Splatt3R: Zero-shot Gaussian Splatting from Uncalibrated Image Pairs

Brandon Smart[1] Chuanxia Zheng[2] Iro Laina[2] Victor Adrian Prisacariu[1]
[1]Active Vision Lab, University of Oxford [2]Visual Geometry Group, University of Oxford
{brandon, cxzheng, iro, victor}@robots.ox.ac.uk

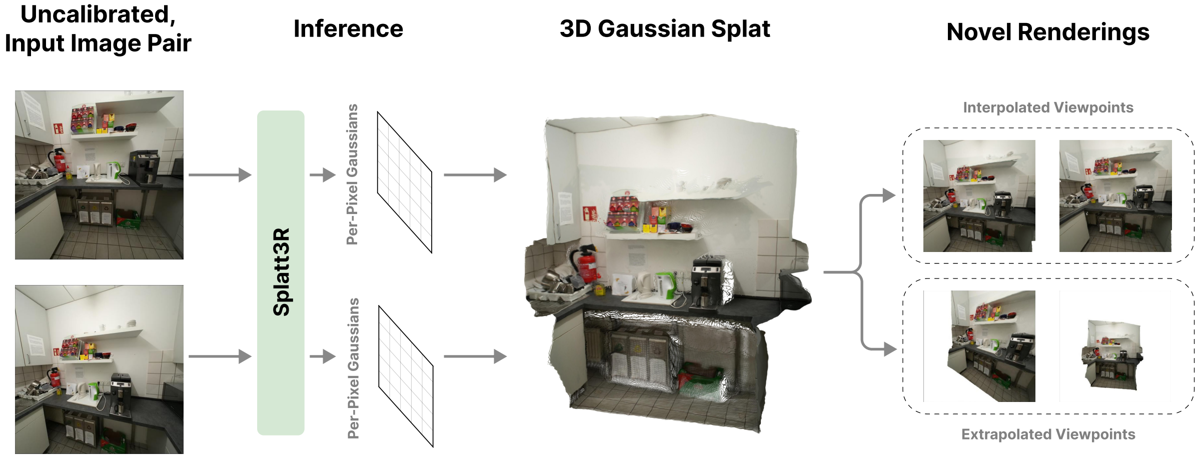

Figure 1. We introduce *Splatt3R*, a feed-forward model that can directly predict a 3D Gaussian Splat from a stereo pair of images with unknown camera parameters. We base our work on MASt3R, and following their simple architecture, we avoid any explicit prediction of camera poses, intrinsics or monocular depth. Splatt3R can perform both interpolation and extrapolation of novel views from the input.

Abstract

In this paper, we introduce Splatt3R, a pose-free, feed-forward method for in-the-wild 3D reconstruction and novel view synthesis from stereo pairs. Given uncalibrated natural images, Splatt3R can predict 3D Gaussian Splats without requiring any camera parameters or depth information. For generalizability, we build Splatt3R upon a "foundation" 3D geometry reconstruction method, MASt3R, by extending it to deal with both 3D structure and appearance. Specifically, unlike the original MASt3R which reconstructs only 3D point clouds, we predict the additional Gaussian attributes required to construct a Gaussian primitive for each point. Hence, unlike other novel view synthesis methods, Splatt3R is first trained by optimizing the 3D point cloud's geometry loss, and then a novel view synthesis objective. By doing this, we avoid the local minima present in training 3D Gaussian Splats from stereo views. We also propose a novel loss masking strategy that we empirically find is critical for strong performance on extrapolated viewpoints. We train Splatt3R on the ScanNet++ dataset and demonstrate excellent generalisation to uncalibrated, in-the-wild images. Splatt3R can reconstruct scenes at 4FPS at 512×512 *resolution, and the resultant splats can be rendered in real-time.*

1. Introduction

We consider the problem of 3D scene reconstruction and novel view synthesis from sparse, *uncalibrated natural images* in just one forward pass of a trained model. While recent breakthroughs have been made in 3D reconstruction and novel view synthesis by using neural scene representations, *e.g.* SRN [47], NeRF [41], LFN [48], and non-neural scene representations, *e.g.* 3D Gaussian Splatting (3D-GS) [29], these methods are far from being accessible to casual users, due to expensive, iterative, per-scene optimization procedures, which are often slow and are unable to utilize learned priors from training datasets. More importantly, reconstruction quality is poor when trained from only a pair of stereo images, as these methods require a dense collection of dozens or hundreds of images to produce high-quality results.

To mitigate these issues, generalizable 3D reconstructors [8, 12, 16, 27, 56, 63], aim to predict pixel-aligned features for radiance fields from sparse *calibrated images* using feed-forward networks. These models are trained by differentiably rendering the predicted, parameterized representations from target viewpoints and supervising them with ground truth images captured from the same camera pose. By learning priors across large datasets of input scenes, these models avoid the failure cases of traditional per-scene optimization from sparse images. To avoid the expensive volumetric processing in NeRF, several feed-forward Gaussian Splatting models [7, 10, 51, 52, 68] have been proposed to explore 3D reconstruction from sparse views. They use a cloud of pixel-aligned 3D Gaussian primitives [29] to represent the scene. The 3D locations of these Gaussian primitives are parameterized using their depth along the ray, which is explicitly calculated using the *known intrinsic and extrinsic camera parameters* from the input images.

Due to their reliance on known camera parameters, these methods can *not* be directly used on "in-the-wild" uncalibrated images. Ground truth poses are assumed to be available, or camera pose estimation is implied as a preprocessing step — existing methods are typically tested on datasets where poses have been reconstructed by running SfM software on dozens or hundreds of images of the same scene. Methods which attempt to use SfM or multi-view stereo (MVS) pipelines typically use a string of algorithms for matching points, triangulating them, finding essential matrices, and estimating camera extrinsics and intrinsics.

In this paper, we introduce Splatt3R, a feed-forward model that takes as input two uncalibrated images, and outputs 3D Gaussians to represent the scene. Specifically, we use a feed-forward model to predict pixel-aligned 3D Gaussian primitives for each image, and then render novel views using a differentiable renderer. We achieve this without relying on any additional information such as camera intrinsics, extrinsics, or depth.

Without explicit pose information, one key challenge is identifying where to place the 3D Gaussian centers. Even with pose information, iterative 3D Gaussian Splatting optimization is susceptible to local minima [7, 29]. Our solution is to jointly address the lack of camera poses and the problem of local minima by explicitly supervising and regressing the ground truth 3D point clouds for each training sample. In particular, we observe that the architecture used to produce MASt3R's pixel-aligned 3D point clouds [31] closely aligns with the existing pixel-aligned 3D Gaussian splatting architectures using in feed-forward Gaussian methods [7, 10, 51, 52]. Therefore, we seek to show that simply adding a Gaussian decoder to a large-scale pre-trained "foundation" 3D MASt3R model, without any bells and whistles, is sufficient to develop a *pose-free*, *generalizable* novel view synthesis model.

One notable limitation of most existing generalizable 3D-GS methods is that they only supervise novel viewpoints which are between the input stereo views [7, 10], rather than learning to extrapolate to farther viewpoints. The challenge with these extrapolated viewpoints is that they often see points that are obscured to the input camera views, or are outside of their frustums entirely. Thus, supervising the novel view rendering loss for these points is counterproductive, and can be destructive to the model's performance. By only supervising the novel view rendering loss for views that are between the two context images, existing works avoid attempting to reconstruct many unseen parts of the scene. However, this means that the model is not trained to accurately generate novel view renderings for views beyond the stereo baseline. To address this, we employ a loss masking strategy based on frustum culling and covisibility testing, calculated using the ground truth poses and depth maps known during training. We apply mean squared error and LPIPS loss only to the parts of the rendering that can be feasibly reconstructed, preventing updates to our model from unseen parts of the scene. This allows training with wider baselines, and for supervising novel views that are beyond the stereo baseline.

We present, for the first time, a method that predicts 3D Gaussian Splats for scene reconstruction and novel view synthesis from a pair of unposed images in a single forward pass of a network. We construct baselines out of existing work and show that our method surpasses them in visual quality and perceptual similarity to the ground truth images. More impressively, our trained model is capable of generating photorealistic novel view synthesis from in-the-wild uncalibrated images. This significantly relaxes the need for dense image inputs with precise camera poses, addressing a major challenge in the field.

2. Related Work

2.1. Novel View Synthesis

Many representations have been used for 3D Novel View Synthesis (NVS), such as luminagraphs [19], light fields [32], and plenoptic functions [1]. Neural Radiance Fields (NeRFs) have achieved photo-realistic representations of 3D scenes using view-dependent, ray-traced radiance fields, encoded by neural networks trained through per-scene optimization on densely collected image sets [3, 41, 42]. Recently, 3D Gaussian Splatting [29] has greatly increased the training and rendering speed of radiance fields by training a set of 3D Gaussian primitives to represent the radiance of each point in space, and rendering them through an efficient rasterization process.

To avoid intensive per-scene optimization, generalizable NVS pipelines have been developed, which infer 3D representations directly from multi-view images [8, 12, 27, 33, [36, 37, 46, 49, 50, 56, 59, 63, 68]. Rather than performing per-scene optimization, these methods are trained across large datasets of scenes, allowing data-driven priors to be learned that can ground reconstruction for newly observed scenes. By leveraging these data-driven priors, these methods have evolved to work with sparse image sets [10, 34, 38, 43] and even stereo image pairs [7, 16, 30, 69], significantly reducing the number of reference images required to obtain a radiance field for NVS.

Recent methods, such as pixelSplat [7], MVSplat [10], GPS-Gaussian [68], SplatterImage [52] and Flash3D [52] use a cloud of 3D Gaussian primitives placed along camera rays explicitly calculated from camera parameters, aiming to predict one (or multiple) 3D Gaussian primitives per-pixel in each image. However, these existing methods assume the availability of camera intrinsics and extrinsics for each image at testing time, which limits their applicability to in-the-wild photo pairs. Many methods have been proposed for per-scene optimization with unknown camera poses [4, 5, 25, 35, 58], however these depend on large collections of images. Recent studies propose methods to jointly predict camera parameters and 3D representations in a generalizable manner, although these are limited to sparse setups [26, 54]. In contrast, we propose Splatt3R to address the gap in generalizable stereo NVS with unknown camera parameters. Among closely related works, FlowCam [49] removes the need for pre-computed cameras using dense correspondences from optical flow, but it requires sequential input and shows limited rendering performance.

By integrating the recent stereo reconstruction work MASt3R with 3D Gaussians, our method effectively handles larger baselines without the need for pre-processed cameras. GGRt [33] also seeks to model 3D Gaussian Splats without known camera poses or intrinsics, but instead focuses on processing video sequences with small baselines between frames, introducing caching and deferred backpropogation techniques to aid reconstruction from long video sequences. DBARF [9] also aims to jointly learn camera poses and reconstruct radiance fields, but uses a NeRF-based approach and focuses on calculating poses using cost maps derived from learned features.

2.2. Stereo Reconstruction

Traditionally, the stereo reconstruction task involves a sequence of steps. Starting with keypoint detection and feature matching [14, 20, 39, 53], camera parameters are estimated using fundamental matrices [40, 44, 67]. Next, dense correspondence is established through epipolar line search [2, 24, 28] or stereo matching [6, 64, 66], enabling the triangulation of 3D points [21–23]. This process can be optionally refined by photometric bundle adjustment [13, 60]. With the advent of deep learning, numerous methods have been proposed to integrate certain steps, such as joint depth and camera pose estimation, and optical flow [11, 17, 18, 55, 62, 65]. However, all these methods rely on explicit correspondence, making them challenging to apply when the overlap between images is limited.

Recently, DUSt3R [57] introduced an innovative approach to address this challenge by predicting point maps for a pair of uncalibrated stereo images in one coordinate system with implicit correspondence searching. The follow-up paper MASt3R [31] primarily focuses on improvements to image matching, but improves on DUSt3R by predicting points in metric space and achieving greater accuracy. These methods have shown promising stereo reconstruction results even when there is little or no overlap between the images. While the raw point maps are sufficiently accurate for several downstream applications like pose estimation, they are not designed to be directly rendered. In contrast, our method augments MASt3R to predict 3D Gaussian primitives, which enables fast and photorealistic NVS.

3. Method

Given two uncalibrated images $\mathcal{I} = \{\mathbf{I}^i\}_{i=\{1,2\}}$, ($\mathbf{I}^i \in \mathbb{R}^{H\times W\times 3}$), our goal is to learn a mapping Φ that takes as input $\mathcal{I}$ and outputs 3D Gaussian parameters for both geometry and appearance. We achieve this by simply adding a third branch to MASt3R to output the additional attributes required for 3D Gaussians. Before outlining the details of our proposed method, we provide a brief overview of 3D Gaussian Splatting in Sec. 3.1, followed by an overview of MASt3R in Sec. 3.2. We then describe how we modify the MASt3R architecture to predict 3D Gaussian Splats for novel view synthesis in Sec. 3.3. Finally, we outline our training and evaluation protocols in Sec. 3.4.

3.1. 3D Gaussian Splatting

Scenes as sets of 3D Gaussians. We begin by briefly reviewing 3D Gaussian Splatting (3D-GS) [29]. 3D-GS represents the radiance field of a scene using a set of anisotropic, 3D Gaussians, each of which represents the radiance emitted in the spatial region around a point. Each Gaussian is parameterized using its mean position $\boldsymbol{\mu} \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, covariance $\Sigma \in \mathbb{R}^{3\times 3}$ and view-dependent color $S \in \mathbb{R}^{3\times d}$ (here parameterized using d-degree spherical harmonics). Like other works, we reparameterize the covariance matrix with a rotation quaternion $q \in \mathbb{R}^4$ and scale $s \in \mathbb{R}^3$ to ensure positive semi-definite covariance matrices. In our experiments, we focus on constant, view-independent color for each gaussian ($S \in \mathbb{R}^3$), and ablate view-dependent spherical harmonics. Original 3D-GS uses an iterative process to fit the Gaussian Splats to a single scene, but Gaussian primitives have vanishingly small gradients if the distance to their 'correct' location is greater than a few standard deviations, and can often get stuck in

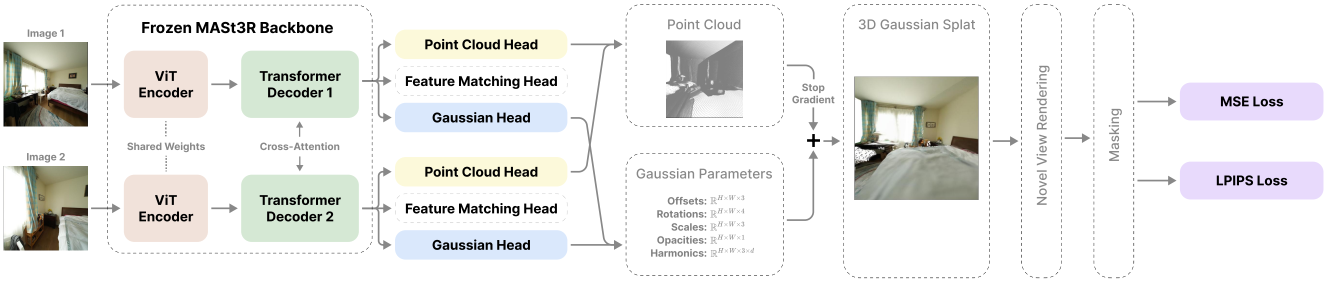

Figure 2. **Method overview.** We encode the two uncalibrated images using MASt3R's pretrained ViT encoder and cross-attention transformers, which we freeze during training. In addition to MASt3R's prediction head for point positions and confidences, we introduce a Gaussian head that predicts offsets, spherical harmonics, rotations, scales and opacities. We supervise novel renderings of the generated 3D Gaussian Splats using mean squared error (MSE) and LPIPS.

local optima during optimization [7]. 3D-GS partially overcomes these problems using initialization from SfM point clouds and non-differentiable 'adaptive density control' to split and prune gaussians [29]. This method is effective, but requires a dense collection of images, and cannot be used for generalizable, feed-forward models, which directly predict Gaussians without per-scene optimization.

Feed-forward 3D Gaussians. Very recently, given a set of N images $\mathcal{I} = \{\mathbf{I}^i\}_{i=1}^N$, the generalizable 3D-GS methods [7, 10, 51, 52] predict pixel-aligned 3D Gaussian Primitives. In particular, for each pixel $\boldsymbol{u} = (u_x, u_y, 1)$, the parameterized Gaussian is predicted with its opacity α, depth d, offsets Δ, covariance Σ expressed as rotation and scale, and the parameters of the colour model S. The location of each Gaussian is given by $\boldsymbol{\mu} = K^{-1}\boldsymbol{u}d + \Delta$, where K is the camera intrinsics. Of particular note, pixelSplat predicts a probabilistic distribution over depth, which seeks to avoid the problem of local minima by tying the probabilistic density to the opacity of the Gaussian primitives sampled [7]. However, these parameterizations cannot be directly applied to 3D-GS prediction from *uncalibrated images*, which have unknown camera rays. Instead, we directly supervise the positions of per-pixel Gaussian primitives using 'ground-truth' point clouds. This allows the Gaussian corresponding to each pixel to have a direct path of monotonically decreasing loss leading to its correct position during training.

3.2. MASt3R Training

As discussed, we wish to directly supervise the 3D location of each pixel in a pair of uncalibrated images. This task has recently been explored by DUSt3R [57] (and its follow-up work MASt3R [31]), a multi-view stereo reconstruction method that directly regresses a model for predicting 3D point clouds. For simplicity, we collectively refer to these methods as 'MASt3R' for the remainder of the paper.

Given two images $\mathbf{I}^1, \mathbf{I}^2 \in \mathbb{R}^{W\times H\times 3}$, MASt3R learns to predict the 3D locations for each pixel $\hat{X}^1, \hat{X}^2 \in \mathbb{R}^{W\times H\times 3}$, alongside corresponding confidence maps $C^1, C^2 \in \mathbb{R}^{W\times H}$. Here, the model aims to predict both point maps in the coordinate frame of the first image, which removes the need for transforming point clouds from one image's coordinate frame to the other using camera poses. This representation, like generalizable 3D reconstruction approaches, assumes the existence of a single, unique location where the ray corresponding to each pixel intersects with the surface geometry, and does not attempt to model non-opaque structures like glass or fog.

Given ground truth pointmaps X^1 and X^2, for each valid pixel i in each view $v \in \{1, 2\}$, the training objective L_{pts} is defined as:

$$L_{pts} = \sum_{v\in\{1,2\}} \sum_i C_i^v L_{regr}(v, i) - \gamma \log(C_i^v) \quad (1)$$

$$L_{regr}(v, i) = \left\| \frac{1}{z} X_i^v - \frac{1}{\bar{z}} \hat{X}_i^v \right\| \quad (2)$$

L_{pts} is a confidence-based loss used to handle points with ill-defined depths, such as points corresponding to the sky, or to translucent objects. The hyperparameter γ governs how confident the network should be, while z and $\bar{z}$ are normalization factors used for non-metric datasets (set to $z = \bar{z} = 1$ for metric datasets). In our experiments, we use a frozen MAST3R model pre-trained with this objective, and only apply novel view rendering losses during training. We experiment with fine-tuning using this loss in Tab. 2.

3.3. Adapting MASt3R for Novel View Synthesis

We now present *Splatt3R*, a feed-forward model that predicts 3D Gaussians from uncalibrated image pairs. Our key motivation derives from the conceptual similarity between MASt3R and generalizable 3D-GS models, such as pixelSplat [7] and MVSplat [10]. First, these methods all use feed-forward, cross-attention network architectures to extract information between input views. Second, MASt3R predicts pixel-aligned 3D points (and confidence) for each image, whereas generalizable 3D-GS models [7, 10, 51, 52] predict pixel-aligned 3D Gaussians for each image. Thus, we follow the spirit of MASt3R, and show that a simple modification to the architecture, alongside a well-chosen training loss, is sufficient to achieve strong novel view synthesis results.

Formally, given a set of uncalibrated images $\mathcal{I}$, MASt3R encodes each image $\mathcal{I}^i$ simultaneously using a vision transformer (ViT) encoder [15], then passes them to a transformer decoder which performs cross-attention between each image. Normally, MASt3R has two prediction heads, one that predicts a 3D point (x) and confidence (c) for each pixel, and a second which is used for feature matching, which is not relevant to our task, and can be ignored. We introduce a third head, which we refer to as the *'Gaussian head'*, that runs *in parallel* to the existing two heads. This head predicts covariances (parameterized by rotation quaternions $q \in \mathbb{R}^4$ and scales $s \in \mathbb{R}^3$), spherical harmonics ($S \in \mathbb{R}^{3\times d}$) and opacities ($\alpha \in \mathbb{R}$) for each point. Additionally, we predict an offset ($\Delta \in \mathbb{R}^3$) for each point, and parameterize the mean of the Gaussian primitive as $\mu = x + \Delta$. This allows us to construct a complete Gaussian primitive for each pixel, which we can then render for novel view synthesis.

During training, we only train the Gaussian prediction head, relying on a pre-trained MASt3R model for the other parameters. Following MASt3R's point prediction head, we use the DPT architecture [45] for our Gaussian head. An overview of the model architecture is shown in Fig. 2.

Following existing generalizable 3D-GS works, we use different activation functions for each Gaussian parameter type, including normalization for quaternions, exponential activations for scales and offsets, and sigmoid activations for opacities. Additionally, to aid in the learning of high-frequency color, we seek to predict the residual between each pixel's color and the color we apply to that pixel's corresponding Gaussian primitive.

Following MAST3R's practice of predicting the 3D locations of all points in the first image's camera frame, the predicted covariances and spherical harmonics are considered as being in the first image's camera frame. This avoids the need to use ground truth transformations to convert these parameters between reference frames, which existing methods do [7]. The final set of Gaussian primitives is the union of the Gaussian primitives predicted from both images.

3.4. Training Procedure and Loss Calculation

To optimize our Gaussian parameter predictions we supervise novel view renderings of the predicted scene, as in existing work [7, 10, 52]. During training, each sample consists of two input 'context' images which we use to reconstruct the scene, and a number of posed 'target' images which we use to calculate rendering loss.

Some of these target images may contain regions of the scene that were not visible to the two context views due to being obscured, or outside of the context view frustums entirely. Supervising the rendering loss for these pixels would be counterproductive and potentially destructive to the model's performance. Existing generalizable, feed-forward radiance field prediction methods attempt to avoid this problem by only synthesizing novel views for viewpoints that are between the input stereo views [7, 10, 16], reducing the number of unseen points that need to be reconstructed. Instead, we seek to train ours to extrapolate to farther viewpoints that are not necessarily an interpolation between the two input images.

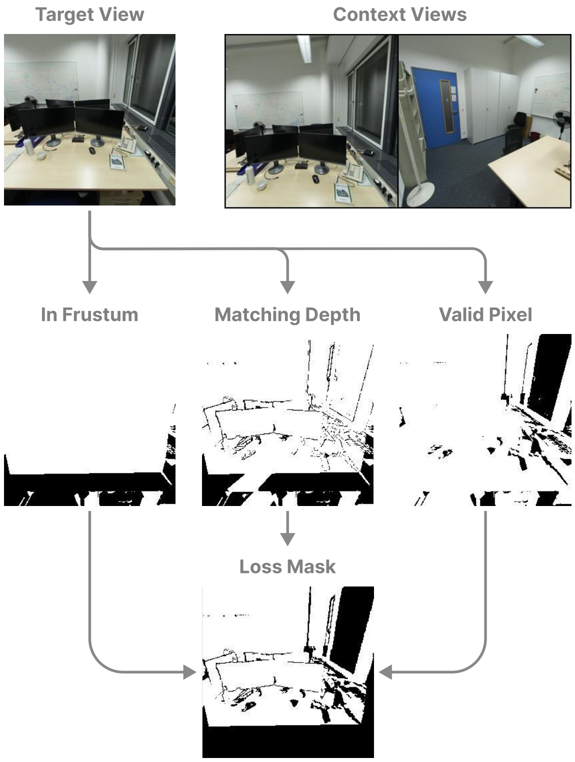

Figure 3. **Our loss masking approach.** Valid pixels are considered to be those that are: inside the frustum of at least one of the views, have their reprojected depth match the ground truth, and are considered valid pixels with valid depth in their dataset.

To address this, we introduce a loss masking strategy. For each target image, we calculate which pixels are visible in at least one of the context images. We unproject each point in the target image and reproject it onto each of the context images, checking if the rendered depth closely matches the ground truth depth. We show the construction of an example loss mask Fig. 3.

Like existing generalized 3D-GS approaches [7, 10, 52], we train using a *weighted* combination of mean squared error loss (MSE) and perceptual similarity. Given, our rendered images ($\hat{\mathbf{I}}$), ground truth images ($\mathbf{I}$), and rendered loss

	Close ($\phi = 0.9, \psi = 0.9$)			Medium ($\phi = 0.7, \psi = 0.7$)			Wide ($\phi = 0.5, \psi = 0.5$)			Very Wide ($\phi = 0.3, \psi = 0.3$)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Splatt3R (Ours)	**19.66** **(14.72)**	**0.757** -	**0.234** **(0.237)**	**19.66** **(14.38)**	**0.770** -	**0.229** **(0.243)**	**19.41** **(13.72)**	**0.783** -	**0.220** **(0.247)**	**19.18** **(12.94)**	**0.794** -	**0.209** **(0.258)**
MASt3R (Point Cloud)	18.56 (13.57)	0.708 -	0.278 (0.283)	18.51 (12.96)	0.718 -	0.259 (0.280)	18.73 (12.50)	0.739 -	0.245 (0.293)	18.44 (11.27)	0.758 -	0.242 (0.322)
pixelSplat (MASt3R cams)	15.48 (10.53)	0.602 -	0.439 (0.447)	15.96 (10.64)	0.648 -	0.379 (0.405)	15.94 (10.14)	0.675 -	0.343 (0.394)	16.46 (10.12)	0.708 -	0.302 (0.373)
pixelSplat (GT cams)	15.67 (10.71)	0.609 -	0.436 (0.443)	15.92 (10.61)	0.643 -	0.381 (0.407)	16.08 (10.33)	0.672 -	0.407 (0.392)	16.56 (10.20)	0.709 -	0.299 (0.370)

Table 1. **Comparisons with the state of the art.** Performances are averaged over test scenes in ScanNet++. For each scene, the model takes two, unposed images as input and renders novel views for evaluation. Splatt3R shows improvements on all visual metrics.

masks M, the masked reconstruction loss is:

$$\begin{aligned} L = &\lambda_{MSE} L_{MSE}(M \odot \hat{\mathbf{I}}, M \odot \mathbf{I}) \\ &+ \lambda_{LPIPS} L_{LPIPS}(M \odot \hat{\mathbf{I}}, M \odot \mathbf{I}) \end{aligned} \tag{3}$$

During training, existing methods [7, 10, 51] assume that the images of each scene are in a video sequence. These methods use the number of frames between chosen context images as a proxy for the distance and overlap between the images, and select intermediary frames as the target frames for novel view synthesis supervision. We seek to generalize this approach to work with datasets of frames that are not in the form of a linear sequence, and to allow supervision from views that are not in-between the context images. During preprocessing, we calculate the overlap masks for each pair of images for each scene in the training set. During training, we select context images such that at least ϕ% of the pixels in the second image have direct correspondences in the first, and target images such that at least ψ% of the pixels are present in at least one of the context images.

4. Experimental Results

Next, we describe our experimental setup (Sec. 4.1), evaluate our method with a comparison to baselines (Sec. 4.2), and assess the significance of our model's components with an ablation study (Sec. 4.3).

4.1. Training and Evaluation Setup

Training details. During each epoch, we randomly sample two input images, and three target images from each scene in the training split. As described in Section 3.4, we select views using ϕ and ψ parameters which we set at $\phi = \psi = 0.3$. We train our model for 2000 epochs ($\approx$ 500,000 iterations) at a resolution of 512 × 512, using $\lambda_{MSE} = 1.0$ and $\lambda_{LPIPS} = 0.25$. We optimize using the Adam optimizer at learning rate of 1.0×10^{-5}, with a weight decay of 0.05, and a gradient clip value of 0.5.

Training data. We train our model using ScanNet++ [61], which is a dataset of 450+ indoor scenes with ground truth depth obtained from high-resolution laser scans. We use the official ScanNet++ training and validation splits.

Testing datasets. We construct four testing subsets from ScanNet++ scenes to represent close together views (for high ϕ and ψ) and farther views with less overlap (for low ϕ and ψ). The test scenes are not seen during training. We ignore the frames marked as 'bad' in the ScanNet++ dataset, and scenes that contain frames with no valid depth.

Metrics are calculated after applying the loss masks to the rendered and target images. Metrics are reported both across the entire image, and averaging across just the pixels in the loss mask (in parenthesis for PSNR and LPIPS).

Baselines. To the best of our knowledge, Splatt3R is the first model that performs 3D reconstruction from a wide, unposed, stereo pair of images for novel view synthesis in a feed-forward manner. To evaluate our method, we construct baselines from existing works. We test our method against directly rendering MASt3R's prediction as a colored point cloud, giving each point the color of its corresponding pixel. We wish to reconstruct and render the entire 3D scene, therefore we do not filter out points with low confidences from our point cloud renderings. We also compare our method against pixelSplat [7], a generalizable 3D-GS reconstruction method that requires poses for reconstruction. We evaluate pixelSplat using ground truth camera poses, and also using camera poses estimated using the point clouds predicted from MASt3R. Please see the MASt3R paper for details on pose regression from MASt3R's predictions [31]. We retrain baselines with the same dataloaders and training curricula where appropriate to present a fair comparison. Due to memory constraints when training pixelSplat, we train at 256x256, and initialize the model using the pretrained weights from the pixelSplat authors. We observe that when trained using the same data schedule, pixelSplat achieves very low accuracy. Therefore we adapt pixelSplat's curriculum learning strategy for our data, initially training the model at $\phi = \psi = 0.7$, and decreasing these values to $\phi = \psi = 0.3$ at the end of training.

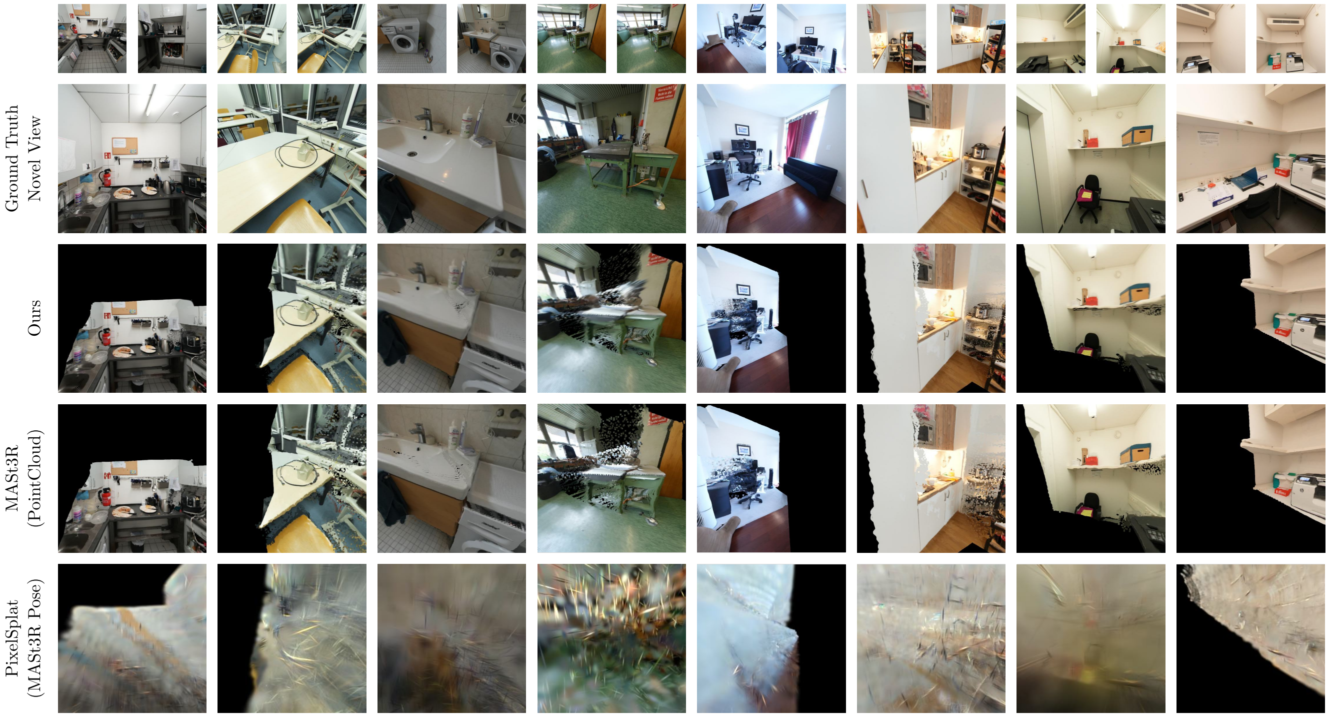

Figure 4. **Qualitative comparisons on ScanNet++.** We compare different methods on ScanNet++ testing examples. The two context camera views for each image are included in the first row of the table.

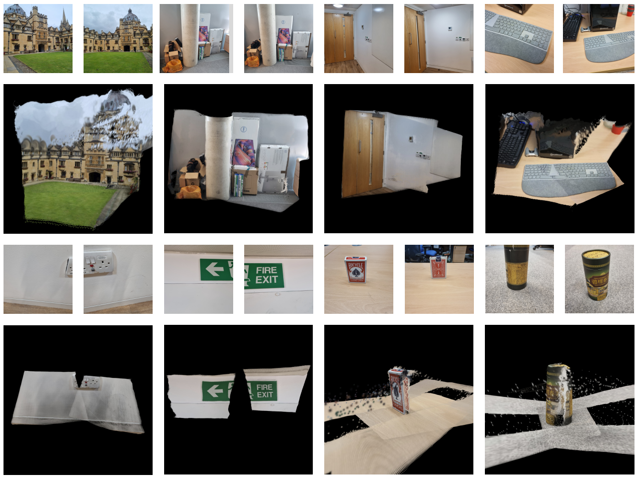

Figure 5. **Examples of Splatt3R generalizing to in-the-wild testing examples.** The bottom row showcases examples with few direct pixel correspondences between the two context images.

4.2. Results

Quantitative evaluation. We begin by reporting our quantitative results for ScanNet++ in Tab. 1. Our method outperforms both directly rendering the MASt3R point cloud, and reconstructing the scene using pixelSplat across all stereo baseline sizes. Critically, we find that our method outperforms pixelSplat even when pixelSplat is evaluated using the ground truth poses for each camera. When trained using the stereo baselines in our dataset, and when supervised from viewpoints which contain information not visible to the input cameras, we observe that the quality of reconstructions from pixelSplat significantly degrades.

Qualitative comparisons. Next, we provide a qualitative comparison of each method using examples from ScanNet++ in Fig. 4. We see that our method, like MASt3R is able to reconstruct the visible regions of the scene, while not attempting to reconstruct areas which are not visible to the context views. By masking our novel view rendering loss, our model does not learn to guess unseen regions of the scene. pixelSplat has a very poor reconstruction quality, visibly attempting to predict regions of the scene which cannot be seen from the input context views, and achieving poor accuracy even in reconstructable regions of the scene. We also note the visual artifacts which are present when directly rendering the point clouds from MASt3R. Our learned 3D Gaussian representation is able to reduce the number of these artifacts, resulting in marginally higher quality renderings.

Here, we also note that our model is reconstructing the scene in metric scale. We can observe the accuracy of this scale prediction by noting how closely the viewpoint of the rendered image matches the ground truth image taken from

	Close ($\phi = 0.9$, $\psi = 0.9$)			Medium ($\phi = 0.7$, $\psi = 0.7$)			Wide ($\phi = 0.5$, $\psi = 0.5$)			Very Wide ($\phi = 0.3$, $\psi = 0.3$)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	19.66 (14.72)	0.757 -	0.234 (0.237)	19.66 (14.38)	0.770 -	0.229 (0.243)	19.41 (13.72)	0.783 -	0.220 (0.247)	19.18 (12.94)	0.794 -	0.209 (0.258)
+ Finetune w/ MASt3R	20.97 (16.03)	0.780 -	0.199 (0.201)	20.41 (15.13)	0.781 -	0.214 (0.226)	20.00 (14.32)	0.793 -	0.207 (0.232)	19.69 (13.45)	0.803 -	0.197 (0.241)
+ Spherical Harmonics	18.04 (13.10)	0.730 -	0.254 (0.257)	18.57 (13.29)	0.752 -	0.248 (0.259)	18.50 (12.82)	0.768 -	0.236 (0.262)	18.40 (12.16)	0.781 -	0.226 (0.272)
- LPIPS Loss	19.62 (14.68)	0.763 -	0.277 (0.282)	19.65 (14.37)	0.776 -	0.261 (0.278)	19.41 (13.73)	0.787 -	0.245 (0.278)	19.22 (12.98)	0.797 -	0.230 (0.285)
- Offsets	19.38 (14.44)	0.757 -	0.249 (0.252)	19.25 (13.97)	0.775 -	0.242 (0.256)	19.14 (13.46)	0.792 -	0.225 (0.253)	19.09 (12.85)	0.805 -	0.209 (0.255)
- Loss Masking	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 2. **Ablations on the ScanNet++ dataset.** When trained without loss masking, the memory requirements of rendering grow until training cannot continue.

Method	Pose Est.	Encoding
Ours	-	0.268
MASt3R (Point Cloud)	-	0.263
PixelSplat (w/ MASt3R poses)	10.72	0.156

Table 3. **Average time in seconds** required for position estimation (if relevant) and scene prediction.

that location. In only a few instances, such as the example in the third column, is there a significant misalignment between our rendered image and the ground truth image.

In Fig. 5, we attempt to generalize from our model, trained on ScanNet++, to real world data captured by a mobile phone. By only training our Gaussian head, we maintain MASt3R's ability to generalize to different scenes, such as the outdoor scene in the top left of the figure. Our predicted Gaussians are able to generalize from object-scale scenes up to large outdoor environments. We make a particular note of the bottom row of Fig. 5, where we show examples of reconstructing a scene from two images with little or no direct pixel correspondences, due to being taken directly side-by-side or from opposite sides of the same object. Traditional multi-view stereo systems based on image correspondences would fail in these scenarios, however MASt3R's data-driven approach allows these scenes to be reconstructed accurately.

Runtime comparisons. Next, we benchmark the time taken to reconstruct poses and perform scene reconstruction using each of these methods. Our method, and MASt3R, do not need to perform any explicit pose estimation, as all points and Gaussians are directly predicted in the same coordinate space. We see that our method can reconstruct scenes at ~4 FPS on an RTX2080ti at 512x512 resolution. Because pixelSplat needs to use MASt3R and perform explicit point cloud alignment to estimate the poses of the images, our total runtime is significantly less than the time taken to estimate the poses for pixelSplat.

4.3. Ablation studies

In Tab. 2, we run ablations on our method. We find that finetuning our MASt3R's 3D point predictions to ScanNet++ improves testing performance on ScanNet++, but we omit this finetuning from our other experiments for fair comparison with MASt3R. When training with spherical harmonics (with a degree of 4) instead of constant color Gaussians, we find that our performance decreases, likely due to overfitting spherical harmonics to our collection of training scenes. Like other works, we find that using an LPIPS loss term meaningfully increases the visual quality of the reconstructions. Our introduced offsets slightly improve performance across all metrics as well. Finally, if we omit our loss masking strategy, we find that the size of the Gaussians grows in an unbounded manner, until the memory cost of rendering the Gaussians causes training to halt.

5. Conclusion

We present Splatt3R, a feed-forward generalizable model for generating 3D Gaussian Splats from uncalibrated stereo images, without relying on camera intrinsics, extrinsics, or depth information. We find that simply using the MASt3R architecture to predict 3D Gaussian parameters, in combination with a loss-masking strategy during training, allows us to accurately reconstruct both 3D appearance and geometry from wide baselines. As we demonstrate in our experiments, Splatt3R outperforms both MASt3R and the current state-of-the-art in feed-forward splatting.

References

[1] Edward H Adelson and James R Bergen. *The plenoptic function and the elements of early vision*. MIT Press, 1991. 2

[2] Stephen T Barnard and Martin A Fischler. Computational stereo. *ACM Computing Surveys (CSUR)*, 1982. 3

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[4] Jia-Wang Bian, Wenjing Bian, Victor Adrian Prisacariu, and Philip Torr. Porf: Pose residual field for accurate neural surface reconstruction. In *ICLR*, 2023. 3

[5] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 2023. 3

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 3

[7] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 2, 3, 4, 5, 6

[8] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*, 2021. 2

[9] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *CVPR*, 2023. 3

[10] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *ECCV*, 2024. 2, 3, 4, 5, 6

[11] Cheng Chi, Qingjie Wang, Tianyu Hao, Peng Guo, and Xin Yang. Feature-level collaboration: Joint unsupervised learning of optical flow, stereo depth and camera motion. In *CVPR*, 2021. 3

[12] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *CVPR*, 2021. 2

[13] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *CVPR*, 2014. 3

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 3

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkorei, and Neil Houlsy. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 5

[16] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 2, 3, 5

[17] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 3

[18] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 3

[19] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Computer Graphics and Interactive Techniques*, 1996. 2

[20] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 3

[21] Richard Hartley and Frederik Schaffalitzky. L/sub/spl infin//minimization in geometric reconstruction problems. In *CVPR*, 2004. 3

[22] Richard I Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 1997.

[23] Richard I Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *CVPR*, 1992. 3

[24] Hiroshi Ishikawa and Davi Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *ECCV*, 1998. 3

[25] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *ICCV*, 2021. 3

[26] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. In *ICLR*, 2023. 3

[27] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *CVPR*, 2022. 2

[28] Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano, and Masaya Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *CVPR*, 1996. 3

[29] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 1, 2, 3, 4

[30] Haechan Lee, Wonjoon Jin, Seung-Hwan Baek, and Sunghyun Cho. Generalizable novel-view synthesis using a stereo camera. *CVPR*, 2024. 3

[31] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 2, 3, 4, 6

[32] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 2

[33] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. *ECCV*, 2024. 2, 3

[34] Yaokun Li, Chao Gou, and Guang Tan. Taming uncertainty in sparse-view generalizable nerf via indirect diffusion guidance. *arXiv preprint arXiv:2402.01217*, 2024. 3

[35] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 3

[36] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Fast generalizable gaussian splatting reconstruction from multi-view stereo. *ECCV*, 2024. 3

[37] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping

Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, 2022. 3

[38] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022. 3

[39] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3

[40] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *IJCV*, 1996. 3

[41] 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 *ECCV*, 2020. 1, 2

[42] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *SIGGRAPH*, 2022. 2

[43] Zhangkai Ni, Peiqi Yang, Wenhan Yang, Hanli Wang, Lin Ma, and Sam Kwong. Colnerf: Collaboration for generalizable sparse input neural radiance field. In *AAAI*, 2024. 3

[44] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 3

[45] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 5

[46] 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 *ICCV*, 2021. 3

[47] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 2019. 1

[48] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 2021. 1

[49] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. In *NeurIPS*, 2023. 3

[50] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *ECCV*, 2022. 3

[51] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 2, 4, 6

[52] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 2, 3, 4, 5

[53] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 1998. 3

[54] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023. 3

[55] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *CVPR*, 2021. 3

[56] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2, 3

[57] Shuzhe Wang, Vincent Leroy, Yohan Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 3, 4

[58] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

[59] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *ECCV*, 2024. 3

[60] Oliver J Woodford and Edward Rosten. Large scale photometric bundle adjustment. In *BMVC*, 2020. 3

[61] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 6

[62] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 3

[63] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 3

[64] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015. 3

[65] 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 *CVPR*, 2018. 3

[66] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, 2019. 3

[67] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 1995. 3

[68] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *CVPR*, 2024. 2, 3

[69] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 3