3D Reconstruction with Spatial Memory

Hengyi Wang   Lourdes Agapito
Department of Computer Science, University College London
{hengyi.wang.21, l.agapito}@ucl.ac.uk

Figure 1. **Overview.** Given a set of ordered or unordered image collections without prior knowledge of camera parameters, the proposed Spann3R can incrementally reconstruct the 3D geometry by directly regressing the pointmap of each image in a common coordinate system. Spann3R does not require any optimization-based alignment during inference, i.e., the 3D reconstruction of each image can be solved by a simple forward pass with a transformer-based architecture, thus enabling online reconstruction in real-time. The qualitative examples shown are reconstructed from some self-captured images to illustrate the generalization ability of Spann3R.

Abstract

We present Spann3R, a novel approach for dense 3D reconstruction from ordered or unordered image collections. Built on the DUSt3R paradigm, Spann3R uses a transformer-based architecture to directly regress pointmaps from images without any prior knowledge of the scene or camera parameters. Unlike DUSt3R, which predicts per image-pair pointmaps expressed in a local coordinate frame, Spann3R predicts per-image pointmaps expressed in a global coordinate system, thus eliminating the need for optimization-based global alignment. The key idea behind Spann3R is to manage an external spatial memory that learns to keep track of all previous relevant 3D information. Spann3R then queries this spatial memory to predict the 3D structure of the next frame in a global coordinate system. Taking advantage of DUSt3R's pre-trained weights, and further fine-tuning on a subset of datasets, Spann3R shows competitive performance and generalization ability on various unseen datasets and can process ordered image collections in real-time. Project page: https://hengyiwang.github.io/projects/spanner

1. Introduction

Reconstructing dense geometry from images is one of the fundamental problems in computer vision that has been researched for decades [31]. This task offers numerous applications in autonomous driving, virtual reality, robotics, medical imaging, and more. The inherent ambiguities in interpreting 3D structures have led traditional solutions evolve into various sub-fields, including keypoint detection and matching [10, 46, 47, 60], Structure-from-Motion (SfM) [2, 19, 64, 68, 75, 84, 85], Bundle Adjustment (BA) [3, 77, 86], Multi-View Stereo (MVS) [29, 30, 65], Simultaneous Localization and Mapping (SLAM) [21, 41, 54], etc. Each of these sub-fields addresses different aspects of the problem using a variety of handcrafted heuristics, requiring substantial engineering effort to integrate them into a complete dense reconstruction pipeline [64, 65].

Recent attention has shifted towards replacing handcrafted features with learned structural priors from large-scale datasets [12, 22, 32, 61, 73, 79, 89, 94]. These modern approaches typically integrate learning-based models into each step of the traditional pipeline. Thus, the sequential structure of traditional pipelines, involving matching, trian-

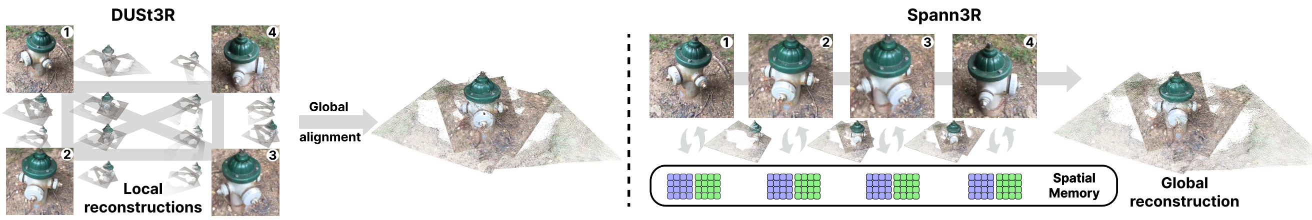

Figure 2. **Motivation.** DUSt3R [81] directly regresses the pointmap of each image pair in a local coordinate system. In contrast, Spann3R predicts a global pointmap in a common coordinate system via a spatial memory that stores all previous predictions. Thus, our method can enable online incremental reconstruction without the need to build a dense pairwise graph and a final optimization-based alignment.

gulation, sparse reconstruction, camera parameter estimation, and dense reconstruction is mostly maintained. While these methods have made significant progress with learned priors, the inherent limitations of this complex pipeline persist, making it sensitive to noise at each step and still demanding substantial engineering effort for integration.

To address these issues, DUSt3R [81] introduces a radical and novel paradigm shift that was often considered impossible - directly regressing the pointmap, a common representation in visual localization [13–15, 59], from a pair of images without prior scene information. Since the pointmap is expressed in the local coordinate system of the image pair, a global alignment is introduced for the reconstruction of more than just an image pair. This involves per-scene optimization to align the predicted pointmap with a dense pairwise graph. Trained on millions of image pairs with ground-truth annotations for depth and camera parameters, DUSt3R [81] shows unprecedented performance and generalization across various real-world scenarios with different camera sensors. However, operating on a pair of images and the need for per-scene optimization-based global alignment limit its ability for real-time incremental reconstruction and scalability to many images.

In this paper, we present Spann3R, a framework that adopts a Spatial Memory for 3D Reconstruction. Building on the paradigm of DUSt3R [81], we take a step further by eliminating the need for per-scene optimization-based alignment (See Fig. 2). That being said, our model enables incremental reconstruction by predicting the pointmap of each image in a common coordinate system with a simple forward pass on our transformer-based architecture. The key idea is to maintain an external memory that keeps track of previous states and learns to query all relevant information from this memory for predicting the next frame, a concept often referred to as memory networks [51, 72, 83].

We employ a lightweight transformer-based memory encoder to encode previous predictions as memory values. To retrieve information from this memory, we project geometric features from two decoders into query features and memory keys using two multilayer perceptron (MLP) heads. Our model is trained on sequences of five frames randomly sampled from videos, with a curriculum training strategy that adjusts the sample window size throughout the training process. This allows Spann3R to learn both short and long-term dependency across frames. During inference, we apply a memory management strategy inspired by X-Mem [16], which mimics human memory model [5], to maintain a compact memory representation. Compared to DUSt3R [81], our method aligns point on-the-fly (like a spanner) purely based on neural network (NN), enables real-time online incremental reconstruction at over 50 frames per second (fps) without test-time optimization. Experiments on various unseen datasets show competitive dense reconstruction quality and generalization ability.

2. Related Works

Classic 3D Reconstruction. 3D Reconstruction from visual signals has been investigated for decades [31]. Structure-from-motion (SfM) [2, 19, 56, 64, 68, 75, 84, 85] is often considered the de-facto standard for obtaining sparse geometry and accurate camera poses. Starting from feature correspondence search (keypoint detection and description [10, 46, 47, 60], matching [2, 85], and geometric verification [64]), SfM selects an image pair for initialization, followed by image registration, triangulation, and bundle adjustment [3, 77, 86]. Finally, multi-view stereo [29, 30, 65] is used to obtain dense 3D geometry. These methods usually require lengthy offline optimization. In contrast, visual SLAM focuses on online reconstruction in real-time. Given the calibrated cameras, visual SLAM can perform sparse [21, 28, 41, 52] or dense [27, 54] reconstruction via minimizing either reprojection error (indirect) [21, 41, 52] or photometric error (direct) [27, 28, 54]. To obtain accurate reconstruction, these methods either require a depth/LiDAR sensor [53] or careful initialization and various assumptions about the camera motion and scene appearance [21, 52, 54].

Learning-based 3D Reconstruction. Built upon the success of the classic reconstruction pipeline, recent approaches usually leverage learning-based techniques to improve each sub-task, i.e., feature extraction [22, 93], matching [45, 61], BA [44], monocular depth estimation [23, 38, 94], multi-view depth estimation [26, 63, 89], optical

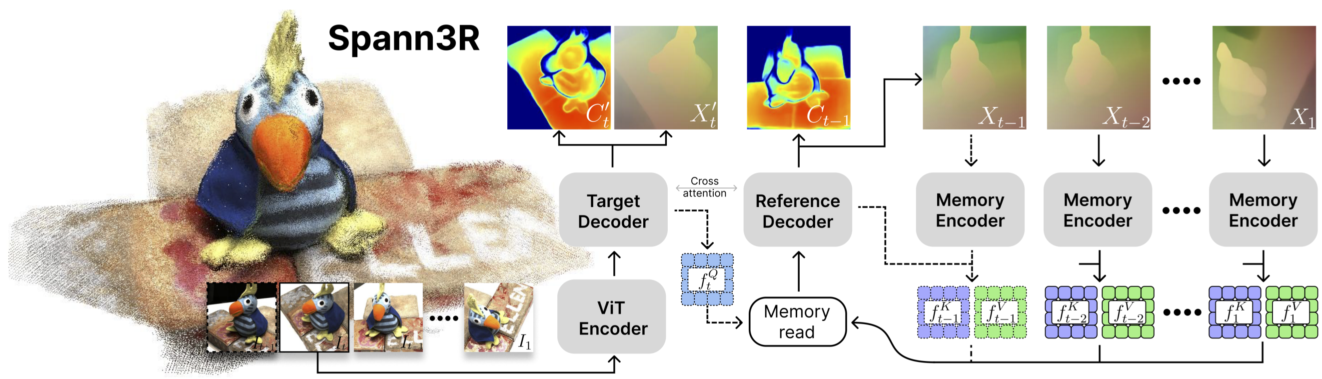

Figure 3. **Overview of Spann3R.** Our model uses a ViT [25] encoder and two intertwined decoders as in DUSt3R [81]. Our target decoder outputs query features for memory read while the reference decoder predicts a pointmap and confidence based on the memory readout using query and memory features. A lightweight memory encoder receives previously predicted pointmaps together with features from reference decoder as input and encodes them into memory key and value features. Dashed lines indicate an operation in the next time step.

flow [76], point tracking [24, 37, 87], etc. However, classic pipelines usually involve a sequential structure vulnerable to noise in each sub-task. To avoid this, DUSt3R [81] unifies all sub-tasks by directly learning to map an image pair to 3D, followed by an optimization-based global alignment to bring all image pairs into a common coordinate system. In this work, we take a step further to replace the optimization step with an end-to-end learning-based framework, enabling real-time online incremental reconstruction in a feed-forward way.

Neural Rendering for 3D Reconstruction. Recent progress in differentiable rendering, i.e., Neural Radiance Field (NeRF) [50] and its follow-up works [7, 8, 33, 35, 36, 40, 80, 91] has enabled high-fidelity scene reconstruction using images with known camera parameters obtained via SfM [64]. Several other works leverage neural rendering for SfM [11] and SLAM [42, 71, 78, 95]. However, despite the significant progress in accelerating neural rendering, these methods still require lengthy optimization time. For instance, Gaussian splatting [40] and its variants in SLAM [34, 39, 48] can achieve over 100fps rendering. However, they still require minutes of test-time optimization for scene reconstruction.

Memory Networks. Memory networks were originally introduced in the context of question-answering [51, 72, 83] in natural language processing, where an external memory is used to reason over long-term dependencies. This architecture is naturally suitable for processing sequential data and can thus be adapted to various vision tasks, such as video object segmentation (VOS) [16–18, 55], video understanding [69], etc. Our work is inspired by STM [55], the first method to employ memory networks for VOS, and XMem [16], which further extended the idea to long video sequences via a memory consolidation strategy that mimics the human memory model [5].

3. Method

Fig. 3 shows an overview of Spann3R. We repurpose DUSt3R [81] into an end-to-end incremental reconstruction framework that regresses pointmaps in a global coordinate frame. Specifically, given a sequence of images $\{I_t\}_{t=1}^{N}$, our goal is to train a network $\mathcal{F}$ that maps each I_t to its corresponding pointmap X_t, expressed in the coordinate frame of I_1. To enable this, we introduce a spatial memory that encodes previous predictions to reason about the next frame. Sec. 3.1 describes our network architecture , Sec. 3.2 the spatial memory, and Sec. 3.3 training and inference.

3.1. Network Architecture

Feature encoding. In each forward pass, our model takes a frame I_t and a previous query f_{t-1}^{Q} as input. A ViT [25] is used to encode the frame I_t into visual feature f_t^I:

$$f_t^I = \mathrm{Encoder}^I(I_t). \tag{1}$$

The query features f_{t-1}^{Q} are used to retrieve features in our memory bank to output the fused feature f_{t-1}^{G}:

$$f_{t-1}^{G} = \mathrm{Memory\_read}(f_{t-1}^{Q}, f^K, f^V), \tag{2}$$

where f^K and f^V are memory key and value features.

Feature decoding. The fused feature f_{t-1}^{G} and the visual feature f_t^I are fed into two intertwined decoders that process them jointly via cross-attention. This enables the model to reason about the spatial relationship between two features:

$$f_t^{H'}, f_{t-1}^{H} = \mathrm{Decoder}(f_t^I, f_{t-1}^{G}). \tag{3}$$

where H'/H refers to features decoded by the target/reference decoders. The feature $f_t^{H'}$ decoded by the target decoder is fed into an MLP head to generate the query feature for the next step:

$$f_t^{Q} = \mathrm{head}_{\mathrm{query}}^{\mathrm{target}}(f_t^{H'}, f_t^I). \tag{4}$$

The feature f_{t-1}^H decoded by the reference decoder is fed into an MLP head to generate the pointmap and confidence:

$$X_{t-1}, C_{t-1} = \text{head}_{\text{out}}^{\text{ref}}(f_{t-1}^H). \tag{5}$$

Note that we also generate a pointmap X_t' and confidence C_t' from $f_t^{H'}$ only for supervision.

Memory encoding. The feature and predicted pointmap of the reference decoder are used for encoding the memory key and value features:

$$f_{t-1}^K = \text{head}_{\text{key}}^{\text{ref}}(f_{t-1}^H, f_{t-1}^I), \tag{6}$$
$$f_{t-1}^V = \text{Encoder}^V(X_{t-1}) + f_{t-1}^K. \tag{7}$$

Since memory key and value features have both geometric and visual information, it enables memory readout based on both appearance and distance.

Initialization. We decode two visual features for initialization, which makes the initialization identical to DUSt3R.

Discussion. Compared to DUSt3R [81], Spann3R has one more lightweight memory encoder and two MLP heads for encoding the query, memory key and value features. For decoders, DUSt3R [81] contains two decoders - a reference decoder that reconstructs the first image in the canonical coordinate system, and a target decoder that reconstructs the second image in the coordinate system of the first image. In contrast, we repurpose two decoders in DUSt3R [81]. The target decoder is mainly used to produce features for querying the memory while the reference decoder takes the fused features from memory for reconstruction.

3.2. Spatial Memory

Fig. 4 shows an overview of the spatial memory that consists of a dense working memory, a sparse long-term memory, and a memory query mechanism for extracting features from the memory, which we will describe next.

Memory query. The spatial memory stores all key and value feature $f^K, f^V \in \mathbb{R}^{Bs \times (T \cdot P) \times C}$, where $Bs, (T \cdot P), C$ are batch size, number of tokens and channels. To compute fused feature f_{t-1}^G, we apply a cross attention using query feature $f_{t-1}^Q \in \mathbb{R}^{Bs \times P \times C}$ for memory reading:

$$f_{t-1}^G = A_{t-1} f^V + f_{t-1}^Q, \tag{8}$$

where $A_{t-1} \in \mathbb{R}^{Bs \times P \times (T \cdot P)}$ is the attention map:

$$A_{t-1} = \text{Softmax}(\frac{f_{t-1}^Q (f^K)^\top}{\sqrt{C}}). \tag{9}$$

This attention map contains a dense attention weight for each token in the current query with respect to all tokens in memory keys (See Fig. 8). During training, We apply an attention dropout of 0.15 to encourage the model to reason the geometry from a subset of the memory values. In practice, we observe that at inference, most of the attention weights are relatively small, as illustrated in the cumulative histogram of Fig. 4. However, despite their small weights, the corresponding patches can be significantly distant from the query patches or even outliers. In the end, their memory values might still have a non-negligible impact on the fused features. To mitigate the impact of these outlier features, we apply a hard clipping threshold of 5×10^{-4} and re-normalize the attention weights at inference time.

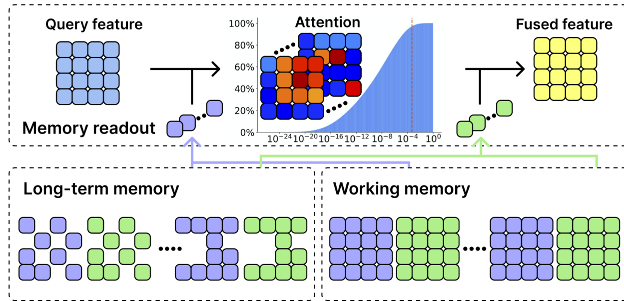

Figure 4. **Overview of our spatial memory.** Our memory contains a dense working memory chunk and a sparse long-term memory chunk. For each memory query, all tokens in both long-term memory and working memory will be used for generating attention weight and the fused feature. We also visualize the cumulative histogram of the values in attention weight.

Eq. 8 contains a weighted average of value features along $(T \cdot P)$. Thus, the memory query mechanism, by design, can handle features with different $(T \cdot P)$. This enables us to prune key and value features along this dimension at inference. To this end, we divide memory into working memory and long-term memory.

Working memory. This consists of dense memory features from the recent 5 frames. For each incoming memory feature, we first correlate its key feature with each key feature in working memory. We only insert new key and value features into working memory if their maximum similarity is less than 0.95. Once the working memory is full, the oldest memory features are drained into long-term memory.

Long-term memory. At inference, long-term memory accumulates over time, which increases GPU memory usage and slows down speed. Inspired by XMem [16], we design a similar strategy to sparsify long-term memory. Specifically, for each token in long-term memory keys, we keep track of its accumulated attention weights (i.e., A in Eq. 9). Once the long-term memory reaches a predefined threshold, we perform memory sparsification by retaining only the top-k memory keys and their corresponding value features.

3.3. Training and Inference

Objective function. Following DUSt3R [81], we train our model by a simple confidence-aware regression loss. We additionally include a scale loss to encourage the average

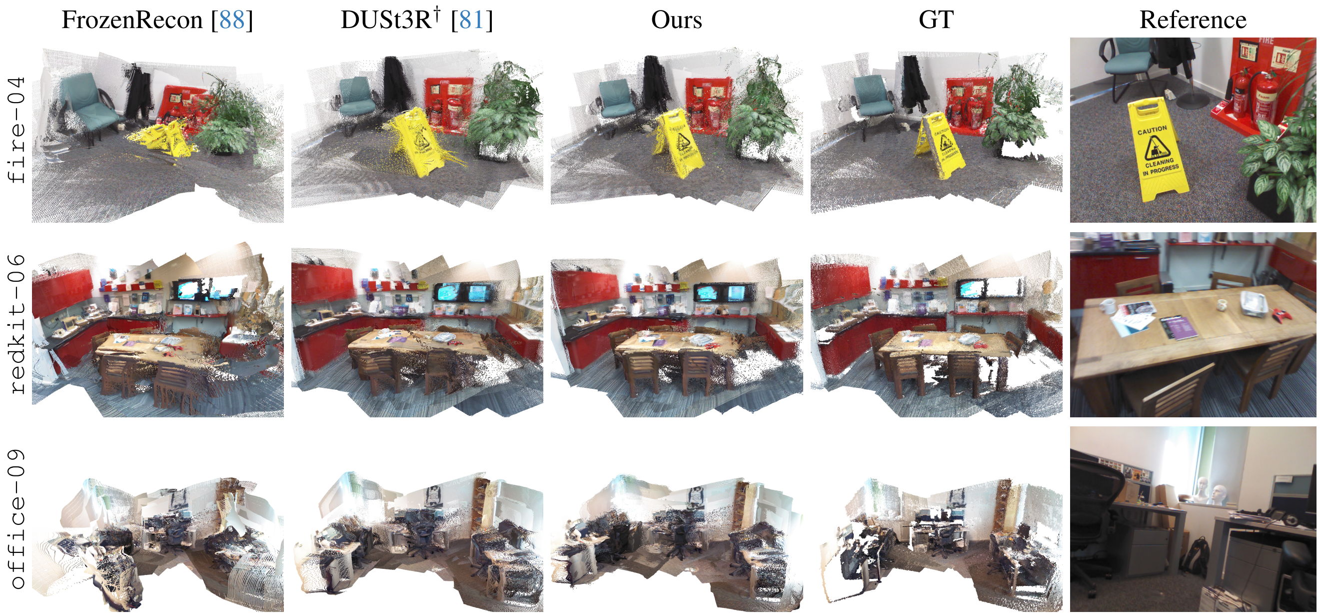

Figure 5. **Qualitative examples.** We show qualitative examples of DUSt3R† [81], FrozenRecon [88] for a comprehensive comparison. Our method shows competitive results in comparison to other offline methods. However, since our method runs online without any optimization-based alignment, it can potentially lead to drift issues in some challenging scenarios (See Office-09).

distance of the predicted point cloud to become smaller than the ground truth. The overall loss is $\mathcal{L} = \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{scale}}$.

Note that to compute $\mathcal{L}_{\text{conf}}$, both the predicted and ground truth pointmaps are normalized by their average distance. We tried to fix this scale based on the first two-view prediction during initial experiments, but it does not work well due to the presence of outliers and the unbounded nature of the outdoor scene, Co3D [58], for instance.

Curriculum training. Due to GPU memory constraints, we train our model by randomly sampling 5 frames per video sequence. Thus, the memory bank contains only a 4-frame memory at maximum during training. To ensure the model adapts to diverse camera motions and long-term feature matching, we gradually increase the sample window size throughout the training. For the last 25% epochs, we gradually decrease the window size to ensure the training frame interval aligns with the inference frame interval.

Inference. Our model naturally fits sequential data, i.e. video sequence. For unordered image collections, we can build a dense pairwise graph as in DUSt3R [81]. The pair with the highest confidence will be used for initialization. Then, we can either build a minimum spanning tree based on pairwise confidence or directly feed the remaining images into our model to identify the next best image based on the predicted confidence. Note that the confidence map in DUSt3R [81] involves an exponential function, which tends to overweight patches with higher confidence. In our case, we find that map it back to a sigmoid function for view selection can improve the robustness of the reconstruction.

4. Experiments

4.1. Setup

Datasets. DUSt3R [81] adopts a mixture of 8 datasets: Habitat [62], MegaDepth [43], ARKitScenes [9], Static Scenes 3D [49], BlendedMVS [90], ScanNet++ [92], Co3D-v2 [58], and Waymo [74]. We choose a subset of datasets: Habitat [62], ScanNet [20], ScanNet++ [92], ARKitScenes [9], BlendedMVS [90], Co3D-v2 [58] for training our model. Note that for Habitat [62], we only use a small subset of the scenes to synthesize data for training. To demonstrate the generalization ability of our model, we quantitatively evaluate our model on 3 unseen datasets: 7Scenes [67], NRGBD [6], and DTU [1].

Baselines. We consider DUSt3R [81] and FrozenRecon [88] as our baselines. FrozenRecon is a test-time optimization method that jointly optimizes camera parameters with the scale and shift factor of the depth map from the off-the-shelf monocular depth estimation model. All evaluations are performed on a single NVIDIA 4090 GPU with 24GB of memory. DUSt3R† denotes running DUSt3R's final weight with 224×224 images as running full reconstruction on 512×384 images cannot fit in 24GB memory. We include both results of DUSt3R on few-view reconstruction. Ours* denotes offline reconstruction using predicted confidence for next-view selection.

Metrics. We use *accuracy*, *completion* and *normal consistency* as in prior works [6, 78, 95]. The predicted dense

Method	Optim.	Onl.	7 scenes						NRGBD						FPS
			Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑		
			Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	
F-Recon [88]	✓		0.1243	0.0762	0.0554	0.0231	0.6189	0.6885	0.2855	0.2059	0.1505	0.0631	0.6547	0.7577	<0.1
Dust3R† [81]	✓		**0.0286**	**0.0123**	0.0280	0.0091	**0.6681**	**0.7683**	**0.0544**	**0.0251**	0.0315	**0.0103**	**0.8024**	**0.9529**	0.78
Ours		✓	0.0342	0.0148	**0.0241**	**0.0085**	0.6635	0.7625	0.0691	0.0315	**0.0291**	0.0110	0.7775	0.9371	**65.49**
Dust3R [81] (FV)	✓		**0.0188**	**0.0087**	**0.0234**	**0.0096**	**0.7851**	0.8985	**0.0392**	**0.0167**	**0.0342**	**0.0121**	**0.8765**	**0.9757**	0.48
Dust3R† [81] (FV)	✓		0.0279	0.0133	0.0276	0.0108	0.7630	0.8841	0.0591	0.0266	0.0409	0.0136	0.8305	0.9556	1.42
Ours⋆ (FV)			0.0233	0.0108	0.0246	0.0104	0.7791	**0.9003**	0.0587	0.0239	0.0390	0.0132	0.8384	0.9616	5.83
Ours (FV)		✓	0.0239	0.0111	0.0247	0.0103	0.7768	0.8985	0.0611	0.0254	0.0392	0.0135	0.8330	0.9593	**72.04**

Table 1. **Quantitative results on 7Scenes [67] and NRGBD [6] datasets.** DUSt3R† indicates using DUSt3R's final weights on 224×224 images, same as our input resolution, to fit within 24GB GPU memory. For few-view (FV) reconstruction, we use the 8-frame pairs [26] as in SimpleRecon [63]. Note that evaluating DUSt3R at the original resolution may benefit from increased visual overlap.

Method	Opt.	Onl.	Acc↓		Comp↓		NC↑	
			Mean	Med.	Mean	Med.	Mean	Med.
Dust3R [81]	✓		**2.114**	**1.159**	**2.033**	**0.914**	**0.749**	**0.849**
Dust3R† [81]	✓		2.296	1.297	2.158	1.002	0.747	0.848
Ours⋆			2.902	1.273	2.120	0.937	0.732	0.836
Ours		✓	4.785	2.268	2.743	1.295	0.721	0.823
Dust3R [81] (FV)	✓		**2.128**	**1.241**	**2.464**	**1.228**	**0.797**	**0.889**
Dust3R† [81] (FV)	✓		2.511	1.484	2.661	1.230	0.788	0.883
Ours⋆ (FV)			3.055	1.600	2.878	1.345	0.781	0.878
Ours (FV)		✓	3.375	1.782	2.870	1.338	0.777	0.875

Table 2. **Quantitative results on DTU [1] dataset.** For few-view (FV) reconstruction, we use pairs provided in MVSNet [89].

Method	Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.
w/o lm	0.2554	0.1419	0.1470	0.0872	0.5964	0.6523
w/o clip	0.0349	0.0161	0.0249	0.0090	0.6627	0.7614
Full	**0.0342**	**0.0148**	**0.0241**	**0.0085**	**0.6635**	**0.7625**

Table 3. **Ablation studies on spatial memory.** w/o lm: use working memory only. w/o clipping the attention weight.

pointmap is directly compared with the back-projected per-point depth, excluding invalid and background points if applicable. Since the reconstruction is up to an unknown scale, we align the reconstruction following DUSt3R [81]. For DUSt3R† and our method, 224×224 inputs are generated using resizing and center cropping. Evaluation on DUSt3R and FrozenRecon with full-resolution images is restricted to the same 224×224 region for fairness. However, evaluating on full-resolution (4:3) may benefit from increased visual overlapping compared to 224×224 (1:1).

Implementation details. We initialize part of our model with pre-trained weights from DUSt3R [81, 82] with ViT-large [25] encoder, ViT-base decoders, and a DPT head [57]. For the memory encoder, we employ a light-weight ViT containing 6 self-attention blocks with the embedding dimension of 1024. Due to the computational constraint, we only train our model on 224×224 images for 120 epochs using AdamW optimizer with a learning rate of $5e-5$ and $\beta = (0.9, 0.95)$. The training takes around 10 days on 8 V100 GPUs, each with 32GB memory. The batch size is 4 per GPU, which leads to the effective batch size of 32.

4.2. Evaluation

Scene-level reconstruction. We compare the reconstruction quality with FrozenRecon [88] and DUSt3R [81], both of which are offline dense reconstruction methods that involve optimization-based alignment. As shown in Tab. 1, our model shows competitive online reconstruction quality compared to the other two offline methods while being significantly faster. This is because our model is able to predict the pointmap in a common coordinate system without the need for test-time optimization. For few-view reconstruction, our model achieves performance on par with DUSt3R†. However, since our model is trained on 224×224 images, it shows a performance gap compared to DUSt3R which uses 512×384 images for reconstruction, especially on the NRGBD [6] dataset, which contains many thin structures. Fig. 5 shows three qualitative examples on the 7scenes [67] dataset, where our model demonstrates comparable results to DUSt3R†. However, due to the absence of bundle adjustment, our model may drift. This is shown in Office-09, where a strong specular reflection in the corner causes inaccurate prediction, eventually leading to drift.

Object-level reconstruction. In Tab. 2, we evaluate the object-level reconstruction on DTU [1] dataset. DTU contains a challenging camera trajectory, starting from a top-down view, which makes online reconstruction particularly difficult. For offline reconstruction, our method achieves performance on par with DUSt3R† in terms of median Acc, Comp, and NC. It is important to note that DTU contains a black background with many thin structures (see Fig. 8). As a result, our model may produce floaters around the edges, which receive significant penalties in terms of mean Acc.

Run-time and memory footprint. Our default setting of online reconstruction can run around 65fps with 11GB GPU memory on a single 4090 GPU.

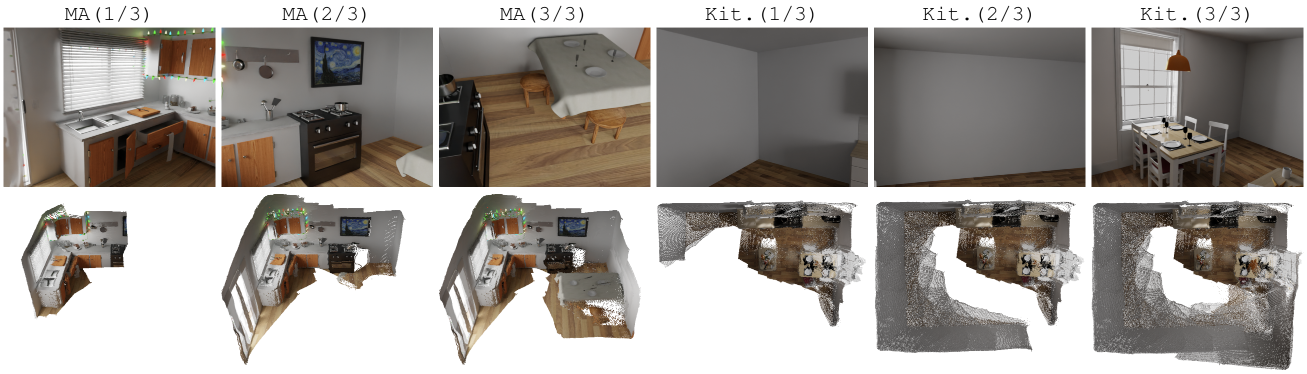

Figure 6. **Online reconstruction.** We visualize the process of our online reconstruction in two indoor scenes. In both cases, our model shows its understanding of the regularity of the indoor scene, i.e., the Manhattan World Assumption. Our model can infer the geometry of the textureless wall based on those learned regularity. However, during loop closing, our model may not fill the geometry accurately due to the accumulated errors and outliers (noisy points around the window in the second scene.)

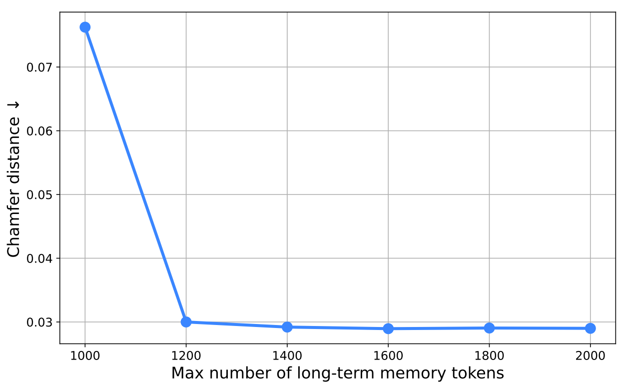

Figure 7. **Ablation study on memory size.** We plot Chamfer distance against the max number of tokens in long-term memory.

4.3. Analysis

Effect of the memory bank. We conduct an ablation study on our memory bank in Tab. 3. Without long-term memory (w/o lm), the model tends to drift quickly when relying only on the working memory, which consists of the most recent 5 frames. Additionally, without clipping attention weight, the performance of our model can degrade in certain scenes. This occurs because despite most attention weight values being small (See Fig. 4), the corresponding memory values can still differ significantly, especially when the geometry prediction contains outliers. Filtering out those small attention weights can improve the robustness of our reconstruction pipeline in various challenging scenarios. Fig. 7 shows the reconstruction quality with respect to different long-term memory sizes. In practice, we find that 4000 memory tokens are sufficient for most scenes.

Online reconstruction. We visualize the online reconstruction of two indoor scenes in Fig. 6. Our method can achieve the online reconstruction of the indoor scene even in some challenging scenarios (textureless walls). This shows a certain level of understanding of the regularity presented in indoor scenes, i.e. Manhattan World Assumption. However, one limitation is that due to the accumulated errors and outliers, our model may not fill the geometry correctly when the loop closes (See last column of Fig. 6).

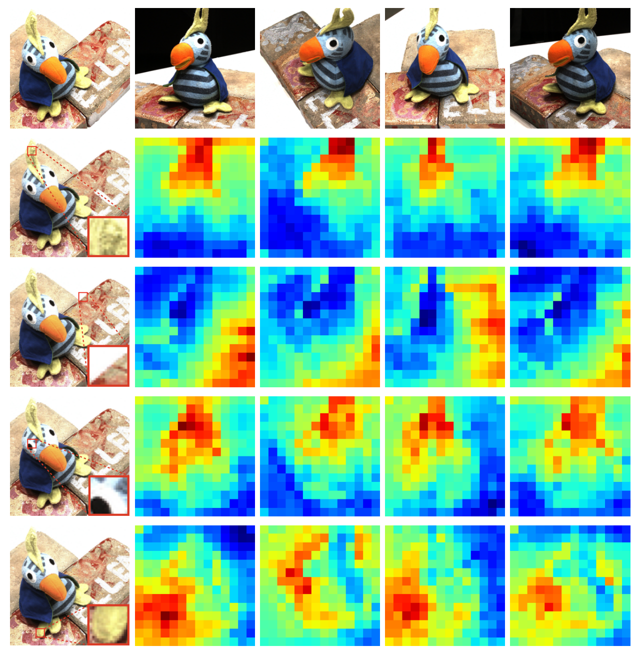

Figure 8. **Visualization of the attention map.** We visualize the attention weight of selected patches with respect to all tokens in the memory. The results show robustness toward visually similar patches (e.g., right eye/feet).

Visualization of affinity map. In Fig. 8, we visualize the attention weights corresponding to different patches in the query frame throughout the memory frames. To read out from memory, we project visual features and geometry features from two decoders into query features and memory key features as in Eq. 4 and Eq. 7. This can help to distinguish the parts with similar appearance and semantics but

Figure 9. **Qualitative examples in various real-world datasets.** We visualize several reconstruction results of Spann3R on Map-free Reloc [4], ETH3D [66], MipNeRF-360 [7], NeRF [50] and TUM-RGBD [70] datasets to demonstrate the generalization ability of our methods on different type of scenes, including indoor, outdoor, object-level, scene-level reconstruction.

in different locations (e.g. the right eye and foot of the toy).

Generalization to other unseen datasets. We demonstrate the generalization capability of Spann3R through qualitative examples of reconstructions shown in Fig. 9. These examples include results from the Map-free Reloc [4], ETH3D [66], MipNeRF-360 [7], NeRF [50] and TUM-RGBD [70] datasets. The results illustrate that Spann3R can generalize to different types of scenes and has a certain level of robustness across various challenging scenarios.

4.4. Discussion

Despite showing competitive results across various datasets, our method still has some inherent limitations. We will describe several limitations and potential directions next.

Large-scale scene reconstruction. Our model can deal with large-scale object-centric scenes fairly well. However, in cases where the camera continuously moves forward or reconstructs large multi-room scenes, our model might fail. This limitation arises due to the limited memory size during training. Since our training process assumes the camera pose of the first frame is the identity, training on just 5 frames typically spans only a limited spatial region. To address this issue, one approach could be to restart our model every few frames and then align the different fragments using PnP-RANSAC. Alternatively, a more scalable sampling strategy in training or a more structured memory system at inference is needed to overcome this challenge.

Bundle adjustment. For an incremental reconstruction pipeline, bundle adjustment is usually of great importance for mitigating error accumulation. In the case of Spann3R, the question would be: Can we learn to update and fuse our memory when incorporating new observations? Alternatively, since the concept of Spann3R is to predict the next frame based on previous predictions, we could potentially integrate traditional bundle adjustment techniques to correct the geometry. The model could then encode this corrected geometry into the spatial memory, leading to more accurate predictions in subsequent frames.

Training data. Due to the constraint of the computational resources, we only train our model across 6 datasets using five 224×224 images sampled from the entire sequence. We expect training on the entire datasets of DUSt3R [81], either with more than five images or at a higher 512 resolution, could further improve the accuracy. Moreover, the current model relies on a substantial amount of posed RGB-D data. It is worth exploring how to effectively learn data-driven prior from casual videos using self-supervised training.

5. Conclusion

We have presented Spann3R, a model capable of achieving incremental reconstruction from RGB images without requiring prior knowledge of the camera parameters. By introducing the concept of spatial memory, which encodes previous states for next-frame prediction, Spann3R reconstructs scenes through a simple forward pass with a transformer-based architecture, eliminating the need for test-time optimization. This enables online reconstruction in real time. Trained on various large-scale datasets, Spann3R demonstrates competitive reconstruction quality and generalization ability across various scenarios. Future work includes extending our method to handle large-scale scenes, incorporating bundle adjustment techniques, and exploring self-supervised training on casual videos.

Acknowledgements. The research presented here has been supported by a sponsored research award from Cisco Research and the UCL Centre for Doctoral Training in Foundational AI under UKRI grant number EP/S021566/1. This project made use of time on Tier 2 HPC facility JADE2, funded by EPSRC (EP/T022205/1). We thank Weilue Luo and Jingyi Zhang for their helpful suggestions.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision (IJCV)*, 120:153–168, 2016. 5, 6

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 72–79, 2009. 1, 2

[3] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 29–42, 2010. 1, 2

[4] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian 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)*, 2022. 8

[5] Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. In *Psychology of Learning and Motivation*, pages 89–195. Elsevier, 1968. 2, 3

[6] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. 5, 6

[7] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 3, 8

[8] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 19697–19705, 2023. 3

[9] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 5

[10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 1, 2

[11] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2023. 3

[12] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018. 1

[13] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6684–6692, 2017. 2

[14] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5044–5053, 2023.

[15] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. *arXiv preprint arXiv:2404.14351*, 2024. 2

[16] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 640–658. Springer, 2022. 2, 3, 4

[17] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11781–11794, 2021.

[18] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3151–3161, 2024. 3

[19] David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3001–3008, 2011. 1, 2

[20] 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)*, pages 5828–5839, 2017. 5

[21] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007. 1, 2

[22] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 224–236, 2018. 1, 2

[23] Eric Dexheimer and Andrew J Davison. Learning a depth covariance function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13122–13131, 2023. 2

[24] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisser-

man. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10061–10072, 2023. 3

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 3, 6

[26] 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)*, pages 15324–15333, 2021. 2, 6

[27] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 834–849, 2014. 2

[28] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2017. 2

[29] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(8):1362–1376, 2009. 1, 2

[30] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 873–881, 2015. 1, 2

[31] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1, 2

[32] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[33] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, pages 1–11, 2024. 3

[34] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21584–21593, 2024. 3

[35] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 12949–12958, 2021. 3

[36] Wonbong Jang and Lourdes Agapito. Nvist: In the wild new view synthesis from a single image with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10181–10193, 2024. 3

[37] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3

[38] 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)*, pages 9492–9502, 2024. 2

[39] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21357–21366, 2024. 3

[40] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139–1, 2023. 3

[41] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, pages 1–10, 2007. 1, 2

[42] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J Davison. vmap: Vectorised object mapping for neural field slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 952–961, 2023. 3

[43] 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)*, pages 2041–2050, 2018. 5

[44] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5987–5997, 2021. 2

[45] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 17627–17638, 2023. 2

[46] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999. 1, 2

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

[48] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18039–18048, 2024. 3

[49] 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)*, pages 4040–4048, 2016. 5

[50] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020. 3, 8

[51] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, pages 1400–1409, 2016. 2, 3

[52] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[53] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011. 2

[54] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011. 1, 2

[55] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9226–9235, 2019. 3

[56] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes Lutz Schönberger. Global structure-from-motion revisited. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2

[57] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 6

[58] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10901–10911, 2021. 5

[59] Jerome Revaud, Yohann Cabon, Romain Brégier, JongMin Lee, and Philippe Weinzaepfel. Sacreg: Scene-agnostic coordinate regression for visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 688–698, 2024. 2

[60] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011. 1, 2

[61] 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)*, pages 4938–4947, 2020. 1, 2

[62] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9339–9347, 2019. 5

[63] 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)*, pages 1–19, 2022. 2, 6

[64] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2, 3

[65] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–518, 2016. 1, 2

[66] Thomas Schops, Johannes L Schonberger, 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)*, pages 3260–3269, 2017. 8

[67] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013. 5, 6

[68] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006. 1, 2

[69] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, 2024. 3

[70] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, 2012. 8

[71] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6229–6238, 2021. 3

[72] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2440–2448, 2015. 2, 3

[73] 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)*, pages 8922–8931, 2021. 1

[74] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2446–2454, 2020. 5

[75] Chris Sweeney, Torsten Sattler, Tobias Hollerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 801–809, 2015. 1, 2

[76] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 3

[77] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, pages 298–372, 2000. 1, 2

[78] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Coslam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13293–13302, 2023. 3, 5

[79] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21686–21697, 2024. 1

[80] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27171–27183, 2021. 3

[81] 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)*, pages 20697–20709, 2024. 2, 3, 4, 5, 6, 8

[82] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pretraining for stereo matching and optical flow. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 17969–17980, 2023. 6

[83] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 2, 3

[84] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 61–75, 2014. 1, 2

[85] Changchang Wu. Towards linear-time incremental structure from motion. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 127–134, 2013. 1, 2

[86] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore bundle adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064, 2011. 1, 2

[87] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20406–20417, 2024. 3

[88] Guangkai Xu, Wei Yin, Hao Chen, Chunhua Shen, Kai Cheng, and Feng Zhao. Frozenrecon: Pose-free 3d scene reconstruction with frozen depth models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9276–9286, 2023. 5, 6

[89] 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)*, pages 767–783, 2018. 1, 2, 6

[90] 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)*, pages 1790–1799, 2020. 5

[91] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4805–4815, 2021. 3

[92] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 5

[93] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 467–483, 2016. 2

[94] 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)*, pages 9043–9053, 2023. 1, 2

[95] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12786–12796, 2022. 3, 5