d'intérêt, LoFTR extrait des caractéristiques denses, à basse et haute résolution, couvrant toute l'image.
- **Communication grossière entre les deux images.** Ensuite, comme SuperGlue, LoFTR utilise de l'auto-attention et de l'attention-croisée pour créer de la communication intra et inter-images. Cependant, la communication se fait cette fois-ci entre les représentations denses des images. Pour des questions de complexité de l'opération d'attention, ce sont les cartes de descripteurs grossières à  $\frac{1}{8}$ <sup>ème</sup> de résolution qui sont utilisées.
- **Sélection des correspondances.** Une fois les représentations grossières mises à jour par l'attention, des cartes de correspondances sont créées en calculant le produit scalaire entre toutes les features de l'image source et celles de l'image cible. Deux softmax sont appliqués sur les deux dimensions de cette matrice de correspondances pour faire ressortir les correspondances les plus prometteuses, et qui se valident dans les sens source  $\rightarrow$  cible et cible  $\rightarrow$  source. Les correspondances sélectionnées sont celles ayant un score de correspondance supérieur à un seuil fixé.
- **Raffinement des prédictions grossières.** Après avoir identifié ces correspondances, LoFTR utilise un processus de raffinement à une échelle plus fine. Dans les cartes de descripteurs haute résolution, des patchs sont extraits autour des correspondances grossières. Puis des auto-attentions et attentions-croisées sont appliquées sur ces patchs et des cartes de correspondances locales sont calculées pour trouver les correspondances finales.
- **Optimisation de bout en bout.** L'entraînement se fait de bout en bout, ce qui signi-



fié que tous les paramètres du modèle, y compris ceux du CNN siamois qui extrait les descripteurs, sont optimisés conjointement pour maximiser la performance de correspondance via une entropie croisée.

LoFTR offre plusieurs avantages majeurs, notamment sa capacité à effectuer un matching semi-dense sans dépendre de détecteurs de points d'intérêt, ce qui le rend particulièrement efficace dans des scènes avec peu de texture ou des variations géométriques complexes. Cependant, LoFTR présente une complexité computationnelle plus élevée que SuperGlue due à l'utilisation de l'attention de manière dense.

### 3.2.3 Changement de paradigme, besoin d'évaluation équitable

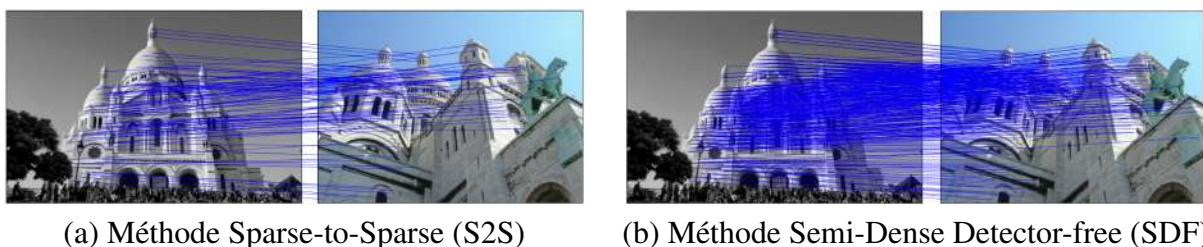


FIGURE 3.2 – **Comparaison des prédictions entre les méthodes S2S (a) et SDF (b).** On constate que les SDF produisent plus de correspondances, dont certaines dans des régions peu texturées. Les S2S sont contraintes à établir des correspondances dans les régions où sont détectés les points d'intérêt, généralement des régions texturées.

Les méthodes éparses et semi-denses représentent deux approches distinctes pour la mise en correspondance d'images. Chaque approche a ses propres avantages et limitations, mais comme nous l'avons vu dans les sections 3.2.1 et 3.2.2, elles peuvent être très similaires dans leur mise en œuvre. SuperGlue (S2S) et LoFTR (SDF) utilisent toutes les deux autant de couches d'auto-attention et d'attention-croisée pour intégrer le contexte des deux images dans leurs représentations, les correspondances sont établies de la même manière avec un produit scalaire, et elles sont optimisées avec la même fonction de coût. Cependant, LoFTR surpasse très largement SuperGlue en estimation relative de pose de caméra (Tableau 3.1).

Paradigme	Méthode	Estimation de pose AUC ↑		
		@5°	@10°	@20°
S2S	SP+SuperGlue	42.18	61.16	75.96
SDF	LoFTR	<b>52.80</b>	<b>69.19</b>	<b>81.18</b>

TABLE 3.1 – **Comparaison des méthodes S2S et SDF en estimation de pose de caméra sur MegaDepth-1500.** LoFTR offre une meilleure estimation de pose de caméra que SuperGlue avec le détecteur SuperPoint.

La première problématique nous vient directement de la métrique utilisée pour évaluer les méthodes. Alors que les méthodes de matching sont toutes optimisées pour produire des correspondances précises, leur comparaison se fait à travers la tâche *proxy* qu'est l'estimation de pose de caméra. L'hypothèse derrière cela est que plus une méthode est capable de générer des correspondances précises, meilleure elle sera en estimation de pose. Cette hypothèse est vraie si l'on

compare deux méthodes S2S qui établissent les correspondances aux mêmes points d'intérêt venant du même détecteur, mais le passage au paradigme SDF introduit de nouvelles problématiques. Par exemple, comme on peut l'observer dans la Figure 3.2, alors que les méthodes S2S produisent en général quelques centaines de correspondances, les méthodes SDF peuvent en produire plusieurs milliers. Pour une estimation de pose précise, est-il préférable d'avoir une méthode qui produit une centaine de correspondances précises ou une méthode qui en produit un millier moins précises ? On note aussi que les SDF peuvent établir des correspondances dans les régions homogènes alors que les S2S non. Cela peut-il avoir un impact sur l'estimation de pose ? Les méthodes SDF ne proposant pas d'évaluation de la précision de leurs correspondances, il est très difficile de trouver des réponses et donc de comprendre la cause réelle des très bonnes performances des méthodes SDF en estimation de pose de caméra.

D'après les auteurs de LoFTR, le gain de performance en estimation de pose peut s'expliquer par *"La capacité de LoFTR à produire des correspondances de grande qualité même dans les régions non-distinctives avec peu de texture"*. Mais aucune analyse quantitative n'est proposée.

### 3.3 Notre approche

D'après les présentations précédentes des méthodes S2S et SDF, on peut formuler les hypothèses suivantes sur les facteurs qui pourraient expliquer les bons résultats en estimation de pose des méthodes SDF :

- **Plus de correspondances.** Générer un plus grand nombre de correspondances peut mieux contraindre la pose lors de l'estimation par l'algorithme RANSAC.
- **Correspondances dans les régions non-texturées.** Les méthodes S2S sont limitées à établir des correspondances dans les régions où sont détectés les points d'intérêt, les régions texturées. Les SDF peuvent également matcher dans les régions homogènes, ce qui pourrait avoir un impact sur l'estimation de pose.
- **Meilleure précision générale des correspondances.** Les méthodes SDF trouvent peut-être simplement des correspondances plus précises, indépendamment de leur nombre ou localisation, ce qui amènerait à une meilleure estimation de pose.

Dans leur article, les auteurs de LoFTR supposent que l'amélioration des performances en estimation de pose de caméra viendrait de la capacité de LoFTR à matcher précisément dans les régions non-texturées. Cependant, à notre connaissance, le lien entre cette capacité à établir des correspondances dans ces régions et la qualité de la pose estimée qui en résulte a jusqu'à présent reçu peu d'attention. Notre méthode constitue une première tentative pour étudier ce lien.

#### 3.3.1 Matching sur demande

Nous cherchons à créer une méthode agnostique qui puisse à la fois réaliser des correspondances uniquement dans les régions texturées comme les S2S, mais qui puisse également matcher dans les régions homogènes comme les SDF. Nous formulons ces caractéristiques à travers le paradigme de *matching sur demande* (*On-Demand-Matching* ou ODM) qui nous aidera à envisager de nouvelles architectures de réseau.

**Définition.** Étant donné un ensemble de positions 2D requêtes  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$  dans une image source  $I_s$ , le ODM consiste à estimer leurs correspondances 2D  $\{\mathbf{p}_{t,i}\}_{i=1\dots L}$  dans une image cible  $I_t$ . Ainsi, une méthode  $\mathcal{M}$  qui implémente ODM peut être formulée de la manière suivante :

$$\{\mathbf{p}_{t,i}\}_{i=1\dots L} = \mathcal{M}(I_s, I_t, \{\mathbf{p}_{s,i}\}_{i=1\dots L}). \quad (3.1)$$

Notons que les approches SDF et S2S peuvent être considérées comme deux implémentations de ODM. Une méthode S2S implémente ODM car l'étape de détection des points d'intérêt contraint à la fois les positions des requêtes  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$  dans l'image source, et également les positions des correspondants  $\{\mathbf{p}_{t,i}\}_{i=1\dots L}$  dans l'image cible. Une approche SDF peut aussi être utilisée pour implémenter ODM puisque son étape de matching grossier produit des correspondances pour tous les descripteurs grossiers de la source. Comme nous l'avons vu dans la section 2.2.2.2 sur le matching semi-dense, si le matching grossier se fait à  $\frac{1}{8}$ ème de la résolution d'origine, après l'étape de raffinement, cela revient à trouver des correspondances pour toutes les positions sur une grille  $\frac{1}{8}$ ème de l'image source. Pour les approches SDF, cette grille est donc notre ensemble de positions requêtes  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$  dans le paradigme ODM. Cette implémentation consiste donc à résoudre le problème de ODM pour chaque descripteur de l'image source à une résolution grossière.

À résolution fine, les positions de  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$  dans l'image source sont contraintes à une grille couvrant  $\frac{1}{8}$ ème des pixels pour LoFTR, mais certaines améliorations comme celle proposée dans [Zhou et al., 2021] retirent cette contrainte. À noter que d'autres méthodes dites fonctionnelles comme COTR [Jiang et al., 2021] (voir section 2.2.2.3) entrent aussi dans le paradigme ODM en prédisant les  $L$  correspondants  $\{\mathbf{p}_{t,i}\}_{i=1\dots L}$  de  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$  de manière indépendante. Cette implémentation est extrême, car d'une part elle rend l'approche élégante, mais d'autre part, en traitant les positions requêtes de manière séquentielle, elle rend le temps de calcul déraisonnable.

Cette formulation sous la forme d'un matching *sur demande* d'un ensemble de positions requêtes nous aidera à envisager une nouvelle architecture de réseau permettant de se comparer équitablement aux architectures SDF et S2S.

**Métrique d'évaluation** Notre objectif est de voir si la précision de mise en correspondance dans les régions faiblement texturées est l'ingrédient clé des bonnes performances en estimation de pose des méthodes SDF utilisant de l'attention. Étant donné qu'il est difficile d'identifier efficacement les points appartenant à ces régions, nous décidons de rapporter deux métriques : la précision des correspondances globale (*Matching Accuracy* ou MA) et la précision des correspondances dans les régions texturées ( $MA_{text}$ ), où les positions sont calculables efficacement grâce à des détecteurs comme SIFT. De cette manière, nous pouvons efficacement analyser les performances des méthodes dans les régions texturées et non texturées.

Étant donné un ensemble de test  $\mathcal{D}_{test}$ , une méthode  $\mathcal{M}$  à évaluer traite d'abord chaque échantillon de  $\mathcal{D}_{test} = \left\{ I_{s_k}, I_{t_k}, \left\{ \mathbf{p}_{s_k,i}, \mathbf{p}_{t_k,i}^{GT}, v_{t_k,i} \right\}_{i=1\dots L_k} \right\}_{k=1\dots K}$ . Pour chaque prédiction  $\mathbf{p}_{t_k,i}$  de  $\mathcal{M}$ , la distance euclidienne  $d_{k,i}$  par rapport à la position 2D de vérité terrain  $\mathbf{p}_{t_k,i}^{GT}$  peut être calculée :  $d_{k,i} = \left\| \mathbf{p}_{t_k,i} - \mathbf{p}_{t_k,i}^{GT} \right\|_2$ . Les méthodes SDF sont entraînées pour détecter ou filtrer les cas où le correspondant  $\mathbf{p}_{t_k,i}$  du point  $\mathbf{p}_{s_k,i}$  n'est pas covisible dans l'image cible. Par conséquent, en tant que critère d'évaluation, nous ignorons les prédictions qui ne sont pas marquées comme

covisibles ( $[v_{t_k,i} = 0]$ ) et considérons les précisions de matching (MA) comme dans [Truong et al., 2020], c'est-à-dire le ratio de correspondances correctes, pour différents seuils d'erreur en pixels ( $\eta$ ).

$$\text{MA}(\eta) = \frac{\sum_{k=1}^K \sum_{i=1}^{L_k} [v_{t_k,i} = 1] [d_{k,i} < \eta]}{\sum_{k=1}^K \sum_{i=1}^{L_k} [v_{t_k,i} = 1]}. \quad (3.2)$$

où  $[.]$  correspond au crochet d'Iverson. Nous parlerons de  $\text{MA}_{\text{ext}}(\eta)$  lorsque les correspondances sont issues de régions texturées.

Nous souhaitons comparer la précision des correspondances dans différentes régions avec les performances d'estimation de pose de caméra. Nous rapporterons donc également l'AUC de l'erreur angulaire d'estimation de pose comme décrit dans la section 2.2.1.2.

### 3.3.2 Architecture SAM

**Motivations.** Nous proposons une architecture de réseau basée sur l'attention, spécialement conçue pour le paradigme ODM (équation 3.1) introduit dans la section précédente. Adopter ce paradigme permet de développer une architecture novatrice, sans aucune couche d'attention dense (à aucune résolution), contrairement aux architectures SDF, ce qui conduit à un réseau simple et efficace sur le plan computationnel. Dans ce sens, elle ressemble aux architectures S2S, mais l'utilisation d'un espace latent (inspiré de l'architecture Perceiver [Jaegle et al., 2021] [Jaegle et al., 2022]) lui permet d'extraire le contexte de l'image cible pour établir des correspondances précises.

Schématisée dans la Figure 3.3, notre architecture peut se décomposer en plusieurs étapes : l'extraction de descripteurs denses basse résolution, la création d'un espace latent composé des descripteurs des positions requêtes et de vecteurs appris, la communication au sein de l'espace latent, la mise en correspondance et le raffinement des prédictions. Dans la suite, nous détaillons les différents éléments qui composent cette architecture SAM (Structured-Attention Matching).

**Extraction des descripteurs.** Notre méthode prend en entrée une image source  $I_s$  ( $H_s \times W_s \times 3$ ), une image cible  $I_t$  ( $H_t \times W_t \times 3$ ) et un ensemble de positions 2D  $\{\mathbf{p}_{s,i}\}_{i=1\dots L}$ , dites de requête. La première étape de SAM est une étape classique d'extraction de cartes de descripteurs denses. À partir de l'image source  $I_s$  ( $H_s \times W_s \times 3$ ) et de l'image cible  $I_t$  ( $H_t \times W_t \times 3$ ), des descripteurs visuels denses  $F_s$  ( $\frac{H_s}{4} \times \frac{W_s}{4} \times 128$ ) et  $F_t$  ( $\frac{H_t}{4} \times \frac{W_t}{4} \times 128$ ) sont extraits respectivement pour la source et la cible à l'aide d'un CNN siamois. Un encodage positionnel (PE) appris est utilisé, calculé à l'aide d'un MLP [Sarlin et al., 2020], puis concaténé avec les descripteurs visuels des images source et cible pour obtenir deux tenseurs  $H_s$  ( $\frac{H_s}{4} \times \frac{W_s}{4} \times 256$ ) et  $H_t$  ( $\frac{H_t}{4} \times \frac{W_t}{4} \times 256$ ). Pour chaque point de requête 2D  $\mathbf{p}_{s,i}$ , un descripteur  $\mathbf{h}_{s,i}$  de taille 256 est extrait de  $H_s$ . En pratique, nous utilisons des emplacements de requête entiers, il n'est donc pas nécessaire d'effectuer une interpolation ici.

**Étape de communication et espace latent.** Afin de permettre aux descripteurs de requêtes  $\mathbf{h}_{s,i=1\dots L}$  de communiquer et de s'ajuster par rapport à  $H_t$ , nous nous inspirons de Perceiver et considérons un ensemble latent de  $N = M + L$  vecteurs, composé de  $M$  vecteurs latents appris  $\{\mathbf{m}_i\}_{i=1\dots M}$  et des  $L$  descripteurs de requêtes  $\{\mathbf{h}_{s,i}\}_{i=1\dots L}$ . Ces vecteurs latents  $\{\{\mathbf{m}_i\}_{i=1\dots M}, \{\mathbf{h}_{s,i}\}_{i=1\dots L}\}$  sont utilisés comme requêtes dans une couche d'attention-croisée pour extraire les informations pertinentes de  $H_t$ , et obtenir finalement un ensemble mis à jour de vecteurs latents

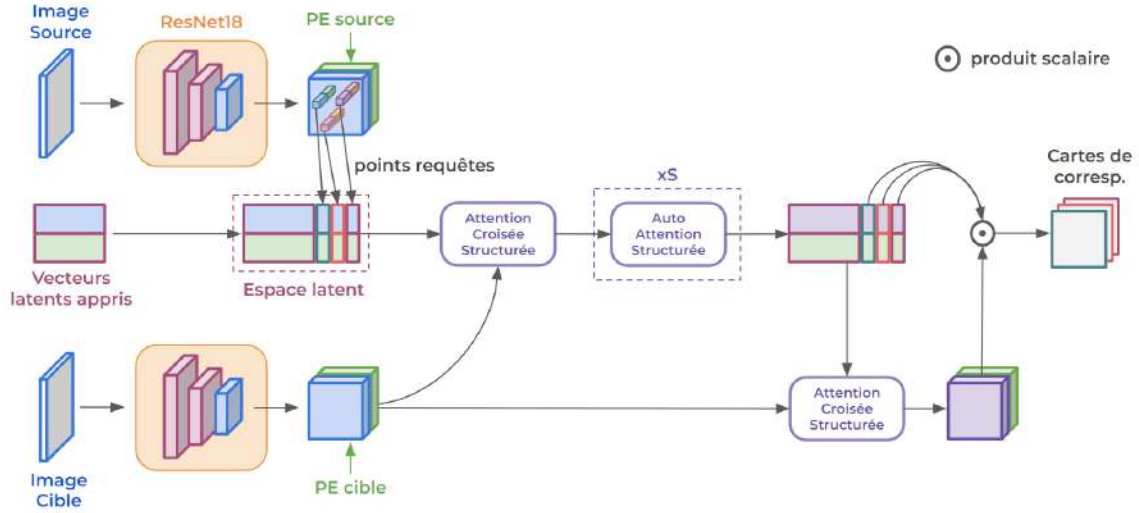


FIGURE 3.3 – **Aperçu de la méthode de matching basée sur l'attention structurée (SAM) proposée.** L'architecture de mise en correspondance commence par extraire les caractéristiques des images source et cible à  $\frac{1}{4}$  de la résolution d'origine. Ensuite, un ensemble de vecteurs latents appris est utilisé en parallèle avec les descripteurs des emplacements de requête, et une attention-croisée structurée est appliquée en entrée avec les caractéristiques denses de l'image cible. L'espace latent est ensuite traité par une succession de couches d'auto-attention structurée. Une attention-croisée structurée en sortie est appliquée pour mettre à jour les caractéristiques de l'image cible avec les informations provenant de l'espace latent. Enfin, les cartes de correspondance sont obtenues en utilisant un produit scalaire.

$\left\{ \left\{ \mathbf{m}_i^{(0)} \right\}_{i=1 \dots M}, \left\{ \mathbf{h}_{s,i}^{(0)} \right\}_{i=1 \dots L} \right\}$ . Cette première attention-croisée peut être vue comme une manière d'encoder l'information de l'image cible dans l'espace latent.

D'une part, les descripteurs  $\left\{ \mathbf{h}_{s,i}^{(0)} \right\}_{i=1 \dots L}$  contiennent désormais l'information nécessaire pour trouver leurs correspondants respectifs au sein de  $H_t$ . D'autre part, les descripteurs  $\left\{ \mathbf{m}_i^{(0)} \right\}_{i=1 \dots M}$  extraient une représentation générale de  $H_t$  puisqu'ils ne sont pas informés des emplacements 2D des requêtes.

Ensuite, une série de  $S$  couches d'auto-attention est appliquée aux vecteurs latents pour obtenir  $\left\{ \left\{ \mathbf{m}_i^{(S)} \right\}_{i=1 \dots M}, \left\{ \mathbf{h}_{s,i}^{(S)} \right\}_{i=1 \dots L} \right\}$ . Dans ces couches, tous les vecteurs latents peuvent communiquer et s'ajuster les uns par rapport aux autres. Par exemple, la communication entre les vecteurs  $\left\{ \mathbf{h}_{s,i}^{(S)} \right\}_{i=1 \dots L}$  peut lever certaines ambiguïtés de correspondance, et la communication entre  $\left\{ \mathbf{m}_i^{(S)} \right\}_{i=1 \dots M}$  et  $\left\{ \mathbf{h}_{s,i}^{(S)} \right\}_{i=1 \dots L}$  peut permettre d'affiner la précision des correspondances par une communication avec le contexte général de l'image cible.

Enfin,  $H_t$  est utilisé comme requête dans une couche d'attention-croisée en sortie pour extraire les informations pertinentes des vecteurs latents. Le tenseur résultant est noté  $H_t^{\text{out}}$ . Pour chaque descripteur mis à jour  $\mathbf{h}_{s,i}^{(S)}$ , une carte de correspondance  $C_{t,i}$  (de taille  $\frac{H_t}{4} \times \frac{W_t}{4}$ ) est obtenue en calculant le produit scalaire entre  $\mathbf{h}_{s,i}^{(S)}$  et  $H_t^{\text{out}}$ . Enfin, pour chaque position de requête 2D  $\mathbf{p}_{s,i}$ , le correspondant prédit  $\hat{\mathbf{p}}_{t,i}$  est défini comme étant l'argmax de  $C_{t,i}$ .

Notons que chacune des couches d'attention utilisées ici est une version structurée de la softmax-attention, dont nous détaillerons le fonctionnement dans la Section 3.3.3.

Dans la Figure 3.4, nous proposons une visualisation des vecteurs latents appris par SAM. Pour les besoins de la visualisation, nous avons utilisé  $L = 64$ . La carte de requête moyenne est



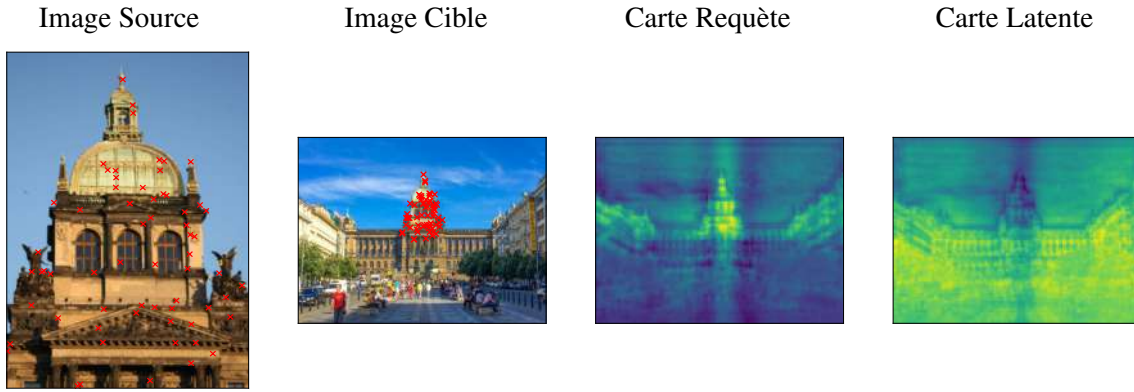


FIGURE 3.4 – **Visualisation des vecteurs latents appris par SAM.** La carte de requête moyenne est obtenue en moyennant 64 cartes de correspondance des 64 emplacements de requête (croix **rouges**), tandis que la carte latente moyenne est obtenue en moyennant les 128 cartes de correspondance des 128 vecteurs latents appris. Nous constatons que la carte de requête moyenne est principalement activée autour des correspondants, tandis que la carte latente moyenne est activée dans les autres régions, qui peuvent être interprétées comme les régions contextuelles.

donc obtenue en faisant la moyenne des 64 cartes de correspondance des 64 positions de requête, tandis que la carte latente moyenne est obtenue en faisant la moyenne des 128 cartes de correspondance des 128 vecteurs latents appris. Nous observons que la carte de requête moyenne est principalement activée autour des correspondants, tandis que ces régions sont moins activées dans la carte latente moyenne.

**Étape de raffinement des prédictions.** L’architecture décrite précédemment produit des cartes de correspondance grossières avec une résolution de  $1/4$ . Ainsi, les correspondants prédits  $\{\mathbf{p}_{t,i}\}_{i=1\dots L}$  doivent être affinés à pleine résolution. Pour ce faire, nous utilisons simplement un second CNN siamois qui génère des caractéristiques denses pour l’image source et l’image cible à pleine résolution. Pour chaque position de requête  $\mathbf{p}_{s,i}$ , une carte de correspondance est calculée sur une fenêtre de taille 11 centrée autour de la prédiction grossière. La position correspondante 2D prédite  $\mathbf{p}_{t,i}$  est définie comme l’argmax de cette carte de correspondance. Ce réseau d’affinement est entraîné séparément en utilisant la même fonction de coût d’entropie croisée (eq. 3.6).

**Détails de l’architecture.** L’architecture proposée repose sur un backbone classique ResNet18, avec une profondeur de descripteurs réduite à 128. L’encodage de la position est réalisé par trois blocs successifs comprenant chacun une couche de convolution, suivie d’une BatchNorm et d’une activation ReLU, et prenant les coordonnées  $(x,y)$  en entrée. Chaque couche d’attention structurée est composée de 8 heads. Pour la communication dans l’espace latent, nous utilisons 16 couches d’auto-attention, dont la taille est fixée à  $M = 128$  et  $L = 1024$  pour l’entraînement, garantissant une gestion efficace du nombre de requêtes.

### 3.3.3 Attention structurée

**Motivations.** Dans le mécanisme d’attention, il est nécessaire d’ajouter l’information de position aux caractéristiques, car l’opération est invariante aux permutations, et les pixels d’une



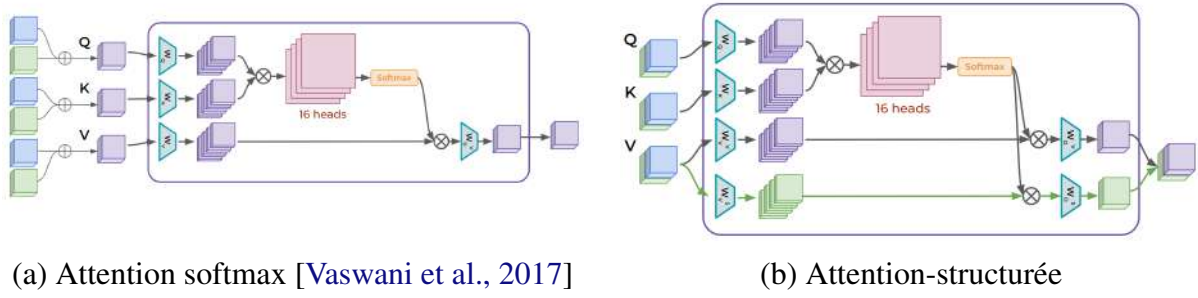


FIGURE 3.5 – **Comparaison entre l’attention softmax (a) et notre attention-structurée (b).** Les volumes verts représentent des cartes de descripteurs contenant uniquement de l’information visuelle, les bleus uniquement de l’information positionnelle, et les violets un mix d’information visuo-positionnelle. Alors que l’attention softmax produit uniquement une représentation visuo-positionnelle, notre attention-structurée produit en parallèle une représentation uniquement positionnelle. Les cartes d’attention étant partagées entre les deux branches, l’attention-structurée n’a qu’un très faible coût calculatoire supplémentaire.

image sont corrélés par leur localité. La manière classique de faire est d’encoder la position des pixels et de l’ajouter aux caractéristiques visuelles. L’attention pourra alors créer, au fur et à mesure des couches, une représentation visuo-spatiale de haut niveau. Si, pour certaines tâches de vision comme la reconnaissance d’objets, l’information de position peut être imprécise, pour le matching, elle est cruciale. Nous cherchons à trouver des correspondances précises entre deux images d’une scène 3D. Mélanger la position aux caractéristiques visuelles pourrait mener à une perte d’information. Nous proposons donc de structurer l’opération d’attention afin de construire, en parallèle de la représentation visuo-spatiale classique, une représentation purement positionnelle. De cette manière, le réseau peut créer des relations uniquement basées sur l’information de position. Un schéma comparant l’attention-structurée avec l’attention softmax classique est proposé dans la Figure 3.5.

**Formalisation.** À la fin de l’étape d’extraction de cartes de descripteurs, la moitié supérieure de chaque vecteur correspond aux caractéristiques visuelles, tandis que la moitié inférieure correspond aux caractéristiques positionnelles.

Pour faire apparaître toutes les matrices de poids apprises, l’attention softmax [Vaswani et al., 2017] sur  $H$  heads pour une requête  $\mathbf{x}$  et un ensemble de clés et valeurs  $\{\mathbf{y}_n\}_{n=1\dots N}$  peut se réécrire comme suit :

$$\sum_{h=1}^H \sum_{n=1}^N \underbrace{s_{h,n}}_{1 \times 1} \underbrace{\mathbf{W}_{o,h}}_{D \times D_H D_H \times D} \underbrace{\mathbf{W}_{v,h}}_{D \times D} \mathbf{y}_n, \quad (3.3)$$

$$\text{Avec, } s_{h,n} = \frac{\exp(\underbrace{\mathbf{x}^\top}_{1 \times D} \underbrace{\mathbf{W}_{q,h}^\top}_{D \times D_H D_H \times D} \underbrace{\mathbf{W}_{k,h}}_{D \times D} \underbrace{\mathbf{y}_n^\top}_{D \times 1})}{\sum_{m=1}^N \exp(\mathbf{x}^\top \mathbf{W}_{q,h}^\top \mathbf{W}_{k,h} \mathbf{y}_m)}. \quad (3.4)$$

Dans chaque couche d’attention (attention-croisée en entrée, auto-attention de communication et attention-croisée en sortie), nous avons considéré important de structurer les matrices de transformation linéaire  $\mathbf{W}_{o,h}$  et  $\mathbf{W}_{v,h}$  (équation 3.5) afin de contraindre la moitié inférieure de

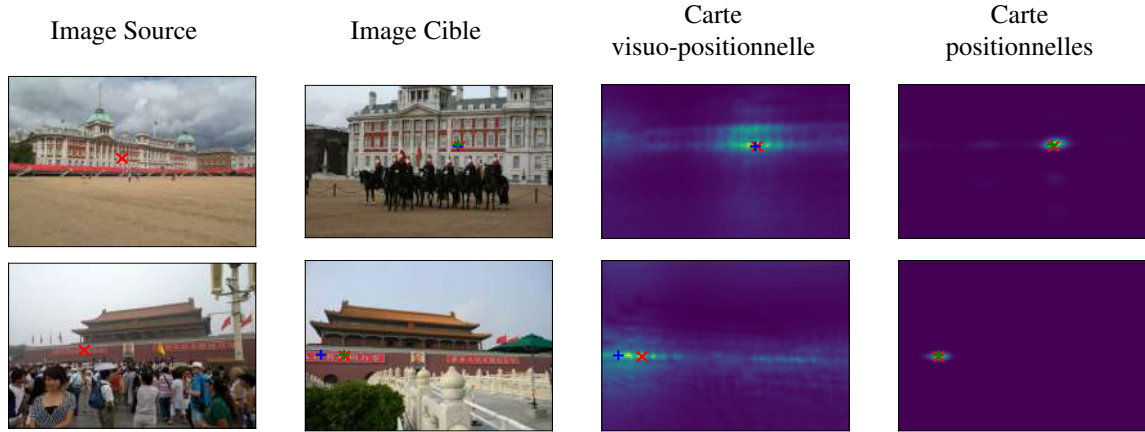


FIGURE 3.6 – **Visualisation - Attention structurée.** Les cartes visuo-positionnelles et positionnelles sont calculées avant la cross-attention en sortie. Les croix **rouges** représentent les correspondances de vérité terrain. Les croix **bleues** et **vertes** correspondent respectivement aux maxima des cartes visuo-positionnelles et positionnelles. On peut observer que les cartes visuo-positionnelles sont fortement multimodales (c'est-à-dire, sensibles aux structures répétitives) tandis que les cartes positionnelles sont presque unimodales.

chaque vecteur de sortie à ne contenir qu'une transformation linéaire des encodages positionnels.

$$\underbrace{W_{o,h}}_{D \times D_H} = \left[ \begin{array}{c|c} \underbrace{0}_{\frac{D}{2} \times \frac{D_H}{2}} & \underbrace{W_{o,h,up}}_{\frac{D}{2} \times D_H} \\ \hline & \underbrace{W_{o,h,low}}_{\frac{D}{2} \times \frac{D_H}{2}} \end{array} \right]. \quad (3.5)$$

Les matrices  $W_{v,h}$  et les couches MLP à la fin de chaque module d'attention sont structurées exactement de la même manière. En conséquence, tout au long du réseau, la moitié inférieure de chaque vecteur latent ne contient que des transformations des encodages positionnels, sans aucune caractéristique visuelle. La Figure 3.5 schématise la différence entre la *Softmax-attention* et notre *Attention-structurée*.

Dans la Figure 3.6, nous proposons une visualisation de deux cartes de correspondance différentes avant l'étape de cross-attention en sortie. La première carte est produite en utilisant les 128 premières dimensions de la représentation des caractéristiques et contient des caractéristiques visuo-positionnelles de haut niveau. La seconde carte est produite en utilisant les 128 dernières dimensions de la représentation des caractéristiques et ne contient que des encodages positionnels. En construisant ces cartes de correspondance distinctes, nous pouvons observer que, tandis que la représentation visuo-positionnelle est sensible aux structures répétitives, la représentation purement positionnelle tend à s'activer uniquement dans les zones voisines de la correspondance.

### 3.3.4 Entraînement

Pendant l’entraînement, nous utilisons une entropie croisée (CE) comme fonction de coût [Germain et al., 2020] sur chaque carte de correspondance  $C_{t,i}$  afin de maximiser son score à la position de vérité terrain  $\mathbf{p}_{t,i}$  :

$$CE(C_{t,i}, \mathbf{p}_{t,i}) = -\ln \left( \frac{\exp C_{t,i}(\mathbf{p}_{t,i})}{\sum_{\mathbf{q}} \exp C_{t,i}(\mathbf{q})} \right) \quad (3.6)$$

L’utilisation de l’entropie croisée permet de favoriser les correspondances correctes tout en pénalisant les mauvaises correspondances. La Figure 3.7 présente l’évolution des fonctions de coût de SAM et du modèle de raffinement au cours de l’apprentissage.

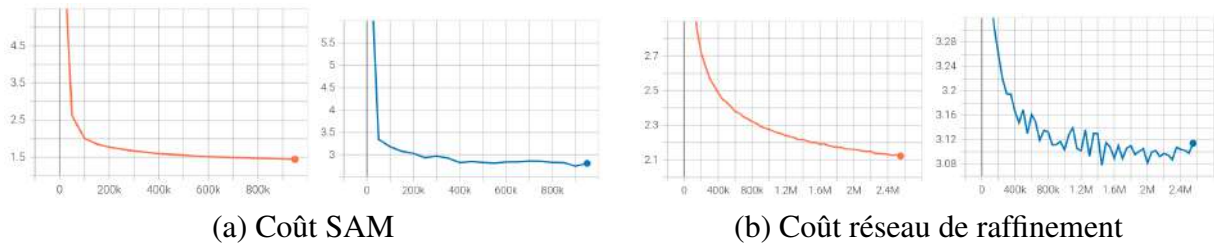


FIGURE 3.7 – Coût de training (orange) et validation (bleu) pour (a) SAM et (b) le module de raffinement. L’axe des abscisses représente le nombre de mini-batches vus par le modèle et l’axe des ordonnées la valeur de l’entropie croisée.

La première partie du modèle à être entraînée est le modèle de raffinement (CNN+FPN). Il est entraîné pendant 50 heures sur 4 GPU Nvidia V100, en utilisant l’optimiseur AdamW, un taux d’apprentissage constant de  $10^{-3}$  et une taille de batch de 1 (Figure 3.7 (b)). Nous minimisons l’entropie croisée des cartes de correspondance en pleine résolution produites par le FPN.

Le backbone CNN de SAM est initialisé avec les poids d’un réseau CNN siamois précédemment entraîné à faire de la mise en correspondance. Nous entraînons SAM pendant 100 heures. La même configuration est utilisée, comprenant 4 GPU Nvidia V100, l’optimiseur AdamW et une taille de batch de 1. Concernant l’évolution du taux d’apprentissage, nous appliquons une période de chauffe (*warm-up*) linéaire sur 5000 étapes (de 0 à  $10^{-4}$ ), suivi d’une décroissance exponentielle vers  $10^{-5}$  (Figure 3.7 (a)). Nous minimisons l’entropie croisée des cartes de correspondance de résolution  $\frac{1}{4}$  produites par SAM.

## 3.4 Experiences

Dans cette partie, nous présentons des résultats qualitatifs et quantitatifs pour notre méthode ainsi que pour les méthodes SDF les plus performantes. L’objectif est d’évaluer la précision de mise en correspondance, particulièrement dans les régions texturées, et de la comparer aux performances d’estimation de pose. Nous chercherons également à comprendre quelles parties de notre architecture conduisent à ces résultats, et nous nous intéresserons à un cas particulier de l’attention structurée.

### 3.4.1 Estimation de pose de caméra, homographie et séquence d'images

Dans ces expériences, nous nous concentrons sur l'évaluation de six réseaux SDF (LoFTR [Sun et al., 2021], QuadTree [Tang et al., 2022b], ASpanFormer [Chen et al., 2022], 3DG-STFM [Mao et al., 2022], MatchFormer [Wang et al., 2022], TopicFM [Giang et al., 2023]) ainsi que sur notre architecture proposée SAM. Dans tous les tableaux suivants, les meilleurs et les deuxièmes meilleurs résultats sont respectivement en gras et soulignés.

**Implémentations.** Pour chaque réseau, nous utilisons le code et les poids (entraînés sur MegaDepth) mis à disposition par les auteurs. En ce qui concerne SAM, nous l'entraînons de manière similaire aux réseaux SDF sur MegaDepth [Li and Snavely, 2018], pendant 100 heures, sur quatre GPU NVIDIA V100 (16GB).

**Positions de requête.** Concernant les méthodes SDF, les positions de requête sont définies comme les emplacements de la grille source avec un pas de 8. Ainsi, afin de pouvoir comparer les performances de SAM à ces méthodes, nous utilisons exactement les mêmes positions de requête pour calculer MA, ainsi que le sous-ensemble de positions situées dans des régions texturées pour calculer  $MA_{text}$ . Pour ce faire (rappelons que SAM a été entraîné avec  $L = 1024$  positions de requête), nous mélangeons simplement les emplacements de la grille source (avec un pas de 8) et les transmettons à SAM par lots de taille 1024 (les caractéristiques du CNN sont mises en cache, rendant efficace le traitement de chaque mini-lot).

#### 3.4.1.1 Estimation de pose de caméra, évaluation sur MegaDepth.

TABLE 3.2 – Évaluation sur MegaDepth1500 [Sarlin et al., 2020]. La méthode SAM proposée surpasse les méthodes SDF en termes d'estimation de pose, tandis que les méthodes SDF sont nettement meilleures en termes de MA. Cependant, lorsque les régions uniformes sont ignorées ( $MA_{text}$ ), SAM dépasse souvent les méthodes SDF. Ces résultats soulignent une forte corrélation entre la capacité à établir des correspondances précises dans les régions texturées et la précision de l'estimation de pose qui en résulte.

Méthode	Précision de matching (MA) $\uparrow$						Précision de matching régions texturées ( $MA_{text}$ ) $\uparrow$						Estimation de pose grille 1/8 (AUC) $\uparrow$		
	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=20$	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=20$	@5°	@10°	@20°
LoFTR	49.7	73.2	81.6	87.4	90.5	91.8	55.3	75.2	81.6	86.9	89.9	91.9	52.8	69.2	82.0
MatchFormer	51.1	73.4	81.0	86.9	89.5	90.9	56.5	75.6	81.8	87.1	89.6	90.9	52.9	69.7	82.0
TopicFM	51.4	75.4	83.7	89.9	92.9	93.5	59.8	77.6	84.8	90.4	92.9	93.7	54.1	70.1	81.6
3DG-STFM	51.6	73.7	80.7	86.4	89.0	90.7	57.0	75.8	81.8	86.8	88.8	90.5	52.6	68.5	80.0
ASpanFormer	<b>52.0</b>	<b>76.2</b>	<b>84.5</b>	<b>90.7</b>	<b>93.7</b>	<b>94.8</b>	<u>62.2</u>	<u>80.3</u>	<u>85.9</u>	<b>91.0</b>	<b>93.7</b>	<u>94.7</u>	<u>55.3</u>	<u>71.5</u>	<u>83.1</u>
LoFTR+QuadTree	<u>51.6</u>	<u>75.9</u>	<u>84.1</u>	<u>90.2</u>	<u>93.1</u>	<u>94.0</u>	61.7	79.9	85.5	90.5	93.3	94.1	54.6	70.5	82.2
<b>SAM</b>	48.5	70.4	78.0	83.0	85.4	86.4	<b>67.9</b>	<b>83.8</b>	<b>87.3</b>	<u>90.6</u>	<u>93.6</u>	<b>95.2</b>	<b>55.8</b>	<b>72.8</b>	<b>84.2</b>

Nous considérons le benchmark MegaDepth1500 [Sarlin et al., 2020]. Nous utilisons exactement les mêmes paramètres que ceux des méthodes SDF, avec une résolution d'image de 1200 pixels. Les résultats sont présentés dans le Tableau 3.2. Nous rapportons la précision de mise en correspondance (3.2) pour plusieurs seuils  $\eta$ , calculée sur l'ensemble des emplacements de requête semi-denses (grille source avec un pas de 8) ayant des correspondants de vérité terrain disponibles (MA). Les correspondants de vérité terrain sont obtenus à partir des cartes de profondeur et des poses de caméra disponibles. Par conséquent, de nombreux emplacements de requête situés dans des régions non texturées ont un correspondant de vérité terrain. Ainsi, nous

rapportons également la précision de mise en correspondance calculée uniquement sur les emplacements de requête situés dans une région texturée de l'image source ( $MA_{\text{text}}$ ). Concernant les métriques d'estimation de pose, nous rapportons les AUC classiques à 5, 10 et 20 degrés.

La méthode SAM proposée surpasse les méthodes SDF en termes d'estimation de pose, tandis que les méthodes SDF sont significativement meilleures en termes de MA. Cependant, lorsque les régions uniformes sont ignorées ( $MA_{\text{text}}$ ), SAM surpasse souvent les méthodes SDF. Ces résultats soulignent une forte corrélation entre la capacité à établir des correspondances précises dans les régions texturées et la précision de l'estimation de pose qui en résulte. Dans la Figure 3.10, nous présentons des résultats qualitatifs qui illustrent visuellement les observations précédentes.

### 3.4.1.2 Estimation d'homographie, évaluation sur HPatches.

Nous évaluons les différentes architectures sur HPatches [Balntas et al., 2017] (voir Tableau 3.3). Nous utilisons les mêmes paramètres que ceux des méthodes SDF. Nous présentons l'accuracy de mise en correspondance (3.2) pour plusieurs seuils  $\eta$ , calculée sur l'ensemble des emplacements de requête semi-denses (grille source avec un pas de 8) ayant des correspondants de vérité terrain disponibles (MA). Les correspondants de vérité terrain sont obtenus grâce aux matrices d'homographie disponibles. De ce fait, de nombreux emplacements de requête situés dans des régions non texturées ont un correspondant de vérité terrain. Nous rapportons donc également la précision de mise en correspondance calculée uniquement sur les emplacements de requête situés dans une région texturée de l'image source ( $MA_{\text{text}}$ ). Pour les métriques d'estimation d'homographie, nous rapportons les AUC classiques à 3, 5 et 10 pixels.

TABLE 3.3 – **Évaluation sur HPatches** [Balntas et al., 2017]. La méthode SAM proposée est équivalente aux méthodes SDF en termes d'estimation d'homographie, tandis que les méthodes SDF sont nettement meilleures en termes de MA. Cependant, lorsque les régions uniformes sont ignorées ( $MA_{\text{text}}$ ), SAM atteint les performances des méthodes SDF. Ces résultats soulignent une forte corrélation entre la capacité à établir des correspondances précises dans les régions texturées et la précision de l'homographie estimée qui en résulte.

Méthode	Précision de matching (MA) $\uparrow$			Précision de matching régions texturées ( $MA_{\text{text}}$ ) $\uparrow$			Estimation d'homographie (AUC) $\uparrow$		
	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=3$	$\eta=5$	$\eta=10$	@3px	@5px	@10px
LoFTR	66.8	74.3	77.3	67.6	75.3	78.4	65.9	75.6	84.6
MatchFormer	66.2	74.9	78.2	67.7	75.8	79.1	65.0	73.1	81.2
TopicFM	<u>72.7</u>	<u>85.0</u>	<u>87.5</u>	<b>74.0</b>	<u>86.0</u>	<u>88.5</u>	<u>67.3</u>	<b>77.0</b>	<u>85.7</u>
3DG-STFM	64.9	75.1	78.2	66.2	74.3	77.6	64.7	73.1	81.0
ASpanFormer	<b>76.2</b>	<b>86.2</b>	<b>88.7</b>	<u>73.9</u>	85.8	88.4	<b>67.4</b>	<u>76.9</u>	85.6
LoFTR+QuadTree	70.2	83.1	85.9	73.5	84.3	86.9	67.1	76.1	85.3
<b>SAM</b>	62.4	70.9	74.2	73.4	<b>86.6</b>	<b>89.3</b>	67.1	<u>76.9</u>	<b>85.9</b>

La méthode SAM proposée est comparable aux méthodes SDF en termes d'estimation d'homographie, tandis que les méthodes SDF sont significativement meilleures en termes de MA. Cependant, lorsque les régions uniformes sont ignorées ( $MA_{\text{text}}$ ), SAM égale les performances des méthodes SDF. Ces résultats mettent en évidence une forte corrélation entre la capacité à établir des correspondances précises dans les régions texturées et la précision de l'homographie estimée qui en découle. Dans la Figure 3.11, nous présentons des résultats qualitatifs qui illustrent visuellement ces conclusions.

### 3.4.1.3 Matching dans une séquence d'images, évaluation sur ETH3D.

Nous évaluons les différents réseaux sur plusieurs séquences du jeu de données ETH3D [Schöps et al., 2019], comme proposé dans [Truong et al., 2020]. Différents taux d'échantillonnage d'intervalle de trames  $r$  sont considérés. À mesure que le taux  $r$  augmente, le recouvrement entre les paires d'images diminue, rendant ainsi le problème de mise en correspondance plus difficile. Les résultats sont présentés dans le Tableau 3.4. Nous rapportons la précision de mise en correspondance (3.2) pour plusieurs seuils  $\eta$ , calculée sur l'ensemble des emplacements de requête semi-denses (grille source avec un pas de 8) ayant des correspondants de vérité terrain disponibles (MA).

TABLE 3.4 – Évaluation sur ETH3D [Schöps et al., 2019]. Comme les paires sont issues de séquences d'images, nous rapportons les résultats pour différents intervalles de d'images  $r$ . Pour ETH3D, les correspondants de vérité terrain sont basés sur les trajectoires obtenues par *structure from motion*. Par conséquent, le MA ignore déjà les régions non texturées des images sources, ce qui explique pourquoi SAM parvient à surpasser les méthodes SDF.

Méthode	Précision de matching (MA) $\uparrow$														
	$r = 3$					$r = 7$					$r = 15$				
	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$
LoFTR	44.8	76.5	88.4	97.0	99.4	39.7	73.1	87.6	95.9	98.5	33.3	66.2	84.8	92.5	96.3
MatchFormer	45.5	77.1	89.2	97.2	99.7	40.4	73.8	87.8	96.6	99.0	34.2	66.7	84.9	93.5	97.0
TopicFM	45.1	76.9	89.0	97.2	99.6	39.9	73.5	87.9	96.4	99.0	33.8	66.4	85.0	92.8	96.5
3DG-STFM	43.9	76.3	88.0	96.9	99.3	39.3	72.7	87.4	95.5	98.3	32.4	65.7	84.7	92.0	96.0
ASpanFormer	45.8	77.6	89.6	97.8	<b>99.8</b>	40.6	73.8	88.1	96.8	99.0	34.3	66.8	85.3	93.9	97.3
LoFTR+QuadTree	45.9	77.5	89.5	97.8	99.7	40.8	74.0	88.3	97.0	99.2	34.5	66.8	85.4	94.0	97.3
<b>SAM</b>	<b>53.4</b>	<b>79.9</b>	<b>91.5</b>	<b>98.0</b>	<b>99.8</b>	<b>48.6</b>	<b>78.6</b>	<b>91.7</b>	<b>98.2</b>	<b>99.4</b>	<b>40.1</b>	<b>70.2</b>	<b>87.8</b>	<b>95.4</b>	<b>97.8</b>

Pour ETH3D, les correspondants de vérité terrain sont basés sur les trajectoires obtenues par un algorithme d'acquisition de structure à partir d'un mouvement (*structure from motion* ou SfM). Par conséquent, le MA ignore les régions non texturées des images sources, ce qui explique pourquoi SAM parvient à surpasser les méthodes SDF en termes de MA. Dans la Figure 3.12, nous présentons des résultats qualitatifs qui illustrent la précision de la méthode SAM proposée.

**Discussion des résultats.** Les résultats des expériences montrent que les méthodes semi-denses sans détecteur (SDF) surpassent SAM en termes de précision globale de mise en correspondance, notamment grâce à leur capacité à établir des correspondances dans des régions uniformes ou faiblement texturées. Cependant, dans les régions texturées, SAM surpasse les méthodes SDF en termes de précision de mise en correspondance. Cela montre que SAM est particulièrement efficace dans les zones riches en détails visuels, où la fiabilité des correspondances est essentielle. De plus, lorsque l'on se concentre uniquement sur l'estimation de pose de caméra, qui repose sur les mêmes points de correspondance pour SAM et les méthodes SDF, SAM atteint ou dépasse souvent les performances des méthodes SDF. Cela souligne une forte corrélation entre la capacité à établir des correspondances précises dans les régions texturées et la qualité de l'estimation de pose qui en résulte. Les méthodes SDF exploitent des zones plus larges des images grâce au matching dense grossier, que ce soit pendant l'entraînement et l'évaluation, ce qui pourrait expliquer leur meilleure précision globale. D'un autre côté, SAM, bien que travaillant avec une représentation dense de l'image cible, ne possède qu'une représentation locale de la source à travers les positions de requête calculées par lots de 1024. Ceci ne semble pas poser de problème dans les régions riches en détails visuels où SAM montre une



supériorité en précision de matching, mais semble important pour la précision dans les régions non texturées. Cette distinction est cruciale, car elle reflète les compromis entre la densité des correspondances et leur précision, un aspect directement lié à la problématique de l'estimation de pose de caméra. Alors que LoFTR supposait que le gain en estimation de pose venait de la précision dans les régions uniformes, SAM montre que l'on peut concevoir une méthode produisant des correspondances très précises dans les régions texturées, moins précises dans les régions uniformes, et atteindre des résultats d'estimation de pose comparables voire supérieurs.

### 3.4.2 Étude d'ablation

TABLE 3.5 – **Étude d'ablation de la méthode SAM proposée.** Les résultats sont calculés sur notre ensemble de validation MegaDepth (10 scènes). Nous partons d'un réseau siamois en ajoutant progressivement les éléments de SAM pour analyser leur impact sur les résultats.

Méthode	MA $\uparrow$					
	$\eta=1$	$\eta=2$	$\eta=5$	$\eta=10$	$\eta=20$	$\eta=50$
CNN siamois	0.029	0.112	0.436	0.581	0.622	0.687
+ AC entrée et AA (x16)	0.137	0.321	0.671	0.734	0.767	0.822
+ espace latent et AC sortie	0.132	0.462	0.823	0.871	0.898	0.935
+ PE concaténé	0.121	0.419	0.796	0.868	0.902	0.939
+ attention structurée	<u>0.140</u>	<u>0.487</u>	<u>0.857</u>	<u>0.902</u>	<b>0.922</b>	<b>0.947</b>
+ Raffinement ( <b>modèle complet</b> )	<b>0.673</b>	<b>0.791</b>	<b>0.870</b>	<b>0.902</b>	<u>0.921</u>	<u>0.946</u>

Dans le Tableau 3.5, nous proposons une étude d'ablation de SAM afin d'évaluer l'impact de chaque composant de notre architecture. Cette étude est réalisée sur les scènes de validation de MegaDepth. En partant d'un CNN siamois standard, nous montrons qu'un gain significatif de performance peut être obtenu simplement en ajoutant la couche d'attention-croisée (AC) en entrée (ici, l'encodage positionnel est ajouté et non concaténé) et les couches d'auto-attention (AA). Nous ajoutons ensuite des vecteurs latents appris (LV) dans l'espace latent et utilisons une cross-attention en sortie, ce qui améliore à nouveau significativement les performances. La concaténation des informations d'encodage positionnel au lieu de leur addition aux caractéristiques visuelles réduit la précision de mise en correspondance à  $\eta = 2$  et  $\eta = 5$ . Cependant, en la combinant avec l'attention structurée, on observe une amélioration significative en termes de MA. Enfin, comme prévu, l'étape de raffinement améliore la précision de mise en correspondance pour les faibles seuils d'erreur en pixels.

### 3.4.3 Étude de l'attention structurée

Ici, nous voulons explorer les capacités de l'*attention structurée*. Pour rappel, l'attention structurée consiste en une structuration des matrices de poids  $W_{o,h}$  et  $W_{v,h}$  afin de conserver une représentation purement positionnelle, mise à jour par l'attention, en parallèle de la représentation visuo-positionnelle de haut niveau (Figure 3.8 (a)). Ces matrices de poids peuvent être structurées de différentes manières pour atteindre différents comportements. Schématisé dans la Figure 3.8 (b), nous pouvons structurer nos matrices pour totalement dissocier l'information visuelle de l'information positionnelle dans le réseau :

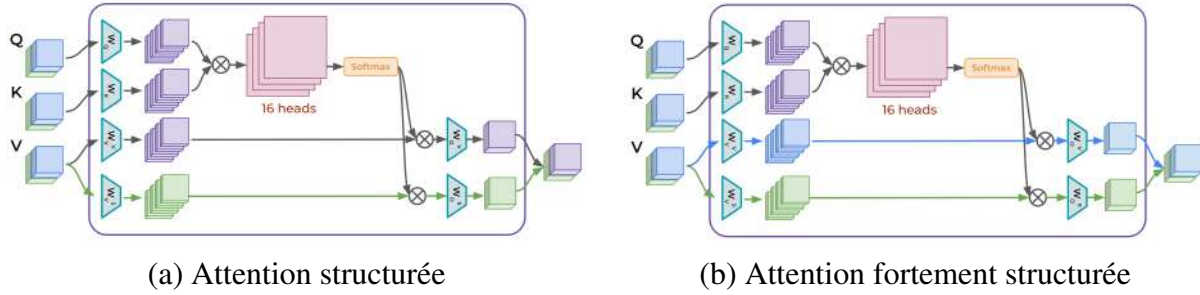


FIGURE 3.8 – **Comparaison entre l'Attention structurée (a) et une version sans représentation visuo-positionnelle de haut niveau (b).** Les volumes verts représentent des descripteurs contenant uniquement de l'information visuelle, les bleus uniquement de l'information positionnelle, et les violets un mélange d'information visuo-positionnelle.

$$\underbrace{W_{o,h}}_{D \times D_H} = \left[ \begin{array}{c|c} \underbrace{W_{o,h,up}}_{\frac{D}{2} \times \frac{D_H}{2}} & \underbrace{0}_{\frac{D}{2} \times \frac{D_H}{2}} \\ \hline \underbrace{0}_{\frac{D}{2} \times \frac{D_H}{2}} & \underbrace{W_{o,h,low}}_{\frac{D}{2} \times \frac{D_H}{2}} \end{array} \right]. \quad (3.7)$$

Cette *attention fortement structurée* mélange les informations visuelles et positionnelles uniquement pour le calcul des cartes d'attention. Les poids ainsi calculés sont utilisés pour mettre à jour séparément les deux représentations. Cette méthode pourrait faire penser à certaines techniques d'encodage positionnel relatif comme [Ho et al., 2019], mais au lieu d'ajouter une information constante de position aux cartes d'attention, ici nous faisons évoluer l'information de position à travers le réseau. Nous obtenons donc des caractéristiques purement visuelles de haut niveau ainsi que des caractéristiques purement positionnelles de haut niveau. Contre-intuitivement, les performances de cette *attention fortement structurée* ne sont que légèrement inférieures à celles de l'*attention structurée* (tableau 3.6), alors qu'aucune représentation visuo-positionnelle de haut niveau n'est construite au fur et à mesure dans le réseau.

TABLE 3.6 – **Résultats comparatifs entre attention structurée et attention fortement structurée.** Les résultats sont calculés sur notre ensemble de validation MegaDepth (10 scènes), sans module de raffinement.

Méthode	Précision de matching (MA) ↑					
	$\eta=1$	$\eta=2$	$\eta=5$	$\eta=10$	$\eta=20$	$\eta=50$
Attention structurée	0.140	0.487	0.857	0.902	0.922	0.947
Attention fortement structurée	0.138	0.478	0.849	0.897	0.920	0.947

### 3.4.4 Évolution de SAM

Nous avons trouvé que notre architecture SAM pouvait avoir des performances d'estimation de pose de caméra similaires à celles des méthodes SDF, sans établir de correspondances précises pour les points situés dans les régions homogènes. Mais alors, comment expliquer le gain de performance entre les méthodes S2S et SDF? Un point commun que SAM partage avec les

SDF est une plus grande prise en compte du contexte des images par rapport aux méthodes S2S. En effet, les méthodes S2S se limitent à une représentation des images par les descripteurs locaux des points d'intérêt. Les SDF, de leur côté, bénéficient d'une représentation dense à basse résolution des deux images. SAM se situe entre les deux, avec des descripteurs de requêtes pour l'image source et une représentation dense pour l'image cible.

TABLE 3.7 – **Variation du nombre de requêtes pour SAM.** Les résultats sont calculés sur notre ensemble de validation MegaDepth (10 scènes).

Méthode		Précision de matching $\uparrow$					
		$\eta=1$	$\eta=2$	$\eta=5$	$\eta=10$	$\eta=20$	$\eta=50$
SAM	32 requêtes	0.647	0.765	0.846	0.880	0.907	0.936
SAM	256 requêtes	0.651	0.774	0.856	0.888	0.911	0.938
SAM	1024 requêtes	0.673	0.791	0.870	0.902	0.921	0.946

Pour voir si le contexte est important pour une mise en correspondance précise, nous limitons le nombre de points requêtes pour SAM. Les résultats sont présentés dans le Tableau 3.8. En donnant moins de points requêtes à SAM, nous limitons son contexte de l'image source. Bien que les performances restent bonnes, la diminution du contexte de l'image source fait baisser de plusieurs points la précision de matching de SAM.

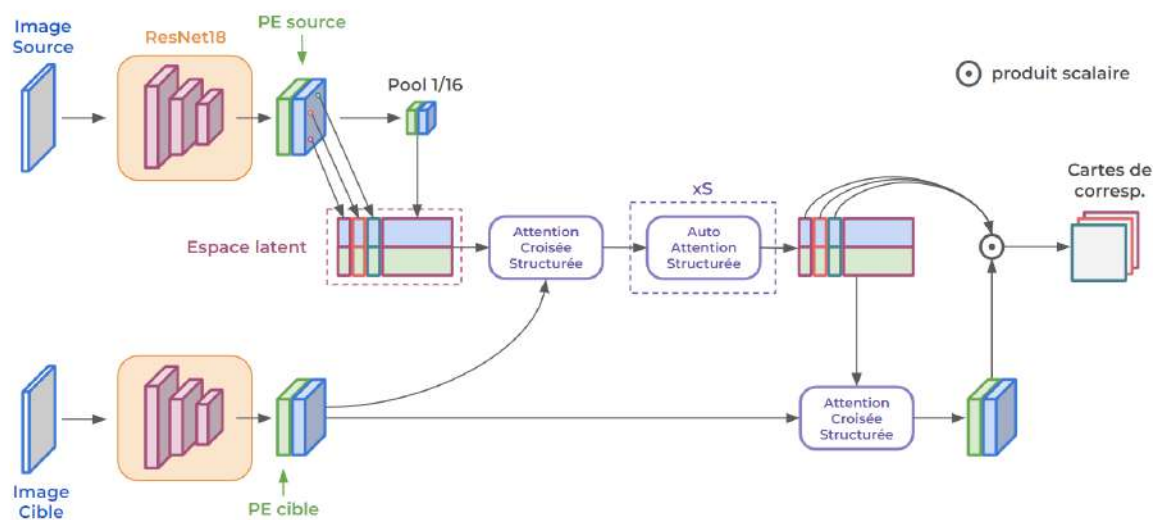


FIGURE 3.9 – **Évolution de l'architecture SAM.** Ajout du contexte de l'image source.

Pour intégrer de manière plus homogène le contexte de la source, nous modifions l'architecture de SAM en remplaçant les vecteurs appris de l'espace latent par un pooling des cartes de descripteurs de l'image source (schématisé par la Figure 3.9). La communication à travers l'attention structurée et la création des cartes de correspondances restent inchangées. Cependant, le pooling  $\frac{1}{16}$  produisant plus de descripteurs que le nombre de vecteurs appris que possédait SAM, cette architecture devient significativement plus coûteuse en mémoire, et il n'est possible d'avoir que 32 points de requête, la rendant difficilement utilisable.

TABLE 3.8 – **Résultats comparatif SAM sans et avec contexte de la source.** Les résultats sont calculés sur notre ensemble de validation MegaDepth (10 scènes).

Méthode	Précision de matching $\uparrow$					
	$\eta=1$	$\eta=2$	$\eta=5$	$\eta=10$	$\eta=20$	$\eta=50$
SAM sans contexte source	0.673	0.791	0.870	0.902	0.921	0.946
SAM avec contexte source	0.678	0.795	0.871	0.899	0.917	0.940

Ajouter le contexte de l'image source améliore légèrement la précision de la mise en correspondance (Tableau 3.8). En revanche, retirer les vecteurs appris de notre architecture semble diminuer sa robustesse pour des seuils d'erreur plus élevés ( $\eta$ ). Ces résultats très proches ne nous permettent pas d'établir une conclusion certaine ; néanmoins, la prise en compte d'un contexte plus large et plus précis nous semble être un axe de recherche intéressant. Nous aborderons ceci plus en détail dans le chapitre 4 sur la mise en correspondance dense d'images.

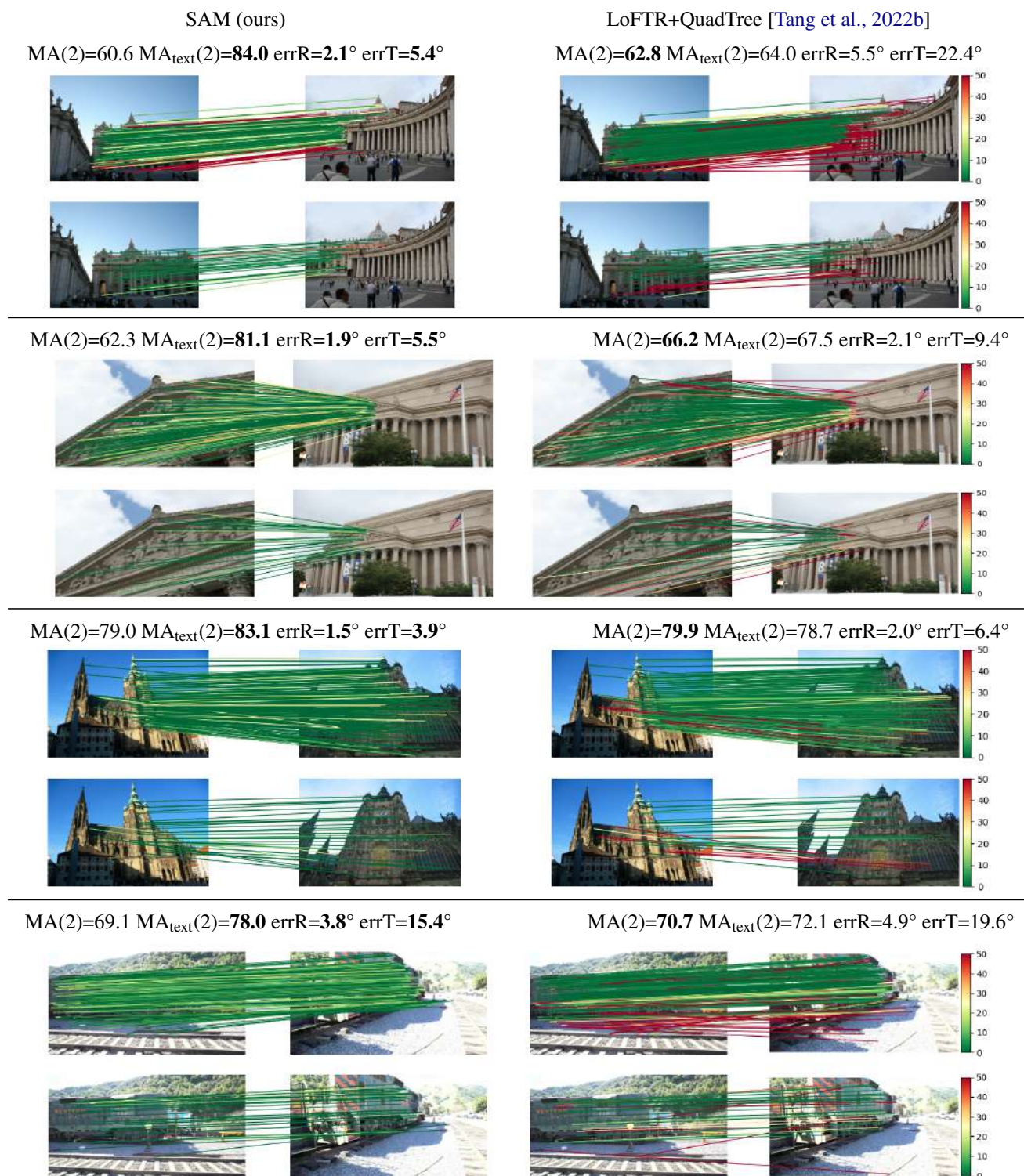


FIGURE 3.10 – **Résultats qualitatifs sur MegaDepth1500.** Pour chaque paire d'images : (ligne du haut) Visualisation des correspondances établies utilisées pour calculer le MA, (ligne du bas) Visualisation des correspondances établies utilisées pour calculer le  $MA_{\text{text}}$ . Les couleurs des lignes indiquent la distance en pixels par rapport au correspondant de vérité terrain.



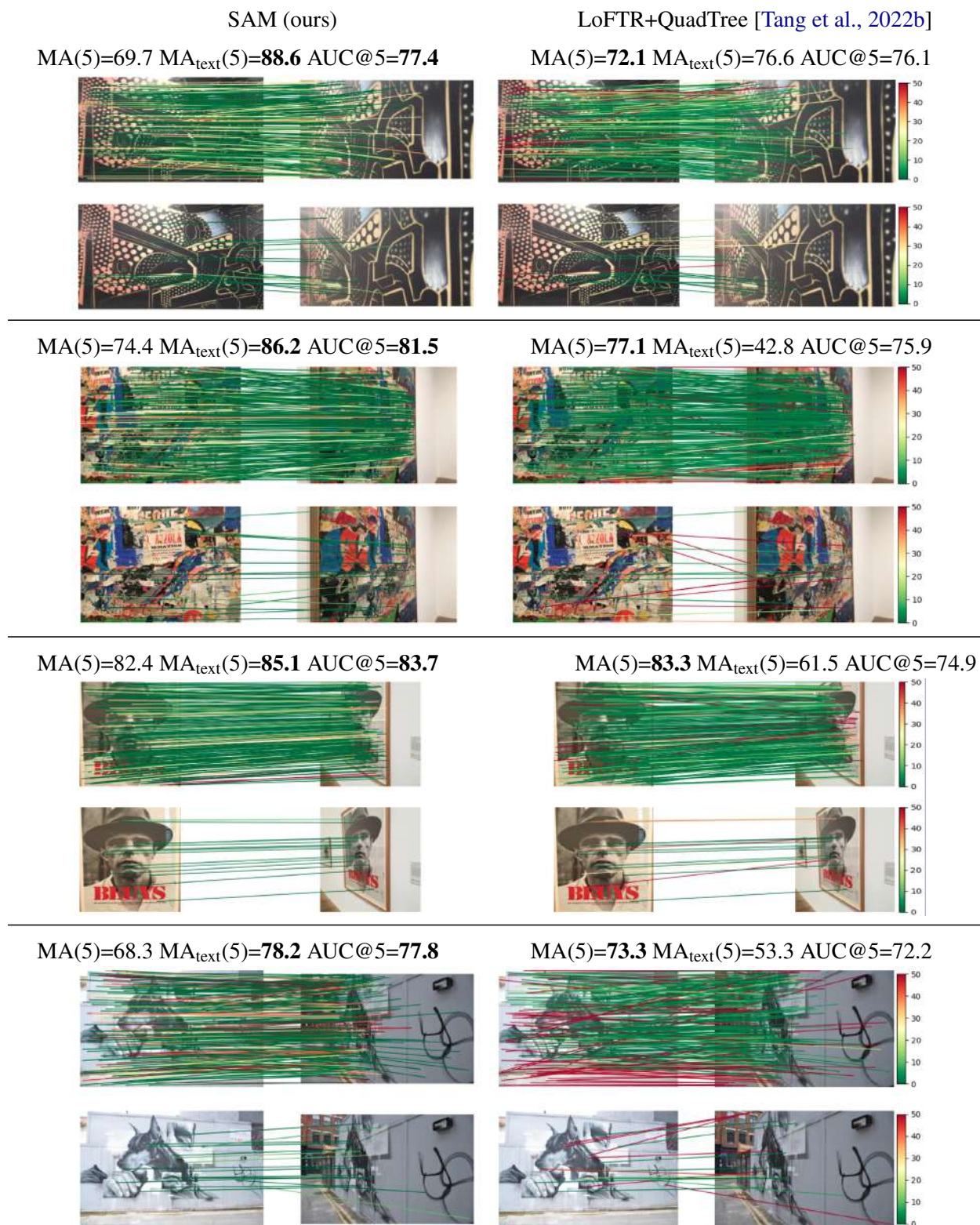


FIGURE 3.11 – **Résultats qualitatifs sur HPatches.** Les couleurs des lignes indiquent la distance en pixels par rapport au correspondant de vérité terrain.



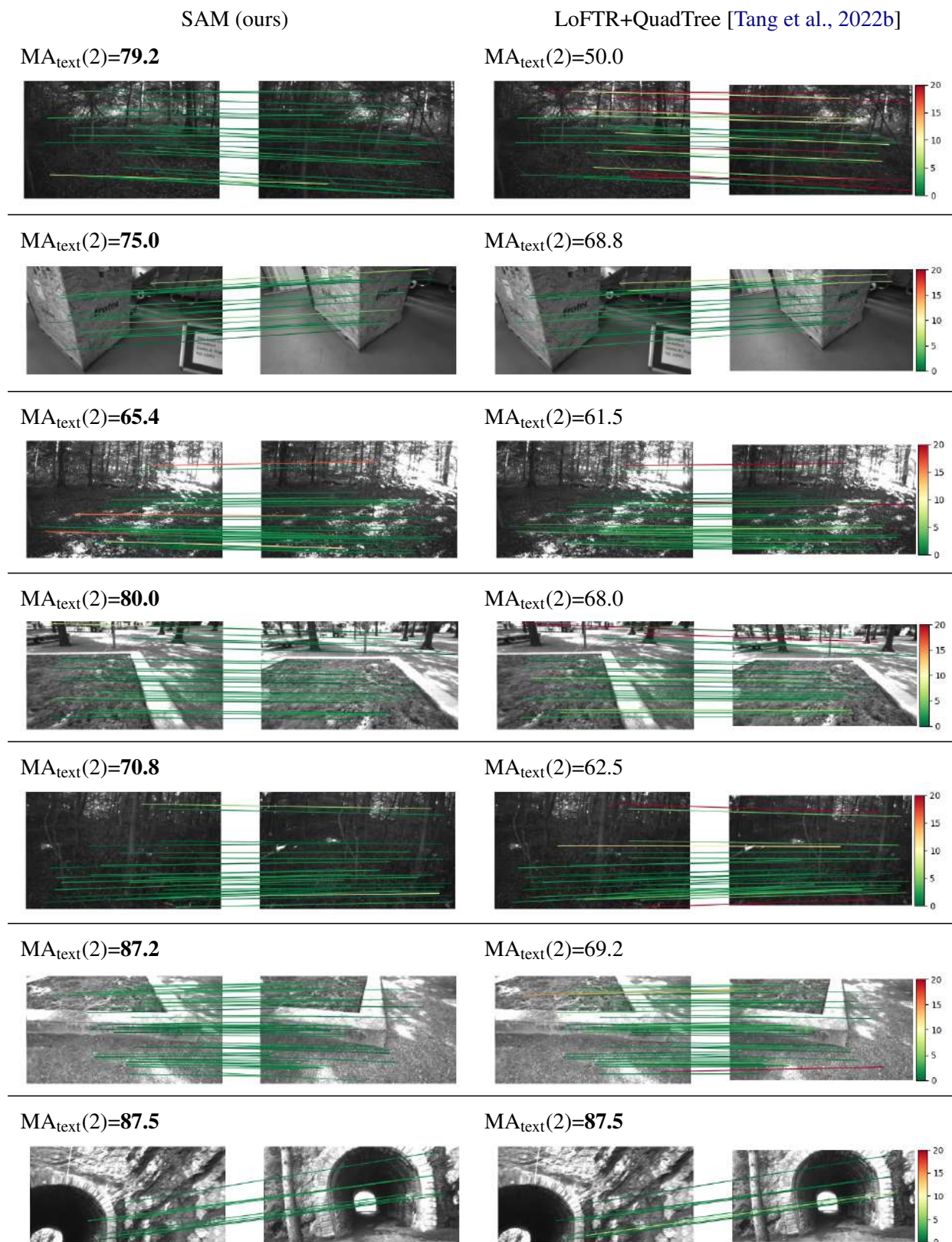


FIGURE 3.12 – **Résultats qualitatifs sur ETH3D.** Les couleurs des lignes indiquent la distance en pixels par rapport au correspondant de vérité terrain.

### 3.5 Conclusion

Les méthodes semi-denses sans détecteurs (SDF), comme LoFTR, sont devenues des références pour la mise en correspondance d'images en raison de leurs performances supérieures en estimation de pose de caméra par rapport aux méthodes éparses (S2S). Toutefois, jusqu'à présent, l'évaluation de ces méthodes s'est concentrée principalement sur cette métrique d'estimation de pose, sans prendre en compte de manière approfondie le lien direct entre la précision de la mise en correspondance et la qualité des poses estimées. Cette lacune dans la littérature existante a motivé notre exploration approfondie de cette relation cruciale.

Dans ce travail, nous avons proposé une nouvelle architecture de mise en correspondance d'images, basée sur l'attention structurée (SAM), conçue pour évaluer de manière plus fine la capacité des méthodes de matching à établir des correspondances précises pour tout type de points. Nos résultats ont révélé un constat surprenant : en centrant l'analyse sur les régions texturées, SAM démontre une précision supérieure des correspondances par rapport aux méthodes SDF, alors que ces dernières se montrent généralement plus performantes en termes de précision globale. SAM parvient pourtant à égaler, voire surpasser, les méthodes SDF en estimation de pose de caméra, ce qui remet en question l'hypothèse selon laquelle la capacité des méthodes SDF à établir des correspondances précises dans les régions uniformes serait la cause de leurs bonnes performances en estimation de pose.

La forte corrélation observée entre la précision du matching dans les régions texturées et la qualité des poses estimées suggère qu'il ne semble pas nécessaire d'établir des correspondances précises dans les régions homogènes pour obtenir une bonne estimation de pose de caméra. Cependant, la prise en compte d'un contexte plus global des deux images semble être un ingrédient clé des performances des méthodes SDF. Notre étude ouvre également des perspectives sur l'importance de l'évaluation de la qualité des correspondances dans des contextes plus spécifiques et non seulement sur des métriques proxy, comme l'estimation de pose de caméra.

Le passage des méthodes semi-denses aux méthodes entièrement denses, comme DKM [Edstedt et al., 2023], représente une évolution prometteuse dans le domaine de la mise en correspondance d'images. Les méthodes denses, qui exploitent l'intégralité des informations visuelles présentes dans chaque pixel d'une image, pourraient surmonter certaines des limitations des approches semi-denses en offrant une prise en compte complète et précise du contexte des deux images. Cependant, cette transition vers des méthodes denses pose des défis majeurs en termes de complexité computationnelle.

## **Chapitre 4**

### **Vers une mise en correspondance d'images dense**

## Table des matières

4.1	Introduction . . . . .	81
4.2	État de l'art . . . . .	82
4.2.1	L'approche régressive de DKM . . . . .	82
4.2.2	Classification en cascade avec CasMTR . . . . .	84
4.2.3	Discussion . . . . .	85
4.3	Matching dense par recherche en faisceaux . . . . .	86
4.3.1	Problème de multimodalité . . . . .	87
4.3.1.1	Validation expérimentale . . . . .	88
4.3.1.2	Recherche en faisceaux . . . . .	89
	Motivation. . . . .	89
	Matching <i>grossier à fin</i> . . . . .	89
	Recherche en faisceaux. . . . .	91
4.3.1.3	Beam-attention . . . . .	92
4.3.2	Prédiction de la covisibilité . . . . .	93
4.3.2.1	Problème de l'échantillonnage de correspondances . . . . .	93
4.3.2.2	Prédire la covisibilité entre deux images . . . . .	95
	Prédiction de la covisibilité. . . . .	95
	Communication dans le réseau. . . . .	97
4.3.3	Notre architecture BEAMER . . . . .	97
4.3.4	Étape d'entraînement . . . . .	99
4.4	Expériences . . . . .	102
4.4.1	Analyse de l'architecture . . . . .	102
4.4.1.1	Visualisation de la recherche en faisceaux . . . . .	102
4.4.1.2	Analyse de la prédiction de covisibilité . . . . .	102
4.4.1.3	Stratégies d'échantillonnage . . . . .	105
4.4.1.4	Étude du coût . . . . .	106
4.4.1.5	Études d'ablations . . . . .	106
4.4.2	Estimation de pose de caméra et précision de correspondance . . . . .	108
4.4.2.1	Résultats sur MegaDepth . . . . .	108
4.4.2.2	Résultats sur HPatches . . . . .	112
4.4.2.3	Résultats sur ETH3D . . . . .	115
4.4.2.4	Discussion des résultats . . . . .	115
4.5	Conclusion . . . . .	118

## 4.1 Introduction

Les méthodes semi-denses se sont imposées comme des solutions fiables pour la mise en correspondance d'images, et ont dominé à leur introduction les autres approches en estimation de pose de caméra. Elles reposent sur une approche *grossier puis fin*, où des correspondances grossières sont établies puis raffinées à pleine résolution. Toutefois, avec cette approche elles n'établissent qu'une correspondance fine par région grossière, omettant ainsi une grande partie des correspondances fines. Or, ces correspondances manquantes peuvent jouer un rôle essentiel dans l'amélioration de la précision de l'estimation de pose, notamment dans les régions où les détails fins sont déterminants. La mise en correspondance dense (*Dense Image Matching* ou DIM), longtemps mise de côté en raison de sa complexité mémoire et de son coût computationnel élevé, a récemment surpassée les autres méthodes de l'état de l'art en estimation de pose de caméra. Contrairement aux méthodes semi-denses, le matching dense permet d'établir des correspondances pour chaque pixel des images à pleine résolution, capturant ainsi les détails les plus fins.

Cependant, l'introduction du paradigme dense soulève de nouveaux défis. Le premier est le coût mémoire nécessaire pour établir des correspondances pour tous les pixels. La création de cartes de correspondances n'est pas triviale ; les calculer de manière dense pour chaque pixel directement à pleine résolution nécessiterait trop de mémoire. Les méthodes denses utilisent une stratégie *grossier à fin*, où des correspondances grossières sont d'abord établies, puis elles sont progressivement raffinées et densifiées à des résolutions plus élevées jusqu'à atteindre la pleine résolution. L'utilisation de cette stratégie *grossier à fin*, contrairement à la stratégie *grossier puis fin* des méthodes semi-denses, pose un nouveau problème de multimodalité des correspondances que nous détaillerons dans ce chapitre. Un autre défi lié au paradigme dense est l'échantillonnage des correspondances finales. En effet, les méthodes denses produisent des correspondances pour tous les pixels des images, mais certains de ces pixels peuvent être occultés ou ne pas apparaître dans la seconde image. Il est donc nécessaire d'établir une stratégie pour filtrer ces points non covisibles que nous ne voulons pas utiliser pour l'estimation de pose de caméra.

L'architecture BEAMER, introduite dans ce chapitre, répond à ces défis en adoptant une approche basée sur la recherche en faisceaux (*beam search*). Ce mécanisme permet d'affiner progressivement les correspondances en se concentrant sur les zones les plus prometteuses. Cette formulation du matching dense permet de réduire intelligemment le volume de correspondances à traiter et permet ainsi d'utiliser de l'attention à tous les niveaux de l'architecture de manière dense ou éparse. En intégrant également la prédiction de la covisibilité entre les images, BEAMER facilite l'échantillonnage pour assurer une bonne répartition spatiale des correspondances.

Dans ce chapitre, nous commencerons par aborder les nouvelles problématiques liées au paradigme dense à travers l'étude de deux méthodes de l'état de l'art. Puis, nous formaliserons le problème de multimodalité et détaillerons notre solution à travers la recherche en faisceaux. Nous verrons ensuite comment échantillonner efficacement des correspondances en intégrant la prédiction de la covisibilité. Enfin, nous présenterons l'architecture finale de BEAMER et évaluerons ses performances en mise en correspondance d'images ainsi qu'en estimation de pose de caméra.

## 4.2 État de l'art

La mise en correspondance d'images dense (DIM) est utilisée depuis l'avènement des réseaux de neurones convolutifs pour des tâches comme le matching d'images stéréo [Chang and Chen, 2018, Xu and Zhang, 2020, Zhang et al., 2021, Lipson et al., 2021, Li et al., 2022, Xu et al., 2022a] ou l'estimation de flux optique [Dosovitskiy et al., 2015, Ilg et al., 2017, Sun et al., 2018, Teed and Deng, 2020, Xu et al., 2022b, Huang et al., 2022, Sui et al., 2022, Shi et al., 2023]. Pour ces tâches de mise en correspondance où la transformation entre les deux images n'est pas très importante, l'utilisation d'architectures fortement basées sur des réseaux de convolution est suffisamment précise et peu coûteuse. Pour des tâches comme l'estimation de pose de caméra pour faire de la reconstruction 3D, par exemple, les transformations entre les images peuvent être beaucoup plus complexes et nécessitent des architectures plus avancées et plus coûteuses. D'autres paradigmes comme le matching éparsé ou semi-dense, naturellement moins coûteux et permettant de développer des architectures plus complexes, ont donc pendant longtemps été préférés au paradigme dense. Comme nous l'avons vu dans la section 2.2.2.3, récemment, DKM [Edstedt et al., 2023] et d'autres méthodes ont proposé des architectures denses avec d'excellentes performances en estimation de pose de caméra. La plupart des approches denses [Melekhov et al., 2019, Truong et al., 2020, Truong et al., 2021b, Truong et al., 2023, Zhu and Liu, 2023, Ni et al., 2023, Edstedt et al., 2023] considèrent le problème comme un *warping* dense et le présentent comme une série de tâches de régression *grossière à fine*. Le *warping* dans le contexte du matching dense d'images fait référence à une transformation non rigide appliquée à une image pour aligner ou faire correspondre des caractéristiques entre les deux images. Il s'agit de modifier l'image source de manière à ce qu'elle corresponde spatialement à l'image cible. Cette approche par régression est différente de l'approche par classification, basée sur des cartes de correspondances, qui a prouvé être très efficace pour la mise en correspondance à travers les méthodes semi-denses. Utiliser des cartes de correspondances dans le paradigme dense n'est pas trivial ; leur calcul dense à haute résolution est beaucoup trop coûteux en mémoire. Parallèlement aux travaux qui seront présentés dans ce chapitre, CasMTR [Cao and Fu, 2023] a commencé à étudier l'utilisation de cartes de correspondances de manière *grossier à fin* et propose une méthode permettant de produire des correspondances denses précises à la moitié de la résolution des images d'origine.

La description des principales méthodes de matching dense ayant été faite dans la section 2.2.2.3, nous nous concentrerons ici sur l'analyse de deux méthodes : DKM [Edstedt et al., 2023], car elle fut la première méthode dense à proposer des performances supérieures aux approches semi-denses et nous servira de point de comparaison tout au long du chapitre ; puis CasMTR [Cao and Fu, 2023], car, même si elle ne propose pas un matching entièrement dense, elle explore la mise en correspondance *grossière à fine* comme une tâche de classification de manière similaire aux méthodes semi-denses.

### 4.2.1 L'approche régressive de DKM

DKM (*Dense Kernelized Feature Matching*) [Edstedt et al., 2023] repose sur une mise en correspondance *grossière à fine* par régression, schématisée dans la Figure 4.1. Dans un premier temps, une pyramide de caractéristiques multi-échelles est extraite des images d'entrée en utilisant un ResNet. Les caractéristiques grossières sont ensuite utilisées par leur matcheur global, qui exploite un processus gaussien, pour déterminer des *warps* ainsi qu'une estimation de la certitude de ces *warps*. Le *warp* de l'image source est l'ensemble des coordonnées des correspondances dans la cible pour toutes les régions grossières de la source. Ces *warps* sont



ensuite utilisés pour créer une nouvelle représentation grossière de la source. Ces représentations se composent des caractéristiques de l'image concaténées aux *warps* de l'autre image, à l'estimation de certitude et à un encodage des positions. Ces représentations grossières vont ensuite être successivement raffinées jusqu'à la résolution originale des images. Pour ce faire, les représentations sont suréchantillonnées pour obtenir des représentations deux fois plus grandes, passées à travers plusieurs couches de convolutions, puis décodées pour obtenir de nouveaux *warps* et une nouvelle estimation de la certitude. Ce processus est répété jusqu'à atteindre la résolution d'origine. L'objectif est d'améliorer progressivement la précision des correspondances en utilisant des informations plus locales.

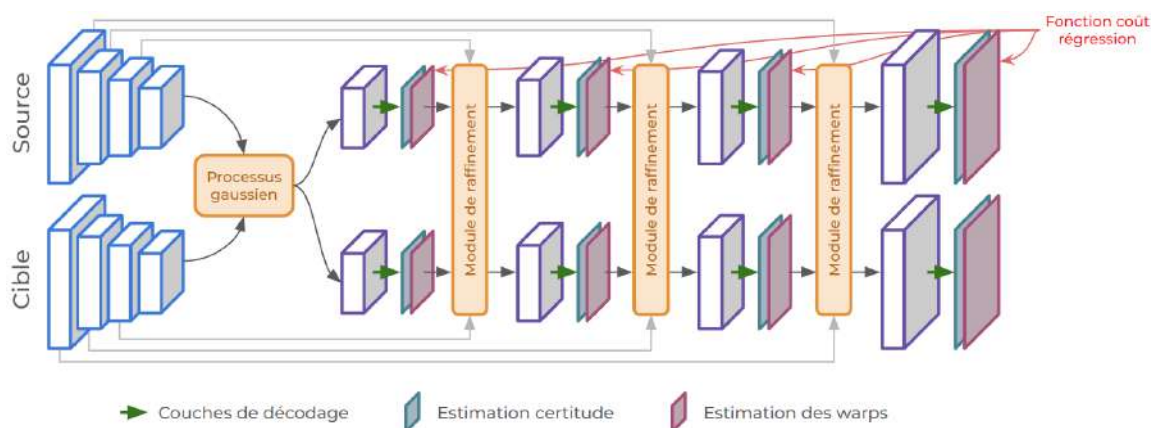


FIGURE 4.1 – **Représentation schématique de l'architecture de DKM.** DKM est construit comme une succession de tâches de régression pour établir les correspondances.

Les *warps* finaux obtenus contiennent donc des correspondances pour tous les pixels des images. Le réseau produit également des cartes de confiance pour toutes ces correspondances. Les correspondances sont entraînées via une régression sur tous les points covisibles disponibles dans la vérité terrain. Pour entraîner la prédiction de confiance du réseau, une entropie croisée est utilisée, où tous les points covisibles dans la vérité terrain sont définis comme confiants et tous les autres comme non-confiants.

Cette estimation de la confiance est cruciale dans le cadre du matching dense. En effet, DKM est capable de déterminer des correspondances pour tous les pixels des deux images. Cependant, certaines régions des images ne sont pas covisibles ; l'estimation de la confiance permet donc au réseau d'écarter ces régions. Lors de l'inférence du réseau, ces cartes de confiance sont utilisées pour pondérer l'échantillonnage des correspondances qui seront utilisées par l'algorithme RANSAC pour l'estimation de la pose de caméra.

Les résultats expérimentaux démontrent que DKM surpasse les méthodes semi-denses sur plusieurs benchmarks d'estimation de pose de caméra. Sur le benchmark MegaDepth-1500, DKM améliore la performance de +5,1 AUC@5° par rapport à la meilleure méthode semi-dense et de +18,2 AUC@5° par rapport à la meilleure méthode éparsée. Les auteurs soulignent cependant quelques limitations. Le raffinement des *warps* est unimodal, ce qui pose problème lorsqu'il existe des discontinuités dans le *warp*. Cela peut entraîner une perte de précision dans ces zones critiques. De plus, la méthode tend à être trop incertaine dans certaines régions, telles que les petits objets près du ciel. Malgré cela, DKM montre une grande robustesse et prouve l'intérêt du paradigme dense pour des tâches complexes comme l'estimation de pose.

### 4.2.2 Classification en cascade avec CasMTR

CasMTR (Cascade feature Matching Transformer) [Cao and Fu, 2023] reprend la méthodologie des approches semi-denses comme LoFTR [Sun et al., 2021] en formulant la tâche de matching comme une tâche de classification sur des cartes de correspondances, en l'adaptant à l'approche *grossier à fin* à travers une mise en correspondance en cascade (voir Figure 4.2). Dans un premier temps, une pyramide de caractéristiques multi-échelles est extraite des images d'entrée en utilisant un ResNet. De manière analogue à LoFTR, CasMTR utilise une succession de couches d'auto-attention et d'attention-croisée pour créer de la communication entre les caractéristiques grossières des deux images, puis calcule des cartes de correspondances par un produit scalaire entre les caractéristiques grossières mises à jour de la source et de la cible. Le cœur de CasMTR réside dans ses modules de mise en correspondance en cascade, conçus pour améliorer la précision des correspondances de manière progressive. Après avoir calculé les cartes de correspondances à  $\frac{1}{8}$ <sup>ème</sup> de résolution, on souhaiterait calculer les cartes à résolution  $\frac{1}{4}$ <sup>ème</sup>, mais il serait trop coûteux en mémoire de calculer ces dernières de manière dense. CasMTR explore deux stratégies :

- Fenêtre locale : Pour tous les points de la source à  $\frac{1}{4}$ <sup>ème</sup> de résolution, au lieu de calculer le produit scalaire avec toute la cible, CasMTR propose d'utiliser la carte de correspondance de la résolution précédente pour déterminer la région la plus prometteuse, et de calculer une carte de correspondance locale autour de cette région.
- K régions éparses : Pour tous les points de la source à  $\frac{1}{4}$ <sup>ème</sup> de résolution, on peut aussi se servir des cartes de correspondances pour trouver les K régions les plus prometteuses, et cette fois-ci calculer le produit scalaire uniquement avec ces K régions, réduisant ainsi les calculs à des cartes de correspondances éparses.

Calculer des cartes de correspondances locales ou éparses pour tous les pixels réduit drastiquement le coût mémoire. De plus, cela permet à CasMTR d'utiliser l'attention sur ces régions éparses afin de créer de la communication entre les images. Ces nouvelles cartes de correspondances sont ensuite utilisées de la même manière pour trouver toutes les correspondances à la résolution  $\frac{1}{2}$ .

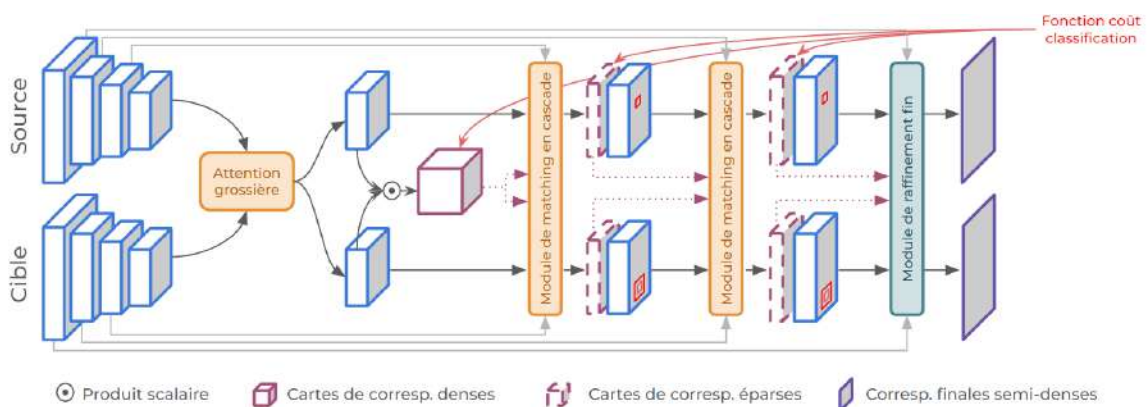


FIGURE 4.2 – **Représentation schématique de l'architecture de CasMTR.** CasMTR est construit comme une succession de tâches de classification sur les cartes de correspondances.

Le réseau peut alors être entraîné de la même manière que LoFTR, en utilisant une dérivée de l'entropie croisée [Ross and Dollár, 2017] sur les cartes de correspondances. CasMTR produit donc des correspondances denses à  $\frac{1}{2}$  de la résolution d'origine des images. Pour sélectionner les correspondances pertinentes pour l'estimation de pose de caméra, les auteurs

proposent d'utiliser une variante de l'algorithme de suppression non-maximale (NMS) sur les cartes de correspondances produites par le réseau.

Malgré ses bonnes performances en estimation de pose de caméra, CasMTR présente certaines limitations. La méthode semble peu efficace dans des scénarios d'intérieur comportant des régions peu texturées ou des variations de mouvement importantes. De plus, l'utilisation de la suppression non-maximale pour filtrer les correspondances les plus informatives dépend fortement du choix de la taille du noyau utilisé pour la détection des maxima locaux. Cependant, CasMTR montre qu'il est possible de formaliser la mise en correspondance *grossier à fin* comme une tâche de classification de manière similaire aux méthodes semi-denses.

### 4.2.3 Discussion

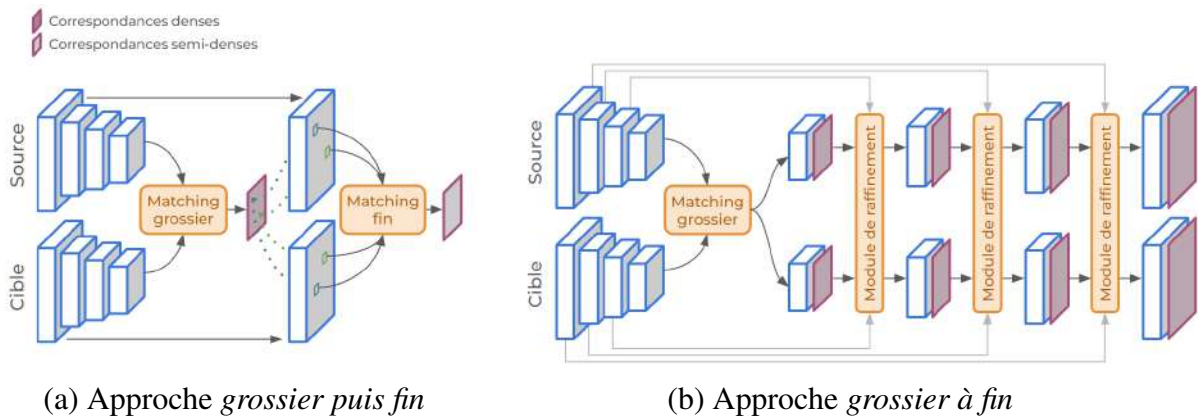


FIGURE 4.3 – **Représentation schématique des approches *grossier puis fin* des méthodes semi-denses (a), et *grossier à fin* des méthodes denses (b).** (a) Les méthodes semi-denses (comme LoFTR [Sun et al., 2021]) construisent des correspondances grossières denses qu'elles utiliseront pour déterminer des régions fines, pour lesquelles elles trouvent une correspondance fine par région. (b) Les méthodes denses (comme DKM [Edstedt et al., 2023]) construisent également des correspondances grossières denses, qui sont par la suite affinées et densifiées progressivement jusqu'à la pleine résolution.

A travers l'étude de ces deux articles, nous avons vu l'importance d'adopter une approche *grossier à fin* pour la mise en correspondance dense. DKM l'aborde comme une tâche de régression, alors que CasMTR la traite comme une tâche de classification. Contrairement à l'approche *grossier puis fin* des méthodes semi-denses, les correspondances sont construites de manière progressive au fur et à mesure de la montée en résolution (voir Figure 4.3). L'approche de CasMTR utilisant des cartes de correspondances et une fonction de coût de classification semble plus prometteuse. En effet, l'utilisation de l'entropie croisée pour la supervision semble plus riche, car elle encourage les correspondances correctes et pénalise les mauvaises, et a démontré sa robustesse dans le paradigme semi-dense. Ces deux méthodes soulignent également des problématiques inhérentes au matching dense, comme l'échantillonnage des correspondances finales et la gestion du coût mémoire de l'architecture.

Dans la section suivante, nous allons continuer l'analyse de la mise en correspondance dense *grossier à fin*, soulignant de nouvelles problématiques et proposant une nouvelle architecture basée sur l'attention et entraînée comme une succession de tâches de classification.

### 4.3 Matching dense par recherche en faisceaux

La mise en correspondance dense d'image (DIM) vise à établir des correspondances entre tous les pixels de deux images, assurant ainsi une couverture complète des images. L'objectif serait d'étendre l'approche des méthodes semi-denses (SDF), qui construisent des volumes de correspondances à résolution grossière. Cependant, le calcul des correspondances à résolution fine présente des défis majeurs en termes de mémoire. En effet, si l'on souhaite calculer le volume de correspondances directement à résolution fine, pour chaque descripteur de pixel de l'image source, il faudrait calculer le produit scalaire avec tous les descripteurs de l'image cible. Typiquement, pour des images de taille  $H = 480$  et  $W = 640$ , cela nécessiterait environ 377 Go de mémoire pour stocker ce volume, rendant cette approche impraticable. Pour établir des correspondances fines, les méthodes SDF utilisent une stratégie de classification *grossier puis fin* où chaque région grossière est utilisée pour trouver une correspondance fine, résultant à une mise en correspondance d'un sous-ensemble de tous les pixels des images. Dans le cadre du DIM, nous aimerions établir une stratégie *grossier à fin* où les correspondances grossières sont utilisées pour trouver des correspondances pour tous les pixels des images. Nous verrons dans la section 4.3.1 que l'utilisation d'une telle stratégie pour trouver des correspondances denses pose une nouvelle problématique de multimodalité des correspondances.

Une seconde problématique inhérente au paradigme de mise en correspondance dense est la sélection des correspondances. En effet, les méthodes denses produisent des correspondances pour tous les pixels des images, mais tous les pixels de l'image source n'ont pas réellement un correspondant dans l'image cible. Certains pixels peuvent être occultés dans l'image cible, ne pas apparaître sous le point de vue de la cible ou encore appartenir à des structures non rigides de la scène 3D observée (exemples : le ciel, les feuilles d'un arbre, ...). Dans tous ces cas, la méthode dense va tout de même produire une correspondance, mais nous ne souhaitons pas utiliser ces correspondances pour des tâches annexes comme l'estimation de pose de caméra. Il faut donc concevoir un mécanisme permettant de filtrer ces correspondances. Dans la section 4.3.2, nous verrons différentes techniques pour ne sélectionner que des correspondances fiables et comment nous intégrons la prédiction de la covisibilité au sein de notre architecture pour pallier ce problème.

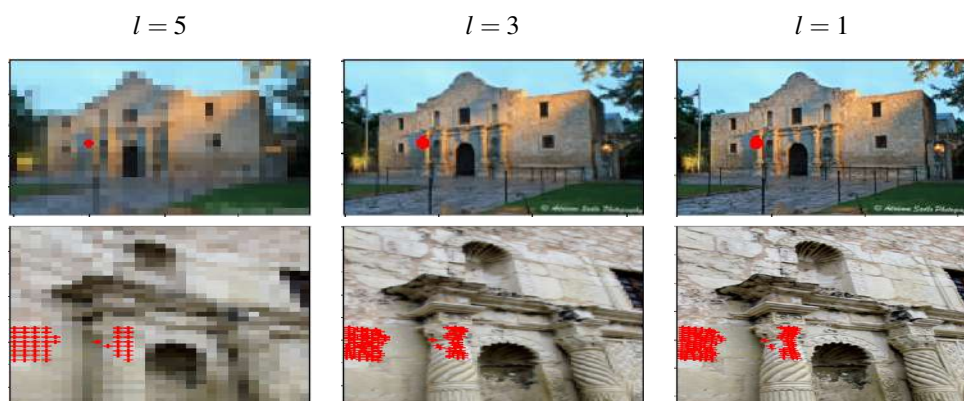


FIGURE 4.4 – **Illustration d'une paire d'images avec une forte multimodalité.** A résolution 1 (à droite) les 256 correspondances (+) sont bien distinctes dans les deux images. Lorsque l'on descend à résolution  $\frac{1}{16}$  (à gauche), les 256 points dans la source ne correspondent plus qu'à une région dans l'image du haut mais à 51 régions distinctes dans l'image du bas. De plus les 51 régions sont distantes et ne peuvent donc pas être toute capturées par une fenêtre locale. Dans cette paire, le problème est exacerbé par le zoom extrême.

### 4.3.1 Problème de multimodalité

Pour établir une stratégie de classification *grossier à fin*, supposons que nous disposons d'un réseau de neurones prenant en entrée une paire d'images : l'image source  $I_s$  ( $H_s \times W_s \times 3$ ) et l'image cible  $I_t$  ( $H_t \times W_t \times 3$ ), et produisant en sortie  $L$  ensembles multi-échelles de caractéristiques denses pour l'image source et l'image cible :  $\{F_s^l\}_{l=1\dots L}$  et  $\{F_t^l\}_{l=1\dots L}$ . Ici,  $F_s^l$  et  $F_t^l$  ont respectivement une taille de  $\frac{H_s}{2^{l-1}} \times \frac{W_s}{2^{l-1}} \times d_l$  et  $\frac{H_t}{2^{l-1}} \times \frac{W_t}{2^{l-1}} \times d_l$ , où  $d_l$  est la dimension des descripteurs à l'échelle  $l$ . À chaque échelle  $l = 1, 2, \dots, L$ , la grille de coordonnées 2D source  $\Omega_s^l$  et cible  $\Omega_t^l$  ont respectivement une taille de  $\frac{H_s}{2^{l-1}} \times \frac{W_s}{2^{l-1}}$  et  $\frac{H_t}{2^{l-1}} \times \frac{W_t}{2^{l-1}}$ .

Notre objectif est de réaliser une mise en correspondance dense d'images, c'est-à-dire d'établir des correspondances entre les positions 2D fines de la source  $\{\mathbf{p}_{s,i}^1\}_{i=1,\dots,|\Omega_s^1|}$  et les positions 2D fines de la cible  $\{\mathbf{p}_{t,i}^1\}_{i=1,\dots,|\Omega_t^1|}$ . Dans le reste de cette partie, une position 2D  $\mathbf{p}$  est considérée comme un vecteur 2D d'entiers. Par simplicité de notation,  $F_s^l(\mathbf{p}_{s,i}^l)$  fera référence au descripteur de  $F_s^l$  à la position  $\mathbf{p}_{s,i}^l$  dans la source à l'échelle  $l$ .

**Énoncé** - Supposons qu'à l'échelle la plus fine  $l = 1$ , le problème de mise en correspondance soit un-à-un, c'est à dire  $\mathbf{p}_{s,i}^1$  a un seul correspondant dans la grille cible  $\Omega_t^1$  (voir Figure 4.5 (a)). À l'échelle plus grossière  $l = 2$ , la mise en correspondance peut ne plus être un-à-un, mais un-à-quatre, c'est à dire  $\mathbf{p}_{s,i}^2 = \left\lceil \frac{\mathbf{p}_{s,i}^1}{2} \right\rceil$  peut avoir 4 correspondants dans la grille cible  $\Omega_t^2$ . (Voir Figure 4.5 (b)) Ainsi, en théorie, à l'échelle  $l$ , le problème de mise en correspondance devient un-à- $4^{l-1}$ , dans le pire des cas. Nous appelons cette relation de correspondances un-à- $m$  lors de la diminution de résolution la multimodalité.

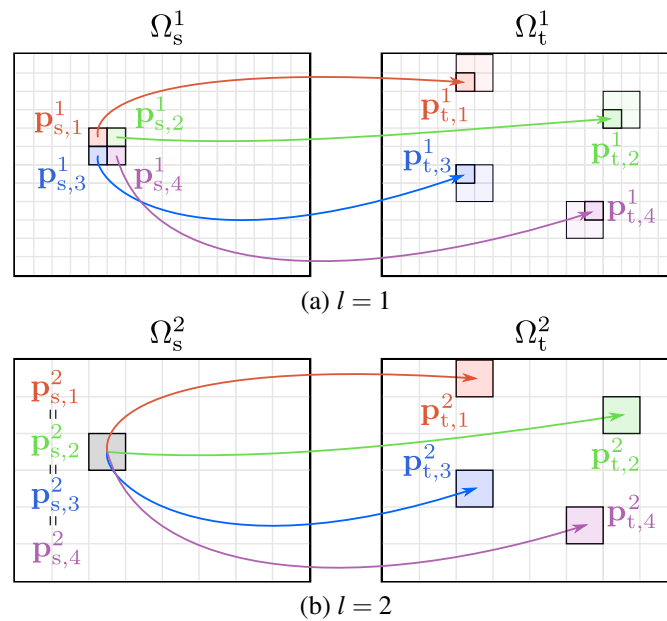


FIGURE 4.5 – **Illustration du problème de multimodalité** engendré par une approche *grossier à fin* de mise en correspondance dense. A une résolution fine (a), quatre correspondances voisines distinctes peuvent conduire à une correspondance un-à-quatre à une échelle plus grossière (b). (voir le texte pour les notations)



Dans le cadre du DIM, nous souhaitons commencer par une mise en correspondance grossière, puis affiner ces prédictions jusqu'à la résolution la plus fine. Une stratégie simple serait, en partant d'une correspondance grossière, de chercher le correspondant à la résolution suivante dans la région  $2 \times 2$  trouvée précédemment, et ainsi de suite jusqu'à la résolution la plus fine. Mais avec la multimodalité présente dans les résolutions les plus grossières, une telle stratégie n'est pas souhaitable, car à ces résolutions, les correspondances ne sont plus un-à-un mais un-à- $m$ . La Figure 4.4 montre un exemple de paire d'images où 256 correspondances fines résultent en une correspondance un-à-51 pour  $l = 5$  ( $\frac{1}{16}$ ème de la résolution d'origine).

À noter que les méthodes semi-denses ne souffrent théoriquement pas de la multimodalité, car elles n'établissent qu'une seule correspondance par région grossière. De ce fait, si elles trouvent la bonne région grossière correspondante et que leur fenêtre de raffinement est au moins de taille  $2^{L-1} \times 2^{L-1}$ , alors elles peuvent trouver la bonne correspondance fine.

#### 4.3.1.1 Validation expérimentale

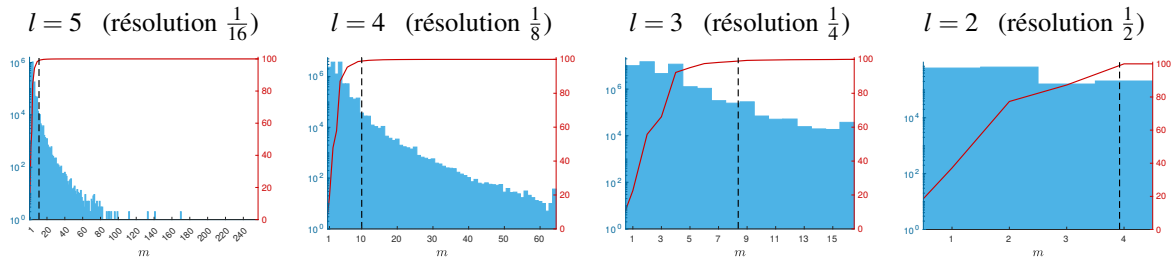


FIGURE 4.6 – **Analyse expérimentale d'hypothèses multiples sur MegaDepth.** Pour chaque résolution, l'axe de gauche est en échelle logarithmique et l'histogramme correspondant indique le nombre de fois qu'un problème de correspondance de un à  $m$  a été trouvé dans les correspondances de vérité terrain. La courbe rouge (axe de droite) est l'histogramme cumulatif. La ligne verticale en pointillés marque la valeur de  $m$  qui englobe 99% de la distribution. À la résolution la plus fine ( $l = 2$ ), la limite théorique de correspondances de un à 4 est très souvent atteinte. Au contraire, à la résolution la plus grossière ( $l = 5$ ), la limite théorique de correspondances de un à 256 n'est presque jamais atteinte.

Dans cette partie nous cherchons à valider expérimentalement l'énoncé théorique précédent. En utilisant 10 000 paires d'images d'entraînement de MegaDepth [Li and Snavely, 2018]. Pour chaque paire d'images, un ensemble de correspondances de vérité terrain (GT)  $\{(\mathbf{p}_{s,k}^{\text{GT},1}, \mathbf{p}_{t,k}^{\text{GT},1})\}$  est obtenu en utilisant les cartes de profondeur et les poses de caméras de GT disponibles. Pour plusieurs échelles  $l = 2, \dots, 5$ , nous sous-échantillons les correspondances GT  $\left\{(\mathbf{p}_{s,k}^{\text{GT},l} = \left\lfloor \frac{\mathbf{p}_{s,k}^{\text{GT},1}}{2^{l-1}} \right\rfloor, \mathbf{p}_{t,k}^{\text{GT},l} = \left\lfloor \frac{\mathbf{p}_{t,k}^{\text{GT},1}}{2^{l-1}} \right\rfloor)\right\}$ . Ensuite, pour chaque échelle, nous calculons le nombre de positions sources  $\mathbf{p}_{s,i}^l \in \Omega_s^l$  ayant  $m$  correspondances GT différentes, pour  $m = 1, 2, \dots, 4^{l-1}$ .

Dans la Figure 4.6, nous constatons qu'à l'échelle  $l = 2$ , une grande partie des correspondances GT respecte la limite théorique, tandis qu'à l'échelle  $l = 5$ , la limite théorique de un-à-256 n'est presque jamais atteinte. Nous pensons que ce résultat provient du fait que les paires d'images sont sélectionnées pour avoir un chevauchement minimal, tandis que les cas extrêmes de un-à-256 se produisent généralement dans des zooms extrêmes où le chevauchement est par conséquent très faible.



Dans la Figure 4.7, nous montrons des paires d'images présentant un problème de multimodalité lorsqu'on applique la stratégie *grossier à fin*. On peut voir que les problèmes de multiples hypothèses fortes proviennent soit de changements de point de vue importants, soit de cas où l'image cible est un zoom de l'image source. Dans des paires d'images naturelles, comme celles de nos différents jeux de données, il s'agit souvent d'une combinaison de ces deux cas.

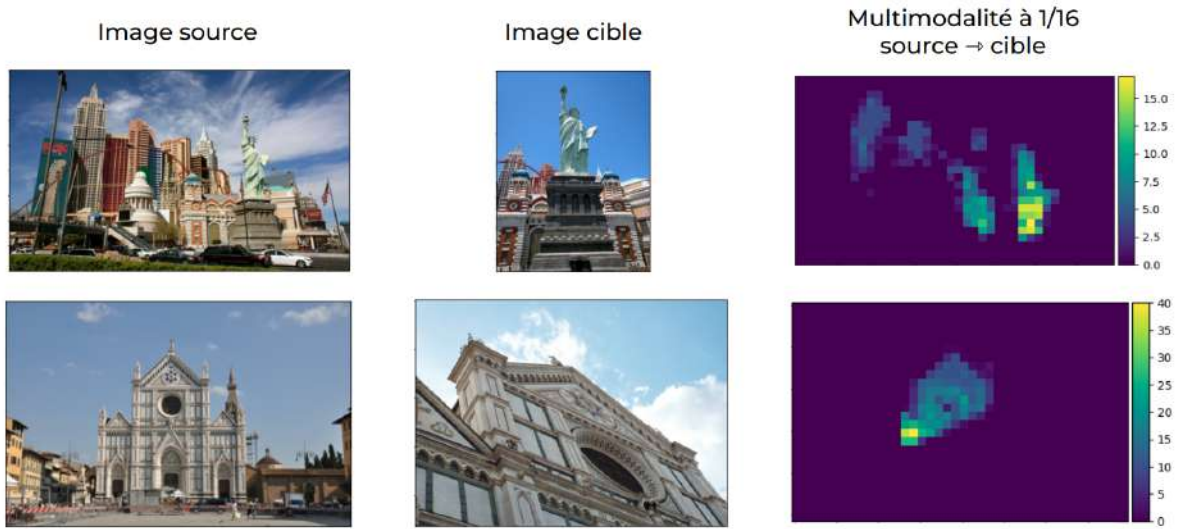


FIGURE 4.7 – Exemple de multimodalité pour des paires d'images de MegaDepth. À droite à visualise la valeur  $m$  des correspondances un-à- $m$  créée par la descente à la résolution  $\frac{1}{16}$  ème ( $l = 5$ ).

#### 4.3.1.2 Recherche en faisceaux

**Motivation.** Pour établir des correspondances denses de manière *grossier à fin*, nous souhaitons concevoir une architecture capable de conserver l'ambiguïté induite par la multimodalité présente dans la résolution la plus grossière, puis de lever ces ambiguïtés au fur et à mesure de la montée en résolution, pour finir avec des correspondances un-à-un à pleine résolution.

**Matching grossier à fin.** Supposons que nous disposions du même réseau de neurones décrit précédemment, prenant en entrée une paire d'images : l'image source  $I_s$  et l'image cible  $I_t$ , et produisant en sortie des ensembles multi-échelles de caractéristiques denses  $\{F_s^l\}_{l=1\dots L}$  et  $\{F_t^l\}_{l=1\dots L}$ .

Le DIM peut être formulé comme une série de tâches de classification progressive du grossier au fin : la mise en correspondance dense est effectuée à l'échelle la plus grossière  $l = L$ , puis progressivement affinée jusqu'à atteindre l'échelle fine  $l = 1$ .

Plus précisément, au lieu d'essayer directement de trouver la correspondance de  $\mathbf{p}_{s,i}^1 \in \Omega_s^1$ , nous commençons par chercher les correspondances grossières des points  $\mathbf{p}_{s,i}^L = \left\lceil \frac{\mathbf{p}_{s,i}^1}{2^{L-1}} \right\rceil \in \Omega_s^L$ , où  $\lceil \cdot \rceil$  est l'opérateur du plus proche entier. Une carte de correspondance grossière **dense** est calculée :

$$C_{\mathbf{p}_{s,i}^L}^L = \text{softmax}(F_s^L(\mathbf{p}_{s,i}^L) \odot F_t^L), \quad (4.1)$$

où  $C_{\mathbf{p}_{s,i}^L}^L$  est de taille  $\frac{H_t}{2^{L-1}} \times \frac{W_t}{2^{L-1}}$ .

Ensuite, nous considérons les positions  $\mathbf{p}_{s,i}^{L-1} = \left\lceil \frac{\mathbf{p}_{s,i}^1}{2^{L-2}} \right\rceil \in \Omega_s^{L-1}$  pour la résolution suivante. Ici,  $C_{\mathbf{p}_{s,i}^L}^L$  est utilisée pour définir un ensemble de positions de pixels  $\Omega_{t,i}^{L-1} \subset \Omega_t^{L-1}$  où une carte de correspondance **éparse** est évaluée :

$$\tilde{C}_{\mathbf{p}_{s,i}^{L-1}}^{L-1} = \text{softmax}(F_s^{L-1}(\mathbf{p}_{s,i}^{L-1}) \odot F_t^{L-1}(\Omega_{t,i}^{L-1})). \quad (4.2)$$

Le même processus est répété jusqu'à atteindre l'échelle  $l = 1$  :  $\mathbf{p}_{s,i}^l = \left\lceil \frac{\mathbf{p}_{s,i}^1}{2^{l-1}} \right\rceil \in \Omega_s^l$  est considérée ;  $\tilde{C}_{\mathbf{p}_{s,i}^{l+1}}^{l+1}$  est utilisée pour définir un ensemble de localisations de pixels  $\Omega_{t,i}^l \subset \Omega_t^l$  où une carte de correspondance **éparse** est évaluée :

$$\tilde{C}_{\mathbf{p}_{s,i}^l}^l = \text{softmax}(F_s^l(\mathbf{p}_{s,i}^l) \odot F_t^l(\Omega_{t,i}^l)), \quad (4.3)$$

La correspondance finale de  $\mathbf{p}_{s,i}^1$  est définie comme l'argmax de la dernière carte de correspondance éparse  $\tilde{C}_{\mathbf{p}_{s,i}^1}^1$ .

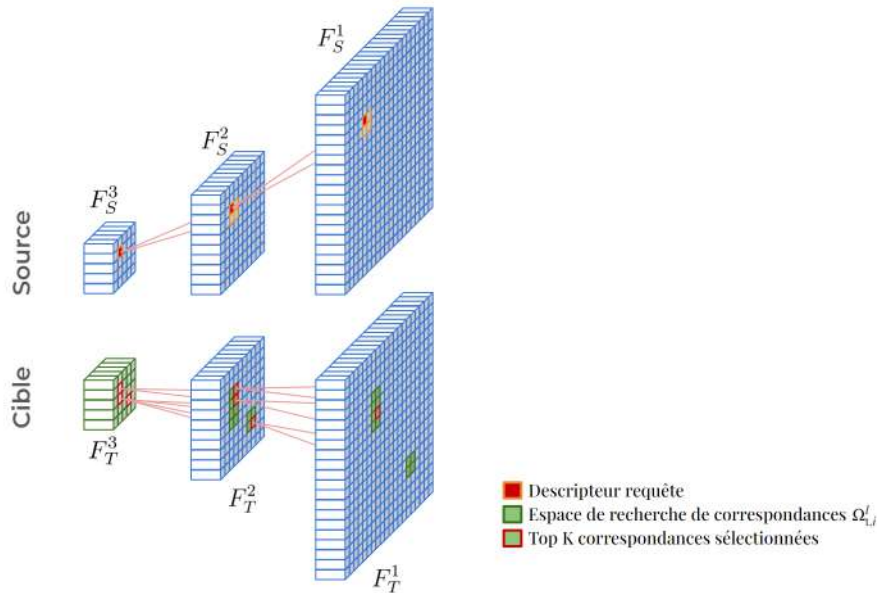


FIGURE 4.8 – **Schéma de la recherche en faisceaux.** À résolution grossière, l'espace de recherche (la carte de correspondance) est dense. Puis on sélectionne les régions les plus prometteuses pour définir l'espace de recherche de la résolution suivante. Cette fois-ci la carte de correspondance est éparse. On sélectionne à nouveau les régions les plus prometteuses pour définir l'espace de recherche à pleine résolution. Enfin, l'argmax de la carte de correspondance à pleine résolution est notre prédiction finale pour la correspondance de notre point requête.

**Recherche en faisceaux.** Nous avons dit précédemment que  $\mathcal{C}_{\mathbf{p}_{s,i}}^l$  était utilisée pour définir un ensemble de positions de pixels  $\Omega_{t,i}^{l-1} \subset \Omega_t^{l-1}$  qui seront utilisées pour créer nos cartes de correspondances **éparse**. Pour implémenter cette approche grossière-fine, nous devons décider comment  $\Omega_{t,i}^l$  dans l'Eq. 4.3 doit être calculée, en supposant que  $\tilde{\mathcal{C}}_{\mathbf{p}_{s,i}}^{l+1}$  est donnée.

Garder uniquement l'argmax de  $\tilde{\mathcal{C}}_{\mathbf{p}_{s,i}}^{l+1}$  pour définir  $\Omega_{t,i}^l$  ne prendrait pas en compte les correspondances un-à-m présentes à la résolution grossière. Notre analyse de la multimodalité (section 4.3.1.1) nous a montré qu'un sous-ensemble  $\Omega_{t,i}^{l-1}$  restreint de  $\Omega_t^{l-1}$  était suffisant pour prendre en compte toute la multimodalité.

Ces résultats précédents nous amènent à envisager une stratégie de recherche en faisceaux (voir schéma Figure 4.8), où les  $K_l$  positions de  $\tilde{\mathcal{C}}_{\mathbf{p}_{s,i}}^l$  (Eq. 4.3) les plus prometteuses sont utilisées pour définir l'ensemble des positions 2D  $\Omega_{t,i}^{l-1}$ . Plus précisément, chaque position  $\mathbf{p}$  parmi les  $K_l$  meilleures est transformée en quatre localisations :  $(2\mathbf{p}+[00]^\top, 2\mathbf{p}+[10]^\top, 2\mathbf{p}+[01]^\top, 2\mathbf{p}+[11]^\top)$ . Ainsi, en pratique, une carte de correspondance éparse à l'échelle  $l$  est évaluée sur  $4K_{l+1}$  positions.

En supposant que nous disposons de descripteurs "parfaits"  $\{\mathbf{F}_s^l\}_{l=1\dots L}$  et  $\{\mathbf{F}_t^l\}_{l=1\dots L}$ , pour pouvoir établir correctement toutes les correspondances, en théorie, nous devrions définir  $K_l = 4^{l-1}$ . Néanmoins, dans la Figure 4.6, nous pouvons voir qu'il est possible d'établir correctement 99% des correspondances avec des valeurs inférieures à  $4^{l-1}$ . Cependant, en pratique, nous ne disposons pas de descripteurs "parfaits". Il est donc nécessaire de choisir des valeurs plus élevées. Nous avons trouvé expérimentalement que  $K_2 = 8, K_3 = 16, K_4 = 24, K_5 = 32$  représentent un bon compromis entre les besoins en mémoire et la capacité à établir correctement les correspondances. Pour obtenir ces valeurs, nous avons commencé avec les valeurs issues de la Figure 4.6 et les avons progressivement augmentées jusqu'à ce que l'empreinte mémoire (pendant l'entraînement) de notre réseau atteigne la capacité de notre GPU (16 Go). La Figure 4.9 propose une visualisation de la recherche d'une correspondance pour une pair d'images de MegaDepth.

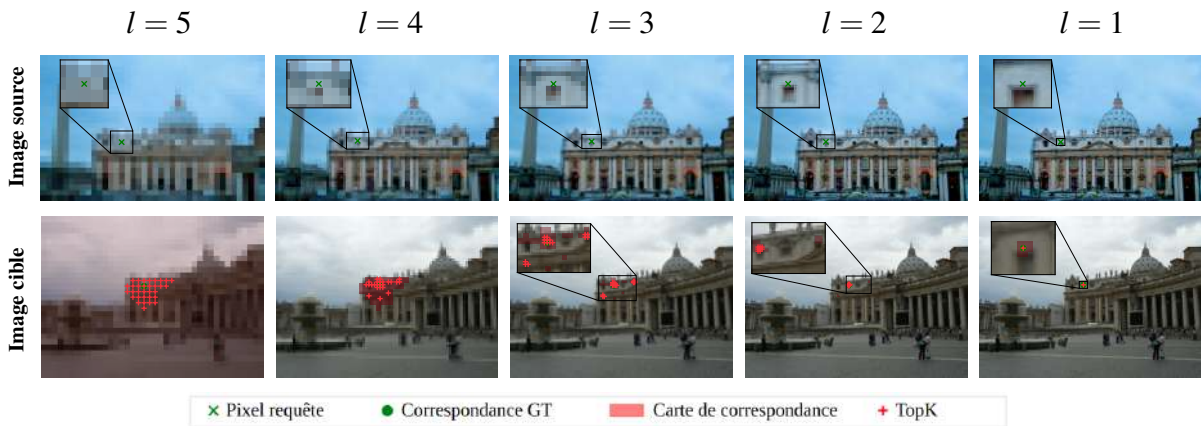


FIGURE 4.9 – **Visualisation de la recherche en faisceaux.** Les cartes de correspondances sont successivement à partir des régions les plus prometteuses de l'échelle précédente.

### 4.3.1.3 Beam-attention

Considérons un modèle de référence entièrement convolutionnel utilisant un FPN (*Feature Pyramid Network* [Lin et al., 2017]) pour extraire les ensembles multi-échelles de caractéristiques denses  $\{F_s^l\}_{l=1\dots L}$  et  $\{F_t^l\}_{l=1\dots L}$ . On peut utiliser notre recherche en faisceaux pour produire des correspondances denses à partir de cette architecture, mais les simples convolutions  $1 \times 1$  du FPN ne permettent pas une communication intra- et inter-images capable d'enlever les ambiguïtés liées aux structures répétitives ou aux larges changements de point de vue. On peut ajouter, comme dans les méthodes semi-denses, de l'attention dense à la résolution la plus grossière  $l = 5$  pour commencer à créer de la communication entre les images, mais nous aimerions également utiliser l'attention pour spécifiquement lever l'ambiguïté créée par la multimodalité.

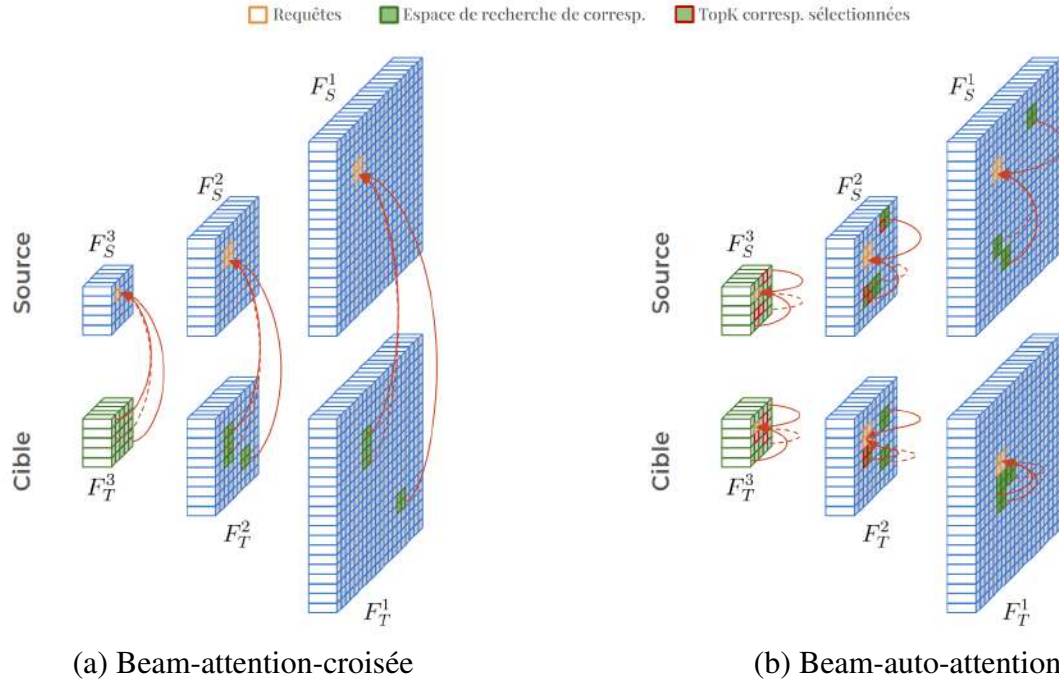


FIGURE 4.10 – Visualisation de la beam-attention-croisée (a) et la beam-auto-attention (b). La recherche en faisceaux est utilisée pour effectuer de l'attention éparsée. Ceci réduit drastiquement le coût calculatoire par rapport à l'attention dense et nous permet de créer de la communication intra- et inter-images même à pleine résolution.

Ajouter des couches d'attention dense à des échelles plus fines est computationnellement impraticable, car dans ce cas, les scores d'attention de chaque tête seraient stockés dans une matrice de taille  $Q \times N = |\Omega_s^l| \times |\Omega_t^l|$ . À la place, nous pouvons utiliser le raffinement par recherche en faisceaux pour créer des couches de attention-croisée standard. Notons que l'Eq. 4.3 est très similaire à l'équation de l'attention softmax [Vaswani et al., 2017] (4.4), les éléments d'une carte de correspondance éparsée peuvent être interprétés comme des scores d'attention dans une couche d'attention-croisée. Ainsi, à l'échelle  $l$  (avec  $l < L$ ), pour chaque  $\mathbf{p}_{s,i}^l \in \Omega_s^l$ , nous proposons d'effectuer une opération de attention-croisée entre le vecteur de caractéristiques source  $F_s^l(\mathbf{p}_{s,i}^l)$  et les vecteurs de caractéristiques cibles aux localisations  $F_t^l(\Omega_{t,i}^l)$  :

$$Attention(Q, K, V) = softmax \left( \frac{QK^T}{\sqrt{d_q}} \right) V,$$

(4.4)

$$\begin{aligned} \text{Avec, } Q &= F_s^l(\mathbf{p}_{s,i}^l)W_q, \\ K &= F_t^l(\Omega_{t,i}^l)W_k, \\ V &= F_t^l(\Omega_{t,i}^l)W_v, \end{aligned}$$

Lorsque  $\Omega_{t,i}^l$  est défini en utilisant notre recherche en faisceaux proposée, cette opération de cross-attention permet à chaque vecteur de caractéristiques source  $F_s^l(\mathbf{p}_{s,i}^l)$  de communiquer avec les  $4K_{l+1}$  vecteurs de caractéristiques cibles, provenant des  $K_{l+1}$  meilleures localisations de la carte de correspondance plus grossière  $\tilde{C}_{\lfloor \mathbf{p}_{s,i}^l/2 \rfloor}^{l+1}$  (schématisé dans la Figure 4.10 (a)). Dans notre architecture, à chaque échelle  $l < L$ , nous proposons d'utiliser de telles couches d'attention-croisée basées sur la recherche en faisceaux de manière bidirectionnelle (voir schéma de l'architecture complète Figure 4.12 dans la section 4.3.3 abordant l'architecture BEAMER).

Nous avons vu dans les chapitres précédents qu'il était également important de permettre une communication intra-image pour une mise en correspondance précise. En parallèle de la recherche en faisceaux pour trouver les correspondances, nous pouvons appliquer, sur le même principe, des recherches en faisceaux des images entre elles. Au lieu d'effectuer la recherche entre  $\{F_s^l\}_{l=1\dots L}$  et  $\{F_t^l\}_{l=1\dots L}$ , nous effectuons une recherche en faisceaux entre  $\{F_s^l\}_{l=1\dots L}$  et  $\{F_s^l\}_{l=1\dots L}$  pour la source, et entre  $\{F_t^l\}_{l=1\dots L}$  et  $\{F_t^l\}_{l=1\dots L}$  pour la cible (dans la Figure 4.10 (b)). De cette manière, nous pouvons créer des espaces de correspondance éparse pour nos auto-attentions sur la source et la cible. À noter que ces recherches en faisceaux ne sont pas utilisées pour la recherche de correspondances, mais uniquement pour créer de la communication intra-image.

Nous verrons dans la section 4.3.3 comment la beam-attention-croisée et la beam-auto-attention sont utilisées au sein de l'architecture BEAMER.

## 4.3.2 Prédiction de la covisibilité

### 4.3.2.1 Problème de l'échantillonnage de correspondances

La sélection de correspondances dans le matching dense pose un problème fondamental en raison de la nature même de ce paradigme, qui vise à établir des correspondances pour chaque pixel de l'image source avec un pixel dans l'image cible. Contrairement aux méthodes éparse ou semi-denses, où seules des correspondances sont établies pour un ensemble limité de points clés ou de régions d'intérêt, le matching dense doit gérer un grand volume de correspondances potentielles. Il serait trop long en pratique de fournir toutes les correspondances à l'algorithme d'estimation de pose de caméra, surtout que certaines de ces correspondances prédites sont établies dans des régions qui ne sont pas covisibles entre les images. Pour échantillonner les correspondances qui serviront à l'estimation de pose de caméra parmi toutes les correspondances produites par une méthode dense, on peut distinguer plusieurs stratégies :

**La stratégie bidirectionnelle** repose sur le principe de consistance cyclique pour garantir la robustesse des correspondances. Cette méthode ne conserve une correspondance entre un



point  $\mathbf{p}_{S,i}$  de l'image source  $I_S$  et un point  $\mathbf{p}_{T,i}$  de l'image cible  $I_T$ , que si, lors du matching inverse (de  $I_T$  vers  $I_S$ ), le point  $\mathbf{p}_{S,i}$  est également retrouvé comme correspondant à  $\mathbf{p}_{T,i}$ . Cela signifie que chaque correspondance est vérifiée dans les deux sens (source vers cible et cible vers source), afin d'éliminer les correspondances ambiguës ou erronées. Cette approche est utilisée dans les méthodes semi-denses, qui s'appuient sur un softmax bidimensionnel appliqué aux cartes de correspondances pour garder les correspondances qui se valident dans les deux directions. Ce mécanisme permet de filtrer les correspondances erronées mais n'assure pas une meilleure consistance globale entre les correspondances.

La seconde stratégie repose sur l'**évaluation de la confiance** du modèle dans chaque correspondance potentielle. Dans les méthodes semi-denses, cette confiance est calculée via un softmax appliqué sur les cartes de correspondances, fournissant une distribution de probabilité pour chaque correspondance. Le modèle peut ainsi conserver les correspondances pour lesquelles il a une haute certitude et éliminer celles jugées incertaines. Cependant, l'interprétation du softmax sur les cartes de correspondance peut être remise en question [Szegedy et al., 2014]. Cette méthode fonctionne bien dans le cadre du semi-dense où les correspondances sont plus structurées, mais elle présente des limites dans le paradigme dense. En effet, pour notre architecture dense, les cartes de correspondances  $C_{\mathbf{p}_{S,i}}^l$  sont éparpillées aux niveaux  $l \neq L$ , ce qui empêche l'obtention d'une distribution de probabilité couvrant tous les pixels. La probabilité générée par le softmax ne reflète alors qu'une certitude locale, ce qui rend son utilisation difficile pour distinguer les pixels entre eux. Des méthodes comme PDCNet [Truong et al., 2021a] adoptent une approche probabiliste plus avancée en régressant non seulement une position moyenne pour chaque correspondance, mais aussi une variance, permettant de quantifier l'incertitude associée à chaque correspondance.

La troisième stratégie repose sur l'**estimation de la covisibilité** entre les pixels des deux images. Cette approche vise à éviter l'échantillonnage de correspondances dans des régions où il est impossible ou non pertinent d'établir une correspondance, telles que les zones occultées dans l'image cible, les zones se reprojétant en dehors de l'image cible ou les objets non rigides de la scène (ciel, foule, etc.). Par exemple, DKM [Edstedt et al., 2023] apprend à détecter les points *consistants* dans les deux images afin d'éviter de sélectionner des correspondances dans des régions non covisibles. Un point de l'image source est dit *consistant* si, lorsqu'il est reprojété dans l'espace de l'image cible à l'aide de la vérité terrain, sa valeur de profondeur correspond à celle de la carte de profondeur de l'image cible (avec un certain seuil de tolérance). Les points *consistants* sont alors considérés comme covisibles, et tous les autres comme non covisibles. Bien que cette méthode permette de réduire les erreurs dues à des correspondances irréalistes ou impossibles, elle ne prend pas en compte les imperfections de la vérité terrain dans les jeux de données de matching. En effet, certaines cartes de profondeur servant à déterminer la covisibilité des points sont incomplètes, en particulier dans les zones uniformes ou avec des variations importantes de profondeur (voir Figure 4.11). DKM traite les points sans profondeur comme non covisibles, alors qu'ils peuvent être covisibles, ce qui peut introduire un signal d'apprentissage incorrect lors de l'estimation des zones visibles.

Ces trois stratégies représentent les principales méthodes utilisées, mais on peut noter que d'autres approches existent, comme l'élagage hiérarchique [Chen et al., 2024], l'utilisation de contraintes épipolaires [Zhou et al., 2023] ou la suppression de correspondances locales non maximales [Bailo et al., 2018]. L'objectif de la phase d'échantillonnage des correspondances dans le matching dense est à la fois de filtrer les correspondances établies dans des régions incohérentes, de sélectionner les correspondances les plus précises possible et de choisir celles qui permettraient une bonne estimation de pose de caméra. Dans la suite, nous verrons comment nous intégrons l'apprentissage de la prédiction de la covisibilité dans notre architecture, sur



laquelle sera basée notre stratégie d'échantillonnage.

#### 4.3.2.2 Prédire la covisibilité entre deux images

Nous aimerions que notre architecture soit capable de prédire la covisibilité entre les images en parallèle de l'estimation des correspondances. Inspirée par DKM, qui prédit les régions *consistantes* dans les deux images, nous cherchons, de notre côté, à obtenir la *covisibilité complète* entre les images. L'objectif est d'enrichir l'information de covisibilité afin de fournir un signal d'apprentissage supplémentaire sur la géométrie 3D de la scène. Nous souhaitons également prendre en compte les absences de vérité terrain dans les cartes de profondeurs pour éviter d'introduire des biais dans l'apprentissage et de faussement guider l'estimateur. La prédiction de la *covisibilité complète* pourrait non seulement aider à filtrer les correspondances erronées, mais également améliorer la cohérence et la précision des correspondances estimées en fournissant un signal d'apprentissage plus riche.

Pour construire la vérité terrain pour la prédiction complète de la covisibilité, nous allons au-delà de la simple distinction entre les points visibles et non visibles en définissant plusieurs catégories de points (représentées dans la Figure 4.11). Tout d'abord, un point est considéré comme **visible** s'il apparaît dans les deux images. Ensuite, un point est classé comme **occulté** s'il est visible dans une image mais caché dans l'autre en raison d'un obstacle ou d'un changement de point de vue. Un point est défini comme **extérieur** lorsque sa projection dans une image sort des limites de l'autre image. Nous incluons également une catégorie **sans vérité terrain (noGT)** pour les points pour lesquels nous ne disposons pas d'informations de vérité terrain fiables. Enfin, nous définissons une catégorie pour le **ciel**, qui est un élément non rigide dans la scène. Les points du ciel ne possèdent pas de profondeur dans la vérité terrain ; la création d'une catégorie spécifique pour ces points permet d'éviter les confusions avec les autres catégories de covisibilité lors de l'inférence du modèle. Ce découpage en plusieurs classes de covisibilité a été introduit dans [Germain et al., 2022] pour l'*hallucination* de correspondances.

Ces différentes catégories sont construites à partir des cartes de profondeur fournies dans les jeux de données de mise en correspondance. Cependant, il est important de noter que ces cartes de profondeur sont souvent incomplètes et parfois imprécises, ce qui peut affecter la qualité de la vérité terrain et, par conséquent, la précision de la méthode. Pour construire la catégorie **ciel**, nous utilisons un détecteur de ciel [Kirillov et al., 2023] sur toutes les images de notre base de données d'entraînement MegaDepth. Comme on peut le voir dans la Figure 4.11, distinguer les points non visibles en points occultés, extérieurs ou appartenant au ciel ajoute une information non négligeable sur la relation géométrique entre les deux images.

**Prédiction de la covisibilité.** Pour intégrer l'apprentissage de la covisibilité dans notre architecture, nous utilisons une approche similaire à celle employée pour la mise en correspondance. Nous introduisons quatre vecteurs appris  $\{\mathbf{v}_v, \mathbf{v}_o, \mathbf{v}_e, \mathbf{v}_s\}$  de taille  $d$ , chacun représentant respectivement une des catégories définies précédemment : visible, occulté, extérieur, et ciel. Pour chaque point  $\mathbf{p}_{s,i}^l$  de l'image source à une résolution  $l$  donnée, notre objectif est que son descripteur associé  $F_s^l(\mathbf{p}_{s,i}^l)$  soit le plus similaire possible au descripteur de sa catégorie de covisibilité, tout en étant dissemblable aux descripteurs des autres catégories. Ce principe s'applique également à l'image cible. Sur cette base, nous construisons des cartes de covisibilité :

$$\text{COV}_{\mathbf{p}_{s,i}^l}^l = \text{softmax} \left( F_s^l(\mathbf{p}_{s,i}^l) \odot [\mathbf{v}_v \quad \mathbf{v}_o \quad \mathbf{v}_e \quad \mathbf{v}_s]^\top \right), \quad (4.5)$$

où chaque point est associé à une probabilité d'appartenir à une catégorie spécifique de covisibilité. Ces cartes sont construites de manière hiérarchique, suivant le même principe que

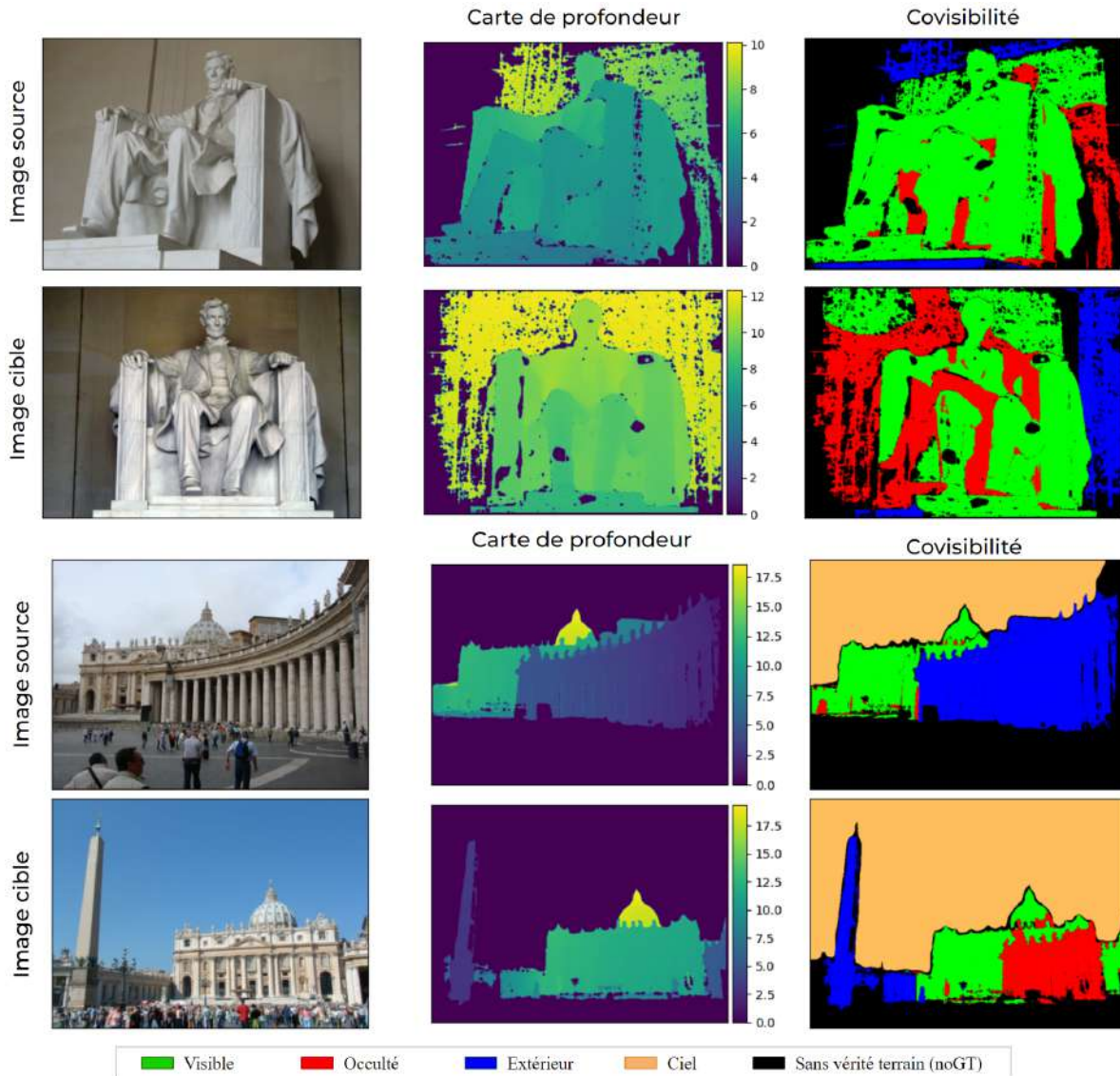


FIGURE 4.11 – **Visualisation de nos catégories de covisibilité pour une paire de MegaDepth.** Dans les cartes de profondeur, la valeur 0 signifie qu'il n'y a pas d'information de profondeur pour ce pixel. Dans la première paire d'images (Haut) on voit qu'il manque beaucoup d'information de profondeur, notamment dans les régions uniformes du mur. Nous décidons de ne pas superviser ces pixels, contrairement à DKM qui les considère comme des points *non consistants*. Notre découpage en plusieurs classes de covisibilité est également plus informatif sur la géométrie 3D de la scène liant les deux images, en comparaison à DKM qui regroupe les points **occultés**, **extérieurs**, **ciels** et **sans vérité terrain** comme une seule classe *non consistants*.

pour la mise en correspondance. À la résolution la plus fine  $l = 1$ , la catégorie finale prédite pour chaque point est déterminée en prenant l'*argmax* sur les scores de la carte de covisibilité. Les cartes de covisibilité sont construites de manière dense dans tout le réseau, même dans les couches les plus fines, car le produit scalaire avec les quatre descripteurs n'est pas très coûteux en mémoire par rapport au calcul des cartes de correspondances éparses.

**Communication dans le réseau.** Les descripteurs de covisibilité sont des vecteurs appris qui seront optimisés pendant l'entraînement. Cependant, pour rendre le produit scalaire entre ces vecteurs et les cartes de descripteurs des images encore plus discriminant, nous souhaitons permettre au réseau de neurones de créer de la communication entre tous ces descripteurs. Nous allons donc utiliser une opération d'attention-croisée classique [Vaswani et al., 2017] (Eq 4.4) de  $H$  heads avec les cartes de descripteurs des images comme requête ( $Q$ ) et les vecteurs de covisibilité comme clés ( $K$ ) et valeurs ( $V$ ). L'attention sur les deux images source et cible est faite séparément et en parallèle. De cette manière, le réseau peut injecter l'information des descripteurs de covisibilité appris à toutes les positions des images.

Grâce à cette approche, le réseau peut apprendre à prédire de manière dense la catégorie de covisibilité la plus probable pour chaque point d'image. Ceci va nous permettre de donner un signal d'apprentissage plus riche sur la géométrie de la scène et de prédire des régions covisibles précises pour mieux sélectionner les correspondances.

### 4.3.3 Notre architecture BEAMER

Nous combinons la recherche en faisceaux (*BEAM search*) de correspondances denses et ses opérations de beam-auto-attention et beam-attention-croisée, avec l'estimation de la covisibilité pour créer l'architecture BEAMER (BEAM matchER) schématisée dans la Figure 4.12.

Nous considérons un réseau siamois ResNet-18 [He et al., 2015] avec un FPN (*Feature Pyramid Network*) [Lin et al., 2017] pour extraire les pyramides de  $L = 5$  cartes de descripteurs nécessaires à notre recherche en faisceaux (voir section 4.3.1.2). Au niveau le plus grossier, 8 modules d'attention dense (Figure 4.13 (a)) sont utilisés pour faire communiquer la source et la cible et créer des cartes de correspondances denses ainsi que des cartes de covisibilité. Chacun de ces modules se compose de 6 couches Transformer (voir section 2.3.3 pour les détails d'une couche Transformer) : 2 couches d'auto-attention sûr les descripteurs de chaque image pour la communication intra-image, 2 couches d'attention-croisée entre les descripteurs de la source et de la cible pour la communication inter-images, ainsi que 2 couches d'attention-croisée sur les descripteurs de covisibilité et les descripteurs des images pour intégrer l'information de covisibilité au sein des images. Pour les résolutions suivantes, nous utilisons des modules de beam-attention (Figure 4.13 (b)) pour générer les cartes de correspondances éparses utiles à notre recherche en faisceaux, ainsi que des cartes de covisibilité denses. Ces modules de beam-attention ont une architecture similaire au module d'attention dense, mais utilisent de la beam-attention. Pour les résolutions  $\frac{1}{8}$ ,  $\frac{1}{4}$ ,  $\frac{1}{2}$  et 1, nous utilisons respectivement 2, 2, 1 et 1 modules de beam-attention consécutifs. À noter également que nous n'utilisons pas la même taille de descripteur pour toutes les résolutions afin de réduire le coût mémoire. Au niveau grossier, les descripteurs sont de taille  $d = 256$ , puis  $d = 128$  pour  $l = 4$ ,  $l = 3$ , et enfin  $d = 64$  pour  $l = 2$  et  $l = 1$ . Grâce à la recherche en faisceaux, BEAMER est la première méthode de l'état de l'art à pouvoir utiliser de l'attention de manière *grossier à fin* à toute les résolutions.

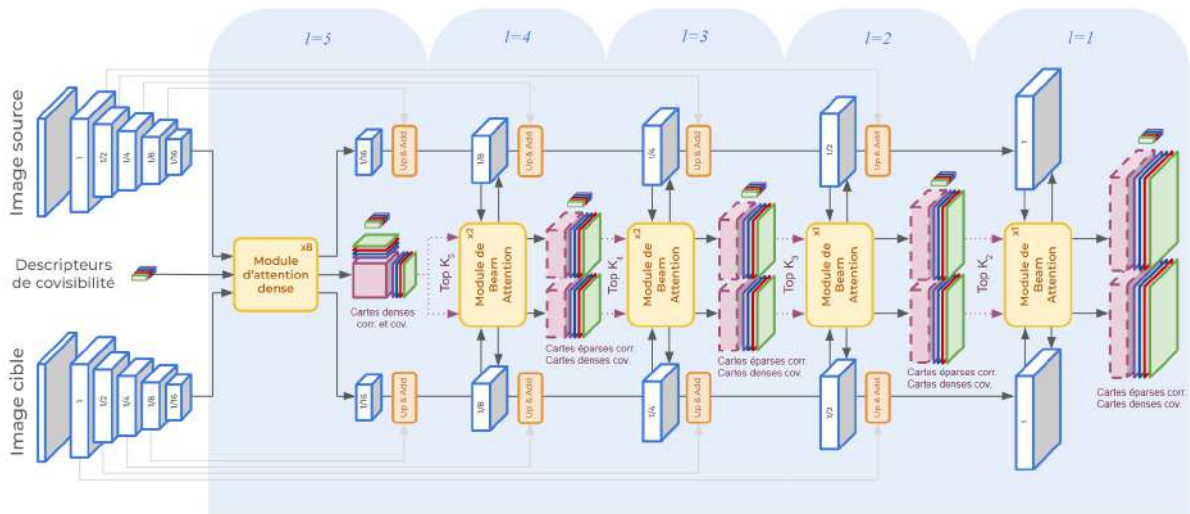
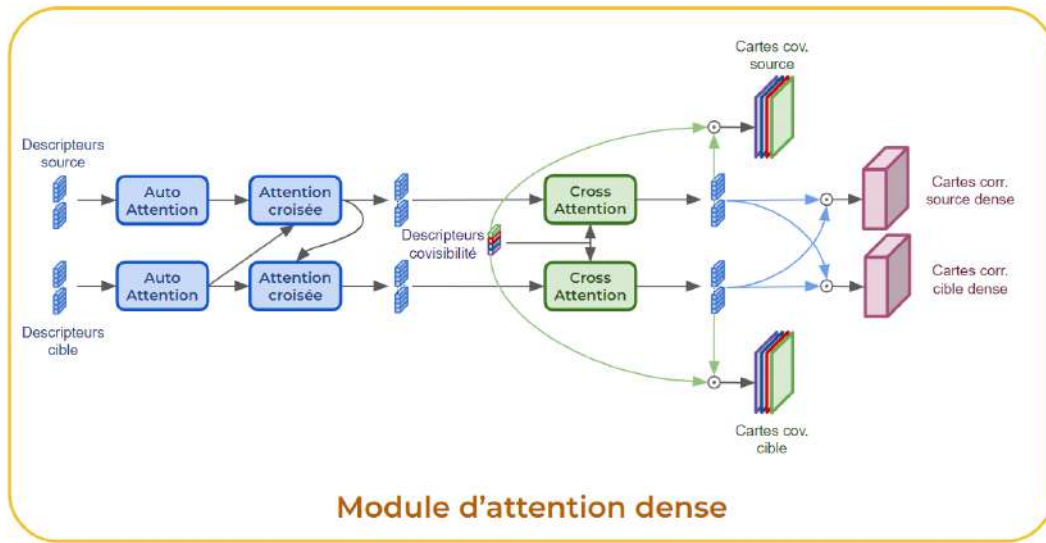


FIGURE 4.12 – Notre nouvelle architecture **BEAMER** établit des correspondances denses de manière bidirectionnelle, de manière *grossier à fin*. Au plus grossier, des cartes de correspondance denses sont calculées qui permettent d’initialiser la recherche de faisceaux. Puis à chaque échelle, le module de beam-attention utilise les correspondants les plus prometteurs des cartes de correspondance précédentes pour mettre en œuvre la recherche en faisceaux. Les beam-attentions de ce module permettent aux représentations source et cible de communiquer entre elles et de produire des cartes de correspondance éparse. En parallèle, les modules d’attention produisent une estimation dense de la covisibilité entre les images. Voir le texte pour plus de détails.

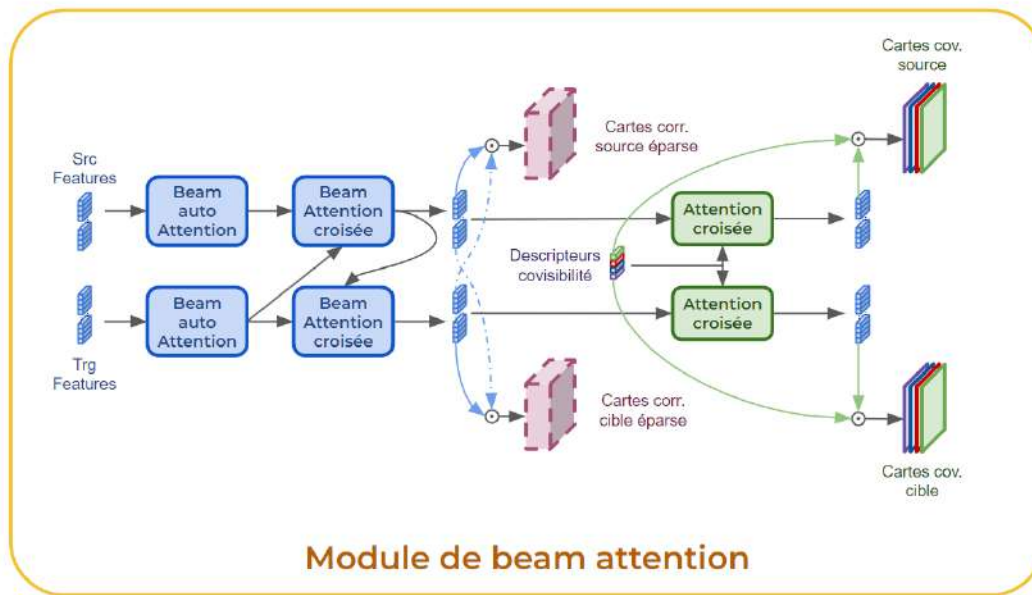
Contrairement à la convolution, l’opération d’attention est invariante par permutation et ne prend pas en compte la relation locale entre les pixels, et notre beam-attention conserve ces propriétés. Cependant, la disposition spatiale des pixels est importante et doit être prise en compte. Pour ce faire, nous apprenons un encodeur de position composé de trois couches de convolutions  $1 \times 1$ , qui prend les coordonnées  $(x, y)$  des pixels normalisées entre  $[-1, 1]$  et retourne un encodage positionnel  $PE$  de taille  $H \times W \times d$ . À chaque couche, nous sélectionnons une grille  $PE^l \subset PE$  qui correspond à la résolution de la couche et l’ajoutons aux caractéristiques. Notons que si une couche  $l$  travaille avec des caractéristiques de dimension  $d'$ , nous utilisons une convolution  $1 \times 1$  pour projeter les  $d$  dimensions de  $PE$  vers les  $d'$  dimensions de  $PE^l$ .

Tout au long de l’architecture, la mise en correspondance se fait de manière bidirectionnelle, source  $\rightarrow$  cible et cible  $\rightarrow$  source. De cette manière, nous pouvons vérifier la consistance cyclique des correspondances en une seule passe dans le réseau, mais également car nous avons remarqué qu’une des directions de matching était souvent préférable à l’autre. En effet, si l’on prend l’exemple d’un zoom, la direction de matching correspondant au zoom avant comportera une forte multimodalité, alors que le dézoom ne le sera pas (voir Figure 4.4). Nous étudions plus en détails ce matching bidirectionnel dans la section 4.4.1.





(a) Module d'attention dense



(b) Module de beam-attention

FIGURE 4.13 – **Visualisation de l'architecture des modules dense et éparse.** Ici chaque bloc d'attention correspond à une couche transformer entière (Attention → normalisation → perceptron multi-couches → normalisation).

#### 4.3.4 Étape d'entraînement

Pendant l'entraînement, nous disposons de paires d'images et de correspondances de vérité terrain (GT). Pour chaque paire d'images source/cible, un ensemble de correspondances GT  $(\mathbf{p}_{s,k}^{GT,1}, \mathbf{p}_{t,k}^{GT,1})_{k=1\dots N}$  est disponible, ainsi qu'un ensemble de points possédant une étiquette de covisibilité GT  $\{v_{s,m}^{GT,1}\}_{m=1\dots M}$ .

Pour la partie correspondance, notre objectif est de maximiser la vraisemblance de chaque correspondance à chaque échelle  $l = 1 \dots L$ . Dans notre cadre de classification, cela équivaut à minimiser la somme des termes suivants de log-vraisemblance négative (également appelée entropie croisée). Pour la partie classification de covisibilité, nous maximisons également la vraisemblance dans les cartes de covisibilité, ce qui au final nous donne :

$$\sum_{k=1}^N \mathcal{L}_{corr}(\mathbf{p}_{s,k}^{GT,1}, \mathbf{p}_{t,k}^{GT,1}) + \sum_{m=1}^M \mathcal{L}_{cov}(\mathbf{p}_{s,m}^{GT,1}), \quad (4.6)$$

où,

$$\mathcal{L}_{corr}(\mathbf{p}_{s,k}^1, \mathbf{p}_{t,k}^1) = -\ln\left(\mathcal{C}_{\mathbf{p}_{s,k}}^L(\mathbf{p}_{t,k}^L)\right) - \sum_{l=1}^{L-1} \ln\left(\tilde{\mathcal{C}}_{\mathbf{p}_{s,k}}^l(\mathbf{p}_{t,k}^l)\right), \quad (4.7)$$

$$\mathcal{L}_{cov}(\mathbf{p}_{s,m}^1) = -\sum_{l=1}^L \ln\left(\text{COV}_{\mathbf{p}_{s,k}}^l(v_{s,m}^{GT,l})\right), \quad (4.8)$$

avec  $\mathbf{p}_{s,k}^l = \left\lceil \frac{\mathbf{p}_{s,k}^1}{2^{l-1}} \right\rceil$  et  $\mathbf{p}_{t,k}^l = \left\lceil \frac{\mathbf{p}_{t,k}^1}{2^{l-1}} \right\rceil$ . Par convention, lorsque une correspondance de vérité terrain  $\mathbf{p}_{t,k}^l$  n'appartient pas à la carte de correspondances éparses issue de  $\Omega_{t,k}^l$ , alors  $\ln\left(\mathcal{C}_{\mathbf{p}_{s,k}}^l(\mathbf{p}_{t,k}^l)\right) := -\infty$  et par conséquent nous considérons que son score vaut 0 et que le gradient est nul. La supervision des échelles fines dépend donc de la précision des échelles précédentes, et le modèle apprend progressivement à produire des correspondances de plus en plus fines. Pendant l'entraînement nous monitorons le taux d'appartenance aux cartes de correspondances, disponible dans la Figure 4.14.

Il est important de noter que lorsque les correspondances de vérité terrain (GT) sont suffisamment denses (ce qui est le cas avec MegaDepth, car les correspondances GT sont obtenues à partir des cartes de profondeur), minimiser l'équation 4.6 conduit automatiquement à apprendre à produire des cartes de correspondance qui sont très multi-modales à des échelles grossières, c'est-à-dire qu'une carte de correspondance présente plusieurs pics à plusieurs localisations. Elles deviennent moins multi-modales aux échelles intermédiaires, et finalement uni-modales à la résolution  $l = 1$ . L'évolution des valeurs de ces différentes fonctions de coût lors de l'apprentissage est présentée dans la Figure 4.14.

Les entraînements sont réalisés de manière distribuée sur plusieurs GPU NVIDIA V100 16 Go, avec une paire d'images par GPU. Malgré l'aspect éparses de notre attention et les diverses optimisations que nous avons réalisées, l'architecture est très gourmande en mémoire lors de l'entraînement. Le calcul de l'inférence du réseau et de la rétropropagation du gradient (calcul des cartes de gradient) dépasse significativement les 16 Go de nos cartes. Pour réussir à entraîner notre architecture, nous réduisons (pendant l'entraînement uniquement) l'encodage des poids du réseau en demi-précision (16 bits). Nous utilisons également du *checkpointing*, qui nous permet de ne pas garder en mémoire toutes les matrices calculées lors de l'inférence du réseau et de les recalculer au moment de la rétropropagation, ralentissant significativement l'entraînement.



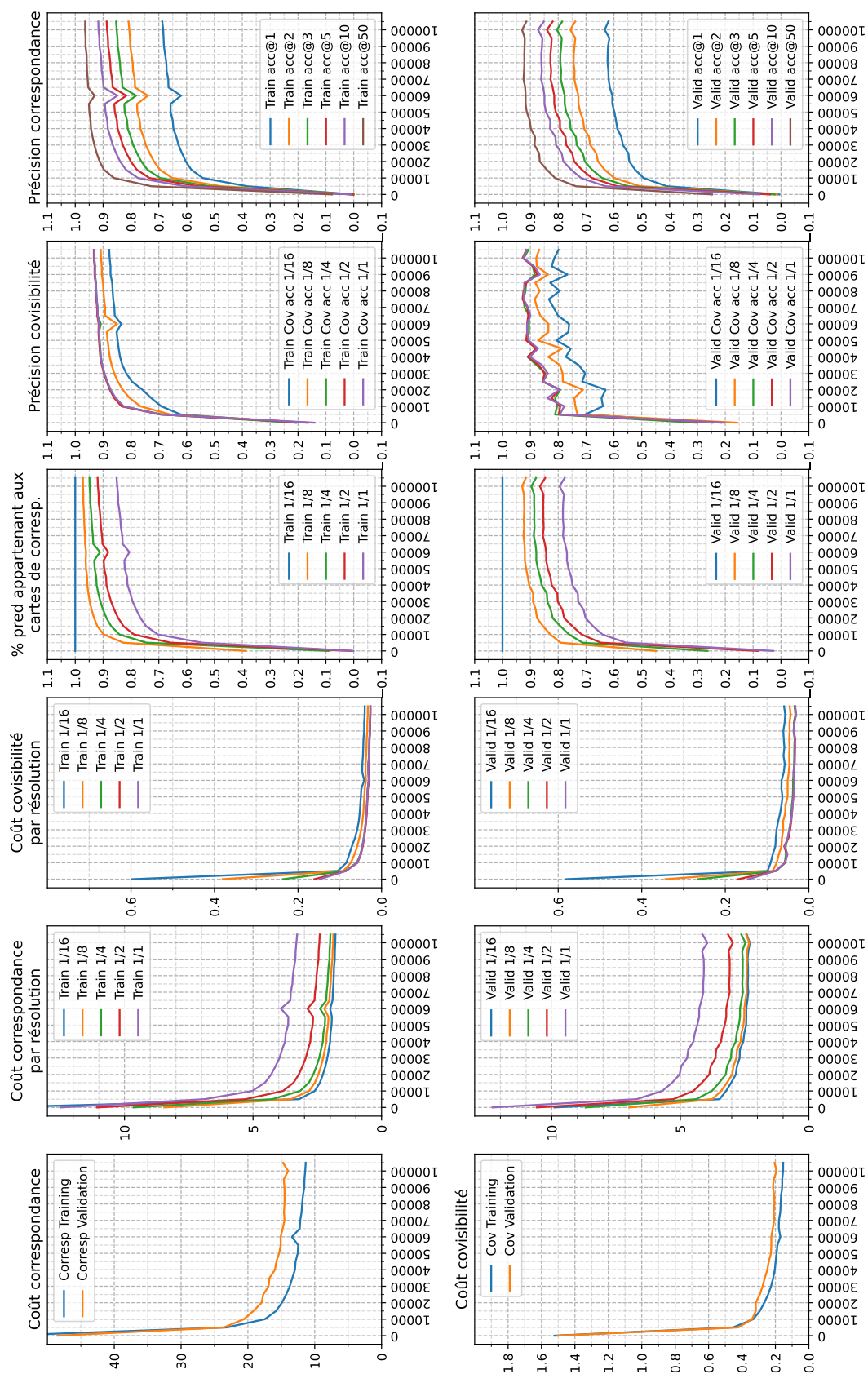


FIGURE 4.14 – **Évolution des différentes métriques au cours de l'entraînement.** Les métriques sont présentées sur l'ensemble de données d'entraînement (train) et l'ensemble de validation (valid).

## 4.4 Expériences

Dans cette partie, nous présentons des résultats qualitatifs et quantitatifs de notre méthode BEAMER et des principales méthodes de mise en correspondance de l'état de l'art. Dans un premier temps nous présentons des analyses et visualisations permettant de mieux comprendre le fonctionnement de différents éléments de l'architecture. Enfin nous présentons des résultats sur les bases de données MegaDepth, HPatches et ETH3D.

### 4.4.1 Analyse de l'architecture

#### 4.4.1.1 Visualisation de la recherche en faisceaux

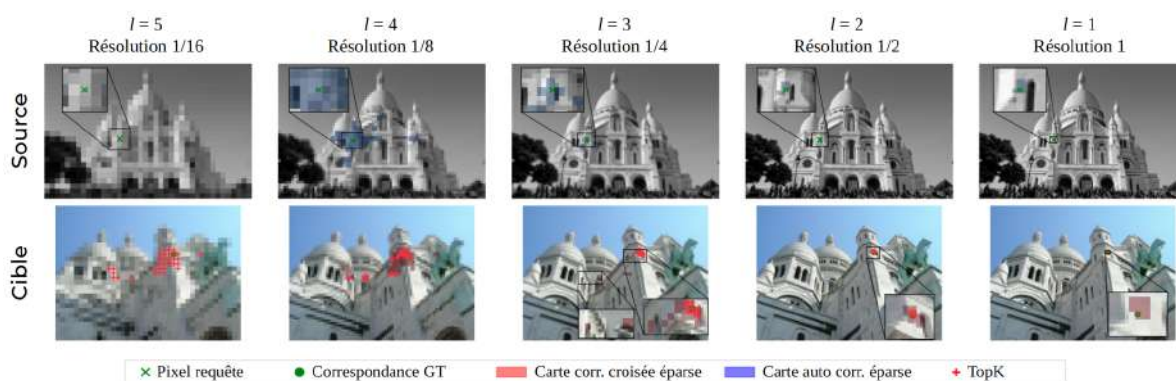


FIGURE 4.15 – Visualisation de la recherche en faisceaux pour une paire d'images de MegaDepth.

La recherche en faisceaux est l'élément central de l'architecture BEAMER. Dans la Figure 4.15, nous présentons une visualisation de cette recherche pour une correspondance. L'architecture utilise un traitement grossier à fin pour, dans un premier temps, sélectionner les régions grossières de correspondance potentielle. Puis, au fur et à mesure de la montée en résolution, le réseau affine sa prédiction en utilisant de l'auto-attention et de l'attention-croisée pour éliminer les régions ambiguës, et finit par fournir une prédiction de correspondance finale à pleine résolution. Nous avons choisi cette correspondance car elle montre une forte multimodalité au niveau grossier ( $l = 5$ ) entre des régions distantes. Comme on peut le voir dans l'image en bas à gauche, le réseau conserve cette multimodalité au niveau grossier, puis élimine les mauvaises régions une après l'autre lorsqu'il a accès à des informations plus précises, et finit par trouver le bon pixel correspondant. La partie supérieure de la Figure 4.15 montre les régions d'auto-correspondance. On peut voir qu'elles se concentrent principalement sur une région locale autour du point de requête, mais également sur les structures répétitives (symétrie du bâtiment) et des régions qui partagent certaines similarités visuelles. Le fait que le réseau se concentre principalement sur une région autour du point de requête dans les couches les plus fines lui permet certainement d'extraire des informations locales détaillées qui lui servent à trouver précisément le correspondant.

#### 4.4.1.2 Analyse de la prédiction de covisibilité

Nous nous intéressons maintenant à la prédiction de la covisibilité par l'architecture BEAMER. Nous avons défini 5 catégories de point : **visible**, **occulté**, **extérieur**, **ciel** et **sans GT**.

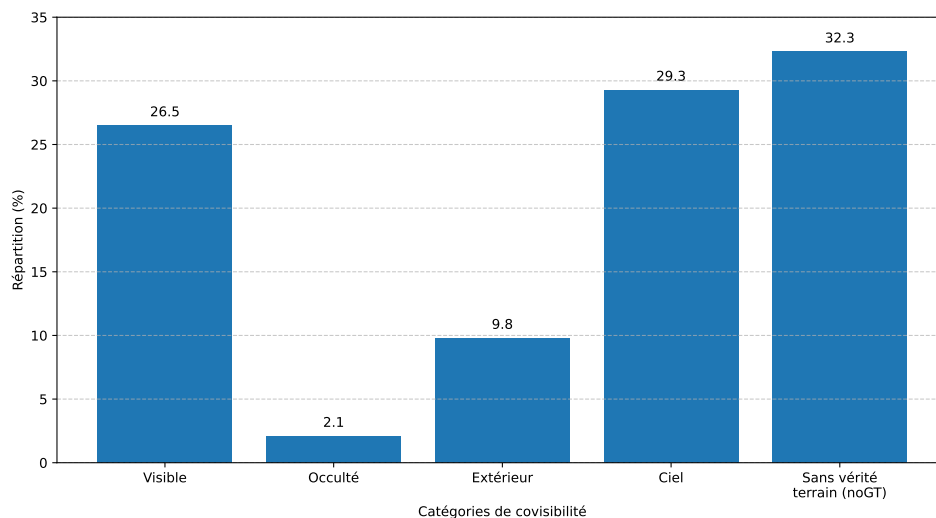


FIGURE 4.16 – **Distribution des catégories de covisibilité pour MegaDepth.** Étude faite sur 10000 paires d’images sélectionnées uniformément sur toutes les scènes de MegaDepth.

Dans un premier temps, nous réalisons une analyse sur MegaDepth pour connaître comment nos classes sont réparties à travers la base de données. Nous comptons donc le nombre de points appartenant à chaque classe pour 10000 paires d’images issues de toutes les scènes de MegaDepth.

Comme on peut le voir dans la Figure 4.16, la distribution des catégories n’est pas uniforme. La catégorie **visible** est beaucoup plus présente que les catégories **occulté** et **extérieur**. D’un côté, c’est une bonne chose car nous avons un grand nombre de points sur lesquels entraîner la partie mise en correspondance du réseau, mais d’un autre côté, la sous-représentation des catégories **occulté** et **extérieur** peut poser problème lors de l’apprentissage de la covisibilité. Cette analyse nous a permis de trouver des coefficients pour pondérer les gradients en fonction de la catégorie du point afin que le réseau puisse apprendre les différentes catégories équitablement.

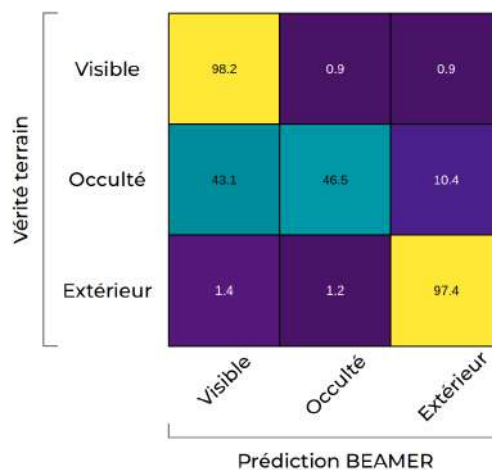


FIGURE 4.17 – **Matrice de confusion des prédictions de covisibilité de BEAMER.** Résultats obtenus sur notre ensemble de validation de MegaDepth composé de 10 scènes.

Dans la Figure 4.17, nous représentons la matrice de confusion établie par BEAMER entre les classes **visible**, **occulté** et **extérieur**. Malgré notre pondération des gradients, le réseau a du mal à distinguer les points **occulté** et les confond souvent avec des points **visible**. Cela peut s'expliquer par la difficulté de reconnaître si un point est occulté ou non, mais également par le bruit dans les cartes de profondeur servant à créer les catégories. En effet, comme la profondeur est imprécise dans certaines régions, certains points **visible** sont parfois classés comme **occulté**. Toujours sur la vérité terrain, on peut noter dans la Figure 4.16 qu'une grande partie des points n'ont pas de vérité terrain. Ces points peuvent appartenir à n'importe quelle catégorie, c'est pourquoi nous ne les utilisons pas lors de l'entraînement, contrairement à DKM qui les considère comme des points *non consistants* et bruite donc énormément le signal d'apprentissage.

La Figure 4.18 illustre comment notre architecture BEAMER estime la covisibilité entre les deux images à différents niveaux du réseau. On constate que BEAMER a une très bonne représentation générale de la covisibilité finale, même pour les régions **occulté** et **extérieur**, ce qui montre que l'architecture possède une certaine représentation de la géométrie 3D de la scène liant les deux images. On note également que cette représentation évolue au fur et à mesure de la montée en résolution, principalement pour l'image source dans notre exemple. Si la représentation grossière semble peu précise, elle devient de plus en plus fine avec la montée en résolution. Cela est assez contre-intuitif, car on pourrait penser que l'estimation de la covisibilité ne nécessite pas de détails très fins, mais cette observation est peut-être due à la multimodalité présente dans les couches les plus grossières.

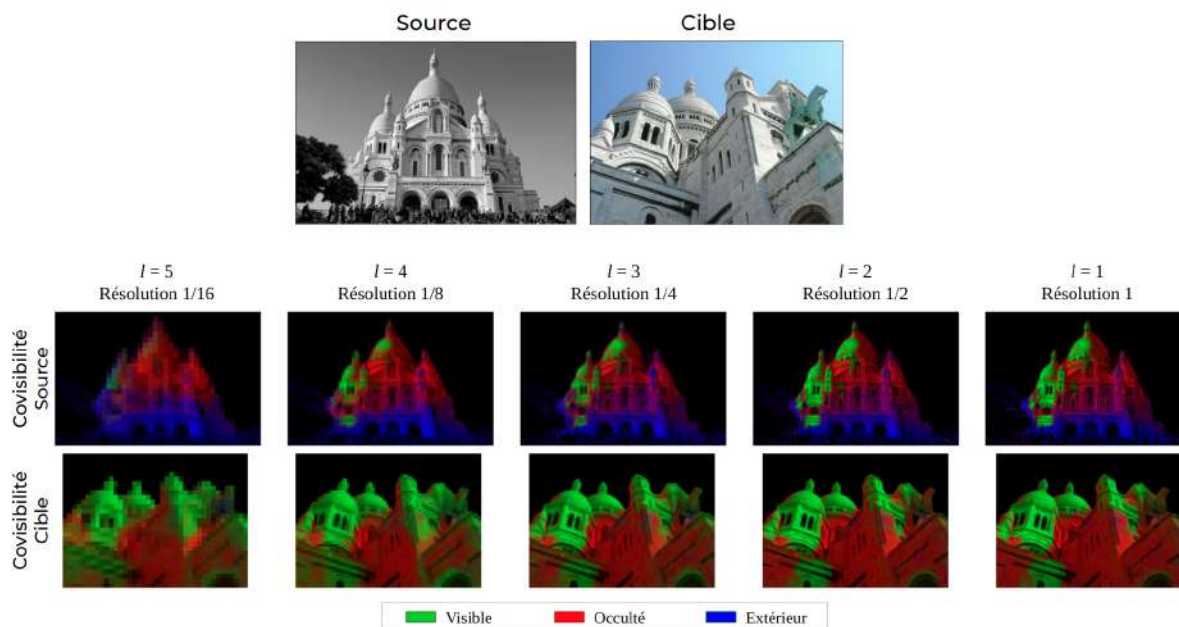


FIGURE 4.18 – **Visualisation de la prédiction de covisibilité de BEAMER.** Dans cette visualisation, les scores de covisibilité pour les classes **visible**, **occulté** et **extérieur** contrôlent respectivement l'intensité des canaux vert, rouge et bleu des images. Notre architecture prédit la covisibilité à chaque résolution. On observe que le réseau affine progressivement sa prédiction de covisibilité.

### 4.4.1.3 Stratégies d'échantillonnage

Nous nous intéressons maintenant à la stratégie d'échantillonnage des correspondances. Dans la section 4.3.2, nous avons vu comment, pour chaque point  $\mathbf{p}_{s,i}^l$  à l'échelle  $l$ , BEAMER calcule une carte de covisibilité  $\text{COV}_{\mathbf{p}_{s,i}^l}^l$ . Pour pondérer notre échantillonnage, nous utilisons le score de la classe **visible** dans cette carte de covisibilité à la résolution la plus fine  $l = 1$ . De cette manière, les points classifiés comme **occulté**, **extérieur** ou **ciel** ont une probabilité nulle ou très faible d'être sélectionnés. BEAMER permet également de considérer d'autres stratégies. Sur le même principe que les méthodes semi-denses comme LoFTR [Sun et al., 2021], nous pouvons utiliser les scores des cartes de correspondances pour mesurer la *confiance* qu'a le réseau pour chaque correspondance. Dans notre cas, pour chaque point fin  $\mathbf{p}_{s,i}^L$ , nous pouvons extraire une carte de correspondance à chaque échelle qui nous donne une distribution de probabilités. Nous utilisons donc la probabilité jointe pour pondérer l'échantillonnage des correspondances. Enfin, comme BEAMER produit des prédictions source  $\rightarrow$  cible et cible  $\rightarrow$  source, il est possible d'utiliser la prédiction de covisibilité pour vérifier que la correspondance trouvée tombe bien dans une région **visible**. Ce test supplémentaire peut être utilisé en combinaison avec les deux stratégies précédentes. Les résultats en estimation de pose de BEAMER utilisant ces stratégies sont présentés dans le tableau 4.1. Nous visualisons également dans la Figure 4.19 les cartes de scores obtenus pour les stratégies de covisibilité et de probabilité jointe, ainsi que les correspondances qu'elles produisent.

TABLE 4.1 – **Étude de la stratégie d'échantillonnage.** L'erreur de pose en AUC est rapportée en pourcentage sur MegaDepth-1500. Plus l'AUC est haute, meilleure est l'estimation de pose.

Stratégie échantillonnage	AUC@5°	AUC@10°	AUC@20°
Probabilité jointe	45.7	64.5	78.4
Probabilité jointe + bidirection	45.8	64.6	78.7
Covisibilité	52.6	69.2	81.3
Covisibilité + bidirection	53.8	70.0	81.9

Le Tableau 4.1 montre qu'utiliser les cartes de correspondances pour pondérer l'échantillonnage des correspondances est bien moins efficace que d'utiliser les cartes de covisibilité. En effet, dans la Figure 4.19, on observe que les scores issus des cartes de correspondances sont beaucoup plus faibles dans les régions uniformes par rapport aux régions texturées. On note également que l'aspect éparé des cartes de correspondances crée des artefacts dans notre carte de scores. De son côté, l'utilisation de la prédiction de la covisibilité n'est pas sensible aux régions uniformes. Cela résulte en un échantillonnage de correspondances spatialement bien mieux réparti lorsque l'on utilise la covisibilité, ce qui est indispensable pour une bonne estimation de pose de caméra. En revanche, l'estimation de la covisibilité est indépendante de la précision des correspondances, il est donc possible que de mauvaises correspondances soient sélectionnées. Le Tableau 4.1 montre également que la vérification bidirectionnelle de la covisibilité permet d'améliorer l'estimation de pose quelle que soit la stratégie d'échantillonnage.



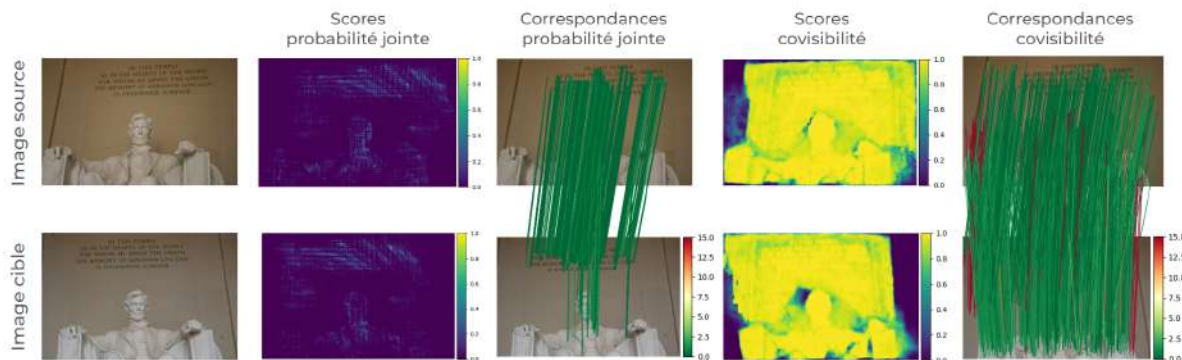


FIGURE 4.19 – **Comparaison des différentes stratégies d'échantillonnage de correspondances.** L'utilisation de la prédiction de la covisibilité permet un échantillonnage bien mieux répartie spatialement.

#### 4.4.1.4 Étude du coût

Dans cette partie nous nous intéressons au coût en temps de calcul et en mémoire de l'architecture BEAMER. Les résultats sont présentés dans le Tableau 4.2.

TABLE 4.2 – **Comparaison du temps de calcul et de l'empreinte mémoire.** Calculé pour des images  $640 \times 640$ . L'empreinte mémoire est calculée avec l'outil de gestion de mémoire GPU NVidia-SMI.

Méthodes	Temps d'inférence (s)	Mémoire GPU inférence (Gbits)	Mémoire GPU entraînement (Gbits)
LoFTR	0.12	4.5	-
DKM	0.45	7.8	-
BEAMER	0.48	9.1	15.5

Comme on peut le voir dans le Tableau 4.2, BEAMER a un temps d'inférence comparable à celui de DKM [Edstedt et al., 2023]. Cependant ces deux méthodes denses sont significativement plus lente que LoFTR [Sun et al., 2021] qui utilise le paradigme semi-dense. Même remarque pour l'empreinte mémoire, cependant BEAMER est cette fois-ci significativement plus lourde que DKM. Ceci s'explique par le fait que les cartes de correspondances et les cartes d'attentions, même si elles sont éprases à haute résolution, ont une forte empreinte mémoire. Ceci peut poser problème pour traiter des images plus haute résolution. Pendant l'entraînement nous avons utilisé des méthodes de calcul itératif des gradient (voir section 4.3.4) pour réduire l'empreinte mémoire du modèle, au détriment du temps nécessaire pour effectuer l'inférence et la modification des poids du réseau.

#### 4.4.1.5 Études d'ablations

Dans cette partie nous nous intéressons à l'analyse de différents composants de l'architecture. Le Tableau 4.3 montre l'impacte d'apprendre la covisibilité en parallèle de la mise en correspondance et le Tableau 4.4 montre l'intérêt d'une architecture bidirectionnelle.

Dans le Tableau 4.3 sont présentées les performances de deux architectures : Une architecture entraînée uniquement pour prédire des correspondances via la recherche en faisceaux, et une architecture similaire à laquelle on ajoute la tâche de prédiction de covisibilité selon le formalisme décrit dans la section 4.3.2. Pour la première (sans covisibilité) nous utilisons une



TABLE 4.3 – **Analyse de l’ajout de la prédiction de covisibilité.** Les résultats sont calculés sur MegaDepth-1500. L’ajout de la tâche de prédiction de covisibilité permet d’améliorer la précision de la mise en correspondance et permet l’utilisation d’une meilleure stratégie d’échantillonnage pour l’estimation de pose de caméra.

Méthodes	Précision de matching (MA) $\uparrow$						Estimation de pose (AUC) $\uparrow$		
	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=20$	@5°	@10°	@20°
Sans covisibilité	69.1	83.9	87.2	90.0	92.1	94.5	45.2	64.2	78.0
Avec covisibilité	71.4	84.9	87.8	90.3	92.3	94.6	53.8	70.0	81.9

stratégie d’échantillonnage basée sur la probabilité jointe pour l’estimation de pose de caméra, alors que pour la seconde (avec covisibilité) nous utilisons une stratégie basée sur l’estimation de la covisibilité. On peut voir que, en plus de permettre l’utilisation d’une meilleure stratégie d’échantillonnage, l’apprentissage de la covisibilité permet à BEAMER de produire des correspondances plus précises. Ceci montre l’intérêt d’intégrer la prédiction de la covisibilité dans l’architecture. Notre découpage en plusieurs classes de covisibilité semble apporter une information supplémentaire sur la structure 3D de la scène observée, permettant à BEAMER de produire des correspondances plus précises.

TABLE 4.4 – **Analyse de la direction de matching.** Les résultats d’estimation de pose de caméra sont calculés sur MegaDepth-1500. La prédiction bidirectionnelle de correspondances permet une meilleure estimation de pose de caméra car, pour des paires avec de fortes multimodalités, un sens est toujours plus simple qu’un autre.

Méthodes	AUC@3°	AUC@5°	AUC@10°
Prédictions unidirectionnelles	49.1	67.9	80.4
Prédictions bidirectionnelles	53.8	70.0	81.9

Dans le Tableau 4.4, nous présentons les résultats de l’estimation de pose de caméra de BEAMER en utilisant : uniquement les prédictions source  $\rightarrow$  cible (unidirectionnelles), et une combinaison des prédictions source  $\rightarrow$  cible et cible  $\rightarrow$  source (bidirectionnelles). On observe que l’utilisation des correspondances provenant uniquement de la source entraîne une chute significative des performances par rapport à l’utilisation des correspondances provenant des deux images. Cela s’explique par le fait que la multimodalité d’une paire d’images n’est pas la même dans les deux directions. Par exemple, lorsque l’image cible est un zoom avant de l’image source, la recherche *grossier à fin* dans le sens source  $\rightarrow$  cible génère de nombreuses correspondances un-à- $m$  aux niveaux grossiers, tandis que dans le sens cible  $\rightarrow$  source, ces correspondances grossières deviennent un-à-un. L’image cible comporte également beaucoup plus de pixels que la source pour la même région covisible de la scène 3D, ce qui offre une meilleure répartition spatiale lors de l’échantillonnage des correspondances directement dans la cible. Pour ces raisons, il est donc bénéfique de développer des méthodes permettant de produire des correspondances dans les deux directions (source  $\rightarrow$  cible et cible  $\rightarrow$  source) afin d’améliorer l’estimation de pose de caméra.

## 4.4.2 Estimation de pose de caméra et précision de correspondance

Dans cette partie nous présenterons les performances de BEAMER en estimation de pose relative de caméra, en estimation d'homographie et en précision de matching (voir section 2.2.1.2). Notre méthode sera comparée aux méthodes éparses, semi-denses et denses les plus performantes de l'état de l'art.

**Implémentations.** Pour chaque réseau, nous récupérons les résultats présentés par les auteurs, résultats que nous avons vérifiés pour les méthodes LoFTR, QuadTree, MatchFormer, TopicFM, 3DG-STFM, ASpanFormer, ECO-TR et DKM. En ce qui concerne BEAMER, nous l'entraînons pendant 200 heures sur MegaDepth avec quatre GPU NVIDIA V100 (16GB).

### 4.4.2.1 Résultats sur MegaDepth

Dans cette section, nous présentons des résultats quantitatifs et qualitatifs de notre architecture BEAMER sur MegaDepth [Li and Snavely, 2018]. Pour les visualisations, nous comparons nos résultats aux prédictions faites par DKM [Edstedt et al., 2023].

Pour MegaDepth, l'évaluation se fait généralement sur un sous-ensemble de scènes appelé MegaDepth-1500, comprenant 1500 paires d'images prises sur uniquement 2 scènes, ce qui limite l'évaluation de la généralisation de la méthode évaluée. Les images de MegaDepth-1500 sont également à une résolution supérieure aux autres jeux de données de test. Alors que HPatches utilise des images d'une taille maximale de 640 pixels et ETH3D d'une taille maximale de 520 pixels, MegaDepth-1500 propose d'évaluer les méthodes sur des images de taille 1200 pixels (résolution 1200). L'entraînement de BEAMER se fait sur des images de taille 640 pixels (résolution 640), et son architecture ne lui permet pas de réaliser une mise en correspondance à une aussi haute résolution. Nous avons donc évalué BEAMER sur ces mêmes images mais à une résolution maximale de 640. Établir des correspondances à une plus haute résolution va permettre une meilleure contrainte de l'estimation de la pose de caméra. Pour tout de même pouvoir se comparer sur ce jeu de données, nous avons évalué les performances de la méthode DKM [Edstedt et al., 2023] sur ces mêmes images de résolution 640. Pour référence, nous présentons également dans le Tableau 4.5 les performances des autres méthodes sur les images à résolution 1200.

Pour l'estimation de pose de caméra réalisée sur MegaDepth 1500 (voir Tableau 4.5), nous pouvons voir que BEAMER surpasse SuperGlue (matching épars), LoFTR (matching semi-dense) ainsi qu'ECO-TR et PDC-NET+ (matching dense), alors que ces méthodes utilisent des images à résolution 1200. En revanche, BEAMER propose une estimation de pose de caméra moins précise que DKM, même lorsque ce dernier utilise des images à résolution 640. Dans la Figure 4.20, nous analysons plus en détail ces résultats en comparant l'estimation de pose à la précision de la mise en correspondance. Les résultats sont présentés sous la forme de courbes cumulatives, sans se limiter aux trois seuils d'AUC présentés dans le Tableau 4.5. Nous décomposons les résultats pour les deux scènes de MegaDepth-1500 et ajoutons l'erreur de rotation ainsi que l'erreur de translation. Pour rappel, l'erreur de pose est la valeur maximale entre l'erreur de rotation et l'erreur de translation (voir section 2.2.1.2).

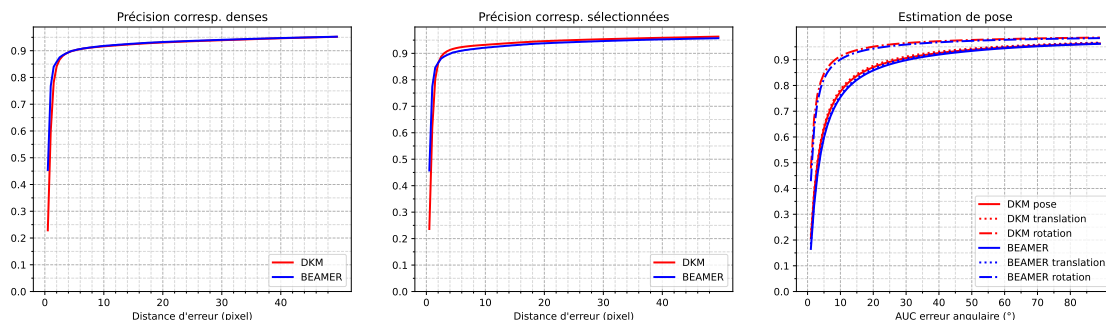
Dans la Figure 4.20, nous comparons la précision de la mise en correspondance dense, c'est-à-dire la précision sur tous les pixels avec une vérité terrain, la précision des correspondances sélectionnées par BEAMER et DKM, ainsi que l'erreur d'estimation de pose de caméra. Nous

TABLE 4.5 – Estimation de la pose de la caméra sur MegaDepth-1500. L’erreur de pose en AUC est rapportée en pourcentage. Plus l’AUC est haute, meilleure est l’estimation de pose.

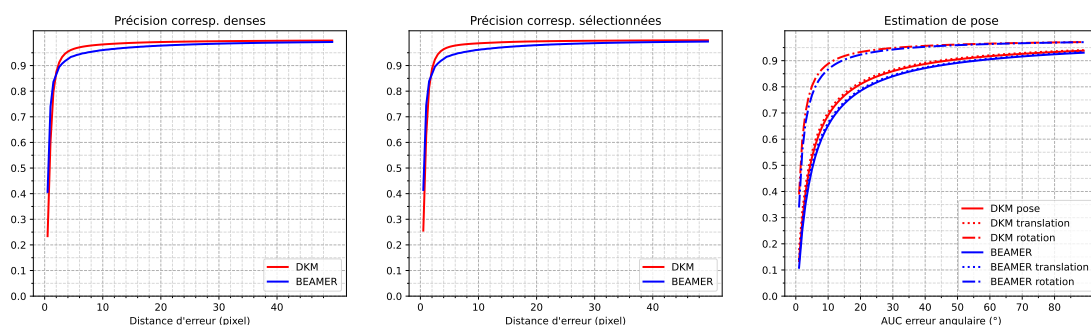
	Méthodes	AUC ↑		
		@5°	@10°	@20°
Images 1200 pixels	SuperGlue [Sarlin et al., 2020] CVPR’19	42.2	61.2	76.0
	LoFTR [Sun et al., 2021] CVPR’21	52.8	69.2	81.2
	QuadTree [Tang et al., 2022b] ICLR’22	54.6	70.5	82.2
	MatchFormer [Wang et al., 2022] ACCV’22	52.9	69.7	82.0
	TopicFM [Giang et al., 2023] ACCV’22	54.1	70.1	81.6
	3DG-STFM [Mao et al., 2022] ECCV’22	52.6	68.5	80.0
	ASpanFormer [Chen et al., 2022] ECCV’22	55.3	71.5	83.1
	DenseGAP [Kuang et al., 2022] ICPR’22	41.2	56.9	70.2
	ECO-TR [Tan et al., 2022b] ECCV’22	48.3	65.8	78.5
	PDC-Net+ [Truong et al., 2023] TPAMI’23	51.5	67.2	78.5
	ASTR [Ni et al., 2023] CVPR’23	58.4	73.1	83.8
	CasMTR [Cao and Fu, 2023] ICCV’23	59.1	74.3	84.8
	DKM [Edstedt et al., 2023] CVPR’23	60.4	74.9	85.1
Images 640 pixels	DKM [Edstedt et al., 2023] CVPR’23	57.5	73.2	84.3
	<b>BEAMER</b>	53.8	70.0	81.9

remarquons que, pour tous les pixels avec une vérité terrain, BEAMER et DKM ont des performances très similaires, BEAMER étant plus précis mais DKM légèrement plus robuste. En revanche, en comparant ces résultats à la précision des points sélectionnés par les deux méthodes, on constate que l’écart entre les deux méthodes augmente, ce qui laisse suggérer que la stratégie d’échantillonnage de BEAMER est moins efficace que celle de DKM. Cet écart est particulièrement visible pour la seconde scène de MegaDepth-1500. Cette scène a pour particularité de contenir de forts zooms entre les images (voir paire n°2 dans la Figure 4.21), et l’écart d’erreur de translation entre DKM et BEAMER semble plus important que l’écart d’erreur de rotation, laissant supposer que BEAMER pourrait avoir des difficultés pour les paires contenant des zooms. À noter que BEAMER est entraîné sur significativement moins de paires d’images que DKM, environ 350 000 pour BEAMER contre environ 10 millions pour DKM. DKM utilise également de l’augmentation de données, comme de nombreuses méthodes de l’état de l’art, alors que BEAMER ne le fait pas. Ces résultats semblent suggérer que BEAMER ne généralise pas encore suffisamment pour traiter avec précision certaines paires contenant de forts zooms.

Dans la Figure 4.21, nous présentons des visualisations qualitatives des performances de mise en correspondance de BEAMER. Nous affichons la prédiction de la covisibilité pour les images source et cible, un sous-ensemble des correspondances sélectionnées par BEAMER, ainsi que les warping pour BEAMER et DKM. Les warping *source*←*cible* présentés dans les figures correspondent à la reconstruction de l’image source avec l’information de l’image cible en suivant les correspondances prédites entre la source et la cible. Nous pondérons l’intensité de chaque pixel par sa probabilité d’appartenir à la classe **visible**. Un point ayant une pro-



(a) Résultats sur la première scène de MegaDepth-1500



(b) Résultats sur la seconde scène de MegaDepth-1500

FIGURE 4.20 – Comparaison entre les précisions de matching et l'estimation de pose pour les deux scènes de MegaDepth-1500.

tabilité nulle d'être **visible** apparaîtra donc en blanc, tandis qu'un point avec une probabilité égale à 1 d'être **visible** sera une copie de son pixel correspondant. Nous reconstruisons sur le même principe l'image cible à partir de l'information de l'image source. Nous rapportons également l'erreur moyenne dense (Err Dense), qui représente la distance moyenne en pixels entre les prédictions et les correspondances de vérité terrain pour tous les pixels visibles de la paire d'images, ainsi que l'erreur moyenne de sélection (Err Select), qui représente cette même distance d'erreur mais uniquement calculée sur les points sélectionnés par notre méthode pour l'estimation de pose.

On voit dans la Figure 4.21 que BEAMER produit des correspondances très précises et spatialement bien réparties grâce à une bonne prédiction de la covisibilité entre les images. BEAMER trouve des correspondances précises, que ce soit dans les régions texturées ou dans les régions uniformes, ce qui semble montrer que le modèle a une bonne représentation de la géométrie de la scène. En revanche, on peut observer dans la prédiction de la covisibilité que BEAMER a tendance à classer des points **occultés** comme des points **visibles**, ce qui vient compléter notre analyse de la matrice de confusion présentée dans la Figure 4.17. Notre estimation de la covisibilité semble bien plus uniforme que celle de DKM, qui semble beaucoup plus centrée sur les régions texturées. Ceci s'explique par le fait que DKM considère les points sans vérité terrain comme des points non visibles. Cette estimation de la covisibilité aboutit à des warping bien plus denses pour BEAMER.

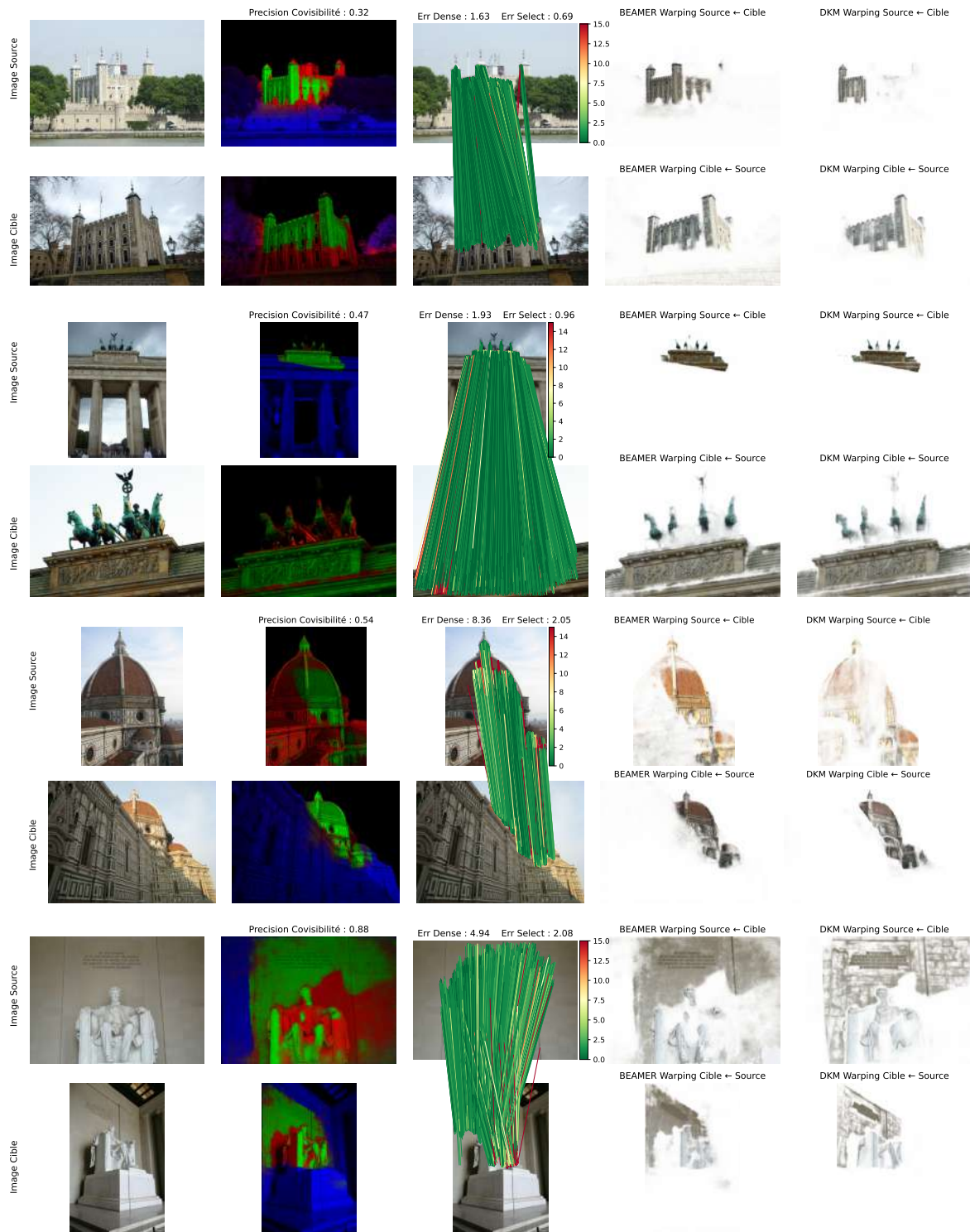


FIGURE 4.21 – **Resultats qualitatifs sur MegaDepth.** La couleur des lignes fait référence à la distance d’erreur entre le prédiction et la vérité terrain en pixel. On rapporte également la précision de covisibilité, l’erreur moyenne de mise en correspondance en pixel sur tout les points avec une vérité terrain (Err Dense) et sur les points sélectionnés (Err Select).



#### 4.4.2.2 Résultats sur HPatches

Dans cette section, nous présentons des résultats quantitatifs et qualitatifs de notre architecture BEAMER sur HPatches [Li and Snavely, 2018]. Pour les visualisations, nous comparons nos résultats aux prédictions faites par DKM [Edstedt et al., 2023].

Dans le Tableau 4.6, nous présentons les résultats d'estimation d'homographie pour HPatches. Les mêmes paires d'images sont utilisées pour évaluer toutes les méthodes. Nous rapportons l'AUC de la distance d'erreur de la reprojection des quatre coins.

TABLE 4.6 – **Estimation de l'homographie sur HPatches.** L'AUC de l'erreur de reprojection des coins est rapportée en pourcentage. Plus l'AUC est haute, meilleure est l'estimation de pose.

Méthodes	AUC $\uparrow$	@3px	@5px	@10px
SuperGlue [Sarlin et al., 2020] <small>CVPR'19</small>		53.9	68.3	81.7
LoFTR [Sun et al., 2021] <small>CVPR'21</small>		65.9	75.6	84.6
TopicFM [Giang et al., 2023] <small>ACCV'22</small>		67.3	77.0	85.7
3DG-STFM [Mao et al., 2022] <small>ECCV'22</small>		64.7	73.1	81.0
ASpanFormer [Chen et al., 2022] <small>ECCV'22</small>		67.4	76.9	85.6
PDC-Net+ [Truong et al., 2023] <small>TPAMI'23</small>		67.7	77.6	86.3
ASTR [Ni et al., 2023] <small>CVPR'23</small>		71.7	80.3	88.0
CasMTR [Cao and Fu, 2023] <small>ICCV'23</small>		71.4	80.2	87.9
PMatch [Zhu and Liu, 2023] <small>CVPR'23</small>		71.9	80.7	88.5
DKM [Edstedt et al., 2023] <small>CVPR'23</small>		71.3	80.6	88.5
BEAMER		70.0	79.2	87.3

Le Tableau 4.6 nous montre que BEAMER a des performances supérieures aux méthodes éparées, aux méthodes semi-denses et à PDC-Net+ pour les méthodes denses. Ses performances sont de l'ordre des meilleures méthodes denses mais restent inférieures. Dans la Figure 4.22, nous analysons plus en détail les différentes scènes de HPatches pour essayer de trouver les raisons de cet écart.

Dans la Figure 4.22, nous affichons la différence de distance d'erreur de reprojection des quatre coins entre BEAMER et DKM pour toutes les paires d'images de HPatches. Lorsqu'une barre rouge apparaît, cela signifie que BEAMER a fait une erreur supérieure à celle de DKM ; une barre verte indique que l'erreur était inférieure pour BEAMER, et une barre noire montre que les deux méthodes ont fait de larges erreurs. On peut observer que, pour la très grande majorité des scènes, BEAMER et DKM font de très faibles erreurs. Les différences ne se concentrent que sur quelques scènes. En visualisant ces scènes, nous remarquons que les erreurs sont commises sur des paires d'images contenant des rotations importantes (voir Figure 4.23, paire n°2). À nouveau, ceci semble montrer que BEAMER ne généralise pas encore suffisamment pour les paires contenant des rotations importantes. L'utilisation d'augmentations de données lors de l'entraînement pourrait aider à mieux généraliser ou accélérer le modèle.

Dans la Figure 4.23, nous présentons des visualisations qualitatives des performances de mise en correspondance de BEAMER. Nous affichons la prédiction de la covisibilité pour les images source et cible, un sous-ensemble des correspondances sélectionnées par BEAMER, ainsi que les warping pour BEAMER et DKM. Ces visualisations confirment nos observations



faites précédemment sur MegaDepth-1500 : BEAMER propose une grande précision de mise en correspondance et sa prédiction de la covisibilité, plus dense que celle de DKM, lui permet de construire un meilleur warping.

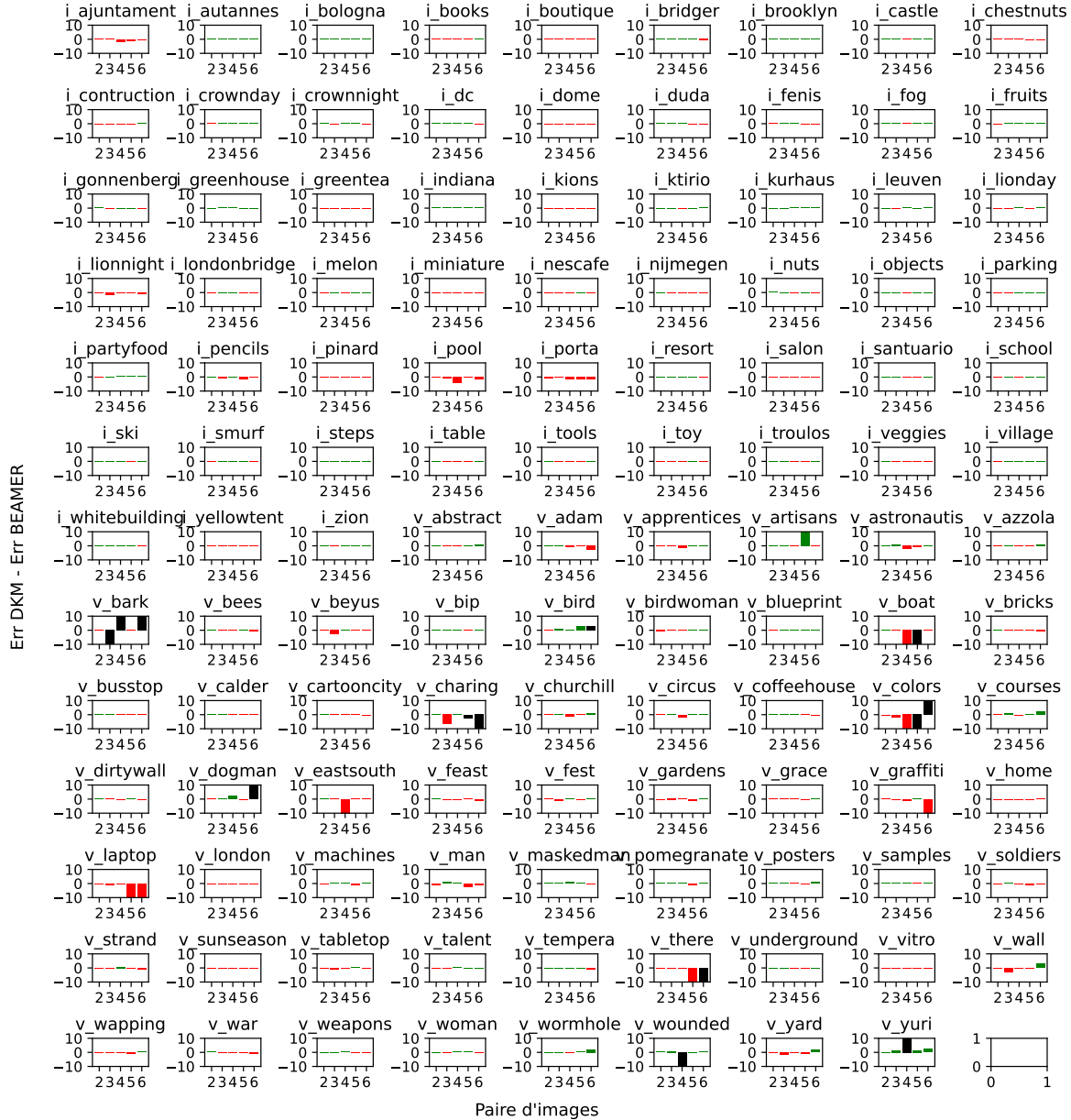


FIGURE 4.22 – Comparaison des erreurs faites entre BEAMER et DKM sur toutes les scènes de HPatches. Chaque graphe représente une scène et contient 5 paires d’images. On affiche la différence entre la distance d’erreur de reprojection des quatre coins faite par DKM et celle faite par BEAMER, en pixels. Lorsque l’erreur de BEAMER est supérieure à celle de DKM, la barre est affichée en rouge. Si l’erreur de DKM est supérieure à celle de BEAMER, la barre est en vert. Lorsque les deux méthodes font des erreurs supérieures à 10 pixels, la barre est en noir. On observe que sur 6 scènes DKM et BEAMER font tout les deux des erreurs importantes. DKM est significativement meilleure que BEAMER sur une dizaine de paires d’images, dont la grande majorité sont des rotations. Pour toutes les autres paires d’images, les performances de DKM et BEAMER sont comparables.

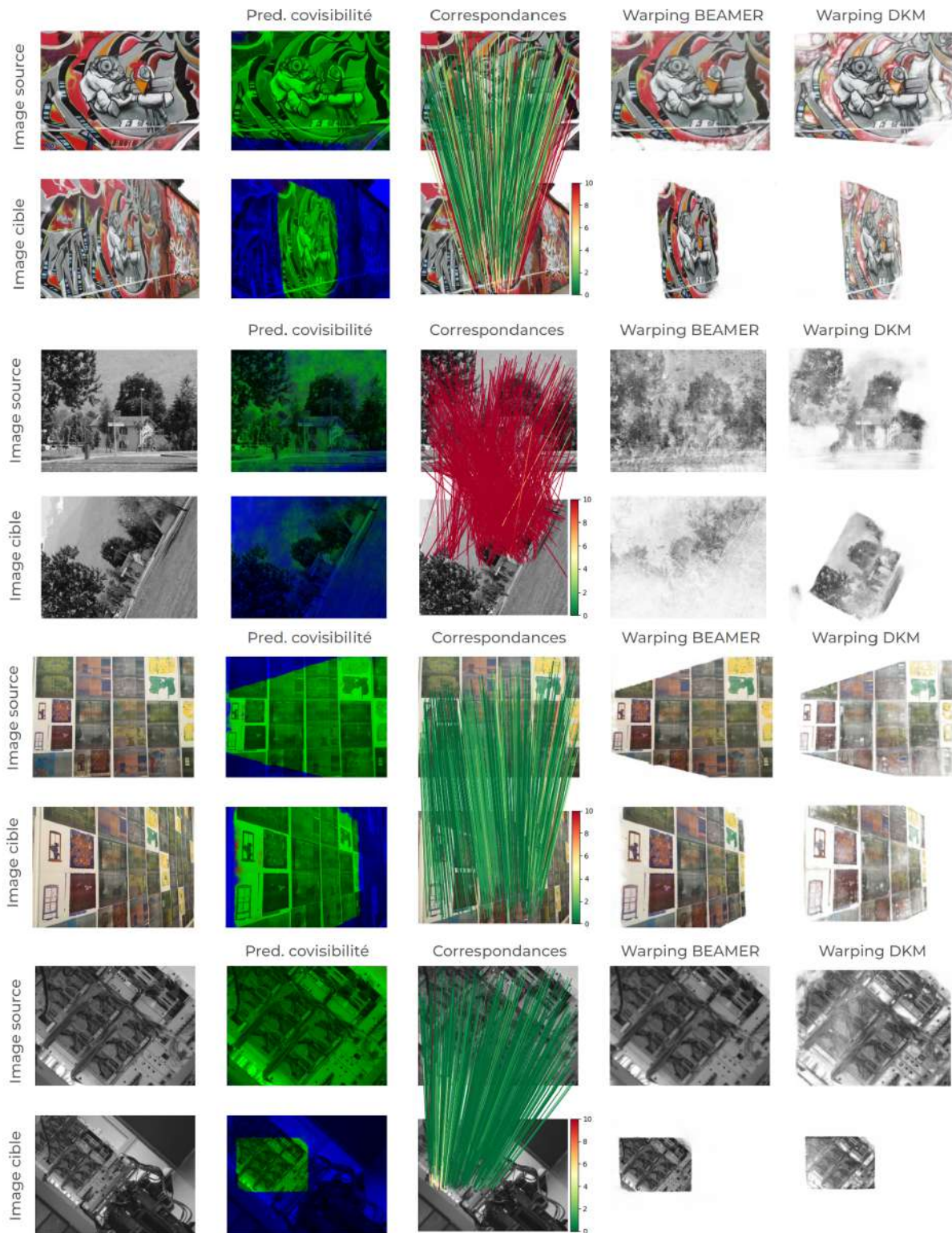


FIGURE 4.23 – **Resultats qualitatifs sur HPatches.** La couleur des lignes fait référence à la distance d'erreur entre le prédiction et la vérité terrain en pixels.

### 4.4.2.3 Résultats sur ETH3D

Dans cette section, nous présentons des résultats quantitatifs et qualitatifs de notre architecture BEAMER sur ETH3D [Schöps et al., 2019]. Pour les visualisations, nous comparons nos résultats aux prédictions faites par DKM [Edstedt et al., 2023].

ETH3D est un dataset où les paires sont issues de séquences d’images. Différents taux d’échantillonnage d’intervalle de trames  $r$  sont considérés. À mesure que le taux  $r$  augmente, le recouvrement entre les paires d’images diminue, rendant ainsi le problème de mise en correspondance plus difficile. Les résultats sont présentés dans le Tableau 4.7. Nous rapportons la précision de mise en correspondance pour plusieurs seuils  $\eta$ , calculée sur l’ensemble des points de vérité terrain. Ces points, issus d’un algorithme de Structure acquise à partir d’un mouvement (SfM), se situent tous dans des régions texturées.

TABLE 4.7 – **Evaluation sur ETH3D** pour différents taux d’échantillonnage d’intervalles d’images  $r$ . Nous rapportons la précision de correspondance pour plusieurs seuils  $\eta$

Méthodes	Précision de mathcing $\uparrow$														
	$r = 3$					$r = 7$					$r = 15$				
	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$	$\eta=1$	$\eta=2$	$\eta=3$	$\eta=5$	$\eta=10$
LoFTR	44.8	76.5	88.4	97.0	99.4	39.7	73.1	87.6	95.9	98.5	33.3	66.2	84.8	92.5	96.3
MatchFormer	45.5	77.1	89.2	97.2	99.7	40.4	73.8	87.8	96.6	99.0	34.2	66.7	84.9	93.5	97.0
TopicFM	45.1	76.9	89.0	97.2	99.6	39.9	73.5	87.9	96.4	99.0	33.8	66.4	85.0	92.8	96.5
3DG-STFM	43.9	76.3	88.0	96.9	99.3	39.3	72.7	87.4	95.5	98.3	32.4	65.7	84.7	92.0	96.0
ASpanFormer	45.8	77.6	89.6	97.8	99.8	40.6	73.8	88.1	96.8	99.0	34.3	66.8	85.3	93.9	97.3
LoFTR+QuadTree	45.9	77.5	89.5	97.8	99.7	40.8	74.0	88.3	97.0	99.2	34.5	66.8	85.4	94.0	97.3
DKM	<u>55.6</u>	<u>79.5</u>	<u>91.0</u>	<u>98.6</u>	99.6	<u>53.3</u>	<u>77.0</u>	<u>89.7</u>	<u>98.4</u>	<u>99.5</u>	<b>49.7</b>	<b>74.5</b>	<b>88.6</b>	<b>97.9</b>	<b>99.2</b>
<b>BEAMER</b>	<b>57.0</b>	<b>81.6</b>	<b>92.1</b>	<b>99.3</b>	<b>99.9</b>	<b>54.0</b>	<b>78.7</b>	<b>90.6</b>	<b>98.7</b>	<b>99.5</b>	<u>49.1</u>	<u>74.1</u>	<u>87.6</u>	<u>96.8</u>	<u>98.1</u>

Le Tableau 4.7 montre que BEAMER a des performances de mise en correspondance bien supérieures à toutes les méthodes semi-denses, quelle que soit la fréquence d’échantillonnage des images. BEAMER a également une meilleure précision de matching que DKM pour les fréquences d’échantillonnage  $r = 3$  et  $r = 5$ , mais est légèrement inférieur pour  $r = 15$ . Ces résultats confirment nos remarques faites sur MegaDepth-1500 et HPatches : BEAMER propose une grande précision de matching, comparable aux méthodes les plus performantes de l’état de l’art. La Figure 4.24 présente des résultats de mise en correspondance en comparaison avec DKM pour des paires d’images issues d’une fréquence d’échantillonnage  $r = 15$ . Dans la grande majorité des cas, les correspondances sont très précises.

### 4.4.2.4 Discussion des résultats

Les performances de BEAMER démontrent une robustesse notable en matching dense, surpassant des méthodes telles que SuperGlue, LoFTR, et d’autres architectures semi-denses ou denses dans certains scénarios, notamment sur MegaDepth. BEAMER parvient à établir des correspondances précises dans de nombreuses configurations d’images, tant dans les régions texturées que dans les zones uniformes, ce qui montre une bonne compréhension de la géométrie 3D de la scène. Cela se traduit par une estimation de pose de caméra et d’homographie généralement précises, validant l’efficacité de l’architecture proposée. L’estimation de la visibilité complète entre les images sous la forme de plusieurs classes modélisant la géométrie 3D de la scène surpasse nettement la prédiction de points *consistants* de DKM. Le warping produit par BEAMER est bien plus dense et modélise mieux les régions uniformes que celui produit par DKM.



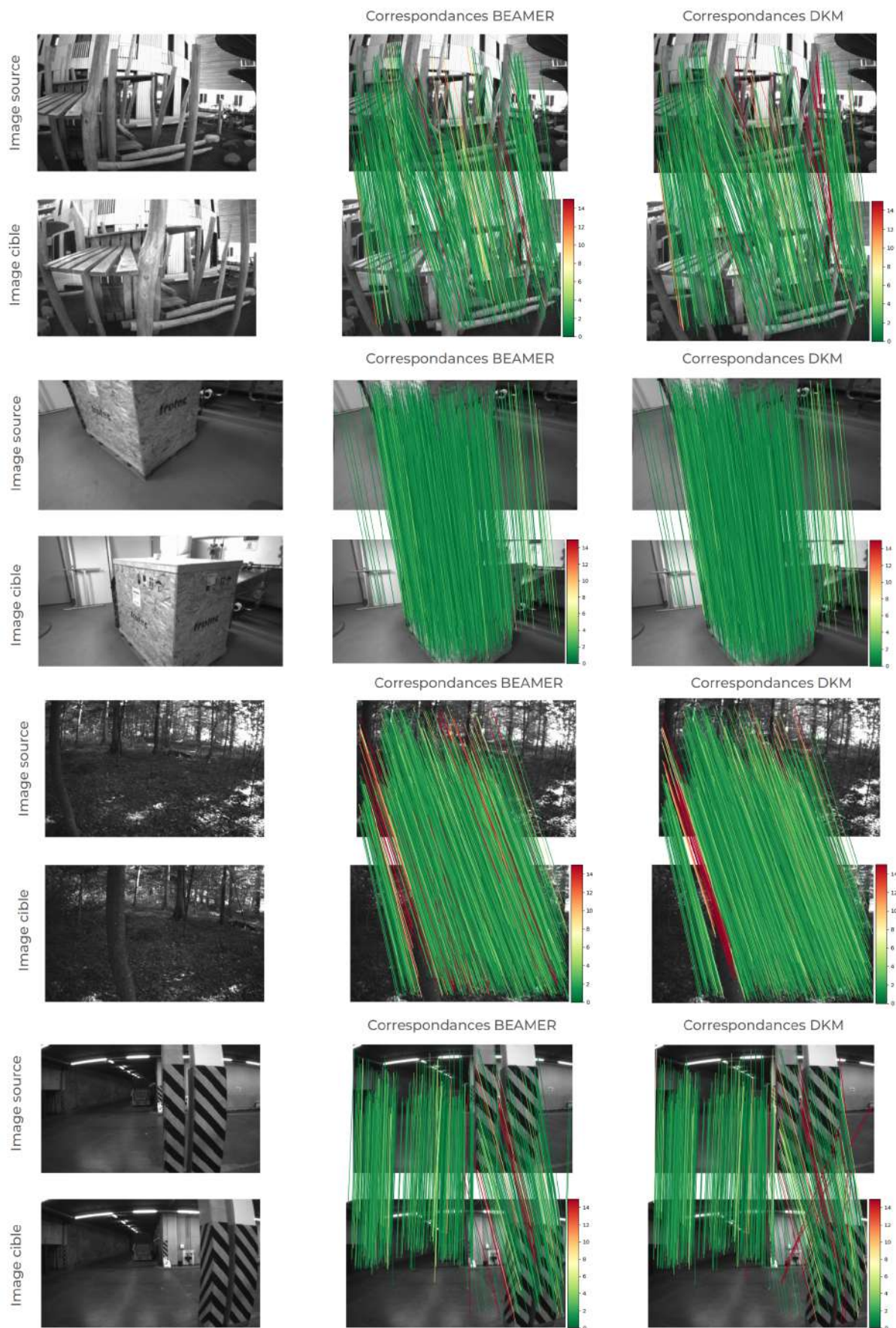


FIGURE 4.24 – Résultats qualitatifs sur ETH3D. La couleur des lignes fait référence à la distance d'erreur entre le prédiction et la vérité terrain en pixel.

Cependant, BEAMER présente certaines limitations, notamment une performance moindre comparée à DKM dans l'estimation de pose de caméra sur des images à fort zoom ou avec des rotations importantes. Ces faiblesses peuvent être attribuées à la stratégie d'échantillonnage et à une moindre capacité à généraliser dans ces configurations complexes. BEAMER présente un coût en mémoire élevé, rendant plus lent son entraînement et donc sa capacité à généraliser, et limitant sa capacité à travailler directement sur des résolutions plus élevées. Un autre axe d'amélioration se manifeste par la tendance de BEAMER à classifier certains points occultés comme visibles, ce qui suggère un défi persistant dans la prédiction de covisibilité, probablement dû au bruit dans les cartes de profondeur des datasets.

Malgré ces limitations, les résultats globaux de BEAMER démontrent une forte promesse pour le matching dense. La recherche en faisceaux couplée à la beam-attention et la prédiction de covisibilité lui permet de surpasser une majorité des méthodes de l'état de l'art alors qu'il est entraîné sur significativement moins de paires d'images. Cela souligne des axes d'amélioration, tels que l'utilisation de l'augmentation de données et l'amélioration de la stratégie d'échantillonnage, pour renforcer davantage la précision des correspondances dans des environnements variés.

## 4.5 Conclusion

Contrairement aux méthodes semi-denses, qui établissent une seule correspondance fine par région grossière des images, le matching dense vise à établir des correspondances pour chaque pixel, garantissant ainsi une couverture complète de la scène. Cela permet de capturer des détails fins et de mieux répartir spatialement les correspondances, un aspect critique pour des tâches telles que l'estimation de pose de caméra. Cependant, cette approche est confrontée à des problématiques telles que la gestion de la multimodalité des correspondances, le coût en mémoire pour les volumes de correspondances à haute résolution, et la sélection efficace des correspondances pertinentes dans des environnements visuellement complexes.

L'architecture BEAMER, présentée dans ce chapitre, a été développée pour répondre à ces défis. En adoptant une approche basée sur la recherche en faisceaux (*beam search*), BEAMER parvient à réduire de manière significative le volume de correspondances à traiter en se concentrant sur les zones les plus prometteuses à chaque échelle. Ce raffinement progressif, associé à des mécanismes d'attention-croisée et d'auto-attention, permet non seulement de gérer efficacement la multimodalité des correspondances, mais aussi d'améliorer la précision des correspondances globales. De plus, l'intégration de la prédiction de la covisibilité permet de filtrer les correspondances non pertinentes et garantit une bonne répartition spatiale.

Les expériences menées sur des benchmarks comme MegaDepth, HPatches et ETH3D ont montré que BEAMER surpasse les méthodes semi-denses en termes de précision de pose de caméra et affiche des performances compétitives avec les méthodes denses comme DKM. Notre prédiction complète de la covisibilité sous la forme de plusieurs classes modélisant la structure 3D de la scène permet à BEAMER de produire des warping plus dense que DKM, particulièrement dans les régions homogènes des images. Toutefois, certaines limites subsistent, notamment en ce qui concerne la capacité de BEAMER à gérer des résolutions plus élevées, en raison de son coût mémoire. De plus, bien que BEAMER réussisse à capturer des correspondances fines et précises dans la majorité des cas, des améliorations sont encore nécessaires pour mieux gérer les paires d'images comportant de forts zooms ou rotations.

En termes d'améliorations, plusieurs pistes peuvent être envisagées. Tout d'abord, une optimisation de l'architecture pour réduire davantage le coût mémoire serait importante pour permettre à BEAMER de traiter des images de plus haute résolution. Ceci permettrait également un entraînement plus rapide, et par conséquent, d'entraîner BEAMER sur plus de paires d'images et ainsi améliorer sa capacité de généralisation pour les rotations et les zooms. Pour réduire ce coût mémoire, nous pourrions dans un premier temps parfaire l'implémentation de nos couches de beam-attention par une implémentation CUDA optimisant la gestion mémoire de nos volumes éparses. D'un côté plus algorithmique, nous pourrions utiliser de la prédiction hiérarchique de la covisibilité pour limiter la recherche en faisceaux uniquement aux régions estimées comme covisibles, et ainsi limiter le nombre de cartes de correspondances calculées et le nombre d'opérations d'attention. Enfin, il serait intéressant d'explorer une nouvelle stratégie d'échantillonnage, toujours basée sur la covisibilité, mais intégrant des contraintes de répartition spatiale, comme le propose DKM et de nombreuses autres méthodes de l'état de l'art.

En conclusion, bien que le matching dense offre des avancées significatives en estimation de pose de caméra, il reste un domaine en évolution. Si le matching dense a prouvé sa grande précision de correspondance, le compromis entre puissance de calcul nécessaire et précision d'estimation de pose est questionnable. Pour l'entraînement, DKM et d'autres méthodes denses de l'état de l'art utilisent des GPU NVidia A100 avec 80Go de mémoire. Ces cartes sont coûteuses.



teuses à l'achat et à l'utilisation, rendant difficile le développement de méthodes similaires pour de nombreux acteurs de la communauté de mise en correspondance d'images. De plus, même pour l'inférence, les stratégies actuelles pour établir des correspondances de manière *grossier à fin* sont significativement plus lentes que les approches éparses et semi-denses. Ceci contraint les méthodes du paradigme dense aux applications hors-ligne et empêche leur utilisation pour des applications temps réel ou embarquées. La recherche en faisceaux, bien que coûteuse pour le moment avec de l'attention, permet de réduire drastiquement les espaces de recherche de correspondances tout en conservant la multimodalité existante aux niveaux grossiers. Cette approche nous semble prometteuse pour réduire le coût calculatoire des futures méthodes denses.



## **Chapitre 5**

## **Conclusion**

Dans ce chapitre, nous présenterons un résumé concis des contributions exposées dans ce manuscrit, ainsi que plusieurs pistes de recherche futures s'appuyant sur nos méthodes. Nous examinerons nos contributions dans la Section 5.1 et, enfin, nous présenterons des perspectives de travaux futurs dans la Section 5.2.

## 5.1 Contributions et discussion

**Évaluation des méthodes semi-denses avec SAM.** Dans le Chapitre 3, nous avons introduit un nouveau paradigme de mise en correspondance : le matching *sur demande*. Ce paradigme a pour but d'évaluer simplement les méthodes semi-denses (SDF) pour comprendre l'ingrédient essentiel ayant permis à ces méthodes de surpasser les méthodes éparses (S2S) basées sur la détection de points d'intérêts. Pour cela, nous avons conçu une architecture flexible, SAM (*Structured Attention image Matching*), reposant sur notre nouvelle attention structurée et un espace latent. Alors que les auteurs de LoFTR [Sun et al., 2021] supposaient que le gain de performance en estimation de pose de caméra provenait de la capacité des méthodes semi-denses à établir des correspondances précises dans les régions uniformes, nos expériences ont démontré qu'il était possible de développer une méthode peu précise dans ces régions uniformes mais très précise dans les régions texturées, tout en atteignant des performances d'estimation de pose comparables. Cette forte corrélation entre la précision des correspondances dans les régions texturées et les performances d'estimation de pose de caméra laisse supposer qu'il ne faut que quelques correspondances précises dans les régions uniformes pour contraindre suffisamment l'estimation de pose. Améliorer la précision des correspondances dans les régions texturées reste donc un facteur clé pour une bonne estimation de pose de caméra.

**Passage au paradigme de mise en correspondance dense avec BEAMER.** Dans le Chapitre 4, nous avons mis en lumière différentes problématiques survenant lors de la création de correspondances denses. Nous avons proposé une analyse montrant que le raffinement progressif de correspondances de manière *grossier à fin* entraîne naturellement un problème de multimodalité des correspondances aux niveaux grossiers. Cette multimodalité est particulièrement marquée dans les régions présentant des discontinuités de profondeur, pour des paires d'images avec de forts changements de perspectives ou comprenant des zooms importants. Ces régions avaient été soulignées par les auteurs de DKM [Edstedt et al., 2023] comme étant des zones où leur méthode avait tendance à commettre des erreurs. Pour permettre aux architectures de mieux gérer cette multimodalité, nous avons proposé de formaliser le matching *grossier à fin* sous la forme d'une recherche en faisceaux, où seules les régions les plus prometteuses sont explorées, tout en couvrant la multimodalité possible à chaque échelle. Cette formulation en faisceaux nous a permis de proposer la *beam-auto-attention* et la *beam-attention-croisée*, qui permettent de faire communiquer des espaces épars et d'utiliser le mécanisme d'attention au niveau pixellique jusqu'à pleine résolution. Au sein de la même architecture, nous avons également proposé une nouvelle manière de prédire la covisibilité complète entre les images, afin de faciliter l'échantillonnage des correspondances. Notre architecture BEAMER (*BEAM matchER*) s'est révélée très précise dans sa mise en correspondance et offre des performances d'estimation de pose prometteuses, supérieures aux méthodes semi-denses et comparables aux méthodes denses. Cependant, le coût en mémoire nécessaire pour établir des correspondances denses est très élevé, ce qui ralentit son entraînement. Néanmoins, ses premiers résultats sont prometteurs et laissent entrevoir de meilleures performances à mesure que le modèle généralisera suffisamment pour traiter avec précision des transformations comme les zooms importants ou les rotations.

## 5.2 Perspectives de travaux futurs

Les observations que nous avons faites tout au long de ce manuscrit suggèrent plusieurs pistes de recherche futures s'appuyant sur nos contributions.

**Généralisation de l'attention structurée.** Lors de nos travaux sur l'architecture SAM, nous avons développé l'attention structurée, une méthode visant à séparer l'information de position de l'information visuelle afin de construire des représentations plus riches. Cette séparation permet au réseau de créer une représentation purement positionnelle de haut niveau en parallèle de la représentation visuo-positionnelle classique. En utilisant l'attention structurée, le modèle peut mieux raisonner sur des informations spatiales, ce qui nous semble important pour la tâche de mise en correspondance nécessitant une grande précision. Nos expériences ont montré que cette approche améliore la précision des correspondances tout en n'engendrant qu'un faible surcoût calculatoire.

Il serait pertinent d'évaluer l'intégration de l'attention structurée dans d'autres méthodes, telles que les approches semi-denses comme LoFTR [Sun et al., 2021] ou denses comme notre architecture BEAMER, en remplacement de l'attention linéaire et l'attention softmax respectivement. Cela permettrait de vérifier si les gains de performance observés avec SAM se généralisent à d'autres paradigmes de mise en correspondance d'images. De plus, il serait intéressant de tester l'efficacité de l'attention structurée sur d'autres tâches de vision par ordinateur. Bien que son intérêt soit moins évident pour des tâches où l'information de position est moins critique, comme la classification d'images, elle pourrait apporter des avantages significatifs pour des tâches comme la détection d'objets ou la segmentation sémantique, où la capacité à raisonner sur des représentations purement positionnelles semble prometteur.

**Mise en correspondance multi-vues.** Les algorithmes de structure acquise à partir d'un mouvement (SfM) reconstruisent une scène en 3D à partir d'images non structurées en enchaînant plusieurs mises en correspondance d'images par paires. Une perspective intéressante serait de développer des méthodes permettant de réaliser directement la mise en correspondance entre une image source et plusieurs images cibles simultanément. Cette approche permettrait d'intégrer plus de contexte provenant des différentes vues, ce qui pourrait lever des ambiguïtés pour certaines correspondances difficiles à établir en utilisant seulement deux images. En ayant accès à davantage de vues, le réseau pourrait obtenir une meilleure compréhension de la scène 3D globale, ce qui améliorerait la précision des correspondances. Pour éviter un coût calculatoire trop élevé, l'utilisation d'un espace latent encodant toutes les images, comme cela a été fait avec l'architecture SAM, semble être une solution prometteuse. Une telle stratégie permettrait de maintenir une représentation compacte et riche de l'information tout en réduisant la complexité inhérente au traitement simultané de multiples images.

**Utilisation de la prédiction hiérarchique de la covisibilité.** Le matching dense permet d'estimer des correspondances pour chaque pixel des deux images. Or, comme nous l'avons vu dans le Chapitre 4, une grande partie de ces points ne sont pas covisibles entre les deux images. Une grande partie du calcul effectué par les méthodes de matching dense *grossier à fin* est donc dédiée à des points pour lesquels il n'existe pas de correspondance. Pour réduire le coût calculatoire de ces architectures, nous pourrions utiliser la prédiction de covisibilité à chaque échelle afin de limiter la recherche de correspondance aux régions estimées comme covisibles. Alors que ceci semble difficile pour les méthodes utilisant des réseaux de convolution comme DKM [Edstedt et al., 2023], notre formulation en recherche en faisceaux semble prometteuse,

car il suffirait de prolonger la recherche uniquement pour les régions estimées comme covisibles à l'échelle précédente. De cette manière, l'attention ne serait calculée que pour ces régions, réduisant ainsi drastiquement le coût mémoire. Cependant, comme nous l'avons observé dans l'analyse de la covisibilité (section 4.4.1.2), l'estimation de la covisibilité à un niveau grossier n'est pas toujours correcte, et arrêter la recherche en faisceaux pour des régions faussement classées comme non-covisibles pourrait impacter fortement la répartition spatiale des correspondances finales. Il faudrait probablement modifier la fonction de coût pour retirer la multimodalité de covisibilité aux niveaux grossiers et favoriser la classification des points comme visibles.

### **Amélioration du raisonnement géométrique des méthodes de mise en correspondance.**

Les méthodes actuelles de mise en correspondance d'images reposent fortement sur l'information visuelle, ce qui peut poser des problèmes dans des situations complexes, comme la présence de structures répétitives. Ces difficultés montrent la nécessité d'améliorer le raisonnement géométrique des modèles pour surmonter les limitations dues à une dépendance excessive à l'apparence visuelle.

Une première piste serait de demander aux modèles de correspondance d'extrapoler des correspondances pour des points situés en dehors du champ de vue de l'image cible, permettant ainsi de mieux capturer la continuité spatiale de la scène. Certaines approches [Germain et al., 2022] visant à *halluciner* des correspondances dans des régions sans information visuelle commencent à aborder cette problématique et montrent son intérêt pour l'estimation de pose de caméra.

Une autre piste consisterait à explorer de nouvelles méthodes de supervision, telles que l'apprentissage auto-supervisé, pour entraîner les modèles à extraire des relations géométriques de manière plus autonome. Les modèles de fondation visuelle (comme DINO [Oquab et al., 2023] ou CLIP [Radford et al., 2021]) ont montré une certaine compréhension de la structure 3D des images qu'ils observent [El Banani et al., 2024], alors qu'ils sont simplement entraînés de manière non supervisée sur des singletons d'images. Il semble prometteur de prolonger ces travaux dans le cadre spécifique de la mise en correspondance d'images.

Enfin, une approche alternative pourrait consister à ne plus chercher à établir des correspondances point à point, mais à prédire directement les transformations géométriques (rotation et translation) entre les images [Blanton et al., 2022]. Cela permettrait au modèle de se concentrer sur l'estimation de la relation géométrique entre les vues, sans avoir besoin de correspondances exactes, ce qui pourrait être particulièrement bénéfique dans des environnements où l'information visuelle est insuffisante ou ambiguë.



# Bibliographie

- Agarwal, S., Furukawa, Y., Sna, N., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Reconstructing rome. Computer (Long Beach Calif.), pages 40–47.
- Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. Communications of the ACM, page 105–112.
- Anderson, M. A. (2020). Structure from motion and archaeological excavation : experiences of the via consolare project in pompeii. Studies in Digital Heritage, pages 78–107.
- Apple (2024). Apple vision pro. <https://www.apple.com/fr/apple-vision-pro/>.
- Bailo, O., Rameau, F., Joo, K., Park, J., Bogdan, O., and Kweon, I. S. (2018). Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. Pattern Recognition Letters, pages 53–60.
- Balntas, V., Lenc, K., Vedaldi, A., and Mikolajczyk, K. (2017). HPatches : a benchmark and evaluation of handcrafted and learned local descriptors. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. European Conference on Computer Vision (ECCV), page 404–417.
- Bello, I. (2021). Lambdanetworks : Modeling long-range interactions without attention. International Conference on Learning Representations (ICLR).
- Blanton, H., Workman, S., and Jacobs, N. (2022). A structure-aware method for direct pose estimation. IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2019–2028.
- Broadbent, D. E. (1958). Perception and communication. Pergamon Press.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping : Toward the robust-perception age. IEEE Transactions on Robotics, pages 1309–1332.
- Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief : Binary robust independent elementary features. European Conference on Computer Vision (ECCV), page 778–792.
- Cao, B., Araujo, A., and Sim, J. (2020). Unifying deep local and global features for image search. European Conference on Computer Vision (ECCV).
- Cao, C. and Fu, Y. (2023). Improving transformer-based image matching by cascaded capturing spatially informative keypoints. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Cao, S.-Y., Yu, B., Luo, L., Chen, S.-J., Li, C., and Shen, H.-L. (2022). Pcnets : A structure similarity enhancement method for multispectral and multimodal image registration. Information Fusion.

- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. European Conference on Computer Vision (ECCV).
- Chang, J.-R. and Chen, Y.-S. (2018). Pyramid stereo matching network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5410–5418.
- Chen, B., Dao, T., Winsor, E., Song, Z., Rudra, A., and Ré, C. (2021). Scatterbrain : Unifying sparse and low-rank attention approximation. International Conference on Neural Information Processing Systems (NeurIPS).
- Chen, H., Luo, Z., Zhou, L., Tian, Y., Zhen, M., Fang, T., McKinnon, D., Tsin, Y., and Quan, L. (2022). Aspanformer : Detector-free image matching with adaptive span transformer. European Conference on Computer Vision (ECCV).
- Chen, Y., Liu, Y., Wu, K., Nie, Q., Xu, S., Ma, H., Wang, B., and Wang, C. (2024). Hcpm : Hierarchical candidates pruning for efficient detector-free matching. arXiv.
- Chidananda Gowda, K. and Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern Recognition, page 105–112.
- Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., and Shen, C. (2021). Twins : Revisiting the design of spatial attention in vision transformers. International Conference on Neural Information Processing Systems (NeurIPS).
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017). ScanNet : richly-annotated 3D reconstructions of indoor scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. (2022). Flashattention : Fast and memory-efficient exact attention with io-awareness. International Conference on Neural Information Processing Systems (NeurIPS).
- DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). Superpoint : Self-supervised interest point detection and description. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert : Pre-training of deep bi-directional transformers for language understanding. Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. arXiv.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet : Learning optical flow with convolutional networks. IEEE International Conference on Computer Vision (ICCV), pages 2758–2766.
- Edstedt, J., Athanasiadis, I., Wadenbäck, M., and Felsberg, M. (2023). DKM : dense kernelized feature matching for geometry estimation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- El Banani, M., Raj, A., Maninis, K.-K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., and Jampani, V. (2024). Probing the 3d awareness of visual foundation models. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 21795–21806.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, page 381–395.

- Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination : Consistency properties. International Statistical Review, page 238.
- Germain, H., Bourmaud, G., and Lepetit, V. (2020). S2DNet : learning image features for accurate sparse-to-dense matching. European Conference on Computer Vision (ECCV).
- Germain, H., Lepetit, V., and Bourmaud, G. (2022). Visual correspondence hallucination. International Conference on Learning Representations (ICLR).
- Giang, K. T., Song, S., and Jo, S. (2023). TopicFM : Robust and interpretable feature matching with topic-assisted. Association for the Advancement of Artificial Intelligence (AAAI).
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference (ACV), pages 147–151.
- Hassaballah, M., Alshazly, H., and Ali, A. (2019). Analysis and evaluation of keypoint descriptors for image matching. Studies in Computational Intelligence, pages 113–140.
- Hassanin, M., Anwar, S., Radwan, I., Khan, F. S., and Mian, A. (2024). Visual attention methods in deep learning : An in-depth survey. Information Fusion Journal.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Heinly, J., Schonberger, J. L., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the world\* in six days \*(as captured by the yahoo 100 million image dataset). IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. (2019). Axial attention in multidimensional transformers. arXiv.
- Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. (2018). Music transformer. arXiv.
- Huang, H., Nielsen, J., Nelson, M. D., and Liu, L. (2005). Image-matching as a medical diagnostic support tool (dst) for brain diseases in children. Computerized Medical Imaging and Graphics, pages 195–202.
- Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., Dai, J., and Li, H. (2022). Flowformer : A transformer architecture for optical flow. European Conference on Computer Vision (ECCV), pages 668–685.
- Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O'Connor, J., and Rosette, J. (2019). Structure from motion photogrammetry in forestry : A review. Current Forestry Reports, pages 155–168.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0 : Evolution of optical flow estimation with deep networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2462–2470.
- Itti, L. and Koch, C. (2001). Computational modelling of visual attention. Nature Reviews Neuroscience, pages 194–203.
- Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1254–1259.
- Jacquet, P. (2014). Map digital photography blog. <https://miseaupoint.org/blog/en/photo-stitching-introduction.html>.

- Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Henaff, O. J., Botvinick, M., Zisserman, A., Vinyals, O., and Carreira, J. (2022). Perceiver IO : A general architecture for structured inputs & outputs. International Conference on Learning Representations (ICLR).
- Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (2021). Perceiver : General perception with iterative attention. International Conference on Machine Learning (ICML).
- James, A. P. and Dasarthy, B. V. (2014). Medical image fusion : A survey of the state of the art. Information Fusion, pages 4–19.
- Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., and Yi, K. M. (2021). COTR : Correspondence transformer for matching across images. IEEE International Conference on Computer Vision (ICCV).
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Židek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. Nature, page 583–589.
- Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are RNNs : Fast autoregressive transformers with linear attention. International Conference on Machine Learning (ICML).
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything. IEEE International Conference on Computer Vision (ICCV).
- Kitaev, N., Łukasz Kaiser, and Levskaya, A. (2020). Reformer : The efficient transformer. arXiv.
- Kuang, Z., Li, J., He, M., Wang, T., and Zhao, Y. (2022). DenseGAP : graph-structured dense correspondence learning with anchor points. International Conference on Pattern Recognition (ICPR), pages 542–549.
- Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., Yang, L., Liu, J., Fan, H., and Liu, S. (2022). Practical stereo matching via cascaded recurrent network with adaptive correlation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 16263–16272.
- Li, Z. and Snavely, N. (2018). Megadepth : Learning single-view depth prediction from internet photos. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Lindenberger, P., Sarlin, P.-E., and Pollefeys, M. (2023). Lightglue : Local feature matching at light speed. IEEE International Conference on Computer Vision (ICCV).
- Lipson, L., Teed, Z., and Deng, J. (2021). Raft-stereo : Multilevel recurrent field transforms for stereo matching. International Conference on 3D Vision (3DV), pages 218–227.
- Liu, C., Yuen, J., and Torralba, A. (2011). SIFT flow : dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 978–994.

- Liu, S., Nie, X., and Hamid, R. (2022). Depth-guided sparse structure-from-motion for movies and tv shows. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15980–15989.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer : Hierarchical vision transformer using shifted windows. *IEEE International Conference on Computer Vision (ICCV)*.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1150–1157.
- Mao, R., Bai, C., An, Y., Zhu, F., and Lu, C. (2022). 3DG-STFM : 3D geometric guided student-teacher feature matching. *European Conference on Computer Vision (ECCV)*.
- Melekhov, I., Tiulpin, A., Sattler, T., Pollefeys, M., Rahtu, E., and Kannala, J. (2019). DGC-Net : dense geometric correspondence network. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1034–1042.
- Mur-Artal, R. and Tardos, J. D. (2017). Orb-slam2 : An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, page 1255–1262.
- Ni, J., Li, Y., Huang, Z., Li, H., Bao, H., Cui, Z., and Zhang, G. (2023). PATS : patch area transportation with subdivision for local feature matching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 756–777.
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2 : Learning robust visual features without supervision. *Transactions on Machine Learning Research*.
- Özyeşil, O., Voroninski, V., Basri, R., and Singer, A. (2017). A survey of structure from motion\*. *Acta Numerica*, pages 305–364.
- Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., and Castellanos, J. A. (2023). A survey on active simultaneous localization and mapping : State of the art and new frontiers. *IEEE Transactions on Robotics*, pages 1686–1705.
- Qi, W., Mingkuan, L., Xianhong, C., and Mengwen, X. (2023). Multi-task piano transcription with local relative time attention. *Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 966–971.
- Radenovic, F., Tolias, G., and Chum, O. (2019). Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1655–1668.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning (ICML)*, pages 8748–8763.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *openai.com*.
- Rahman, M. M., Bhattacharya, P., and Desai, B. C. (2007). A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. *IEEE transactions on Information Technology in Biomedicine*, pages 58–69.
- Rocco, I., Arandjelović, R., and Sivic, J. (2020a). Efficient neighbourhood consensus networks via submanifold sparse convolutions. *European Conference on Computer Vision (ECCV)*.

- Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., and Sivic, J. (2018). Neighbourhood consensus networks. International Conference on Neural Information Processing Systems (NeurIPS).
- Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., and Sivic, J. (2020b). NC-Net : neighbourhood consensus networks for estimating image correspondences. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).
- Ross, T.-Y. and Dollár, G. (2017). Focal loss for dense object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2980–2988.
- Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. European Conference on Computer Vision (ECCV), page 430–443.
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb : An efficient alternative to sift or surf. IEEE International Conference on Computer Vision (ICCV).
- Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). Superglue : Learning feature matching with graph neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Sattler, T., Leibe, B., and Kobbelt, L. (2012). Improving image-based localization by active correspondence search. European Conference on Computer Vision (ECCV), page 752–765.
- Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., and Pajdla, T. (2017). Are large-scale 3d models really necessary for accurate visual localization ? IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Schlag, I., Irie, K., and Schmidhuber, J. (2021). Linear transformers are secretly fast weight programmers. International Conference on Machine Learning (ICML).
- Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M. (2016). Pixelwise view selection for unstructured multi-view stereo. European Conference on Computer Vision (ECCV).
- Schöps, T., Sattler, T., and Pollefeys, M. (2019). BAD SLAM : Bundle adjusted direct RGB-D SLAM. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W. R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., and Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. Nature, page 706–710.
- Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).
- Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K. C., See, S., Qin, H., Dai, J., and Li, H. (2023). Flowformer++ : Masked cost volume autoencoding for pretraining optical flow estimation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1599–1610.
- Solsona, S. P., Maeder, M., Tauler, R., and De Juan, A. (2017). A new matching image preprocessing for image data fusion. Chemometrics and Intelligent Laboratory Systems, pages 32–42.
- Song, G. and Li, Y. (2021). Fast feature matching augmented reality method on dynamic image towards multimedia. International Symposium on Artificial Intelligence and its Application on Media (ISAIAM), pages 101–106.



- Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. (2023). Roformer : Enhanced transformer with rotary position embedding. Neurocomputing Journal.
- Sui, X., Li, S., Geng, X., Wu, Y., Xu, X., Liu, Y., Goh, R., and Zhu, H. (2022). Craft : Cross-attentional flow transformer for robust optical flow. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 17602–17611.
- Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). Pwc-net : Cnns for optical flow using pyramid, warping, and cost volume. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8934–8943.
- Sun, J., Shen, Z., Wang, Y., Bao, H., and Zhou, X. (2021). LoFTR : detector-free local feature matching with transformers. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Sun, J. J. (2014). Astronomical image matching based on the cross-correlation algorithm. Advanced Materials Research, pages 3827–3833.
- Sun, K., Yu, J., Tao, W., Li, X., Tang, C., and Qian, Y. (2023). A unified feature-spatial cycle consistency fusion framework for robust image matching. Information Fusion.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. arXiv.
- Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., and Torii, A. (2018). InLoc : Indoor visual localization with dense matching and view synthesis. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Tan, D., Liu, J.-J., Chen, X., Chen, C., Zhang, R., Shen, Y., Ding, S., and Ji, R. (2022a). Eco-tr : Efficient correspondences finding via coarse-to-fine refinement. European Conference on Computer Vision (ECCV).
- Tan, D., Liu, J.-J., Chen, X., Chen, C., Zhang, R., Shen, Y., Ding, S., and Ji, R. (2022b). ECO-TR : efficient correspondences finding via coarse-to-fine refinement. European Conference on Computer Vision (ECCV).
- Tang, L., Yuan, J., and Ma, J. (2022a). Image fusion in the loop of high-level vision tasks : A semantic-aware real-time infrared and visible image fusion network. Information Fusion, page 28–42.
- Tang, S., Zhang, J., Zhu, S., and Tan, P. (2022b). Quadtree attention for vision transformers. International Conference on Learning Representations (ICLR).
- Teed, Z. and Deng, J. (2020). Raft : Recurrent all-pairs field transforms for optical flow. European Conference on Computer Vision (ECCV), pages 402–419.
- Treisman, A. M. (1964). Selective attention in man. British Medical Bulletin, page 12–16.
- Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. Cognitive Psychology, page 97–136.
- Treue, S. (2003). Visual attention : the where, what, how and why of saliency. Current Opinion in Neurobiology, pages 428–432.
- Truong, P., Danelljan, M., Gool, L. V., and Timofte, R. (2021a). Learning accurate dense correspondences and when to trust them. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Truong, P., Danelljan, M., and Timofte, R. (2020). Glu-net : Global-local universal network for dense flow and correspondences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

- Truong, P., Danelljan, M., Timofte, R., and Van Gool, L. (2023). PDC-Net+ : enhanced probabilistic dense correspondence network. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).
- Truong, P., Danelljan, M., Van Gool, L., and Timofte, R. (2021b). Learning accurate dense correspondences and when to trust them. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors : A survey. Foundations and Trends in Computer Graphics and Vision, pages 177–280.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. International Conference on Neural Information Processing Systems (NeurIPS).
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. arXiv.
- Vilain, M., Giraud, R., Germain, H., and Bourmaud, G. (2024). Are semi-dense detector-free methods good at matching local features ? International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP).
- Vyas, A., Katharopoulos, A., and Fleuret, F. (2020). Fast transformers with clustered attention. International Conference on Neural Information Processing Systems (NeurIPS).
- Wang, Q., Zhang, J., Yang, K., Peng, K., and Stiefelhagen, R. (2022). Matchformer : Interleaving attention in transformers for feature matching. Asian Conference on Computer Vision (ACCV).
- Wang, S., Li, B. Z., Khabisa, M., Fang, H., and Ma, H. (2020). Linformer : Self-attention with linear complexity. arXiv.
- Wang, Y., Zhao, R., Liang, L., Zheng, X., Cen, Y., and Kan, S. (2021). Block-based image matching for image retrieval. Journal of Visual Communication and Image Representation, page 102998.
- Waymo (2019). Waymo self driving. <https://waymo.com/>.
- Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow : Large displacement optical flow with deep matching. IEEE International Conference on Computer Vision (ICCV).
- Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., and Zhang, L. (2021a). Cvt : Introducing convolutions to vision transformers. IEEE International Conference on Computer Vision (ICCV).
- Wu, K., Peng, H., Chen, M., Fu, J., and Chao, H. (2021b). Rethinking and improving relative position encoding for vision transformer. In IEEE International Conference on Computer Vision (ICCV), pages 10033–10041.
- Xu, G., Cheng, J., Guo, P., and Yang, X. (2022a). Attention concatenation volume for accurate and efficient stereo matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 12981–12990.
- Xu, H. and Zhang, J. (2020). Aanet : Adaptive aggregation network for efficient stereo matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1959–1968.
- Xu, H., Zhang, J., Cai, J., Rezatofighi, H., and Tao, D. (2022b). Gmflow : Learning optical flow via global matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8121–8130.

- Yadav, S. and Singh, A. (2016). An image matching and object recognition system using web-cam robot. International Conference on Parallel, Distributed and Grid Computing (PDGC), pages 282–286.
- Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). LIFT : Learned invariant feature transform. European Conference on Computer Vision (ECCV).
- Yi, K. M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., and Fua, P. (2018). Learning to find good correspondences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Z., L., Gong, R., Y., C., and K., S. (2023). Fine-grained position helps memorizing more, a novel music compound transformer model with feature interaction fusion. Association for the Advancement of Artificial Intelligence (AAAI).
- Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., and Liao, H. (2019). Learning two-view correspondences and geometry using order-aware network. IEEE International Conference on Computer Vision (ICCV).
- Zhang, S., Wang, Z., Wang, Q., Zhang, J., Wei, G., and Chu, X. (2021). Ednet : Efficient disparity estimation with cost volume combination and attention-based spatial residual. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5433–5442.
- Zhang, Z., Sattler, T., and Scaramuzza, D. (2020). Reference pose generation for long-term visual localization via learned features and view synthesis. International Journal of Computer Vision (IJCV), page 821–844.
- Zhao, Y., Xu, S., Bu, S., Jiang, H., and Han, P. (2019). Gslam : A general slam framework and benchmark. IEEE International Conference on Computer Vision (ICCV).
- Zhou, C., Su, S., Chen, Q., and Fan, R. (2023). E3cm : Epipolar-constrained cascade correspondence matching. Neurocomputing.
- Zhou, Q., Sattler, T., and Leal-Taixe, L. (2021). Patch2Pix : Epipolar-guided pixel-level correspondences. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Zhou, Z., Jin, H., and Ma, Y. (2012). Robust plane-based structure from motion. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1482–1489.
- Zhu, S. and Liu, X. (2023). PMatch : paired masked image modeling for dense geometric matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).