Deep Learning based 3D reconstruction : supervision and representation

François Darmon

▶ **To cite this version:**

François Darmon. Deep Learning based 3D reconstruction : supervision and representation. Artificial Intelligence [cs.AI]. École des Ponts ParisTech, 2022. English. NNT : 2022ENPC0024 . tel-03953709

HAL Id: tel-03953709
https://pastel.hal.science/tel-03953709

Submitted on 24 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École des Ponts
ParisTech

THÈSE DE DOCTORAT

de l'École des Ponts ParisTech

Deep Learning based 3D reconstruction: supervision and representation

École doctorale MSTIC : Mathématiques et Sciences et Technologies de l'Information et de la Communication

Informatique

Thèse préparée au sein du LIGM-IMAGINE

Financement : CIFRE, Thales LAS France

Thèse soutenue le 1 juin 2022, par

François Darmon

Composition du jury:

Gabriele, FACCIOLO ENS Paris-Saclay	*Président du jury*
Julie, DIGNE CNRS, LIRIS	*Rapporteur*
Tamy, BOUBEKEUR Adobe Research	*Rapporteur*
Julie, DELON Université Paris Cité	*Examinatrice*
Pascal, MONASSE École des Ponts ParisTech	*Directeur de thèse*
Mathieu, AUBRY École des Ponts ParisTech	*Co-directeur de thèse*
Bénédicte, BASCLE Thales LAS France	*Co-encadrante*
Jean-Clément, DEVAUX Thales LAS France	*Co-encadrant*

École des Ponts ParisTech
LIGM-IMAGINE
6, Av Blaise Pascal - Cité Descartes
Champs-sur-Marne
77455 Marne-la-Vallée cedex 2
France

Thales LAS France
Service Traitement des Images
2 Av. Gay Lussac
78990 Élancourt France

Abstract

3D reconstruction is a long standing problem in computer vision. Yet, state-of-the-art methods still struggle when the images used have large illumination changes, many occlusions or limited textures. Deep Learning holds promises of improving 3D reconstruction in such setups, but classical methods still produce the best results. In this thesis we analyse the specificity of deep learning applied to multiview 3D reconstruction and introduce new deep learning based methods.

The first contribution of this thesis is an analysis of the possible supervision for training Deep Learning models for sparse image matching. We introduce a two-step algorithm that first computes low resolution matches using deep learning and then matches classical local features inside the matches regions. We analyze several levels of supervision and show that our new epipolar supervision leads to the best results.

The second contribution is also a study of supervision for Deep Learning but applied to another scenario: calibrated 3D reconstruction in the wild. We show that existing unsupervised methods do not work on such data and we introduce a new training technique that solves this issue. We then exhaustively compare unsupervised approach and supervised approaches with different network architectures and training data.

Finally, our third contribution is about data representation. The neural implicit representation was recently used for image rendering. We adapt this representation to the multiview reconstruction problem and we introduce a new method that, similar to classical 3D reconstruction techniques, optimizes photo-consistency between projections of multiple images. Our approach outperforms state-of-the-art by a large margin.

Résumé

La reconstruction 3D est un problème classique en vision par ordinateur. Pourtant, les meilleures méthodes ne fonctionnent toujours pas parfaitement lorsque les images utilisées présentent de grands changements d'illumination et de nombreuses occlusions. L'apprentissage profond (Deep Learning) promet d'améliorer la reconstruction 3D dans de telles configurations, mais les méthodes classiques produisent encore les meilleurs résultats aujourd'hui. Dans cette thèse, nous analysons la spécificité de l'apprentissage profond appliqué à la reconstruction 3D multi-vues et nous introduisons de nouvelles méthodes basées sur l'apprentissage profond.

La première contribution de cette thèse est une analyse des différentes supervisions possibles pour l'entraînement de modèles d'apprentissage profond pour l'appariement d'images. Nous introduisons un algorithme en deux étapes qui calcule d'abord des correspondances à basse résolution en utilisant l'apprentissage profond, puis des correspondances de points d'intérêt classiques à l'intérieur des régions appariées. Nous analysons plusieurs niveaux de supervision et montrons que notre nouvelle supervision épipolaire donne les meilleurs résultats.

La deuxième contribution est également une étude de la supervision pour l'apprentissage profond mais appliquée à un autre scénario : la reconstruction 3D calibrée à partir d'image non contraintes. Nous montrons que les méthodes non supervisées existantes ne fonctionnent pas sur de telles données et nous introduisons une nouvelle technique d'apprentissage qui résout ce problème. Nous comparons ensuite de manière exhaustive l'approche non supervisée et l'approche supervisée avec différentes architectures de réseau et différentes données d'entraînement.

Enfin, notre troisième contribution concerne la représentation des données. Les représentations implicites ont été récemment utilisées pour le rendu d'images. Nous adaptons cette représentation au problème de la reconstruction multi-vues et nous introduisons une nouvelle méthode qui, comme les techniques classiques de reconstruction 3D, optimise la photo-consistance entre les projections de plusieurs images. Notre approche améliore largement les performances de l'état de l'art.

Remerciements

Je remercie tout d'abord Julie Digne, Tamy Boubekeur, Julie Delon et Gabriele Facciolo d'avoir accepté de faire partie de mon jury.

Je tiens à remercier mes directeurs de thèse : Pascal, dont la porte était toujours ouverte pour des questions pointues de géométrie ou d'informatique ; et Mathieu qui a passé quelques nuits blanches sur mes papiers et qui a su me conseiller dans mes choix de recherche et d'après-thèse.

Merci aussi à Bénédicte et Jean-Clément qui ont lancé ce projet de thèse CIFRE et qui m'ont toujours soutenu dans cette collaboration difficile mais fructueuse. Je remercie également tous les membres de l'équipe STI qui m'ont chaleureusement accueilli.

J'ai passé d'excellentes années au sein du laboratoire Imagine. Merci à tous ses membres pour les discussions scientifiques mais aussi pour la bonne ambiance, les foots et autres activités. Je remercie en particulier Thibault, Pierre-Alain et Xi de m'avoir accueilli dans le laboratoire et initié à la recherche.

Merci aussi à tous mes amis qui m'ont soutenu pendant ce long effort grâce, entre autre, à des rendez-vous réguliers qui ont jalonné mes semaines : les Pizza Rossi et les foot à Puteaux. Tom et Bruno méritent un remerciement spécial parmi mes amis pour leur contribution scientifique à cette thèse.

Merci à ma mère, mon frère et mes soeurs d'avoir été là depuis le tout début de mes études et bien avant.

Et enfin merci à Chloé. Tu as été mon meilleur soutien pendant ces quatre ans, je n'aurais pas pu le faire sans toi.

Résumé étendu

Introduction

Le but de cette thèse est d'obtenir une reconstruction 3D à partir d'images en entrée. Cette tâche peut être décomposée en deux : l'étalonnage de caméra qui permet d'obtenir la position et les paramètres internes des caméras qui ont capturé les images et la reconstruction 3D en elle-même à partir des caméras étalonnées.

Ces tâches ont de nombreuses applications pratiques. L'étalonnage de caméra permet de faire de la localisation visuelle pour positionner un appareil à partir des photos qu'il a prises. C'est une alternative intéressante à la technologie GPS car la localisation visuelle ne partage pas les mêmes cas d'échec : elle ne nécessite pas de signal extérieur qui doit être reçu en ligne directe. La reconstruction 3D permet d'avoir des numérisations en 3 dimensions d'un environnement. Cela est très utile pour des applications industrielles pour connaître précisément les dimensions d'un objet ou d'une installation. Cela permet aussi de conserver certains sites historiques, à la fois pour les étudier et pour les visiter virtuellement.

Nous avons choisi dans cette thèse d'essayer de résoudre le problème en utilisant des approches de Deep Learning. Ces approches sont désormais très courantes en vision par ordinateur mais elles ne se sont toujours pas imposées dans le domaine de la reconstruction 3D. Cela peut s'expliquer par deux raisons : premièrement les méthodes de Deep Learning nécessitent une grande quantité de données annotées. Ces annotations sont compliquées à obtenir sur des données 3D car elles ne peuvent pas être obtenues par annotation manuelle. Il faut recourir à des méthodes d'annotations automatiques, soit en générant des données synthétiques, soit en utilisant des appareils de mesure dédiés mais cela ne permet pas de construire de bases de données réalistes à grande échelle. Un deuxième facteur qui explique les difficultés du Deep Learning en reconstruction 3D est le problème de la représentation des données. S'il est facile de représenter des images sous la forme de tableaux de pixels, il n'y a pas de représentation qui combine toutes les bonnes propriétés en 3D : les voxels sont trop gourmands en mémoire, les nuages de points ne permettent pas de modéliser tous les phénomènes, les maillages nécessitent de travailler sur des graphes... Ces deux raisons combinées font que l'application du deep learning à la reconstruction 3D est un problème compliqué.

Nous proposons trois contributions dans cette thèse pour résoudre ces problèmes. La première contribution concerne le développement de méthodes faiblement supervisées et l'analyse des supervisions possibles pour le sous-problème de l'appariement de points

d'interêts. Notre seconde contribution s'intéresse au problème suivant dans les chaînes de reconstructions classiques : la reconstruction multi-vue calibrée. Nous analysons dans ce contexte différentes façons de superviser des réseaux et nous développons une nouvelle méthode non supervisée. Enfin, notre troisième contribution s'intéresse au problème de la représentation des données 3D. Nous adaptons les méthodes implicites neuronales au problème de la reconstruction 3D en apportant des termes de reconstruction multi-vue à l'optimisation.

Deep Learning pour l'appariement de points d'interêts

Dans ce chapitre, nous nous intéressons à la tâche de mise en correspondance d'images : trouver des correspondances précises et robustes de points clés entre les images. Nous proposons une nouvelle méthode de mise en correspondances qui fonctionne en deux étapes. D'abord, un réseau de neurones profond prédit des correspondances à faible résolution puis ces correspondances sont améliorées grâce à des détecteurs et descripteurs traditionnels.

Nous analysons les supervisions qui peuvent être dérivées de la Structure-from-motion à grande échelle pour le réseau de neurone de la première étape. Nous introduisons et étudions trois formes de supervisions : (i) supervision faible au niveau de paires d'images, (ii) supervision épipolaire qui utilise les contraintes épipolaires pour avoir une contrainte supplémentaire sur les correspondances possibles et (iii) supervision par points 3D qui permettent d'obtenir des correspondances vérité terrain. Nous montrons que la supervision à partir de la géométrie épipolaire conduit à de meilleurs résultats que la supervision plus forte mais plus biaisée au niveau des points et constitue une nette amélioration par rapport à une supervision faible au niveau des images.

Nous démontrons ensuite les avantages de notre approche en utilisant les correspondances dans diverses conditions et pour différentes applications. Les correspondances obtenues par notre méthode sont utilisées en entrée d'algorithme d'estimation de géométrie et d'étalonnage de caméra. Nous évaluons la localisation d'images Internet sur le jeu de données YFCC100M et d'images d'intérieur sur le jeu de données SUN3D, la localisation robuste jour-nuit sur Aachen Daynight et la reconstruction 3D dans des conditions difficiles en utilisant les données d'images historiques LTLL.

Stéréo multi-vue par Deep Learning en conditions réelles

Nous nous penchons ensuite sur l'étape suivante dans les chaînes de traitement 3D multi-vue : la reconstruction à partir d'images étalonnées ou stéréo multi-vues (MVS). Ce problème est ici encore envisagé sous l'angle des données d'apprentissage et de la supervision.

Les méthodes Deep MVS ont été développées et largement comparées sur des jeux de données simples où elles surpassent désormais les approches classiques. Dans ce chapitre, nous nous demandons si les conclusions tirées dans des scénarios contrôlés sont toujours valables lorsqu'on travaille avec des collections de photos collectées directement sur Internet sans contrôle sur leur qualité. Nous proposons une méthodologie d'évaluation et explorons l'influence de trois aspects des méthodes deep MVS : l'architecture du réseau, les données d'entraînement et la supervision. Nous faisons plusieurs observations clés, que nous validons de manière extensive quantitativement et qualitativement, tant pour la prédiction de la profondeur que pour les reconstructions 3D complètes.

Premièrement, l'apprentissage supervisé ne fonctionne pas sur des données complexes avec les méthodes existantes. Notre nouvelle approche le rend possible grâce à trois éléments clés : l'agrandissement de la carte de profondeur sortie à la résolution initiale, l'agrégation basée sur un softmin et une fonction de coût uniquement basée sur un terme photo-métrique. Deuxièmement, les méthodes supervisées de Deep MVS constituent l'état de l'art pour les scénarios de reconstructions utilisant peu d'images non controlées. Enfin, notre évaluation fournit des résultats très différents des résultats préexistants obtenus sur des jeux de données contrôlées. Cela montre que l'évaluation dans des scénarios non contrôlés est importante pour le développement de nouvelles architectures.

Reconstruction 3D par fonctions implicites neuronales

Dans ce chapitre, nous nous attaquons finalement au défi de la représentation des données 3D pour le Deep Learning. Les représentations implicites neuronales ont été récemment introduites pour des données 3D. Ces méthodes possèdent de nombreuses propriétés théoriques qui la rendraient supérieure aux représentations traditionnelles. Ce sont des approches volumiques qui permettent de modéliser les notions d'intérieurs et d'extérieurs et surtout elles permettent d'encoder simultanément la géométrie et ses normales dans un seul réseau de neurones. Cette propriété est très désirable en comparaison avec les

cartes de profondeurs généralement utilisées car ces représentations doivent encoder de multiples cartes de profondeurs ainsi que les cartes de normales correspondantes.

Cependant, l'utilisation de méthodes implicites neuronales produit toujours des reconstructions de faible précision comparées aux méthodes classiques. Cela provient de la difficulté d'apprendre et de faire des rendus de textures à haute fréquence avec des réseaux de neurones. En effet, ceux-ci ont un pouvoir expressif limité qui dépend de leur taille. Nous proposons donc de ne pas essayer d'apprendre les textures des images mais d'utiliser les textures provenant d'autres vues. Nous proposons d'ajouter à l'optimisation standard du rendu neuronal un terme de comparaison photo-métrique entre différentes vues. Intuitivement, nous optimisons la géométrie implicite de manière à ce qu'elle déforme les vues les unes sur les autres de manière cohérente.

Nous démontrons que deux éléments sont essentiels au succès d'une telle approche : d'abord l'utilisation de patchs entiers au lieu de pixels puis la gestion des problèmes de visibilité. La déformation de patches entiers peut se faire en utilisant la géométrie et les normales encodées implicitement le long de chaque rayon. Cela permet de comparer les patches avec une fonction photo-métrique robuste (SSIM) et d'être ainsi robustes aux changement de couleur et d'illumination. La gestion des problèmes de visibilité se fait en détectant les points en-dehors de bornes de l'image et en détectant les points qui sont occludés par une partie de l'objet à modéliser.

Nous évaluons notre approche sur les benchmarks standards DTU et EPFL. Ces jeux de données proposent des scènes avec de multiples catégories d'objet et d'aspect de surfaces ainsi qu'un scan vérité terrain de leur géométrie. Nous montrons que notre méthode surpasse les méthodes existantes par fonctions implicites neuronales de plus de 15% sur les deux jeux de données.

Conclusion

Nous nous sommes intéressés dans cette thèse au problème de la reconstruction 3D en utilisant des méthodes de Deep Learning. Nous avons identifié et étudié deux problèmes en particulier : le problème de l'obtention et de l'utilisation des données d'apprentissage et le problème de la représentation des données 3D. Nous proposons des solutions à ces problèmes pour des scénarios d'appariement de points d'intérêts et de reconstruction calibrés. Premièrement, nous avons introduit des méthodes faiblement supervisées qui permettent d'entraîner des réseaux de neurones sans avoir à obtenir des annotations pour les données d'apprentissage. Deuxièmement, nous avons proposé une adaptation des

méthodes de reconstruction par fonctions implicites neuronales dans le but d'améliorer la précision des reconstructions obtenues.

Deux pistes de travail sont envisagées à la suite de cette étude. Premièrement, il est désormais possible d'entraîner des modèles sur des bases de données très grandes et très diverses en mélangeant différentes sources de données et en utilisant une supervision faible. Deuxièmement, la piste de recherche des représentations implicites promet de nombreuses améliorations à l'état de l'art en reconstruction 3D. Il pourrait être intéressant de chercher à développer des méthodes robustes à tous types de données, en particulier des données non contrôlées collectées sur internet. Il serait aussi possible d'améliorer ces méthodes en développant une méthode d'extraction de maillages spécifiques aux fonctions implicites neuronales en remplaçant la méthode générique de Marching Cube.

Table of contents

1 Introduction **1**

1.1 Objectives of this thesis . . . 1
1.2 Motivation . . . 2
1.3 Approach . . . 4
1.4 Challenges . . . 4
1.5 Contributions . . . 6
1.6 Thesis outline . . . 7
1.7 Publications . . . 8

2 Background **9**

2.1 Preliminary: Deep learning . . . 9
2.2 3D geometry . . . 10
2.2.1 Image formation model . . . 11
2.2.1.1 Point projection . . . 11
2.2.1.2 Coordinate systems . . . 11
2.2.2 Epipolar geometry . . . 13
2.2.3 Algorithms . . . 13
2.2.3.1 Triangulation . . . 14
2.2.3.2 Two-view geometry estimation . . . 14
2.2.3.3 Camera calibration . . . 14
2.2.3.4 Bundle adjustment . . . 14
2.2.3.5 Structure from motion . . . 15
2.3 3D datasets . . . 17
2.4 Image matching . . . 19
2.4.1 Classical local feature . . . 19
2.4.2 DL based local features . . . 21
2.4.3 Feature matching . . . 22
2.4.4 Match filtering . . . 23

2.4.5 Dense matching . . . 23
2.4.5.1 Local methods . . . 24
2.4.5.2 Global methods . . . 25
2.4.5.3 Rectified setup . . . 25
2.5 Multiview 3D reconstruction from calibrated camera . . . 26
2.5.1 Point-cloud reconstruction . . . 26
2.5.1.1 Direct point cloud reconstruction . . . 27
2.5.1.2 Depth map based reconstruction . . . 28
2.5.2 Volumetric reconstruction . . . 28
2.5.2.1 Voxel based . . . 29
2.5.2.2 Mesh based . . . 29
2.5.2.3 Deep Learning based . . . 30

3 Deep learning for guiding keypoint matching 33
3.1 Introduction . . . 35
3.2 Related Work . . . 36
3.3 Guided Feature Matching . . . 37
3.4 Learning coarse correspondences . . . 38
3.4.1 Architecture . . . 38
3.4.2 Supervision . . . 39
3.4.2.1 Weak image level supervision . . . 39
3.4.2.2 Weak epipolar supervision . . . 40
3.4.2.3 Point supervision . . . 40
3.4.3 Implementation and training details . . . 41
3.5 Experiments . . . 42
3.5.1 Coarse matching . . . 42
3.5.2 Comparison with guided matching and correspondence pruning . . . 43
3.5.3 Validation on learned keypoint detectors and descriptors . . . 46
3.5.4 Application to challenging 3D reconstruction . . . 47
3.5.5 Limitations . . . 48
3.6 Conclusion . . . 48

4 Deep Learning based Multi-View Stereo in the wild 51
4.1 Introduction . . . 53
4.2 Related Work . . . 54
4.3 MVS Networks in the Wild . . . 55
4.3.1 Network Architectures . . . 55

4.3.2 Training . . . 58
4.3.2.1 Supervised Training. . . . 58
4.3.2.2 Unsupervised Training . . . 58
4.4 Benchmark of Deep MVS Networks in the Wild . . . 59
4.4.1 Data . . . 59
4.4.2 Metrics . . . 61
4.4.3 Common Reconstruction Framework . . . 61
4.5 Experiments . . . 62
4.5.1 Unsupervised Approach Analysis . . . 62
4.5.2 Depth Map Prediction . . . 63
4.5.3 3D Reconstruction . . . 65
4.6 Conclusion . . . 68

5 Multi-view reconstruction with implicit surfaces and patch warping 69
5.1 Introduction . . . 71
5.2 Related Work . . . 72
5.3 Method . . . 73
5.3.1 Volumetric rendering of radiance field . . . 73
5.3.2 Warping images with implicit geometry . . . 75
5.3.3 Optimizing geometry from warped patches . . . 77
5.3.4 Optimization details . . . 79
5.4 Experiments . . . 81
5.5 Conclusion . . . 85

6 Conclusion 87
6.1 Summary of contributions . . . 87
6.2 Future works . . . 88

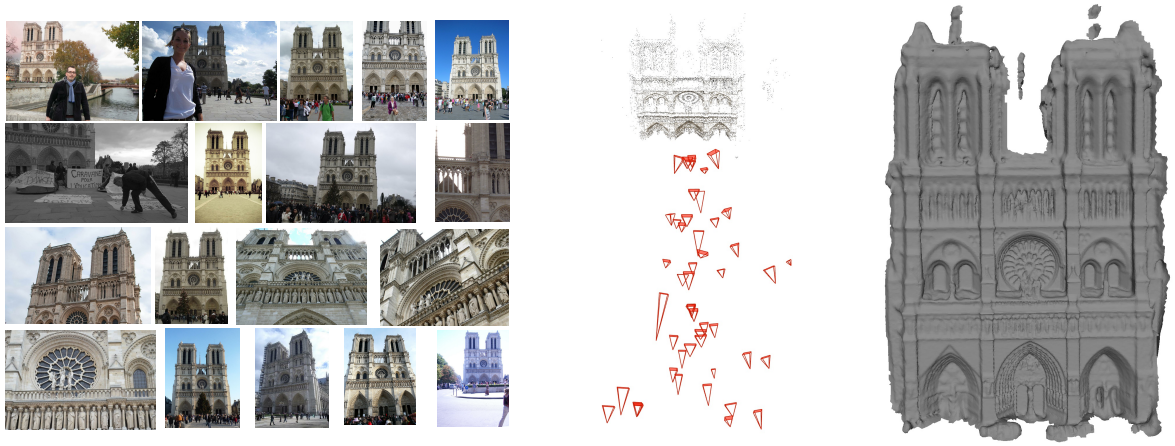
References 89

Chapter 1

Introduction

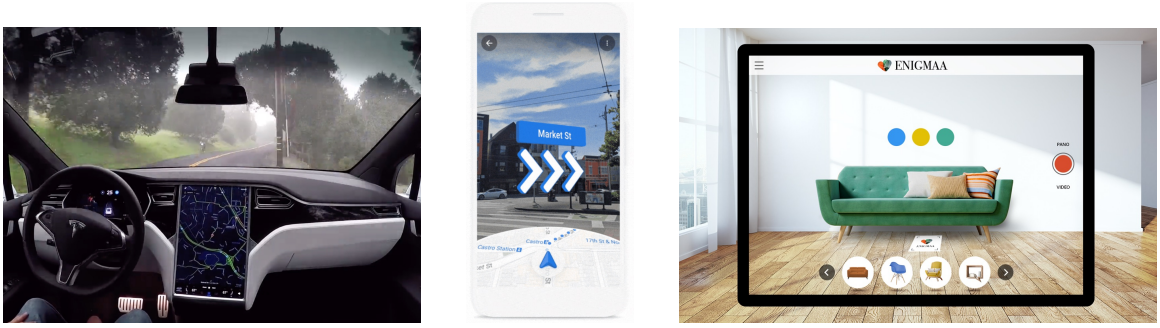
1.1 Objectives of this thesis

The goal of this thesis is to develop algorithms for multi-view 3D reconstruction. Starting from a set of images (Figure 1.1a) the objective is to recover both the positions of the cameras that took the pictures (Figure 1.1b), and the 3D structure of the scene (Figure 1.1c). Taking a picture of a scene consists in projecting a 3D object onto a 2D plane, hereby losing the 3D information. In this thesis, we combine the information from multiple views to recover the true shape of the object.

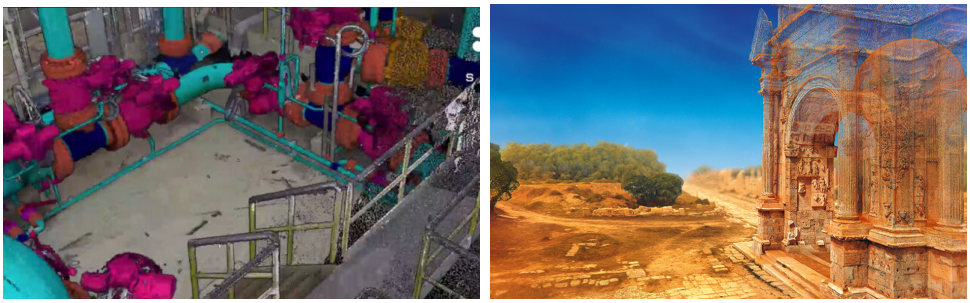


(a) Raw image collection **(b)** Camera calibration **(c)** 3D reconstruction

Fig. 1.1 In this thesis we try to recover 3D information from image data (a)Thomee et al. (2016) : 3D localization of the cameras (b) Schonberger and Frahm (2016) and 3D reconstruction of the object in the images (c) Schönberger et al. (2016).



(a) Autonomous driving[1] **(b)** Navigation[2] **(c)** Augmented Reality[3]



(d) Industrial site scanning[4] **(e)** Digitalization of historical sites[5]

Fig. 1.2 Applications of this thesis: visual localization and multi-view 3D reconstruction.

1.2 Motivation

3D computer vision has multiple practical applications. We distinguish two categories of applications depending on the desired output: either camera localization or shape reconstruction.

Camera localization Camera localization is the process that localizes precisely a camera pose given a picture taken by this camera. The output is the 3D position of the camera center but also the direction the camera looked at. This is used to know where a device is located, which is primordial in many contexts. Some methods like GPS are already present in the industry or everyday life but there are still limitations to these methods: we briefly present a few of them.

- Global Positioning System (GPS) relies on time of flight of signals sent by known satellites to compute a 3D position. The precision can go up to a meter but

[1] https://www.tesla.com/autopilot
[2] https://blog.google/products/maps/new-sense-direction-live-view/
[3] https://helpx.adobe.com/fr/aero/using/help-users-visualize-your-furniture-in-their-homes.html
[4] https://samp.ai/a-quand-le-http-de-l-industrie/
[5] https://www.imarabe.org/fr/expositions/cites-millenaires

it requires a perfect reception of the signal with no reflections. This is often not possible in indoor environment or urban areas. Another limitation of GPS technology is that it can be jammed either by solar flare or by malicious acts.

- Compasses predict an orientation with respect to the magnetic field of the earth, but their precision is very low when used near electronic devices that may produce small magnetic fields.

- Inertial Measurment Units (IMU) use mechanical sensors to measure the value of angular velocity and acceleration at a given time frequency. Knowing the initial position, the current position is computed by integrating those values. The computed positions are known to drift over time because any error in the measurements scales quadratically with the elapsed time. This is the reason why sensors must be very precise, thus heavy, and expensive.

Autonomous driving (Figure 1.2a) requires to know the car location in order to navigate. This is also true for autonomous drones that need to navigate back to their starting points. For such safety critical applications it is necessary to have a substitute to GPS when it is fails. Localization is also used everyday with smartphone based navigation. We show in Figure 1.2b an example of augmented reality for navigation using camera localization in Google Live View. This simple application requires a high precision, especially in the device orientation. However compass based orientation on a smartphone is not precise and requires frequent calibrations. Besides, the most interesting use would be for localization inside an urban environment which is a known failure case of GPS based localization due to multiple path of the signal. Visual localization is therefore a desirable solution for those applications since it relies on pictures only and no exterior signal. Besides, it only requires a cheap sensor: a camera, readily available on smartphones.

Object reconstruction Object reconstruction recovers the shape of the object seen in the pictures or even the whole scene. It is used for example in Augmented Reality (AR) applications. Figure 1.2c shows an application where a user can see what would a furniture look like in a room. This application requires knowledge of the layout of the room to correctly position the furniture on the ground at the correct scale. Object reconstruction is also used for scanning whole industrial scenes (Figure 1.2d). This allows to have a full digitalization of industrial sites, which is helpful for optimization and maintainance: an operator can navigate through a complex site without having to be physically present and therefore shutting down the plant. 3D reconstruction also

has applications in cultural heritage preservation. Figure 1.2e shows an example of a reconstruction made by startup Iconem of Leptis Magna in Lybia. This is a way to keep a digital copy of historical and touristic places when they are located in sensitive areas, politically or environmentally.

1.3 Approach

Image correspondences are at the core of 3D vision: if one knows that regions of two images show the same physical part, it is possible to triangulate the real 3D position of that part. This is depicted in the simple example of Figure 1.3a: each object point and the camera centers form a triangle whose parameters can be computed with trigonometry. All algorithms developed in this thesis involve comparing images to find correspondences.

While it seems easy to find matches for a human operator that has an understanding of the scene structure, it is not straightforward to design an algorithm that does it. Existing methods rely on handcrafted comparison functions designed to be invariant to classical image transformations encountered between two views. However, not all transformations can be modelled: Figure 1.3b shows four views of the same scene taken at different time of the year. In this extreme example, the image transformations are so large that methods based on hand-crafted comparisons are likely to fail.

Deep Learning has a proven track record at handling large image variation for image classification Dosovitskiy et al. (2021); He et al. (2016), detection Carion et al. (2020); Redmon et al. (2016) or segmentation He et al. (2017). The idea is to learn invariances from large scale dataset instead of modelling all the possibilities. Yet classical algorithm Furukawa and Ponce (2009); Schönberger et al. (2016) are still the best methods for 3D reconstruction. In this thesis, we try to bridge this gap: we investigate the application of Deep Learning to 3D reconstruction. We introduce new Deep Learning methods that we carefully evaluate against the existing methods, both deep learning-based and classical ones.

1.4 Challenges

There are two main challenges in the application of Deep Learning to 3D reconstruction. Similar to many computer vision tasks solved with Deep Learning, the first challenge is data collection: what data to use for training the artificial neural network. The second challenge is specific to 3D data: how to represent the data.

(a) 3D position of objects can be recovered from their observations in two views with triangulation

(b) Challenging example of image matching across multiple seasons Badino et al. (2011); Sattler et al. (2018)

(c) Deep Learning is the current state-of-the-art for image classification Simonyan and Zisserman (2015)

Fig. 1.3 Approach: we use correspondences to triangulate the 3D positions of the objects or the position of the cameras. Finding correspondences can be difficult in extreme cases, requiring advanced data-driven algorithms.

Data and supervision Data collection is one of the most important factors for the quality of any Deep Learning method. They require a training step before their prediction can be used. This step consists in optimizing the network so that it best reproduces the example shown in the training dataset. There are three quality criteria for a training dataset: (i) how large and diverse the dataset is, (ii) how close the training examples are to the desired evaluation data, (iii) how accurate the labels are. However, no 3D datasets perfectly meet the three criteria, especially the third criterion about label quality. Contrary to the usual labels used in detection and classification, 3D labels cannot be manually annotated. There exist several approaches to get them: synthetic data, dedicated sensors, or approximated labels, but each method has its own limitations. We investigate in Chapters 3 and 4 several supervision methods to train Deep Learning with different label requirements.

Representation There exist many ways to represent 3D data: among them, point clouds, meshes or voxels. Each has its pros and cons in terms of applicability and efficiency, but not all 3D representations are easy to use in a Deep Learning framework. In addition, there exist intermediate representations that are not directly 3D, e.g. depth maps or correspondences. The challenge in this thesis is to choose the representation so that Deep Learning can be used efficiently and so that it is consistent with available data. Neural implicit representations Mescheder et al. (2019); Park et al. (2019) is a representation with many desirable properties in term of representation power and memory efficiency.

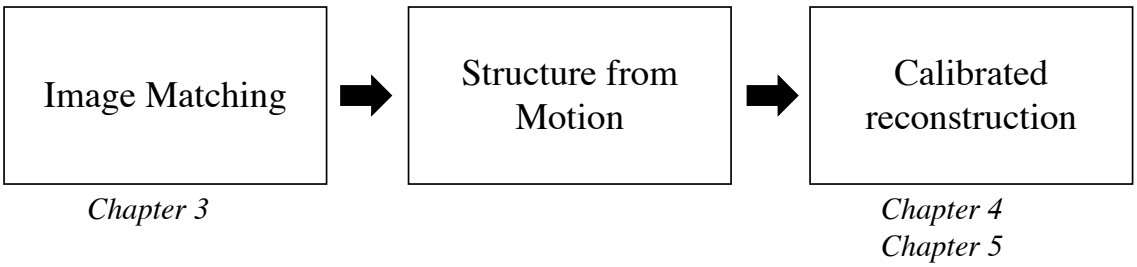


Fig. 1.4 Positioning of our contributions in a typical reconstruction pipeline from raw images.

We introduce in Chapter 5 a new method to do multiview 3D reconstruction using this 3D representation.

1.5 Contributions

We present three contributions to tackle these challenges:

Analysis of data and supervision applied to image matching. This contribution tackles the data and supervision challenge for image matching. We introduce a new method for feature matching: a deep network first matches two images at a low resolution, then those matches are used to guide local feature matching. We compare several levels of supervision for the network: image level, epipolar, and point supervision. We evaluate our sparse matches on downstream tasks like two-view geometry estimation, visual localization, and sparse reconstruction.

Analysis of data and supervision applied to calibrated reconstruction. This chapter also deals with the data challenge but applied to the next task in usual reconstruction pipeline: multiview 3D reconstruction with calibrated cameras. Existing Deep Learning based Multi-View Stereo (MVS) methods Yao et al. (2018), both supervised and unsupervised, are trained and evaluated on controlled environments. We investigate their use on internet collected images which is a more realistic scenario. We show that not all results from the literature hold on this type of data, especially for unsupervised training. We introduce a new unsupervised method and compare several approaches for 3D reconstruction. Our analysis shows that it is possible to do unsupervised training for Deep MVS on internet images and that it can be close to state-of-the-art results.

Multi-view reconstruction with neural implicit representation. This chapter is about the representation challenge for 3D data. Neural Implicit representations Mescheder et al. (2019); Park et al. (2019) were recently introduced for representing 3D data. It has shown impressive results for novel view generation Mildenhall et al. (2020) but its use in multi-view reconstruction is still far from state-of-the-art. We hint that this comes from the difficulty to learn and render high frequency textures with neural networks. In this chapter, we propose to add a direct photometric comparison across different views to solve his issue. It is a way to take advantage of high frequency textures without the need to memorize them in a neural network. We experimentally show that our method improves existing neural implicit methods by a large margin.

1.6 Thesis outline

This thesis is organized as follows.

Chapter 2: Background This chapter reviews the existing work on 3D reconstruction and Deep Learning. The first sections summarizes the background on Deep Learning, geometry and 3D data. We then review the related work on image matching and 3D reconstruction.

Chapter 3: Deep learning for guiding keypoint matching This chapter is about our first contribution tackling the data and supervision challenge for image matching. We first introduce our guided matching technique that uses low resolution matches to guide local feature matching. We then detail how to get those low resolution matches, i.e. the matching network architecture and the three levels of supervisions. We then perform experiments on challenging data like day-night Sattler et al. (2018) and historical data Fernando et al. (2015).

Chapter 4: Deep Learning based Multi-View Stereo in the wild This chapter tackles the data and supervision challenge for calibrated multi-view reconstruction. It is about our second contribution. We first introduce a simple unsupervised method for training deep Multi-View Stereo networks. We next show experimentally that this approach, contrary to existing unsupervised methods, is able to train on uncontrolled data like MegaDepth dataset Li and Snavely (2018). We then perform an analysis of Deep MVS methods along three axes: training data, network architecture and supervision.

Chapter 5: Multi-view reconstruction with implicit surfaces and patch warping In this chapter we introduce our third contribution about data representation. We first introduce a new photo-metric consistency term for optimizing neural implicit surfaces. We then detail the technical contributions required to use this loss: patch warping for using robust comparisons and computation of visibility indicators. We perform experiments showing that our method has state-of-the-art results on DTU dataset Jensen et al. (2014) and EPFL dataset Strecha et al. (2008).

Chapter 6: Conclusion We finally summarize the contributions of the thesis and discuss possible future work directions.

1.7 Publications

Three papers are presented in the manuscript:

- **François Darmon**, Mathieu Aubry, and Pascal Monasse. (2020) Learning to Guide Local Feature Matches. In *International Conference on 3D Vision*.
- **François Darmon**, Bénédice Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. (2021) Deep MVS Gone Wild. In *International Conference on 3D Vision*.
- **François Darmon**, Bénédice Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. (2022) Improving neural implicit surfaces geometry with patch warping. In *To appear in IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

During my PhD, I have also worked on other projects that are not presented in this thesis but that have led to publications:

- Xi Shen, **François Darmon**, Alexei A. Efros, and Mathieu Aubry. (2020) RANSAC-Flow: Generic Two-stage Image Alignment. In *European Conference on Computer Vision*.
- **François Darmon** and Pascal Monasse. (2021) The Polar Epipolar Rectification. In *Image Processing Online*.

Chapter 2

Background

In this chapter, we first describe the general ideas behind Deep Learning. We then explain the geometric foundations required for 3D reconstruction, then the existing 3D datasets. We next review the existing methods for image matching and finally we explore the dense reconstruction litterature. We do not review in detail the works that are most closely related to our contributions in this chapter, but we keep a detailed related work section in each chapter.

2.1 Preliminary: Deep learning

Deep Learning (DL) is a machine learning technique that recently got a lot of attention due to its results for image classification Krizhevsky et al. (2012). It is nowadays used in many computer vision tasks including 3D reconstruction. Most DL methods can be described following three axes: (i) a network architecture, (ii) a loss function and (iii) a training procedure.

Architecture DL networks can be seen as a differentiable parametric function: $\hat{y} = f_\theta(x)$ where x is the input data, $\hat{y}$ is the network output and θ denotes the network parameters or *weights*. Neural networks are often represented using successive layers of elementary operations like convolutions or matrix product. The design choices, i.e. the number, size and type of layers, are referred as the network architecture. We do not review here all possible design choices, e.g. batch normalization Ioffe and Szegedy (2015), skip connection He et al. (2016)... We refer the reader to Goodfellow et al. (2016) for a detailed presentation of the possible choices.

Loss functions The network parameters θ are optimized to minimize a loss function $\mathcal{L}$. It is a scalar function that is designed to quantify how good the network output are. Formally we write it $\mathcal{L}(\hat{y}, X)$ where $\hat{y}$ is the network output and X is an additional data used for computing the loss. There are three types of loss function depending on X: the *supervised losses* use the ground truth label y while the *unsupervised losses* only use the network input x. *Weakly supervised losses* uses additional information but not the exact labels. Such information is generally easier to acquire than the ground truth. We could also define a fourth type of loss: *regularization losses* designed to prevent overfitting. Loss functions in recent Deep Learning methods can be composed of many sublosses, supervised terms, unsupervised terms and regularizations.

Training The goal of the training step is to find the networks weights that minimize the *empirical risk*.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(x,X)\in\mathcal{D}} \mathcal{L}(f_\theta(x), X) \tag{2.1}$$

This is done with first-order optimization methods that work on minibatch since the full dataset $\mathcal{D}$ is generally too big to fit in memory. The most used algorithm are Stochastic Gradient Descent (SGD) and more advanced optimizers like Adam Kingma and Ba (2015) or RMSProp Tieleman et al. (2012) but Deep Learning optimization is a very active research domain that we do not review in this thesis. One of the most important factors that influence the quality of a network prediction is the training dataset $\mathcal{D}$. It must be carefully chosen so that the dataset is large enough for training and so that the examples used for training are similar to the data used in practice. Indeed, when using the network with an input different from the training data, the predictions can be much worse that what was observed on the training data. For example using night-time images when the network was trained on daytime only could give almost random results. This ability of a network trained on one dataset to be robust to new images modality is called the *generalization* of a network. This concept is not restricted to Deep Learning but is present in every Machine Learning model. As of today, Deep learning generalization is not well understood theoretically and it remains an open research problem beyond the scope of this thesis.

2.2 3D geometry

In this section, we review the basics of 3D geometry used in this thesis. We decouple here the notion of 3D geometry and image processing. We only consider the mathematical

operations and the numerical algorithms involved in 3D geometry. The image processing aspects like image matching will be reviewed in other sections of this chapter and we suppose that these problems are already dealt with. We first explain the image formation model, that is to say the mathematical relation between a 3D point and its observation on an image pixel. We then look specifically at the two-view scenarios that can be described with the epipolar geometry Hartley and Zisserman (2004). We finally present the most common algorithms for estimating geometry parameters.

2.2.1 Image formation model

In this section, we detail the image formation model. Given X a 3D point we explain how to compute the pixel position where X projects.

2.2.1.1 Point projection

We assume that the camera is located at the coordinate system origin and that the direction of view, the *principal axis*, is z-axis. The perspective projection of X is written $\pi(X)$ with

$$\pi(X) = (x/z, y/z) \tag{2.2}$$

This operator projects a 3D point onto a 2D plane called the *image plane* of equation $z = 1$. The coordinates on the image plane $\pi(X)$ are still in the world coordinate distance unit, it can be converted to pixel unit using the *internal calibration matrix* K.

$$x = \pi(KX) \tag{2.3}$$

K is an upper triangular matrix. It depends on several internal parameters of the camera such pixel size, pixel shape, focal length... This simple model in called the *pinhole camera model*. It is not realistic in practical situations where the model must take into account geometric distortion. In this thesis, we will only use images with small distortion or images where the distortion was already corrected so the pinhole camera model is valid.

2.2.1.2 Coordinate systems

When using multiple cameras, the coordinate of a 3D point can be expressed in multiple orthonormal coordinate systems: the *World coordinate system* which is fixed for all cameras and the *Camera coordinate system* whose origin is the camera centers and the z axis direction is the principal axis of the cameras (see Figure 2.1). Camera coordinate system is linked to world coordinate system with the *camera pose* or the *extrinsic*

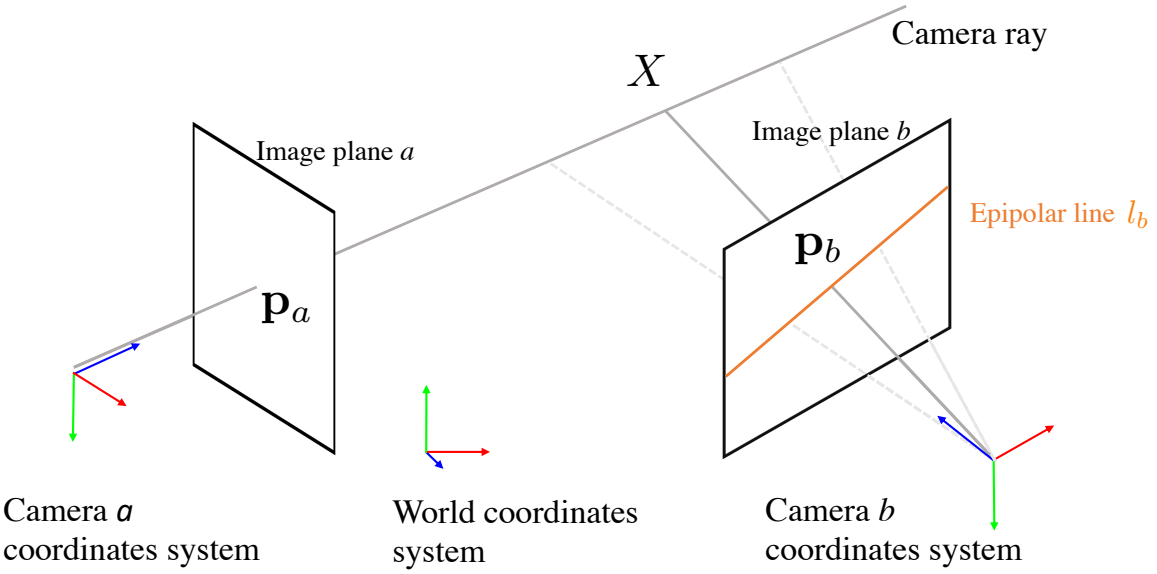


Fig. 2.1 Overview of the geometry notions introduced in Section 2.2. A pixel $\mathbf{p}_a$ in image a corresponds to any point X along the camera ray. This ray projects on camera b on the epipolar line. If $\mathbf{p}_a$ and $\mathbf{p}_b$ match, then the position of the true points X along the ray is computed with triangulation. Inversely, if we know that X projects to a specific pixel p_b, this gives information about the localization of camera b.

calibration $(R, \mathbf{t})$. R is a 3×3 rotation matrix and $\mathbf{t}$ is a 3D translation vector. If X is expressed in world coordinate system, its coordinates in camera coordinate system are

$$\tilde{X} = RX + \mathbf{t} \tag{2.4}$$

In this thesis, we will always use this convention for camera poses and change of coordinates. In summary, the projection x of point X can be computed with

$$x = \pi(PX) \tag{2.5}$$

Here, X is implicitly used with $4D$ homogeneous coordinates. P is a 3×4 matrix called *projection matrix*

$$P = \begin{pmatrix} KR & K\mathbf{t} \end{pmatrix} \tag{2.6}$$

2.2.2 Epipolar geometry

We now explain the two-view scenario. Suppose we have access to two images I_a and I_b with respective poses $R_a, \mathbf{t}_a$ and $R_b, \mathbf{t}_b$. We use $R_{ab} = R_b R_a^T$ and $\mathbf{t}_{ab} = \mathbf{t}_b - R_{ab}\mathbf{t}_a$ the *relative pose* from a to b. This pose transfers points in camera a coordinates to camera b coordinates.

A 3D point seen by the two cameras will correspond to a pixel in image a and another in image b. In this case those pixels form a match or a correspondence. The epipolar geometry establishes constraints on the possible matches between two images. For a given point $\mathbf{p}_a$ in image a it can only match points in image b along the associated *epipolar line* ℓ_b. An example of such line is shown in Figure 2.1. ℓ_b is represented with a $3D$ vector that represents the coefficients of the line equation. It is computed with:

$$\ell_b = F^T \mathbf{p}_a \tag{2.7}$$

with $\mathbf{p}_a$ in homogeneous coordinates. Inversely, the epipolar line associated to $\mathbf{p}_b$ is $\ell_a = F\mathbf{p}_b$. F is called the *Fundamental matrix*, it is formed using the camera calibrations.

$$F = K_b^{-T}[\mathbf{t}_{ab}]_\times R_{ab} K_a^{-1} \tag{2.8}$$

where $[\mathbf{v}]_\times$ is the skew-symetric matrix associated to cross-product with $\mathbf{v}$ such that for all $\mathbf{x}$, $[\mathbf{v}]_\times \mathbf{x} = \mathbf{v} \times \mathbf{x}$. We also use $E = [\mathbf{t}_{ab}]_\times R_{ab}$ called the *Essential matrix*. It defines point-to-line correspondences between points on the respective image planes.

2.2.3 Algorithms

We review here the fundamental algorithms used for computing geometric information like 3D position of points or camera calibration. The intuition behind these algorithms is shown in Figure 2.1. $\mathbf{p}_a$, $\mathbf{p}_b$, X and the camera poses are linked with the projection equations and the epipolar geometry. If some terms are known the other can be recovered by solving the equation. This is the reason why geometric algorithms use matches as input. 2D-to-2D matches fix the coordinates of the pixels $\mathbf{p}_a$ and $\mathbf{p}_b$ so that the true point X can be recovered, 3D-to-2D matches fix X and $\mathbf{p}_a$ providing constraints on the camera poses. We suppose for now that the matches are known. We will review how to get them in Section 2.4.

2.2.3.1 Triangulation

Matches between a calibrated image pair can be used to recover the $3D$ coordinates of the corresponding point. The idea is to find the intersection between the two camera rays that pass through each point. Such intersection does not always exist and several methods exist to find the best intersection point. An easy solution is to use Direct Linear Transform (DLT) method Hartley and Zisserman (2004), it has the advantage of being generalizable to the multiview scenario, but many other method exists

2.2.3.2 Two-view geometry estimation

Matches between two images provide information about the relative transformation between the two images. If the camera intrinsics are known, the matches can be converted to the image plane and the essential matrix is estimated with the 5 points algorithm Nistér (2004). Indeed, the essential matrix has only 5 independent parameters and each match results in a single scalar equation involving them. The essential matrix can be decomposed into 4 different relative poses Hartley and Zisserman (2004) but only one is feasible when looking at the cheirality condition. Note that this relative pose can only be known up to a scale factor since any global scaling of the cameras and 3D points would give the exact same matches.

2.2.3.3 Camera calibration

Camera calibration can also be estimated from matches. Classical algorithms use matches between 3D points to 2D pixels in the image to register. Several algorithms exist depending on the available information about the intrinsic calibration.

- Known intrinsics: PnP problem Lepetit et al. (2009)
- Unknown focal length Bujnak et al. (2008)
- Unknown intrinsics: Camera resection Tsai (1987)

All algorithms rely on an optimization of the camera pose and/or intrinsics so that the 3D points are projected close to their matching 2D pixels.

2.2.3.4 Bundle adjustment

Bundle adjustment Triggs et al. (1999) is an algorithm designed to jointly optimize a noisy 3D reconstruction with noisy camera calibrations. It uses matches from 3D points

to 2D pixels in multiple views and their camera calibrations. It optimizes the coordinates of the 3D points and the camera calibrations so that the *reprojection error* is small. Suppose there are n images and m 3D points $X_1 \dots X_m$. X_j is matched to a pixel x_{ij} in camera i, then the objective function is:

$$\sum_{i=1}^{n} \sum_{j} \|\pi(P_i X_j) - x_{ij}\|^2 \tag{2.9}$$

This non-linear least square optimization is generally solved using Levenberg Marquardt (LM) algorithm. It is a second order optimization method that requires inverting a large Hessian matrix. A commonly used way to decrease the size of the matrix to perform the Schur complement trick Brown (1958). This allows to solve for the camera parameters only, then getting the points solution with back-substitution. Even with this trick, inverting the reduced Hessian is a costly operation. For small number of images, most approaches use a Cholesky matrix factorization. For larger number, Sameer et al. (2010) uses conjugate gradient method instead of explicitly inverting the Hessian. Wu et al. (2011a) further improve this algorithm for multicore hardware architectures.

2.2.3.5 Structure from motion

Structure-from-motion (SfM) is the combination of all the above problems. Starting from a set of uncalibrated images, the goal is to calibrate each of them and to get a point cloud reconstruction of the scene. There are two classes of algorithms: global methods and incremental methods.

Global reconstruction Global reconstruction methods start with the estimation of all the two-view geometries from the input matches. This provides for each image pair (i, j) a relative pose $R_{ij}, \mathbf{t}_{ij}$. The relative poses are related to the absolute poses with:

$$R_{ij} = R_j R_i^T \tag{2.10}$$

$$\mathbf{t}_{ij} \propto \mathbf{t}_j - R_{ij}\mathbf{t}_i \tag{2.11}$$

Global reconstruction methods solve this system of equations with unknowns $R_i, \mathbf{t}_i$. There are several challenges: first the two-view geometries are outlier contaminated since some image pairs do not have any co-visible points and some image pairs are pure rotation, making their estimation with the essential matrix ill-defined. Second, the space of rotation matrices $SO(3)$ is not Euclidean so classical algorithms cannot be used. Third,

the relative translation is only known up to a scale factor so there are additional scaling unknowns λ_{ij}.

It is therefore important to detect and discard bad two-view geometries. Zach et al. (2010) introduce an algorithm that looks at cycles inside the graph to detect invalid rotations. Wilson and Snavely (2014) projects the problem to random directions so that invalid translation directions can be easily found in 1D. Moulon et al. (2013) find the translation directions using a third view instead of using essential matrix decomposition making it more robust to noise and small baseline. Most approaches first solve for the rotations R_i then the translation $\mathbf{t}_i$. Martinec and Pajdla (2007) simply solves a least square problem on the Euclidean matrix space then project the solution on $SO(3)$ while Chatterjee and Govindu (2013) develop an iterative algorithm based on the Lie algebra properties of $SO(3)$. Once the rotation are known, estimating the translations and the scaling parameters λ_{ij} can be done with an ℓ_∞ optimization Kahl (2005).

Algorithm 1 Incremental Structure-from-motion

Input: Matches between image pairs $\mathcal{M}_{ij}, (i,j) = 1 \dots n$
Output: Calibrations $(R_i, \mathbf{t}_i), i = 1 \dots n$, Sparse reconstruction $\mathcal{P}$

Select an initial image pair (i, j)
$R_i, \mathbf{t}_i \leftarrow Id, 0$ ▷ Start with camera i as world coordinate system
$R_j, \mathbf{t}_j \leftarrow$ twoViewGeometry$(\mathcal{M}_{ij})$ ▷ Pose of camera set the relative pose from i to j
$\mathcal{P} \leftarrow$ triangulation$(\mathcal{M}_{ik}, i \in \mathcal{I})$ ▷ Initialize triangulated point cloud
$\mathcal{I} = \{i, j\}$ ▷ Initialize list of processed images
while $|\mathcal{I}| < n$ **do**
    Choose an image k to register
    $R_k, \mathbf{t}_k \leftarrow$ cameraCalibration$(\mathcal{P}, \mathcal{M}_{ik}, i \in \mathcal{I})$ ▷ PnP with the matches between $\mathcal{P}$ and image k
    $\mathcal{P} \leftarrow \mathcal{P} \cup$ triangulation$(\mathcal{M}_{ik}, i \in \mathcal{I})$ ▷ Add newly triangulated points from image k
    $\mathcal{I} = \mathcal{I} \cup \{k\}$
end while
bundleAdjustment$(\mathcal{P}, \mathcal{M}, R_i, \mathbf{t}_i)$

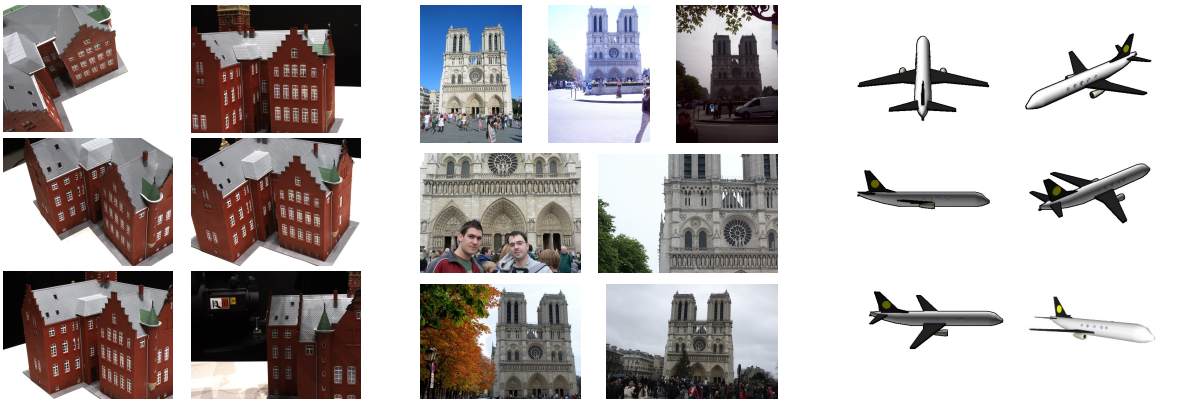
Incremental reconstruction Another approach is to add images one-by-one to the model and keeping track of an increasing number of points. The idea is to start from a two-view reconstruction and fix the poses of the two first cameras. This image pair also provides a few 3D points from the triangulation of its matches. Then the algorithm iteratively adds new images to the reconstruction. It registers new images with PnP using the matches from the already triangulated points and the points from the new image. Those matches are formed by keeping track of the original 2D points used for

triangulating the points so that the input 2D-to-2D matches can be transferred to 3D-to-2D matches. Now that a new image is added to the model, new points can be added with triangulation. The general algorithm is presented in algorithm 1. A functional incremental Structure-from-motion is much more complicated in practice: e.g. bundle adjustment is performed frequently to avoid drifting overtime; a re-triangulation step Wu et al. (2011b) can be used after bundle adjustment to retry triangulation of failed points after the camera poses have been updated. Another very important technical choice is what initial pair to start with Beder and Steffen (2006) and what image to add at each step Schonberger and Frahm (2016). Incremental reconstruction is the method of choice for large scale SfM with well-known software packages: VisualSFM Wu et al. (2011b), Bundler Snavely (2008), OpenMVG Moulon et al. (2016) that also implements global SfM and COLMAP Schonberger and Frahm (2016). We will use COLMAP in all the experiments of this thesis since it is the most used one and it is open-source.

2.3 3D datasets

We review in this section the existing datasets that can be used for 3D vision. Those datasets are used for two main reasons: evaluation and training. The most common way to compare algorithm in computer vision is to run them on a same dataset and compare the results obtained. Another use of data is for training data-driven techniques like Deep Learning networks: data collection is an important part of the design of a DL method, especially for supervised methods that require ground truth labels. For 3D tasks, the label may be 3D reconstructions, poses of cameras, matches, or any 3D related output. Contrary to traditional DL task like image classification, such labels cannot be manually annotated. There are three main ways to gather 3D annotation, each with its pros and cons, (i) acquisition with hardware, (ii) large-scale collection and estimation with an existing algorithm, or (iii) synthesis with a rendering framework. An example of each data type is shown in Figure 2.2.

Acquired data 3D reconstructions can be acquired with laser sensors or structured light device, while 3D poses can be obtained with Inertial Measurment Units (IMU). Such approach can generate high quality ground truth labels, but they need expensive hardware that cannot be used on a large scale. This is the reason why acquired ground truths are mainly used for evaluation on small-scale dataset. Tanks and Temples (T&T) Knapitsch et al. (2017) and ETH3D Schöps et al. (2017) are datasets with acquired ground truth but they cannot be used for training since there are not enough scenes. However KITTI Geiger



(a) Acquired Data Jensen et al. (2014) **(b)** Large scale data Thomee et al. (2016) **(c)** Synthetic data Chang et al. (2015)

Fig. 2.2 Example of possible data used for training deep 3D network

et al. (2013) and DTU Jensen et al. (2014) both consist of a large number of image in many scenes. This allows to train networks on such data but the network may not generalize since the data lacks diversity. DTU is acquired in a fixed laboratory environment while KITTI images are only from a camera on the top of a car.

Large scale data A common way to collect large scale datasets is to use internet images of touristic places on the internet. Such datasets can be composed of hundreds of scenes of thousands of images each. However, only the images are available online and there are no 3D labels. The labels must therefore be estimated with existing 3D software. YFCC dataset Thomee et al. (2016) was run through a SfM pipeline in Heinly et al. (2015). Megadepth Li and Snavely (2018) consists of many scenes from the internet with their sparse reconstruction and depth maps obtained with COLMAP Schonberger and Frahm (2016); Schönberger et al. (2016). Similarly, Image Matching Benchmark Jin et al. (2020) perform evaluation of 3D methods using ground truth labels from the COLMAP estimation. The idea behind such approaches is that reconstruction using thousands of images will be of much better quality than when using few images. Since DL networks usually take few input images they can be trained and evaluated on such datasets. However, labels obtained from existing 3D softwares even with thousands of image are not perfect, inducing noisy training data.

Synthetic data Image rendering techniques can produce images from a given viewpoint and a 3D model. Starting from 3D models, it is thus possible to render images with perfect ground truth for a large variety of objects. Shapenet Chang et al. (2015) is a

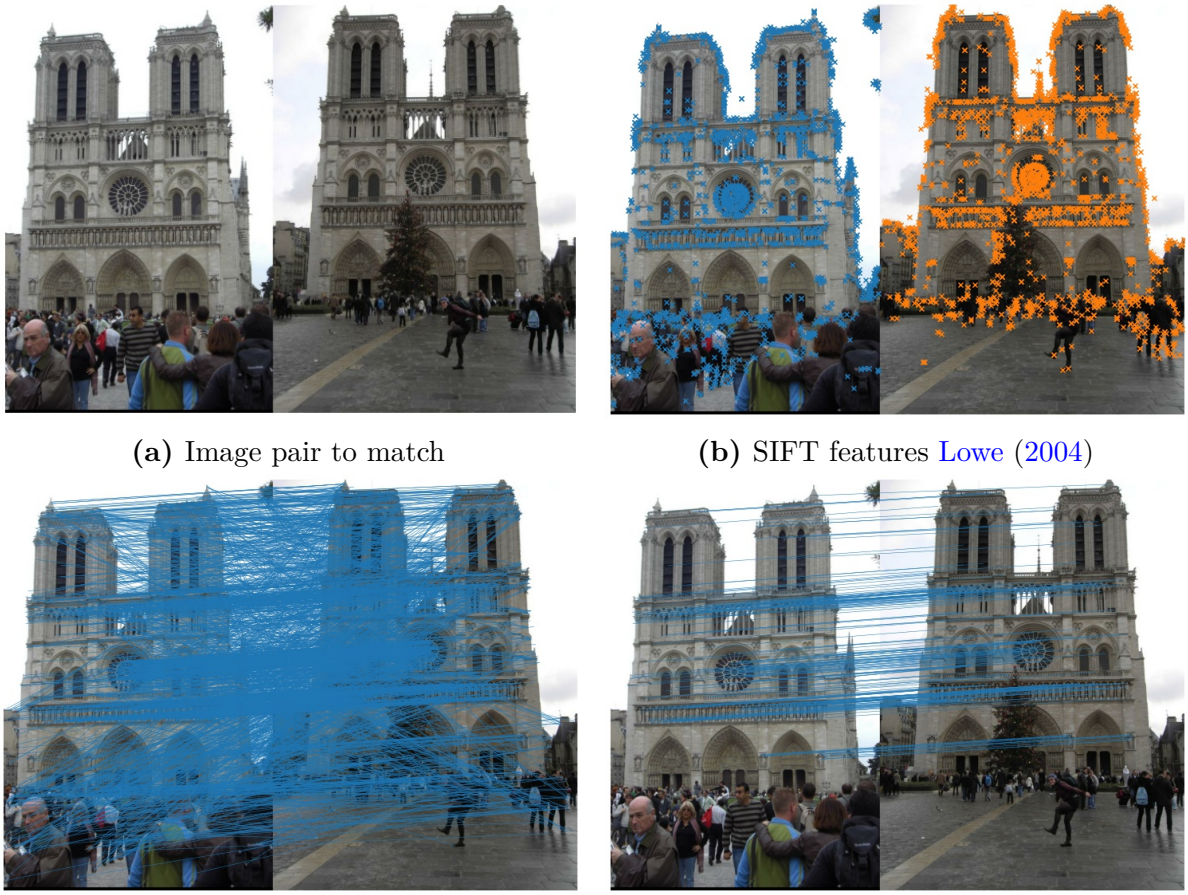
collection of shapes widely used in 3D DL. However, the rendered images are not realistic since the objects are not rendered in a realistic environment and the shapes lack diversity. Networks trained on such data therefore poorly generalize to real data. BlendedMVS Yao et al. (2020a) try to remedy this issue by combining real data with renderings from photogrammetry in order to produce more realistic images with perfect ground truth.

2.4 Image matching

We now review how to get the image matches used in the algorithms of Section 2.2. Since the objective is to get matches for downstream Structure-from-motion or visual localization, it is not required to have a match for every pixel. Most approaches are sparse matching techniques, matching only a set of selected *features* per images. The main advantages is efficiency both for the matching and for the followings steps, especially bundle adjustment. We first review classical methods for extracting local features, then the Deep Learning based ones. We then review the techniques for matching local features and how to detect wrong matches. The different steps for image matching are illustrated in Figure 2.3. Finally, we review the dense matching techniques.

2.4.1 Classical local feature

The first step for sparse image matching is to choose which pixels to match. The goal is to select pixels based on how distinctive they are. For example, pixels from textureless regions must not be chosen since they cannot be precisely retrieved in another image. Another required property is that the detection must be invariant to image transformations: the same points should be detected if we apply a rotation/scaling/color transform to the image. Most detection methods rely on the image gradient with the idea that salient points in the image must have high gradient magnitude. However, not all points with high gradient will be distinctive because some points may be on a line with large gradient but poor distinctiveness. Harris corners Harris et al. (1988) are points with high magnitude gradient in two orthogonals directions by looking at the second moment matrix. Harris-Laplace detector Mikolajczyk and Schmid (2004) makes this approach scale invariant by using multiple scales. Other approaches rely on finding *blobs* in the images. Among them are Laplacian of Gaussian efficiently approximated with Difference of Gaussians (DoG) Lowe (2004). The idea is to detect extrema, both in space and scale, of the Laplacian operator.



(a) Image pair to match

(b) SIFT features Lowe (2004)

(c) Matched features based on nearest neighbor between descriptors

(d) Filtered matches with ratio test and RANSAC Fischler and Bolles (1981)

Fig. 2.3 Illustration of the main steps of image matching

Keypoints will be matched based on the comparisions of the patch around each keypoint. However, simple square patch extraction is not invariant to common image transformation like scaling and rotation. It can be made scale invariant by computing a characteristic scale Lindeberg (1998). Similarly a characteristic orientation can be computed from the principal orientation of the gradient histogram Lowe (2004). Combining both would make the patch extraction invariant to similarity transforms. However, in the case of 3D movements, similarity is not general enough. A better approximation is affine transformation as estimated in Mikolajczyk and Schmid (2004). Most of recent methods use SIFT detector Lowe (2004) which is based on DoG and extraction of characteristic scale and orientation. There exist many more detection approaches, some are compared in Mikolajczyk and Schmid (2005).

Now that keypoints are detected, with their extracted patches, they must be described for later matching. The idea is to use a small vector (typically of size 128) to summarize an image patch. That way, feature will be efficiently matched with nearest neighbor in descriptor space. Descriptors must be robust to color transformations so that images with different lighting and ambient colors can be matched while being discriminative so that the number of false matches is low. There exist many description methods: SIFT Lowe (2004) builds a histogram of the gradient orientation in the patches. SURF Bay et al. (2006) is based on wavelet decomposition of the patches. A key performance factor of a descriptor method is the size of the descriptors in memory and how fast they can be compared. While SIFT is an integer histogram, BRIEF Calonder et al. (2010) is a set of binary pixel value comparison that can be very efficiently stored and compared using bit manipulation.

2.4.2 DL based local features

Training DL networks for local features is possible using ground truth matches. Their are two ways to obtain such annotations: supervised methods use 3D ground truth to compute matches between two images; and self-supervised methods generate synthetic pairs by applying two different transformations to the same image. We first review learned descriptors then review methods used for both extraction and detection.

DL is often used for description by feeding a patch to a CNN that outputs a fixed size vector. The networks are supervised with a variation of triplet loss. Such losses uses triplet of patches, two matching patches and another non-matching patch. The objective is to have the minimum distance between the predictions for the matching patches while having the maximal distance for the non matching patches. One key factor is how to choose the positive and negative pairs Luo et al. (2018); Mishchuk et al. (2017); Mishkin et al. (2018). Contextdesc Luo et al. (2019b) uses additional information to describe the patch, instead of just using the patch, it uses in addition the keypoints distribution in the form 2D PointNet encoder Qi et al. (2017) and global features from the feature map of a CNN looking at the full input image. LogPolarDesc Ebel et al. (2019) also describes keypoints using DL but using a circular region instead of square patches.

Combined detection and description is done with a CNN that looks at the full image and outputs a dense descriptor map and a detection heatmap. See Figure 2.4 for an example of a network architecture. Such networks are harder to train than patch description since the exact definition of keypoint is unclear. Most approaches use a loss that penalizes (i) keypoints detected in one image and not the matching point in another image, (ii) keypoints whose descriptor does not give a valid match. This is not enough

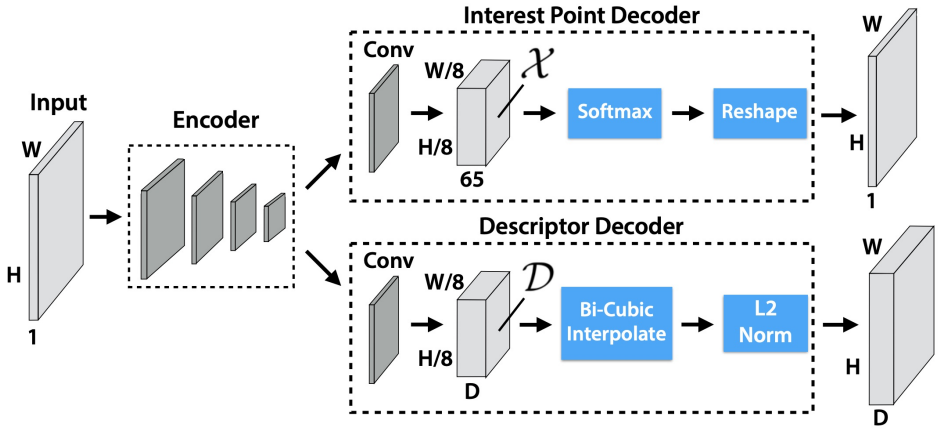


Fig. 2.4 Superpoint DeTone et al. (2018) architecture for joint detection and description of local features.

for training a network since there is an obvious solution that is to detect no keypoints. DeTone et al. (2018); Yi et al. (2016) solve this issue by using a supervised loss to force detection on known keypoints, Revaud et al. (2019) uses a specific loss term to force non-constant heatmaps and Ono et al. (2018) forces the extraction of a fixed number of patches at training time while Tyszkiewicz et al. (2020) uses a reinforcement learning framework that gives positive feedback to new keypoints as long as they lead to a correct match. Finally, D2-net Dusmanu et al. (2019) does not explicitly predict a heatmap, it just computes it from the local maxima of the descriptor heatmap.

2.4.3 Feature matching

Once the features are described, they can be matched with nearest neighbors. This is an expensive step: in an SfM scenario each possible pair must be matched so this step has quadratic complexity in the number of images. A solution to this issue is to do a pre-matching step where a set of candidate images are estimated for each image. This step must be much faster than feature matching. It is usually done with image retrieval techniques Philbin et al. (2007).

For the feature matching step itself, Sarlin et al. (2020) developed a transformer Vaswani et al. (2017) based architecture dubbed Superglue. The idea is that matching each feature individually is not optimal so they use a graph neural network that works on the graph formed by features in both images. This method improves the quality of the matches got with nearest neighbor.

2.4.4 Match filtering

Matches obtained from nearest neighbors are often outlier contaminated. The filtering step is designed to detect invalid matches. It is usually done in two steps: non-parametric filtering or correspondence pruning then geometric filtering. Non parametric filtering uses simple heuristics to remove ambiguous matches. For a given match, the ratio test looks at the distance between the two descriptors as well as the distance to the second best descriptor. If the two distances are too close, it means that there is no way to decide which point is the correct one and the match should be discarded. Another simple method is the bidirectional check: each match should be the nearest neighbor when starting from the point in image a and when starting from image b. More elaborate techniques Bian et al. (2017); Lin et al. (2017); Ma et al. (2019) also remove false matches with the observation that keypoint matches should be consistent with their close neighbors.

Geometric filtering is then used to further remove false matches using a geometric model. The most known method is RANdom SAmple Consensus (RANSAC) Fischler and Bolles (1981). This algorithms picks k matches randomly where k is the number of matches needed to estimate the geometric model ($k = 4$ for homographies, $k = 5$ for essential matrices, etc.). It then estimates the model and uses it to count the total number of matches that are consistent with this model. The algorithm does it iteratively and keeps in the end the model which led to the highest number of inliers. RANSAC has been improved for increased efficiency Chum and Matas (2005) or solving degeneracies in planar structures Chum et al. (2005). Recently, DL has been applied to do the same task as RANSAC. Brachmann and Rother (2019); Moo Yi et al. (2018); Ranftl and Koltun (2018); Zhang et al. (2019b) learn outlier filtering by neural networks. They typically consider the matches as a 4D point cloud (pairs of 2D coordinates) and use a pointcloud segmentation network e.g. Qi et al. (2017) to predict whether a match should be kept. These networks are supervised with epipolar geometry: if the fundamental matrix between both images is available, each match can be assigned a label as inlier or outlier depending on its epipolar distance.

2.4.5 Dense matching

Contrary to feature matching, dense matching is the task that matches each pixel of the input images. This is generally not possible for SfM, but it is starting to become possible with the increase of computing power. Dense matching is linked to the optical flow problem that matches pixels between frames of video. This subject has been much studied with the seminal work of Horn and Schunck (1981); Lucas and Kanade (1981)

and recently DL networks Dosovitskiy et al. (2015); Ilg et al. (2017); Ranjan and Black (2017); Sun et al. (2018); Teed and Deng (2020). Optical flow methods are however not used in 3D reconstruction since the movement between two images is usually much bigger and there can be imaging condition changes. We review here methods adapted to matching in this scenario. There are two classes of methods: local methods look at each pixel individually while global methods solve for the full image at once.

2.4.5.1 Local methods

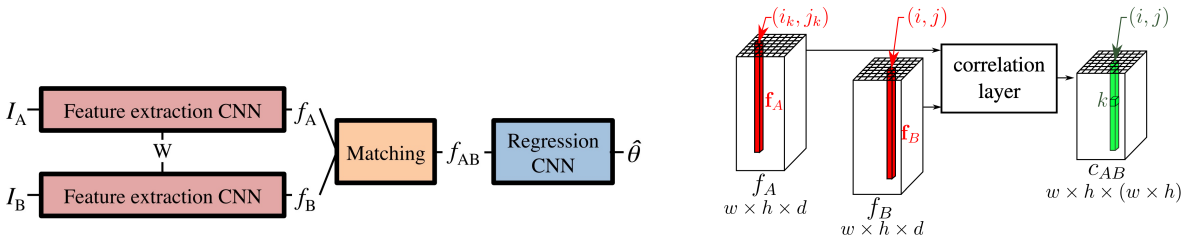
The easiest approach is to directly compare patches exhaustively using a photometric comparison function and to choose the best match according to this metric. Several comparison functions exist, the easiest is the cross-correlation, but it is not invariant to affine color changes. More advanced functions are therefore used, using normalizations based on the patch means μ_a, μ_b and their variance σ_a, σ_b.

- Cross-correlation:
$$\text{cc}(\mathbf{P}_a, \mathbf{P}_b) = \frac{1}{n} \sum_{x,y} \mathbf{P}_a(x,y)\mathbf{P}_b(x,y) \tag{2.12}$$

- Zero-normalized Cross-correlation (ZNCC):
$$\text{zncc}(\mathbf{P}_a, \mathbf{P}_b) = \frac{1}{n} \sum_{x,y} \frac{1}{\sigma_a \sigma_b} (\mathbf{P}_a(x,y) - \mu_a)(\mathbf{P}_b(x,y) - \mu_b) \tag{2.13}$$

- Structural Similarity Index Measure (SSIM) Wang et al. (2004)
$$\text{ssim}(\mathbf{P}_a, \mathbf{P}_b) = \frac{(2\mu_a\mu_b + c_1)(2cc(\mathbf{P}_a, \mathbf{P}_b) + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)} \tag{2.14}$$

Patch descriptor can also be used: Daisy Tola et al. (2009) is a descriptor method that, similar to SIFT, uses gradient orientation histograms but computed efficiently for dense description. DL methods can also be used for direct patch comparison, in Zagoruyko and Komodakis (2015); Zbontar and LeCun (2015), the two image patches are used as input to a classification network that decides whether the two patches match or not. Comparing exhaustively each patch in image a to each patch in image b can be very heavy for large images. PatchMatch Barnes et al. (2009) uses the assumption that two neighbor pixels in a are likely to have also neighbor matches in b. It compares only a few number of hypotheses and propagates the best ones to the neighboring pixels. This is a way to drastically reduce the number of patch comparisons.



(a) Full architecture: shared feature extraction followed by matching layer and final layer

(b) The matching layer explicitly compare all pairwise similarities

Fig. 2.5 Architecture from Rocco et al. (2017) designed for matching image pairs with large variations.

2.4.5.2 Global methods

Global methods do not look individually at each pixel but solve for the full image. SiftFlow Liu et al. (2010) builds an energy model where the data term is based on SIFT Lowe (2004) descriptor distance and the regularization term favors the displacement to be small and spacially consistent. This is solved using Loopy Belief algorithm in a coarse-to-fine approach. Deep Learning methods taking both images as input and that outputs matches can also be seen as global methods. Simple CNN architectures cannot be used for this problem: stacking both images as a 6-channel image and use a CNN as in some optical flow papers does not work since the displacement can be much bigger than the CNN receptive field. Rocco et al. (2017) add an explicit feature similarity layer where each feature of image a is matched to every feature in image b (Figure 2.5). This is further improved with $4D$ convolutions Rocco et al. (2018). Another approach is to use a multi-step approach where a coarse matching is first estimated, then traditional CNN architectures are used on images warped with the coarse matching Shen et al. (2020); Truong et al. (2020). Finally LoFTR Sun et al. (2021) extend Superglue Sarlin et al. (2020) to match dense CNN features instead of interest points.

2.4.5.3 Rectified setup

An interesting scenario is the calibrated case. When the calibration is known, one can compute the epipolar geometry and do image rectification so that the epipolar lines are all located on horizontal lines Darmon and Monasse (2021); Loop and Zhang (1999); Pollefeys et al. (1999). The correspondence search is now a simpler 1D search along the horizontal line. This topic has been much studied with classical methods Scharstein and Szeliski (2002) both local and global as well as DL methods. A more diverse DL architecture choice can be made without being memory limited since the problem is only

1D. The first architecture Kendall et al. (2017) used explicit feature comparison into a 3D volume processed with 3D convolution. Later methods Chang and Chen (2018); Xu and Zhang (2020); Zhang et al. (2019a) improved its efficiency. Recently, transformers were adapted for stereo matching tasks Li et al. (2021).

2.5 Multiview 3D reconstruction from calibrated camera

3D reconstruction is the task that tries to recover the 3D structure of a scene from its projection on 2D images. In this thesis, we focus on the calibrated case where the camera calibrations are known e.g. estimated with SfM. The general idea behind every reconstruction method is to find the 3D position of points that leads to the best alignment of the input images after reprojction. Formally, let $I_1 \dots I_N$ be the input image, $I_i[\mathbf{p}]$ denotes the color value of image i at pixel $\mathbf{p}$. Let $\mathcal{M}_{\mathcal{S}}^{ij}(\mathbf{p})$ be the match in view j of $\mathbf{p}$ in i that can be derived from the reprojection based on surface $\mathcal{S}$. Similarly $V_{\mathcal{S}}^{ij}(\mathbf{p})$ is the visibility indicator of this match. The goal of MVS method is to optimize the following energy with respect to the surface $\mathcal{S}$:

$$\sum_{i \neq j} \sum_{\mathbf{p} \in I_i} V_{\mathcal{S}}^{ij}(\mathbf{p}) \phi(I_i[\mathbf{p}], I_j[\mathcal{M}_{\mathcal{S}}^{ij}(\mathbf{p})]) \tag{2.15}$$

where ϕ is a photometric comparison function. MVS methods differ in the visibility computation, the choice of photometric function, the algorithm chosen to solve the optimization and the way the surface is parameterized Seitz et al. (2006). Figure 2.6 shows a graphical overview of this section: we first review point-cloud based reconstruction methods then volumetric ones. See Furukawa and Hernández (2015); Seitz et al. (2006) for a more detailed overview of the classical reconstruction methods.

2.5.1 Point-cloud reconstruction

Point cloud is a 3D representation that represents 3D as a list of coordinates of 3D points, possibly with their surface normal. It is an intuitive data structure since it naturally comes from image pixels. The perfect reconstruction, although highly redundant, would be a point cloud where each point corresponds to a pixel in one of the input image at the correct depth. We distinguish here two approaches to do point cloud reconstruction. First direct reconstruction that uses no intermediate 3D parameterization then depth

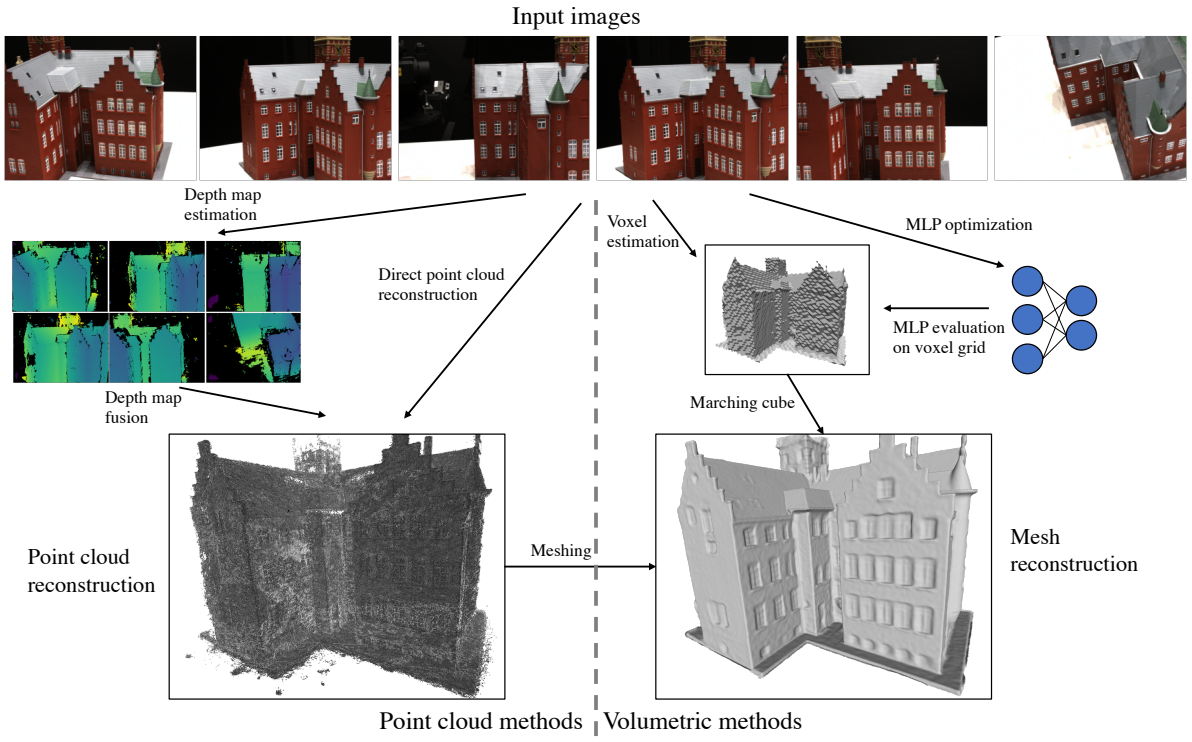


Fig. 2.6 Overview of different approaches for dense 3D reconstruction. There exist several representations each used by different methods as final output or as intermediate representation. Note that the final output may vary: each framed representation may be considered as final output depending on the way the reconstruction will be used.

map reconstruction that computes an intermediate representation: depth maps for each input view.

2.5.1.1 Direct point cloud reconstruction

Direct reconstruction works on a point cloud reconstruction directly. Sparse reconstruction from SfM algorithms of section 2.2.3.5 are such techniques, but they work on uncalibrated cameras. If the calibrations are known, the geometry can be better estimated using specific methods. PMVS Furukawa and Ponce (2009) starts with a sparse reconstruction then it tries to fit a local homography to each keypoint matches so that it aligns patches around keypoints in multiple images. This provides a list of 3D points with their normals and visibility indicators, i.e. the list of images that see the point. Points are then iteratively added to the reconstruction by propagating the depth, normal and visibility to points that projects close to an existing point projection until all the pixels have been

tried. This method provides good reconstructions results even with challenging data, but its iterative nature makes it hard to parallelize for GPU or modern CPU architectures.

2.5.1.2 Depth map based reconstruction

A very successful approach is to decouple the point-cloud reconstruction into smaller subproblems of depth maps estimation, then fusing the depth maps into a point cloud. The goal is now to estimate depth maps for an input image called *reference image* given other views called *source images*. Similar to image matching discussed in Section 2.4.5, local methods set the depth for each pixel individually. It consists in choosing the depth that leads to the lowest photometric distance between the reference image and the reprojected source images. These method are often referred to as plane-sweep methods Gallup et al. (2007). GIPUMA Galliani et al. (2015) is such a method focused on computational efficiency. It uses a patch-match formulation on the GPU. Global methods computes a full depthmap based on an energy minimization. This can be done with a Markov Random Field (MRF) Campbell et al. (2008). In order to work with unconstrained data collection, Zheng et al. (2014) improve the probabilistic model by adding a pixelwise visibility indicator. The goal is not only to predict depth for each pixel but also a binary indicator of other views that see this pixel. This model is optimized with a Patch-Match formulation. This algorithm is further improved with COLMAP Schönberger et al. (2016) that adds a normal estimation that allows for accurate patch comparison on slanted surfaces.

Estimation of depth maps can also be done with Deep Learning. MVSNet Yao et al. (2018) introduces a DL architecture that predicts a depth map given a reference image and a set of source images. It uses a correlation volume layer that aggregates features from the reference view and the source view reprojected from every tested depth. This supervised architecture produces state-of-the-art results on controlled datasets. Another approach Sinha et al. (2020) uses differentiable sparse matching to triangulate points and produce a sparse depth map that can be completed by a simple CNN that looks at the reference image and the sparse depth map.

2.5.2 Volumetric reconstruction

Point-cloud reconstruction is often not enough for practical use of the reconstruction like rendering. A volumetric rendering is more useful. There are many volumetric parameterizations, the only requirement is to be able to say whether a point in space is empty or occupied. This requires to estimate a continuous function of space. This

can only be done using a discretization of the function space. We distinguish here three ways to do it, i.e. three different parametrizations: voxel based, mesh based and Deep Learning based.

2.5.2.1 Voxel based

Voxels are a generalization to 3D of pixels. It is a regular grid of a fixed region of 3D space. It is one of the first studied parametrization since it is easy to process. The early work of Seitz and Dyer (1999) iteratively gives a color to each voxel by looking at the color consistency of its reprojection in the input image. Space carving methods Kutulakos and Seitz (2000) uses silhouette of objects as input and "carve out" each voxel that reprojects outside a silhouette. This approach cannot recover fine details but only an englobing shape, the visual hull. Faugeras and Keriven (1998); Pons et al. (2007) introduce the level set framework for surface reconstruction. The surface is encoded as the zero-level of a scalar field, generally implemented as the interpolation of the values on a regular grid. This surface can be optimized with a variational formulation for photo-consistency of the projections on the input images. Deep learning methods can be used to predict values on a voxel grid. 3D-R2N2 Choy et al. (2016) uses a recurrent network that is fed multiple views of a same object and outputs a voxel occupancy grid. Surfacenet Ji et al. (2017) convert each input image to a colored voxel grid then fuses each voxel grid in a unique occupancy grid with a 3D CNN. Finally, Murez et al. (2020) uses a similar encoding to predict an SDF grid of indoor scenes. However, the memory requirements of voxel methods grow cubically with the resolution making these method often impractical for real world scenes.

2.5.2.2 Mesh based

Meshes are an extension of point clouds with additional connectivity between points. We only consider triangular meshes where points are only linked by triangles. Intuitively, a watertight mesh is a mesh without any holes. From a watertight mesh, one can describe any point in space as inside the mesh or outside. Meshes are one of the most often used data structures in Computer Graphics and are therefore a desirable output for reconstruction algorithms. Meshes can be computed from point clouds with meshing algorithms. Vu et al. (2011) starts from a Delaunay triangulation, then solve a binary classification problem to decide the inside/outside labelling. Screened Poisson Reconstruction (sPSR) Kazhdan and Hoppe (2013) is one of the most used methods nowadays, starting from a point cloud with its normal, it optimizes a scalar field to vanish on the points and to have a gradient close to the normals. Meshes can also be

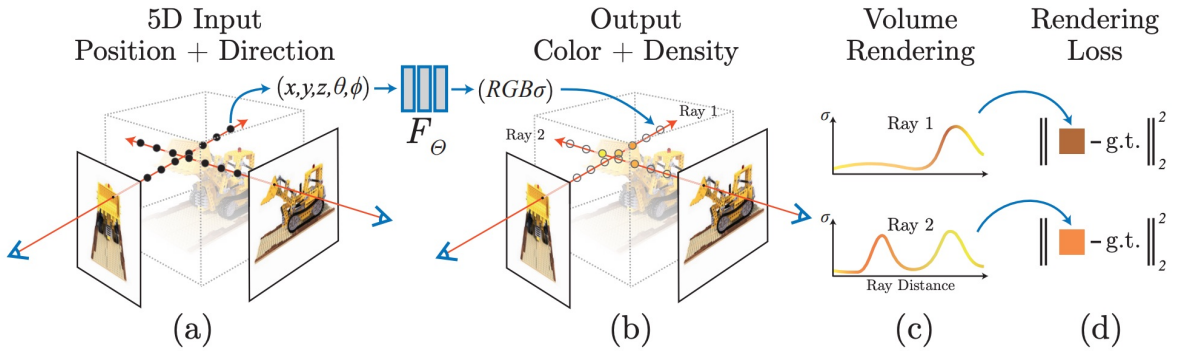


Fig. 2.7 Neural Radiance Field (NeRF): Point sampling along the camera ray (a) with their associated color and density (b) that are aggregated with volumetric rendering (c). The networks are optimized with a rendering loss (d).

converted from voxel parameterization with Marching Cubes algorithm Lorensen and Cline (1987). Once a watertight mesh is estimated, the reconstruction can be further improved by optimizing the vertices position for photo-consistency of the continuous reprojections computed from the mesh Furukawa and Ponce (2009); Vu et al. (2011).

2.5.2.3 Deep Learning based

A successful way to have function discretization with good fidelity while keeping a low memory consumption is to use a Neural Network to encode geometry. We refer to this approach as *Neural Implicit Surfaces*. Similar to the classical level-set framework Osher and Sethian (1988) where the surface is represented as the zero level of a function, Implicit Neural Surfaces methods represent the surface as the zero level of a neural network. The idea is to optimize a Multi-Layer Perceptron (MLP) to reproduce the Signed Distance Function (SDF) Park et al. (2019) or occupancy indicator Mescheder et al. (2019) of a shape. It is different from existing DL techniques, the network is not used for predicting parameters of the representation but as the representation itself. This representation is still a discretization of a continuous field but the discretization is hidden in the MLP weights. It can reproduce shapes at a much higher resolution, with low memory consumption compared to voxels. However, this comes at a cost with a larger computation time since a neural network must be evaluated instead of accessing a value in memory. This parameterization was recently used in a multi-view setup. The parameterization network is optimized to produce the best renderings of the input images. DVR Niemeyer et al. (2020) explicitly computes the intersection of camera rays with the surface and derives a gradient for this operation, but this leads to unstable trainings

that require silhouette inputs. NeRF Mildenhall et al. (2020) (Figure 2.7) solves this issue by using volumetric rendering Max (1995). Although primarily intended for highly realistic renderings the volumetric rendering can be used for 3D reconstruction Oechsle et al. (2021); Yariv et al. (2021).

Chapter 3

Deep learning for guiding keypoint matching

(a) The precise feature match (green) is disambiguated relative to concurrent ones (red) by the coarse matching.

(b) SIFT features matched with our method.

Fig. 3.1 In challenging conditions, local information might not be enough to disambiguate local feature matches. We thus propose to guide the matches using coarse image-level deep correspondences.

Abstract

In this chapter, we look at the image matching task: finding accurate and robust keypoint correspondences between images. We perform an analysis of the possible supervisions for Deep Learning methods. We study the various supervisions that can be derived from large-scale Structure-from-motion. We show that weak supervision from epipolar geometry leads to better results than the stronger but more biased point level supervision and is a clear improvement over weak image level supervision. The trained networks are used to guide traditional local feature matching. We demonstrate the benefits of our

approach in a variety of conditions by evaluating our guided keypoint correspondences for localization of internet images on the YFCC100M dataset and indoor images on the SUN3D dataset, for robust localization on the Aachen day-night benchmark and for 3D reconstruction in challenging conditions using the LTLL historical image data.

The work presented in this chapter was initially presented in:

Learning to guide local feature matches (2020) François Darmon, Mathieu Aubry and Pascal Monasse, In *International Conference on 3D Vision (3DV)*

3.1 Introduction

Image matching is a fundamental task in computer vision and in particular a crucial step of Structure from Motion algorithms. Local feature detectors and descriptors are an essential tool for this task, providing both accuracy and high robustness. However, relying exclusively on local information to match images can be misleading in particular in the case of repeated or nearly repeated structures. We thus propose to complement and guide local keypoint matching using learned image-level coarse correspondences.

This idea represents an important shift compared to the dominant paradigm where global information and geometric constraints are usually introduced after keypoints have been matched, typically by performing RANSAC Fischler and Bolles (1981) to filter matches that are geometrically consistent. Indeed, recent work on applying deep learning to local feature correspondences has mostly focused on improving keypoint detection and description DeTone et al. (2018); Dusmanu et al. (2019); Luo et al. (2019b) or improving outlier rejection Moo Yi et al. (2018); Zhang et al. (2019b). To the best of our knowledge, we are the first to propose a combination of learned coarse correspondences and local keypoint matching, combining the benefits of both approaches.

As illustrated in Figure 3.1 our approach is especially beneficial in challenging conditions and in typical failure cases of classical features. First, when there are repeated structures in the image, they are likely to be disambiguated by the coarse matching and can thus be identified reliably by our guided keypoint matching. Second, when large appearance variations make descriptor matching less reliable, for example in the case of historical images, the number of candidate keypoint matches is reduced strongly by our method and false matches are less likely to appear.

We demonstrate that our approach boosts the results obtained with the standard SIFT descriptor to a level similar to the most advanced state-of-the-art deep descriptors. Our method can also be used with more advanced detectors and descriptors and we demonstrate it also boosts their performance, though by a smaller margin. This is a hint that a large part of the improvement brought by modern deep keypoint descriptors comes actually from their ability to consider global image information instead of exclusively at the local level. Note that this is explicitly targeted in some approaches and deep architectures such as ContextDesc Luo et al. (2019b).

The main challenge for guided matching is to predict coarse image correspondences. We build on an architecture computing correlation between base deep features and filtering them using a 4D convolutional network Rocco et al. (2018). This approach has the advantage to be able to handle any displacement and to leverage geometric consistency via the 4D convolutions. It is possible to train it with only weak image

supervision, providing the network with matching and non-matching image pairs. We introduce and study two other levels of supervision: weak epipolar supervision and point supervision. Indeed we can exploit large scale databases of 3D models reconstructed via Structure from Motion Heinly et al. (2015); Li and Snavely (2018) that provide camera calibration, from which we can infer epipolar constraints for all points as well as a sparse set of reconstructed points that can be used as ground truth matches. This data is, of course noisy and biased since points could only be reconstructed when traditional approaches succeeded, but we demonstrate it can still be used to boost performances.

Contrary to Rocco et al. (2018), both the weak epipolar supervision and point supervision improved results by fine-tuning the base features.
Our three main contributions are the following:

1. We propose the first learned guided correspondence approach for local keypoint matching.

2. We study different possible levels of supervision to learn coarse image matching, in particular weak supervision from epipolar geometry.

3. We demonstrate our method benefits all the studied keypoint descriptors. In some cases, it boosts the traditional SIFT descriptor to the performance of the latest learned descriptors, hinting it is mainly due to their discriminating power by considering global image characteristics.

3.2 Related Work

Several works Feng and Yuan (2011); Hartley and Zisserman (2004); Maier et al. (2016); Shah et al. (2015) introduced the idea of using an existing geometric model to guide keypoint matches. Hartley and Zisserman (2004) proposes to use a homography model, Shah et al. (2015) a fundamental matrix model, Feng and Yuan (2011) a combination of both and Maier et al. (2016) a specifically designed keypoint-based statistical optical flow. However, all these methods require an accurate initial keypoint based estimation of the geometric model in order to get more keypoint matches. For challenging scenarios such as day-night matching this is not realistic and adding guided matches from an incorrect geometry would add even more false matches. Other approaches Taira et al. (2019); Widya et al. (2018) match features of a pre-trained CNN in a hierarchical manner by first matching coarse deep features then higher resolutions features inside the receptive field of the matched features. Although very intuitive, it also requires good initial matches and we show that using pre-trained CNN features does not lead to good matches.

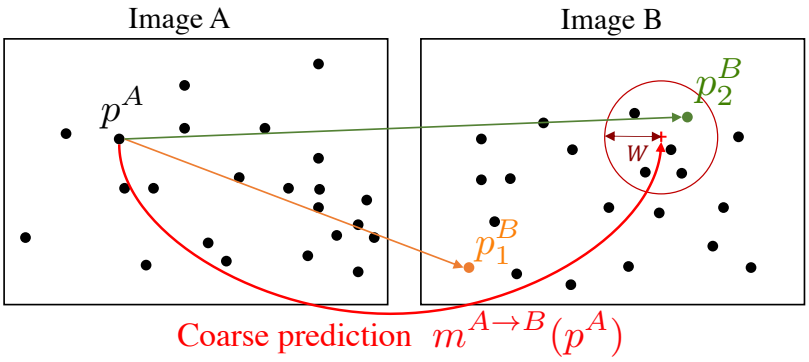


Fig. 3.2 Guiding keypoint matching with a coarse match. The orange match p_1^B is the closest in term of descriptor distance, but is not consistent with the coarse prediction. The correctly selected match, p_2^B (in green), is the closest in descriptor space being consistent with the coarse prediction.

3.3 Guided Feature Matching

Local keypoints have clear advantages for robust image matching. Indeed, they are naturally robust to occlusion of part of the image, localized changes, and clutter. Keypoint detectors are also designed to localize points with sub-pixel accuracy and to be robust to changes of viewpoint. However, local image regions are insufficient to reliably match keypoints in the presence of repetitive structures, which only large scale image information can help disambiguate. More generally, matches have to be identified among all the keypoints in the target image, and thus good matches have to be distinguished from a large number of false correspondences. We propose to make keypoint matching easier by first using a neural network to predict coarse correspondences at image level, and using them to guide keypoint matching, considering candidate matches only in a small image region.

This idea is illustrated in Figure 3.2. Let us assume we have access to an approximate match $m^{A\to B}$ between images A and B. We want to match a keypoint at position p^A in image A, described by a feature f^A to the keypoints detected in image B at positions p_i^B, described by features f_i^B, for $i = 1 \dots N$. We will leverage $m^{A\to B}$ by comparing f^A only to features of keypoints close to its approximate match $m^{A\to B}(p^A)$. The index j of the optimal match is given by:

$$j = \underset{i:\|m^{A\to B}(p^A)-p_i^B\|<W}{\arg\min} \|f^A - f_i^B\|, \tag{3.1}$$

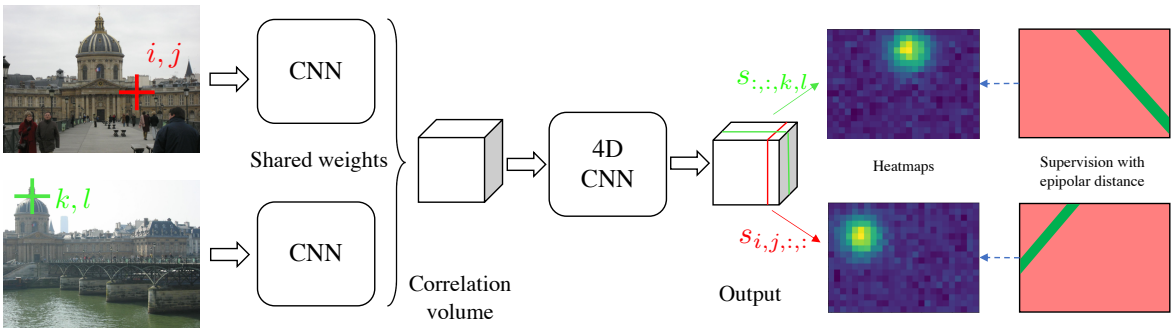


Fig. 3.3 Overview of our coarse matching network architecture and training: a shared CNN extracts coarse features from the two images. These features are then combined via the dot product into a 4D correlation volume. This volume is finally filtered with a CNN based on 4D convolutions, which can be trained with weak epipolar supervision.

where $W > 0$ is a parameter of our method. Note that using $W = \infty$ leads back to the standard matching. Similarly, the matching can be performed from image B to image A and the mutual matching test can remove outliers.

3.4 Learning coarse correspondences

In this section, we present our deep learning approach to predict approximate correspondences between images. The key elements of our approach are visualized in Figure 3.3. In the following, we first discuss our architecture, then present losses corresponding to three levels of supervision, and finally provide details of our implementation and training.

3.4.1 Architecture

We build on the NCNet Rocco et al. (2018) architecture. We first compute feature maps f^A and f^B for both input images and aggregate them in a 4D correlation volume $c_{ijkl} = \langle f^A_{ij} | f^B_{kl} \rangle$ that contains the correlation between every feature in image A and every feature in image B. We use a 4D convolutional neural network to filter the correlation volume into a new volume s, trained to have high values only in positions corresponding to valid correspondences. Rocco et al. (2018) motivates this architecture and the use of 4D convolutions by the idea of neighborhood consensus: the quality of a match between feature (i, j) in image A and (k, l) in image B should be decided not only based on the correlation c_{ijkl} but also on the correlation of the neighbor features.

The coarse matches between features in each direction are extracted from s using an argmax over the target image's dimensions. Such matches can be interpolated at pixel level: given a point in pixel coordinate (x, y) in image A, its coarse match $m^{A\to B}(x, y)$ in image B is computed using bilinear interpolation of the feature matches of its four nearest features. Inversely, $m^{B\to A}(x, y)$ denotes the coarse match in image A of pixel (x, y) in image B.

3.4.2 Supervision

We now introduce three different levels of supervision corresponding to different information about the ground truth matches and the associated losses. First, we consider an image level supervision, given in the form of pairs of matching and non matching images. Second, we introduce an epipolar supervision, which in addition leverages geometry information to infer a line where positive matches can lie. Third, we discuss a loss for point supervision, which uses ground truth matches between images.

In the rest of the section, we assume we are given a set $\mathcal{T}$ of training image pairs $(A_m, B_m)_{m=1...M}$ and we minimize the loss:

$$\mathcal{L} = \sum_{(A,B)\in\mathcal{T}} l^{A\to B}(s^{A\to B}) + l^{B\to A}(s^{B\to A}) \tag{3.2}$$

where $s^{A\to B}$ (resp. $s^{B\to A}$) is the result of applying a softmax to s in the dimensions corresponding to image B (resp. A) and $l^{A\to B}$ and $l^{B\to A}$ are the losses associated to the matches in both directions. For simplicity we only explain $l^{A\to B}(s^{A\to B})$ in the following subsections.

3.4.2.1 Weak image level supervision

For image level supervision, we use the same loss as Rocco et al. (2018). For each pair (A, B) of images, we write $y_{AB} = 1$ if both images represent the same scene and $y_{AB} = -1$ otherwise. We then define the loss by:

$$l^{A\to B}_{image} = -y_{AB} \sum_{(i,j)} \max_{(k,l)} s^{A\to B}_{ijkl}. \tag{3.3}$$

This loss encourages the maxima of $s^{A\to B}$ to be 1 for as many features as possible when the image pair is positive, which amounts to making the maxima in s sharper, and on the contrary when the pair is negative encourages the maxima of $s^{A\to B}$ to be small, which amounts to having almost constant values in s. In order to balance the influence

of negative and positive examples, the training batch consists of one half positive and one half negative image pairs. This supervision has been shown to achieve good results for semantic matching when image pair label is typically the only supervision available. However, we argue that additional information provided by 3D reconstruction datasets improve the matches.

3.4.2.2 Weak epipolar supervision

We propose to leverage epipolar geometry Hartley and Zisserman (2004) to better supervise the matches. Given a position (i, j) in image A, it is possible to use the camera calibrations (internal parameters and 6D pose) to predict the epipolar line on which the corresponding point in image B will lie. The distance between a position (k, l) in image B and this line is called the epipolar distance

$$d^F((i,j),(k,l)) = \frac{|(k,l,1)F(i,j,1)^\top|}{\sqrt{(F(i,j,1)^\top)^2_{[1]} + (F(i,j,1)^\top)^2_{[2]}}}, \tag{3.4}$$

where $\mathbf{t}_{[i]}$ denotes the ith coordinate of vector $\mathbf{t}$ and F is the fundamental matrix associated to the image pair, computed from the full calibration. We design a loss to leverage this information. Instead of trying to increase all maxima in positive image pairs, we try to increase only the ones consistent with epipolar geometry. Let $\mathcal{P}^{A\to B}$ be the subset of features in image A whose matches are consistent with epipolar geometry,

$$\mathcal{P}^{A\to B} = \left\{(i,j) | d^F\left((i,j), \underset{(k,l)}{\text{argmax}}\ s^{A\to B}_{ijkl}\right) < \lambda\right\}, \tag{3.5}$$

where λ is a threshold on the epipolar distance, and $\mathcal{N}^{A\to B}$ the complementary set of $\mathcal{P}^{A\to B}$, which correspond to matches that are not consistent with epipolar geometry. We propose to use as loss: As in the previous section, we use images from different scenes for half the batch. We consider that all the points for such image pairs are in $\mathcal{N}^{A\to B}$ and that the second term is zero. The division by 2 of the negative part of the loss then balances the positive and negative parts.

3.4.2.3 Point supervision

Point supervision is the strongest form of supervision we consider. It relies on sparse ground truth match labels. Let us assume that we are given a set of N ground truth correspondences between images $(p^A_1, p^B_1) \dots (p^A_N, p^B_N)$. Let $\mathcal{M}^{A\to B}(i,j)$ be the set of

features in image B that have a ground truth match with feature at (i, j) in image A. The loss we use for point supervision is

$$l_{points}^{A\to B} = -\sum_{ij} \max_{(k,l)\in\mathcal{M}^{A\to B}(i,j)} s_{ijkl}^{A\to B}. \tag{3.6}$$

This loss simply encourages $s^{A\to B}$ to be as close to 1 as possible for the best corresponding feature. Note that we could also use negative contributions as for the image level and epipolar supervision, or inversely consider only positive contributions for the epipolar supervision. We experimented with these variations and found that they lead to results worse than the losses we have discussed.

3.4.3 Implementation and training details

Similar to D2-NetDusmanu et al. (2019) we train the coarse matching network on MegaDepth dataset Li and Snavely (2018). This dataset consists of 196 sets of images collected from the same physical scene. COLMAP Schonberger and Frahm (2016) was run on these scenes to obtain a sparse 3D reconstruction. We removed from the training set all the scenes that are used in the evaluation: the Tanks and Temples scenes from FM benchmark Bian et al. (2019), the 4 YFCC scenes Heinly et al. (2015); Thomee et al. (2016) evaluated in OANet Zhang et al. (2019b), the 6 YFCC scenes of Image Matching Workshop Jin et al. (2020) and the buildings from LTLL Fernando et al. (2015). This reduces the training set to 175 scenes. We use the provided calibration for our weak supervision and choose as positive image pairs the ones that see at least 30 common 3D points in the reconstruction.

We use Resnet101 He et al. (2016) Conv4 features pretrained on ImageNet to extract feature maps from the input images. The 4D CNN is composed of three successive 4D convolutions layers with 16 channels and kernel of size 3. Similar to NCNet we ensure the output volume is independent of the image order by feeding the images in both orders successively and by taking the average of the outputs. The networks are trained with the Adam optimizer, an initial learning rate of 10^{-3}, and a batch size of 8 for 25000 iterations. For the epipolar supervision, λ is set to the distance between two consecutive features. The networks are initially trained with frozen feature extractors. Then after convergence, the feature extractors can be fine-tuned with a smaller learning rate. We limit the image resolution at 401 pixels at training time and keep the original aspect ratio with zero padding. At test time we limit the resolution to 497, which gives a feature resolution of at most 32×32. For a typical 1600×1600 image, each feature will correspond approximately to a 50 pixels square.

	Frozen features			Finetuned features		
Threshold	8	16	32	8	16	32
Image	34.5	55.0	65.36	36.3	57.8	68.7
Epipolar	43.1	62.4	70.7	**47.7**	**67.6**	**75.8**
Point	40.3	58.5	67.8	45.0	63.5	72.5

Table 3.1 Proportion of ground truth SfM points from MegaDepth correctly predicted by the coarse matcher. The threshold is in pixel units in the resized image coordinates. Here, 16 pixels is the distance between two consecutive features. Guiding with epipolar supervision leads to the highest proportion of matches in the guidance.

3.5 Experiments

In this section, we compare our approach with other guided matching methods, correspondence filtering techniques and state of the art feature extraction methods. First we validate and analyse the performance of our coarse matching network. Second, we compare our approach to other guided matching and correspondence pruning techniques. Third, we use our guided matching with different keypoint detectors and descriptors and show that our method consistently improves their results. Finally, we show that our method can help 3D reconstruction on challenging scenes.

3.5.1 Coarse matching

We first evaluate our coarse matching using the 3D points provided by MegaDepth as ground truth matches for a set of 1600 test image pairs. For each ground truth match (p^A, p^B), we compute the distance $\|m^{A\to B}(p^A) - p^B\|$. The proportion of distances below a threshold is used for evaluation. We use as threshold 8, 16 and 32 pixels since the distance between two nearby coarse matches is 16 pixels. We report in Table 3.1 the results obtained with our different supervisions as well as fine-tuning or not the ResNet-101 feature extractor, which was reported to degrade performances in the test database of Rocco et al. (2018). However, in our experiments finetuning the feature extractor leads to better matching, its effect being stronger with the epipolar and point supervisions. As can be expected, image supervision leads to the worst results. Although it is trained with a stronger supervision, point supervision has worse performances than epipolar supervision. This may be because point supervision is sparse and biased, providing information on specific areas of the image only. With a window size $W = 16$, the performance of epipolar supervision is close to 70%, which seems acceptable for

Matches	Pre-filtering	YFCC (internet)			Sun3D (indoor)		
		5°	10°	20°	5°	10°	20°
Raw	None	8.45	13.80	22.4	2.34	4.70	9.61
	Bidirectional check	27.70	36.43	47.73	6.96	11.72	19.89
	Ratio test	41.75	51.63	62.23	13.48	20.93	31.48
	Ratio test + bid. check	46.80	57.41	67.80	14.52	22.74	34.22
	Ratio test + GMS	30.43	38.30	48.16	11.49	17.89	27.46
Raw	CNNet	47.98	58.13	68.67	15.98	-	-
	N^3	49.13	-	-	15.38	-	-
	DFE	49.45	-	-	16.45	-	-
	OANet	**52.08**	**62.38**	**72.66**	**17.25**	**26.60**	**39.50**
Guided epipolar	Ratio test + bid. check	45.88	55.59	65.20	15.86	24.52	36.31
Guided homography	Ratio test + bid. check	46.00	55.65	65.46	15.15	23.55	35.36
Guided VGG4	Ratio test + bid. check	31.23	40.49	51.51	3.97	7.23	13.16
Ours image guided	Ratio test + bid. check	43.50	52.99	63.24	15.45	23.84	35.81
Ours point guided	Ratio test + bid. check	47.43	57.71	68.59	15.61	24.24	36.37
Ours epipolar guided	Ratio test + bid. check	49.60	60.36	71.37	15.72	24.35	36.40

Table 3.2 Comparison with various correspondence filtering and guided matching methods on 2-view geometry estimation. We report the AUC for a given tolerance for rotation and translation direction. The matches are computed from 2000 SIFT keypoints. "Ours Image/Point/Epipolar guided" is our guided matching with the different supervisions. A final RANSAC filtering follows any used pre-filtering.

guiding keypoint matching; we use this threshold to filter our matches in the rest of the experiments.

3.5.2 Comparison with guided matching and correspondence pruning

There is no direct benchmark for sparse matching. However, as mentioned earlier, sparse matching is the backbone of many 3D related tasks for which datasets exist and allow to indirectly evaluate the quality of matches. We compare our method for matching SIFT features with a posteriori filtering techniques and traditional guided matching on 2-view geometry estimation, both outdoor and indoor.

First, we use the setup of Zhang et al. (2019b) to evaluate 2-view geometry accuracy on pairs of images from the YFCC100M and Sun3D datasets. The YFCC100M dataset Thomee et al. (2016) is a very large collection of internet images that was used for Structure from Motion in Heinly et al. (2015). Four scenes and 1000 image pairs per scene are used for evaluation. Sun3D Xiao et al. (2013) data come from RGBD indoor

Fig. 3.4 Example of matching for SIFT with and without our method on Aachen daynight Sattler et al. (2018). Learned guided matching is able to correctly match keypoints in difficult conditions when the descriptor alone is not.

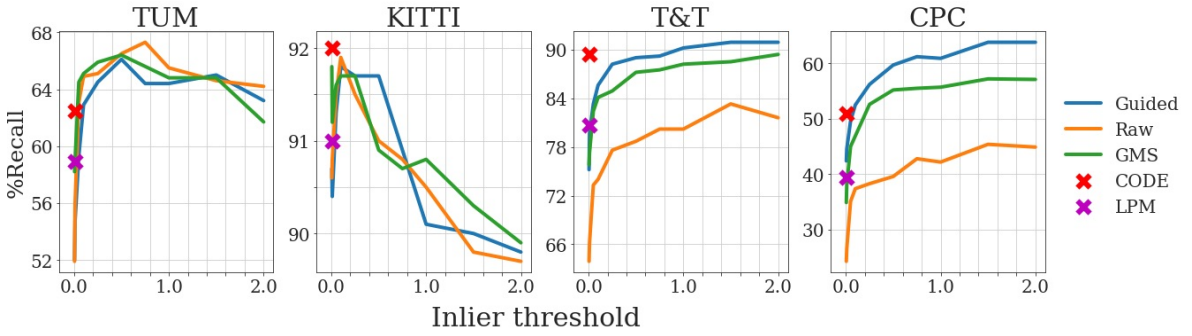


Fig. 3.5 FM benchmark results for a varying RANSAC threshold. A different RANSAC threshold must be carefully chosen for every dataset and every method for fair comparison. The compared methods perform similarly on KITTI and TUM but our guided matching performs the best on the wide baseline datasets.

videos. 15 indoor scenes and 1000 image pairs per scene are used for the evaluation. On both datasets, for each image pair, the matches provided by different approaches are used to estimate the essential matrix with RANSAC, which in turn is used to compute the relative pose (rotation and translation) Hartley and Zisserman (2004).

Our results are reported in Table 3.2. We compare our method for matching 2000 SIFT features with several correspondence pruning methods after classical nearest neighbor matching (raw matching). We also report the results for traditional guided matching baselines. Following Shah et al. (2015) the top 20% features in term of scale are first matched in order to estimate a geometric model. The model is then used to guide feature

Features	Matching	YFCC two-view geometry estimation			Sun3D two-view geometry estimation			Aachen day/night visual localization		
		5°	10°	20°	5°	10°	20°	(0.25m, 2°)	(0.5m, 5°)	(5m, 10°)
SIFT	Raw	46.80	57.41	67.80	14.52	22.74	34.22	38.8	51.0	58.2
	Ours Epip.	**49.60**	**60.36**	**71.37**	**15.72**	**24.35**	**36.40**	**66.3**	**84.7**	**96.9**
Contex-xtDesc	Raw	**55.40**	**66.58**	**77.38**	**16.83**	**25.77**	**37.99**	60.2	74.5	87.8
	Ours Epip.	51.95	62.60	73.33	16.50	25.43	37.56	**75.5**	**85.7**	**98.0**
Super-point	Raw	32.48	42.84	54.25	15.39	**24.27**	**36.37**	70.4	77.6	85.7
	Ours Epip.	**38.10**	**49.06**	**61.48**	**15.60**	24.23	36.33	**75.5**	**89.8**	**99.0**
D2-Net	Raw	**25.20**	**35.63**	49.43	13.52	22.67	35.61	**78.6**	85.7	**100**
	Ours Epip.	24.68	35.30	**49.55**	**14.10**	**22.87**	**35.63**	76.5	**87.8**	99.0

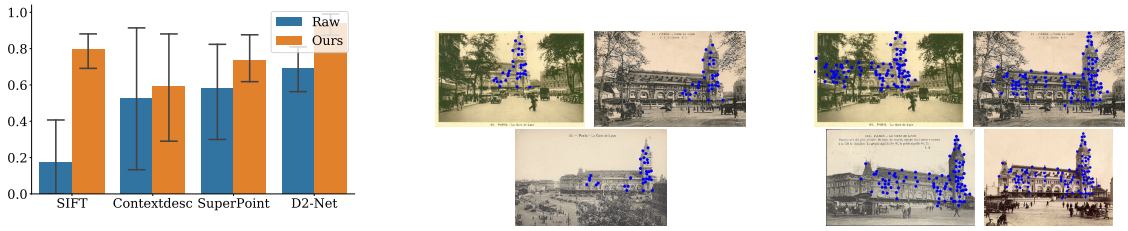
Table 3.3 Comparison with state of the art keypoint detectors and descriptors. We report AUC on several localization thresholds for YFCC and Sun3D, and the proportion of image sucessfully localized for Aachen benchmark. Raw descriptor denotes classical matching with mutual test and RANSAC. Ours Epip. is our guided matching with epipolar supervisions, mutual test, and RANSAC. We only show the best results for several ratio test thresholds (including no ratio test at all) before the other outlier filtering steps. Note that D2-Net's training set intersects YFCC100M test set.

matching. We evaluate two geometric models: homography and fundamental matrix Shah et al. (2015). We also compare with the pretrained VGG4 guided matching of Taira et al. (2019); Widya et al. (2018): for each mutual match between VGG4 features, we match the SIFT features located inside the receptive field of the corresponding VGG features. For clarity purpose, we only report for the ratio test experiments the results with the ratio that performed the best among 0.8, 0.9 and 0.95. Our method ranks second for two view geometry estimation after OANet. Interestingly, as noted in Sun et al. (2019), the ratio test is very important for SIFT matching; combined with bidirectional check, it is a very strong baseline. We note again that the epipolar supervision performs clearly better than the point supervision.

Second, we evaluate on the FM Benchmark Bian et al. (2019), a combination of scenes of Tanks and Temple (T&T) Knapitsch et al. (2017), TUM Sturm et al. (2012), KITTI Geiger et al. (2013) and Community Photo Collection (CPC) Wilson and Snavely (2014) datasets. Similar to the previous setup, the sparse matches are used to estimate the fundamental matrix that is compared with the ground truth. Each method is compared using the recall: the proportion of fundamental matrices correctly estimated. This metric is very sensitive to the inlier threshold chosen for RANSAC so we show in Figure 3.5 the recall of raw matches, our method and GMS Bian et al. (2017) for various inlier

thresholds. We also show results of the benchmark at the default threshold of 0.01 for CODE Lin et al. (2017) and LPM Ma et al. (2019). Since TUM dataset is an indoor dataset with short baseline, the difficulty lies more in the keypoint detection than on the matching and it is not surprising that all methods provide similar results. For KITTI, the results seem saturated and every method also performs similarly. On the two wide baseline scenes, our method shows a large improvement on raw SIFT matching and outperforms GMS by a significant margin.

3.5.3 Validation on learned keypoint detectors and descriptors



(a) Proportion of image successfully registered

(b) SuperPoint raw descriptor matching

(c) SuperPoint with our guided matching

Fig. 3.6 3D reconstruction results on challenging image sets Fernando et al. (2015). We show the average proportion of image registered in each scene of the historic sets in (a) and a comparison of the 3D reconstructions of scene "Sacre Coeur" for Superpoint raw descriptor (b) and our method (c). We show the projection of the reconstructed point cloud on every image added into the model. Our matching not only helps incorporate more images in the reconstruction but it also make it more dense.

In this section, we show that our guided matching benefits to many features, including the most recent learned deep features, by improving their results for 2-view geometry estimation and visual localization. In addition to the two-view geometry estimation results on YFCC100M and SUN3D, we report visual localization results on the local feature challenge from the Aachen day/night benchmark Sattler et al. (2018). The challenge provides a list of image pairs to match, from daytime to daytime, and from nighttime to daytime. The daytime to daytime matches are used to build a 3D point cloud. Then the nighttime to daytime matches are used to register the nighttime images to this model. The evaluation measure is the mean average precision (mAP) of the localization of all the query nighttime images. Note that since the evaluation is performed on 98 images only, small differences in performance should not be over-interpreted.

Our results on Aachen daynight as well as YFCC100M and SUN3D are reported in Table 3.3. Most features are improved by our method. Our guidance does not benefit ContextDesc on YFCC and Sun3D, hinting that this method is effective in adding global context. However, its performance is still improved on the harder Aachen day/night benchmark. The results for D2-net are inconclusive on the Aachen day/night benchmark but improvements are visible on SUN3D. We show qualitative examples in Figure 3.4.

Fig. 3.7 Examples of matching on LTLL Fernando et al. (2015) for SIFT features with and without our method compared with raw D2-Net. SIFT alone performs really poorly but with our guidance it provides better looking matches than state of the art D2-Net.

3.5.4 Application to challenging 3D reconstruction

We demonstrate that our approach can help 3D reconstruction in its most challenging cases by performing 3D reconstruction on the LTLL dataset Fernando et al. (2015). This dataset contains 25 sets of historical and recent pictures of the same scene. We tried to reconstruct the scenes both using historical images only and using recent and historical images. Many scenes are either too small or too complicated for 3D reconstruction with

any method. We focus our analysis on the 5 scenes that D2-Net could reconstruct from historical images only, and the 8 scenes it could reconstruct from all photographs.

In Figure 3.6a, we report the average proportion of registered images for the historical set using 4 different features with and without guidance from our network trained with epipolar supervision. For every feature the guided matching helps registering more images. In particular, guiding SIFT features with our methods registers the most images. We show in Figure 3.7 examples of image pairs matched with our method and in Figure 3.6, example of reconstruction of the old set of "Gare de Lyon" scene for SuperPoint with and without guidance. Guiding SuperPoint features helps registering one more image and the obtained point cloud is way more complete.

3.5.5 Limitations

Our method has two main drawbacks. First, it has a small but non negligible computational cost since the coarse matching adds an extra 70ms of computation time per image pair, to compare to the 30ms necessary to match 10000 SIFT keypoints. However it is dominated by the time of the 4D convolutions, which is currently based on loops of 3D convolution and could be made much faster by a direct CUDA implementation. Second, similar to traditional guided matching methods, it cannot be used to compute image visibility graphs for large scenes. Indeed, since they are trained on matching images, the coarse matches tend to be geometrically consistent even for input images representing different scenes.

3.6 Conclusion

We have presented a new paradigm to perform local feature matching. Our key idea is to use a deep learning model to predict coarse matches between images and use them to guide classical feature matches. We discussed several possible supervisions for this coarse matching model, and demonstrated the benefits of a weak epipolar supervision. Our method boosts the performances that can be obtained with SIFT features to the level of recent learning-based features. We also showed it leads to state of the art results in 3D tasks such as visual localization and 3D reconstruction in challenging conditions.

Now that we have studied data and supervision for the image matching task, and shown how Deep Learning improves downstream Structure-from-motion, we move on to the next subtasks in the usual reconstruction pipeline. We can now assume that the calibration are known and study the calibrated reconstruction task. In next chapter,

we are going to perform a similar analysis of data and supervision but applied to this different task.

Chapter 4

Deep Learning based Multi-View Stereo in the wild

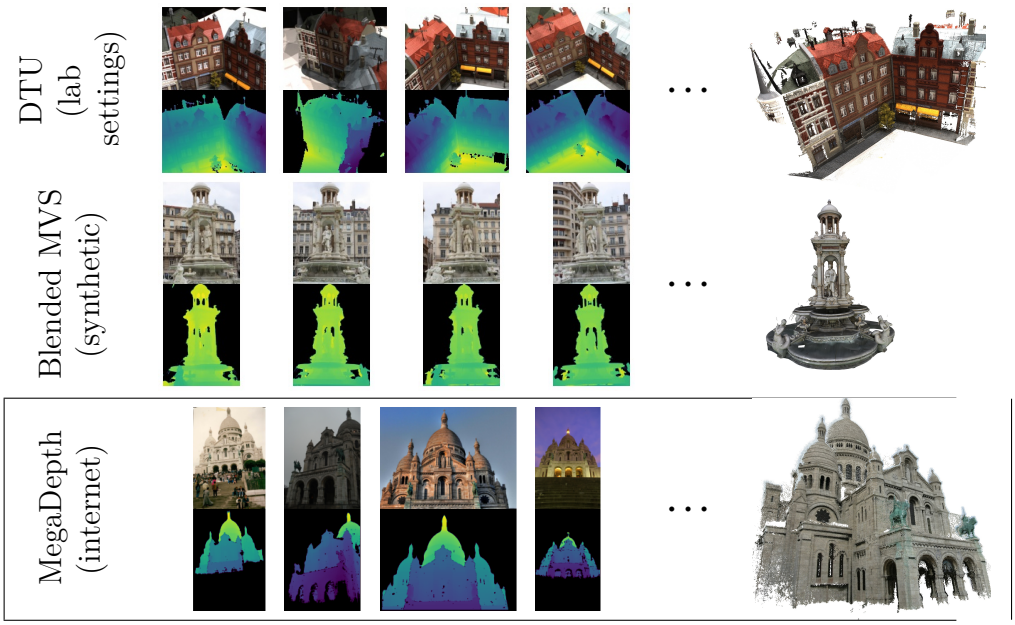


Fig. 4.1 We apply deep multi-view stereo networks to internet images and study the influence of architecture, supervision and training data over the quality of the reconstructed 3D models.

Abstract

We look at the calibrated reconstruction or Multi-View Stereo (MVS) task and how to best train Deep Learning networks for it, in term of data and supervision. Deep MVS methods have been developed and extensively compared on simple datasets, where they now outperform classical approaches. In this chapter, we ask whether the conclusions reached in controlled scenarios are still valid when working with Internet photo collections. We propose a methodology for evaluation and explore the influence of three aspects of deep MVS methods: network architecture, training data, and supervision. We make several key observations, which we extensively validate quantitatively and qualitatively, both for depth prediction and complete 3D reconstructions. First, complex unsupervised approaches cannot train on data in the wild. Our new approach makes it possible with three key elements: upsampling the output, softmin based aggregation and a single reconstruction loss. Second, supervised deep depthmap-based MVS methods are state-of-the art for reconstruction of few internet images. Finally, our evaluation provides very different results than usual ones. This shows that evaluation in uncontrolled scenarios is important for new architectures.

The work presented in this chapter was initially presented in:

Deep Multi-View Stereo gone wild (2021) François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse and Mathieu Aubry, In *International Conference on 3D Vision (3DV)*

4.1 Introduction

Deep Learning holds promises to improve Multi-View Stereo (MVS), i.e. the reconstruction of a 3D scene from a set of images with known camera parameters. However, deep MVS algorithms have mainly been developed and demonstrated in controlled settings, e.g. using pictures taken with a single camera, at a given time, specifically for 3D reconstruction. In this chapter, we point out the challenges in using such approaches to perform 3D reconstructions “in the wild" and propose an evaluation procedure for such a task. We analyze quantitatively and qualitatively the impact of network architecture and training for 3D reconstructions with a varying number of images. We give particular attention to unsupervised methods and propose a new approach which performs on par with state of the art while being much simpler. The reference dataset for the development of the vast majority of deep MVS approaches Chen et al. (2019); Gu et al. (2020); Luo et al. (2019a); Yao et al. (2018, 2019); Yu and Gao (2020) has been the DTU Jensen et al. (2014) dataset, consisting of 80 small scenes in sequences of 49 to 64 images. It was acquired in a lab, together with a reference 3D model obtained with a structured light scanner. While the scale of this dataset makes it suitable to train deep networks, it corresponds to a relatively simple reconstruction scenario, where many images taken in similar conditions are available. Recently, deep MVS approaches have started to evaluate their performance on larger outdoor datasets, such as *Tanks and Temples* Knapitsch et al. (2017) (T&T) and ETH3D Schöps et al. (2017). For example, VisMVSNet Zhang et al. (2020) outperformed all classical methods on the T&T dataset. Yao et al. Yao et al. (2020a) have recently proposed a synthetic dataset, BlendedMVS, specifically designed to train deep MVS approaches to generalize to such outdoor scenarios. We aim to go further and evaluate whether deep MVS networks can generalize to the more challenging setting of 3D reconstruction from Internet image collections. We are particularly interested in the challenging case with few images available for the reconstruction, where classical approaches struggle to reconstruct dense point clouds, and we propose an evaluation protocol for such a setup.

Such a generalization to datasets with different properties, where obtaining ground truth measurements is challenging, has been a major motivation for unsupervised deep MVS approaches Dai et al. (2019); Huang et al. (2020); Khot et al. (2019); Xu et al. (2021). We found however that many of these methods cannot train the network on real data. We thus propose a simpler alternative providing results on par with the best competing methods. It relies on minor modifications of the standard MVSNet Yao et al. (2018) architecture and is simply trained by comparing the images warped using the predicted depth.

Our experiments give several interesting insights. First, we show that existing unsupervised method are not suited for images in the wild but our proposed method is. Second, we show that depthmap-based MVS methods are state-of-the-art for reconstruction of few internet images. Third, in the wild evaluation provides very different results than evaluation on DTU and T&T. We therefore advocate for its use when developing new architectures.

Our main contributions are:

- We introduce a new experimental setup[1] for comparing different algorithms on images in the wild.
- We use this protocol to compare methods along three axes: network supervision, network architecture and training dataset.
- We show that existing unsupervised methods fail on images in the wild and thus we introduce a new unsupervised method.

4.2 Related Work

Supervised MVSNet MVSNet Yao et al. (2018) introduced a network architecture specific to depth map based MVS and end-to-end trainable. Further works improved the precision, speed and memory efficiency of MVSNet. Yan et al. (2020); Yao et al. (2019) propose a recurrent architecture instead of 3D convolutions. MVS-CRF Xue et al. (2019) uses a CRF model with learned parameters to infer depth maps. Many approaches introduce a multistep framework where a first depth map is produced with an MVSNet-like architecture, then the results are refined either with optimization Yao et al. (2019); Yu and Gao (2020), graph neural networks Chen et al. (2020, 2019) or multiscale neural networks Gu et al. (2020); Yang et al. (2020); Yi et al. (2020); Zhang et al. (2020). Another line of work aims at improving the per-view aggregation by predicting aggregation weights Luo et al. (2019a); Zhang et al. (2020). All these methods were evaluated in controlled settings and it is unclear how they would generalize to internet images.

Unsupervised MVSNet Unsupervised training is a promising approach for such a generalization. Dai et al. (2019); Huang et al. (2020); Khot et al. (2019); Xu et al. (2021) propose unsupervised approaches using several photometric consistency and regularization losses. However, to the best of our knowledge, these methods were never evaluated on

[1] Our PyTorch code and data are available at https://github.com/fdarmon/wild_deep_mvs.

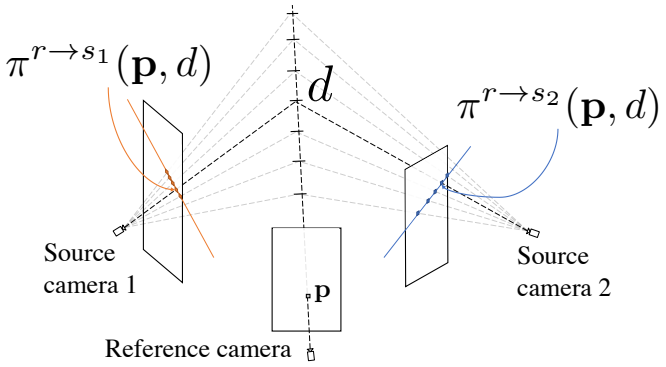


Fig. 4.2 Notations: A pixel $\mathbf{p}$ in reference frame is projected on source frames at a position depending on its depth.

Internet-based image datasets. Besides, their loss formulation is composed of many terms and hyperparameters, making them difficult to reproduce and likely to fail on new datasets. In this chapter we propose a simpler unsupervised training approach, which we show to perform on par with these more complex ones on standard data, and generalizes to Internet images.

4.3 MVS Networks in the Wild

In this section, we present standard deep depthmap-based MVS architectures and training which we compare, and propose some modifications, including our new unsupervised training, to better handle images in the wild.

4.3.1 Network Architectures

Overview and notations Starting from a set of calibrated images, deep MVS methods typically try to estimate a depth map for each image. The depth maps are then fused into a 3D point cloud using an independent method. The MVS network will take as input a *reference image* $\mathbf{I}^r$ and a set of *source images* $\mathbf{I}^{s_1} \dots \mathbf{I}^{s_N}$, and try to predict the depth $\mathbf{D}^r$ of the reference image.

Multi-view stereo architectures build on the idea that depth can be used to establish correspondences between images. More formally, as visualized in Figure 4.2, we write $\pi^{r\to s_k}(\mathbf{p}, d)$ the 2D position of the projection in the camera associated to $\mathbf{I}^{s_k}$ of the 3D point at depth d on the ray associated to the pixel $\mathbf{p} \in \mathbb{R}^2$ in $\mathbf{I}^r$. If d is the ground truth depth of the pixel $\mathbf{p}$ then the appearance of the reference image should be consistent with all the appearances of the source images $\mathbf{I}^{s_k}$ at pixels $\pi^{r\to s_k}(\mathbf{p}, d)$ Similar to the classic

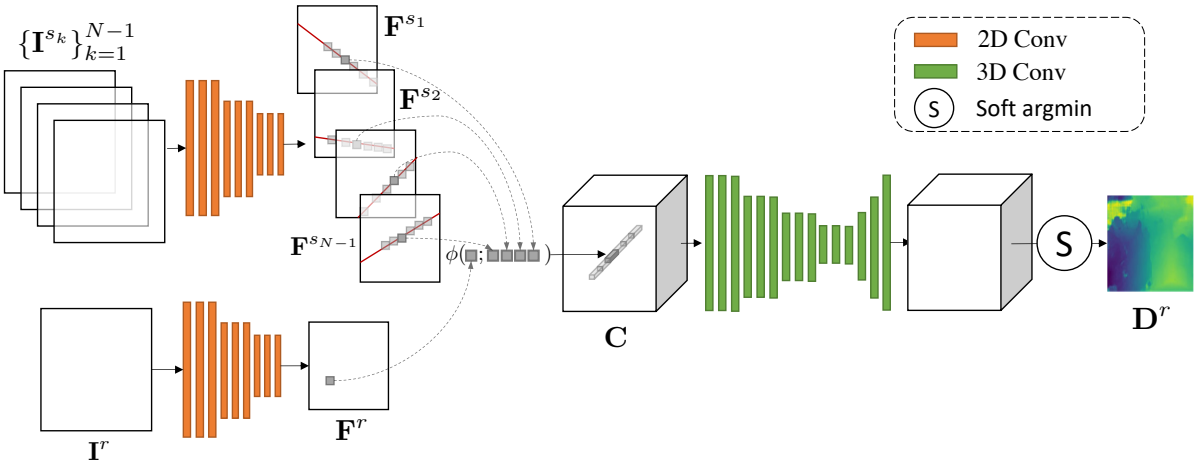


Fig. 4.3 MVSNet architecture: Given a set of source image and a reference view, this network architecture compute the depth map of reference frame. Feature maps are extracted with a shared CNN, then aggregated accross each view for a given depth using the aggregation function ϕ. The resulting 3D volume is then processed with 3D convolution in order to output a depth map with soft-argmax operation.

plane sweep stereo methods, the best performing deep MVS methods explicitly sample candidate depths and build a cost volume that computes for every pixel a score based on the correspondences given in every view by the candidate depths. This score can be seen as the cost of assigning a given depth to a pixel. The depth map can be extracted from the cost volume by looking at the minimum in the depth axis. MVSNet Yao et al. (2018) was the first deep architecture implementing this idea.

MVSNet MVSNet is shown in Figure 4.3. It starts by extracting a feature representation of each input image using a shared convolutional neural network (CNN). We write $\mathbf{F}^r$ the feature map of the reference image and $\mathbf{F}^{s_1} \dots \mathbf{F}^{s_N}$ the feature maps of the source images. The network also takes as input the depth range $[d_{\min}, d_{\max}]$ to consider. It is usually provided in standard datasets, and we explain in section 4.4.1 how we defined them in the context of reconstruction in the wild. The information from the feature maps are then aggregated into a 3D cost volume C for depth d regularly sampled from $d_{\min}$ to $d_{\max}$.

$$\mathbf{C}[\mathbf{p}, d] = \phi\left(\mathbf{F}^r[\mathbf{p}]; \{\mathbf{F}^{s_k}[\pi^{r \to s_k}(\mathbf{p}, d)], k = 1 \dots N\}\right), \tag{4.1}$$

where $\mathbf{F}[\cdot]$ denotes the interpolation of the feature map $\mathbf{F}$ and ϕ is an aggregation function. In MVSNet the channel-wise variance is used for aggregation:

$$\phi(\mathbf{f}_r; \{\mathbf{f}_{s_1}, \dots \mathbf{f}_{s_N}\}) = \text{var}(\mathbf{f}_r, \mathbf{f}_{s_1}, \dots \mathbf{f}_{s_N}). \tag{4.2}$$

The cost volume only encodes local information. A further step of 3D convolutions is then performed to propagate this local information. The depth map is finally extracted from the refined cost volume using a differentiable argmin operation.

Softmin-based aggregation The variance aggregation of MVSNet can be problematic when applied to images in the wild. Indeed, we can expect large viewpoint variations and many occlusions. Both would lead to large appearance differences in the pixels corresponding to the actual depth, and such outlier images would have a strong influence on the variance. Instead, we propose an alternative aggregation:

$$\phi(\mathbf{f}_r; \{\mathbf{f}_{s_1}, \dots \mathbf{f}_{s_N}\}) = \frac{\sum_{k=1}^{N} \exp(-\lambda \|\mathbf{f}_r - \mathbf{f}_{s_k}\|^2)(\mathbf{f}_r - \mathbf{f}_{s_k})^2}{\sum_{k=1}^{N} \exp(-\lambda \|\mathbf{f}_r - \mathbf{f}_{s_k}\|^2)}, \tag{4.3}$$

where λ is a learnable parameter. This score has two main differences with the variance aggregation. First, it breaks the symmetry between the reference and source images, each source feature is compared to the reference instead of the mean feature. Second, in our aggregation, the contribution of each source feature is weighted using a softmin weight. The intuition is that the features most similar to the ones of the reference image should be given higher weight compared to features that are completely different, which might be outliers. We will show in our experiments that such an aggregation is particularly beneficial in the presence of a high number of potential source images.

Multi-scale architectures. A known limitation of MVSNet is its large memory consumption. The consequence is that the resolution, both for image and depth is limited by the available memory. Several methods have been proposed to increase the resolution without increasing the memory requirements. Among them are the multiscale approaches that first scan coarsely the whole depth range using low resolution feature maps then refine the depth at higher resolutions. We used two successful multi-scale approaches: CVP-MVSNet Yang et al. (2020) and Vis-MVSNet Zhang et al. (2020). CVP-MVSNet starts from a downsampled version of the images to predict a coarse depth map using the same architecture as MVSNet. Then, at each scale, this depth map is upsampled and refined using the same network until full scale prediction is reached. This architecture is very light and can work with a variable number of scales. Vis-MVSNet uses a fixed number of scales with a different network for each scale. Moreover, contrary

to CVP-MVSNet, the lowest scale features are extracted from the full scale image using strided convolutions. Finally, Vis-MVSNet leverages a specific multi-view aggregation. It first predicts a confidence for each source image using a Bayesian framework Kendall and Gal (2017), then uses the confidence to aggregate the pairwise cost volumes associated with every source image. While this aggregation is complex, we believe it is one of the key ingredients that allows Vis-MVSNet to generalize well to Internet images.

4.3.2 Training

4.3.2.1 Supervised Training.

We first consider the case where a partial ground truth depth map D_{GT} is available, together with a binary mask M_{GT} encoding pixels for which the ground truth is valid. Supervised training is typically performed using the $\ell 1$ loss between the predicted depth map D and the ground truth depth map D_{GT}, weighted by the mask M_{GT} and normalized by the scene's depth range,

$$l(D, D_{GT}, M_{GT}) = \frac{\|M_{GT}(D - D_{GT})\|}{(d_{\max} - d_{\min})\|M_{GT}\|}. \tag{4.4}$$

4.3.2.2 Unsupervised Training

Motivated by the difficulty to obtain ground truth depth maps, unsupervised approaches attempt to rely solely on camera calibration. Approaches proposed in the literature typically combine a photometric consistency term with several additional losses. Instead, we propose to rely exclusively on the structural similarity between the reference image and the warped source images.

Similarly to MVS^2 Dai et al. (2019), we adopt a multi-reference setting where each image in a batch is successively used as a reference image. Let us consider a batch of images $\mathcal{I} = \{\mathbf{I}^1, \ ..., \mathbf{I}^{N+1}\}$ and the associated depth maps $\mathcal{D} = \{D^1, \ ..., D^{N+1}\}$ predicted by our MVS network. We write $\mathbf{W}^{b\to a}(D^a)$ the image resulting from warping $\mathbf{I}^b$ on $\mathbf{I}^a$ using the estimated depth map D^a:

$$\mathbf{W}^{b\to a}(D^a)[\mathbf{p}] = \mathbf{I}^b[\pi^{a\to b}(\mathbf{p}, D^a[\mathbf{p}])]. \tag{4.5}$$

Note that this warped image is not defined everywhere, since the 3D point defined from the pixel $\mathbf{p}$ and the depth $D^a[\mathbf{p}]$ in $\mathbf{I}^a$ might project outside of $\mathbf{I}^b$. Even when the warped image is defined, it might not be photoconsistent in case of occlusion. We thus follow MVS^2 Dai et al. (2019) and define an occlusion aware mask in $\mathbf{I}^a$, $M^{b\to a}(D^a, D^b)$, based

on the consistency between the estimated depths D^a and D^b in I^a and I^b (see Dai et al. (2019) for details).

We then define the loss for the batch as:

$$l(\mathcal{I}, \mathcal{D}) = \sum_{r=1}^{N} \frac{\sum_{s \neq r} \| M^{s \to r}(D^r, D^s) \text{SSIM}\left(\mathbf{I}_r, \mathbf{W}^{s \to r}(D^r)\right) \|}{\sum_{s \neq r} \| M^{s \to r}(D^r, D^s) \|} \tag{4.6}$$

where SSIM is the structural similarity Wang et al. (2004), which enables to compare images acquired in very different conditions. To avoid the need for further regularization outlined by other unsupervised approaches, we use the same approach as Shen et al. (2020): the depth maps, which are predicted at the same resolution as the CNN feature maps, are upsampled to the resolution of the original images before computing the loss.

Note that this loss is much simpler than in MVS^2 Dai et al. (2019), a combination of $\ell 1$ color loss, $\ell 1$ of the image gradients, SSIM and census transform. However, we show that it leads to results on par with state of the art and can be successfully applied to train a network from Internet image collections.

4.4 Benchmark of Deep MVS Networks in the Wild

To analyze the performance of different networks, we define training and evaluation dataset, evaluation metrics, as well as reference choices for several steps of the reconstruction pipeline, including reference depth to consider and a strategy to merge estimated depths into a full 3D model.

4.4.1 Data

Training datasets We experiment training on the DTU Jensen et al. (2014), BlendedMVS Yao et al. (2020a) and MegaDepth Li and Snavely (2018) datasets. Training an MVS network requires images with their associated calibration and depth maps. It also requires to select the set of source images for each reference image and the depth range for building the cost volume. DTU is a dataset captured in laboratory conditions with structured light scanner ground truth. It was preprocessed, organized, and annotated by Yao et al. Yao et al. (2018), who generated depth maps by meshing and rendering the ground truth point cloud. BlendedMVS was introduced in order to train deep models for outdoor reconstruction. It is composed of renderings of 3D scenes blended with real images to increase realism. Depth range and list of source views are given in the dataset. These datasets have reliable ground truth but there is a large domain gap between their

images and Internet images. Thus, we also experiment training on MegaDepth, which is composed of many Internet images with pseudo ground truth depth and sparse 3D models obtained with COLMAP Schonberger and Frahm (2016); Schönberger et al. (2016). We generate a training dataset from MegaDepth, excluding the training scenes of the Image Matching Benchmark Jin et al. (2020) which we will use for evaluation. We selected sets of training images by looking at the sparse reconstruction. We randomly sample a reference image and two source images uniformly among all the images that have more than 100 reconstructed 3D points, with a triangulation angle above 5 degree, in common with the reference. Once these sets of three images are sampled, we select the 3D points observed by at least three images and use their minimum and maximum depth as $d_{\min}$ and $d_{\max}$.

Depth map evaluation To evaluate depth map prediction, we used the synthetic BlendedMVS scenes, as well as the images from YFCC-100M Heinly et al. (2015); Thomee et al. (2016) used in the image matching benchmark of Jin et al. (2020). Indeed, Jin et al. (2020) provides 14 sets of few thousands of images with filtered ground truth depth maps obtained with COLMAP Schonberger and Frahm (2016); Schönberger et al. (2016).

3D reconstruction evaluation Because our ultimate goal is not to obtain depth maps but complete 3D reconstructions, we also select data to evaluate it. We used the standard DTU test set, but also constructed an evaluation setting for images in the wild. We start from the verified depth maps of the Image Matching Workshop Jin et al. (2020). We fuse them into a point cloud with COLMAP fusion using very conservative parameters: reprojection error below half a pixel and depth error below 1%. The points that satisfy such conditions in 20 views are used as reference 3D models. We manually check that the reconstructions were of high enough quality. The scenes *Westminster Abbey* and *Hagia Sophia interior* were removed since their reconstruction was not satisfactory. We then randomly selected sets of 5, 10, 20, and 50 images as test image sets and compare the reconstructions using only the images in these sets to the reference reconstruction obtained from the reference depth maps of a very large number of images. While the reference reconstructions are of course biased toward COLMAP reconstructions, we argue that the quality and extension of the reconstructions obtained from the small image sets is much lower than the reference one, and thus evaluating them with respect to the reference makes sense. Moreover, we never observed any case where part of the scene was reconstructed using deep MVS approaches on the test image sets and not in the reference reconstruction.

4.4.2 Metrics

Depth estimation We follow the same evaluation protocol as BlendedMVS Yao et al. (2020a). The predicted and ground truth depth maps are first normalized by $(d_{\max} - d_{\min})/128$ which makes the evaluation comparable for images seeing different depth ranges. We use the following three metrics: end point error (EPE), the mean absolute error between the inferred and the ground truth depth maps; $\mathbf{e_1}$, the proportion in % of pixels with an error larger than 1 in the scaled depth maps; $\mathbf{e_3}$, the proportion in % of pixels with an error larger than 3.

3D reconstruction On DTU Jensen et al. (2014) we follow the standard evaluation protocol and use as metrics: Accuracy, the mean distance from the reconstruction to ground truth, completeness from ground truth to reconstruction, and the overall metric defined as the average of the previous two.

For our evaluation in the wild, we use the same metrics as T&T Knapitsch et al. (2017): precision, recall and F-score at a given threshold. The reference reconstruction is known only up to a scale factor therefore the choice of such threshold is not straightforward. Let D_k be the ground truth depth map of image I_k, we define the threshold as:

$$t = \underset{k}{\text{median}} \left(\underset{\|\mathbf{p}-\mathbf{p}'\|=2}{\text{median}} \|D_k[\mathbf{p}]\mathbf{p} - D_k[\mathbf{p}']\mathbf{p}'\| \right). \tag{4.7}$$

4.4.3 Common Reconstruction Framework

To fairly compare different deep MVS approaches for 3D reconstruction, we need to fuse the predicted depth maps in a consistent way. Several fusion methods exist, each with variations in the point filtering and the parameters used. For DTU reconstructions, we use Fusibile Galliani et al. (2015) with the same parameters for all compared methods. Since Fusibile is only implemented for fixed resolution images, we use a different method for Internet images. We use the fusion procedure of COLMAP Schönberger et al. (2016) with a classic pre-filtering step similar to the one used in MVSNet: we only keep depths consistent in three views. The consistency criterion is based on three thresholds: the reprojected depth must be closer than 1% to original depth, the reprojection distance must be less than 1 pixel, and the triangulation angle must be above 1°.

Table 4.1 Ablation study of unsupervised learning. We compare networks trained with and without upsampling (Up.) of the predicted depth map, with and without occlusion masking (Occ.) and using as aggregation function (ϕ) either Variance aggregation (V, eq. 4.2) or our Softmin aggregation (S, eq. 4.3).

ϕ	Up.	Occ.	DTU reconstruction			YFCC depth maps		
			Prec.	Comp.	Over.	EPE	e_1	e_3
V			1.000	0.803	0.901	34.74	91.88	80.99
V	✓		0.614	0.580	0.597	21.68	67.48	48.44
S	✓		**0.607**	0.560	0.584	21.88	66.39	46.61
V	✓	✓	0.610	0.545	0.578	18.75	63.07	43.17
S	✓	✓	0.608	**0.528**	**0.568**	**18.22**	**61.97**	**41.34**
Unsup. MVSNet Khot et al. (2019)			0.881	1.073	0.977			
MVS^2 Dai et al. (2019)			0.760	**0.515**	0.637			
M^{3}VSNet Huang et al. (2020)			0.636	0.531	0.583	40.57	82.45	69.91
JDACS Xu et al. (2021)			**0.571**	**0.515**	**0.543**	34.37	92.36	80.45

4.5 Experiments

We first provide an ablation study of our unsupervised approach and compare it with state of the art unsupervised methods (Section 4.5.1). Then, we compare the different network architecture and supervisions for depth map estimation (Section 4.5.2) and full 3D reconstruction (Section 4.5.3).

4.5.1 Unsupervised Approach Analysis

We validate and ablate our unsupervised approach on both DTU 3D reconstruction and YFCC depth map estimation in Table 4.1. We test the importance of upsampling the depth map as a way to regularize the network, compare variance and softmin aggregation functions as well as results with and without occlusion aware masking. We also compare our approach with existing work.[2] The main observation is that upsampling is the key factor to obtain meaningful results. Softmin aggregation is consistently better than variance for both datasets and evaluations. Occlusion masking also consistently improves the results, by a stronger margin on YFCC where occlusions are expected to be more common. Using upsampling and occlusion masking, our simple training loss is on par with state of the art unsupervised methods on DTU but is is much better for YFCC trainings where competing method have poor results.

[2]We use open source implementations of M3VSNet and JDACS provided by the authors.

Table 4.2 Direct depth map evaluation: Comparison of architectures MVSNet Yao et al. (2018), our MVSNet with softmin aggregation (MVSNet-S), CVP-MVSNet Yang et al. (2020) and Vis-MVSNet Zhang et al. (2020), trained on BlendedMVS (B), DTU or MegaDepth (MD), with or without depth supervision (Sup.).

Archi	Train	Sup.	BlendedMVS			YFCC		
			EPE	e_1	e_3	EPE	e_1	e_3
MVSNet	B	✓	**1.49**	**21.98**	**8.32**	21.56	67.93	49.75
	DTU	✓	4.90	39.25	22.01	32.41	79.74	65.42
	DTU		3.94	31.68	16.61	24.61	70.77	53.44
	MD	✓	2.29	30.08	12.37	**17.61**	**62.54**	43.09
	MD		3.24	32.94	15.25	18.75	63.07	**42.17**
MVSNet-S	B	✓	**1.35**	**25.91**	**8.55**	20.98	69.57	49.86
	DTU	✓	4.18	42.07	19.87	27.18	78.25	60.53
	DTU		4.05	32.50	16.01	21.07	68.64	50.35
	MD	✓	1.91	32.91	11.93	**15.57**	**60.44**	**38.18**
	MD		2.88	33.44	14.14	18.22	61.97	41.34
CVP-MVSNet	B	✓	**1.90**	**19.73**	**10.24**	40.07	85.88	76.25
	DTU	✓	10.99	46.79	35.59	73.69	95.29	90.69
	DTU		5.25	29.33	19.77	45.36	89.81	82.17
	MD	✓	3.07	24.33	14.40	34.39	**78.49**	**66.57**
	MD		3.39	21.67	12.94	**32.74**	78.92	66.99
Vis-MVSNet	B	✓	**1.47**	**18.47**	**7.59**	19.60	64.98	46.38
	DTU	✓	3.70	30.37	18.16	27.46	72.89	58.37
	DTU		7.22	37.03	20.75	38.32	73.24	56.17
	MD	✓	2.05	22.21	10.14	**16.01**	**56.71**	**38.20**
	MD		3.88	31.64	17.50	19.21	66.27	47.34

4.5.2 Depth Map Prediction

In Table 4.2 we compare different approaches for depth prediction on the YFCC data and BlendedMVS validation set. We study different architectures, the standard MVSNet architecture with either the standard variance aggregation or our proposed softmin aggregation and two state-of-the-art multiscale architecture CVP-MVSNet Yang et al. (2020) and Vis-MVSNet Zhang et al. (2020).[3] We compare results for networks trained on DTU, BlendedMVS (B) and MegaDepth (MD) with either the supervised $\ell 1$ loss or our occlusion masked unsupervised loss (except for BlendedMVS which we only use in the supervised setting since it is a synthetic dataset designed specifically for this purpose).

[3]For both CVP-MVSNet and Vis-MVSNet, we adapted the public implementation provided by the authors

MVSNet architecture As expected, the best performing networks on a given dataset are the ones trained with a supervised loss on the same dataset. Confirming our analysis in the unsupervised setup, we can also see that MVSNet modified with our softmin-based aggregation systematically outperforms the variance-based aggregation. The results for unsupervised settings are more surprising. First, networks trained on DTU in the unsupervised setting generalize better both to blended-MVS and YFCC, hinting that unsupervised approaches might lead to better generalization even without changing the training data. Also, the unsupervised network trained on MegaDepth performs better on YFCC than the supervised networks trained on BlendedMVS, outlining again the potential of unsupervised approaches.

Multi-scale architectures CVP-MVSNet, which is our best performing architecture on DTU (see next section) and achieves the second best results in terms of e_1 error on BlendedMVS, did not achieve satisfactory results on YFCC images even when supervised on MegaDepth. This may be because at the lowest resolution the architecture takes as input a low resolution downsampled version of the image, on which correspondences might be too hard to infer for images taken in very different conditions.

On the contrary, Vis-MVSNet supervised on MegaDepth data is the best performing approach in terms of e_1 error. However, its performance is worse in the unsupervised setting compared to MVSNet. This is likely explained by the complexity of the loss function of Vis-MVSNet. In the original paper, the loss combines pairwise estimated depth maps and final depth maps. The pairwise loss is computed in a Bayesian setting Kendall and Gal (2017) and our direct adaptation of this loss by replacing $\ell 1$ with photometric loss might be too naive.

Number of source views Until now, all experiments were performed using one reference view and four source views as input to the network at test time. However, in real scenarios, one might have access to more views, and it is thus important for the network to benefit from an increased number of views at inference time. We thus investigate the use of more source images at test time, still training with two source images only, and report the results in Figure 4.4. The results for MVSNet architecture with variance-based aggregation get worse as the number of source views is increased. This may be because the relevant information in the cost volume is more likely to be masked by noise when the number of source images increases. On the contrary, the performances of our softmin-based aggregation or Vis-MVSNet aggregation improve when using a higher number of source views.

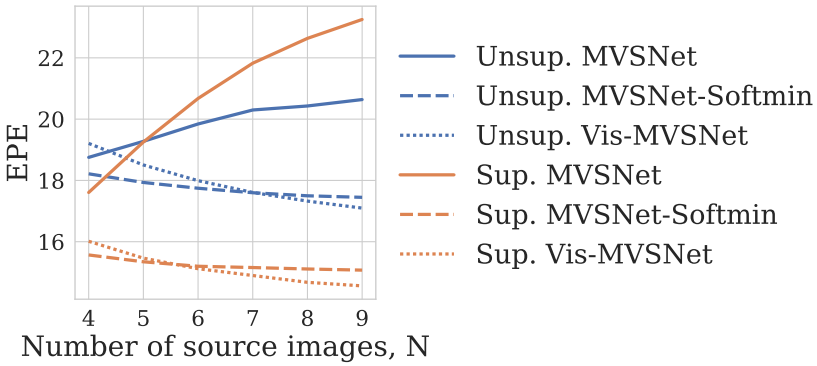


Fig. 4.4 EPE on YFCC depth map estimation as a function of the number of source images N for networks trained on MegaDepth, supervised (Sup.) or unsupervised (Unsup.). Contrary to the other architectures MVSNet with variance aggregation does not benefit from using more images.

4.5.3 3D Reconstruction

We now evaluate the quality of the full 3D reconstruction obtained with the different architectures. The predicted depth maps are combined with a common fusion algorithm and the evaluation is performed on the fused point cloud. We first report the results on DTU in Table 4.3. Interestingly, our softmin-based adaptation degrades performance when training on DTU in a supervised setting, but not in an unsupervised setting. Also note that the performance on DTU, and in particular the completeness of the reconstruction, is largely improved when the networks are trained on MegaDepth data. The best performance is even obtained in an unsupervised setting with CVP-MVSNet. However, this network performs poorly on YFCC data.

Finally, we evaluate the quality of the 3D reconstructions obtained from small sets of YFCC images. For fair comparison with COLMAP Schönberger et al. (2016), the depth maps are evaluated with up to 19 source images for these experiments, except when using the original MVSNet architecture with variance aggregation, which does not benefit from using more images and which we continue to use with four source images. Since depth estimation on YFCC images completely failed with CVP-MVSNet as well as with networks trained on DTU, we did not perform reconstructions in these cases. Quantitative results are reported in Table 4.4 and qualitative results are shown in Figure 4.5.

Most of the trends observed for depth map prediction can also be observed in this experiment. In particular, networks trained on MegaDepth perform better than those trained on BlendedMVS, our softmin-based aggregation improves results for the MVSNet

Table 4.3 3D reconstruction evaluation on DTU. Comparison of trainings on BlendedMVS, MegaDepth and DTU for various architectures with or without supervision ('Sup.' column).

Archi.	Training data	Sup.	DTU reconstructions		
			Acc.	Comp.	Overall
MVSNet	Blended	✓	0.487	0.496	0.491
	DTU	✓	**0.453**	**0.488**	**0.470**
	DTU		0.610	0.545	0.578
	MD	✓	0.486	0.547	0.517
	MD		0.689	0.645	0.670
MVSNet softmin	Blended	✓	0.631	0.738	0.684
	DTU	✓	**0.598**	0.531	**0.564**
	DTU		0.609	**0.528**	0.568
	MD	✓	0.625	0.548	0.586
	MD		0.690	0.614	0.652
CVP-MVSNet	Blended	✓	0.364	0.434	0.399
	DTU	✓	0.396	0.534	0.465
	DTU		**0.340**	0.586	0.463
	MD	✓	0.360	0.376	0.368
	MD		0.364	**0.370**	**0.367**
Vis-MVSNet	Blended	✓	0.504	0.411	0.458
	DTU	✓	**0.434**	0.478	0.456
	DTU		0.456	0.596	0.526
	MD	✓	0.438	**0.345**	**0.392**
	MD		0.451	0.873	0.662

Table 4.4 3D reconstruction evaluation on our 'in the wild' benchmark. We compare several architectures trained with and without supervision ('Sup.' column) on BlendedMVS (Blended) and MegaDepth (MD). We report the best between using four source views (*) and all source views (up to 19).

Arch.	Train. data	Sup.	5 images			10 images			20 images			50 images		
			Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
MVS-Net	B	✓	**84.68**	11.51	19.16	**91.37***	23.27*	35.83*	**91.46***	36.46*	51.73*	**93.40***	51.50*	66.08*
	MD	✓	84.31	14.84	23.87	91.05*	27.62*	41.51*	89.94*	39.55*	54.60*	92.07*	52.69*	66.75*
	MD		80.44	**15.31**	**24.39**	90.17*	**31.05***	**45.46***	88.15*	**43.84***	**58.22***	92.22*	**57.27***	**70.39***
MVS-Net -s	B	✓	74.67	2.14	3.95	83.66	11.68	18.97	83.43	19.64	29.74	**89.93**	37.85	52.40
	MD	✓	**84.60**	11.70	19.36	**88.65**	30.99	44.43	**85.93**	43.16	56.45	87.69	57.10	68.70
	MD		82.02	**14.71**	**23.71**	83.67	**37.63**	**50.95**	79.09	**51.40**	**61.44**	81.74	**63.66**	**71.29**
Vis-MVS-Net	B	✓	87.43	13.19	21.58	87.98	35.82	49.87	85.66	53.65	65.41	89.26	64.99	75.03
	MD	✓	**91.44**	**21.72**	**33.30**	**93.23**	**48.22**	**63.05**	**90.25**	**64.40**	**74.85**	**90.55**	**72.88**	**80.67**
	MD		84.25	8.90	14.91	84.19	32.45	45.69	81.10	53.27	63.52	86.13	68.03	75.79
COLMAP			89.95	17.30	27.85	93.20	42.76	57.97	94.88	62.68	75.11	96.78	73.64	83.46

architecture, in particular when many views are available and in the unsupervised setting, and Vis-MVSNet outperforms MVSNet in the supervised setting.

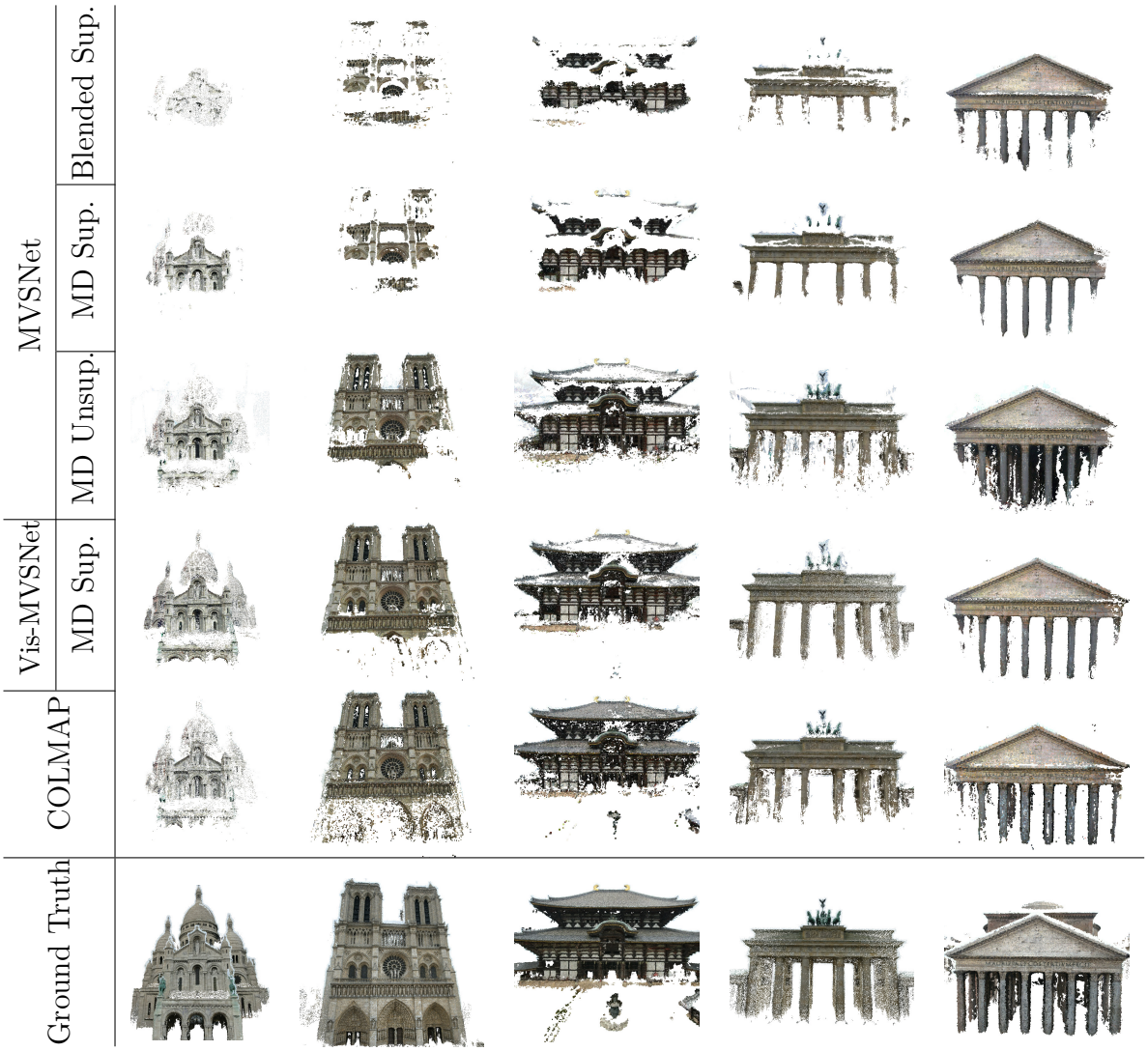


Fig. 4.5 Comparison of reconstructions from ten images using different architectures trained on BlendedMVS (Blended) or MegaDepth (MD), supervised (Sup.) or unsupervised (Unsup.).

A new important observation is that the unsupervised MVSNet performs better than the supervised version, especially for the recall metric. This can also be seen in Figure 4.5 where the reconstructions appear more complete in the unsupervised setting. We argue that this is a strong result that advocates for the development of unsupervised approaches suitable for the more advanced architectures. Interestingly, this result was not noticeable in our depth map evaluation.

Another significant observation is that Vis-MVSNet trained with supervision on MegaDepth quantitatively outperforms COLMAP when using very small image sets (5 or 10 images) with both higher precision and higher recall. This is also a very encouraging

result, showing that Deep MVS approaches can provide competitive results in challenging conditions. However, one should note that this quantitative result is not obvious when looking at the qualitative results where either COLMAP or Vis-MVSNet achieves better looking results, depending on the scene.

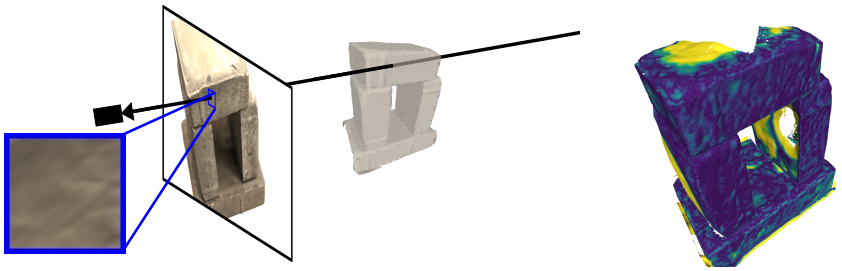
4.6 Conclusion

We have presented a study of the performance of deep MVS methods on Internet images, a setup that, to the best of our knowledge has never been explored. We discussed the influence of training data, architecture and supervision. For this last aspect, we introduced a new unsupervised approach which outperforms state-of-the art. Our analysis revealed several interesting observations, which we hope will encourage systematic evaluation of deep MVS approaches with challenging Internet images and stimulate research on unsupervised approaches, to ultimately have deep learning bring clear performance boosts in this important scenario.

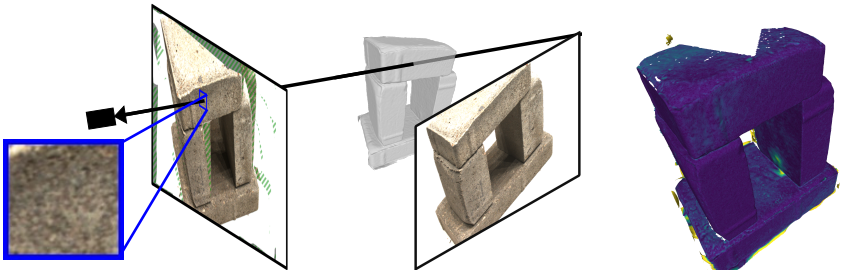
We have studied depthmap-based deep MVS since this kind of data where the desired output is an image is well suited for Deep Learning prediction. But Deep learning can be used as the 3D representation itself. This is the idea behind neural implicit representations. It has many theoretical advantages over depthmap representations: (i) it encodes in a single representation both the surface position and its normal, (ii) it can be directly converted to meshes, and (iii) it could encode higher resolution with a smaller memory requirement. However, neural implicit reconstructions are still worse than existing methods in term of surface accuracy. In next chapter, we develop a new neural implicit method for calibrated multiview reconstruction that improves upon state-of-the-art by using insights from traditional MVS methods.

Chapter 5

Multi-view reconstruction with implicit surfaces and patch warping



(a) Volumetric rendering (left) and geometric error map (right).



(b) Image warping (left) and geometric error map with our method (right)

Fig. 5.1 Standard neural implicit surface approaches jointly optimize a geometry and color network, but struggle to represent high frequency textures and therefore lack accuracy (top). We propose to additionally warp image patches with the implicit geometry, which allows to directly optimize photo-consistency between images and significantly improves the reconstruction accuracy (bottom).

Abstract

In this chapter, we tackle the challenge of representation of 3D data for Deep Learning. Neural implicit representations have many theoretical properties that would make it superior to traditional representations such as point clouds or depth-maps. However, its use in calibrated multi-view reconstruction still produces low accuracy reconstructions. We argue that this comes from the difficulty to learn and render high frequency textures with neural networks. We thus propose to add to the standard neural rendering optimization a direct photo-consistency term across different views. Intuitively, we optimize the implicit geometry so that it warps views on each other in a consistent way. We demonstrate that two elements are key to the success of such an approach: (i) warping entire patches, using the predicted occupancy and normals of the 3D points along each ray, and measuring their similarity with a robust structural similarity (SSIM); (ii) handling visibility and occlusion in such a way that incorrect warps are not given too much importance while encouraging a reconstruction as complete as possible. We evaluate our approach on the standard DTU and EPFL benchmarks and show it outperforms state-of-the-art unsupervised implicit surfaces reconstructions by over 15% on both datasets.

The work presented in this chapter will appear in:

Improving neural implicit surfaces geometry with patch warping (2022) François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse and Mathieu Aubry, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

5.1 Introduction

Multi-view 3D reconstruction is the task of recovering the geometry of objects by looking at their projected views. Multi-View Stereo (MVS) methods rely on the photo-consistency of multiple views and typically provide the best results Schönberger et al. (2016); Zhang et al. (2020). However, they require a cumbersome multi-step procedure, first estimating then merging depth maps. Recent 3D optimization methods Mildenhall et al. (2020); Niemeyer et al. (2020); Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021, 2020) avoid this issue by representing the surface implicitly and jointly optimizing neural networks encoding occupancy and color for all images, but their accuracy remains limited. In this chapter, we bridge these two types of approaches by optimizing multi-view photo-consistency for a geometry represented by implicit functions. We show that this enables our method to leverage high-frequency textures present in the input images that existing implicit methods struggle to represent, resulting in significant accuracy gains.

The idea behind our approach is visualized on Fig. 5.1. The top row shows a rendering and the geometric error map (5.1a) for a state of the art implicit method Yariv et al. (2021). The rendering fails at producing high frequency textures, resulting in low 3D accuracy. To overcome this limitation we use the original images, reprojecting them using the geometry described by the implicit occupancy function. This is shown on the bottom row (5.1b) where our warped patch includes high frequency texture. Consequently, we can optimize the geometry much more accurately, resulting in smaller geometric errors in the reconstruction.

Optimizing the implicit geometry for photo-consistency poses two main challenges. First, since we do not have perfectly Lambertian materials, directly minimizing the difference between colors is not meaningful and would lead to artefacts. We thus compare entire patches using a robust similarity (SSIM Wang et al. (2004)) which requires performing patch warping using the implicit geometry. Building on the volumetric neural implicit surface framework, we start by sampling 3D points on the ray associated to each pixel in a reference image. We then propose to warp for each sampled point a source image patch to the reference image using a planar scene approximation, and finally combine all warped patches. Second, opposite to standard neural rendering methods that can associate a color to each 3D point, a warping-based approach must deal with the fact that many 3D points do not project correctly in the source view, e.g. are not visible or are occluded. This will typically happen for points sampled on any ray, we thus define for each reference image pixel and each source image a soft visibility mask. We then completely remove from the loss pixels in reference image that have no valid reprojection in any of the source views and, for the other pixels, weight the loss associated to each

source view depending on how reliable the associated projection is. This downweights invalid reprojections, while encouraging a reconstruction as complete as possible.

We evaluate our method on the DTU Jensen et al. (2014) and EPFL Strecha et al. (2008) benchmarks. Our method outperforms current state-of-the-art unsupervised neural implicit surfaces methods by a large margin: the 3D reconstruction metrics are on average improved by 15%. We also show qualitatively that our image warps are able to capture high frequency details.

To summarize, we present:

- a method to warp patches using implicit geometry;
- a loss function able to handle incorrect reprojections;
- an experimental evaluation demonstrating the very significant accuracy gain on two standard benchmarks and validating each element of our approach.

5.2 Related Work

Recently, new implicit representations of 3D surfaces with neural network were introduced. The surface is represented by a neural network which will output either an occupancy field Mescheder et al. (2019); Peng et al. (2020) or a Signed Distance Function (SDF) Park et al. (2019). These representations are used to perform multi-view reconstruction following two different paradigms Oechsle et al. (2021): surfacic Niemeyer et al. (2020); Yariv et al. (2020); Zhang et al. (2021) and volumetric Mildenhall et al. (2020); Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021). Surfacic approaches compute the surface then backpropagate through this step with implicit differentiation Atzmon et al. (2019). They are hard to optimize and typically require additional supervision: silhouette masks in Niemeyer et al. (2020); Yariv et al. (2020) or the output of a pretrained depth map estimator Zhang et al. (2020) in MVSDF Zhang et al. (2021). Volumetric approaches were introduced in NeRF Mildenhall et al. (2020). It combines classical volumetric rendering Max (1995) with implicit functions to produce high quality renderings of images. The main focus of NeRF Mildenhall et al. (2020) was the quality of rendering, therefore the geometry was not evaluated. Further work adapted the geometric output Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021) to make it better suited for surface extraction. UNISURF Oechsle et al. (2021) uses an occupancy network Mescheder et al. (2019) whereas VolSDF Yariv et al. (2021) and NeuS Wang et al. (2021a) use an SDF Park et al. (2019). We build our method on VolSDF Yariv

et al. (2021) but we believe it could be adapted to fit any volumetric neural implicit framework.

Image warping and neural implicit surfaces: In this chapter we combine the color matching idea of traditional MVS with neural implicit surfaces. Closest to this idea is MVSDF Zhang et al. (2021) that also uses a loss based on correspondences and works on accurate geometry optimization. However, it optimizes consistency between CNN features and the optimization requires a network pretrained on multi-view datasets Yao et al. (2020b). Our approach does not require any pretrained network and we show that it outperforms MVSDF.

The idea of projecting information from source views to 3D then using the neural radiance field framework to render a target view has also been used in learning-based approaches Bergman et al. (2021); Chen et al. (2021); Chibane et al. (2021); Trevithick and Yang (2021); Wang et al. (2021b); Yu et al. (2021). These approaches train on multiple scenes networks that take as input features from the source views aggregated at a given 3D point and output radiance and occupancy for this point. These papers however focus on the generalization to new scenes and the quality of rendered views, but to the best of our knowledge, the quality of their predicted geometry has not been evaluated. On the contrary, we focus on the optimization framework, i.e. we do not train our network on several scenes, and we optimize the quality of geometry.

5.3 Method

In this section, we present our technical contributions. Section 5.3.1 introduces the volumetric rendering framework Mildenhall et al. (2020); Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021) on which we build. Section 5.3.2 explains how we warp a patch from a source image to a target image given a 3D scene represented by a geometry network predicting occupancy for each 3D point. Section 5.3.3 discusses questions related to visibility and how we mask invalid points during the optimization. Finally, Section 5.3.4 presents our full optimization. An overview of our approach and notations can be seen in Figure 5.2.

5.3.1 Volumetric rendering of radiance field

Neural volumetric rendering was introduced in Mildenhall et al. (2020) for novel view rendering. The idea is to represent the characteristics of a 3D scene with two implicit

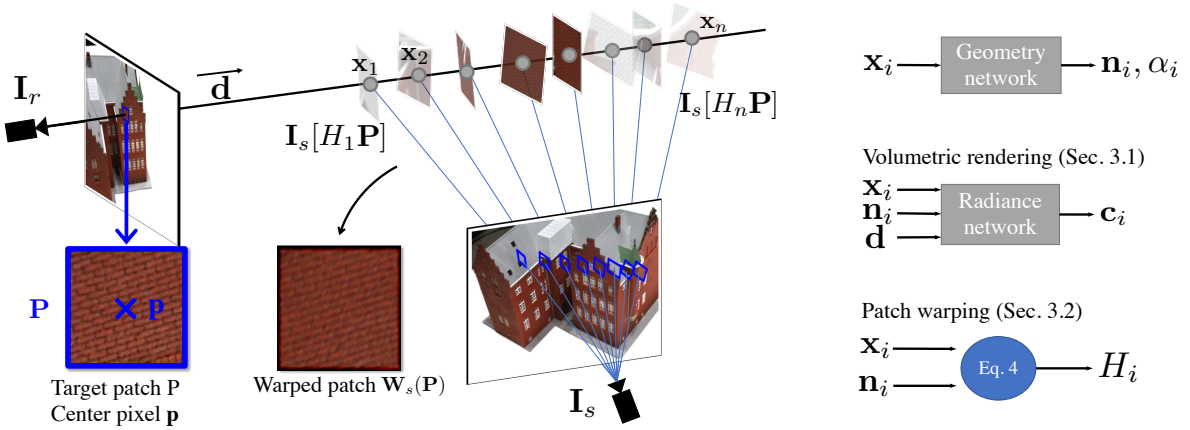


Fig. 5.2 Approach overview. We combine volumetric rendering with a new patch warping technique. Both approaches aggregate color from points sampled along the camera ray: radiance predicted by the radiance network for volumetric rendering and patch extracted from source views for our patch warping.

functions that are approximated with neural networks. The geometry network encodes the geometry of the scene as a Signed Distance Field (SDF) Wang et al. (2021a); Yariv et al. (2021). The radiance network encodes the color emitted by any region in space in all directions. The idea is to optimize jointly the two neural networks so that rendering the associated scene reconstructs a set of given views of the scene. Let us consider a reference image $\mathbf{I}_r$. The two networks are optimized using $\ell 1$ loss between the colors in reference image $\mathbf{I}_r[\mathbf{p}]$ and the volumetric rendering $R[\mathbf{p}]$ for a pixel $\mathbf{p}$:

$$\mathcal{L}_{\text{vol}} = \sum_{\mathbf{p}} |\mathbf{I}_r[\mathbf{p}] - \mathbf{R}[\mathbf{p}]|. \tag{5.1}$$

The rendered color $\mathbf{R}[\mathbf{p}]$ is computed from both networks in a differentiable way with respect to their parameters using volumetric rendering Kajiya and Von Herzen (1984). Let $\mathbf{x}_i, i = 1 \dots n$ be an ordered set of points sampled along the ray going through the reference camera center and the pixel $\mathbf{p}$[1]. The rendering of the scene at pixel $\mathbf{p}$ is approximated as a weighted sum of the color $\mathbf{c}_i = \mathbf{c}(\mathbf{x}_i, \mathbf{n}_i, \mathbf{d})$ predicted by the radiance network $\mathbf{c}$ for the direction $\mathbf{d}$ of the ray, at the points x_i for surface normals $\mathbf{n}_i$, computed by differentiation of the geometry network as the different positions x_i. The dependency on the viewing direction is the key factor to model view dependent effects and IDR Yariv et al. (2020) noted that although it seems redundant with $\mathbf{x}_i$, using $\mathbf{n}_i$ for computing the radiance leads to more robust results. The colors on the ray $\mathbf{c}_i, i = 1 \dots N$ are averaged

[1]For simplicity, we drop the dependency in the pixel to render $\mathbf{p}$ and the reference image index r in the point names.

using weights computed from the geometry network, α. For simplicity, we see the output of the geometry network outputs as an occupancy between 0 and 1. In practice, our geometry network outputs a SDF and we refer to Yariv et al. (2021) for a detailed explanation of the mapping of SDF to occupancy values. Intuitively, the color $\mathbf{c}_i$ will contribute to the rendering if $\mathbf{x}_i$ has a high occupancy and if no point on the ray between $\mathbf{x}_i$ and the reference camera has a high occupancy value. Formally, the rendered color can be approximated with an alpha blending formula:

$$\mathbf{R}[\mathbf{p}] = \sum_{i=1}^{N} \alpha_i \prod_{j<i} (1 - \alpha_j)\mathbf{c}_i, \tag{5.2}$$

where the $\alpha_i = \alpha(\mathbf{x}_i)$ are the occupancy values and $\mathbf{c}_i = \mathbf{c}(\mathbf{x}_i, \mathbf{n}_i, \mathbf{d})$ the radiance values predicted at the sampled points. This equation is the approximation of an integral along the camera ray. The choice of sampling points $\mathbf{x}_i$ is therefore a key element and is discussed in Mildenhall et al. (2020); Oechsle et al. (2021); Yariv et al. (2021). Those methods improve geometry estimation by improving the sampling, but the reconstructions are still worse than traditional MVS techniques. We hypothesize this comes from the difficulty for the radiance network to represent high frequency textures (see Fig. 3.1).

5.3.2 Warping images with implicit geometry

Instead of memorizing all the color information present in the scene with the radiance network, we propose to directly warp images onto each other relying only on the geometry network. We consider a reference image $\mathbf{I}_r$ and a source image $\mathbf{I}_s$. Similar to the above section, we want to obtain the color of a pixel $\mathbf{p}$ or a patch centered around $\mathbf{p}$ using the occupancies $\alpha_i, i = 1 \dots N$ from the geometry network but this time using colors from projections from the source image and not the one predicted by the radiance network. In this section, we assume these projections and their colors are well defined, and deal with the general case in Section 5.3.3. We start by explaining how a source image can be warped to a target image pixel-by-pixel, which we refer to as pixel warping. We then extend this idea to warping full patches, a classical idea in MVS Furukawa and Ponce (2009); Galliani et al. (2015); Schönberger et al. (2016); Zheng et al. (2014).

Pixel warping: Instead of using a radiance network to compute the color of each 3D point $\mathbf{x}_i$ on the ray associated to pixel $\mathbf{p}$, we use the color of their projection in source

image. Formally, we define the warped value of $\mathbf{p}$ from source image s as:

$$\mathbf{W}_s[\mathbf{p}] = \sum_{i=1}^{N} \alpha_i \prod_{j<i} (1-\alpha_j) \mathbf{I}_s[\pi_s(\mathbf{x}_i)], \tag{5.3}$$

where $\mathbf{I}_s[\pi_s(\mathbf{x})]$ denotes the bilinear interpolation of colors from $\mathbf{I}_s$ at the point $\pi_s(\mathbf{x})$ where the 3D point $\mathbf{x}$ projects in $\mathbf{I}_s$. In this section, we assume that every 3D points has a valid projection in source image so that $\mathbf{I}_s[\pi_s(\mathbf{x})]$ is always defined. Eq. (5.3) is similar to Eq. (5.2) but the color comes from pixel values in source images instead of network predictions. Intuitively, the warped value is a weighted average of the source image colors along the epipolar line. Similar to Eq. 5.1, one could optimize the geometry using a $\ell 1$ loss function $\ell = \sum_{\mathbf{p}} |\mathbf{W}_s[\mathbf{p}] - I_r[\mathbf{p}]|$. However, this does not model changes in intensity related to the cameras viewpoint, and in particular specularities, and can create artifacts in the reconstruction. One solution to this issue is to use a robust patch-based photometric loss function.

Patch warping: We now explain how to warp entire patches instead of single pixels, by locally approximating the scene at each point $\mathbf{x}_i$ as a plane in a way similar to standard techniques used in classical MVS Furukawa and Ponce (2009). Let $\mathbf{n}_i$ be the surface normal at $\mathbf{x}_i$, which can be computed with automatic differentiation of the geometry network at $\mathbf{x}_i$. Let H_i be the homography between $\mathbf{I}_r$ and $\mathbf{I}_s$ induced by the plane through $\mathbf{x}_i$ of normal $\mathbf{n}_i$. It can be computed as a 3×3 matrix acting on 2D homogeneous coordinates:

$$H_i = K_s \left(R_{rs} + \frac{\mathbf{t}_{rs} \mathbf{n}_i^T R_r^T}{\mathbf{n}_i^T (\mathbf{x}_i + R_r^T \mathbf{t}_r)} \right) K_r^{-1}, \tag{5.4}$$

where K_r and K_s are the internal calibration matrices of reference and source camera, $(R_{rs}, \mathbf{t}_{rs})$ is the relative motion from $\mathbf{I}_r$ to $\mathbf{I}_s$ represented with a 3×3 rotation matrix R_{rs} and a 3D translation vector $\mathbf{t}_{rs}$ and $(R_r, \mathbf{t}_r)$ is the pose of reference frame in world coordinates with the same representation. This homography associates any pixel $\mathbf{q}$ in $\mathbf{I}_r$ to a pixel $H\mathbf{q}$ in $\mathbf{I}_s$.

With a slight abuse we extend this notation to patches: for a patch $\mathbf{P}$ centered around pixel $\mathbf{p}$, we write $H_i\mathbf{P}$ the application of the homography to all pixels of the patch and $\mathbf{I}_s[H_i\mathbf{P}]$ the color interpolated at those locations in $\mathbf{I}_s$. Intuitively, $H_i\mathbf{P}$ is the location of a patch in source image that would correspond to $\mathbf{P}$ in reference frame if the true geometry was a plane of normal $\mathbf{n}_i$ passing through $\mathbf{x}_i$. We can now average the patches corresponding to each $\mathbf{x}_i$ in a manner similar to Eq. (5.2) to produce a warped patch

$\mathbf{W}_s[\mathbf{P}]$:

$$\mathbf{W}_s[\mathbf{P}] = \sum_{i=1}^{N} \alpha_i \prod_{j<i} (1-\alpha_j) \mathbf{I}_s[H_i \mathbf{P}]. \tag{5.5}$$

In all our experiments, we follow COLMAP Schönberger et al. (2016) and use a patch size of 11×11. Note that using a patch size of 1 would provide the same equation as Eq. (5.3).

5.3.3 Optimizing geometry from warped patches

We now want to define a loss to optimize geometry based on the warped patches. This cannot be done directly by using Eq. (5.5) to warp patches and maximizing SSIM. Indeed, we assumed that all 3D points $\mathbf{x}_i$ on the camera ray have a valid projection in the source image, but in practice this is not true for many points, which may for example project outside of the source image. In that case $\mathbf{I}_s[H_i\mathbf{P}]$ is not defined and we use instead a constant (grey) padding color in Eq. (5.5). This will of course affect the quality of the warped patch $\mathbf{W}_s[\mathbf{P}]$. Intuitively, if all non-valid 3D points on a ray are far from the implicit surface seen in the reference image, the padding value will contribute very little to the final warped patch which can be used in the loss. On the contrary, if there are invalid points near the implicit surface, the warped patch becomes invalid and should not be used in the loss. We formalize this intuition by associating to a patch $\mathbf{P}$ centered around pixel $\mathbf{p}$ in the reference image a mask value $M_s[\mathbf{P}] \in [0,1]$ for each source image s. In the rest of the section, we first explain how we define our loss for a given reference image based on the validity masks M_s associated with each source image. We then explain how we define the validity masks.

Warping-based loss: We start by selecting for a given reference image the set $\mathcal{V}$ of patches we consider in the loss. Since we want to discard patches $\mathbf{P}$ for which no source image gives any reasonable warp, as quantified by $M_s(\mathbf{P})$, we define it as $\mathcal{V} = \{\mathbf{P} : \sum_s M_s(\mathbf{P}) > \epsilon\}$. In practice, we use $\epsilon = 0.001$ in all of our experiments. We then define our warping based loss such that every valid patch in the reference image is given the same weight, but also such that invalid warpings are given less weight:

$$\mathcal{L}_{\text{warp}} = \sum_{\mathbf{P}\in\mathcal{V}} \frac{\sum_{s\in\mathcal{S}} M_s[\mathbf{P}]\, d(\mathbf{I}_r[\mathbf{P}], \mathbf{W}_s[\mathbf{P}])}{\sum_{s\in\mathcal{S}} M_s[\mathbf{P}]} \tag{5.6}$$

where $\mathbf{I}_r[\mathbf{P}]$ is the color patch $\mathbf{P}$ in $\mathbf{I}_r$ and d is a photometric distance between image patches. We use the SSIM Wang et al. (2004) except in our ablation where we use $\ell 1$ for pixel warping.

Validity masks: We now explain how we define the validity mask M_s. We consider two reasons for warps not to be valid, hence two masks: (i) a projection mask M_s^{proj} for cases in which the projection is not valid for geometric reasons, and (ii) an occlusion mask M_s^{occ} for cases where the patch is occluded by the reconstructed scene in the source image. The final mask is the product of both: $M_s[\mathbf{P}] = M_s^{\text{proj}}[\mathbf{P}]M_s^{\text{occ}}[\mathbf{P}]$.

To define the projection mask, we introduce a binary indicator V_i^s which is 0 when the projection associated with $\mathbf{x}_i$ is not valid and 1 otherwise. The projection can be invalid for three reasons: first, when the projection of the point in the source view is outside the source image; second, when the reference and source views are on two different sides of the plane defined by $\mathbf{x}_i$ and the normal $\mathbf{n}_i$; third when a camera center is too close to the plane defined by $\mathbf{x}_i$ and the normal $\mathbf{n}_i$, for which we use a threshold of 0.001 in practice. We obtain $M_s^{\text{proj}}[\mathbf{P}]$ by averaging the validity indicators for all 3D points sampled on the ray associated to $\mathbf{P}$ weighted by the α values:

$$M_s^{\text{proj}}[\mathbf{P}] = \sum_{i=1}^{N} \alpha_i \prod_{j<i} (1-\alpha_j) V_i^s \tag{5.7}$$

Note that (5.7) produces a soft mask value between 0 and 1, which is important to make it differentiable with respect to the α factors, which we found to be important in practice.

To define the occlusion mask M_s^{occ} we check whether there are occupied regions on the ray between the points $\mathbf{x}_i$ and the source camera center. We compute how occluded a 3D point $\mathbf{x}$ is with its transmittance in source view: $T_s(\mathbf{x}) = 1 - \prod_{k=1}^{N}(1-\alpha_k^s)$, where the α_k^s are the occupancy values predicted by the geometry network on 3D points sampled on the ray from $\mathbf{x}$ to the center of view s. Intuitively, $T_s(\mathbf{x})$ is close to 1 if there are no point with a large density between the source image and $\mathbf{x}$ and close to 0 in the opposite case.We could average the $T_s(\mathbf{x}_i), i = 1 \dots N$ in the same manner as Eq. (5.7) but this would require computing the transmittance T_s for every point on the ray. For computational efficiency we choose instead to compute an intersection point on the ray corresponding to the patch $\mathbf{P}$ in the reference view and evaluate transmittance in the source views on this point only:

$$M_s^{\text{occ}}[\mathbf{P}] = T_s\left(\sum_{i=1}^{N} \alpha_i \prod_{j<i} (1-\alpha_j)\mathbf{x}_i\right) \tag{5.8}$$

This mask is again soft because T_s outputs a continuous value between 0 and 1, which helps handling thin surfaces occlusion: if a ray comes very close to a surface without being strictly occluded, it will still have a lesser influence on the warping loss compared to a ray that is far from any surface.

5.3.4 Optimization details

We now explain the details of our method. We first present out full loss and optimization. We then detail the network architecture. We finally discuss how we selected the source images for each reference image.

Full optimization: We optimize the geometry and radiance networks to minimize the sum of the volumetric rendering loss (Eq. 5.1) and the patch warping loss (Eq. 5.6). In order to encourage the geometry network to output a function similar to a signed distance field, we also add the eikonal loss $\mathcal{L}_{\text{eik}}$ Gropp et al. (2020) which is minimum when the gradient of the output function at each point in space is of norm 1. This results in the following complete loss:

$$\mathcal{L} = \lambda_{\text{vol}}\mathcal{L}_{\text{vol}} + \lambda_{\text{warp}}\mathcal{L}_{\text{warp}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} \tag{5.9}$$

where λ_{vol}, λ_{warp} and λ_{eik} are scalar hyperparameters. We use $\lambda_{\text{vol}} = 1$ and $\lambda_{\text{eik}} = 0.1$ for all experiments. We first optimize the networks using $\lambda_{\text{warp}} = 0$ in the same setting as VolSDF. After $100k$ iterations with a learning rate exponentially decayed from 5e$-$4 to 5e$-$5, we finetune for another $50k$ iterations using $\lambda_{\text{warp}} = 1$ and a fixed learning rate of 1e$-$5. The networks are initialized with the sphere initialization of Atzmon and Lipman (2020). We start optimizing with volumetric rendering only because the normals are initially too different from the scene ones to compute meaningful homographies. During the first phase of training, without patch warping, we train with batches of 1024 pixels but we finetuned with patch warping on batches of 512 patches due to GPU memory constraints. Also, we do not backpropagate the loss through the homography parameters in equation (5.4) since we noticed it leads to unstable optimization.

Architecture: We use the same architecture as concurrent works Oechsle et al. (2021); Yariv et al. (2021). Both radiance and geometry network are Multi-Layer Perceptrons (MLP). The geometry network has 8 layers with 256 hidden units. The radiance network has 4 layers with 256 hidden units. Similar to Oechsle et al. (2021); Yariv et al. (2021,

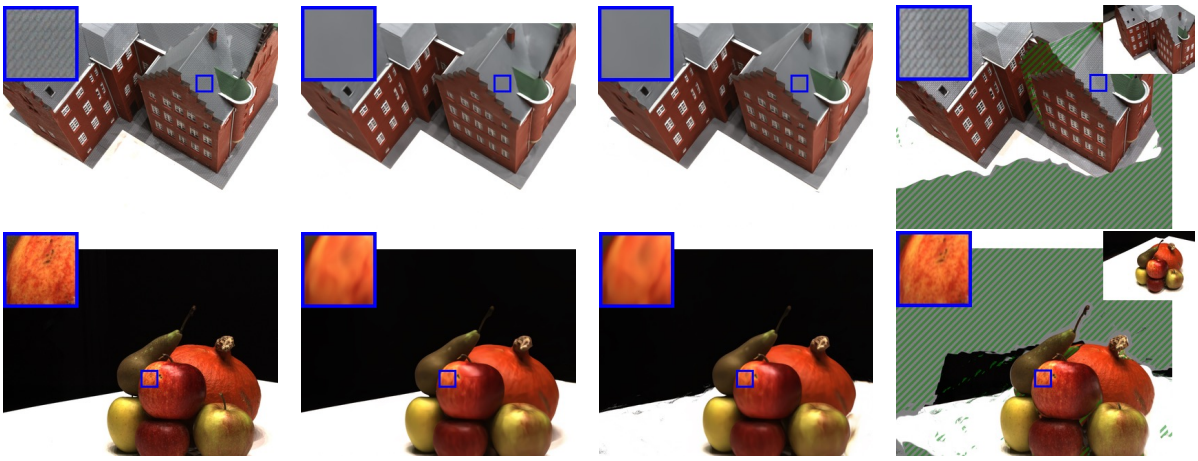
Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
IDR	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
MVSDF*	0.83	1.76	0.88	0.44	1.11	0.90	0.75	1.26	1.02	1.35	0.87	0.84	0.34	0.47	0.46	0.88
COLMAP	0.45	0.91	0.37	0.37	0.90	1.00	0.54	1.22	1.08	0.64	0.48	0.59	0.32	0.45	0.43	0.65
NeRF	1.90	1.60	1.85	0.58	2.28	1.27	1.47	1.67	2.05	1.07	0.88	2.53	1.06	1.15	0.96	1.49
UNISURF	1.32	1.36	1.72	0.44	1.35	0.79	0.80	1.49	1.37	0.89	**0.59**	1.47	0.46	0.59	0.62	1.02
NeuS	1.37	1.21	0.73	**0.40**	1.20	**0.70**	**0.72**	**1.01**	1.16	0.82	0.66	1.69	**0.39**	**0.49**	**0.51**	0.87
VolSDF	1.14	1.26	0.81	0.49	1.25	**0.70**	**0.72**	1.29	1.18	**0.70**	0.66	1.08	0.42	0.61	0.55	0.86
Ours	**0.53**	**0.90**	**0.45**	0.52	**0.95**	0.85	0.87	1.26	**1.00**	0.73	0.62	**0.72**	0.41	0.63	**0.51**	**0.73**

Table 5.1 Quantitative comparison on DTU. Note that MVSDF results are not directly comparable since they are the raw output while results with all other methods have been cleaned using the visual hull. The bottom part of the table compares our result with neural implicit surfaces approaches that do not use additional inputs (masks, supervised depth estimation). **Bold** results have the best score and underlined the second best. Our method outperforms existing work by a large margin, more than 15% on the mean metrics.

2020) we encode 3D position using positional encoding with 6 frequencies and viewing direction with 4 frequencies.

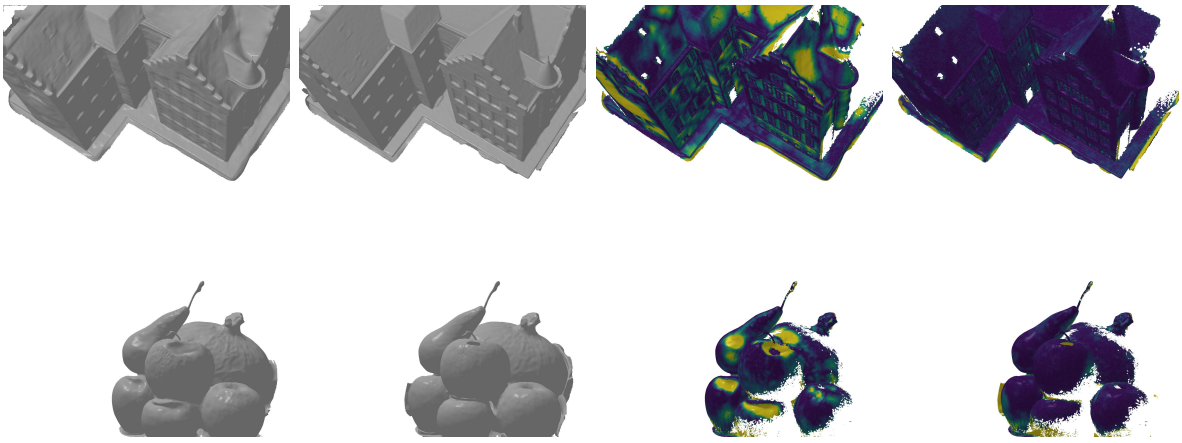
Rays sampling: We follow VolSDF Yariv et al. (2021) for choosing the points $\mathbf{x}_1 \dots \mathbf{x}_N$ on the camera rays but with a small modification. We first estimate the opacity function with the algorithm introduced by VolSDF, we then sample $N = 64$ points on the camera ray with 90% sampled from the Opacity distribution as in VolSDF but the remaining 10% sampled uniformly along the whole camera ray.

Choice of source images: Our method uses a set of 19 source images $\mathcal{S}$ following COLMAP Schönberger et al. (2016) for each reference image. Those source images must be carefully chosen since very similar viewpoints will carry little geometric information and very different viewpoints will have few common points. We first build a sparse point cloud with a Structure from Motion software Schonberger and Frahm (2016), we then compute for each image pair the total number of sparse points observed by both (co-visible points) and remove the pairs for which more than 75% of the co-visible points are observed with a triangulation angle below 5° . We finally select the 19 top views in number of co-visible points.



Ground truth image | VolSDF Yariv et al. (2021) volume rendering | Our volume rendering using our radiance network | Ours warping from a source image and validity mask in green

Fig. 5.3 Examples of DTU renderings using VolSDF Yariv et al. (2021) and our method. For both methods, volumetric rendering cannot produce high frequency texture whereas image warping can precisely recover them. (better viewed in electronic version)



VolSDF mesh | Ours mesh | VolSDF errormap | Ours errormap

Fig. 5.4 Rendering and geometric error maps of two DTU scenes Jensen et al. (2014) (blue=low, yellow=high). Compared to VolSDF on which we build, our method significantly improves accuracy.

5.4 Experiments

In this section, we first show that our method outperforms state-of-the-art unsupervised neural implicit surface approaches on the DTU dataset Jensen et al. (2014), then on the

EPFL benchmark Strecha et al. (2008), we present an ablation study to evaluate each of our technical contributions and finally we discuss the limitations of our method.

DTU benchmark: The DTU benchmark Jensen et al. (2014) includes scenes with 49 to 64 images associated to reference point clouds acquired with laser sensor. Each scene covers a different object: some have challenging specular materials while others have large textureless regions. The evaluation of IDR Yariv et al. (2020) selected 15 scenes and manually annotated object masks. We compare our method with existing work on the same scenes, using the evaluation code provided by the authors. The metric is the average of accuracy and completeness: the chamfer distance of prediction to reference point cloud and inversely. Similar to existing work Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021, 2020), we clean the output meshes with the visibility masks dilated by 50 pixels.

We compare with multiple baselines in Table 5.1. The results for each methods are taken from there original paper, except COLMAP, NeRF and IDR that we took from Oechsle et al. (2021). Similar to Oechsle et al. (2021); Wang et al. (2021a); Yariv et al. (2021), we only compare in the bottom part of the Table deep implicit surfaces approaches that do not use masks or other data during training. In particular MVSDF Zhang et al. (2021) uses a supervised depth estimation network. Our method outperforms existing methods by a large margin. As could be expected, the improvement is more important on highly textured scenes but our method performs on par with other methods on weakly textured scenes. Fig. 5.3 compares original images, volume rendering obtained by VolSDF, volume rendering obtained with out radiance network and our image warping (using the pixel warping approach described in Section 5.3.2). Volumetric rendering only renders smoothed texture, whereas our warpings is able to render high frequency texture information. As can be seen qualitatively in Figure 3.1 and 5.4, this leads to important improvements in accuracy.

EPFL benchmark: The EPFL benchmark Strecha et al. (2008) consists of two outdoor scenes of 7 and 11 high resolution images with a high resolution ground truth mesh. Since the extent of the ground truth does not exactly overlap the cameras viewing angle and inversely, it is necessary to remove points from both predicted and ground truth mesh for evaluation. MVSDF Zhang et al. (2021) uses manual masks to remove vertices from the ground truth mesh, which we argue might be biased. We instead automatically remove vertices from the ground truth mesh when they do not project in any input image. Similar to DTU evaluation, we use silhouette masks to clean the predicted mesh

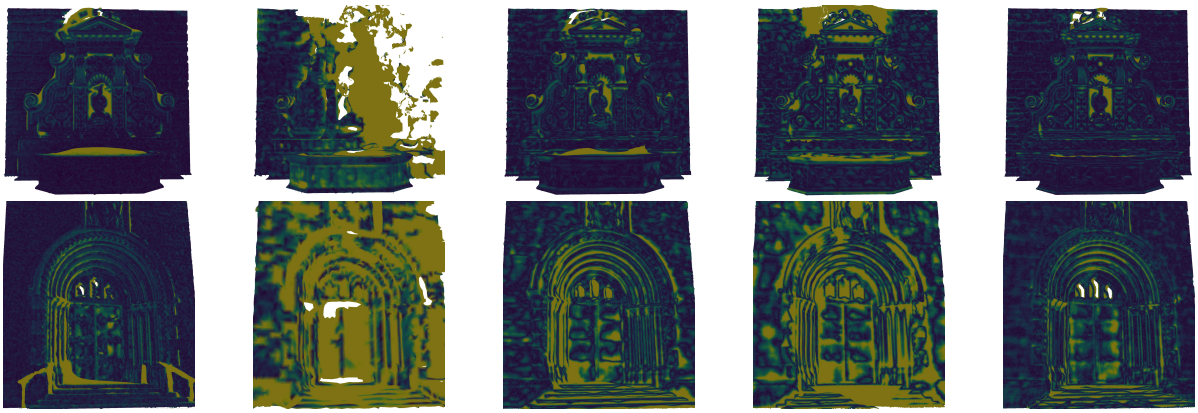


Fig. 5.5 Geometric error maps of reconstructed meshes on EPFL benchmark Strecha et al. (2008) (blue=low, yellow=high). Our method produces better results than other neural implicit surface approaches, almost on par with COLMAP Schönberger et al. (2016).

with the scene visual hull. We generate silhouette masks by rendering the ground truth mesh on each input viewpoint and marking pixels which are not covered as outside of the silhouette. Finally, we also remove from the predicted mesh any triangles that are not rendered in any of the images, which removes in particular faces closing the volume behind the object. To compute the distance between the filtered ground truth and predicted mesh, we sample 1 million point from each and compute their chamfer distance. We call this metrics the *full* chamfer distance. It is mainly influenced by the completeness of the reconstruction, e.g. it compares how well methods reconstruct the ground plane or rarely seen points. We therefore introduce another metric referred to as the *center* chamfer distance which only evaluates the chamfer distance in a box at the center of the scene which we manually defined so that it only includes the central part of the scene, which is reconstructed by all methods. This metric thus focuses more on the accuracy of the reconstruction.

We compare our method with several baselines with these two metrics in Table 5.2. We ran COLMAP followed by sPSR Kazhdan and Hoppe (2013) with trim 5, used the official UNISURF implementation [2], evaluated the MVSDF meshes communicated by the authors and ran our own reimplementation of VolSDF. Qualitative comparisons between the reconstructions and error maps for each method can be seen in Figure 5.5. Similar to the DTU results, our method outperforms other neural implicit surfaces by more than 15%. COLMAP Schönberger et al. (2016) is the best method for the full metric.

[2] https://github.com/autonomousvision/unisurf

Method	Fountain-P11		Herzjesu-P7		**Mean**	
	Full	Center	Full	Center	**Full**	**Center**
COLMAP	**6.47**	2.45	7.95	2.31	**7.21**	2.38
UNISURF	26.16	17.72	27.22	13.72	26.69	15.72
MVSDF	6.87	2.26	11.32	2.72	9.10	2.49
VolSDF*	12.89	2.99	13.61	4.58	13.25	3.78
Ours	8.51	**1.98**	**6.80**	**1.95**	7.66	**1.96**

Table 5.2 Quantitative evaluation on EPFL dataset. We use the chamfer distance on the Full scene (Full) and on a manually defined bounding box at the center of the scene (Center). Results of VolSDF Yariv et al. (2021) come from our implementation. Although COLMAP is the best method for the full reconstruction, our method has the best results for the center metrics. It outperforms existing neural implicit surfaces by 20% on center metrics.

Method	$\mathcal{L}_{\text{vol}}$	$\mathcal{L}_{\text{warp}}$	M_s^{occ}	Chamfer dist.
VolSDF	✓	None		0.87
Pixel	✓	Pixel	✓	0.83
Patch no occ.	✓	Patch		0.79
Patch no vol.		Patch	✓	0.79
Ours full	✓	Patch	✓	**0.73**

Table 5.3 Ablation study on all scenes of DTU: $\mathcal{L}_{\text{vol}}$ denotes volumetric rendering, $\mathcal{L}_{\text{warp}}$ warping consistency for which we try none, pixel and patch warpings. M_s^{occ} denotes whether we detect self occlusions or not.

This is in large part because it is the only method able to reconstruct accurately the ground plane on both scenes. For the center metrics however, our method outperforms even COLMAP, but this might be due to some details reconstructed by COLMAP (e.g. railing) not being included in the ground truth and COLMAP still qualitatively seems to recover finer details.

Ablation study: To evaluate the effect of our technical contributions, we perform an ablation study on the DTU dataset. Starting from the same models trained without photometric consistency, we finetune different versions of our model for 50000 iterations and compare the results. The average chamfer distance over all 15 scenes is shown in Table 5.3. We first compare the results without our warping loss ('VolSDF Yariv et al. (2021)' line), with pixel warping ('Pixel' line) and with patch warping ('Ours full' line). Both pixel and patch warping improve the results, with a clear advantage for patches. We then evaluate the importance of masking. Removing the projection mask (not reported in the table) does not lead to meaningful reconstructions. Without the occlusion mask

('Patch no occ.' line) our method still improves over the baseline, but by a smaller margin. Finally, we tried to remove the volumetric rendering loss completely, i.e. use $\lambda_{\text{vol}} = 0$ ('Patch no vol.' line). Again, this improves over the baseline, but is worse than combining the volumetric and warp losses.

Limitations: Our method has several limitations. First, compared to COLMAP it struggles to reconstruct high frequency geometry. We believe this is due to the difficulty of optimizing a geometry network at high resolution. Second, computing our loss increase the computational cost of the optimization, in particular the occlusion mask adds processing time and processing patches increases memory footprint. Finally simply comparing patches does not model reflections, which can lead to artefacts even using a robust patch similarity.

5.5 Conclusion

We have presented a new method to perform multiview reconstruction with implicit functions, using image warpings in combination with volumetric rendering. Unlike existing neural implicit surface methods, our approach can easily take advantage of high frequency texture. We show that this leads to strong performance improvements on the classical DTU and EPFL datasets.

Chapter 6

Conclusion

In this chapter, we summarize the contributions presented in the thesis. We then discuss a few research directions opened by this work.

6.1 Summary of contributions

- In Chapter 3 we have studied data and supervision applied to image matching. We have introduced a two-step method that first produces low resolution matches using a Deep Learning Network then use those matches for guiding local feature matching. We have shown that weak supervision from epipolar geometry provides useful information for training the network and that our method is able to match images in challenging conditions like day-night or historical data.

- We then focused on the calibrated dense reconstruction scenario. In Chapter 4 we have studied MVS network in the challenging setup of "in the wild images". We introduce a new unsupervised method that works on such data, contrary to existing work. We have performed a detailed analysis of the application of MVS networks to this hard setup and we have shown that Deep Learning methods are on par with classical methods.

- In Chapter 5 we still focused on the dense reconstruction problem but using a different 3D parameterization with better theoritical properties. We develop a new Neural Implicit Surface method which is much more accurate than existing works. Our method relies on an added patch-based photo-consistency that better constrains the optimization. This approach shows very promising results and it opens many research directions for Neural Implicit MVS.

6.2 Future works

- We have developed weakly supervised methods that work on very challenging data for two 3D tasks of the typical 3D reconstruction pipeline. They allow to use much larger datasets without the need to manually label the data or to use specific acquisition hardware. A promising research direction is therefore to create large-scale datasets with very diverse data and train unsupervised networks to get the best robustness. Using much bigger datasets is a common trend in recent computer vision with the use of transformers architectures Vaswani et al. (2017) that require more and more training data Carion et al. (2020); Dosovitskiy et al. (2021). Such architectures were already adapted to image matching Sarlin et al. (2020); Sun et al. (2021) but they could be used also for other 3D tasks.

- We have improved Neural Implicit methods to a point close to classical MVS methods on DTU dataset Jensen et al. (2014). An obvious continuation would be to try harder reconstructions on the same data as the one of chapter 4. This would require to deal with moving objects and very different imaging conditions similar to Martin-Brualla et al. (2021). Another research direction is to work on the implicit formulation itself. As shown in Yu et al. (2022), when using additional regularization, Neural Rendering method can be optimized without any Neural Network. The only requirement is to have a parametric function differentiable with respect to its parameters. It would therefore be interesting to study both in theory and in practice the properties of implicit parameterizations and to derive an optimal one in term of optimization speed and representative power. If such an approach becomes state-of-the-art, it might be interesting to also improve the mesh extraction procedure. Marching Cubes Lorensen and Cline (1987) work on a regular grid, but we could instead adapt the vertex density and optimize the mesh so that the vertices and normals correspond perfectly to the surface encoded in the implicit function.

References

Atzmon, M., Haim, N., Yariv, L., Israelov, O., Maron, H., and Lipman, Y. (2019). Controlling neural level sets. In *Advances in Neural Information Processing Systems*.

Atzmon, M. and Lipman, Y. (2020). Sal: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Badino, H., Huber, D., and Kanade, T. (2011). The CMU Visual Localization Data Set. http://3dvis.ri.cmu.edu/data-sets/localization.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *Communications of the ACM*, 28(3):24.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer.

Beder, C. and Steffen, R. (2006). Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Joint Pattern Recognition Symposium*, pages 657–666. Springer.

Bergman, A. W., Kellnhofer, P., and Wetzstein, G. (2021). Fast training of neural lumigraph representations using meta learning. In *Advances in Neural Information Processing Systems*.

Bian, J., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., and Cheng, M.-M. (2017). GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4181–4190.

Bian, J.-W., Wu, Y.-H., Zhao, J., Liu, Y., Zhang, L., Cheng, M.-M., and Reid, I. (2019). An evaluation of feature matchers for fundamental matrix estimation. In *British Machine Vision Conference*.

Brachmann, E. and Rother, C. (2019). Neural-guided RANSAC: Learning where to sample model hypotheses. In *International Conference on Computer Vision*, pages 4322–4331.

Brown, D. C. (1958). *A solution to the general problem of multiple station analytical stereotriangulation*. D. Brown Associates, Incorporated.

Bujnak, M., Kukelova, Z., and Pajdla, T. (2008). A general solution to the p4p problem for camera with unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer.

Campbell, N. D., Vogiatzis, G., Hernández, C., and Cipolla, R. (2008). Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, pages 766–779. Springer.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chang, J.-R. and Chen, Y.-S. (2018). Pyramid stereo matching network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418.

Chatterjee, A. and Govindu, V. M. (2013). Efficient and robust large-scale rotation averaging. In *International Conference on Computer Vision*, pages 521–528.

Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., and Su, H. (2021). Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *International Conference on Computer Vision*.

Chen, R., Han, S., Xu, J., et al. (2020). Visibility-aware point-based multi-view stereo network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, R., Han, S., Xu, J., and Su, H. (2019). Point-based multi-view stereo network. In *International Conference on Computer Vision*.

Chibane, J., Bansal, A., Lazova, V., and Pons-Moll, G. (2021). Stereo radiance fields (srf): Learning view synthesis from sparse views of novel scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer.

Chum, O. and Matas, J. (2005). Matching with prosac-progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 220–226. IEEE.

Chum, O., Werner, T., and Matas, J. (2005). Two-view geometry estimation unaffected by a dominant plane. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–779. IEEE.

Dai, Y., Zhu, Z., Rao, Z., and Li, B. (2019). Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry. In *International Conference on 3D Vision*.

Darmon, F. and Monasse, P. (2021). The polar epipolar rectification. *Image Processing On Line*, 11:56–75.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

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. In *International Conference on Computer Vision*, pages 2758–2766.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., and Sattler, T. (2019). D2-net: A trainable CNN for joint description and detection of local features. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ebel, P., Mishchuk, A., Yi, K. M., Fua, P., and Trulls, E. (2019). Beyond cartesian representations for local descriptors. In *International Conference on Computer Vision*, pages 253–262.

Faugeras, O. and Keriven, R. (1998). Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions Image Processing*.

Feng, T. and Yuan, J. (2011). Feature point detection and matching of wide baseline image based on scale space theory and guided matching algorithm. In *International Conference on Multimedia Technology*, pages 538–542. IEEE.

Fernando, B., Tommasi, T., and Tuytelaars, T. (2015). Location recognition over large time lags. *Computer Vision and Image Understanding*, 139.

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*.

Furukawa, Y. and Hernández, C. (2015). Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148.

Furukawa, Y. and Ponce, J. (2009). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8).

Galliani, S., Lasinger, K., and Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *International Conference on Computer Vision*.

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q., and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y. (2020). Implicit geometric regularization for learning shapes. In *Proc. of Machine Learning and Systems*.

Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., and Tan, P. (2020). Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Harris, C., Stephens, M., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, pages 10–5244.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *International Conference on Computer Vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

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). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3287–3295.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*.

Huang, B., Huang, C., He, Y., Liu, J., and Liu, X. (2020). M^{3}VSNet: Unsupervised multi-metric multi-view stereo network. *arXiv preprint arXiv:2005.00363*.

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. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., and Aanæs, H. (2014). Large scale multi-view stereopsis evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ji, M., Gall, J., Zheng, H., Liu, Y., and Fang, L. (2017). Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *International Conference on Computer Vision*.

Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K. M., and Trulls, E. (2020). Image matching across wide baselines: From paper to practice. *International Journal on Computer Vision*, 129(2):517–547.

Kahl, F. (2005). Multiple view geometry and the l1/2-norm. In *International Conference on Computer Vision*.

Kajiya, J. T. and Von Herzen, B. P. (1984). Ray tracing volume densities. *ACM Transactions on Graphics*.

Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics*.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., and Bry, A. (2017). End-to-end learning of geometry and context for deep stereo regression. In *International Conference on Computer Vision*, pages 66–75.

Khot, T., Agrawal, S., Tulsiani, S., Mertz, C., Lucey, S., and Hebert, M. (2019). Learning unsupervised multi-view stereopsis via robust photometric consistency. *CVPR workshop*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Knapitsch, A., Park, J., Zhou, Q.-Y., and Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105.

Kutulakos, K. N. and Seitz, S. M. (2000). A theory of shape by space carving. *International Journal on Computer Vision*, 38(3):199–218.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International Journal on Computer Vision*, 81(2):155.

Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F. X., Taylor, R. H., and Unberath, M. (2021). Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *International Conference on Computer Vision*, pages 6197–6206.

Li, Z. and Snavely, N. (2018). Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050.

Lin, W.-Y., Wang, F., Cheng, M.-M., Yeung, S.-K., Torr, P. H., Do, M. N., and Lu, J. (2017). CODE: Coherence based decision boundaries for feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):34–47.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal on Computer Vision*, 30(2):79–116.

Liu, C., Yuen, J., and Torralba, A. (2010). Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994.

Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131. IEEE.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *ACM Transactions on Graphics*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

Luo, K., Guan, T., Ju, L., Huang, H., and Luo, Y. (2019a). P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *International Conference on Computer Vision*, pages 10452–10461.

Luo, Z., Shen, T., Zhou, L., Zhang, J., Yao, Y., Li, S., Fang, T., and Quan, L. (2019b). Contextdesc: Local descriptor augmentation with cross-modality context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2527–2536.

Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., and Quan, L. (2018). Geodesc: Learning local descriptors by integrating geometry constraints. In *European Conference on Computer Vision*, pages 168–183.

Ma, J., Zhao, J., Jiang, J., Zhou, H., and Guo, X. (2019). Locality preserving matching. *International Journal on Computer Vision*, 127(5):512–531.

Maier, J., Humenberger, M., Murschitz, M., Zendel, O., and Vincze, M. (2016). Guided matching based on statistical optical flow for fast and robust correspondence analysis. In *European Conference on Computer Vision*, pages 101–117. Springer.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S. M., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Martinec, D. and Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Vision and Computer Graphics*, 1(2):99–108.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal on Computer Vision*, 60(1):63–86.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*.

Mishchuk, A., Mishkin, D., Radenovic, F., and Matas, J. (2017). Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837.

Mishkin, D., Radenovic, F., and Matas, J. (2018). Repeatability is not enough: Learning affine regions via discriminability. In *European Conference on Computer Vision*, pages 284–300.

Moo Yi, K., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., and Fua, P. (2018). Learning to find good correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674.

Moulon, P., Monasse, P., and Marlet, R. (2013). Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision*, pages 3248–3255.

Moulon, P., Monasse, P., Perrot, R., and Marlet, R. (2016). OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer.

Murez, Z., van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., and Rabinovich, A. (2020). Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conference on Computer Vision*.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. (2020). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Oechsle, M., Peng, S., and Geiger, A. (2021). UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *International Conference on Computer Vision*.

Ono, Y., Trulls, E., Fua, P., and Yi, K. M. (2018). LF-Net: Learning local features from images. In *Advances in Neural Information Processing Systems*, pages 6234–6244.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In *European Conference on Computer Vision*.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Pollefeys, M., Koch, R., and Van Gool, L. (1999). A simple and efficient rectification method for general motion. In *International Conference on Computer Vision*, pages 496–501. IEEE.

Pons, J.-P., Keriven, R., and Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal on Computer Vision*, 72(2):179–193.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.

Ranftl, R. and Koltun, V. (2018). Deep fundamental matrix estimation. In *European Conference on Computer Vision*, pages 284–299.

Ranjan, A. and Black, M. J. (2017). Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Revaud, J., Weinzaepfel, P., De Souza, C., and Humenberger, M. (2019). R2D2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, pages 12405–12415.

Rocco, I., Arandjelovic, R., and Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6148–6157.

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., and Sivic, J. (2018). Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems*, pages 1651–1662.

Sameer, A., Noah, S., Seitz, S., and Szeliski, R. (2010). Bundle adjustment in the large. In *European Conference on Computer Vision*, pages 29–42.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4938–4947.

Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al. (2018). Benchmarking 6DOF outdoor visual localization in changing conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47(1):7–42.

Schonberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schönberger, J. L., Zheng, E., Frahm, J.-M., and Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518.

Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., and Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528. IEEE.

Seitz, S. M. and Dyer, C. R. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal on Computer Vision*, pages 151–173.

Shah, R., Srivastava, V., and Narayanan, P. (2015). Geometry-aware feature matching for structure from motion applications. In *IEEE Winter Conference on Applications of Computer Vision*, pages 278–285. IEEE.

Shen, X., Darmon, F., Efros, A. A., and Aubry, M. (2020). Ransac-flow: generic two-stage image alignment. In *16th European Conference on Computer Vision*.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Sinha, A., Murez, Z., Bartolozzi, J., Badrinarayanan, V., and Rabinovich, A. (2020). Deltas: Depth estimation by learning triangulation and densification of sparse points. In *European Conference on Computer Vision*.

Snavely, N. (2008). Bundler: Structure from motion (SfM) for unordered image collections. *http://phototour.cs.washington.edu/bundler/*.

Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943.

Sun, J., Shen, Z., Wang, Y., Bao, H., and Zhou, X. (2021). Loftr: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8922–8931.

Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., and Yi, K. M. (2019). Attentive context normalization for robust permutation-equivariant learning.

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., and Torii, A. (2019). Inloc: Indoor visual localization with dense matching and view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Teed, Z. and Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

Tieleman, T., Hinton, G., et al. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Tola, E., Lepetit, V., and Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830.

Trevithick, A. and Yang, B. (2021). Grf: Learning a general radiance field for 3d representation and rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer.

Truong, P., Danelljan, M., and Timofte, R. (2020). Glu-net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6268.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344.

Tyszkiewicz, M. J., Fua, P., and Trulls, E. (2020). Disk: learning local features with policy gradient. In *Advances in Neural Information Processing Systems*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vu, H.-H., Labatut, P., Pons, J.-P., and Keriven, R. (2011). High accuracy and visibility-consistent dense multiview stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 889–901.

Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., and Wang, W. (2021a). NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*.

Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J. T., Martin-Brualla, R., Snavely, N., and Funkhouser, T. (2021b). Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. In *IEEE Transactions Image Processing*.

Widya, A. R., Torii, A., and Okutomi, M. (2018). Structure from motion using dense CNN features with keypoint relocalization. *IPSJ Transactions on Computer Vision and Applications*, 10(1):6.

Wilson, K. and Snavely, N. (2014). Robust global translations with 1DSFM. In *European Conference on Computer Vision*, pages 61–75. Springer.

Wu, C., Agarwal, S., Curless, B., and Seitz, S. M. (2011a). Multicore bundle adjustment. In *CVPR 2011*, pages 3057–3064. IEEE.

Wu, C. et al. (2011b). VisualSFM: A visual structure from motion system.

Xiao, J., Owens, A., and Torralba, A. (2013). Sun3d: A database of big spaces reconstructed using SfM and object labels. In *International Conference on Computer Vision*, pages 1625–1632.

Xu, H. and Zhang, J. (2020). Aanet: Adaptive aggregation network for efficient stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1959–1968.

Xu, H., Zhou, Z., Qiao, Y., Kang, W., and Wu, Q. (2021). Self-supervised multi-view stereo via effective co-segmentation and data-augmentation. In *AAAI*.

Xue, Y., Chen, J., Wan, W., Huang, Y., Yu, C., Li, T., and Bao, J. (2019). MVSCRF: Learning multi-view stereo with conditional random fields. In *International Conference on Computer Vision*, pages 4312–4321.

Yan, J., Wei, Z., Yi, H., Ding, M., Zhang, R., Chen, Y., Wang, G., and Tai, Y.-W. (2020). Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision*, pages 674–689. Springer.

Yang, J., Mao, W., Alvarez, J. M., and Liu, M. (2020). Cost volume pyramid based depth inference for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yao, Y., Luo, Z., Li, S., Fang, T., and Quan, L. (2018). MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision*.

Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., and Quan, L. (2019). Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., and Quan, L. (2020a). BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799.

Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., and Quan, L. (2020b). BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yariv, L., Gu, J., Kasten, Y., and Lipman, Y. (2021). Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*.

Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., and Lipman, Y. (2020). Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems*.

Yi, H., Wei, Z., Ding, M., Zhang, R., Chen, Y., Wang, G., and Tai, Y.-W. (2020). Pyramid multi-view stereo net with self-adaptive view aggregation. In *European Conference on Computer Vision*, pages 766–782. Springer.

Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). LIFT: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer.

Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., and Kanazawa, A. (2022). Plenoxels: Radiance fields without neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021). pixelNeRF: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yu, Z. and Gao, S. (2020). Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and Gauss-Newton refinement. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1949–1958.

Zach, C., Klopschitz, M., and Pollefeys, M. (2010). Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433. IEEE.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361.

Zbontar, J. and LeCun, Y. (2015). Computing the stereo matching cost with a convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599.

Zhang, F., Prisacariu, V., Yang, R., and Torr, P. H. (2019a). Ga-net: Guided aggregation net for end-to-end stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194.

Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., and Liao, H. (2019b). Learning two-view correspondences and geometry using order-aware network. In *International Conference on Computer Vision*, pages 5845–5854.

Zhang, J., Yao, Y., Li, S., Luo, Z., and Fang, T. (2020). Visibility-aware multi-view stereo network. In *British Machine Vision Conference*.

Zhang, J., Yao, Y., and Quan, L. (2021). Learning signed distance field for multi-view surface reconstruction. In *International Conference on Computer Vision*.

Zheng, E., Dunn, E., Jojic, V., and Frahm, J.-M. (2014). PatchMatch based joint view selection and depthmap estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.