TESIS DE LA UNIVERSIDAD DE ZARAGOZA
2026
24

Sergio Izquierdo Barranco

# Deep Spatial Perception: Localization & Reconstruction

Director/es
Civera Sancho, Javier



ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza
1542

Tesis Doctoral

# DEEP SPATIAL PERCEPTION: LOCALIZATION & RECONSTRUCTION


Autor

Sergio Izquierdo Barranco

Director/es

Civera Sancho, Javier



**UNIVERSIDAD DE ZARAGOZA**
**Escuela de Doctorado**

2025

Universidad Zaragoza
1474

# Tesis Doctoral

## Deep Spatial Perception: Localization & Reconstruction


Autor/a

Sergio Izquierdo Barranco

Director/a

Javier Civera Sancho


Programa de Doctorado en
Ingeniería de Sistemas e Informática
**Escuela de Doctorado**

2025

A doctoral thesis on Computer Science and Systems

# Deep Spatial Perception: Localization & Reconstruction

Author:
Sergio Izquierdo

Supervised by:
Javier Civera

September 10, 2025


Universidad Zaragoza

No por mucho madrugar amanece más temprano.

*No matter how early you get up, sunrise won't be any sooner.*

– Spanish proverb

# Abstract

Determining the location of an agent and representing its surroundings are two essential capabilities for the successful deployment of intelligent systems with spatial awareness. Commonly referred to as localization and reconstruction or mapping, these tasks have been studied for decades in the computer vision community, as many applications—such as augmented reality and robotics—rely heavily on them to operate coherently within the physical world.

One of the first steps in localizing an agent is retrieving its coarse location, for which Visual Place Recognition (VPR) provides an effective solution when a database of georeferenced images is available. A key challenge in VPR lies in designing compact, informative, and discriminative descriptors that remain robust under strong viewpoint changes, structural variations, and lighting conditions. In this context, the first part of this thesis proposes two complementary directions to advance VPR. First, we introduce a novel feature aggregation method based on optimal transport, paired with a powerful vision transformer backbone, to produce more robust image descriptors. Second, we propose a new training strategy that enhances the geographic sensitivity of these descriptors by selecting hard training samples based on both visual similarity and spatial distance. Together, these contributions advance towards effective, large-scale, and general VPR pipelines, significantly improving metrics at popular benchmarks, like MSLS Challenge, where we improved recall@1 from 67.4% to 82.7% and Nordland, from 58.4% to 90.7%.

Within the broader task of scene reconstruction or mapping, monocular depth estimation is one of the core pieces. While it is well understood how multiple views naturally provide geometric cues to resolve ambiguities and improve accuracy, the enduring question is how to design methods that can robustly exploit this information across diverse scenarios in a general-purpose manner. The second part of this thesis proposes two novel methods for leveraging multi-view constraints for depth estimation. First, we introduce a test-time refinement method that uses sparse 3D points from Structure-from-Motion to guide single-view depth networks during inference. This preserves the learned priors of single-view depth networks while injecting additional multi-view constraints. Second, we propose a general-purpose multi-view stereo architecture designed to operate robustly across diverse environments and depth scales. Our contributions focus on versatility, training on multiple datasets, addressing low overlap and dynamic objects, and removing restrictions like a priori depth range knowledge. Together, these contributions demonstrate the potential of combining learned priors with geometric constraints, showing promising steps towards a seamless integration of multi-view information in depth estimation. More precisely, our proposed refinement improved all considered single-view depth models, and our general-purpose multi-view stereo system obtained state-of-the-art results on the Robust Multi-View Depth Benchmark.

# Resumen

Determinar la ubicación de un agente y saber representar su entorno son dos capacidades esenciales para el correcto funcionamiento de sistemas inteligentes con conocimiento espacial. Estas tareas, conocidas comúnmente como localización y reconstrucción o mapeado, han sido estudiadas durante décadas en la comunidad de visión por computador, ya que muchas aplicaciones, como la realidad aumentada o la robótica, dependen en gran medida de ellas para interactuar de forma coherente en el mundo físico.

Uno de los primeros pasos en la localización de un agente es obtener una estimación aproximada de su ubicación, para lo cual el Reconocimiento Visual de Lugares, conocido como VPR por sus siglas en inglés, ofrece una solución eficaz cuando se dispone de una base de datos de imágenes georreferenciadas. Uno de los principales desafíos en VPR consiste en diseñar descriptores que sean compactos, informativos y discriminativos, pero que además se mantengan robustos ante fuertes cambios de punto de vista, variaciones estructurales o de iluminación. En este contexto, la primera parte de esta tesis propone dos direcciones complementarias para avanzar en VPR. En primer lugar, presentamos un método de agregación de características basado en la teoría de transporte óptimo. Además proponemos utilizar una potente arquitectura como red neuronal para obtener descriptores de imagen más robustos. En segundo lugar, proponemos una nueva estrategia de entrenamiento que mejora la sensibilidad geográfica de los descriptores seleccionando ejemplos difíciles basándonos tanto en similitud visual como en distancia espacial. Estas contribuciones suponen un avance hacia sistemas de VPR efectivos, escalables y versátiles, mejorando significativamente los resultados en benchmarks populares como MSLS Challenge o Nordland.

Dentro del campo de la reconstrucción o mapeado de escenas, la estimación de profundidad a partir de una sola imagen se suele considerar una de las tareas clave. Si bien es conocido que el uso de múltiples vistas aporta información geométrica que permite resolver ambigüedades y mejorar la precisión, el problema a resolver es cómo diseñar métodos capaces de aprovechar esta información de forma robusta en escenarios diversos y de propósito general. La segunda parte de esta tesis propone dos métodos para aprovechar las información multivista en la estimación de profundidad. Primero, presentamos un método de refinamiento en tiempo de inferencia que utiliza nubes de puntos 3D no densas obtenidas mediante Structure-from-Motion para guiar a las redes de profundidad monocular durante su ejecución, preservando así los conocimientos de la red mientras se incorporan restricciones geométricas adicionales. En segundo lugar, proponemos una arquitectura multivista de propósito general diseñada para operar de forma robusta en entornos variados y con rangos de profundidad diversos. Nuestras contribuciones se centran en la versatilidad: entrenando con múltiples conjuntos de datos, afrontando escenas con poco solapamiento y objetos dinámicos, y eliminando restricciones como el conocimiento previo del rango de profundidades. Conjuntamente, estas contribuciones muestran el potencial de combinar lo aprendido por las redes con restricciones geométricas, dando pasos hacia una integración fluida de la información multivista en la estimación de profundidad. Concretamente, el refinamiento que hemos propuesto ha mejorado todos los métodos de profundidad que probamos y el sistema de profundidad multivista que hemos desarrollado obtiene los mejores resultados actuales en el Robust Multi-View Depth Benchmark.

# Acknowledgements

It has been almost four years since I began this journey—a short and long time, all at once. The fact that I will remember this chapter of my life with joy in the future is thanks, almost entirely, to the people who surrounded me along the way.

The first person who made this journey so smooth has been my advisor, Javier. I've always seen myself as an optimistic person, but that self-perception was quickly humbled and recalibrated after witnessing Javier's unlimited enthusiasm and positivity—at any time, in any place, under any condition. He showed me the path to becoming a researcher, helped shape the direction when times were difficult, and taught me how to turn every experience into a positive lesson. In a single word, Javier's supervision has been: *fabulosa*.

The time spent at the lab wouldn't have been nearly as fun and stimulating without my colleagues from the 1.08 gang and associates. Special thanks to Lorenzo, Tomás, Edu, Javi, Javi, Víctor and Juanjo. From all the fun in Japan to the beaches of Sicily and camping at Mt. Rainier—those moments live rent-free in my memory.

I would also like to thank Kalpana and Gabe for providing me with the opportunity to spend two unforgettable summers at Skydio and Niantic, where I learned a great deal and had a fantastic experience. Special thanks to Jamie for all his support on my project during my stay in London.

Outside academia, I am very grateful to my amazing group of friends. Even without fully understanding what I work on, you've always shown genuine interest and honest curiosity about my progress. You always helped me put things into perspective. Thanks for all the good times, the beers, the trips, and the hikes.

To my family—my brother, my dad, and my mum. You've always supported me, ever since I was a kid, encouraging my curiosity and constantly motivating me to keep learning. Each of you is a great example and a daily inspiration for who I want to become.

A rather unusual thank you goes to the outdoors, especially the Pyrenees. Humans need a connection with nature to find peace—and PhD students even more so. You've been my refuge and my compass; hiking and cycling through your hills and valleys helped me find myself again and again. I am forever grateful and committed to your conservation.

The last paragraph goes to the most important person in this journey—Julia. You helped me discover my passion for research and opened a path I had never imagined for myself. In a mixture of luck and determination, we managed to share all of our PhD together—living in the same places, supporting each other through challenges, and creating unforgettable memories along the way. Not once feeling alone throughout this chapter has been a privilege for which I will always be grateful to you. You are the best colleague I could have found. You are the best friend I could have by my side. And you are the best partner I could have ever hoped for.

# Contents

# 1 Outline of the Thesis



Lo bueno, si breve, dos veces bueno.
*Brevity is the soul of wit.*

*-Spanish proverb*

Intelligent behavior in the physical world requires, in general, an understanding of tridimensional space and motion. Whether it is a robot navigating an unfamiliar building, a drone mapping terrain, or a smartphone overlaying digital content in Augmented Reality (AR), all these systems must have the ability to localize themselves, estimate a consistent representation of their surroundings, and interpret and interact with their environment in a spatially coherent manner. The field that aims to endow machines with these abilities is known as Spatial Artificial Intelligence (Spatial AI)—the study and development of systems that can perceive, model, and operate effectively within physical environments [1].

At the core of Spatial AI lies spatial perception—the process by which an agent interprets raw sensory data (images in this thesis) to infer spatial structure and egomotion. This includes a wide range of individual tasks, from low-level operations like image or feature matching [2–4] to mid-level tasks such as visual localization [5, 6], camera pose estimation [7–9] and depth prediction [10–12], and up to high-level processing pipelines like Simultaneous Localization And Mapping (SLAM) [13, 14].

Broadly speaking, these tasks aim to answer two central questions: *Where am I?* and *What surrounds me?* This thesis investigates key research challenges associated with both, and it is structured into two parts aligned with these two questions.

Part I of this thesis is focused on the *where*, and more precisely on Visual Place Recognition (VPR)—the task of retrieving a coarse location by matching a query image against a set of references. The usual approach to this task is to use a backbone model that extracts visual features, followed by an aggregation module that generates compact descriptors from these features. These models are typically trained end-to-end using metric learning losses.

Most of the research in VPR has focused on designing deep architectures or training pipelines to improve their accuracy and robustness. In Chapter 4, we introduce architectural contributions that leverage a large pre-trained vision model as the feature extractor, combined with a novel aggregation module to create highly discriminative global descriptors. This results in a powerful and easy-to-train model that achieves state-of-the-art results in common benchmarks. In Chapter 5, we shift focus to the training process, proposing a new hard-negative mining strategy that curates batches by sampling cliques of very similar-looking images. This significantly improves the recall in very aliased or densely sampled datasets.

Part II of this thesis addresses the *what*, specifically exploring multi-view depth estimation—the process of inferring the scene geometry leveraging information from multiple views. While a significant portion of recent research has focused on single-view depth, leading to large, powerful, and robust models, all of these are fundamentally limited by the ill-posed nature of the single-view

setup. Incorporating additional views offers a principled way to overcome their limitations and enhance robustness.

In this thesis, we propose two different manners of leveraging multi-view cues for monocular depth. First, in Chapter 8, we enhance single-view depth models with a Test-Time Refinement (TTR) strategy that uses sparse depth from a Structure-from-Motion (SfM) reconstruction as supervision. This improves the reconstruction's accuracy, especially at large depths, where our method leverages potentially large baselines from SfM. Then, in Chapter 9, we present a large general-purpose multi-view depth network trained on diverse datasets. Our model overcomes the limitations of previous multi-view stereo approaches regarding varying scales, unknown depth ranges, dynamic environments, and generalization to unseen environments.

## 1.1 List of Contributions

The contributions to this thesis, listed in what follows, stem from research publications, industry collaboration, open source development and peer reviewing of academic manuscripts.

### Publications

- Sergio Izquierdo and Javier Civera
  'SFM-TTR: Using Structure from Motion for Test-Time Refinement of Single-View Depth Networks'
  *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023
- Sergio Izquierdo and Javier Civera
  'Optimal Transport Aggregation for Visual Place Recognition'
  *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024
- Sergio Izquierdo and Javier Civera
  'Close, But Not There: Boosting Geographic Distance Sensitivity in Visual Place Recognition'
  *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024
- Sergio Izquierdo et al.
  'MVSAnywhere: Zero Shot Multi-View Stereo'
  *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025
- Blanca Lasheras-Hernandez et al.
  'Single-Shot Metric Depth from Focused Plenoptic Cameras'
  *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025

### Industry Collaboration

- June 2023 - September 2023: Autonomy Intern at Skydio, San Mateo (US). Supervised by Kalpana Seshadrinathan.
- July 2024 - December 2024: Research Intern at Niantic Labs, London (UK). Supervised by Gabriel Brostow.

### Open Source Development

- SfM-TTR.
  Code for [15] is available at:
  `https://github.com/serizba/SfM-TTR`.
  Licensed under the GNU General Public License v3.0.
- DINOv2 SALAD.
  Code and models for [16] are available at:
  `https://github.com/serizba/salad`.
  Licensed under the GNU General Public License v3.0.
- CliqueMining.
  Code and models for [17] are available at:
  `https://github.com/serizba/cliquemining`.
  Licensed under the GNU General Public License v3.0.
- MVSAnywhere.
  Code and models for [18] are available at:
  `https://github.com/nianticlabs/mvsanywhere`.
  Licensed allowing for non-commercial use only.

### Peer Reviewing

- IROS (2022, 2024)
- ICCV (2023)
- RA-L (2024, 2025, 2025)
- TPAMI (2024, 2025)
- ECCV (2024)
- CVPR (2025) (Outstanding reviewer)
- NeurIPS (2025)

## 1.2 Funding

This thesis has been funded by the Spanish Government with the pre-doctoral grant FPU20/02342. It has also been supported with projects from the Spanish Government (PID2021-127685NB-I00 and TED2021-131150B-I00) and the Aragón Government (DGA T45 23R, DGA FSE-T45 20R).

# I

# Visual Place Recognition

# 2 Motivation and Contributions

De noche todos los gatos son pardos.
*All cats are dark at night.*

*-Spanish proverb*

The ability to identify a place based on visual information is a fundamental human skill and a crucial component of our spatial awareness. This capability allows us to revisit familiar places, identify landmarks, and navigate through known environments effortlessly.

The fascination with this cognitive function expanded beyond its biological roots, inspiring popular games like GeoGuessr [20] and WhereTaken [21]. In these games, players are challenged to determine locations based on visual cues, showcasing the remarkable aptitude of humans to match patterns and recognize places[1].

1: Popular youtuber from the GeoGuessr community recognises places within 0.1 seconds on Geoguessr https://www.youtube.com/watch?v=ff6E4mrUkBY

Unsurprisingly, beyond humans requiring these skills for perception and engaging in them for fun, these abilities are also a fundamental building block for Spatial AI. They are crucial for SLAM [13, 14], AR [22], and absolute visual localization [23]. In SLAM, they enable loop closing so agents can identify previously visited locations and correct accumulated errors [24, 25]. In AR, they help recognize landmarks that applications may use to enhance the experience. They are also the first step in absolute visual localization, obtaining a first coarse location that is then refined with precise feature matching.

This part of the thesis focuses on the task of Visual Place Recognition, a subfield of visual perception concerned with determining the location of a query image by matching it to a database of geo-tagged reference images [26]. The term VPR is widely used in the research community to refer to this specific problem.

VPR is typically formulated as an image retrieval problem, where visual features of the images are extracted and aggregated to generate compact but descriptive representations. These representations are then compared to identify the most visually similar matches. For the retrieval to be effective, the representations must be both robust and discriminative, effectively handling challenges such as illumination changes, varying weather conditions, and structural transformations.

Recent advancements in VPR have primarily focused on two broad areas: namely the neural architectures and training process. Architectural improvements involve designing or adopting novel backbone models [27, 28], which are responsible for extracting meaningful and dense visual features from the input images. The aggregation module, which combines deep features into a compact and descriptive representation, has also been the subject of significant research [5, 29, 30]. Regarding the training pipeline, recent research encompasses strategies and techniques to optimize the model's performance, such as the losses [31–33], mining procedures [34], or datasets [6, 33] used during training.

In this thesis, we present architectural contributions, including using a pre-trained large vision model as the backbone model and a novel aggregation module, which are detailed in Chapter 4. Additionally, a novel mining strategy for the training pipeline is presented in Chapter 5. These contributions effectively

advance the robustness and generality of VPR methods, obtaining unprecedented results in well-established and challenging benchmarks. Specifically, we improve Recall@1 in the MSLS Challenge from 67.4% to 82.7% and in Nordland from 58.4% to 90.7%.

# 3 Related Work

The significant research efforts on VPR have been exhaustively compiled in a number of surveys and tutorials over the years [26, 35–38]. Current research addresses a wide variety of topics, such as novel loss functions [6, 32], image sequences [39–41], extreme viewpoint changes [42] or text features [43]. In this section, we focus on work related to the architecture, i.e. feature extraction and aggregation, as well as to the training pipeline, as there lie our contributions presented in Chapters 4 and 5.

## 3.1 Architectures

Early approaches to VPR used either aggregations of handcrafted local features [44–46] or global descriptors [47, 48]. In both cases, geometric [49] and temporal [49, 50] consistency was sometimes enforced for enhanced performance. With the emergence of deep neural networks, features pre-trained for recognition tasks, without fine-tuning, showed a significant performance boost over handcrafted ones [27]. However, training or fine-tuning specifically for VPR tasks using contrastive or triplet losses [31] offers an additional improvement and is standard nowadays.

Most of the backbones to extract image features used to be based on the ResNet architecture [5, 30, 51]. More recent works in VPR have shifted towards Vision Transformer (ViT) backbones [52, 53], obtaining significant improvements over previous models. This shift opened the door to use large pretrained foundation models like DINOv2 [54]. Although AnyLoc [28] first proposed to use DINOv2 in VPR, they use the model frozen, while in our research we show how finetuning this model can further improve performance.

NetVLAD [5] is one of the most popular architecture explicitly designed for VPR, mimicking the classical Vector of Locally Aggregated Descriptors (VLAD) aggregation [45] but jointly learning from data both convolutional features and cluster centroids. Later, Radenović et al. [30] proposed the Generalized Mean Pooling (GeM) to aggregate feature activations, also a popular baseline due to its simplicity and competitive performance. In addition to these, several other alternatives have been proposed in the literature. For example, Teichmann et al. [55] aggregates regions instead of local features. Recently, MixVPR [29] has presented the best results in the literature by aggregating deep features with a Multi-Layer Perceptron (MLP) layer.

A notable trend in VPR has been the adoption of a two-stage approach to enhance retrieval accuracy [51, 53, 56–59]. After a first stage with any of the methods presented in the previous paragraph, the top retrieved candidates are re-ranked attending to the un-aggregated local features, either assessing the geometric consistency to the query image or predicting their similarity. This re-ranking stage adds a considerable overhead, which is why it is only applied to a few candidates, but generally improves the performance. Re-ranking is out of the scope of our research but, notably, we outperform all baselines that employ re-ranking even if our model does not include such stage (and hence it is substantially faster).

In Chapter 4, we propose a novel aggregation module that uses optimal transport to assign features to clusters. Optimal transport has found a significant number of applications in graphics and computer vision [60]. Specifically, related to our research, it has been used for image retrieval [61], image matching [62] and feature matching [3, 4]. Recently, Zhang et al. [63] used optimal transport at the re-ranking stage in a retrieval pipeline. However, ours is the first work that proposes the formulation of local feature aggregation from an optimal transport perspective.

## 3.2 Training Pipelines

Overall, training details matter in image retrieval and are task-specific. Typically, contrastive [64] and triplet [65] losses are used to train a deep model that maps images into an embedding space, in which similar samples are close together and dissimilar ones are far apart. Although other losses have been proposed in the literature, *e.g.* [34, 66–72], Musgrave *et al.* [31] and Roth *et al.* [73] showed a higher saturation than the one reported in the previous literature. The particularities of VPR, however, can be leveraged in task-specific losses. For example, Leyva-Vallina *et al.* [32] grade similarity based on spatial overlap to make losses more informative. Ali-bey *et al.* [33] showed that the multi-similarity loss [74] can be effectively used for VPR tasks. They curated a dataset, GSV-Cities, and organized it on sparse places that, combined with the multi-similarity loss led to significant performance gains. As other recent works [29, 75], our contributions builds on top of the multi-similarity loss on GSV-Cities. However, the sparse nature of the GSV-Cities dataset [33] limits the effectiveness of the models in densely sampled data, present in many benchmarks [39, 76]. We argue that densely sampled data is relevant in VPR as it is a prevalent condition in numerous applications, owing to the proliferation of mobile computational platforms capturing video (such as cars, drones, glasses and phones) and the availability of tools to crowdsource and store big data.

Mining informative batches matters as much or even more than the chosen losses [34]. "Easy" samples contribute with small loss values, which may slow down or plateau the training [31]. On the other hand, using only "hard" samples produces noisy gradients and may overfit or converge to local minima [34, 77], which suggests a sweet spot in mixed strategies [78]. As another taxonomy, mining can be done offline after a certain number of iterations [79–81], with high computational costs, or online within each batch [82, 83]. In practice, "hard" negatives samples are typically used, as they are easy to mine and informative [5, 39, 84, 85]. "Hard" positive mining [86–89] is more challenging to implement, as it is sometimes caused by occlusions, large scale changes or low overlap, which may be misleading and harm generalization [30]. Wang *et al.* [74] generalizes sampling schemes by weighting pairs in the multi-similarity loss according to their embedding distance.

# 4 Optimal Transport Aggregation



*The task of Visual Place Recognition aims to match a query image against references from an extensive database of images from different places, relying solely on visual cues. State-of-the-art pipelines focus on the aggregation of features extracted from a deep backbone, in order to form a global descriptor for each image. In this context, we introduce SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors), which reformulates NetVLAD's soft-assignment of local features to clusters as an optimal transport problem. In SALAD, we consider both feature-to-cluster and cluster-to-feature relations and we also introduce a 'dustbin' cluster, designed to selectively discard features deemed non-informative, enhancing the overall descriptor quality. Additionally, we leverage and fine-tune DINOv2 as a backbone, which provides enhanced description power for the local features, and dramatically reduces the required training time. As a result, our single-stage method not only surpasses single-stage baselines in public VPR datasets, but also surpasses two-stage methods that add a re-ranking strategy with significantly higher cost.*

Chapter based on [16]:
Sergio Izquierdo and Javier Civera
'Optimal Transport Aggregation for Visual Place Recognition'
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024

The code and models are available at `https://github.com/serizba/salad`

Recognizing a place solely from images becomes a challenging task when scenes undergo substantial changes in their structure or appearance. Such capability is referred to in the scientific and technical literature as VPR, and is essential for agents to navigate and understand their surroundings autonomously in a wide array of applications, such as robotics [90–94] or AR [37]. Specifically, it is present in SLAM [13, 95] and absolute pose estimation [96, 97] pipelines.

In practice, VPR is framed as an image retrieval problem, wherein typically a query image serves as the input and the goal is to obtain an ordered list of top-k matches against a pre-existing database of geo-localized reference images. Images are represented as an aggregation of appearance pattern descriptors, which are subsequently compared via nearest neighbour. The effectiveness of this matching relies on generating discriminative per-image descriptors that exhibit robust performance even for challenging variations such as fluctuating illumination, structural transformations, temporal changes, weather and seasonal shifts. Most recent research on VPR have thus focused on the two key components of this general pipeline, namely the deep neural backbones for feature extraction and methods for aggregating such features.

For years, ResNet-based neural networks have been the predominant backbones for feature extraction [5, 30, 51]. Recently, given the success of ViT for different computer vision tasks [98–101], some methods have introduced ViT in the field of VPR [52, 53]. AnyLoc [28] proposed to leverage foundation models, using DINOv2 [54] as a feature extractor for VPR. However, AnyLoc uses DINOv2 'as is', while we show in this chapter that fine-tuning the model for VPR brings a significant increase in performance.

Regarding aggregation, NetVLAD [5], the learned counterpart to the traditional handcrafted VLAD [45], is among the most popular choices. Alternative methods include pooling layers like GeM [30] or learned global aggregation, like the

recent MixVPR [29]. In this chapter, we propose optimal transport aggregation, setting a new state of the art in VPR.

As a summary, in this work, we present a single-stage approach to VPR that obtains state-of-the-art results in the most common benchmarks. To achieve this, we present two key contributions:

- First, we propose SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors), a reformulation of the feature-to-cluster assignment problem through the lens of optimal transport, allowing more effective distribution of local features into the global descriptor bins. To further improve the discriminative power of the aggregated descriptor, we let the network discard uninformative features by introducing a 'dustbin' mechanism.
- Secondly, we integrate the representational power of foundation models into VPR, using DINOv2 as the backbone for feature extraction. Unlike previous approaches that utilized DINOv2 in its pre-trained form, our method involves fine-tuning the model specifically for the task. This fine-tuning process converges extremely fast, in just four epochs, and allows DINOv2 to capture more relevant and distinctive features pertinent to place recognition tasks.

The fusion of these two novel components results in DINOv2 SALAD, which can be efficiently trained in less than one hour and sets unprecedented recall in VPR benchmarks, with 75.0% Recall@1 in MSLS Challenge and 76.0% in Nordland. All of this with a single-stage pipeline, without requiring expensive post-processing steps and with an inference speed of less than 3 ms per image.

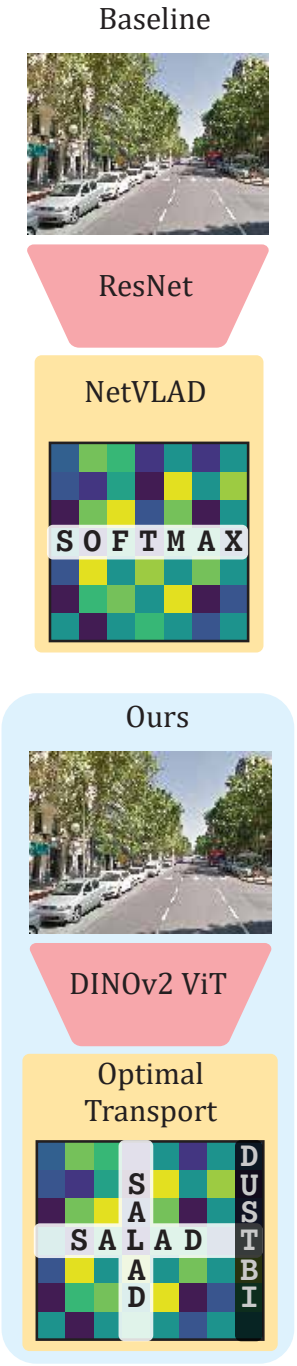



**Figure 4.1: Illustration of a VPR baseline (left) and our contribution (right).** The left column outlines a typical VPR baseline, a ResNet backbone followed by NetVLAD aggregation [5]. On the right column, we replace ResNet with a partially fine-tuned DINOv2 [54] backbone, and incorporate SALAD, our novel optimal transport aggregation using the Sinkhorn Algorithm. Our model achieves unprecedented state-of-the-art results on common VPR benchmarks.

## 4.1 Method: SALAD

DINOv2 SALAD is based on NetVLAD, but we propose to use and fine-tune the DINOv2 backbone (Subsection 4.1.1) and propose a novel module (SALAD) for the assignment (Subsection 4.1.2) and aggregation (Subsection 4.1.3) of features.

### 4.1.1 Local Feature Extraction

Effective local feature extraction lies in striking a balance: features must be robust enough to withstand substantial changes in appearance, such as those between seasons or from day to night, yet they should retain sufficient information on local structure to enable accurate matching.

Inspired by the success of ViT architectures in many computer vision tasks and by AnyLoc [28], that leverages the exceptional representational capabilities of foundation models [102], we adopt DINOv2 [54] as our backbone. However, differently from AnyLoc, we use a supervised pipeline and include the backbone in the end-to-end training for the specific task, yielding improved performance.

DINOv2 adopts a ViT architecture that initially divides an input image $\mathbf{I} \in \mathbb{R}^{h \times w \times c}$ into $p \times p \times c$ patches, with $p = 14$. These patches are sequentially projected with transformer blocks, resulting in the output tokens $\{\mathbf{t}_1, \ldots, \mathbf{t}_n, \mathbf{t}_{n+1}\}$, $\mathbf{t}_i \in \mathbb{R}^d$, where $n = hw/p^2$ is the number of input patches and there is an additional global token $\mathbf{t}_{n+1}$ that aggregates class information. Although the DINOv2's authors reported that fine-tuning the model only brings dim improvements, we

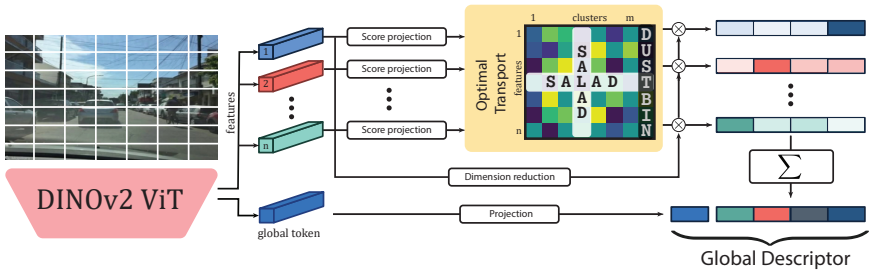



**Figure 4.2: Overview of our method**. First, the DINOv2 backbone extracts local features and a global token from an input image. Then, a small MLP, score projection, computes a score matrix for feature-to-cluster and dustbin relationships. The optimal transport module uses the Sinkhorn algorithm to transform this matrix into an assignment, and subsequently, dimensionality-reduced features are aggregated into the final descriptor based on this assignment and concatenated with the global token.

found that at least for VPR there are substantial gains in selectively unfreezing and training the last blocks of the encoder.

### 4.1.2 Assignment

In NetVLAD, a global descriptor is formed by assigning a set of features to a set of clusters, $\{C_1, \ldots, C_j, \ldots, C_m\}$, and then aggregating all features that belong to each cluster. For the assignment, NetVLAD computes a score matrix $\mathbf{S} \in \mathbb{R}_{>0}^{n \times m}$, where the element in its $i^{\text{th}}$ row and $j^{\text{th}}$ column, $s_{i,j} \in \mathbb{R}_{>0}$, represents the cost of assigning a feature to a cluster $C_j$. In other words, $\mathbf{S}$ quantifies the affinity of each feature to each cluster. While SALAD draws inspiration from NetVLAD, we identify several crucial aspects in their assignment and propose alternatives to address these.

**Reduce assignment priors.** When building the score matrix $\mathbf{S}$, NetVLAD introduces certain priors. Specifically, it initializes the linear layer that computes $\mathbf{S}$ with centroids derived from k-means. While this may accelerate the training, it introduces inductive bias and potentially makes the model more susceptible to local minima. In contrast, we propose to learn each row $\mathbf{s}_i$ of the score matrix from scratch with two fully connected layers initialized randomly:

$$\mathbf{s}_i = \mathbf{W}_{s_2}(\sigma(\mathbf{W}_{s_1}(\mathbf{t}_i) + \mathbf{b}_{s_1})) + \mathbf{b}_{s_2} \tag{4.1}$$

where $\mathbf{W}_{s_1}$, $\mathbf{W}_{s_2}$ and $\mathbf{b}_{s_1}$, $\mathbf{b}_{s_2}$ are the weights and biases of the layers, and $\sigma$ is a non-linear activation function.

**Discard uninformative features.** Some features, such as those representing the sky, might contain negligible information for VPR. NetVLAD does not account for this, and the contribution of all features is preserved in the final descriptor. Contrary, we follow recent works on keypoint matching and introduce a 'dustbin' where non-informative features are assigned to. For that, we augment the score matrix, from $\mathbf{S}$ to $\bar{\mathbf{S}} = [\mathbf{S}, \bar{\mathbf{s}}_{i,m+1}] \in \mathbb{R}_{>0}^{n \times m+1}$, by appending the column $\bar{\mathbf{s}}_{i,m+1}$ representing the feature-to-dustbin relation. As in SuperGlue [4], this score is modeled with a single learnable parameter $z \in \mathbb{R}$:

$$\bar{\mathbf{s}}_{i,m+1} = z\mathbf{1}_n \tag{4.2}$$

being $\mathbf{1}_n = [1, \ldots, 1]^\top \in \mathbb{R}^n$ a $n$-dimensional vector of ones.

**Optimal assignment.** The original NetVLAD assignment computes a per-row softmax over $\mathbf{S}$ to obtain the distribution of each feature's mass across

the clusters. However, this approach only considers the feature-to-cluster relationship and overlooks the reverse –the cluster-to-feature relation. For this reason, we reformulate the assignment as an optimal transport problem where the features' mass, $\boldsymbol{\mu} = \mathbf{1}_n$, must be effectively distributed among the clusters or the 'dustbin', $\boldsymbol{\kappa} = [\mathbf{1}_m^\top, n-m]^\top$. We follow SuperGlue [4] and use the Sinkhorn Algorithm [103, 104] to obtain the assignment $\bar{\mathbf{P}} \in \mathbb{R}^{n\times(m+1)}$ such that

$$\bar{\mathbf{P}}\mathbf{1}_{m+1} = \boldsymbol{\mu} \quad \text{and} \quad \bar{\mathbf{P}}^\top \mathbf{1}_n = \boldsymbol{\kappa}. \tag{4.3}$$

This algorithm finds the optimal transport assignment between distributions $\boldsymbol{\mu}$ and $\boldsymbol{\kappa}$ iteratively normalizing rows and columns from $\exp\left(\bar{\mathbf{S}}\right)$. Finally, we drop the dustbin column to obtain the assignment $\mathbf{P} = \left[\mathbf{p}_{*,1}, \ldots, \mathbf{p}_{*,m}\right]$, where $\mathbf{p}_{*,j}$ stands for the $j^{\text{th}}$ column of $\mathbf{P}$.

### 4.1.3 Aggregation

Once the feature assignment in our SALAD framework is computed as detailed in Subsection 4.1.2, we focus on the aggregation of these assigned features to form the final global descriptor. The aggregation process in NetVLAD involves combining all features assigned to each cluster $C_j$. However, we introduce three variations:

**Dimensionality reduction.** To efficiently manage the final descriptor size, we first reduce the dimensionality of the tokens from $\mathbb{R}^d$ to $\mathbb{R}^l$. This is achieved by processing the features through two fully connected layers, precisely adjusting the size of the feature vectors while retaining the essential information from the task.

$$\mathbf{f}_i = \mathbf{W}_{f_2}(\sigma(\mathbf{W}_{f_1}(\mathbf{t}_i) + \mathbf{b}_{f_1})) + \mathbf{b}_{f_1} \tag{4.4}$$

**Aggregation.** Based on the assignment matrix derived using the Sinkhorn Algorithm, each feature is aggregated into its assigned cluster. Differently from NetVLAD, we do not subtract the centroids to get the residuals. We directly aggregate these features with a summation, reducing the incorporated priors about the aggregation. Viewing the resulting VLAD vector as a matrix $\mathbf{V} \in \mathbb{R}^{m\times l}$, each element $V_{j,k} \in \mathbb{R}$ is computed as follows:

$$V_{j,k} = \sum_{i=1}^{n} P_{i,k} \cdot f_{i,k} \tag{4.5}$$

where $f_{i,k}$ corresponds to the $k^{\text{th}}$ dimension of $\mathbf{f}_i$, with $k \in \{1, \ldots, l\}$.

**Global token.** To include global information about the scene not easily incorporated into local features, we also incorporate a scene descriptor $g$ computed as:

$$\mathbf{g} = \mathbf{W}_{g_2}(\sigma(\mathbf{W}_{g_1}(\mathbf{t}_{n+1}) + \mathbf{b}_{g_1})) + \mathbf{b}_{g_1} \tag{4.6}$$

where $\mathbf{t}_{n+1}$ is the global token from DINOv2. We then concatenate $\mathbf{g}$ with $\mathbf{V}$ flattened. Following NetVLAD, we do an L2 intra-normalization and an entire L2 normalization of this vector, which yields the final global descriptor.

## 4.2 Experiments

To rigorously evaluate the effectiveness of our proposed contributions, we conducted exhaustive experiments following standard evaluation protocols.

### 4.2.1 Implementation Details

We ground our training and evaluation setups on the publicly provided framework by MixVPR [1].

1: https://github.com/amaralibey/MixVPR

For the **architecture**, we opt for a pretrained DINOv2-B backbone, targeting a balance between computational efficiency and representational capacity. We only fine-tune the final 4 layers of the encoder, which significantly enhances the performance without markedly increasing training time. For the fully connected layers, the weights of the hidden layers $\mathbf{W}_{s_1}$, $\mathbf{W}_{f_1}$ and $\mathbf{W}_{g_1}$ have 512 neurons and use ReLU for the activation function $\sigma$. To optimize feature handling, we employ a dimensionality reduction, compressing feature token dimensions from $d = 768$ to $l = 128$, and the global to 256. We use $m = 64$ clusters, resulting in a global descriptor of size $128 \times 64 + 256$. We also report results with smaller descriptors, with size $512 + 32$ ($m = 15,\ l = 32$), and $2048 + 64$ ($m = 32,\ l = 64$).

We **train** on GSV-Cities [33], a large dataset of urban locations collected from Google Street View. Given the impressive representation power of DINOv2, our pipeline achieves training convergence within just 4 epochs. Using a batch size of 60 places, each represented by 4 images, the training is completed in 30 minutes on a single NVIDIA RTX 3090. We use the multi-similarity loss [74] and AdamW [105] for the optimization, with an initial learning rate set to 6e−5. To ensure an effective learning rate, we linearly decay the initial rate at every iteration so at the end of the training is 20% of the initial value. We use a dropout rate of 0.3 on the score projection and dimensionality reduction neurons. As our model is agnostic to the image input size (as long as it can be divided in $14 \times 14$ patches), we evaluate on images of size $322 \times 322$ but train on $224 \times 224$ to speedup training time.

To **validate** our experiments and select the hyperparameters, we monitored the recall in the Pittsburg30k-test [106]. We observed that, in the long run, most configurations perform similarly, but rapid convergence on a few epochs is more sensitive to the hyperparameters.

### 4.2.2 Results

We benchmarked our model against several single-stage baselines, namely NetVLAD [5] and GeM [30] as two representative tradicional baselines, and Conv-AP [33], CosPlace [6], MixVPR [29] and EigenPlaces [107] as the four most recent and best performing baselines in the literature. The evaluation spanned a diverse array of well-established datasets: MSLS Validation and Challenge [39], which are comprised of dashcam images; Pittsburgh250k-test [106], featuring urban scenarios; SPED [92], a collection from surveillance cameras; NordLand, notable for its seasonal variations from images captured from the front of a train traversing Norway; and SF-XL [6], a large urban dataset to evaluate VPR at scale. We use Recall@k (R@k) as the metric for all our experiments, as it is standard in related work. We use evaluation data and code from MixVPR [29], which considers retrieval as correct if an image at less than 25 meters (or two frames for Nordland) from the query is among the top-k predicted candidates.

**Table 4.1: Comparison against single-stage baselines.** We compare DINOv2 SALAD against two popular baselines [5, 30] and the four baselines that show best results in recent literature [6, 29, 33, 107]. Our slim version already obtains state-of-the-art results in all metrics. Our full model outperforms all previous results by a significant margin. Note, in particular, the large improvement in the most challenging benchmarks, MSLS Challenge and NordLand. † We reproduced GeM results training during 80 epochs following MixVPR training pipeline.

| Method | Desc. size | Latency (ms) | MSLS Challenge | | MSLS Val | | NordLand | | Pitts250k-test | | SPED | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 | R@1 | R@5 |
| NetVLAD [5] | 32768 | 1.41 | 35.1 | 47.4 | 82.6 | 89.6 | 32.6 | 47.1 | 90.5 | 96.2 | 78.7 | 88.3 |
| GeM [30]† | 1024 | 1.14 | 49.7 | 64.2 | 78.2 | 86.6 | 21.6 | 37.3 | 87.0 | 94.4 | 66.7 | 83.4 |
| Conv-AP [33] | 8192 | 1.22 | 54.2 | 66.6 | 83.1 | 90.3 | 42.7 | 58.9 | 92.9 | 97.7 | 79.2 | 88.6 |
| CosPlace [6] | 2048 | 2.59 | 67.2 | 78.0 | 87.4 | 93.0 | 44.2 | 59.7 | 92.1 | 97.5 | 80.1 | 89.6 |
| MixVPR [29] | 4096 | 1.37 | 64.0 | 75.9 | 88.0 | 92.7 | 58.4 | 74.6 | 94.6 | 98.3 | 85.2 | 92.1 |
| EigenPlaces [107] | 2048 | 2.65 | 67.4 | 77.1 | 89.3 | 93.7 | 54.4 | 68.8 | 94.1 | 98.0 | 69.9 | 82.9 |
| DINOv2 SALAD | 512 + 32 | 2.33 | 70.8 | 83.6 | 89.3 | 94.9 | 61.2 | 78.9 | 93.0 | 97.4 | 88.5 | 94.7 |
| DINOv2 SALAD | 2048 + 64 | 2.35 | 73.7 | 85.9 | 90.5 | 95.4 | 70.4 | 85.7 | 94.8 | 98.3 | 89.5 | 94.9 |
| DINOv2 SALAD | 8192 + 256 | 2.41 | **75.0** | **88.8** | **92.2** | **96.4** | **76.0** | **89.2** | **95.1** | **98.5** | **92.1** | **96.2** |

**Table 4.2: Comparison against baselines with re-ranking.** We compare our single-stage DINOv2 SALAD with methods that perform a re-ranking stage to improve performance. Without using re-ranking, our DINOv2 SALAD outperforms all other methods while being orders of magnitude faster and more memory-efficient. Latency metrics obtained from [53] using a RTX A5000. Latency for DINOv2 SALAD was computed using a RTX 3090. Memory footprint is calculated on the MSLS Val dataset, which includes around 18,000 images.

| Method | Desc. size | | Memory (GB) | Latency (ms) | | MSLS Challenge | | | MSLS Val | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Global | Local | | Retrieval | Reranking | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| Patch-NetVLAD [51] | 4096 | $2826 \times 4096$ | 908.30 | 9.55 | 8377.17 | 48.1 | 57.6 | 60.5 | 79.5 | 86.2 | 87.7 |
| TransVPR [52] | **256** | $1200 \times 256$ | 22.72 | 6.27 | 1757.70 | 63.9 | 74.0 | 77.5 | 86.8 | 91.2 | 92.4 |
| R2Former [53] | **256** | $500 \times 131$ | 4.7 | 8.88 | 202.37 | 73.0 | 85.9 | 88.8 | 89.7 | 95.0 | 96.2 |
| **DINOv2 SALAD (ours)** | 8192 + 256 | | **0.63** | **2.41** | **0.0** | **75.0** | **88.8** | **91.3** | **92.2** | **96.4** | **97.0** |

**Table 4.3: Ablations**. The first two rows correspond to two baselines in the literature [5, 28], the rest to different aggregations appended to DINOv2 including our DINOv2 SALAD. Note that only DINO NetVLAD, with a significantly bigger descriptor size than ours, is able to show competitive results. We outperform all the rest DINOv2 baselines of similar descriptor sizes by a large margin.

| Method | Desc. size | MSLS Challenge | | | MSLS Val | | | NordLand | | | Pitts250k-test | | | SPED | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| ResNet NetVLAD [5] | 32768 | 35.1 | 47.4 | 51.7 | 82.6 | 89.6 | 92.0 | 32.6 | 47.1 | 53.3 | 90.5 | 96.2 | 97.4 | 78.7 | 88.3 | 91.4 |
| DINOv2 AnyLoc [28] | 49152 | 42.2 | 53.5 | 58.1 | 68.7 | 78.2 | 81.8 | 16.1 | 25.4 | 30.4 | 87.2 | 94.4 | 96.5 | 85.3 | 94.4 | 95.4 |
| ResNet SALAD | 8192 | 57.4 | 70.8 | 74.9 | 83.2 | 89.5 | 91.8 | 33.3 | 49.6 | 55.8 | 91.4 | 96.9 | 97.9 | 75.0 | 86.7 | 89.8 |
| ConvNext [108] SALAD | 8192 | 63.9 | 75.2 | 80.1 | 85.5 | 92.4 | 94.5 | 47.8 | 64.3 | 70.3 | 93.9 | 97.9 | 98.8 | 83.5 | 90.9 | 92.9 |
| DINOv2 GeM | 4096 | 62.6 | 78.3 | 83.0 | 85.4 | 93.9 | 95.0 | 35.4 | 52.5 | 59.6 | 89.5 | 96.5 | 98.0 | 83.0 | 92.1 | 93.9 |
| DINOv2 MixVPR | 4096 | 72.1 | 85.0 | 88.3 | 90.0 | 95.1 | 96.0 | 63.6 | 80.1 | 84.6 | 94.6 | 98.3 | **99.3** | 89.8 | 94.9 | 96.1 |
| DINOv2 NetVLAD | 24576 | **75.8** | 86.5 | 89.8 | **92.4** | 95.9 | 96.9 | 71.8 | 86.5 | 90.1 | **95.6** | **98.7** | **99.3** | 90.8 | 95.7 | **96.7** |
| DINOv2 NetVLAD (dim. red.) | 8192 | 73.3 | 85.6 | 88.3 | 90.1 | 95.4 | 96.8 | 70.1 | 86.5 | 90.2 | 95.4 | 98.4 | 99.1 | 90.6 | 95.4 | **96.7** |
| **DINOv2 SALAD (ours)** | 8192 + 256 | 75.0 | **88.8** | **91.3** | 92.2 | **96.4** | **97.0** | **76.0** | **89.2** | **92.0** | 95.1 | 98.5 | 99.1 | **92.1** | **96.2** | 96.5 |

As shown in Table 4.1, our model outperforms all previous methods on all datasets and all metrics. Even the smaller 512 + 32 version already surpasses previous models with bigger descriptors on most datasets. It is worth highlighting the metrics saturation observed in MSLS Val, Pitts250k-test and SPED, and on the other hand the challenging nature of MSLS Challenge and NordLand. The MSLS Challenge dataset, with its diversity, extensive size and closed labels, and NordLand, with its extreme sample similarity and seasonal shifts, emerge then as key benchmarks for assessing VPR performance. Although our DINOv2 SALAD shows a significant improvement on *all* benchmarks, it is precisely in MSLS Challenge and NordLand where we obtain the most substantial recall increases, with +7.6%, +11.7% and +17.6%, +14.6% for R@1, R@5 respectively over the second best.

For SF-XL, as shown in Table 4.4, our method also achieves the best results to date. This is remarkable, considering that the previous state of the art was trained on this dataset, whereas our method never used any image of San Francisco when it was fine-tuned.

In Table 4.2, we compare our DINOv2 SALAD method, which solely operates on a single retrieval stage, against the leading two-stage VPR techniques. In this comparison, we include the best performing models in the literature, namely R2Former [53], TransVPR [52], and Patch-NetVLAD [51], which incorporate a re-ranking refinement. Note how our DINOv2 SALAD, despite being orders of magnitude faster and smaller in memory, significantly outperforms all these two-stage methods on all benchmarks. This finding not only highlights the efficiency of our model but also demonstrates the effectiveness of global retrieval using our novel SALAD aggregation. Additionally, considering our method's reliance on local features, we believe that a re-ranking stage could also be applied, potentially increasing our recall metrics but at the price of a higher computational footprint.

### 4.2.3 Ablation Studies

**Effect of DINOv2**. We assess the impact of the DINOv2 backbone and our optimal transport aggregation SALAD separately. For this, we compare with the existing baselines of ResNet NetVLAD or AnyLoc, this last one applying a VLAD on top of a pretrained DINOv2 encoder. We integrate the DINOv2 backbone with various aggregation modules, obtaining a handful of performant techniques that improve their respective previous results. As shown in Table 4.3, all of these outperform the baselines, even though AnyLoc already uses DINOv2. This validates the DINOv2's integration in end-to-end fine-tuning to refine its capabilities.

**Effect of SALAD**. Our experiments in Table 4.3 show that aggregation also matters. Even the recent MixVPR aggregation coupled with DINOv2 does not match the performance of DINOv2 NetVLAD and DINOv2 SALAD. We believe that the DINOv2 backbone is especially suitable for local feature aggregation, as its features work remarkably well in dense visual perception tasks [54, 109, 110].

| Method | Desc. size | SF-XL Test v1 | SF-XL Test v2 |
|---|---|---|---|
| CosPlace [6] | 2048 | 76.4 | 88.8 |
| EigenPlaces [107] | 2048 | 84.1 | 90.8 |
| DINOv2 SALAD | 8192 + 256 | **88.6** | **94.8** |

**Table 4.4: Results on SF-XL. (R@1)** Our DINOv2 SALAD achieves unprecedented results on SF-XL despite never seeing any single image of San Francisco during VPR finetuning.

| Model | Dim. size | # Params. | Latency (ms) | MSLS Val R@1 |
|---|---|---|---|---|
| S | 384 | 21M | 1.30 | 90.5 |
| B | 768 | 86M | 2.41 | 92.2 |
| L | 1024 | 300M | 7.82 | 92.6 |
| G | 1536 | 1100M | 24.93 | 91.7 |

**Table 4.5: DINOv2 configurations and performances.**

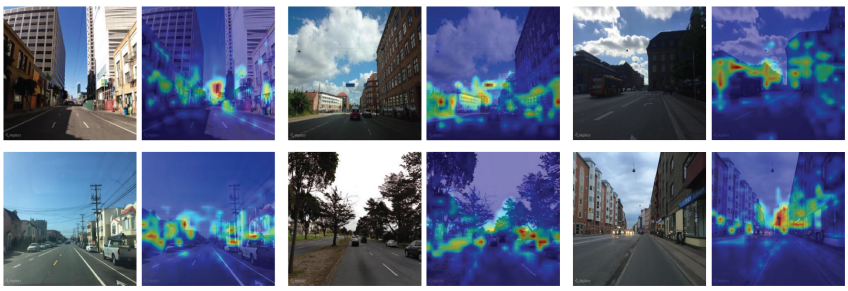


**Figure 4.3: Heatmap of local features importance**. Left images show the original pictures, their right counterparts represent the weights *not* assigned to the 'dustbin'. Note how the network learns to discard uninformative regions like skies, roads or dynamic objects, and instead focus on distinctive patterns in buildings and vegetation. We attribute its focus on distant buildings to their invariance to viewpoint change.

Although DINOv2 NetVLAD achieves comparable performance to SALAD, it employs a descriptor almost three times as big. Besides, the generalization performance of DINOv2 NetVLAD is limited, as observed in NordLand results. We attribute this to NetVLAD's priors initialization with urban scenarios, which constrain the convergence of the system. In our experiments we also trained a slimmer DINOv2 NetVLAD version, whose features are dimensionally reduced as described in Subsection 4.1.3, targetting a final descriptor of roughly the same size as SALAD. In this fairer setup, DINOv2 SALAD clearly outperforms DINOv2 NetVLAD. We also evaluate SALAD on top of ResNet and ConvNext backbones, which improves over baseline ResNet NetVLAD but is significantly worse than using DINOv2. This indicates that SALAD is specially suited for high spatial resolution features, like the ones from DINOv2.

**Effect of hyperparameters**. DINOv2 comes in different sizes that affect the number of parameters, inference speed, and representation capabilities. As shown in Table 4.5, more parameters do not always result in better performance. Excessively big models might be harder to train or prone to overfitting the training set. From these results, we chose the DINOv2-B backbone, which exhibits a great balance between performance and size and speed. Regarding descriptor size, we observed (Table 4.1) that changing $m$ and $l$ allows to get slimmer versions with competitive performance. For the number of blocks to train, as shown in Table 4.6, fine-tuning two or four block report the best results without significant computation overhead.

**Effect of SALAD components**. In Table 4.6, we show how different components of our SALAD pipeline affect the final performance. Both the global token, which

| Method | MSLS Val | | |
|---|---|---|---|
| | R@1 | R@5 | R@10 |
| DINOv2 SALAD (frozen) | 88.5 | 95.0 | 96.2 |
| DINOv2 SALAD (train 2 last blocks) | 92.0 | **96.5** | **97.0** |
| DINOv2 SALAD (train 4 last blocks) | **92.2** | 96.4 | **97.0** |
| DINOv2 SALAD (train 6 last blocks) | 91.6 | 96.2 | **97.0** |
| DINOv2 SALAD (train all blocks) | 89.2 | 95.1 | 96.1 |
| DINOv2 SALAD w/o dustbin | 91.4 | 95.8 | 96.2 |
| DINOv2 SALAD w/o global token | 91.8 | 96.0 | 96.2 |
| DINOv2 SALAD (Dual Softmax) | 91.9 | 95.7 | 96.5 |
| DINOv2 SALAD | **92.2** | **96.4** | **97.0** |

**Table 4.6: Ablation study of the SALAD components.**

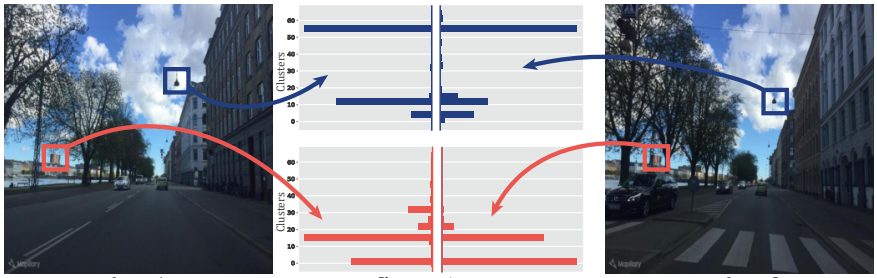



**Figure 4.4: Illustration of feature-to-cluster assignments.** See at the leftmost and rightmost part of the figure two different views of the same place. Framed by red and blue squares we highlight two corresponding patches in each of the images. The central part of the figure shows the feature-to-cluster assignments for these patches. Note how DINOv2 SALAD correctly assigns the features to the same bins for both views, even with different local texture.

appends global information not captured in local features, and the dustbin, which helps in distilling the aggregated features, contribute to the performance of SALAD. We also trained a model using a dual-softmax [111] to solve the optimal transport assignment, following LoFTR and Gluestick [3, 112]. Although dual-softmax achieves only slightly worse performance, the Sinkhorn Algorithm is theoretically sound and provides a better acronym to our method.

### 4.2.4 Introspective Results

We provide an introspection of our model's performance through a series of illustrative figures. Figure 4.3 visualizes the weights that are not assigned to the 'dustbin', offering insight into the parts of the input image that the network considers informative. As the 'dustbin' assignment is completely learnt by the network, some discarded features might be counter-intuitive. However, we observe that it typically removes dynamic objects and focuses on the most distinctive and invariant parts of the image. In Figure 4.4, we display the assignment distribution of patches from two different images depicting the same place. It demonstrates the model's ability to consistently distribute most of the weights into the same bins for patches representing similar regions. Such repeatable and consistent assignment across different images of the same place is crucial for the reliability and performance of the system. Finally, in Figure 4.5, we showcase various query images alongside their respective top-3 retrievals made by our system. DINOv2 SALAD is able to retrieve correct predictions even under challenging conditions, such as severe changes in illumination or viewpoint.

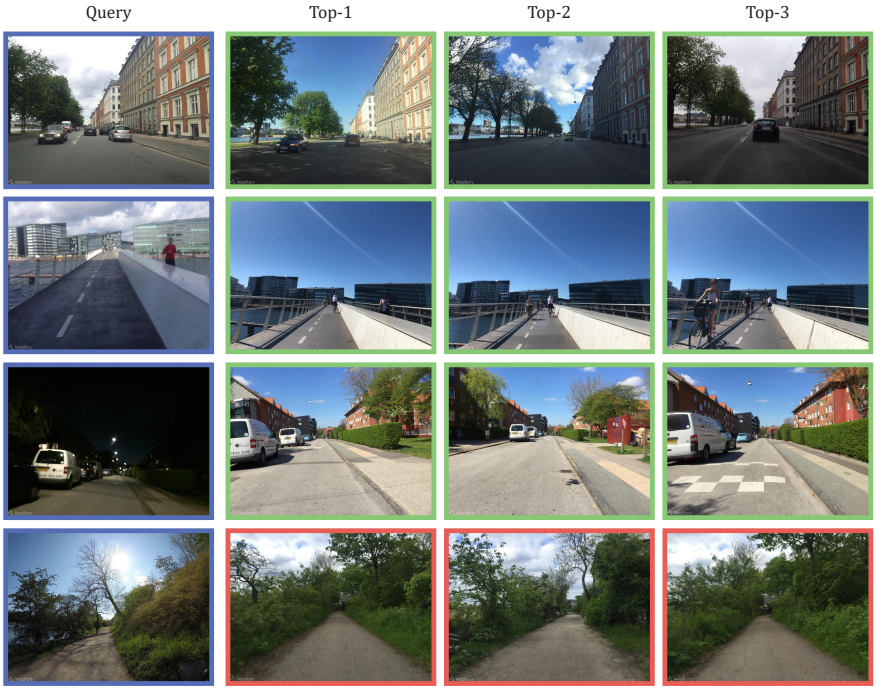



**Figure 4.5: DINOv2 SALAD qualitative results at MSLS.** The left column shows several queries and the three other ones shows the top-3 candidates retrieved by our DINOv2 SALAD. Candidates are framed in green if they correspond to the same place as the query, and in red if they do not. Note the correct retrievals under seasonal, weather, viewpoint and day-night changes. Note also a challenging failure case in the last row, due to non-discriminative image content.

## 4.3 Conclusions and Limitations

In this chapter, we have proposed DINOv2 SALAD, a novel model for VPR that outperforms previous baselines by a substantial margin. This achievement is the result of combining two key contributions: a fine-tuned DINOv2 backbone for enhanced feature extraction and our novel SALAD (Sinkhorn Algorithm for Locally Aggregated Descriptors) module for feature aggregation. Our extensive experiments demonstrate the effectiveness of these modules, highlighting the model's single-stage nature and exceptionally fast training and inference speed.

While our work brings significant improvements in performance, it is not without limitations. Primarily, the adoption of DINOv2 as our backbone results in slower processing speeds compared to ResNet-based methods. Besides, although SALAD is a general aggregation module, its effectiveness is tied to the choice of backbone. It excels with DINOv2, which offers high spatial resolution features, but it is less suited for coarser features. Additionally, in SALAD we use an optimal transport assignment in its simplest form. More sophisticated constraints could improve the resulting assignment, a very relevant aspect for our future work.

# 5 Boosting Geographic Distance Sensitivity



*Visual Place Recognition plays a critical role in many localization and mapping pipelines. It consists of retrieving the closest sample to a query image, in a certain embedding space, from a database of geotagged references. The image embedding is learned to effectively describe a place despite variations in visual appearance, viewpoint, and geometric changes. In this work, we formulate how limitations in the Geographic Distance Sensitivity of current VPR embeddings result in a high probability of incorrectly sorting the top-k retrievals, negatively impacting the recall. In order to address this issue in single-stage VPR, we propose a novel mining strategy, CliqueMining, that selects positive and negative examples by sampling cliques from a graph of visually similar images. Our approach boosts the sensitivity of VPR embeddings at small distance ranges, significantly improving the state of the art on relevant benchmarks. In particular, we raise recall@1 from 75% to 82% in MSLS Challenge, and from 76% to 90% in Nordland.*

Chapter based on [17]:
Sergio Izquierdo and Javier Civera
'Close, But Not There: Boosting Geographic Distance Sensitivity in Visual Place Recognition'
*Proceedings of the European Conference on Computer Vision (ECCV)*, 2024

The code and models are available at `https://github.com/serizba/cliquemining`

Visual Place Recognition refers to identifying a place from a query image $\mathbf{I}_q \in \mathbb{R}^{h \times w \times 3}$, which boils down to retrieving the $K$ closest images $\{\mathbf{I}_1, \dots, \mathbf{I}_K\}$ from a database where they are georeferenced. VPR is fundamental in several computer vision applications. It constitutes the first stage of visual localization pipelines by providing a coarse-grain pose that reduces the search space in large image collections. This pose can be later refined by robust geometric fitting from local feature matches [8, 113]. It is also essential in visual SLAM, in which it is used to detect loop closures and remove geometric drift [13, 95], or as the basis for topological SLAM [114, 115].

In VPR pipelines, every RGB image $\mathbf{I}_i$ is typically mapped to a low-dimensional embedding $x_i \in \mathbb{R}^d$ by a deep neural network $f_\theta : \mathbf{I}_i \rightarrow x_i$ that extracts and aggregates visual features that are relevant for the task. The closest samples are retrieved by a nearest-neighbour search using distances in the embedding space $d_i^e = ||x_q - x_i||_2$, which hopefully correspond to the views with smallest geographic distance $d_i^g = ||p_q - p_i||_2$ between them, with $p_i \in \mathbb{R}^3$ standing for the camera position for $\mathbf{I}_i$. The challenge lies on learning the wide variability in the visual appearance of places, caused among others by environmental, weather, seasonal, illumination and viewpoint variability, or dynamic content. Recent years have witnessed significant advances in VPR, driven among others by enhanced network architectures [16, 29, 33, 53, 116], loss functions [32, 64, 65, 74], or two-stage re-ranking strategies [51–53, 58, 117].

In this work, we start by analyzing the Geographic Distance Sensitivity (GDS) of VPR embeddings, that can be illustrated by a plot of the distribution of embedding distances $d^e$ vs. geographic distances $d^g$, as in the centre of Figure 5.1. The plot shows two cases: in orange the distribution a typical VPR pipeline would achieve, and in blue the distribution that would be obtained by a model with enhanced GDS, result of training using our novel CliqueMining, which we will introduce later. Note how a high variance and a small slope results in a high

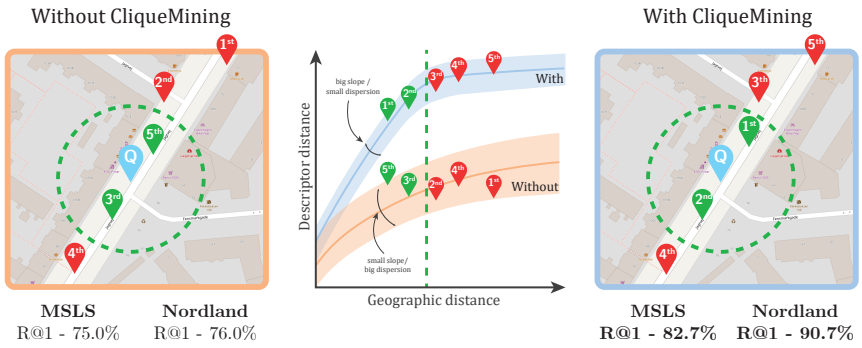



**Figure 5.1: Geographic Distance Sensitivity (GDS).** We illustrate a typical case of top-5 retrieval without (left) and with (right) our proposed CliqueMining. Note how retrievals on the left are not properly sorted based on geographic distance, impacting the recall for the selected threshold (green circle). We conceptualize this effect as GDS in the central plot, which shows the distribution of descriptor distances against geographic distances. A low slope of the mean (orange line) and a high dispersion (orange area), indicative of low GDS, raise the probability of an incorrect order. To address this, we present CliqueMining, a novel batch selection pipeline that increases the GDS of a model (blue line and area) and produces more correct retrievals.

probability of incorrectly sorting the top-5 retrievals. The top-1 retrieval on the left is, as it is written in the title, close but not there. By decreasing the variance and increasing the slope the probability of an incorrect ordering decreases.

Figure 5.2 shows this phenomenon occurring in real datasets when using the state-of-the-art baseline DINOv2 SALAD [16]. Observe how the top-5 retrievals without our CliqueMining in MSLS [39] and Nordland [76] are not properly sorted by real geographic distance. While two-stage re-ranking approaches might assist in alleviating this, their local feature matching stage come with a prohibitive storage and computational footprint. Additionally, recent methods using only global features [16, 59] already surpass those that involve local features for re-ranking. Although mining strategies also aim to improve performance by compiling informative batches during training, existing strategies are not specifically tailored to enhance GDS in densely sampled data.

In addition to analyzing GDS, in this work we propose a novel mining strategy, CliqueMining, explicitly tailored to address it. Our hypothesis is that, in order to boost the GDS, the training batches should include images of highly similar appearance at small distances, that are not explicitly searched for in current mining schemes. We achieve that by organizing our training samples as a graph from which we extract cliques that represent sets of images that are geographically close. Our experiments show that, in this way, using CliqueMining on top of a baseline model obtains substantial improvements in recall metrics.

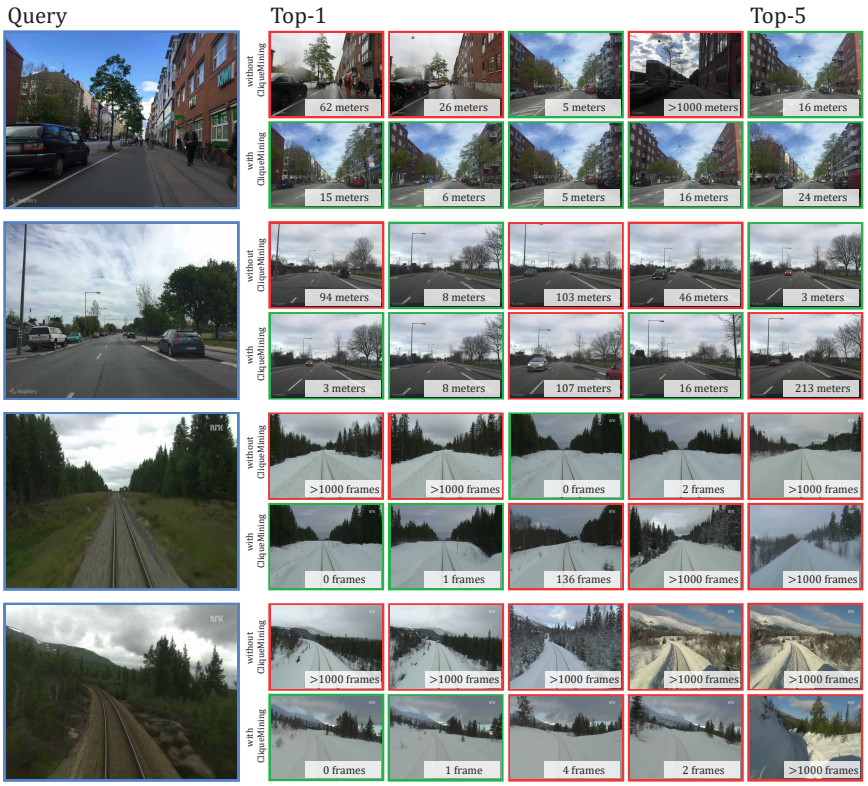



**Figure 5.2:** Top-5 retrievals for DINOv2-SALAD [16] without and with our CliqueMining in MSLS [39] and Nordland [76]. Green frames represent correct retrievals and red frames incorrect ones, under the standard 25-meters (1 frame for Nordland) decision threshold. Our CliqueMining achieves a better sorting of the retrievals with respect to their geographical distance to the query, which positively impacts the recall.

## 5.1 Analysis of Geographic Distance Sensitivity

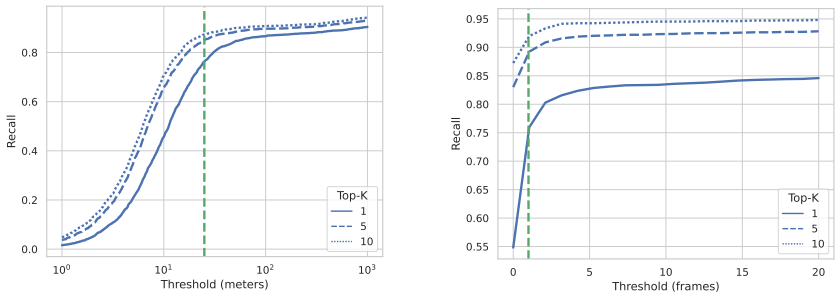
As already said, Figure 5.2 shows examples of DINOv2 SALAD [16] retrievals on MSLS Train [39] and Nordland [76]. Although the recall@1 for these specific queries is zero, dismissing the model's performance as entirely inaccurate would be unfair. Within the top-5 retrievals, some predictions are indeed correct, and most incorrect predictions are relatively close to the decision threshold. These examples uncover a common issue in VPR models: their inability to finely discriminate between similar viewpoints. Note how our novel CliqueMining, that we will describe in next sections, discriminates better for this particular case.

We explain this phenomenon using the concept of GDS, *i.e.*, the model's ability to assign smaller descriptor distances to pairs of images that are geographically closer. VPR models should have a high GDS, that is, they should produce descriptors that maximize the probability $P(d_i^e < d_j^e \mid d_i^g < d_j^g)$. Seeking for a high GDS requires two desiderata to hold.

(i) The expected value of the descriptor distance of a pair should be smaller than that of a pair geographically further from the query $\mathbb{E}[d_i^e - d_j^e \mid d_i^g < d_j^g] < 0$.

(ii) The dispersion of descriptor distances conditioned on a certain geographic distance should be as small as possible $\mathbb{E}[(d_i^e - \mathbb{E}[d_i^e \mid d_i^g])^2 \mid d_i^g] \to 0$.

Failing to achieve these two leads to a high probability of retrieving an incorrect order of candidates. We hypothesize that VPR models struggle to precisely rank between closely spaced locations due to their limited GDS at small distance ranges. This is because current training pipelines are effective at achieving highly invariant representations that encode viewpoints coarsely, but not at learning the subtle cues to disambiguate between close frames.




**(a)** MSLS Train (Val Subset)

**(b)** Nordland

**Figure 5.3: Recall@K vs. decision threshold** on MSLS Train (val) and Nordland for DINOv2-SALAD [16] without CliqueMining. Observe how the steep curve around the decision threshold (green dashed line) indicates a significant number of closely retrieved images. Boosting the GDS of a model would alleviate this, increasing its recall.

This effect can be further assessed in Figure 5.3, which shows the top-{1, 5, 10} recall of the baseline DINOv2 SALAD for different threshold values. The vertical green dashed lines represent the typical thresholds of 25 meters and 1 frame used in MSLS and Nordland. Note how the recall, specially the recall@1, keeps increasing for slightly larger values than the 25 meters and 1 frame thresholds. This indicates that a significant fraction of false negatives is very close to the decision threshold, which lowers the recall.

With our novel CliqueMining strategy, detailed in next section, the reader will assess how we are able increase the GDS for small ranges (Fig. 5.5) and consequently improve recall metrics, as we will show in the experimental results.

## 5.2 Method: CliqueMining

Our novel mining strategy, CliqueMining, selects challenging batches according to geographic and descriptor similarity criteria, alleviating the GDS issues identified in Section 5.1. Figure 5.4 shows an overview of our method. To effectively mine a challenging batch, we first build a graph of image candidates (Subsection 5.2.1) and sample places from it (Subsection 5.2.2). Finally, we select challenging pairs and train the network using the Multi-Similarity (MS) loss (Subsection 5.2.3).

### 5.2.1 Graph Creation

In contrast with the sparse nature of viewpoint sampling in GSV-Cities [33], we propose to use denser batches, with higher spatial continuity, so the the network also learns the subtle changes resulting from small camera motion. To effectively mine such challenging batches, we first create a graph, $G = (V, E)$, representing a cluster of candidates. Vertices from this graph, $v_i \in V$, are frames from sequences with very similar appearance, and two vertices, $v_i$ and $v_j$, are connected by an edge $e_{ij} \in E$ if both frames lie within a given distance threshold in meters, $\tau$.

$$E = \{e_{ij} \mid d(v_i, v_j) < \tau,\ \forall v_i, v_j \in V\} \tag{5.1}$$

To populate the graph, we consider all image sequences as defined in the MSLS training set, as our place-based batches do not require a split between query and database images. We start by sampling a reference sequence from a city, $s_{ref}$, and subsequently, sampling $S$ more different sequences, $\{s_1, \ldots, s_S\}$ based on their similarity with $s_{ref}$. For computational efficiency, we determine the

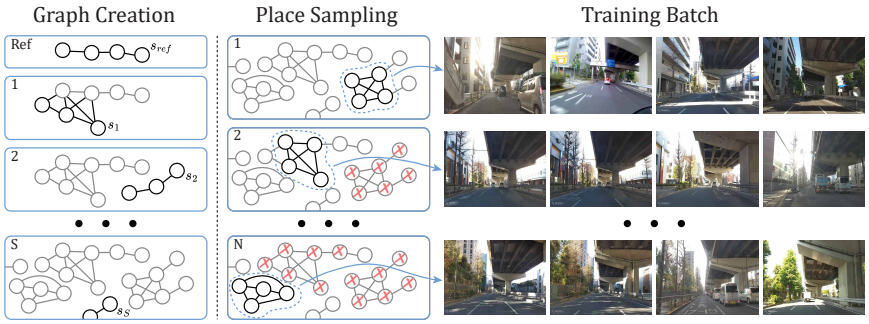



**Figure 5.4: Overview of CliqueMining.** First, we create a graph of candidates by sampling a set of sequences $\{s_1, \ldots, s_S\}$ that are similar to a reference one $s_{ref}$ (left). We then sample places by finding cliques within the graph (center). Observe that the resulting batches contain very similar looking places, which boost the GDS (right).

similarity between two sequences by only comparing the descriptors of their respective central frames. We incorporate every frame from these sequences into the graph, which ensures the presence of adjacent frames within the batches. Edges are determined by the Universal Transverse Mercator (UTM) locations of each frame. Algorithm 1 summarizes this process.

**Algorithm 1** Graph creation.

Initialize $G = (V, E)$ as empty graph
Sample $city$
$V \leftarrow \{v_i | v_i \in s_{ref}\},\ s_{ref} \sim \{s | s \in city\}$
**repeat** S **times**
  $s \sim P(s|s_{ref}) \propto sim(s, s_{ref})$
  $V \leftarrow V \cup \{v_j | v_j \in s\}$
**end**
$E \leftarrow \{e_{ij} | d(v_i, v_j) < \tau, \forall v_i, v_j \in V\}$

### 5.2.2 Place Sampling

To construct a single batch, we start from the graph of candidates $G$, generated as explained in Subsection 5.2.1. $G$ is a convenient representation for place sampling, as it facilitates the identification of distinct viewpoints yet of highly similar appearance, and labels are easily assigned based on connectivity. In our pipeline, we mine batches of $N$ places, each place defined as a set of $K$ images, where each image is within a range $\tau$ of each other. Sampling a place is equivalent to finding a clique, $C$, within $G$

$$C \sim \{C \mid \forall v_i, v_j \in C,\ e_{ij} \in E,\ C \subseteq V,\ |C| = K\}. \tag{5.2}$$

Thus, to compile a batch of $N$ places, we iteratively extract $N$ cliques from $G$. After finding each clique, all its frames, as well as their connected vertices are removed from $G$. This prevents overlap in subsequent cliques, ensuring that each sampled place is at least $\tau$ meters from each other. In the uncommon case of exhausting all cliques in $G$, we create a new graph starting from a new $s_{ref}$ and continue the process. The resulting batches, an example of them shown in Figure 5.4, showcase highly similar yet far apart images, illustrating the effectiveness of our sampling to create difficult batches. Algorithm 2 gives an overview of the sampling procedure.

**Algorithm 2** Graph sampling.

Input: Graph $G = (V, E)$
Initialize empty batch of images $B$
Initialize empty batch of labels $L$
**for all** $n$ from 1 to $N$ **do**
  Sample clique $C \sim \{C \mid \forall v_i, v_j \in C,\ e_{ij} \in E,\ C \subseteq V,\ |C| = K\}$
  $B \leftarrow B \cup C$
  $L \leftarrow L \cup \{n\}$
  $G \leftarrow G - \{v_i \cup Adj(v_i) | v_i \in C\}$
**end for**

### 5.2.3 Training Pipeline

In practice, we mine a large set of batches offline and once, as described in Subsection 5.2.1 and Subsection 5.2.2, and use them during all epochs. To do this, we use the embeddings from a model pre-trained without CliqueMining. Most mining strategies are typically updated every few iterations. However, this increases the computational overhead, and for our CliqueMining we did not observe any improvement by updating the batches.

In order to smooth the gradients from our hard training images, we combine them with images from GSV-Cities. In this manner, we include per batch half of the images from our CliqueMining and half from GSV-Cities, so the network can learn both the fine-grain GDS and the sparse discriminative capabilities from GSV-Cities.

As we use the MS loss [74], during training we use their online selection method for weighted negative and positive pairs. A negative pair, $\{x_i, x_j\}$, is selected from a batch if its distance is lower than the hardest positive pair plus a margin, $\epsilon$,

$$||x_i - x_j||_2 \ < \max_{d^e_{ik} < \tau} ||x_i - x_k||_2 + \epsilon, \tag{5.3}$$

and, conversely, a positive pair is selected when

$$||x_i - x_j||_2 \ > \min_{d^e_{ik} \geq \tau} ||x_i - x_k||_2 - \epsilon. \tag{5.4}$$

## 5.3 Experiments

In this section, we re-train state-of-the-art VPR baseline models using our proposed CliqueMining. Evaluation on various benchmarks showcases the increased discriminative capacity of the models. In the following, we describe the implementation details, benchmarks used, quantitative and qualitative results, as well as ablation studies.

### 5.3.1 Implementation Details

We use CliqueMining with the recent DINOv2 SALAD [16], the current state-of-the-art VPR model as well as on MixVPR [29], a recent model with competitive performance. For each of them, we use their codebase and rigorously follow their training pipelines and hyperparameters. We use batches of size 60 in DINOv2 SALAD and 120 in MixVPR, where half of the places come from our pipeline and the other half from GSV-Cities. We create a new graph for every

| Method | NordLand | | | MSLS Challenge | | | MSLS Val | | | Pitts250k-test | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| NetVLAD [5] | 32.6 | 47.1 | 53.3 | 35.1 | 47.4 | 51.7 | 82.6 | 89.6 | 92.0 | 90.5 | 96.2 | 97.4 |
| GeM [30] | 21.6 | 37.3 | 44.2 | 49.7 | 64.2 | 67.0 | 78.2 | 86.6 | 89.6 | 87.0 | 94.4 | 96.3 |
| CosPlace [6] | 52.9 | 69.0 | 75.0 | 67.5 | 78.1 | 81.3 | 87.6 | 93.8 | 94.9 | 92.3 | 97.4 | 98.4 |
| MixVPR [29] | 58.4 | 74.6 | 80.0 | 64.0 | 75.9 | 80.6 | 88.0 | 92.7 | 94.6 | 94.6 | 98.3 | 99.0 |
| EigenPlaces [107] | 54.4 | 68.8 | 74.1 | 67.4 | 77.1 | 81.7 | 89.3 | 93.7 | 95.0 | 94.1 | 98.0 | 98.7 |
| SelaVPR (global) [116] | 47.2 | 66.6 | 74.1 | 69.6 | 86.9 | 90.1 | 87.7 | 95.8 | 96.6 | 92.7 | 98.0 | 98.9 |
| SelaVPR (re-ranking) [116] | 60.0 | 75.7 | 79.6 | 73.5 | 87.5 | 90.6 | 90.8 | 96.4 | 97.2 | **95.7** | **98.8** | 99.2 |
| DINOv2 SALAD [16] | 76.0 | 89.2 | 92.0 | 75.0 | 88.8 | 91.3 | 92.2 | 96.4 | 97.0 | 95.1 | 98.5 | 99.1 |
| MixVPR [29] CM | 69.6 | 80.7 | 83.5 | 65.6 | 77.1 | 79.2 | 88.8 | 93.9 | 94.6 | 91.8 | 96.7 | 98.1 |
| DINOv2 SALAD [16] CM | **90.7** | **96.6** | **97.5** | **82.7** | **91.2** | **92.7** | **94.2** | **97.2** | **97.4** | 95.2 | **98.8** | **99.3** |

**Table 5.1: Comparison against single-stage baselines and SelaVPR as representative of two-stage baselines.** Observe the significant increase in the recall in MSLS and Nordland when using CliqueMining (CM). Both are the less saturated datasets, hence with most room for improvement, and the most densely sampled, which is the case our novel CliqueMining is tailored for.

batch. We start by sampling $s_{ref}$ from the set of existing sequences. We then sample $S = 15$ sequences from the same city based on the descriptor similarity of their central frames. Edges are assigned with $\tau = 25$. Cliques are searched using the NetworkX library[1] using the unrolled algorithm by Tomita *et al.* [118]. We create offline a large collection of 4000 batch examples before starting the training, and at every iteration, we randomly select one of those. To create the batches we use all the non panoramic images in the MSLS Training set. For the ablation studies we divided this dataset in val and train subsets, setting Melbourne, Toronto, Paris, Amman, Nairobi and Austin for val and the rest 16 cities for train.

1: https://networkx.org/

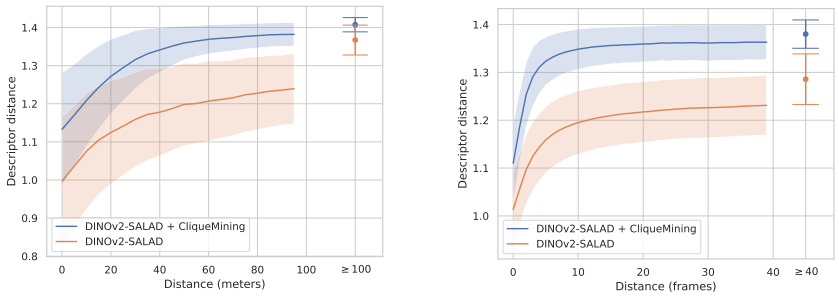
### 5.3.2 Results

We evaluate the effect of our CliqueMining by comparing the performance of two recent high-performing models, DINOv2 SALAD [16] and MixVPR [29], with and without it at training time. We also benchmarked these against classic methods, namely NetVLAD [5] and GeM [30], and recent performant baselines, specifically CosPlace [6], EigenPlace [107], and SelaVPR [116]. Additionally, we include in the comparison results of SelaVPR [116] with re-ranking, as it is the current state of the art among two-stage techniques.

We report results on standard evaluation datasets. **Nordland** [76] is a continuous video sequence taken from a train traveling through Norway across different seasons. The difficulty of this dataset arises from the substantial appearance differences between query (summer) and reference (winter), as well as the dense temporal sampling. **MSLS Challenge and Validation** [39] is a large and dense collection of dashcam images recorded in cities around the globe. The various seasonals, time, and environmental changes depicted make it one of the least saturated datasets in VPR. **Pittsburgh-250k** [106] is known for its significant viewpoint changes, but current pipelines have highly saturated performance.

As previous works, we report recall@$\{1, 5, 10\}$, which measures the rate of correct predictions among the top-$\{1, 5, 10\}$ retrieved images. An image is considered correct if it lies within a 25 meters-radius circle from the query, or at most one frame apart for the Nordland dataset. Results are reported on Table 5.1.

On Nordland, training with our CliqueMining significantly improves both DINOv2 SALAD and MixVPR, obtaining, for the first time, a recall@1 bigger than 90% (+14.7% over the closest baseline). This milestone highlights how our hard batches help in boosting the network's GDS. This is a crucial aspect in




**(a)** MSLS Train (Val Subset)

**(b)** Nordland

**Figure 5.5: Mean ± standard deviation of descriptor distances against geographic distances, without and with CliqueMining.** Our Clique Mining boosts the geographic local sensitivity for small geographic distances, and flattens it for large distances. This results in higher discriminativity around the decision threshold and better metrics. Note the cut in distances and values for high distances aggregated at the right part.

Nordland, where the high similarity between video frames and the strict one-frame distance threshold need outstanding sensitivity. Note that CliqueMining also improves significantly the recall rates for MixVPR.

On MSLS Challenge and Validation, our CliqueMining with the DINOv2 SALAD architecture improves over all previously reported results. The improvement is most notable on the Challenge, where CliqueMining raises +7.7% the recall@1. While training on the MSLS Train dataset contributes to these results, it is noteworthy that SelaVPR, which also trains on MSLS, does not achieve a comparable performance, even with re-ranking. The effect of CliqueMining on MixVPR is dimmer, although it also improves over the baseline without it. We argue that its global aggregation smooths out local details, which are critical for raising the GDS.

On Pittsburgh-250k, our pipeline obtains a slight improvement over the baseline DINOv2 SALAD and obtains comparable performance to SelaVPR with re-ranking. We outperform SelaVPR without re-ranking, which is a more comparable baseline. Note, in any case, that SelaVPR is fine-tuned on Pittsburgh30k before testing on Pittsburgh250k, while ours was trained in GSV-Cities and MSLS. MixVPR with CliqueMining downgrades performance. Training on MSLS data, where almost all images are forward-facing, has a small impact on Pittsburgh250k, which exhibits substantial viewpoint variability.

Note how we sorted the datasets in Table 5.1 from more to less image density, and how this also sorted naturally the recall@1 gains of CliqueMining from bigger to smaller. This supports our observation that GDS issues are more relevant the higher the image density, and that CliqueMining is able to improve them. From these results we can also conclude that a substantial part of the challenge in the less saturated VPR datasets (Nordland and MSLS) is associated to GDS issues, which is a relevant insight.

Observe in Figure 5.5 the effect of CliqueMining on the GDS of the DINOv2-SALAD model [16] in MSLS and Nordland, as a plot of the distribution of the pairwise descriptor distances for different geographic distances. As sought, the GDS is highly boosted (steep curve and low dispersion) by CliqueMining for close geographic distances. Observe the similarity of this result with the illustrative graph in Figure 5.1. Although not specifically tailored for, CliqueMining also reduces the dispersion for large distances, probably due to leveraging batches with more informative gradients. This enables the model to correctly sort candidates that are near, and still discriminate from those too far apart.

We finally remark the low computational footprint of our CliqueMining. CliqueMining is a mining strategy for training, and hence does not increase

at all the computational footprint at inference. This is in contrast to two-stage methods, that increase it by a factor of several orders of magnitude. Additionally, the overhead is modest at training. Our ablations shows that the graph creation only needs to be done once before training, and there is no benefit in updating it. In total, the computational overhead of CliqueMining roughly amounts to only 20% of the total training time in our experiments.

### 5.3.3 Ablation Study

| Method | MSLS Train (Val Subset) | | |
|---|---|---|---|
| | R@1 | R@5 | R@10 |
| DINOv2 SALAD [16] | 76.3 | 85.1 | 87.3 |
| Most-similar | 81.61 ± 0.50 | 89.43 ± 0.53 | 91.02 ± 0.53 |
| Weighted random sampling | 81.98 ± 0.75 | 89.72 ± 0.22 | 91.12 ± 0.14 |
| Uniform random sampling | 80.40 ± 0.70 | 87.33 ± 0.88 | 88.95 ± 0.70 |
| W/o MS mining | 76.87 ± 0.46 | 83.92 ± 0.60 | 86.05 ± 0.76 |
| Naïve GSV-Cities + MSLS | 79.96 ± 0.46 | 89.71 ± 0.32 | 91.80 ± 0.30 |
| Recompute Cliques | 81.96 ± 0.59 | 89.64 ± 0.54 | 91.28 ± 0.39 |
| | Nordland | | |
| | R@1 | R@5 | R@10 |
| DINOv2 SALAD [16] | 76.0 | 89.2 | 92.0 |
| Weighted random sampling | 88.22 ± 0.99 | 95.22 ± 0.45 | 96.52 ± 0.38 |
| Naïve GSV-Cities + MSLS | 68.27 ± 5.47 | 82.92 ± 4.99 | 86.81 ± 4.36 |

**Table 5.2: Ablations.** First row shows the recall for the base DINOv2-SALAD model. Note in the next three rows that random sampling based on sequence similarity outperforms slightly a determininistic sampling of the most similar ones and some more a uniform random sampling. The MS mining also plays a role in the performance. Note how training on GSV-Cities + MSLS w/o CliqueMining, which accounts for the domain change effect, still underperforms at R@1. Finally, note that recomputing cliques every epoch gives metrics that are similar to computing them only once.

We conduct evaluations with different configurations of CliqueMining to assess the importance of its different components. We base all our ablation studies on the DINOv2 SALAD baseline.

**CliqueMining or training on more data.** One of the key contributions of this work is to train state-of-the-art models on a combination of GSV-Cities and MSLS. This raises the question of whether the observed improvements result from training with more data or from CliqueMining. To evaluate this, we re-train DINOv2 SALAD on a combination of GSV-Cities + MSLS without CliqueMining. Thus, batches from MSLS are organized in triplets as usually done in the literature. Table 5.2 shows how, although training on MSLS slightly increases performance, using CliqueMining produces the best results, specially for R@1. We also report, for this ablation, results on Nordland which show more pronounced differences with CliqueMining. This suggest that naïvely training on more data brings limited improvements. CliqueMining creates challenging batches that improve the sensitivity of the model and its recall. Besides, CliqueMining organizes the images in places, so every image can simultaneously act as an anchor, positive or negative, increasing the number of pairwise relations on a batch.

**Geographic distance threshold $\tau$.** We tested the effect of the $\tau$ values in the range 10-30. As shown in Figure 5.6, using the typical decision threshold value $\tau = 25$ achieves the best performance.

**MS mining**. We built our CliqueMining on top of [33], keeping its online mining (Equations 5.3 and 5.4). Deactivating it, keeping only our CliqueMining, has a detrimental effect (see Table 5.2), which indicates that both mining strategies are compatible.

**Sequence sampling.** We evaluate the effect of different sampling strategies to obtain $\{s_1, \ldots, s_S\}$ during the graph creation. We specifically try a weighted

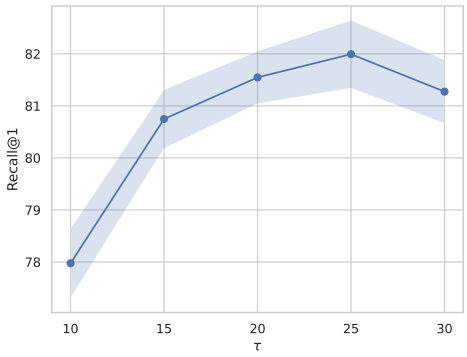



**Figure 5.6: Recall@1 on MSLS Train (val) for different values of $\tau$.** Note how, reasonably, $\tau = 25$ meters, which is equal to the decision threshold, is the best value.

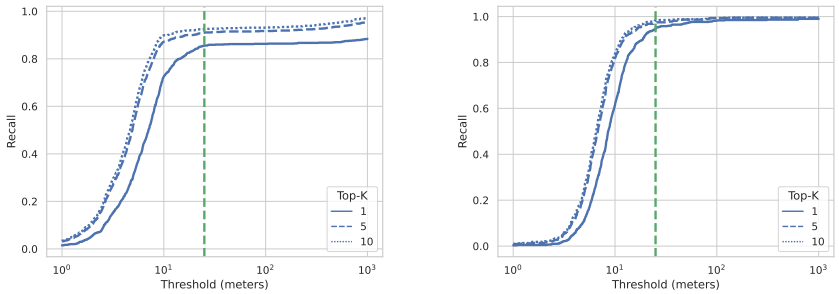
sampling according to similarity, selecting the top $S$ most similar sequences, or randomly. Table 5.2 shows that all three sampling strategies obtain very similar results, but using the most similar sequences produces the best. We argue that the online mining from Equations 5.3 and 5.4 reduces the actual differences between the used selection criteria, as it will further select the hardest pairs. Besides, given the length of some of the sequences, more than one clique might be sampled from the same sequence, reducing the need to find other similar ones.

**Updating the mining every epoch.** Commonly done in literature, updating the mining after every epoch using the recently updated weights can provide some benefits to performance. As shown in Table 5.2, obtained recalls are comparable, and computing the mining after every epoch is computationally expensive.

## 5.4 Limitations

The main limitation of CliqueMining is that it is specifically tailored for VPR, and hence it will not be of use for general image retrieval. In addition, CliqueMining addresses GDS issues, that are mostly relevant for places that are densely sampled with images. We already reported in Table 5.1 the diminishing returns as the sampling density decreases in the benchmarks we used. However, this limitation is softened by the wide range of potential use cases falling into this condition, and also by the remarkable boost in recall@1 in the most dense sampling cases (+14.7% for Nordland).

Additionally to the above, our CliqueMining is strongly dependent on the existence of GDS issues. Even if the dataset is densely sampled, there could be




**(a)** SF-XL test v1

**(b)** SF-XL test v2

**Figure 5.7: Recall@K vs. decision threshold on SF-XL** for DINOv2-SALAD [16] without CliqueMining. Observe how the recall curves are almost flat beyond the decision threshold (green dashed line), indicating a low false negative rate due to limited GDS. Compare it against the recall curves in MSLS and Nordland in Figure 5.3. In this case, enhancing the GDS will not result in better metrics.

a lack of GDS issues, as when viewpoint changes account for the majority of variations. In this cases, the model fails to retrieve close samples, and therefore CliqueMining would not positively impact its recall. We observed this in the recent SF-XL [6], a massive dataset of images from San Francisco, often used to test VPR at scale. Figure 5.7 characterizes the recall in this dataset against the decision threshold. Observe how, in contrast to Figure 5.3, the recall is almost flat in the region immediately after the decision threshold. Enhancing the GDS is not expected to have any effect in this dataset, as the rate of false negatives due to this reason is very small. Even if this is a limitation, we would argue in our favour that every mining strategy is strongly dependent on the data, but in the case of our CliqueMining we have characterized the conditions in which it should or should not offer an improvement.

## 5.5 Conclusions

In this chapter we have identified, formulated and analyzed deficiencies in the GDS of current VPR models. Specifically, we found that they struggle to correlate descriptors and geographic distances for close range views. Based on that, we propose CliqueMining, a tailored batch sampling that selects challenging visually similar places at close ranges, and in particular around the decision threshold. CliqueMining forces the model to incorporate a finer grading of the geographic distances in the embedding. Mining such hard batches is equivalent to finding cliques in a graph of similar image sequences where connectivity represents spatial proximity. Our evaluation of two recent models with and without CliqueMining confirms a boost in the GDS which in turn also boosts the recall. The boost is substantial on densely sampled and unsaturated benchmarks like MSLS Challenge or Nordland, where training with CliqueMining brings unprecedented results.

# II

# Multi-View Cues in Depth Estimation

# 6 Motivation and Contributions

Más ven cuatro ojos que dos.
*Four eyes see more than two.*

*-Spanish proverb*

In our daily lives, 3D perception and reasoning occur continuously, allowing us to interact seamlessly with our environment every time we navigate complex spaces, avoid obstacles, or manipulate objects. Humans instinctively enhance this spatial awareness by leveraging multiple perspectives—adjusting our viewpoints, moving around, and leveraging our stereo vision to gather depth information at close ranges.

These innate human abilities have long been sought after in the fields of robotics and computer vision. The capacity to perceive and reason in 3D is crucial for machines to perform tasks analogous to those humans do effortlessly, like obstacle avoidance in autonomous driving [119, 120], dexterous manipulation in robotics [121], or 3D layout recovery in augmented reality [122]. It is of special interest to perform such tasks with just visual data, as such can be obtained using cheap, readily available cameras, eliminating the need for more expensive sensors.

One of the fundamental tasks in 3D perception is depth estimation. Depth estimation involves determining the distance of objects from a particular viewpoint, effectively creating a depth map of the scene. An appealing and widespread approach is to rely solely on single images [10–12], training deep learning models on vast datasets to infer dense depth maps based on visual clues. This method benefits from the versatility and simplicity of requiring just one single image, without more expensive setups, which is advantageous for both training and inference. However, it faces the challenge of being an ill-posed configuration, where multiple depth hypotheses may correspond to the same 2D image, leading to potential inaccuracies and ambiguities.

For this reason, just as humans seek multiple points of view to enhance their perception, multi-view depth estimation techniques utilize several images to improve accuracy. In multi-view stereo, features across images are matched to triangulate depth, providing accurate, consistent, and scaled 3D information [123–125]. While these approaches benefit from geometric constraints and can offer higher accuracy, they require precise knowledge of camera poses, are more difficult to train, and can struggle with non-Lambertian surfaces or changes in illumination, which affect feature matching.

This part of the thesis focuses on leveraging multi-view cues in dense depth estimation. In Chapter 8, we enhance single-view models on sequences of images by doing a TTR. By using a SfM reconstruction as pseudo ground truth, the models are able to predict more accurate guesses, especially at large distances. In Chapter 9, we propose a general-purpose large model for multi-view depth estimation. Drawing inspiration from recent advancements in single-view depth prediction, we train a ViT on a varied array of datasets. Aiming to overcome some of the limitations of previous multi-view systems, our proposed model can work with any range of depths without an initial guess, can handle dynamic objects, and has strong generalization performance.

# 7 Related Work

In this chapter we discuss related literature about single-view depth (Section 7.1), multi-view depth (Section 7.2) and test-time-refinement (Section 7.3).

## 7.1 Single-View Depth Learning

Although there exists a large corpus of work on single-view depth under certain assumptions on the scene geometry, e.g. [126–131], we focus here on approaches that are mainly based on machine learning and target general scenes.

### 7.1.1 Supervised Methods

Several early works addressed single-view depth learning either directly from the image [132] or via semantic labels [133] before the deep learning era. The seminal works by Eigen et al. [10, 134] significantly improved the prediction accuracy by training deep networks supervised with ground-truth depth from range sensors. Since then, single-view depth networks have received significant attention from the research community, focusing on improving the performance by using more sophisticated architectures and losses, e.g., [135–141]. A re-formulation of the problem as an ordinal regression has led to further improvement [11, 142, 143]. Recently, Bae et al.[144] fuse the single-view depths from multiple images, but differently from our method described in Chapter 8 without a TTR of the network.

Building on highly advanced and effective image backbones [54], more recent monocular methods have focused on making *general-purpose* depth estimation models, which aim to work on arbitrary scenes [145, 146]. Further works have scaled up the size of models and datasets, training on combinations of real and/or synthetic data [147, 148], and have used stronger image-level priors [149, 150]. One of the limitations of models trained from stereo-image-derived supervision without known baselines [146] or human annotations [145] is that these only enable a relative, and not metric (*e.g.* in meters), depth prediction.

Other monocular models predict *metric* depth [12, 151–154]. Not only does this rely on appropriate training data, but also requires an understanding of camera intrinsics, which are often a required additional input to the network.

Conventional monocular methods are inherently limited by only incorporating information from single views at inference time, even when multi-view information is available [155]. On the other hand, with recent advances, they can still provide a very valuable signal when only one image is available. As in [156–158], in Chapter 9, we combine features extracted from a monocular depth model with a multi-view cost volume to better leverage monocular and multi-view cues.

### 7.1.2 Self-supervised Methods

As ground truth depth annotations are uncommon, self-supervised approaches emerged as an alternative, exploiting multi-view photometric consistency [159, 160]. Attracted by the convenience of training without depth labels, many works have further focused on addressing this paradigm, e.g., [161–167]. Close to our work from Chapter 8, SfM has been used as a supervisory signal during training, but limited to probabilistic networks [168], or using disparities [169] that require stereo images. Among self-supervised works, Monodepth2 [170], which proposed a robust loss to handle occlusions and discard invalid pixels, is of particular relevance. Monodepth2 is the base of most state-of-the-art approaches, and specifically of the baselines we chose to validate our refinement on: CADepth [171], that uses self-attention to capture more context, DIFFNet [172], that applies feature fusion to incorporate semantic information, and Many-Depth [155], that leverages more than one frame at inference to improve the predictions.

## 7.2 Multi-View Depth Learning

Multi-View Stereo (MVS) algorithms estimate depth from posed multi-view images using epipolar geometry [173]. Given calibrated cameras, early methods estimated depth by matching image patches [174, 175]. Subsequently, deep learning approaches were introduced, first for stereo matching [176] and later improved via end-to-end learning, typically using plane-sweep cost volumes [123, 124, 177–183]. Subsequent methods introduced advances in architectures [125, 184, 185], increased robustness to occlusion and moving objects [186–188], integrated temporal information [189], improved model efficiency [190, 191], jointly estimated camera pose [192, 193] and ingested prior geometry estimates to improve depths [194].

### 7.2.1 Generalization to Unseen Domains

With some exceptions [195], earlier stereo and MVS methods were traditionally both trained and tested on the same dataset/domain, and were limited in their ability to generalize to out-of-distribution data. This domain generalization issue is a consequence of most performant learning-based MVS methods being data-hungry. Approaches such as training on synthetic [196, 197] or pseudo-labeled depth [198] can be effective, but so far, struggle to span a diverse range of scene types and scales. Self-supervised approaches can be trained without depth supervision, but current methods produce inferior depths compared to fully supervised approaches [199–201]. Concurrent with our work, [202, 203] trained large *binocular* stereo models on large synthetic datasets.

### 7.2.2 Adaptive Cost Volumes.

One of the challenges in developing a general-purpose domain-agnostic MVS method is that different scenes can contain wildly different depth ranges, *e.g.* indoor scenes are limited to a few meters, while outdoor ones can span much larger distances. This is a problem as conventional cost volumes require a known depth range, which is typically just estimated based on the minimum

and maximum depth values in the training set. As a result, there is a need for cost volumes that are not restricted to a pre-defined range or bins, and instead are adaptive. In the context of self-supervised learning with unscaled poses, [155] estimated bin ranges at training time via an exponential moving average of the depth predictions. Another approach is to predict bin centers iteratively in a coarse-to-fine manner, where the outputs from the previous iteration are used to seed the range in the next [181, 204, 205]. Alternatively, the bin offsets can be predicted by a learned network [206] or from estimated depth uncertainty [207, 208]. In Chapter 9, we estimate cost volume depth ranges to enable us to adapt to any range of depths, while prior work has done this when the test time range is known, but they wish to reduce computation or enhance detail.

## 7.3 Test-Time Refinement

In Chapter 8, we employ a TTR of the networks. Here we describe previous attempts and literature on the field.

Multi-view consistency is the basis for both self-supervised depth learning and bundle adjustment [209], this last one naturally occurring at test time. Inspired by that, TTR was proposed [210, 211], updating the network with the same self-supervised losses from training. Similarly, McCraith et al. [212] showed the benefits of encoder-only fine-tuning and proposed two TTR modes: sequence- and instance-wise. Similar approaches were presented by Watson et al. [155], with multiple input images for the network, Shu et al. [166], with a feature-metric loss, and Kuznietsov et al. [213], using a replay buffer. All these TTR methods inherit the small baseline limitations from photometric losses, showing small improvements for medium and large depths for which close views produce small parallax. At these depths, our method from Chapter 8 introduces wide baseline cues, due to the higher invariance of features matching at wide baselines. This leads to significant improvements over the state of the art.

Tiwari et al. [214] iterates over optimizing the parameters of a single-view depth network and running pseudo-RGBD SLAM for pose estimation, but their alignment ignores the depth distributions, which results in smaller improvements compared to ours. Luo et al. work [215] is more related to ours, using SfM and optical flow as geometric constraints. However, despite heavy optimization (taking up to 40 minutes for a sequence of less than 250 frames), their TTR cannot improve over baseline networks on KITTI. Instead of defining derived constraints, we directly optimize the encoder using the sparse reconstruction as pseudo ground truth, resulting in a lighter and more effective pipeline.

# 8 Structure from Motion and Depth Networks



*Estimating a dense depth map from a single view is geometrically ill-posed, and state-of-the-art methods rely on learning depth's relation with visual appearance using deep neural networks. On the other hand, SfM leverages multi-view constraints to produce very accurate but sparse maps, as matching across images is typically limited by locally discriminative texture. In this work, we combine the strengths of both approaches by proposing a novel test-time refinement method, denoted as SfM-TTR, that boosts the performance of single-view depth networks at test time using SfM multi-view cues. Specifically, and differently from the state of the art, we use sparse SfM point clouds as test-time self-supervisory signal, fine-tuning the network encoder to learn a better representation of the test scene. Our results show how the addition of SfM-TTR to several state-of-the-art self-supervised and supervised networks improves significantly their performance, outperforming previous TTR baselines mainly based on photometric multi-view consistency.*

Chapter based on [15].
Sergio Izquierdo and Javier Civera
'SFM-TTR: Using Structure from Motion for Test-Time Refinement of Single-View Depth Networks'
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023

The code is available at `https://github.com/serizba/SfM-TTR`

Obtaining accurate and dense depth maps from images is a challenging research problem and an essential input in a wide array of fields, like robotics [216], AR [215], endoscopy [217], or autonomous driving [120]. Single-view per-pixel depth estimation is even more challenging, as it is geometrically ill-posed in the general case. However, in the last decade, intense research on deep models applied to this task has produced impressive results, showing high promise for real-world applications.

Single-view depth learning was initially addressed as a supervised learning problem, in which deep networks were trained using large image collections annotated with ground truth depth from range (e.g., LiDAR) sensors [134, 135]. At present, this line of research keeps improving the accuracy of single-view depth estimates by better learning models and training methods, as illustrated for example by [11, 218].

In parallel to improving the learning side of the problem, several works are incorporating single- and multi-view geometric concepts to depth learning, extending its reach to more general setups. For example, [219, 220] propose camera intrinsics-aware models, enabling learning and predicting depths for very different cameras. More importantly, many other works (e.g. [170]) use losses based on multi-view photometric consistency, enabling self-supervised learning of depth and even camera intrinsics [221].

Incorporating single- and multi-view geometry into depth learning naturally links the field to classic research on SfM [175, 222], visual odometry [223, 224] and visual SLAM [13, 95]. These methods typically produce very accurate but sparse or semi-dense reconstructions of high-gradient points using only multi-view geometry at test time. Among the many opportunities for cross-fertilization of both fields (e.g., using depth networks in visual SLAM [225] or SfM for training depth networks [168, 169, 226]), our work focuses on using SfM for refining single-view depth networks at test time.

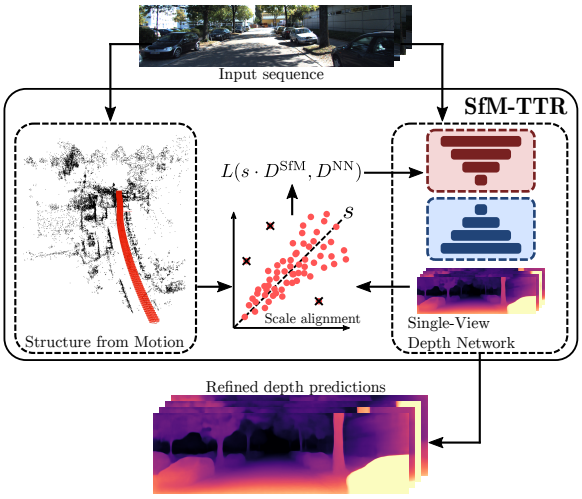



**Figure 8.1: SfM-TTR overview.** Our approach assumes an existing pre-trained depth network and an input sequence at test time. We estimate a SfM 3D reconstruction using the input sequence, and depth maps using a single-view depth network. We align the SfM point cloud with the network's depth to obtain a pseudo-ground truth to refine the network encoder, improving its representation of the test scene and producing significantly more accurate depth estimates.

As single-view depth applications typically include a moving camera, several recent works incorporate multiple views at inference or refine single-view depth networks with multi-view consistency cues [144, 155, 166, 210, 212, 214, 215]. Most approaches, however, rely mainly on photometric losses, similar to the ones used for self-supervised training. These losses are limited to be computed between close views, creating weak geometric constraints. Our contribution in this chapter is a novel method that, differently from the others in the literature, uses exclusively a SfM reconstruction for TTR. Although SfM supervision is sparser than typical photometric losses, it is also significantly less noisy as it has been estimated from wider baselines. Our results show that our approach, which we denote as SfM-TTR, provides state-of-the-art results for TTR, outperforming photometric test-time refinement (Ph-TTR) for several state-of the-art supervised and self-supervised baselines.

## 8.1 Method: SfM-TTR

Our SfM-TTR takes *any* single-view depth network, trained either supervised or self-supervisedly, and fine-tunes it for the test data by a three-stage process. As a brief summary, we first estimate a sparse feature-based reconstruction of the scene from multiple views (Subsection 8.1.1) and predict depth outputs with the network (Subsection 8.1.2). Then, we align the scale of the sparse point cloud and the network's depth (Subsection 8.1.3). Finally, we fine-tune the network using the depths of the aligned sparse point cloud as supervisory signal (Subsection 8.1.4).

### 8.1.1 Multi-View Depth from SfM

We perform a 3D reconstruction of the target scene using an off-the-shelf SfM algorithm. In our current implementation we use COLMAP [175], as it shows a high degree of accuracy and robustness in a wide variety of scenarios, although alternative SfM or visual SLAM implementations could also have been used [95, 227, 228].

From a set of images $\mathcal{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_K\}$, $\mathbf{I}_k \in \mathbb{R}^{w \times h \times 3}$ $\forall k \in \{1, \dots, K\}$ of a scene, COLMAP returns a set of $J$ 6-degrees-of-freedom poses $\mathcal{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_J\}$, $\mathbf{P}_j = \begin{pmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0} & 1 \end{pmatrix} \in \mathbf{SE}(3)$ $\forall j \in \{1, \dots, J\}$, $J \leq K$, corresponding to the cameras that the

method was able to register, and the set of 3D keypoints $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_I\}$, $\mathbf{X}_i \in \mathbb{R}^3 \; \forall i \in \{1, \ldots, I\}$ that were reconstructed, all of them in a common reference frame. The camera with pose $\mathbf{P}_j$ observes a subset of $L_j$ points from the total set of 3D points $\mathcal{X}_j = \{\mathbf{X}_1, \ldots, \mathbf{X}_{L_j}\} \subset \mathcal{X}$. COLMAP final estimates are obtained by minimizing the sum of the squared reprojection errors $\sum_{j=1}^{J} \sum_{l=1}^{L_j} \mathbf{r}_{l,j}^2$.

The depth of each of the $l^{\text{th}}$ point in the $j^{\text{th}}$ camera frame is computed as

$$D_{l,j}^{\text{SfM}} = \mathbf{e}_3^\top \left( \mathbf{R}_j^\top \left( \mathbf{X}_l - \mathbf{t}_j \right) \right) \tag{8.1}$$

where $\mathbf{e}_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$ is the unit vector in the optical axis direction. We will group the depths for the sparse set of points $\mathcal{X}_j$ in the set $\mathcal{D}_j^{\text{SfM}} = \{D_{1,j}^{\text{SfM}}, \ldots, D_{L_j,j}^{\text{SfM}}\}$, $D_{l,j}^{\text{SfM}} \in \mathbb{R}_{>0} \; \forall l \in \{1, \ldots, L_j\}$, and the depths for all images in $\mathcal{D}^{\text{SfM}} = \{\mathcal{D}_1^{\text{SfM}}, \ldots, \mathcal{D}_J^{\text{SfM}}\} \; \forall j \in \{1, \ldots, J\}$.

### 8.1.2 Single-View Depth from Neural Networks

Our SfM-TTR method can be applied to any architecture, and hence its predicted depth $\mathbf{D}_j^{\text{NN}} \in \mathbb{R}^{w \times h}$ for an image $\mathbf{I}_j$ can be generally formulated as

$$\mathbf{D}_j^{\text{NN}} = h \left( g \left( \mathbf{I}_j, \boldsymbol{\theta}_g \right), \boldsymbol{\theta}_h \right) \tag{8.2}$$

where $h(\cdot)$ and $g(\cdot)$ stand respectively for the decoder and encoder parts of the deep networks, and $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_g$ their respective weights, that have been trained either supervised or self-supervisedly.

Note that the depths $\mathcal{D}_j^{\text{SfM}}$ and $\mathbf{D}_j^{\text{NN}}$ correspond to the same image $\mathbf{I}_j$ but are respectively sparse and dense, having hence a different number of elements, and they may have different scales. The scale is unobservable by COLMAP and self-supervised networks, while it is learned from the training data by supervised networks.

In order to estimate the relative scale between $\mathcal{D}_j^{\text{SfM}}$ and $\mathbf{D}_j^{\text{NN}}$ and refine at inference time the deep network, we have to select from $\mathbf{D}_j^{\text{NN}}$ those elements corresponding to the sparse depth of $\mathcal{D}_j^{\text{SfM}}$. For a general element $l$, we use the sampling operator $[\cdot]$ to access the depth corresponding to the pixel coordinates $\mathbf{p}_{l,j}$

$$D_{l,j}^{\text{NN}} = \mathbf{D}_j^{\text{NN}} \left[ \mathbf{p}_{l,j} \right] \tag{8.3}$$

where $\mathbf{p}_{l,j}$ is obtained from the coordinates of the 3D points $\mathbf{X}_l \in \mathcal{X}_j$ and the camera pose $\mathbf{P}_j$ and applying the pinhole projection function, that we will denote as $\pi(\cdot)$

$$\mathbf{p}_{l,j} = \begin{pmatrix} u & v \end{pmatrix}_{l,j}^\top = \pi \left( \mathbf{R}_j^\top \left( \mathbf{X}_l - \mathbf{t}_j \right) \right) \tag{8.4}$$

We finally group the depths predicted by the deep network for the sparse set of points $\mathcal{X}_j$ in a joint set $\mathcal{D}_j^{\text{NNs}} = \{D_{1,j}^{\text{NN}}, \ldots, D_{L_j,j}^{\text{NN}}\}$, $D_{l,j}^{\text{NN}} \in \mathbb{R}_{>0}$ , and the depths for all images in $\mathcal{D}^{\text{NNs}} = \{\mathcal{D}_1^{\text{NNs}}, \ldots, \mathcal{D}_J^{\text{NNs}}\} \; \forall j \in \{1, \ldots, J\}$.

### 8.1.3 Scale Alignment

Scale alignment is not trivial in our setup, as both $\mathcal{D}^{\text{SfM}}$ and $\mathcal{D}^{\text{NNs}}$ are affected by heteroscedastic (depth-dependent) inlier noise and contain a non-negligible rate of outliers. In addition, we are interested in removing outliers from $\mathcal{D}^{\text{SfM}}$, but we do want to keep them in $\mathcal{D}^{\text{NNs}}$, as then our SfM-TTR can reduce their errors. We developed a novel scale alignment method with two stages: we make a first fit with a strict inlier model to obtain an accurate relative scale, and then relax it in the second stage to select the points used for self-supervision from $\mathcal{D}^{\text{SfM}}$.

In the first stage we use RANSAC [229], computing 1D model instantiations, $s_{l,j} = {}^{D^{\text{NN}}_{l,j}}/_{D^{\text{SfM}}_{l,j}}$ and consider in the inlier set $\mathcal{D}^{\text{NNs}\checkmark} \subset \mathcal{D}^{\text{NNs}}$ and $\mathcal{D}^{\text{SfM}\checkmark} \subset \mathcal{D}^{\text{SfM}}$ all depths pairs $\{D^{\text{NN}}_{l',j'}, D^{\text{SfM}}_{l',j'}\}$ for which the following holds

$$\frac{\left(s_{l,j} \cdot D^{\text{SfM}}_{l',j'} - D^{\text{NN}}_{l',j'}\right)^2}{s_{l,j} \cdot D^{\text{SfM}}_{l',j'}} \leq \tau \tag{8.5}$$

where $\tau$ is the inlier threshold.

In most occasions, the distribution of depths in the image is highly unbalanced, with higher frequencies for closer depths. This, together with the heteroscedasticity of the depth errors (errors are smaller for closer depths), causes that the frequently used median scale [215] corresponds to close points, biasing the estimation. Using least squares with all the inlier set $\{\mathcal{D}^{\text{NNs}\checkmark}, \mathcal{D}^{\text{SfM}\checkmark}\}$ is not a good alternative either, the fit will be biased in this case towards large depths as they have larger errors. For these reasons, we use weighted least squares to obtain a refined estimate of $s$ with the depths $D^{\text{NN}\checkmark}_{l,j} \in \mathcal{D}^{\text{NNs}\checkmark}$ and $D^{\text{SfM}\checkmark}_{l,j} \in \mathcal{D}^{\text{SfM}\checkmark}$

$$\hat{s} = \underset{s}{\text{argmin}} \sum_j \sum_l w^s_{l,j} \left(s \cdot D^{\text{SfM}\checkmark}_{l,j} - D^{\text{NN}\checkmark}_{l,j}\right)^2 \tag{8.6}$$

where $w^s_{l,j}$ is a per-pixel weight, that should be proportional to the inverse of the expected depth variance $\sigma^2_{l,j}$. Under the reasonable assumption of similar baselines and matching noises for all reconstructed points, it is well known that the variance grows with the depth squared [222] and hence we can use as weights

$$w^s_{l,j} = {}^1/_{\sigma^2_{l,j}} \approx {}^1/_{\left(D^{\text{NN}\checkmark}_{l,j}\right)^2}. \tag{8.7}$$

Finally, we use $s_j$ from the optimization in Equation 8.6 to obtain the final set of inliers $\{\mathcal{D}^{\text{NNs}\checkmark\checkmark}, \mathcal{D}^{\text{SfM}\checkmark\checkmark}\}$ that we will use for our SfM-TTR. We proceed similarly to Equation 8.5, but this time using the absolute value in the numerator, relaxing in this manner the model and favoring the inclusion of noisy depth predictions from the network depth set $\mathcal{D}^{\text{NNs}}$ in order to have the chance to improve them at test time.

### 8.1.4 Test-Time Refinement

We refine the target network for the selected scene by updating its parameters using the depths in the final inlier set $\mathcal{D}^{\text{SfM}\checkmark\checkmark}$ as supervision. As in [215], we optimize over the complete scene, thus obtaining a refined network with more consistent predictions across all views. This is different from other TTR works, such as [212], in which they refine a different network for each frame of the sequence.

Each batch update works as follows. We sample an image $\mathbf{I}_j$ from the sequence and do a feed-forward pass through the network to obtain the depth prediction $\mathbf{D}_j^{\text{NN}}$. Then we supervise the prediction with the sparse pseudo ground truth $\mathcal{D}_j^{\text{SfM}\checkmark\checkmark}$. This supervision is weighted according to the reliability of the reconstructed 3D points, that we approximate based on their reprojection errors as $w_{l,j}^{\boldsymbol{\theta}} = \exp(-\|\mathbf{r}_{l,j}\|_2^2)$.

$$\mathcal{L} = \frac{1}{|\mathcal{D}_j^{\text{SfM}\checkmark\checkmark}|} \sum_l w_{l,j}^{\boldsymbol{\theta}} \|\hat{s} \cdot D_{l,j}^{\text{SfM}\checkmark\checkmark} - D_{l,j}^{\text{NN}\checkmark\checkmark}\|_1 \tag{8.8}$$

As state-of-the-art depth networks already produce sharp predictions with well-defined object contours, we argue that our refinement should only optimize the internal understanding of the scene. Hence, we follow a similar approach as [212] and only update the encoder parameters during the TTR, keeping the rest of the network fixed. Our TTR optimization can be hence formulated as $\hat{\boldsymbol{\theta}}_g = \text{argmin}_{\boldsymbol{\theta}_g} \mathcal{L}$. In this manner, the frozen decoder $h(\cdot)$ keeps producing sharp predictions, but now they stem from a more informed representation of the underlying scene.

## 8.2 Experiments

### 8.2.1 Implementation Details and Baselines

We validate our proposed SfM-TTR by applying it to different state-of-the-art baselines. Specifically, we provide evaluations with the baselines CADepth [171], DIFFNet [172], and ManyDepth [155] as representative of self-supervised approaches. We also implemented it on AdaBins [11] to benchmark SfM-TTR's performance also with a representative supervised model. The same set of hyperparameters was used for SfM-TTR with all baselines, achieving a substantial improvement in all of them without requiring individual tuning.

For the sparse reconstruction, we run COLMAP [175] with its default parameters, using a single pinhole camera model per sequence and sequential matching. Although we use all available images from a sequence to create the sparse reconstruction, the network is only optimized with the target frames of the evaluation. Regarding our scale alignment, we detect outliers running RANSAC for 20 iterations with inlier threshold $\tau = 0.5$. For the TTR optimization, we use Adam [230] applied to the encoder parameters, $\boldsymbol{\theta}_g$, with a learning rate of $10^{-4}$ for 200 steps.

For comparison, we also implemented the instance-wise photometric refinement (Ph-TTR) from ManyDepth [155][1], based on the work of McCraith et al. [212], which updates the weights of the network encoder during inference using the

1: The TTR code was not available in the authors' repository at the time of writing this document.

photometric loss from the training. Table 8.1 validates our implementation, showing similar performance as the one reported by the authors in [155].

| Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ |
|---|---|---|---|---|
| ManyDepth [155] Ph-TTR ⋄ | 0.087 | 0.696 | 4.183 | 0.167 |
| ManyDepth [155] Ph-TTR ∗ | 0.088 | 0.681 | 4.122 | 0.168 |

**Table 8.1:** ManyDepth Ph-TTR [155] (⋄) and our own implementation (∗) obtain similar metrics.

### 8.2.2 Dataset

We run all evaluations on the KITTI dataset [119], the common benchmark for single- and multi-view depth learning. Regarding the KITTI ground truth for depth learning evaluations, the literature is split among those following Eigen et al. [10], with reprojected LiDAR point clouds, and those using the newer and improved ground truth [231], which aggregates 5 consecutive frames and handles dynamic objects. Given the higher reliability of the new ground truth, we used it to evaluate all the baselines on the Eigen test split with all the images that contain ground truth, a total of 652. We provide evaluation without and with the Eigen cropping, see Table 8.4 and Table 8.5. For fairness and completeness, as some methods present results with the old ground truth, we also include an evaluation with the LiDAR reprojected depths, on the complete Eigen split with 697 images. We report additional results directly taken from the corresponding papers, see Table 8.6.

In a few of the KITTI test scenes the camera motion is insufficient for proper SfM convergence. Our SfM-TTR cannot refine the depth in those cases, but for a fair comparison, we included these sequences in the global metrics using the results of the network without SfM-TTR.

Note that although we have presented a novel scale alignment, for the sake of fairness we align the self-supervised predictions and the ground truth with the per-image median, as commonly done [155, 170]. Also following the common evaluation practices, we set a maximum depth of 80 meters.

### 8.2.3 Comparisons against Baselines

We demonstrate the benefits of our method by comparing the results of applying a photometric refinement (Ph-TTR) and ours (SfM-TTR) on the baseline networks. Table 8.4 shows how our SfM-TTR consistently and significantly improves the predictions of all networks, obtaining superior performance than the photometric refinement. Besides, Ph-TTR fails to improve over CADepth without TTR. The most likely reason is that it requires individual hyperparameter tuning, which was not required for our SfM-TTR.

The advantages of our proposed method are especially noticeable for large depths, where Ph-TTR cannot provide a good supervision signal due to the limited parallax between close frames. Our refinement, instead, leverages SfM, which triangulates points from the complete sequence. This produces better estimates for distant points and better supervision, resulting in a drastic reduction of the RMSE by up to 30%. This effect is clearly visible in Figure 8.2. Although smaller depths show comparable performance for Ph-TTR and SfM-TTR, the photometric loss does not help in areas with large depths. SfM-TTR, instead, provides a significant gain in performance in those areas.

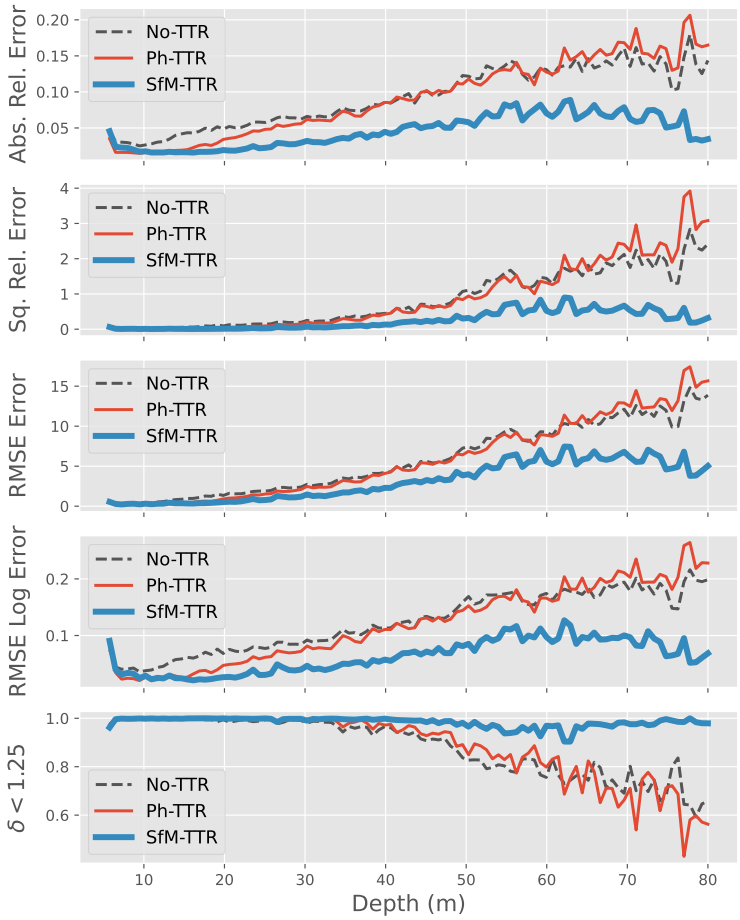



**Figure 8.2: Error metrics for different depths with DIFFNet.** Our SfM-TTR (thick blue) gives a substantial improvement over No-TTR (dashed black) and Ph-TTR (thin red) at medium and large depths. Ph-TTR offers some improvement over No-TTR at close depths, where the small baselines of photometric losses are informative, but it does not improve or it is slightly worse at medium and large depths. The metrics $\delta < 1.25^2$ and $\delta < 1.25^3$ are not plotted, as differences are small (see for example Table 8.4).

The best results are obtained when applying our SfM-TTR to DIFFNet, even though the original DIFFNet without TTR performs slightly worse than ManyDepth. We believe that our TTR has a smaller effect on ManyDepth because it already leverages scene information by using multiple frames at inference time. SfM-TTR can also improve results on AdaBins, for which Ph-TTR cannot be implemented, as AdaBins does not provide a pose estimation module. This further demonstrates the effectiveness of directly optimizing for the 3D points from COLMAP.

Qualitatively, Figure 8.5 shows how predictions after SfM-TTR keep looking sharp with well-defined boundaries despite the sparsity of the pseudo-ground truth. We argue that optimizing the encoder enables a better understanding of the scene while freezing the decoder maintains the previously learned sharpness of the predictions. The error maps from Figure 8.4 reveal the differences between refinements, showing how our method can effectively reduce errors in regions where Ph-TTR cannot. The positive effect of SfM-TTR in distant points is visible in Figure 8.3, where large depths move closer to the ground truth after our refinement.

Regarding runtime efficiency, our method requires roughly 2 seconds per frame during the optimization, similar to Ph-TTR, and faster than other multi-view TTR that also use large baselines [214, 215].

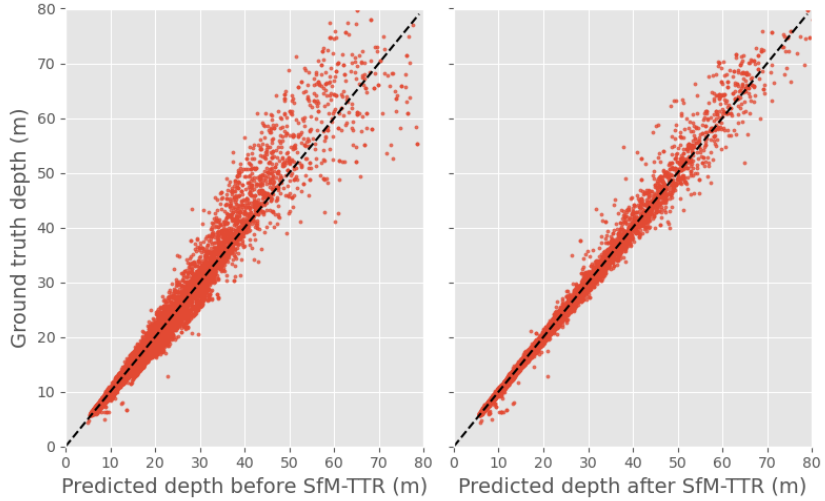



**Figure 8.3: Depth predictions and ground truth before and after SfM-TTR with DIFFNet.** The red dots stand for predicted pixel depths on a KITTI sequence with DIFFNet, the black dashed line stands for zero error. Note how after SfM-TTR the red dots gather closer to the dashed black line, illustrating that the predicted depths are closer to the ground truth ones.

| Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ |
|---|---|---|---|---|
| AdaBins [11] | 0.072 | 0.325 | 3.134 | 0.112 |
| AdaBins [11] + SfM-TTR (full model) | 0.062 | **0.204** | 2.297 | 0.092 |
| **AdaBins [11] + SfM-TTR (encoder)** | **0.060** | **0.204** | **2.260** | **0.091** |
| ManyDepth [155] | 0.064 | 0.345 | 3.116 | 0.103 |
| ManyDepth [155] + SfM-TTR (full model) | 0.059 | **0.293** | 2.655 | 0.096 |
| **ManyDepth [155] + SfM-TTR (encoder)** | **0.057** | 0.294 | **2.648** | **0.094** |
| CADepth [171] | 0.078 | 0.403 | 3.432 | 0.119 |
| CADepth [171] + SfM-TTR (full model) | 0.069 | **0.321** | 2.824 | **0.104** |
| **CADepth [171] + SfM-TTR (encoder)** | **0.068** | 0.328 | **2.821** | 0.106 |
| DIFFNet [172] | 0.071 | 0.361 | 3.230 | 0.110 |
| DIFFNet [172] + SfM-TTR (full model) | 0.057 | **0.273** | 2.621 | **0.092** |
| **DIFFNet [172] + SfM-TTR (encoder)** | **0.056** | **0.273** | **2.600** | 0.093 |

**Table 8.2: Encoder vs. full network TTR.** Note how the best results are achieved with encoder-only TTR.

### 8.2.4 Ablation Studies

To validate the relative importance of the individual components of our SfM-TTR, we perform ablation studies where we dispose some of our key components.

Table 8.2 shows a comparison between refining the complete network and only updating the encoder. Similar to [212], we obtain better results when only updating the encoder, further showing how light refinement schemes should only focus on improving the underlying representation of the network.

As shown in Table 8.3, using the mean of per-image medians [168, 215] alignment in our SfM-TTR, as well as other ablated versions of our method, worsens significantly the performance on AdaBins. The alignment is specially important for supervised models, as their scale is not corrected during the evaluation. With our alignment, we are accounting for outliers with RANSAC and for the heteroscedastic nature of the depth noise with weighted least squares, resulting in substantially more robust and accurate results.

| Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ |
|---|---|---|---|---|
| AdaBins [11] | 0.072 | 0.325 | 3.134 | 0.112 |
| AdaBins [11] + SfM-TTR (median) | 0.074 | 0.263 | 2.509 | 0.103 |
| AdaBins [11] + SfM-TTR ($\mathcal{D}^{\text{NNs}\checkmark}$, $\mathcal{D}^{\text{SfM}\checkmark}$) | 0.065 | 0.278 | 2.787 | 0.103 |
| AdaBins [11] + SfM-TTR (Least Squares) | 0.064 | 0.222 | 2.346 | 0.097 |
| AdaBins [11] + SfM-TTR ($w^{\theta}_{l,j} = 1$) | 0.062 | 0.206 | 2.310 | **0.091** |
| **AdaBins [11] + SfM-TTR** | **0.060** | **0.204** | **2.260** | **0.091** |

**Table 8.3: Alignment ablation study.** Note the substantial improvement of our scaling approach (detailed in Subsection 8.1.3) over other alignments.

**Table 8.4: Quantitative results with new KITTI ground truth, Eigen split and no cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from original papers with ⋄. We compare different architectures without TTR, with Ph-TTR and with our SfM-TTR. † Results from AdaBins differ from [11], as in this table we do not crop during evaluation. For results using cropping, see Table 8.5.

| TTR | Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| ✗ | AdaBins [11] ∗† | 0.072 | 0.325 | 3.134 | 0.112 | 0.941 | 0.990 | **0.998** |
| ✓ | **AdaBins [11] + SfM-TTR** | **0.060** | **0.204** | **2.260** | **0.091** | **0.970** | **0.993** | **0.998** |
| ✗ | ManyDepth [155] ⋄ | 0.064 | 0.345 | 3.116 | 0.103 | 0.949 | 0.989 | **0.997** |
| ✓ | ManyDepth [155] + Ph-TTR ⋄ | **0.056** | 0.322 | 3.034 | 0.096 | 0.961 | **0.992** | **0.997** |
| ✓ | **ManyDepth [155] + SfM-TTR** | 0.057 | **0.294** | **2.648** | **0.094** | **0.963** | 0.990 | **0.997** |
| ✗ | CADepth [171] ∗ | 0.078 | 0.403 | 3.432 | 0.119 | 0.933 | 0.988 | **0.997** |
| ✓ | CADepth [171] + Ph-TTR ∗ | 0.088 | 0.475 | 3.723 | 0.132 | 0.914 | 0.984 | 0.996 |
| ✓ | **CADepth [171] + SfM-TTR** | **0.068** | **0.328** | **2.821** | **0.106** | **0.955** | **0.990** | 0.996 |
| ✗ | DIFFNet [172] ∗ | 0.071 | 0.361 | 3.230 | 0.110 | 0.946 | 0.990 | 0.997 |
| ✓ | DIFFNet [172] + Ph-TTR ∗ | 0.057 | 0.285 | 2.900 | 0.095 | 0.961 | **0.992** | **0.998** |
| ✓ | **DIFFNet [172] + SfM-TTR** | **0.056** | **0.273** | **2.600** | **0.093** | **0.969** | **0.992** | 0.997 |

**Table 8.5: Quantitative results with new KITTI ground truth, Eigen split and Eigen cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from papers with ⋄. † Results from AdaBins + SfM-TTR follow the common KITTI Benchmark cropping from the supervised depth learning literature [11], and the AdaBins results without TTR are taken from the original paper.

| TTR | Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| ✗ | AdaBins [11] ⋄ † | 0.058 | 0.190 | 2.360 | 0.088 | 0.964 | 0.995 | **0.999** |
| ✓ | **AdaBins [11] + SfM-TTR †** | **0.054** | **0.138** | **1.885** | **0.078** | **0.978** | **0.996** | **0.999** |
| ✗ | ManyDepth [155] ∗ | 0.059 | 0.297 | 2.960 | 0.097 | 0.954 | 0.991 | **0.998** |
| ✓ | ManyDepth [155] + Ph-TTR ∗ | **0.053** | **0.252** | 2.774 | **0.089** | 0.962 | **0.993** | **0.998** |
| ✓ | **ManyDepth [155] + SfM-TTR** | 0.054 | **0.252** | **2.510** | **0.089** | **0.966** | 0.992 | **0.998** |
| ✗ | CADepth [171] ∗ | 0.073 | 0.359 | 3.287 | 0.112 | 0.941 | 0.990 | **0.997** |
| ✓ | CADepth [171] + Ph-TTR ∗ | 0.082 | 0.426 | 3.565 | 0.124 | 0.923 | 0.986 | **0.997** |
| ✓ | **CADepth [171] + SfM-TTR** | **0.060** | **0.263** | **2.620** | **0.096** | **0.962** | **0.992** | **0.997** |
| ✗ | DIFFNet [172] ∗ | 0.066 | 0.318 | 3.078 | 0.103 | 0.953 | 0.992 | **0.998** |
| ✓ | DIFFNet [172] + Ph-TTR ∗ | 0.053 | 0.252 | 2.778 | 0.090 | 0.965 | 0.993 | **0.998** |
| ✓ | **DIFFNet [172] + SfM-TTR** | **0.052** | **0.229** | **2.444** | **0.085** | **0.973** | **0.994** | **0.998** |

**Table 8.6: Quantitative results with Eigen (old) KITTI ground truth, Eigen split and Eigen cropping.** Best results per model in **bold**, best results across all self-supervised models underlined. Experimental results are marked with ∗, results from original papers with ⋄. Note how, with this different ground truth, we again outperform the results of the baselines in Tables 8.4 and 8.5 and we further demonstrate improvement over Monodepth2 [170] and the TTR approaches [214, 215] that were evaluated after such architecture in the original papers.

| TTR | Method | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSE log ↓ | $\delta < 1.25$ ↑ | $\delta < 1.25^2$ ↑ | $\delta < 1.25^3$ ↑ |
|---|---|---|---|---|---|---|---|---|
| ✗ | AdaBins [11] ∗ | **0.087** | 0.480 | 3.637 | 0.168 | 0.917 | 0.970 | **0.985** |
| ✓ | **AdaBins [11] + SfM-TTR** | 0.088 | **0.454** | **3.355** | **0.164** | **0.927** | **0.971** | **0.985** |
| ✗ | Monodepth2 (384x112) [170] ⋄ | **0.128** | **1.040** | 5.216 | 0.207 | 0.849 | **0.951** | **0.978** |
| ✓ | Monodepth2 + TTR (from [215]) ⋄ | 0.130 | 2.086 | **4.876** | **0.205** | **0.878** | 0.946 | 0.970 |
| ✗ | Monodepth2 [170] ⋄ | 0.115 | 0.903 | 4.863 | 0.193 | 0.877 | 0.9590 | 0.981 |
| ✓ | Monodepth2 + TTR (from [214]) ⋄ | 0.113 | **0.793** | 4.655 | 0.188 | 0.874 | 0.960 | **0.983** |
| ✓ | **Monodepth2 + SfM TTR** | **0.098** | 0.858 | **4.418** | **0.177** | **0.908** | **0.964** | 0.981 |
| ✗ | ManyDepth [155] ⋄ | 0.093 | 0.715 | 4.245 | 0.172 | 0.909 | 0.966 | **0.983** |
| ✓ | ManyDepth [155] + Ph-TTR ⋄ | **0.087** | **0.696** | 4.183 | **0.167** | **0.918** | **0.968** | **0.983** |
| ✓ | **ManyDepth [155] + SfM-TTR** | 0.090 | 0.718 | **4.040** | 0.168 | 0.917 | 0.967 | **0.983** |
| ✗ | CADepth [171] ⋄ | 0.102 | 0.734 | 4.407 | 0.178 | 0.898 | **0.966** | **0.984** |
| ✓ | CADepth [171] + Ph-TTR ∗ | 0.110 | 0.802 | 4.648 | 0.187 | 0.878 | 0.962 | 0.983 |
| ✓ | **CADepth [171] + SfM-TTR** | **0.095** | **0.703** | **4.073** | **0.173** | **0.912** | **0.966** | 0.982 |
| ✗ | DIFFNet [172] ⋄ | 0.097 | 0.722 | 4.345 | 0.174 | 0.907 | 0.967 | **0.984** |
| ✓ | DIFFNet [172] + Ph-TTR ∗ | **0.087** | 0.667 | 4.138 | 0.167 | 0.920 | 0.968 | **0.984** |
| ✓ | **DIFFNet [172] + SfM-TTR** | **0.087** | **0.660** | **3.948** | **0.165** | **0.925** | **0.969** | **0.984** |

## 8.3 Limitations

As our current implementation of SfM-TTR depends on COLMAP's output, it is inherently offline and its performance is bounded to the quality of the SfM results. Although we achieve good results in KITTI, a natural scenario and standard benchmark, more challenging setups for SfM (for example, dynamic objects, drastic appearance changes or low-parallax motion) are also problematic for SfM-TTR. Works addressing such SfM challenges [232] will also be beneficial for our method. Although we could easily replace COLMAP's reconstruction by that of an online real-time visual SLAM pipeline, e.g. [95], online and real-time refinement of deep models is not straightforward. We find these aspects relevant for our future work.

Although SfM-TTR excels at medium and large depths, we have noticed a comparable or slightly worse performance than Ph-TTR at very close depths, for which even the adjacent views used in Ph-TTR have sufficient parallax. Observe the metrics in Figure 8.2 for depths under 10 meters. This observation suggests a future line of research to combine the best from both Ph-TTR and SfM-TTR.

## 8.4 Conclusion

In this chapter we have presented SfM-TTR, an effective test-time refinement for single-view depth networks that preserves the learned priors of supervised and self-supervised models while also leveraging wide-baseline multi-view constraints at inference. The key ingredient is formulating a TTR loss based on sparse SfM depths, which have been estimated from wider baselines than traditional photometric losses, that only consider adjacent frames. We propose a novel RANSAC-based method for scale alignment between SfM and the depth network that accounts for the depth outliers and its heteroscedastic noise. Very importantly, we use a fixed set of hyperparameters for our SfM-TTR for all experiments, without requiring per-architecture or per-sequence tuning.

Our experiments show that our SfM-TTR improves significantly the depth predictions of different state-of-the-art networks, supervised and self-supervised. We also outperform by a wide margin, in particular at medium and large depths, the common TTR approach that we denote as Ph-TTR, based on the use of photometric losses. These results validate our method as a general TTR approach easy to implement and use after all kinds of networks, current and future ones. Besides, as a more general comment, we believe that the presented contributions provide insights towards a further leverage of SfM in self-supervised depth learning, arising as a promising extension to the widely used photometry-based losses.

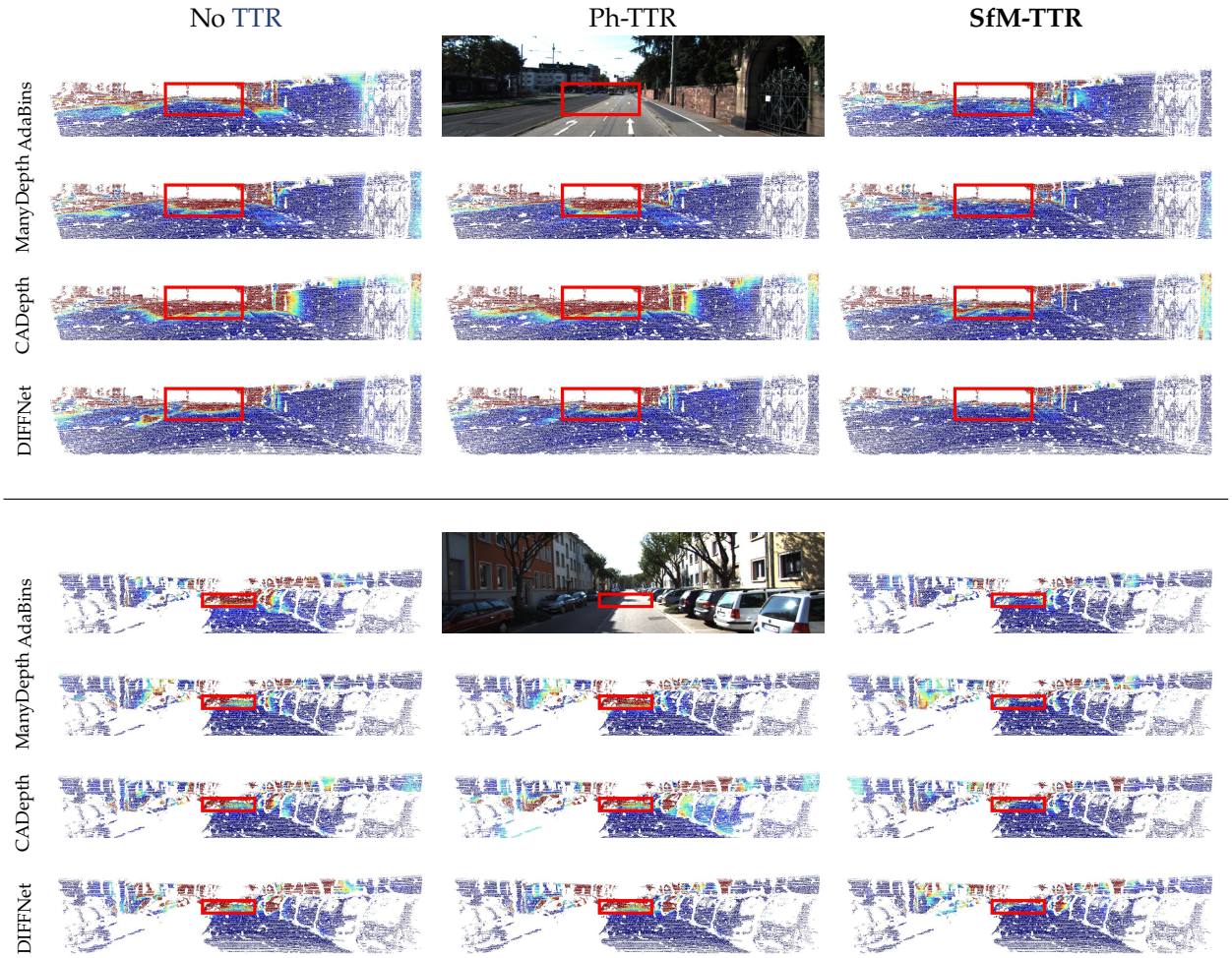



**Figure 8.4: RMSE maps for different baselines architectures (rows) and TTR (columns).** The input image is the center top image, as AdaBins cannot be refined with photometric loss. The benefit of our SfM-TTR is particularly noticeable for large depths (framed by red rectangles). Ph-TTR methods struggle in these areas as they use weak low-parallax constraints, while SfM leverages wider baselines and produces more accurate depth supervision. Figure best viewed in color.

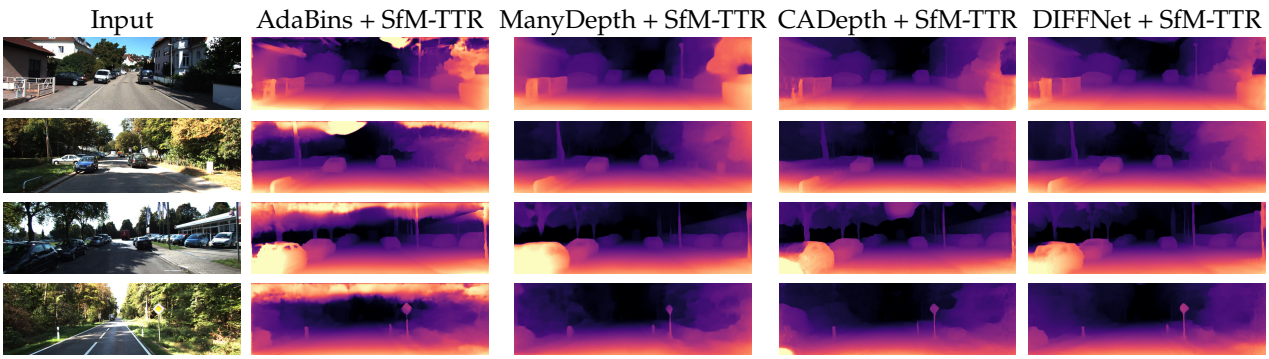



**Figure 8.5:** Qualitative depth maps for different architectures after SfM-TTR on KITTI.

# 9 Zero-Shot Multi-View Stereo



*Computing accurate depth from multiple views is a fundamental and longstanding challenge in computer vision. However, most existing approaches do not generalize well across different domains and scene types (e.g. indoor vs outdoor). Training a general-purpose multi-view stereo model is challenging and raises several questions, e.g. how to best make use of transformer-based architectures, how to incorporate additional metadata when there is a variable number of input views, and how to estimate the range of valid depths which can vary considerably across different scenes and is typically not known a priori? To address these issues, we introduce MVSA, a novel and versatile Multi-View Stereo architecture that aims to work Anywhere by generalizing across diverse domains and depth ranges. MVSA combines monocular and multi-view cues with an adaptive cost volume to deal with scale-related issues. We demonstrate state-of-the-art zero-shot depth estimation on the Robust Multi-View Depth Benchmark, surpassing existing multi-view stereo and monocular baselines.*

Chapter based on [18].
Sergio Izquierdo et al.
'MVSAnywhere: Zero Shot Multi-View Stereo'
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025

Code and models are available at `https://github.com/nianticlabs/mvsanywhere`

Estimating accurate depth from multiple RGB images is a core challenge in 3D vision, and a building block for downstream applications like 3D reconstruction and autonomous driving. Recent approaches in learning-based MVS are capable of generating accurate depths [123, 125, 233]. However, existing methods typically struggle to generalize to scene and camera setups that differ significantly from those in their training data. As a result, there is a pressing need for general-purpose MVS methods that are more robust to differences between the training and test distributions.

We take inspiration from the recent explosion in scene-agnostic *single-view* depth models, which predict plausible metric [12, 151, 152, 154, 234, 235] or up-to-scale [146–149] depth using only a single image as input. These models are typically trained on large curated sets of synthetic and/or real RGB-D data, endowing them with impressive generalization performance on previously unseen data. Single-view models are, however, inherently limited by their input. For our specific depth prediction target, constraining the model's input to just one image forces it to use single-view geometry cues (*e.g.* vanishing points) and learned patterns [236], while losing the stronger multi-view signal. While there are temporal extensions of these single view models [155, 237–240], their focus is on temporal perceptual consistency, and not necessarily multi-view consistency. In application contexts where multiple views are available at inference time, it stands to reason that these lead to significantly more accurate depth estimates [15, 215, 241].

Developing a general-purpose MVS method, however, raises two significant challenges. Firstly, it should be able to deal with arbitrary depth ranges. Existing MVS methods typically require a known range of depths to 'search' over along epipolar lines, corresponding to a discrete set of depth bins used to build a cost volume. These depths are typically either fixed (and chosen from the range of depths in the training data) [190] or are provided at test time for each image [123,

233, 242]. Secondly, the many emerging benefits of ViTs [98] motivate us to find a way to 'upgrade' parts of standard MVS architectures that are still CNNs.

To address these challenges, we introduce a new general-purpose MVS method named Multi-View Stereo Anywhere (**MVSA**). Similarly to recent performant monocular methods, it is trained on a large and diverse set of data, spanning diverse depth ranges. Along with harmonizing these training signals, our main technical contributions are:

- A novel transformer-based architecture that processes the multi-view cost volume, while *also* incorporating monocular features. We propose a Cost Volume Patchifier that tokenizes the cost volume without loosing its details, while also incorporating features from a monocular ViT.
- We propose a view-count-agnostic and scale-agnostic mechanism to construct the cost volume using geometric metadata given any number of input source frames. This is in contrast to the established practice [190] of concatenating geometric metadata from a fixed number of frames to build the cost volume.

MVSA predicts highly accurate and 3D-consistent *depths*, obtaining state-of-the-art results on the Robust Multi-View Depth Benchmark [243], which contains a variety of challenging held-out datasets. We also report scores for some new single- and multi-view methods for comparison. Our better depths result in improved 3D mesh *reconstruction* compared to alternative depth-based reconstruction methods (Figure 9.1).

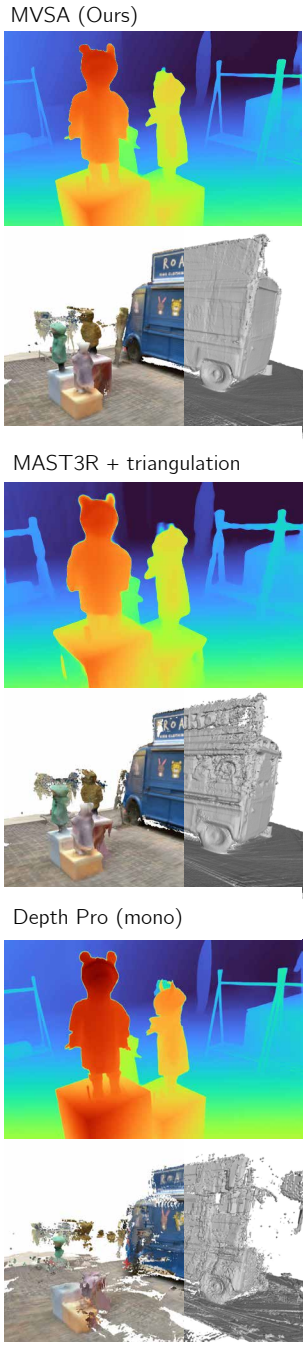



**Figure 9.1:** Our **MVSA** model results in high-quality reconstructions from posed images, and is superior to existing monocular and MVS methods. Here we compare with Depth Pro [153], a recent monocular method which produces sharp and good looking depth maps, but can have inconsistent scaling of depths, which are required for good meshes. We also include a variant of MAST3R [193] that we have augmented with ground truth camera poses. Our model gives sharp depth maps which are also accurate and 3D consistent, producing high-quality meshes in zero-shot environments.

## 9.1 Method: General-purpose Multi-View Stereo

Our model takes as input a $H \times W$ reference image $\mathbf{I}_r$ together with neighboring source frames $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$, each with their relative poses and intrinsics. At test time we aim to predict a dense depth map $\mathbf{D}_r$ for $\mathbf{I}_r$. For ours to be a general-purpose MVS method, we seek to:

1. **Generalize to any domain**. Most current MVS methods are typically trained on and tested on data from similar domains, *e.g.* indoor only or driving only.
2. **Generalize to any range of depths**. Predicted depth maps need to be accurate for nearby surfaces (*e.g.* for robotics) or for more distant ones (*e.g.* for drones and autonomous driving). In some scenarios like SfM, the depths and camera poses are in a non-metric up-to-scale coordinate system. Hence, general-purpose MVS should be robust to the scale of the coordinate system.
3. **Be robust to the number and selection of source frames**. Traditional MVS systems can struggle when there is little overlap between source and reference frames. We also want MVS methods to be agnostic to the number of source frames available at test time.
4. **Predict 3D-consistent depths.** Depths from one viewpoint should be consistent with those predicted from different viewpoints. Fusion of consistent depth maps will produce a mesh with accurate estimates of 3D surfaces.

While prior works have tackled these problems in turn, we are the first model, to the best of our knowledge, to tackle all four problems in a single system.

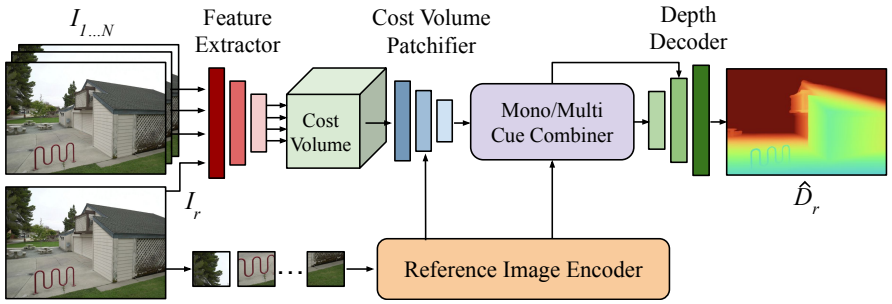



**Figure 9.2: Our general-purpose multi-view depth estimation model**. We start with a cost-volume based architecture, which matches deep features between views at different hypothesized depths. Key for performance are our Cost Volume Patchifier and Mono/Multi Cue Combiner. These also fuse single-view information coming from the Reference Image Encoder and source views.

### 9.1.1 MVSAnywhere

We introduce **MVSAnywhere** (MVSA), a novel general-purpose MVS system which is designed to embody each of the previous properties. To help us learn from diverse datasets and hence **generalize to any domain**, we use a large transformer-based architecture, which takes as input: (1) multi-view information from the reference and source images, and (2) single-view information, which is extracted directly from the reference image via a monocular *reference image encoder*. The overall architecture (Figure 9.2) is broadly inspired by recent MVS approaches, *e.g.* [190]. It comprises five key components:

**Feature extractor.** This encodes the source and reference images into deep feature maps $\mathcal{F}_r$ and $\mathcal{F}_{i\in\{1...N\}}$, that will be processed via a cost volume. We use the first two blocks of a ResNet18 [244] for this encoder, producing feature maps at resolution $H/4 \times W/4$.

**Cost volume.** Following *e.g.* [124, 178, 180, 185], we warp feature maps $\mathcal{F}_i$ from each source view to the reference one using a set of hypothesized depth values (*i.e.* bins) $\mathcal{B}$. We then concatenate these warped features and $\mathcal{F}_r$ with appropriate *metadata*, following [190]. See Subsection 9.1.2 for our specific novel contributions in this matter.

**Reference image encoder.** This extracts powerful deep monocular features for $\mathbf{I}_r$. We use the ViT Base [98] encoder from Depth Anything V2 [148], with their pretrained weights for relative monocular depth estimation, which help us to be **robust to limited overlaps** between source and reference frames. As ViT Base operates on $14 \times 14$ patches, the reference image is resized to $\frac{14H}{16} \times \frac{14W}{16}$ resolution before feeding to ViT Base, such that the extracted features are size $\frac{H}{16} \times \frac{W}{16}$.

**Mono/Multi Cue Combiner.** This converts the "patchified" features of the cost volume and reference image into a sequence of features which go to our depth decoder. Monocular and multi-view cues are combined by a novel component described in Subsection 9.1.3.

**Depth Decoder.** Based on the decoder from [245], MVSA progressively upsamples and processes features from the Mono/Multi Cue Combiner module to produce the final depth map at the reference image resolution.

### 9.1.2 Metadata Agnostic to View Count and Scale

SimpleRecon [190] demonstrated that readily available metadata, *e.g.* geometric and camera pose information, can be incorporated into the cost volume to improve depths. For each pixel location $(u_r, v_r)$ in $\mathbf{I}_r$ and depth bin $k$ in $\mathcal{B}$, we backproject the pixel to a 3D point $P$ and then reproject it into every source view $\mathbf{I}_i$. We enable this formulation to work with arbitrary scales by normalizing

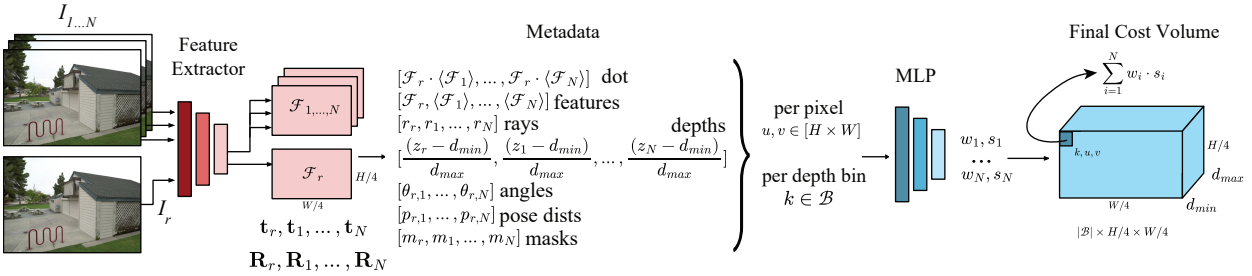



**Figure 9.3: Our metadata cost volume** is agnostic to view count and scale. To be agnostic to view count our MLP produces a weight ($w_i$) and a score ($s_i$) per position that are aggregated into a single value. To be scale agnostic we normalize the metadata that is unit dependent, *i.e.* the depths and the pose distances.

the metadata that depends on the scale. The specific metadata for the bin with coordinates $(u_r, v_r, k)$ in the cost volume includes:

**Feature dot product:** The dot product of the reference image features and each warped source image features, expressed for each source image $i$ as $\mathcal{F}_r \cdot \langle \mathcal{F} \rangle_i$, where $\langle \rangle$ is the warping operation which warps features from the source image to the reference viewpoint at the depth corresponding to bin $k$. This value is often used as the sole matching affinity in cost volumes.

**Visual features:** We also include the features from reference $\mathcal{F}_r$ and for each warped source image $i$, $\langle \mathcal{F} \rangle_i$. This supplements the dot product by also incorporating the visual features that might help to discern the reliability of the matching at that point.

**Ray directions $\mathbf{r}_r^{k,u_r,v_r}$ and $\mathbf{r}_i^{k,u_r,v_r} \in \mathbb{R}^3$:** This is the normalized directions pointing from the camera origins to the 3D location of a point $(k, u_r, v_r)$ in the plane sweep cost volume. We create rays for the reference and all the source images.

**Reference plane depth $z_r^{k,u_r,v_r}$:** This is the depth of the point at position $(k, u_r, v_r)$ in the cost volume, measured perpendicularly from the reference camera. We normalize these values with the minimum and maximum depth of the scene $((z_r - d_{min})/d_{max})$.

**Reprojected depths $z_i^{k,u_r,v_r}$:** This is the perpendicular depth of the 3D point at position $(k, u_r, v_r)$ in the cost volume, relative to the source camera $n$. As with $z_r$, we normalize these values with the minimum and maximum depth of the scene $((z_i - d_{min})/d_{max})$.

**Relative ray angles $\theta_{r,i}$:** This is the angle between the ray directions $\mathbf{r}_r^{k,u_r,v_r}$ and $\mathbf{r}_i^{k,u_i,v_i}$.

**Relative pose distance $p_{r,i}$:** This is the relative pose distance between the reference camera and a source frame, as defined in [189]:

$$p_{r,i} = \sqrt{\|\mathbf{t}_{r,i}\| + \frac{2}{3}\text{tr}(\mathbb{I} - \mathbf{R}_{r,i})}, \tag{9.1}$$

where $\mathbf{t}_{r,i}$ and $\mathbf{R}_{r,i}$ are the relative translation and rotation between views $i$ and $r$. The translation, $\mathbf{t}_{r,i}$, is normalized by the source frame with the biggest pose distance.

**Depth validity masks $m_i^{k,u_r,v_r}$:** This is a binary mask indicating whether the point $(k, u_r, v_r)$ in the cost volume projects in front of the source camera $i$ or not.

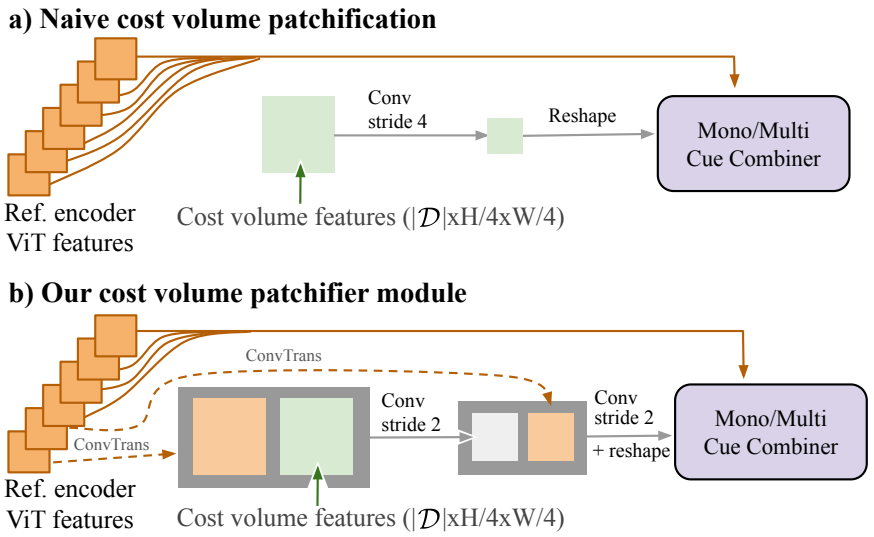



**Figure 9.4: Our cost volume patchifier** enables high-quality information to be extracted from a $|\mathcal{D}| \times \frac{H}{4} \times \frac{W}{4}$ cost volume, ready for input to the Mono/Multi Cue Combiner ViT. (a) Shows the naive approach to patchification. (b) Our approach makes better use of the reference image features.

SimpleRecon [190]'s cost volume concatenates metadata from all eight source frames and runs an MLP to produce one single cost (matching score) per spatial location and depth hypothesis. While this gives good scores, its limitation is that it requires *exactly* eight source frames for every training and test reference image, limiting the model's flexibility (note though that traditional MVS methods are typically already view-count agnostic). To address this limitation, we introduce a *view-count-agnostic metadata* component which enables a single model to **generalize to any number of source views**. For each source frame, we run an MLP that ingests the metadata from the reference frame and the source frame and predicts two values: a score and a weight. This results in $N$ scores and $N$ weights. A weighted sum of the $N$ scores is computed after the $N$ weights go through softmax. This weighted sum is used as the value in the cost volume at every pixel location $(u, v)$ and depth hypothesis $k$. Our novel module enables aggregation of the matching score and confidence for each source frame, while allowing for a variable number of source frames for each $\mathbf{I}_r$.

The source camera poses may be close to the reference, or far from it. To be more invariant to this possible range of scales, we also make the metadata **scene scale-agnostic**. To this end, we normalize the relative pose measures of the metadata using a maximum across all the source frames for a given reference frame. We also normalize the depth hypothesis metadata using the maximum and minimum of $\mathcal{B}$.

As the scene scale information is not provided to the rest of our network, we rescale the depth predictions to match the scale of the input poses. Our depths are predicted with a sigmoid function $\sigma$ over the logit $x$. To align the prediction of the network with the cost volume, the sigmoid output is scaled by the depth range of the cost volume, so

$$\mathbf{D}_r = \exp\left(\log(d_{\min}) + \log(d_{\max}/d_{\min}) \cdot \sigma(x)\right). \tag{9.2}$$

### 9.1.3 Mono/Multi Cue Combiner

Given the cost volume of shape $|\mathcal{B}| \times \frac{H}{4} \times \frac{W}{4}$, and the reference image encoder features of shape $C \times \frac{H}{16} \times \frac{W}{16}$ (outputs of different blocks of the reference image encoder), we pose the question: how can we best combine these features to

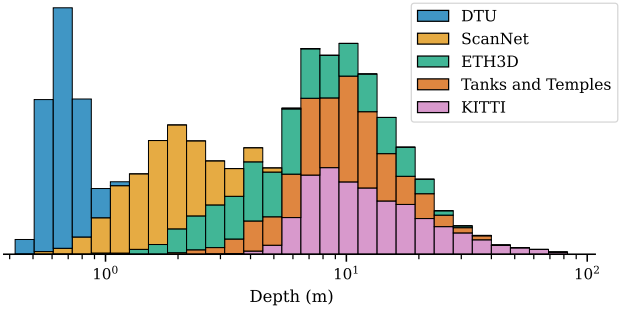



**Figure 9.5: MVS datasets cover a wide range of depth values.** Here we show the distribution of % depths in the DTU [246], ScanNet [247], ETH3D [248], Tanks and Temples [249], and KITTI [119] datasets, as a stacked bar chart. Note the log x-axis. This wide range of depth values can be challenging when it comes to constructing meaningful cost volumes and predicting the final depths.

provide the strongest signal for the decoder? Motivated by the recent success of transformer architectures in single-view depth prediction [148, 153, 245], we use a ViT-Base network to process these features in a *Mono/Multi Cue Combiner* network, which produces a sequence of tokens for the decoder to transform into a depth prediction.

To effectively achieve this we need to i) convert the cost volume into a token sequence without sacrificing information and ii) incorporate the monocular cues to help in decoding sharp depth. For i), a naive approach would be to apply a strided convolution projecting to the ViT token dimensions, resembling how RGB images are patchified. However this is suboptimal, for it lacks contextual information on how to achieve this downsampling. Instead, we propose a **cost volume patchifier** module. This guides the downsampling process with information from the first two blocks of the reference image encoder. We convert the cost volume into tokens using two strided convolutions, but first, concatenate each of them with the monocular features of the first two blocks, transposed and projected at 1/4 and 1/8 of the input resolution, respectively. The output of this module is a sequence of $\frac{H}{16} \times \frac{W}{16}$ tokens, matching the sequence length of the monocular features. These tokens are then fed into a ViT-B initialized with DINOv2 weights (see Fig. 9.4).

For ii) we add the tokens from the cost volume with the ones from the reference image encoder after projecting the latter with a linear layer. We repeat this process at blocks 2, 5, 9, and 11 of the ViT to incorporate multiple levels of monocular cues. This simple mechanism allows the network to refine and regularize the cost volume with the help of the reference image structure.

### 9.1.4 Generalizing to Any Range of Depths

When building a cost volume, a set of depth hypotheses (*i.e.* bins) $\mathcal{B}$ are used to warp feature maps $\mathcal{F}_i$ to $\mathbf{I}_r$. This raises the question: How do we choose $\mathcal{B}$ to **generalize to any range of depths**? Depth ranges vary hugely across datasets (see Figure 9.5), so using the same fixed range is suboptimal.

We address this with a cascaded cost volume approach, first introduced in 3D stereo matching [181, 204, 205]. While these works start from a known 'ground truth' depth range, we use the known intrinsics and extrinsics to infer the minimum and maximum depths that could be matched between $\mathbf{I}_r$ and each $\mathbf{I}_i$. We space our initial depth bins logarithmically within this range, then make an initial depth prediction. The min and max values of this initial estimate are then used to rebuild the cost volume for a final depth prediction. This iterative process occurs only at test time; during training, we use the known depth range. Importantly, previous methods that are provided with an exact depth range

| Name | Scenes | #total scenes | # total images | # training tuples | Metric poses? | Moving objects? |
|---|---|---|---|---|---|---|
| Hypersim [250] | Indoor | 461 | 77K | 45K | Yes | No |
| TartanAIR [251] | Indoor, Outdoor | 30 | 1M | 92K | Yes | Yes |
| BlendedMVG [252] | Indoor, Outdoor, Aerial | 389 | 110K | 97K | No | No |
| MatrixCity [253] | Outdoor, Aerial | 1 | 519K | 40K | Yes | No |
| VKITTI2 [254, 255] | Outdoor | 5 | 21K | 40K | Yes | Yes |
| Dynamic Replica [256] | Indoor | 484 | 145K | 70K | Yes | Yes |
| MVSSynth [124] | Outdoor | 117 | 12K | 3K | No | Yes |
| SAIL-VOS 3D [257, 258] | Indoor, Outdoor | 6807 | 237K | 21K | Yes | Yes |

**Table 9.1:** We train on eight MVS datasets from a variety of domains. All these datasets are synthetically rendered, giving them perfect ground truth depths and camera calibration. However, BlendedMVG uses real textures on their assets.

learn to predict depths that cover all the depth hypotheses. Thus, when using a rough estimate of the range, these methods fail to align the prediction to the actual valid depths. To further mitigate this issue, we augment the ground truth ranges via a random perturbation during training.

### 9.1.5 Implementation Details

**Losses.** We use the supervised losses from [190]. These comprise an L1 loss between the log of the ground truth and the log of the predicted depth values, and a gradient and normals loss. Training losses are applied to four output scales of the decoder. At inference, only the final largest-scale prediction is used. We take as input $640 \times 480$ images, and output depth maps at the same resolution. We use 64 depth bins in $\mathcal{B}$ sampled in log space.

**Keyframes.** For datasets with dense sequences, we choose reference and source frames with the strategy of [189, 190]. To be robust to sparser sets of frames, we also select tuples based on geometry overlap, obtaining tuples of not necessarily consecutive frames.

**Training data.** For MVSA to **generalize across domains**, we train on a large and diverse set of synthetic datasets, as listed in Table 9.1. A subset of these training datasets contain moving objects. Our reference image encoder is initialized from Depth Anything V2 (DAV2) [148], which uses a teacher network trained on synthetic datasets similar to ours, and a student network distilled using various real images that do not overlap with our evaluation benchmarks. DAV2 was initialized from a pretrained DINOv2 [54] network, in turn trained on internet images.

## 9.2 Experiments

We evaluate MVSA on both depth estimation and 3D reconstruction tasks. We also implement and report scores for a set of new baselines, which have never before been evaluated on the benchmarks we use.

### 9.2.1 Baselines

Where possible, we obtain results directly from prior works [192, 243]. We also evaluate and implement other strong baselines that did not previously report performance on diverse MVS benchmarks. These include: (i) A strong monocular baseline in the form of DAV2 [148]. To account for the unknown affine transform, we align its predictions to the ground truth using least squares. (ii) MAST3R [193] (raw depth estimate) which involves passing the reference and

one other source image as input and taking the $z$ component of the point cloud as the depth prediction. (iii) MAST3R (plus our triangulation) which is a novel extension of MAST3R so that it can use provided extrinsics and intrinsics, when available. For each of the available source images, we use MAST3R descriptors to match points with the reference image. We then triangulate points from such matches, rescale the raw depth predictions, and aggregate the point clouds from the different views using a sum weighted by the predicted confidences. Note, this method requires one forward pass and thousands of triangulations per source view, significantly reducing its speed. MAST3R trains on ScanNet [247] and MegaDepth [226] (which contains a subset of the Tanks and Temples dataset [249]).

**Benchmark.** We evaluate 'zero-shot' depth estimation performance on the five multi-view datasets from the Robust Multi-View Depth Benchmark (RMVDB) [243], which are not included in our training data. It contains the KITTI [119] ScanNet [247], ETH3D [248], DTU [246], and Tanks and Temples [249] datasets and represents a diverse set of evaluation scenarios, *e.g.* driving sequences, room scans, building scans, and tabletop objects, among others. We use the evaluation procedure and source view selection procedure from [243], allowing direct comparison to previous approaches.

Methods are grouped into four different types (a-d) depending on the information they are provided, *e.g.* if they are given GT cameras, GT depth ranges, *etc.*. MVSA naturally fits into type (d), where all methods are given GT poses, so need to predict depth directly in metric scale and hence do not need any alignment or knowledge of the GT depth range. Note, some methods train on the training splits of one, or more, of the benchmark datasets, thus achieving very high scores in those cases. We denote these in Table 9.2 with a parenthesis around them.

**Metrics.** We report two commonly used metrics to compare the predicted $\mathbf{D}$ and GT depth $\mathbf{D}^{\text{GT}}$. The absolute relative depth (rel) is computed per-pixel as $|\mathbf{D} - \mathbf{D}^{\text{GT}}|/\mathbf{D}^{\text{GT}}$, while the inlier percentage $\tau$, with threshold 1.03, is computed per-pixel as $[\max(\mathbf{D}/\mathbf{D}^{\text{GT}}, \mathbf{D}^{\text{GT}}/\mathbf{D}) < 1.03]$, where $[]$ is the Iverson bracket. Both metrics are averaged over all valid GT pixels in each test image, before averaging over all images.

**Results.** Table 9.2 depicts the quantitative results, where we outputperform all baselines across most metrics. Qualitative results in Figure 9.8 demonstrate that our MVSA model produces depth maps with superior edge detail and consistent scaling across a variety of scenes, visually outperforming prior methods. MVSA also performs well on moving objects, *e.g.* as found in KITTI; see also Fig. 9.7. Both **MAST3R triangulated** and the **Robust MVD Baseline** exhibit poor edge quality, limiting their suitability for applications such as single-image novel view synthesis [259], which requires sharp depth boundaries. While Depth Pro produces sharp edges, it frequently displays incorrect depth scaling. In contrast, our MVSA model combines competitive quantitative performance with sharper edges, making it ideal for tasks that demand both visual and depth accuracy. Finally, GT depth-based median and least squares scaling of monocular methods and depth from frames methods (w/o poses) is crucial for good scores, while MVSA consistently predicts high-quality and metric depths.

**Alternative Variant of RMVDB.** We further evaluate some of the leading models on a RMVDB variant, in which we change some conditions to better reflect real-world scenarios. In this variant, for ScanNet we use keyframes using the strategy of [189], rather than the temporally sequential keyframes provided

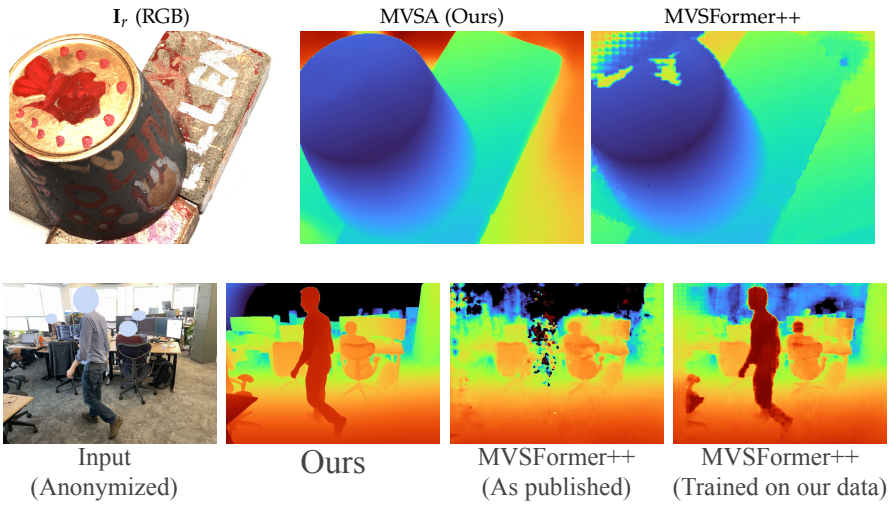



**Figure 9.6: Many MVS models fail in areas of poor frame overlap**. Here we show how MVSFormer++ (right) fails to recover geometry in areas of the image where there are no matching pixels between source and target views (see the top left corner). Our model (middle) handles this situation gracefully.

**Figure 9.7:** We handle **dynamic objects** significantly better than traditional MVS *e.g.* MVSFormer++.

by the benchmark. For ETH3D we undistort both the images and the ground truth using their provided Thin-Prism [260] camera parameters. Results are shown in Table 9.3. On this revised benchmark, we more comprehensively outperform the baselines.

### 9.2.2 Ablations Study

In Table 9.5 we validate our design decisions by turning on and off sections of our system. We train all ablations at a smaller resolution ($512 \times 384$ input), and without using metadata, for efficiency. At this resolution, Row **A** is 'ours' and all other rows are ablations relative to this. Row **B** replaces our standard ViT-B with the smaller ViT Small, both for the cost volume ViT and the reference image encoder. The reference image encoder is initialized from Depth Anything v2 (small). Row **C** uses our training data and pipeline, but with the fully-convolutional architecture from SimpleRecon [190] (without metadata). Row **D** is our system but without adding noise to the ground truth range at training time (Section 9.1.4). Although this method can excel when the initial range is accurate, it can fail to generalize (see DTU). Row **E** is our full architecture but without the pretrained encoder weights from [148]. Instead we initialize with DINOv2 weights. Row **F** is our system without a cascaded cost volume, and instead uses a fixed set of depth bins, losing the ability to refine depth bins and work with arbitrary scales or scene sizes. Row **H** is our full model, but where we take the first depth prediction from the model as our final output, without re-building the cost volume. Even though these bins capture the full range of depths in the test datasets (Figure 9.5), we see that performance degrades. Row **G** uses CNN layers instead of ViT to combine mono/multi features. Row **I** uses naive patchification to preprocess the cost volume for input to the mono/multi cue combiner ViT, as outlined in Figure 9.4.

**Robustness to pose rescaling.**

In Table 9.4 we evaluate the robustness to pose scale in ScanNet by rescaling poses and depths (which are in metric scale) by a factor of 100. This simulates the type of 'non-metric scene scale' we might see if, for example, we reconstructed a scene using SfM package such as COLMAP which does not provide metric scaled reconstructions. Our depths are robust to this scaling (*vs* ours w/o normalization of the metadata, which performs badly.

**Table 9.2: We set a new SOTA in depth estimation on the RMVDB.** See Section 9.2 for details of the metrics, baselines and groupings. Monocular methods with † are given ground truth intrinsics. The best result for each section appears in **bold**, and (parentheses) indicate results where the evaluation dataset is in the training set.

| Approach | GT Poses | GT Range | Align | KITTI rel ↓ | KITTI τ ↑ | ScanNet rel ↓ | ScanNet τ ↑ | ETH3D rel ↓ | ETH3D τ ↑ | DTU rel ↓ | DTU τ ↑ | T&T rel ↓ | T&T τ ↑ | Average rel ↓ | Average τ ↑ | Average time [s] ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Classical SfM approaches** | | | | | | | | | | | | | | | | |
| COLMAP [175, 261] | ✓ | ✗ | ✗ | **12.0** | **58.2** | **14.6** | **34.2** | **16.4** | **55.1** | **0.7** | **96.5** | **2.7** | **95.0** | **9.3** | **67.8** | ≈ 180 |
| COLMAP Dense [175, 261] | ✓ | ✗ | ✗ | 26.9 | 52.7 | 38.0 | 22.5 | 89.8 | 23.2 | 20.8 | 69.3 | 25.7 | 76.4 | 40.2 | 48.8 | ≈ 180 |
| **a) Depth from frames (w/o poses)** | | | | | | | | | | | | | | | | |
| DeMoN [262] | ✗ | ✗ | ‖t‖ | 15.5 | 15.2 | **12.0** | **21.0** | 17.4 | 15.4 | 21.8 | 16.6 | 13.0 | 23.2 | 16.0 | 18.3 | **0.08** |
| DeepV2D KITTI [177] | ✗ | ✗ | med | (**3.1**) | (**74.9**) | 23.7 | 11.1 | 27.1 | 10.1 | 24.8 | 8.1 | 34.1 | 9.1 | 22.6 | 22.7 | 2.07 |
| DeepV2D ScanNet [177] | ✗ | ✗ | med | 10.0 | 36.2 | (4.4) | (54.8) | 11.8 | 29.3 | 7.7 | 33.0 | **8.9** | **46.4** | 8.6 | 39.9 | 3.57 |
| MAST3R [193] (raw output) | ✗ | ✗ | med | **3.3** | **67.7** | (**4.3**) | (**64.0**) | **2.7** | **79.0** | **3.5** | **66.7** | (**2.4**) | (**81.6**) | **3.3** | **71.8** | 0.07 |
| MAST3R [193] (raw output) | ✗ | ✗ | ✗ | 61.4 | 0.4 | (12.8) | (19.4) | 43.8 | 3.1 | 145.8 | 0.5 | (66.9) | (0.0) | 66.1 | 4.7 | 0.07 |
| **b) Depth from frames and poses (with per-image range provided)** | | | | | | | | | | | | | | | | |
| MVSNet [123] | ✓ | ✓ | ✗ | 22.7 | 36.1 | 24.6 | 20.4 | 35.4 | 31.4 | (1.8) | (86.0) | 8.3 | 73.0 | 18.6 | 49.4 | **0.07** |
| MVSNet Inv. Depth [123] | ✓ | ✓ | ✗ | 18.6 | 30.7 | 22.7 | 20.9 | 21.6 | 35.6 | (1.8) | (86.7) | 6.5 | 74.6 | 14.2 | 49.7 | 0.32 |
| CVP-MVSNet [263] | ✓ | ✓ | ✗ | 156.7 | 2.2 | 137.1 | 15.9 | 156.4 | 13.6 | (4.0) | (68.4) | 24.7 | 52.9 | 95.8 | 30.6 | 0.49 |
| Vis-MVSNet [186] | ✓ | ✓ | ✗ | 9.5 | 55.4 | 8.9 | 33.5 | 10.8 | 43.3 | (1.8) | (87.4) | 4.1 | 87.2 | 7.0 | 61.4 | 0.70 |
| PatchmatchNet [242] | ✓ | ✓ | ✗ | 10.8 | 45.8 | 8.5 | 35.3 | 19.1 | 34.8 | (2.1) | (82.8) | 4.8 | 82.9 | 9.1 | 56.3 | 0.28 |
| Fast-MVSNet [191] | ✓ | ✓ | ✗ | 14.4 | 37.1 | 17.0 | 24.6 | 25.2 | 32.0 | (2.5) | (81.8) | 8.3 | 68.6 | 13.5 | 48.8 | 0.30 |
| MVS2D ScanNet [233] | ✓ | ✓ | ✗ | 21.2 | 8.7 | (27.2) | (5.3) | 27.4 | 4.8 | 17.2 | 9.8 | 29.2 | 4.4 | 24.4 | 6.6 | **0.04** |
| MVS2D DTU [233] | ✓ | ✓ | ✗ | 226.6 | 0.7 | 32.3 | 11.1 | 99.0 | 11.6 | (3.6) | (64.2) | 25.8 | 28.0 | 77.5 | 23.1 | 0.05 |
| MVSFormer++ DTU [185] | ✓ | ✓ | ✗ | 26.3 | 42.8 | 16.7 | 28.0 | 30.3 | 40.1 | (**0.8**) | (**95.7**) | 7.2 | 82.3 | 16.3 | 57.8 | 0.78 |
| MVSFormer++ DTU+BlendedMVG [185] | ✓ | ✓ | ✗ | **4.4** | **65.7** | **7.9** | **39.4** | **7.8** | **50.4** | (0.9) | (95.3) | **3.2** | **88.1** | **4.8** | **67.8** | 0.78 |
| **c) Single-view depth** | | | | | | | | | | | | | | | | |
| Depth Pro [153] † | ✗ | ✗ | med | 6.1 | 39.6 | (4.3) | (58.4) | 6.1 | 53.5 | 5.6 | 49.6 | 5.6 | 57.5 | 5.6 | 51.7 | 5.16 |
| Depth Pro [153] † | ✗ | ✗ | ✗ | 13.6 | 14.3 | 9.2 | 19.7 | 28.5 | 8.7 | 161.8 | 3.5 | 38.3 | 4.4 | 50.3 | 10.1 | 5.16 |
| Metric3D [12] † | ✗ | ✗ | med | 5.1 | 44.1 | **2.4** | **78.3** | 4.4 | 54.5 | 10.1 | 39.5 | 6.2 | 48.0 | 5.6 | 52.9 | 0.46 |
| Metric3D [12] † | ✗ | ✗ | ✗ | 8.7 | 13.2 | 6.2 | 19.3 | 12.7 | 13.0 | 890.5 | 1.4 | 16.7 | 13.7 | 187.0 | 12.1 | 0.46 |
| UniDepthV2 [235] † | ✗ | ✗ | med | **4.0** | **55.3** | (2.1) | (82.6) | **3.7** | **66.2** | 3.2 | 72.3 | **3.6** | **68.4** | **3.3** | **68.9** | 0.29 |
| UniDepthV2 [235] † | ✗ | ✗ | ✗ | 13.7 | 4.8 | (3.2) | (61.3) | 15.4 | 11.9 | 964.8 | 1.3 | 16.7 | 12.7 | 202.7 | 18.4 | 0.29 |
| UniDepthV1 [235] † | ✗ | ✗ | med | 4.4 | 51.6 | (**1.9**) | (**84.3**) | 5.4 | 48.4 | 9.3 | 31.8 | 9.6 | 38.7 | 6.1 | 51.0 | 0.21 |
| UniDepthV1 [235] † | ✗ | ✗ | ✗ | 5.2 | 39.5 | (2.7) | (69.4) | 48.2 | 1.8 | 583.3 | 1.0 | 30.7 | 4.2 | 134.0 | 23.2 | 0.20 |
| DepthAnything V2 (ViT-B) [148] | ✗ | ✗ | lstsq † | 6.6 | 38.6 | 4.0 | 58.6 | 4.7 | 56.5 | **2.6** | **74.7** | 4.5 | 57.5 | 4.8 | 54.1 | **0.05** |
| **d) Depth from frames and poses (w/o per-image range)** | | | | | | | | | | | | | | | | |
| DeMoN [262] | ✓ | ✗ | ✗ | 16.7 | 13.4 | 75.0 | 0.0 | 19.0 | 16.2 | 23.7 | 11.5 | 17.6 | 18.3 | 30.4 | 11.9 | 0.08 |
| DeepTAM [264] | ✓ | ✗ | ✗ | 68.7 | 0.4 | (6.7) | (39.7) | 20.4 | 19.8 | 58.0 | 9.1 | 40.0 | 12.9 | 38.8 | 16.4 | 0.85 |
| DeepV2D KITTI [177] | ✓ | ✗ | ✗ | (20.4) | (16.3) | 25.8 | 8.1 | 30.1 | 9.4 | 24.6 | 8.2 | 38.5 | 9.6 | 27.9 | 10.3 | 1.43 |
| DeepV2D ScanNet [177] | ✓ | ✗ | ✗ | 61.9 | 5.2 | (**3.8**) | (60.2) | 18.7 | 28.7 | 9.2 | 27.4 | 33.5 | 38.0 | 25.4 | 31.9 | 2.15 |
| MVSNet [123] | ✓ | ✗ | ✗ | 14.0 | 35.8 | 1568.0 | 5.7 | 507.7 | 8.3 | (4429.1) | (0.1) | 118.2 | 50.7 | 1327.4 | 20.1 | 0.15 |
| MVSNet Inv. Depth [123] | ✓ | ✗ | ✗ | 29.6 | 8.1 | 65.2 | 28.5 | 60.3 | 5.8 | (28.7) | (48.9) | 51.4 | 14.6 | 47.0 | 21.2 | 0.28 |
| CVP-MVSNet [263] | ✓ | ✗ | ✗ | 158.2 | 1.2 | 2289.0 | 0.1 | 1735.3 | 1.2 | (8314.0) | (0.0) | 415.9 | 9.5 | 2582.5 | 2.4 | 0.50 |
| Vis-MVSNet [186] | ✓ | ✗ | ✗ | 10.3 | 54.4 | 84.9 | 15.6 | 51.5 | 17.4 | (374.2) | (1.7) | 21.1 | 65.6 | 108.4 | 31.0 | 0.82 |
| PatchmatchNet [242] | ✓ | ✗ | ✗ | 29.0 | 16.3 | 70.1 | 16.7 | 99.4 | 3.5 | (82.6) | (5.6) | 39.4 | 19.3 | 64.1 | 12.3 | 0.18 |
| Fast-MVSNet [191] | ✓ | ✗ | ✗ | 12.1 | 37.4 | 287.1 | 9.4 | 131.2 | 9.6 | (540.4) | (1.9) | 33.9 | 47.2 | 200.9 | 21.1 | 0.35 |
| MVS2D ScanNet [233] | ✓ | ✗ | ✗ | 73.4 | 0.0 | (4.5) | (54.1) | 30.7 | 14.4 | 5.0 | 57.9 | 56.4 | 11.1 | 34.0 | 27.5 | **0.05** |
| MVS2D DTU [233] | ✓ | ✗ | ✗ | 93.3 | 0.0 | 51.5 | 1.6 | 78.0 | 0.0 | (**1.6**) | (92.3) | 87.5 | 0.0 | 62.4 | 18.8 | 0.06 |
| Robust MVD Baseline [243] | ✓ | ✗ | ✗ | 7.1 | 41.9 | 7.4 | 38.4 | 9.0 | 42.6 | 2.7 | 82.0 | 5.0 | 75.1 | 6.3 | 56.0 | 0.06 |
| MAST3R (plus our triangulation) | ✓ | ✗ | ✗ | 3.4 | 66.6 | (4.5) | (**63.0**) | **3.1** | **72.9** | 3.4 | 67.3 | (2.4) | (83.3) | 3.4 | 70.1 | 0.72 |
| **MVSA** | ✓ | ✗ | ✗ | **3.2** | **68.8** | **3.7** | **62.9** | 3.2 | 68.0 | **1.3** | **95.0** | **2.1** | **90.5** | **2.7** | **77.0** | 0.12 |

| Approach | ScanNet | | ETH3D | |
|---|---|---|---|---|
| | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ |
| Robust MVD Baseline [243] | 6.02 | 47.83 | 5.75 | 71.64 |
| MAST3R Triangulated | (3.88) | (68.68) | 2.37 | 84.90 |
| **MVSA** (Ours) | **3.22** | **69.45** | **1.27** | **93.24** |

**Table 9.3: Our variant of RMVDB.** We use better test-time tuples for ScanNet, and for ETH3D we use the undistorted test images.

| Approach | ScanNet | | ScanNet ×100 | |
|---|---|---|---|---|
| | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ |
| Ours (low res) | 4.22 | 61.80 | 4.22 | 61.83 |
| Ours w/o norm. metadata w/o view count agnostic | 3.97 | 61.23 | 4.34 | 57.59 |

**Table 9.4: Results on an arbitrary scale.** We scale the ScanNet poses by a factor of 100 to evaluate robustness to arbitrary scales.

| | KITTI | | ScanNet | | ETH3D | | DTU | | T&T | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ | rel ↓ | $\tau$ ↑ |
| A Ours (no metadata, low res.) | 3.39 | **66.88** | 3.86 | 60.82 | **3.11** | **70.17** | 2.43 | 92.05 | **2.23** | **88.38** | **3.00** | **75.66** |
| B w/ ViT Small | 3.57 | 64.34 | 4.40 | 56.57 | 3.63 | 64.61 | 2.69 | 91.52 | 2.71 | 84.51 | 3.40 | 72.31 |
| C w/ [190]'s architecture | 3.63 | 65.94 | 5.03 | 51.76 | 3.74 | 63.09 | **1.77** | 90.97 | 2.78 | 87.90 | 3.39 | 71.93 |
| D w/o noise on GT range | **3.33** | 66.83 | 5.14 | 52.77 | 3.53 | 66.32 | 13.45 | 89.21 | 2.34 | 87.64 | 5.32 | 74.89 |
| E w/o DAV2 weights | 3.45 | 65.42 | 4.58 | 57.14 | 3.48 | 65.59 | 2.14 | **92.48** | 2.56 | 86.01 | 3.24 | 73.33 |
| F w/ fixed bins [0-100m] | 3.41 | 64.33 | **3.80** | **61.62** | 3.15 | 67.20 | 4.11 | 65.64 | 2.36 | 85.72 | 3.37 | 68.90 |
| G no MMCC ViT | 3.54 | 65.32 | 4.39 | 56.94 | 3.56 | 65.27 | 3.07 | 90.49 | 2.46 | 87.54 | 3.41 | 73.11 |
| H w/o bin refinement | 3.57 | 63.39 | 5.18 | 51.05 | 3.50 | 67.93 | 6.80 | 82.12 | 2.27 | 87.04 | 4.26 | 70.31 |
| I Naive patchify | 3.66 | 62.61 | 4.27 | 58.86 | 3.18 | 67.27 | 1.95 | 91.69 | 2.46 | 86.52 | 3.11 | 73.39 |

**Table 9.5: Ablation Study.** Here we validate our design decisions on RMVDB [243] by ablating various components. See Subsection 9.2.2 for details. **First** and second best scores are indicated.

### 9.2.3 Meshing and 3D Reconstruction

To judge the 3D-consistency of our predictions we evaluate our model on ScanNet Mesh Evaluation benchmark using the protocol defined in [265] which also uses source frame selection from [189]. The benchmark uses a GT mesh collected with an active RGBD sensor captured in a video. The evaluation computes point-to-point vertex error from GT to predicted (as accuracy), from predicted to GT (as completion), and the average of the two (as chamfer). Additionally, 200k points are sampled uniformly over each mesh and point-to-point errors thresholded at 5cm distance are used to compute precision, recall and F-score. Almost all competing methods are trained on ScanNet, however our method that was not trained on ScanNet performs comparatively, outperforming many of the methods in Table 9.6.

### 9.2.4 Gaussian Splat Regularization using MVSAnywhere

We show a use case of our method as a regularizer for Gaussian Splats. We follow a similar approach as VCR-GauS [270] and DN-Splatter [271] regularizing depth and normals during training as additional losses. In Figure 9.9, we show qualitative results of the meshes obtained after using raw Splatfacto without regularization, and with normal and depth supervision from Metric3D [12] and from MVSA. Note that we used an scale-invariant loss for Metric3D given that the used scenes lay in arbitrary scales. More details on this regularization are available on the code.

**Limitations.** While we use multi-view information to generate depths, we do not enforce or encourage temporal consistency. Techniques for this [198, 215, 240] could work with MVSA. Also, like traditional MVS, our method requires known camera intrinsics and poses; recent works suggest this requirement could be relaxed [272, 273].

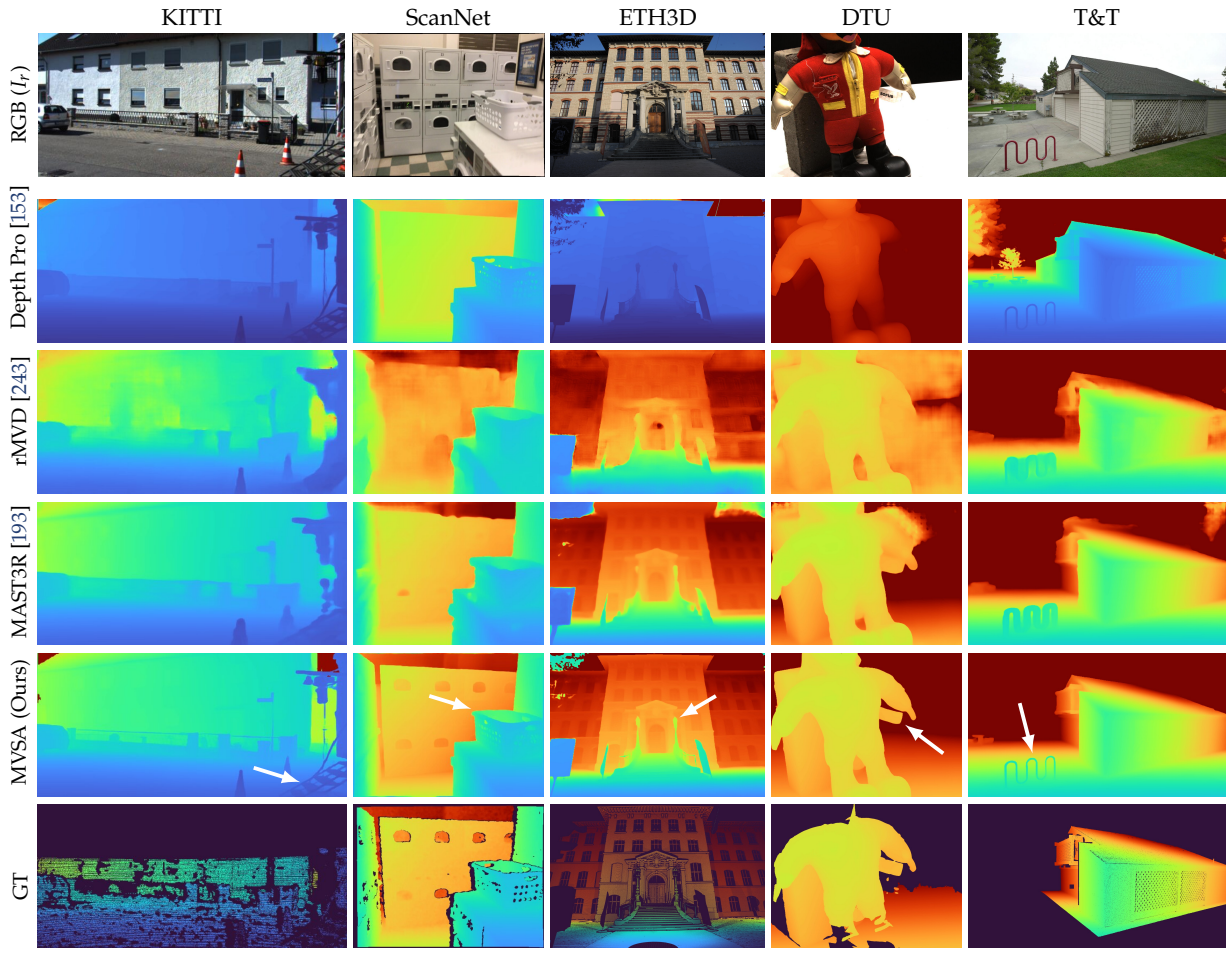



**Figure 9.8: Qualitative comparison of depth prediction results across multiple datasets** (KITTI, ScanNet, ETH3D, DTU, and Tanks & Temples). Rows show different methods: Depth Pro [243], rMVD baseline [243], MAST3R (Triangulated) [193], and our MVSA model, along with RGB inputs ($\mathbf{I}_r$) and ground-truth depths (GT). Depth Pro provides sharp edges but often misestimates depth scale, while our MVSA model captures finer details than MAST3R and rMVD. Depth maps are normalized to ground truth depth range for consistent visualization.

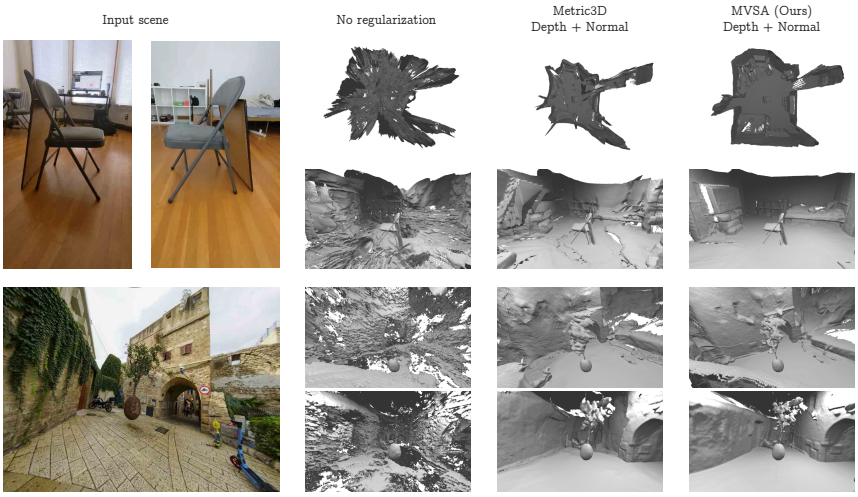



**Figure 9.9: Qualitative comparison of Gaussian Splat meshes using Metric3D and MVSA as regularizers.**

| | | Comp↓ | Acc↓ | Chamfer↓ | Prec↑ | Recall ↑ | F-Score ↑ |
|---|---|---|---|---|---|---|---|
| DeepVideoMVS [189] | **S** | 10.68 | 6.90 | 8.79 | 0.541 | 0.592 | 0.563 |
| ATLAS [266] | **S** | 7.16 | 7.61 | 7.38 | 0.675 | 0.605 | 0.636 |
| NeuralRecon [267] | **S** | 5.09 | 9.13 | 7.11 | 0.630 | 0.612 | 0.619 |
| 3DVNet [268] | **S** | 7.72 | 6.73 | 7.22 | 0.655 | 0.596 | 0.621 |
| TransformerFusion [265] | **S** | 5.52 | 8.27 | 6.89 | 0.728 | 0.600 | 0.655 |
| VoRTX [269] | **S** | **4.31** | 7.23 | **5.77** | **0.767** | 0.651 | **0.703** |
| SimpleRecon [190] | **S** | 5.53 | **6.09** | 5.81 | 0.686 | **0.658** | 0.671 |
| COLMAP [175] | | 10.22 | 11.88 | 11.05 | 0.509 | 0.474 | 0.489 |
| MAST3R [193] (raw depth) | **S+** | 12.35 | 12.69 | 12.52 | 0.265 | 0.283 | 0.272 |
| MAST3R [193] (+ triangulation) | **S+** | 5.38 | 6.78 | 6.08 | 0.572 | 0.655 | 0.608 |
| SimpleRecon [190] (trained on our data) | | 8.07 | 6.67 | 7.37 | 0.501 | 0.597 | 0.544 |
| **MVSA** (Ours) | | **4.93** | **6.39** | **5.66** | **0.616** | **0.696** | **0.652** |

**Table 9.6: ScanNet Mesh Evaluation [265].** Scores adapted from [190, 265]. Rows marked with **S** were trained on ScanNet only, while those marked **S+** were trained on ScanNet and other datasets. Our MVSA model, which was not trained on ScanNet, outperforms many models which were, *e.g.* [266–268].

## 9.3 Conclusions

We introduced MVSAnywhere, a new general-purpose MVS depth estimation approach. We addressed challenges associated with training on diverse MVS datasets, such as how to best leverage ViT-based architectures, how to incorporate geometric metadata, and how to handle variable depth ranges. Through extensive experimentation, we compare to numerous existing and new baselines. Our contributions result in state-of-the-art zero-shot performance on a range of challenging reconstruction and depth estimation test datasets, in some cases even outperforming models trained on the test domains.

# 10 Conclusion

This thesis has explored two fundamental tasks of spatial perception within the broader field of Spatial AI: Visual Place Recognition and multi-view depth estimation.

In the first part of the thesis, we addressed VPR. In Chapter 4 we described how to leverage a large feature extractor effectively and introduced our novel module SALAD. Although previous attempts have been made to use large vision models as backbones in VPR, we demonstrate how fine-tuning DINOv2 improves over the out-of-the-box version and produces state-of-the-art results. Besides, we presented SALAD, a novel aggregation module that can assign features to clusters more effectively than NetVLAD, outperforming previous methods even with smaller descriptors.

In Chapter 5, we analyzed how previous VPR methods, including SALAD, struggle to correlate descriptor and geographic distance, especially around the decision threshold. In light of this, we identify and describe the Geographic Distance Sensitivity of recent models—the ability to assign smaller descriptor distances to pairs of images that are geographically closer. To overcome this, we proposed a novel mining strategy, CliqueMining, which samples batches with challenging hard negative examples. Training with this novel strategy results in a boost in the GDS and improved metrics in densely sampled and visually aliased environments like Nordland or MSLS Challenge.

This first part of the thesis has significantly advanced the state of the art in VPR. With DINOv2 SALAD, deep models are easier and faster to train resulting in higher recalls and improved robustness. Combining this with CliqueMining further improves the metrics on dense and unsaturated datasets.

Despite these advancements, we have observed that current models struggle to correctly rank images just based on descriptor distances. In future work, we aim to explicitly train to retrieve the correct order of the retrieved images. Using contrastive losses reduces VPR to a binary problem, where samples are either positive (same place) or negative (different place). Instead, we suggest incorporating the geographic distance into the loss, to enforce smaller descriptor distances to pairs of images that are indeed closer geographically.

The second part of this thesis focused on leveraging multiple views of a scene for depth estimation. In Chapter 8, we propose an effective approach to enrich single-view depth methods with image sequence information. We developed a test-time refinement of the networks supervising them with the sparse signal of a COLMAP reconstruction. Although this setup poses some challenges, like outliers in the sparse reconstruction or different scales between the network and COLMAP, we propose a RANSAC alignment that correctly retrieves the scale and removes possible outliers. As a result, the refined methods can leverage the wide baseline information from COLMAP and produce much more accurate guesses at large depths, clearly outperforming the commonly used photometric refinement.

Lastly, in Chapter 9, we developed a general-purpose multi-view depth system, that we named MVSAnywhere. This model takes inspiration from single-view depth approaches and addresses the challenges that limit the widespread of multi-view systems. It is trained on multiple synthetic datasets to have strong

generalization, leverages the ViT architecture for single and multi-view, has strong monocular features to handle dynamic or low overlapping scenes, does not require depth range as input, and uses cost-volume metadata but works with different scales. Extensive evaluation shows how MVSAnywhere results in state-of-the-art zero-shot performance on varied and challenging testing datasets.

These two works demonstrate how multi-view information can be leveraged for depth estimation resulting in better accuracies than single-view models. Although this is unsurprising and multi-view depth has been long studied, most of the approaches focused on small datasets or the same train and test distribution, and the community lacked more general systems. In this thesis, we show both the importance and accuracy of multi-view and take some of the first steps towards general-purpose multi-view depth models.

The next steps in this line of research may focus on how to aggregate multi-view information without any knowledge or heuristic about the scene depth range, thus avoiding the two-step refinement. To achieve this, the cost volume may be built by sampling along epipolar lines and concatenating these values with the depth at those points. This will allow for a more effective use of the correlation, as only valid and feasible pixels will be sampled. Another line of research could explore how to leverage powerful matching features in the cost volume, such as those of MAST3R. As these features are trained to be matched, they could be an excellent candidate to build the cost volume, especially texture-less areas, aliased patterns, or strong illumination changes, where current feature extractors tend to fail.

# Conclusión

En esta tesis se han explorado dos tareas fundamentales de la percepción espacial dentro del campo más amplio de la inteligencia artificial espacial: el Reconocimiento Visual de Lugares o VPR y la estimación de profundidad multivista.

En la primera parte de la tesis, abordamos el reconocimiento visual de lugares. En el Capítulo 4 describimos cómo aprovechar eficazmente un extractor de características de gran capacidad e introdujimos el nuevo módulo SALAD. Aunque ha habido intentos previos de utilizar modelos visuales grandes como extractor de características en VPR, nosotros hemos demostrado cómo seguir entrenando DINOv2 para la tarea de VPR mejora sustancialmente los resultados respecto a la versión original. Además, presentamos SALAD, un novedoso módulo de agregación capaz de asignar características a clústeres de forma más efectiva que NetVLAD, superando a métodos anteriores incluso utilizando descriptores más pequeños.

En el Capítulo 5 analizamos cómo los métodos previos de VPR, incluido SALAD, presentan dificultades para correlar la distancia en el espacio de descriptores con la distancia geográfica, especialmente cerca del umbral de decisión. A raíz de este análisis, identificamos y describimos la sensibilidad a la distancia geográfica (GDS, por sus siglas en inglés) de los modelos actuales: la capacidad de asignar distancias menores a los pares de imágenes más cercanas geográficamente. Para solventar este problema, propusimos una nueva estrategia de muestreo de datos, CliqueMining, que selecciona conjuntos de imágenes con ejemplos negativos difíciles para la red. Entrenar con esta nueva estrategia proporciona una mejora en la GDS y aumenta las métricas en entornos densos, como Nordland o MSLS Challenge.

Esta primera parte de la tesis ha traído una mejora significativa del estado del arte en VPR. Con DINOv2 SALAD, los modelos profundos se entrenan de manera más sencilla y rápida, obteniendo mejores métricas y una mayor robustez. Combinando esto con CliqueMining se logran aún mejores resultados en conjuntos de datos densos.

A pesar de estos avances, hemos observado que los modelos actuales todavía presentan dificultades para ordenar correctamente las imágenes únicamente en base a las distancias de sus descriptores. Para trabajos futuros, proponemos entrenar explícitamente para recuperar el orden correcto en la lista de imágenes recuperadas. El uso de funciones de coste basadas en contraste reduce el problema de VPR a una clasificación binaria (mismo o distinto lugar), pero consideramos que incorporar la distancia geográfica en la función de coste permitiría forzar distancias menores en el espacio de descriptores a aquellas imágenes que estén realmente más próximas geográficamente.

La segunda parte de esta tesis se ha centrado en aprovechar múltiples vistas de una escena para la estimación de profundidad. En el Capítulo 8, proponemos un nuevo método para incorporar información multivista en sistemas de estimación de profundidad monoculares. Desarrollamos un refinamiento en tiempo de inferencia que utiliza como objetivo la señal dispersa de una reconstrucción COLMAP. Aunque esta configuración presenta desafíos, como la presencia de errores en la reconstrucción o las diferencias de escala entre la red y COLMAP, propusimos una alineación mediante RANSAC que recupera correctamente

la escala y elimina posibles errores. Como resultado, los métodos, una vez refinados, pueden aprovechar la información multivista proporcionada por COLMAP y producir predicciones mucho más precisas a grandes distancias, superando claramente al refinamiento fotométrico habitual.

Por último, en el Capítulo 9, desarrollamos un sistema de estimación de profundidad multivista de propósito general al que denominamos MVSAnywhere. Este modelo se inspira en sistemas monoculares y aborda los principales desafíos que limitaban la popularización de los métodos multivista. El sistema está entrenado con datos sintéticos variados para lograr generalizar a nuevas escenas, aprovecha arquitecturas ViT tanto para la información monocular como la multivista, presenta características monoculares robustas que permiten manejar escenas con baja superposición o elementos dinámicos, no requiere conocer de antemano el rango de profundidades, e incorpora metadatos en el volumen de costes manteniendo compatibilidad con distintas escalas. Hemos evaluado el sistema de forma exhaustiva, demostrando que MVSAnywhere alcanza los mejores resultados en distintas evaluaciones.

Estos dos trabajos demuestran cómo la información multivista puede aprovecharse para obtener estimaciones de profundidad más precisas que los modelos monoculares. Aunque esto no es nada nuevo y la estimación multivista ha sido estudiada durante décadas, la mayoría de métodos anteriores se limitaban a conjuntos de datos pequeños o a distribuciones de entrenamiento y evaluación similares. En esta tesis mostramos tanto la importancia como el potencial de los métodos multivista, dando algunos de los primeros pasos hacia modelos multivista de propósito general.

Las siguientes líneas de investigación pueden centrarse en cómo agregar información multivista sin necesidad de conocimiento previo o heurísticas sobre el rango de profundidad de la escena, evitando así el proceso de refinamiento en dos pasos. Para lograrlo, el volumen de costes podría construirse muestreando directamente a lo largo de las líneas epipolares y concatenando estos valores con la profundidad en esos puntos. Esto permitiría un uso más eficiente de la correlación, ya que sólo se muestrearían píxeles válidos. Otra línea futura sería investigar cómo aprovechar descriptores potentes entrenados específicamente para tareas de matching, como los de MAST3R. Dado que estas características están entrenadas para ser emparejadas, podrían ser una excelente opción para construir el volumen de costes, especialmente en regiones con baja textura, patrones repetidos o cambios de iluminación, donde los extractores actuales tienden a fallar.

# Bibliography

Here are the references in citation order.


[1] Andrew J Davison. 'FutureMapping: The computational structure of spatial AI systems'. In: *Arxiv preprint* (2018) (cited on page 1).

[2] David G Lowe. 'Distinctive image features from scale-invariant keypoints'. In: *International Journal of Computer Vision (IJCV)* 60 (2004), pp. 91–110 (cited on page 1).

[3] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. 'LoFTR: Detector-free local feature matching with transformers'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 8922–8931 (cited on pages 1, 8, 17).

[4] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 'Superglue: Learning feature matching with graph neural networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4938–4947 (cited on pages 1, 8, 11, 12).

[5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 'NetVLAD: CNN architecture for weakly supervised place recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5297–5307 (cited on pages 1, 5, 7–10, 13, 14, 25).

[6] Gabriele Berton, Carlo Masone, and Barbara Caputo. 'Rethinking visual geo-localization for large-scale applications'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 4878–4888 (cited on pages 1, 5, 7, 13–15, 25, 29).

[7] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. 'Benchmarking 6dof outdoor visual localization in changing conditions'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8601–8610 (cited on page 1).

[8] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. 'Back to the feature: Learning robust camera localization from pixels to pose'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 3247–3257 (cited on pages 1, 19).

[9] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. 'Map-free visual relocalization: Metric pose relative to a single image'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 690–708 (cited on page 1).

[10] David Eigen, Christian Puhrsch, and Rob Fergus. 'Depth map prediction from a single image using a multi-scale deep network'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 27. 2014 (cited on pages 1, 31, 32, 40).

[11] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. 'AdaBins: Depth Estimation using Adaptive Bins'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 4009–4018 (cited on pages 1, 31, 32, 35, 39, 42, 43).

[12] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. 'Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-shot Metric Depth and Surface Normal Estimation'. In: *PAMI* (2024) (cited on pages 1, 31, 32, 46, 55, 56).

[13] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 'Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age'. In: *IEEE Transactions on Robotics (T-RO)* 32.6 (2016), pp. 1309–1332 (cited on pages 1, 5, 9, 19, 35).

[14] Raul Mur-Artal and Juan D Tardós. 'Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras'. In: *IEEE Transactions on Robotics (T-RO)* 33.5 (2017), pp. 1255–1262 (cited on pages 1, 5).

[15] Sergio Izquierdo and Javier Civera. 'SFM-TTR: Using Structure from Motion for Test-Time Refinement of Single-View Depth Networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 21466–21476 (cited on pages 2, 3, 35, 46).

[16] Sergio Izquierdo and Javier Civera. 'Optimal Transport Aggregation for Visual Place Recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 17658–17668 (cited on pages 2, 3, 9, 19–22, 24–28).

[17] Sergio Izquierdo and Javier Civera. 'Close, But Not There: Boosting Geographic Distance Sensitivity in Visual Place Recognition'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2024, pp. 240–257 (cited on pages 2, 3, 19).

[18] Sergio Izquierdo, Mohamed Sayed, Michael Firman, Guillermo Garcia-Hernando, Daniyar Turmukhambetov, Javier Civera, Oisin Mac Aodha, Gabriel J. Brostow, and Jamie Watson. 'MVSAnywhere: Zero Shot Multi-View Stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on pages 2, 3, 46).

[19] Blanca Lasheras-Hernandez, Klaus H Strobl, Sergio Izquierdo, Tim Bodenmüller, Rudolph Triebel, and Javier Civera. 'Single-Shot Metric Depth from Focused Plenoptic Cameras'. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2025 (cited on page 2).

[20] GeoGuessr. *GEOGUESSR Explore the world!* URL: https://www.geoguessr.com/ (visited on 06/12/2025) (cited on page 5).

[21] WhereTaken. *WHERETAKEN GUESS WHERE A PHOTO WAS TAKEN*. URL: https://wheretaken.com/ (visited on 06/12/2025) (cited on page 5).

[22] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. 'Efficient & effective prioritized matching for large-scale image-based localization'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 39.9 (2016), pp. 1744–1756 (cited on page 5).

[23] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. 'LaMAR: Benchmarking Localization and Mapping for Augmented Reality'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022 (cited on page 5).

[24] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. 'A comparison of loop closing techniques in monocular SLAM'. In: *Proceedings of Robotics: Science and Systems (RSS)* 57.12 (2009), pp. 1188–1197 (cited on page 5).

[25] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. 'OverlapNet: Loop Closing for LiDAR-based SLAM'. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2020 (cited on page 5).

[26] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. 'Visual place recognition: A survey'. In: *IEEE Transactions on Robotics (T-RO)* 32.1 (2015), pp. 1–19 (cited on pages 5, 7).

[27] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. 'On the performance of convnet features for place recognition'. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 4297–4304 (cited on pages 5, 7).

[28] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. 'Anyloc: Towards universal visual place recognition'. In: *Arxiv preprint* (2023) (cited on pages 5, 7, 9, 10, 14).

[29] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. 'Mixvpr: Feature mixing for visual place recognition'. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 2998–3007 (cited on pages 5, 7, 8, 10, 13, 14, 19, 24, 25).

[30] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 'Fine-tuning CNN image retrieval with no human annotation'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 41.7 (2018), pp. 1655–1668 (cited on pages 5, 7–9, 13, 14, 25).

[31] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 'A metric learning reality check'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 681–699 (cited on pages 5, 7, 8).

[32] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. 'Data-efficient large scale place recognition with graded similarity supervision'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 23487–23496 (cited on pages 5, 7, 8, 19).

[33] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. 'GSV-Cities: Toward appropriate supervised visual place recognition'. In: *Neurocomputing* 513 (2022), pp. 194–203 (cited on pages 5, 8, 13, 14, 19, 22, 27).

[34] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 'Sampling matters in deep embedding learning'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017, pp. 2840–2848 (cited on pages 5, 8).

[35] Xiwu Zhang, Lei Wang, and Yan Su. 'Visual place recognition: A survey from deep learning perspective'. In: *Pattern Recognition* 113 (2021), p. 107760 (cited on page 7).

[36] Carlo Masone and Barbara Caputo. 'A survey on deep visual place recognition'. In: *IEEE Access* 9 (2021), pp. 19516–19547 (cited on page 7).

[37] Sourav Garg, Tobias Fischer, and Michael Milford. 'Where is your place, visual place recognition?' In: *Arxiv preprint* (2021) (cited on pages 7, 9).

[38] Stefan Schubert, Peer Neubert, Sourav Garg, Michael Milford, and Tobias Fischer. 'Visual Place Recognition: A Tutorial'. In: *Arxiv preprint* (2023) (cited on page 7).

[39] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. 'Mapillary street-level sequences: A dataset for lifelong place recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2626–2635 (cited on pages 7, 8, 13, 20, 21, 25).

[40] Sourav Garg, Madhu Vankadari, and Michael Milford. 'SeqMatchNet: Contrastive learning with sequence matching for place recognition & relocalization'. In: *Conference on Robot Learning (CoRL)*. PMLR. 2022, pp. 429–443 (cited on page 7).

[41] Olga Vysotska and Cyrill Stachniss. 'Effective visual place recognition using multi-sequence maps'. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1730–1736 (cited on page 7).

[42] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. 'Learning deep representations for ground-to-aerial geolocalization'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5007–5015 (cited on page 7).

[43] Ziyang Hong, Yvan Petillot, David Lane, Yishu Miao, and Sen Wang. 'TextPlace: Visual place recognition and topological localization through reading scene texts'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019, pp. 2861–2870 (cited on page 7).

[44] Mark Cummins and Paul Newman. 'FAB-MAP: Probabilistic localization and mapping in the space of appearance'. In: *International Journal of Robotics Research (IJRR)* 27.6 (2008), pp. 647–665 (cited on page 7).

[45] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 'Aggregating local descriptors into a compact image representation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2010, pp. 3304–3311 (cited on pages 7, 9).

[46] Relja Arandjelovic and Andrew Zisserman. 'All about VLAD'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1578–1585 (cited on page 7).

[47] Niko Sünderhauf and Peter Protzel. 'Brief-gist-closing the loop by simple means'. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2011, pp. 1234–1241 (cited on page 7).

[48] Ana C Murillo, Gautam Singh, Jana Kosecka, and José Jesús Guerrero. 'Localization in urban environments using a panoramic gist descriptor'. In: *IEEE Transactions on Robotics (T-RO)* 29.1 (2012), pp. 146–160 (cited on page 7).

[49] Dorian Gálvez-López and Juan D Tardos. 'Bags of binary words for fast place recognition in image sequences'. In: *IEEE Transactions on Robotics (T-RO)* 28.5 (2012), pp. 1188–1197 (cited on page 7).

[50] Michael J Milford and Gordon F Wyeth. 'SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights'. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2012, pp. 1643–1649 (cited on page 7).

[51] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. 'Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 14141–14152 (cited on pages 7, 9, 14, 15, 19).

[52] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. 'Transvpr: Transformer-based place recognition with multi-level attention aggregation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 13648–13657 (cited on pages 7, 9, 14, 15, 19).

[53] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. 'R2former: Unified retrieval and reranking transformer for place recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 19370–19380 (cited on pages 7, 9, 14, 15, 19).

[54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 'Dinov2: Learning robust visual features without supervision'. In: *Arxiv preprint* (2023) (cited on pages 7, 9, 10, 15, 32, 52).

[55] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. 'Detect-to-retrieve: Efficient regional aggregation for image search'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5109–5118 (cited on page 7).

[56] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. 'InLoc: Indoor visual localization with dense matching and view synthesis'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7199–7209 (cited on page 7).

[57] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 'From coarse to fine: Robust hierarchical localization at large scale'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12716–12725 (cited on page 7).

[58] Bingyi Cao, Andre Araujo, and Jack Sim. 'Unifying deep local and global features for image search'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 726–743 (cited on pages 7, 19).

[59] Shihao Shao, Kaifeng Chen, Arjun Karpur, Qinghua Cui, André Araujo, and Bingyi Cao. 'Global Features are All You Need for Image Retrieval and Reranking'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2023, pp. 11036–11046 (cited on pages 7, 20).

[60] Nicolas Bonneel and Julie Digne. 'A survey of optimal transport for computer graphics and computer vision'. In: *Computers Graphics Forum* 42.2 (2023), pp. 439–460 (cited on page 8).

[61] Ofir Pele and Michael Werman. 'Fast and robust earth mover's distances'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE. 2009, pp. 460–467 (cited on page 8).

[62] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. 'Differentiable rendering using rgbxy derivatives and optimal transport'. In: *TOG* 41.6 (2022), pp. 1–13 (cited on page 8).

[63] Chao Zhang, Stephan Liwicki, and Roberto Cipolla. 'Beyond the CLS Token: Image Reranking using Pretrained Vision Transformers'. In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2022 (cited on page 8).

[64] Raia Hadsell, Sumit Chopra, and Yann LeCun. 'Dimensionality reduction by learning an invariant mapping'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE. 2006, pp. 1735–1742 (cited on pages 8, 19).

[65] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. 'Distance metric learning for large margin nearest neighbor classification'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 18. 2005 (cited on pages 8, 19).

[66] Kihyuk Sohn. 'Improved deep metric learning with multi-class n-pair loss objective'. In: *Neural Information Processing Systems (NeurIPS)*. 2016 (cited on page 8).

[67] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 'Deep metric learning with angular loss'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017, pp. 2593–2601 (cited on page 8).

[68] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. 'Signal-to-noise ratio: A robust distance metric for deep metric learning'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4815–4824 (cited on page 8).

[69] Jiankang Deng, J. Guo, and Stefanos Zafeiriou. 'ArcFace: Additive Angular Margin Loss for Deep Face Recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4685–4694 (cited on page 8).

[70] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. 'Deep metric learning to rank'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 1861–1870 (cited on page 8).

[71] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 'Circle Loss: A Unified Perspective of Pair Similarity Optimization'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 6397–6406 (cited on page 8).

[72] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. 'ElasticFace: Elastic Margin Loss for Deep Face Recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 1577–1586 (cited on page 8).

[73] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. 'Revisiting training strategies and generalization performance in deep metric learning'. In: *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR. 2020, pp. 8242–8252 (cited on page 8).

[74] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. 'Multi-similarity loss with general pair weighting for deep metric learning'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5022–5030 (cited on pages 8, 13, 19, 24).

[75] Amar Ali-Bey, Brahim Chaib-draa, and Philippe Giguère. 'BoQ: A place is worth a bag of learnable queries'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 17794–17803 (cited on page 8).

[76] NRK. *Nordlandsbanen: minute by minute, season by season*. 2013. URL: https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/ (cited on pages 8, 20, 21, 25).

[77] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 'CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 3–20 (cited on page 8).

[78] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 'Facenet: A unified embedding for face recognition and clustering'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823 (cited on page 8).

[79] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. 'Smart mining for deep metric learning'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017, pp. 2821–2829 (cited on page 8).

[80] Evgeny Smirnov, Aleksandr Melnikov, Sergey Novoselov, Eugene Luckyanets, and Galina Lavrentyeva. 'Doppelganger mining for face representation learning'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1916–1923 (cited on page 8).

[81] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. 'Stochastic class-based hard example mining for deep metric learning'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 7251–7259 (cited on page 8).

[82] Alexander Hermans, Lucas Beyer, and Bastian Leibe. 'In defense of the triplet loss for person re-identification'. In: *Arxiv preprint* (2017) (cited on page 8).

[83] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. 'Hard-Aware Deeply Cascaded Embedding'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017, pp. 814–823 (cited on page 8).

[84] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 'Hard negative examples are hard, but useful'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 126–142 (cited on page 8).

[85] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 'Hard negative mixing for contrastive learning'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 21798–21809 (cited on page 8).

[86] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, and Francesc Moreno-Noguer. 'Fracking deep convolutional image descriptors'. In: *Arxiv preprint* (2014) (cited on page 8).

[87] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. 'Learned contextual feature reweighting for image geo-localization'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2136–2145 (cited on page 8).

[88] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. 'Self-supervising fine-grained region similarities for large-scale image localization'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 369–386 (cited on page 8).

[89] Feng Lu, Lijun Zhang, Shuting Dong, Baifan Chen, and Chun Yuan. 'Aanet: Aggregation and alignment network with semi-hard positive sample mining for hierarchical place recognition'. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 11771–11778 (cited on page 8).

[90] Zetao Chen, Fabiola Maffra, Inkyu Sa, and Margarita Chli. 'Only look once, mining distinctive landmarks from convnet for visual place recognition'. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 9–16 (cited on page 9).

[91] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. 'Deep learning features at scale for visual place recognition'. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3223–3230 (cited on page 9).

[92] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. 'Learning context flexible attention model for long-term visual place recognition'. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4015–4022 (cited on pages 9, 13).

[93] Ahmad Khaliq, Shoaib Ehsan, Zetao Chen, Michael Milford, and Klaus McDonald-Maier. 'A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes'. In: *IEEE Transactions on Robotics (T-RO)* 36.2 (2019), pp. 561–569 (cited on page 9).

[94] Stephen Hausler, Adam Jacobson, and Michael Milford. 'Multi-process fusion: Visual place recognition using multiple image processing methods'. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1924–1931 (cited on page 9).

[95] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. 'Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam'. In: *IEEE Transactions on Robotics (T-RO)* 37.6 (2021), pp. 1874–1890 (cited on pages 9, 19, 35, 36, 44).

[96] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. 'From structure-from-motion point clouds to fast location recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2009, pp. 2599–2606 (cited on page 9).

[97] Noé Pion, Martin Humenberger, Gabriela Csurka, Yohann Cabon, and Torsten Sattler. 'Benchmarking image retrieval for visual localization'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 483–494 (cited on page 9).

[98] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2021 (cited on pages 9, 47, 48).

[99] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 'A survey on vision transformer'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 45.1 (2022), pp. 87–110 (cited on page 9).

[100] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. 'Mpvit: Multi-path vision transformer for dense prediction'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 7287–7296 (cited on page 9).

[101] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. 'Swin transformer v2: Scaling up capacity and resolution'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12009–12019 (cited on page 9).

[102] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 'On the opportunities and risks of foundation models'. In: *Arxiv preprint* (2021) (cited on page 10).

[103] Marco Cuturi. 'Sinkhorn distances: Lightspeed computation of optimal transport'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 26. 2013 (cited on page 12).

[104] Richard Sinkhorn and Paul Knopp. 'Concerning nonnegative matrices and doubly stochastic matrices'. In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348 (cited on page 12).

[105] Ilya Loshchilov and Frank Hutter. 'Decoupled weight decay regularization'. In: *Arxiv preprint* (2017) (cited on page 13).

[106] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. 'Visual place recognition with repetitive structures'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 883–890 (cited on pages 13, 25).

[107] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. 'EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2023, pp. 11080–11090 (cited on pages 13–15, 25).

[108] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 'A convnet for the 2020s'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11976–11986 (cited on page 14).

[109] Markus Käppeler, Kürsat Petek, Niclas Vödisch, Wolfram Burgard, and Abhinav Valada. 'Few-shot panoptic segmentation with foundation models'. In: *Arxiv preprint* (2023) (cited on page 15).

[110] Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. 'ViTMatte: Boosting image matting with pre-trained plain vision transformers'. In: *Information Fusion* 103 (2024), p. 102091. DOI: https://doi.org/10.1016/j.inffus.2023.102091 (cited on page 15).

[111] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 'Neighbourhood consensus networks'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018 (cited on page 17).

[112] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. 'Gluestick: Robust image matching by sticking points and lines together'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 9706–9716 (cited on page 17).

[113] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. 'Visual Localization using Imperfect 3D Models from the Internet'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 13175–13186 (cited on page 19).

[114] Emilio Garcia-Fidalgo and Alberto Ortiz. 'Hierarchical place recognition for topological mapping'. In: *IEEE Transactions on Robotics (T-RO)* 33.5 (2017), pp. 1061–1074 (cited on page 19).

[115] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. 'Scalable place recognition under appearance change for autonomous driving'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019, pp. 9319–9328 (cited on page 19).

[116] Feng Lu, Lijun Zhang, Xiangyuan Lan, Shuting Dong, Yaowei Wang, and Chun Yuan. 'Towards Seamless Adaptation of Pre-trained Models for Visual Place Recognition'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2024 (cited on pages 19, 25).

[117] Yanqing Shen, Sanping Zhou, Jingwen Fu, Ruotong Wang, Shitao Chen, and Nanning Zheng. 'StructVPR: Distill Structural Knowledge with Weighting Samples for Visual Place Recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 11217–11226 (cited on page 19).

[118] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. 'The worst-case time complexity for generating all maximal cliques and computational experiments'. In: *Theoretical computer science* 363.1 (2006), pp. 28–42 (cited on page 25).

[119] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 'Are we ready for autonomous driving? The KITTI vision benchmark suite'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 3354–3361 (cited on pages 31, 40, 51, 53).

[120] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. 'Full surround monodepth from multiple cameras'. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5397–5404 (cited on pages 31, 35).

[121] Max Argus, Lukas Hermann, Jon Long, and Thomas Brox. 'Flowcontrol: Optical flow based visual servoing'. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7534–7541 (cited on page 31).

[122] Georg Klein and David Murray. 'Parallel tracking and mapping for small AR workspaces'. In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2007, pp. 225–234 (cited on page 31).

[123] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 'MVSNet: Depth inference for unstructured multi-view stereo'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018 (cited on pages 31, 33, 46, 55).

[124] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. 'DeepMVS: Learning multi-view stereopsis'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cited on pages 31, 33, 48, 52).

[125] Chenjie Cao, Xinlin Ren, and Yanwei Fu. 'MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth'. In: *Transactions on Machine Learning Research* (2023) (cited on pages 31, 33, 46).

[126] Peter Sturm and Steve Maybank. 'A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images'. In: *Proceedings of the British Machine Vision Conference (BMVC)*. The British Machine Vision Association (BMVA). 1999, pp. 265–274 (cited on page 32).

[127] Derek Hoiem, Alexei A Efros, and Martial Hebert. 'Geometric context from a single image'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. Vol. 1. IEEE. 2005, pp. 654–661 (cited on page 32).

[128] Olga Barinova, Vadim Konushin, Anton Yakubenko, KeeChang Lee, Hwasup Lim, and Anton Konushin. 'Fast automatic single-view 3-d reconstruction of urban scenes'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2008, pp. 100–113 (cited on page 32).

[129] Jiyan Pan, Martial Hebert, and Takeo Kanade. 'Inferring 3D Layout of Building Facades From a Single Image'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2918–2926 (cited on page 32).

[130] Aamer Zaheer, Maheen Rashid, Muhammad Ahmed Riaz, and Sohaib Khan. 'Single-View Reconstruction using orthogonal line-pairs'. In: *Computer Vision and Image Understanding (CVIU)* 172 (2018), pp. 107–123 (cited on page 32).

[131] Ahmed Ali, Ali Hassan, Afsheen Rafaqat Ali, Hussam Ullah Khan, Wajahat Kazmi, and Aamer Zaheer. 'Real-time vehicle distance estimation using single view geometry'. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 1111–1120 (cited on page 32).

[132] Ashutosh Saxena, Min Sun, and Andrew Y Ng. 'Make3D: Learning 3D Scene Structure from a Single Still Image'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 31.5 (2008), pp. 824–840 (cited on page 32).

[133] Beyang Liu, Stephen Gould, and Daphne Koller. 'Single image depth estimation from predicted semantic labels'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2010, pp. 1253–1260 (cited on page 32).

[134] David Eigen and Rob Fergus. 'Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2015, pp. 2650–2658 (cited on pages 32, 35).

[135] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 'Deeper depth prediction with fully convolutional residual networks'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 239–248 (cited on pages 32, 35).

[136] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. 'Learning depth from single monocular images using deep convolutional neural fields'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 38.10 (2015), pp. 2024–2039 (cited on page 32).

[137] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. 'Structured attention guided convolutional neural fields for monocular depth estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3917–3925 (cited on page 32).

[138] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. 'Single-Image Depth Estimation Based on Fourier Domain Analysis'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 330–339 (cited on page 32).

[139] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. 'Structure-guided ranking loss for single image depth prediction'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 611–620 (cited on page 32).

[140] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. 'Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9685–9694 (cited on page 32).

[141] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. 'P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 1610–1621 (cited on page 32).

[142] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 'Deep ordinal regression network for monocular depth estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2002–2011 (cited on page 32).

[143] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. 'LocalBins: Improving Depth Estimation by Learning Local Distributions'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 480–496 (cited on page 32).

[144] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. 'Multi-View Depth Estimation by Fusing Single-View Depth Probability with Multi-View Geometry'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 2842–2851 (cited on pages 32, 36).

[145] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 'Single-Image Depth Perception in the Wild'. In: *Neural Information Processing Systems (NeurIPS)*. 2016 (cited on page 32).

[146] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 'Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer'. In: *PAMI* (2022) (cited on pages 32, 46).

[147] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 'Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 (cited on pages 32, 46).

[148] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 'Depth Anything V2'. In: *Neural Information Processing Systems (NeurIPS)*. 2024 (cited on pages 32, 46, 48, 51, 52, 54, 55).

[149] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. 'Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 (cited on pages 32, 46).

[150] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. 'LOTUS: Diffusion-based Visual Foundation Model for High-quality Dense Prediction'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2025 (cited on page 32).

[151] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. 'Metric3D: Towards zero-shot metric 3D prediction from a single image'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2023 (cited on pages 32, 46).

[152] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. 'ZoeDepth: Zero-shot transfer by combining relative and metric depth'. In: *Arxiv preprint* (2023) (cited on pages 32, 46).

[153] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. 'Depth Pro: Sharp Monocular Metric Depth in Less Than a Second'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2025 (cited on pages 32, 47, 51, 55, 57).

[154] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. 'MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision'. In: *Arxiv preprint* (2024) (cited on pages 32, 46).

[155] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. 'The temporal opportunist: Self-supervised multi-frame monocular depth'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 1164–1174 (cited on pages 32–34, 36, 39, 40, 42, 43, 46).

[156] Xiaofeng Wang, Zheng Zhu, Guan Huang, Fangbo Qin, Yun Ye, Yijia He, Xu Chi, and Xingang Wang. 'MVSTER: Epipolar transformer for efficient multi-view stereo'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022 (cited on page 32).

[157] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. 'DepthSplat: Connecting Gaussian Splatting and Depth'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 32).

[158] Luca Bartolomei, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. 'Stereo Anywhere: Robust Zero-Shot Deep Stereo Matching Even Where Either Stereo or Mono Fail'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 32).

[159] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. 'Unsupervised learning of depth and ego-motion from video'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1851–1858 (cited on page 33).

[160] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. 'Unsupervised monocular depth estimation with left-right consistency'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 270–279 (cited on page 33).

[161] Zhichao Yin and Jianping Shi. 'GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1983–1992 (cited on page 33).

[162] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. 'Unsupervised learning of geometry from videos with edge-aware depth-normal consistency'. In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. Vol. 32. 1. 2018 (cited on page 33).

[163] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. 'Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 340–349 (cited on page 33).

[164] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. 'Neural RGB-D Sensing: Depth and Uncertainty from a Video Camera'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10986–10995 (cited on page 33).

[165] Adrian Johnston and Gustavo Carneiro. 'Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4756–4765 (cited on page 33).

[166] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. 'Feature-metric loss for self-supervised learning of depth and egomotion'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 572–588 (cited on pages 33, 34, 36).

[167] Zhengming Zhou and Qiulei Dong. 'Self-distilled Feature Aggregation for Self-supervised Monocular Depth Estimation'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 709–726 (cited on page 33).

[168] Maria Klodt and Andrea Vedaldi. 'Supervising the new with the old: learning sfm from sfm'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 698–713 (cited on pages 33, 35, 42).

[169] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. 'Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 817–833 (cited on pages 33, 35).

[170] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. 'Digging into self-supervised monocular depth estimation'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019, pp. 3828–3838 (cited on pages 33, 35, 40, 43).

[171] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. 'Channel-Wise Attention-Based Network for Self-Supervised Monocular Depth Estimation'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 464–473 (cited on pages 33, 39, 42, 43).

[172] Hang Zhou, David Greenwood, and Sarah Taylor. 'Self-Supervised Monocular Depth Estimation with Internal Feature Fusion'. In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2021 (cited on pages 33, 39, 42, 43).

[173] Fangjinhua Wang, Qingtian Zhu, Di Chang, Quankai Gao, Junlin Han, Tong Zhang, Richard Hartley, and Marc Pollefeys. 'Learning-based Multi-View Stereo: A Survey'. In: *Arxiv preprint* (2024) (cited on page 33).

[174] Yasutaka Furukawa and Carlos Hernández. 'Multi-view stereo: A tutorial'. In: *Foundations and Trends in Computer Graphics and Vision* (2015) (cited on page 33).

[175] Johannes L Schonberger and Jan-Michael Frahm. 'Structure-from-motion revisited'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4104–4113 (cited on pages 33, 35, 36, 39, 55, 58).

[176] Jure Zbontar and Yann LeCun. 'Computing the stereo matching cost with a convolutional neural network'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cited on page 33).

[177] Zachary Teed and Jia Deng. 'DeepV2D: Video to Depth with Differentiable Structure from Motion'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2020 (cited on pages 33, 55).

[178] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. 'End-to-end learning of geometry and context for deep stereo regression'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017 (cited on pages 33, 48).

[179] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. 'DPSNet: End-to-end deep plane sweep stereo'. In: *Proceedings of the International Conference on Learning Representations (ICLR)* (2019) (cited on page 33).

[180] Kaixuan Wang and Shaojie Shen. 'MVDepthNet: Real-time multiview depth estimation neural network'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2018 (cited on pages 33, 48).

[181] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. 'Cascade cost volume for high-resolution multi-view stereo and stereo matching'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on pages 33, 34, 51).

[182] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. 'GA-Net: Guided Aggregation Net for End-to-end Stereo Matching'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cited on page 33).

[183] Xinjing Cheng, Peng Wang, and Ruigang Yang. 'Learning Depth with Convolutional Spatial Propagation Network'. In: *PAMI* (2019) (cited on page 33).

[184] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. 'Transmvsnet: Global context-aware multi-view stereo network with transformers'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cited on page 33).

[185] Chenjie Cao, Xinlin Ren, and Yanwei Fu. 'MVSFormer++: Revealing the Devil in Transformer's Details for Multi-View Stereo'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2024 (cited on pages 33, 48, 55).

[186] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. 'Visibility-aware Multi-view Stereo Network'. In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2020 (cited on pages 33, 55).

[187] Xiaoxiao Long, Lingjie Liu, Christian Theobalt, and Wenping Wang. 'Occlusion-Aware Depth Estimation with Adaptive Normal Constraints'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020 (cited on page 33).

[188] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. 'MonoRec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 33).

[189] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. 'DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on pages 33, 49, 52, 53, 56, 58).

[190] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. 'SimpleRecon: 3D Reconstruction Without 3D Convolutions'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022 (cited on pages 33, 46–48, 50, 52, 54, 56, 58).

[191] Zehao Yu and Shenghua Gao. 'Fast-MVSNet: Sparse-to-Dense Multi-View Stereo With Learned Propagation and Gauss-Newton Refinement'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on pages 33, 55).

[192] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. 'DUSt3R: Geometric 3D Vision Made Easy'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 (cited on pages 33, 52).

[193] Vincent Leroy, Yohann Cabon, and Jerome Revaud. 'Grounding Image Matching in 3D with MASt3R'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2024 (cited on pages 33, 47, 52, 55, 57, 58).

[194] Mohamed Sayed, Filippo Aleotti, Jamie Watson, Zawar Qureshi, Guillermo Garcia-Hernando, Gabriel Brostow, Sara Vicente, and Michael Firman. 'DoubleTake: Geometry Guided Depth Estimation'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2024 (cited on page 33).

[195] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. 'Domain-invariant Stereo Matching Networks'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020 (cited on page 33).

[196] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 'A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cited on page 33).

[197] Jiawei Zhang, Jiahe Li, Lei Huang, Xiaohan Yu, Lin Gu, Jin Zheng, and Xiao Bai. 'Robust Synthetic-to-Real Transfer for Stereo Matching'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 (cited on page 33).

[198] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. 'DepthCrafter: Generating Consistent Long Depth Sequences for Open-world Videos'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on pages 33, 56).

[199] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. 'Learning unsupervised multi-view stereopsis via robust photometric consistency'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019 (cited on page 33).

[200] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. 'Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2019 (cited on page 33).

[201] Jiayu Yang, Jose M Alvarez, and Miaomiao Liu. 'Self-supervised learning of depth inference for multi-view stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 33).

[202] Xianda Guo, Chenming Zhang, Youmin Zhang, Dujun Nie, Ruilin Wang, Wenzhao Zheng, Matteo Poggi, and Long Chen. 'Stereo Anything: Unifying Stereo Matching with Large-Scale Mixed Data'. In: *Arxiv preprint* (2024) (cited on page 33).

[203] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. 'FoundationStereo: Zero-Shot Stereo Matching'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 33).

[204] Zhenxing Mi, Chang Di, and Dan Xu. 'Generalized binary search network for highly-efficient multi-view stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cited on pages 34, 51).

[205] Song Zhang, Wenjia Xu, Zhiwei Wei, Lili Zhang, Yang Wang, and Junyi Liu. 'ARAI-MVSNet: A multi-view stereo depth estimation network with adaptive depth range and depth interval'. In: *Pattern Recognition* (2023) (cited on pages 34, 51).

[206] Andrea Conti, Matteo Poggi, Valerio Cambareri, and Stefano Mattoccia. 'Range-Agnostic Multi-View Depth Estimation With Keyframe Selection'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2024 (cited on page 34).

[207] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. 'Deep stereo using adaptive thin volume representation with uncertainty awareness'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on page 34).

[208] Jingliang Li, Zhengda Lu, Yiqun Wang, Jun Xiao, and Ying Wang. 'NR-MVSNet: Learning multi-view stereo based on normal consistency and depth refinement'. In: *IEEE Transactions on Image Processing* (2023) (cited on page 34).

[209] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. 'Bundle adjustment—a modern synthesis'. In: *Proceedings of the International Workshop on Vision Algorithms, in association with ICCV*. Springer. 1999, pp. 298–372 (cited on page 34).

[210] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. 'Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019, pp. 7063–7072 (cited on pages 34, 36).

[211] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. 'Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos'. In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. Vol. 33. 01. 2019, pp. 8001–8008 (cited on page 34).

[212] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. 'Monocular depth estimation with self-supervised instance adaptation'. In: *Arxiv preprint* (2020) (cited on pages 34, 36, 39, 42).

[213] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. 'Comoda: Continuous monocular depth adaptation using past experiences'. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 2907–2917 (cited on page 34).

[214] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. 'Pseudo RGB-D for Self-Improving Monocular SLAM and Depth Prediction'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 437–455 (cited on pages 34, 36, 41, 43).

[215] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. 'Consistent video depth estimation'. In: *TOG* 39.4 (2020), pp. 71–1 (cited on pages 34–36, 38, 39, 41–43, 46, 56).

[216] Pengli Zhu, Siyuan Liu, Tao Jiang, Yancheng Liu, Xuzhou Zhuang, and Zhenrui Zhang. 'Autonomous Reinforcement Control of Visual Underwater Vehicles: Real-Time Experiments Using Computer Vision'. In: *IEEE Transactions on Vehicular Technology* 71.8 (2022), pp. 8237–8250 (cited on page 35).

[217] David Recasens, José Lamarca, José M Fácil, JMM Montiel, and Javier Civera. 'Endo-Depth-and-Motion: Reconstruction and Tracking in Endoscopic Videos using Depth Networks and Photometric Constraints'. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7225–7232 (cited on page 35).

[218] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. 'Neural Window Fully-Connected CRFs for Monocular Depth Estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 3916–3925 (cited on page 35).

[219] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. 'CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11826–11835 (cited on page 35).

[220] Manuel López Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. 'Mapillary planet-scale depth dataset'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 589–604 (cited on page 35).

[221] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. 'Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2019, pp. 8977–8986 (cited on page 35).

[222] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004 (cited on pages 35, 38).

[223] Davide Scaramuzza and Friedrich Fraundorfer. 'Visual odometry [tutorial]'. In: *IEEE Robotics and Automation Magazine* 18.4 (2011), pp. 80–92 (cited on page 35).

[224] Jakob Engel, Vladlen Koltun, and Daniel Cremers. 'Direct sparse odometry'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 40.3 (2017), pp. 611–625 (cited on page 35).

[225] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. 'CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6243–6252 (cited on page 35).

[226] Zhengqi Li and Noah Snavely. 'MegaDepth: Learning Single-View Depth Prediction from Internet Photos'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2041–2050 (cited on pages 35, 53).

[227] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. 'OpenMVG: Open Multiple View Geometry'. In: *International Workshop on Reproducible Research in Pattern Recognition*. Springer. 2016, pp. 60–74 (cited on page 36).

[228] *OpenSfM*. https://github.com/mapillary/OpenSfM (cited on page 36).

[229] Martin A Fischler and Robert C Bolles. 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography'. In: *ACM Transactions on Graphics (TOG)* 24.6 (1981), pp. 381–395 (cited on page 38).

[230] P Kingma Diederik. 'Adam: A method for stochastic optimization'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015 (cited on page 39).

[231] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. 'Sparsity Invariant CNNs'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. IEEE. 2017, pp. 11–20 (cited on page 40).

[232] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. 'Structure and Motion from Casual Videos'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 20–37 (cited on page 44).

[233] Zhenpei Yang, Zhile Ren, Qi Shan, and Qixing Huang. 'MVS2D: Efficient multi-view stereo via attention-driven 2D convolutions'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cited on pages 46, 55).

[234] Vitor Guizilini, Pavel Tokmakov, Achal Dave, and Rares Ambrus. 'GRIN: Zero-Shot Metric Depth with Pixel-Level Diffusion'. In: *Arxiv preprint* (2024) (cited on page 46).

[235] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. 'UniDepth: Universal Monocular Metric Depth Estimation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024 (cited on pages 46, 55).

[236] Duolikun Danier, Mehmet Aygün, Changjian Li, Hakan Bilen, and Oisin Mac Aodha. 'DepthCues: Evaluating Monocular Depth Perception in Large Vision Models'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 46).

[237] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. 'Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cited on page 46).

[238] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. 'Don't forget the past: Recurrent depth estimation from monocular video'. In: *RAL* (2020) (cited on page 46).

[239] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Matteo Poggi, and Yiyi Liao. 'Learning Temporally Consistent Video Depth from Video Diffusion Priors'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 46).

[240] Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. 'Depth Any Video with Scalable Synthetic Data'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2025 (cited on pages 46, 56).

[241] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. 'On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cited on page 46).

[242] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. 'PatchmatchNet: Learned Multi-View Patchmatch Stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on pages 47, 55).

[243] Philipp Schröppel, Jan Bechtold, Artemij Amiranashvili, and Thomas Brox. 'A Benchmark and a Baseline for Robust Multi-view Depth Estimation'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2022 (cited on pages 47, 52, 53, 55–57).

[244] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cited on page 48).

[245] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 'Vision transformers for dense prediction'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2021 (cited on pages 48, 51).

[246] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. 'Large scale multi-view stereopsis evaluation'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cited on pages 51, 53).

[247] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 'ScanNet: Richly-annotated 3D reconstructions of indoor scenes'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cited on pages 51, 53).

[248] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. 'A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cited on pages 51, 53).

[249] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 'Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction'. In: *TOG* (2017) (cited on pages 51, 53).

[250] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. 'Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2021 (cited on page 52).

[251] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. 'TartanAir: A Dataset to Push the Limits of Visual SLAM'. In: *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. 2020 (cited on page 52).

[252] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. 'BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on page 52).

[253] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. 'MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond'. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2023 (cited on page 52).

[254] Yohann Cabon, Naila Murray, and Martin Humenberger. 'Virtual KITTI 2'. In: *Arxiv preprint* (2020) (cited on page 52).

[255] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. 'Virtual worlds as proxy for multi-object tracking analysis'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cited on page 52).

[256] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. 'DynamicStereo: Consistent Dynamic Depth from Stereo Videos'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2023) (cited on page 52).

[257] Y.-T. Hu, H.-S. Chen, K. Hui, J.-B. Huang, and A. G. Schwing. 'SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation – A Synthetic Dataset and Baselines'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cited on page 52).

[258] Y.-T. Hu, J. Wang, R. A. Yeh, and A. G. Schwing. 'SAIL-VOS 3D: A Synthetic Dataset and Baselines for Object Detection and 3D Mesh Reconstruction from Video Data'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 52).

[259] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. '3D photography using context-aware layered depth inpainting'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on page 53).

[260] Juyang Weng, Paul Cohen, Marc Herniou, et al. 'Camera calibration with distortion models and accuracy evaluation'. In: *PAMI* (1992) (cited on page 54).

[261] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 'Pixelwise View Selection for Unstructured Multi-View Stereo'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016 (cited on page 55).

[262] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. 'DeMoN: Depth and Motion Network for Learning Monocular Stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cited on page 55).

[263] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. 'Cost Volume Pyramid Based Depth Inference for Multi-View Stereo'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cited on page 55).

[264] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. 'DeepTAM: Deep Tracking and Mapping'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018 (cited on page 55).

[265] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. 'TransformerFusion: Monocular RGB scene reconstruction using transformers'. In: *Neural Information Processing Systems (NeurIPS)*. 2021 (cited on pages 56, 58).

[266] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. 'Atlas: End-to-end 3D scene reconstruction from posed images'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020 (cited on page 58).

[267] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. 'NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 (cited on page 58).

[268] Alexander Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. '3DVNet: Multi-View Depth Prediction and Volumetric Refinement'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2021 (cited on page 58).

[269] Noah Stier, Alexander Rich, Pradeep Sen, and Tobias Höllerer. 'VoRTX: Volumetric 3D reconstruction with transformers for voxelwise view selection and fusion'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2021 (cited on page 58).

[270] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. 'Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction'. In: *Neural Information Processing Systems (NeurIPS)*. Vol. 37. 2024, pp. 139725–139750 (cited on page 56).

[271] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. 'Dn-splatter: Depth and normal priors for gaussian splatting and meshing'. In: *Arxiv preprint* (2024) (cited on page 56).

[272] Riku Murai, Eric Dexheimer, and Andrew J. Davison. 'MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025 (cited on page 56).

[273] Hengyi Wang and Lourdes Agapito. '3D Reconstruction with Spatial Memory'. In: *Proceedings of the International Conference on 3D Vision (3DV)*. 2025 (cited on page 56).

# List of Acronyms

**A**
**AR** Augmented Reality. 1, 5, 9, 35

**C**
**CM** CliqueMining. 25
**CNN** Convolutional Neural Network. 47

**D**
**DAV2** Depth Anything V2. 52

**G**
**GDS** Geographic Distance Sensitivity. 19–26, 28, 29, 59, 61
**GeM** Generalized Mean Pooling. 7, 9, 13, 25
**GT** Ground Truth. 53, 56, 57

**L**
**LiDAR** Light Detection and Ranging. 35, 40

**M**
**MLP** Multi-Layer Perceptron. 7, 50
**MS** Multi-Similarity. 22, 24, 27
**MVS** Multi-View Stereo. 33, 46–48, 50–52, 54, 56, 58

**R**
**ReLU** Rectified Linear Unit. 13
**RMSE** Root Mean Squared Error. 40, 42, 43, 45
**RMVDB** Robust Multi-View Depth Benchmark. 53, 55, 56

**S**
**SfM** Structure-from-Motion. 2, 31, 33–36, 40, 44, 45, 47, 54
**SLAM** Simultaneous Localization And Mapping. 1, 5, 9, 19, 34–36
**Spatial AI** Spatial Artificial Intelligence. 1, 5, 59

**T**
**TTR** Test-Time Refinement. 2, 31, 32, 34–36, 39–45

**U**
**UTM** Universal Transverse Mercator. 23

**V**
**ViT** Vision Transformer. 7, 9, 10, 31, 47, 48, 50, 51, 54, 58, 60, 62
**VLAD** Vector of Locally Aggregated Descriptors. 7, 9, 12, 15
**VPR** Visual Place Recognition. 1, 5–11, 13–15, 18, 19, 21, 24–26, 28, 29, 59, 61