Splat and Replace: 3D Reconstruction with Repetitive Elements

NICOLÁS VIOLANTE, Inria & Université Côte d'Azur, France and Adobe, USA
ANDREAS MEULEMAN, Inria & Université Côte d'Azur, France
ALBAN GAUTHIER, Inria & Université Côte d'Azur, France
FREDO DURAND, MIT, USA
THIBAULT GROUEIX, Adobe, USA
GEORGE DRETTAKIS, Inria & Université Côte d'Azur, France

Fig. 1. Our method improves 3D reconstruction in unseen views, by leveraging the multi-view information contained in repetitive elements (the two windows in this example). From left to right, we compare Nerfbusters [Warburg et al. 2023], Bayes Rays [Goli et al. 2024], an improved version of 3D Gaussian Splatting [Kerbl et al. 2023] described in Section 5.2, our method, and the ground truth novel test view of a real scene.

We leverage repetitive elements in 3D scenes to improve novel view synthesis. Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) have greatly improved novel view synthesis but renderings of unseen and occluded parts remain low-quality if the training views are not exhaustive enough. Our key observation is that our environment is often full of repetitive elements. We propose to leverage those repetitions to improve the reconstruction of low-quality parts of the scene due to poor coverage and occlusions. We propose a method that segments each repeated instance in a 3DGS reconstruction, registers them together, and allows information to be shared among instances. Our method improves the geometry while also accounting for appearance variations across instances. We demonstrate our method on a variety of synthetic and real scenes with typical repetitive elements, leading to a substantial improvement in the quality of novel view synthesis.

CCS Concepts: • **Computing methodologies** → **Rasterization**; **Point-based models**; **Image segmentation**; **Matching**; **Shape inference**.

Additional Key Words and Phrases: novel view synthesis, radiance fields, 3D Gaussians Splatting, 3D segmentation, 3D matching, repetitions

ACM Reference Format:
Nicolás Violante, Andreas Meuleman, Alban Gauthier, Fredo Durand, Thibault Groueix, and George Drettakis. 2025. Splat and Replace: 3D Reconstruction with Repetitive Elements. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3721238.3730727

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

1 INTRODUCTION

Capturing real scenes from photos and allowing 3D navigation has become widely accessible thanks to progress in novel view synthesis using Neural Radiance Fields (NeRF) [Barron et al. 2022; Mildenhall et al. 2020; Müller et al. 2022] and 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023]. However, good quality novel view synthesis requires careful and exhaustive capture of multiple views of a scene, and the visual quality of renderings from unseen or occluded regions is typically very low. We focus on scenes that exhibit *repetitions*: repetitions can be found everywhere, from urban scenes (pillars), building facades (windows) and furniture (tables and chairs) to decorative elements. We present a method that exploits the information provided by a set of *instances* of the same object in a multi-view capture to improve the visual quality of novel views.

Repetitions have received little attention in novel view synthesis. Symmetry has been used to improve 3D reconstruction [Mitra et al. 2013], and repetitions have been used for single-image reconstruction [Cheng et al. 2023]. Rodriguez et al. [2018] presented a solution to improve novel view synthesis for the restricted case of instances in a plane. In addition, previous methods do not handle the inevitable differences in appearance between instances. In contrast, we aim for a more general solution where repetitive elements can be any 3D object, and appearance variations are handled. This involves three important challenges. First, we need high-quality 3D segmentation of the instances. Second, we need high-quality 3D registration between instances. Finally, each instance usually presents differences in illumination and appearance that need to be accounted for.

Given a base 3DGS reconstruction, our goal is to segment repetitive objects in the scene, i.e. *instances*, and fuse them together into a common and improved *shared representation*.

To address the above challenges, we first exploit learning-based 2D masks, and a single user click per instance, and then segment each instance in 3D. For this we follow previous work that uses contrastive learning [Cen et al. 2024; Choi et al. 2024, 2025; Gu et al. 2024; Kim et al. 2024; Ye et al. 2023; Ying et al. 2024] but show that additional regularization and postprocessing are required to have cleanly segmented 3D instances from a 3DGS reconstruction.

We next find per-instance rigid transformations to bring them all to a common coordinate frame. Standard 3D point-cloud matching methods are ill-suited to 3DGS data, which is noisy and does not correspond well to an underlying geometry because it optimizes image radiance rather than geometry. To overcome this problem we exploit the ability of 3DGS to render additional high-quality views, enabling the use of robust 2D image matching algorithms, which we lift to 3D using depth provided by 3DGS.

Finally, we build a shared representation of a template for all the instances by taking the union of all the 3D Gaussian primitives. A fine-tuning step allows gradients to flow from each instance to the shared representation, improving the overall reconstruction. In particular, partially-occluded instances are completed based on the visible instances. In addition low-quality geometry and appearance caused by poor coverage in the capture are enhanced with information from better-covered instances. We model differences in appearance between instances using an offset representation for Spherical Harmonics (SH); we also show that this simple solution is faster than more complex options such as a multi-layer perceptron (MLP).

In summary, our contributions are:

- A shared representation and a fine-tuning process that improves overall reconstruction and novel view synthesis quality by using information from all instances.
- An improved 3D segmentation based on constrastive learning, using additional regularization and post-processing.
- Introducing the use of novel view synthesis to enable robust 2D matching, which then allows 3D registration of the noisy 3D Gaussians of each primitive.

We evaluate our method on synthetic and real data with independent test trajectories unseen during reconstruction. Our method improves the overall quality of poorly captured or occluded regions in scenes with repetitive elements, outperforming other generalization solutions.

2 RELATED WORK

We build on novel view synthesis and in particular 3D Gaussian Splatting (3DGS), segmentation, 3D registration, symmetries and repetitions, and multi-illumination multi-view capture.

Novel view synthesis and Generalization. Our method strives to improve the quality of novel views by exploiting repetitions. Neural Radiance Fields (NeRF) [Barron et al. 2022; Mildenhall et al. 2020; Müller et al. 2022] revolutionized novel view synthesis, permitting high quality rendering of novel views, albeit at a high computation cost for scene optimization and slow rendering. More recently 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] has permitted much faster optimization and rendering, resulting in widespread adoption (see surveys [Chen and Wang 2024; Fei et al. 2024]).

Several methods attempt to improve novel view synthesis generalization by introducing *priors* [Warburg et al. 2023], or quantifying uncertainty [Goli et al. 2024]. Diffusion models have been used as priors, typically for single or few-view use cases [Gao* et al. 2024; Wu et al. 2024]. For 3DGS, LongLRM [Ziwen et al. 2024] achieves impressive reconstruction and novel view synthesis quality with as few as 32 views, but with significant resource requirements (80Gb GPU). We restrict generalization to repetitive elements, sidestepping the need for expensive diffusion models.

Radiance Field Segmentation. The lack of large-scale segmented 3D datasets has inspired the use of 2D vision models to segment 3D scenes. This creates the challenge of multi-view consistency, even with recent video segmentation [Ravi et al. 2024].

Various authors [Kobayashi et al. 2022; Tschernezki et al. 2022; Zhi et al. 2021] lift 2D class labels, or deep features [Caron et al. 2021; Radford et al. 2021], by augmenting the 3D representation with auxiliary feature channels and optimizing them to match the 2D signal via differentiable rendering. Segmentation is then obtained via nearest neighbors in 3D feature space [Goel et al. 2023]. Such feature distillation has been extended to 3DGS [Lee et al. 2024; Qin et al. 2023; Qiu et al. 2024; Zhou et al. 2024].

Recent approaches propose to directly distill *2D masks* [Bhalgat et al. 2023; Fan et al. 2023]. The lack of association between masks from different views is resolved in 3D via contrastive learning, pushing rendered features closer together when they belong to the same mask in an image and farther otherwise. Many authors [Cen et al. 2024; Choi et al. 2024, 2025; Gu et al. 2024; Kim et al. 2024; Ye et al. 2023; Ying et al. 2024] extend this idea to 3DGS segmentation. Such methods often have residual artifacts and we propose additional regularization and postprocessing that significantly improve results.

Registration. We refer the reader to a recent survey [Huang et al. 2021]. Our specific sub-problem is the registration of two clouds of Gaussian primitives, with *varying density*, *partial overlap*, *large* pose variation and *noise*. Because 3D Gaussians are optimized to reproduce images, they are not necessarily located on the surface; density variation occur across instances depending on their coverage and projected size in the training views. Since our Gaussians are not located on the geometric surface, 3D descriptors [Johnson and Hebert 1999; Qi et al. 2017; Rusu et al. 2009; Wang and Solomon 2019] would yield poor features.

We will exploit the ability of 3DGS to render realistics images to enable the use of 2D image matching. We also use the ability of 3DGS to provide depth to lift the matches to 3D. We build on recent work on 3D reconstruction and 2D matching, in particular DUSt3R [Wang et al. 2023] and MASt3R [Leroy et al. 2024]. Due to its robustness under extreme pose variations, we adopt MASt3R and demonstrate its usefulness in building a registration pipeline for 3DGS, alongside more traditional tools such as PnP-RANSAC [Fischler and Bolles 1981; Lepetit et al. 2009].

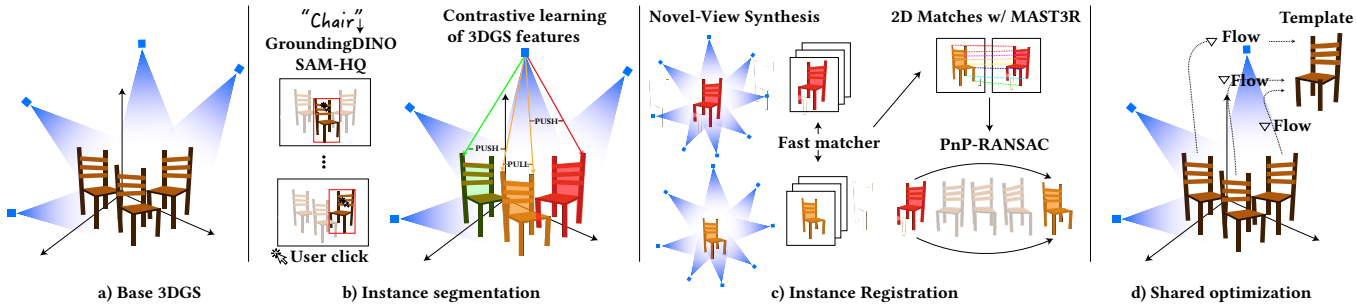

Fig. 2. **Overview of our method. (a)** We start with a base 3DGS reconstruction, then **(b)** use SAM-HQ masks and a user click to identify repetitive instances, and train contrastive features to perform instance segmentation in 3D. **(c)** To register the instances, we first render *additional* views using 3DGS for each instance. After selecting the best pairs of views with a fast 2D matcher, we find robust 2D matches using MASt3R, which we lift to 3D with the depth obtained from 3DGS. Finally, we use a PnP-RANSAC solver to register the instances given the 2D-3D correspondences. **(d)** A finetuning step for our shared representation follows, allowing gradients to flow from the instances to the template. Once our method is complete, we can replace the instances with the optimized template, with significant improvement in visual quality.

Symmetry, self-similarities, repetitions. The concept of symmetry and repetitions has received steady attention in Computer Graphics, with some approaches focusing on detecting symmetry [Je et al. 2024; Mitra et al. 2006, 2013] and others on exploiting it for downstream applications. In this paper, we offload the problem of symmetry detection to the user, who indicates repeating elements with a few clicks, and focus on how to best *exploit* symmetry to improve novel-view synthesis. Previous work has shown the benefits of using symmetry for model compression [Zheng et al. 2010], scan consolidation [Mitra et al. 2006], symmetric triangulation [Podolak et al. 2007] and model completion [Thrun and Wegbreit 2005]. In the latter, the authors propose to use partial symmetries to reconstruct missing regions in 3D scans. Repetitions have been used to compute structure and perform inverse rendering for a single image [Cheng et al. 2023], including in a generative context [Zhang et al. 2023]. Rodriguez et al. [2018] use instances to improve reconstruction as we do, but are restricted to a planar configuration, and only test window instances on a facade. Similarly, we fill in missing details in one instance by borrowing from the other repetitions for 3DGS reconstructions. Naively copying and pasting Gaussians would lead to artifacts because each repetition is observed under different illumination conditions. We thus tackle a multi-view multi-illumination reconstruction problem: geometry and surface properties are shared among instances, but illumination conditions vary.

Multi-view multi-illumination reconstruction. Most novel view synthesis methods assume that the scene has been captured under a single lighting condition. There have been several attempts to model changing conditions, for example NeRF-W [Martin-Brualla et al. 2021] that encodes varying appearance in an additional feature vector for NeRF. Specific methods have been developed to capture scenes under multiple illumination conditions, typically using specialized "dome-like" equipment [Debevec et al. 2000], but also low-end cellphone capture with a flash [Bi et al. 2020]. Recently, 3DGS was extended for relighting [Poirier-Ginter et al. 2024], using a diffusion-model based prior to augment a single illumination capture to multi-illumination.

In our case, each instance can be seen as the same object illuminated under a different lighting configuration; however, the lighting variations are often less severe, since all instances are in the same overall scene. This allows us to represent appearance variation with spherical harmonics offsets.

3 OVERVIEW

We take as input a multi-view image dataset of a scene that contains multiple instances of the same element. Our method (Figure 2) consists of three main stages. First, we perform 3D instance segmentation (Sec. 4.1), with an extended version of 3DGS, where an extra feature per Gaussian is trained using contrastive learning. After segmentation, a user manually selects which instances should be grouped together by simply clicking on them. This requires one click per instance in only one image. Second, we propose a method to merge instances into a single coordinate system (Sec. 4.2). To deal with the geometric ambiguity of 3DGS, we introduce a matching procedure that exploits the novel view synthesis capabilities of 3DGS to generate additional viewpoints. These new views enable robust 2D matching algorithms, which we exploit to do 2D-3D correspondences and 3D registration. Third, we train a shared representation for all instances using common geometry but individual appearance parameters (Sec. 4.3). Each instance is seen from different viewpoints in the input images used during optimization. Our representation allows gradients to flow from each instance to the template using the rendering loss from each such viewpoint. This improves the overall reconstruction quality for each instance despite occlusions or lack of detail in different view of each instance used in training.

4 METHOD

4.1 3D Instance Segmentation

Segmenting objects in 3D in a 3DGS scene is challenging. 3D Gaussian primitives do not strictly model object geometry: sometimes they are placed in space around (but not on) a surface, or are large with low opacity, which allows 3DGS to model specific appearance in the input photos. Inspired by previous work [Bhalgat et al. 2023; Fan

et al. 2023], we handle this task using contrastive learning [Radford et al. 2021] with 2D masks, assigning a feature to each Gaussian.

To segment each training view, the user provides a text query such as "*pillar*" to GroundingDINO [Liu et al. 2024], which outputs a bounding box per instance, fed to SAM-HQ [Ke et al. 2023], which in turn outputs a 2D mask per instance. We call "*background*" the complement of the union of the masks in the training view, i.e. the union of all pixels *not* containing the text query. We train using a cosine distance for the pull and push losses. The push loss includes a margin of 0.3. We add extra loss terms for the hard cases (distance between features larger than 0.5). The pull loss is weighted by the ratio of negative-to-positive pairs to counter the imbalance due to more negative pairs.

During contrastive learning, the trainable per-Gaussian features are rasterized into per-pixel features. To train the features, pairs of pixels are sampled in a given input image and undergo the following contrastive loss. If the pixels belong to the same 2D mask, their features are pulled together. If they belong to different masks, their features are pushed apart.

However, we observe three main challenges in applying contrastive learning to 3DGS: 1) low-opacity Gaussians under the surface of objects do not receive enough gradients because they are occluded by other Gaussians, 2) large Gaussians that are shared among different objects get conflicting gradients and 3) several Gaussian primitives, typically at the border between an instance and the background are "left behind", resulting in visual artifacts and harming the subsequent optimization (Sec. 4.3).

Regularization and Optimization. To address the first two challenges, we add ℓ_1 regularization terms on the opacity and the scaling during 3DGS training to discourage both phenomena

$$\lambda_{\text{opacity}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{o}_i + \lambda_{\text{scale}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i, \tag{1}$$

where N is the number of Gaussian primitives.

After 3DGS training is complete, we train the contrastive features. Contrastive methods are sensitive to the number of pixels sampled per mask to form positive and negative pairs. Therefore, we uniformly sample M_u pixels in the image, so larger masks get more samples, and M_s pixels per mask so even small masks get enough samples to form pairs (we set $M_u = M_s = 4096$).

We include a regularization term to encourage the rendered features $\hat{\mathbf{f}}$ to be unit norm, similar to [Cen et al. 2024].

Interactive segmentation. Once the features are trained, different instances can be segmented based on the similarity of their features. For each instance q, to form a query feature $\mathbf{f}_q$, the user clicks inside the mask of the instance in any training view, and we select the rendered feature corresponding to the clicked pixel. We perform the 3D segmentation by finding the closest Gaussians in the scene with a simple threshold in the contrastive feature space. Thus, the segmented instance $\mathcal{I}_q$ for query $\mathbf{f}_q$ is simply the set of Gaussians with contrastive features $\mathbf{f}_i$ that verify $d(\mathbf{f}_i, \mathbf{f}_q) \leq \tau$. We set $\tau = 0.1$ in our experiments. Since this only requires one click per instance from the user, this interaction is fast and low-effort (please see supplementary video). At the end of this stage, we have a set of instances $\mathcal{I}_1, \ldots \mathcal{I}_M$ formed by their corresponding Gaussian primitives for each object of interest. We compute corresponding 2D masks $\{\mathcal{M}_{q,j} | 1 \leq q \leq M, 1 \leq j \leq N, \}$ of the q-th instance in the j-th training view by comparing the rendered features in the training viewpoints with $\mathbf{f}_q$. The resulting masks are better than the ones produces by SAM-HQ because they aggregate information from all views [Siddiqui et al. 2023].

Post-processing. This process can leave some residual Gaussians out of the segmented instances contributing to the appearance of the instance and the background Fig. 6 (left). It is important to remove them from the reconstruction of the background otherwise each instance will get slightly different gradients, hurting the sharing of information amongst them. We filter them out with a space carving approach. To decide if a Gaussian primitive should be removed from the background, we check if at least one of the following three conditions is met in *every* train image. First, the center projects inside the 2D masks $\{\mathcal{M}_{q,j}\}$ of the instances. Second, the center projects outside the image boundary. Third, the center is not visible in the current camera (this includes, for example, primitives behind the camera). If at least one of the conditions is met for all training views, the Gaussian affects the instance's rendering and is removed from the scene.

In Fig. 6, we show how the regularization in Eq. 1 and the post-processing step significantly improve overall visual quality.

4.2 Novel View Synthesis for Instance Registration

We now have M segmented instances, which we need to register into a common coordinate system with a per-instance rigid transformation, i.e. a rotation R and a translation T. We choose the instance that has the largest number of Gaussian primitives as the template instance $\mathcal{I}_T$.

We have found purely geometric registrations inapplicable because 3D Gaussians are optimized to reproduce the appearance of the scene but not necessarily its geometry. In particular, several Gaussians *behind* the surface often contribute to the final appearance, and the instances have different spatial density of Gaussian primitives, e.g. very detailed regions have much smaller Gaussians than flat regions. Naive 3D point cloud registration does not work very well, as we show in Figure 7.

On the other hand, Gaussian primitives represent a radiance field that allows high quality novel view synthesis, which allows us to leverage 2D image matching. These methods can enable reliable instance registration in 3D if provided with suitable views of the instances. We use novel view synthesis to render such views. The estimation of the pose of a reference object from an image typically has two steps: a coarse estimation stage, and a refinement step [Nguyen et al. 2024].

Our registration process for coarse pose estimation has four steps: 1) Rendering of a dense set of views of each instance from all directions 2) Fast matching of all pairs of all views to find a reduced set of k pairs between each instance $\mathcal{I}_i$ and the template $\mathcal{I}_T$, 3) Dense matching of k pairs of views of $\mathcal{I}_i$ and $\mathcal{I}_T$ to find the best such

pair, 4) Lifting the 2D matches of the chosen pair to 3D and using a PnP-RANSAC solver to find the transformation of $\mathcal{I}_i$ to $\mathcal{I}_T$.

Rendering dense set of views for each instance. We sample 25 cameras pointing towards the center, on a virtual sphere around each instance, by regularly sampling 5 azimuths and 5 elevations. Since we have a base 3DGS representation of the scene, we use these cameras to generate novel views of each instance.

Fast matching to choose k pairs of views. Next, we quickly find candidate pairs of views where the source instances $\mathcal{I}_i$ and target (template) $\mathcal{I}_T$ are seen from similar points of view. For this, we use a fast matcher [Potje et al. 2024], and select the top k pair of source-target views with the most matches as candidates for the next stage. We set k to 10 in our experiments.

Dense matching to find the best pair. Once we have the top k pairs of views, we find *dense* 2D matches with MASt3R [Leroy et al. 2024], a transformer-based solution more robust than fast matching but of higher computational cost, which makes it prohibitively expensive to run directly on all pairs. We denote these matches $P_{2D} = \{x_1 \dots x_N\}$ and $Q_{2D} = \{y_1 \dots y_N\}$. In pixel coordinates, the points x_i and y_i are of the form (u_i, v_i).

Lifting to 3D and Perspective-n-Point solver. We backproject these 2D points using the depth D_i obtained from 3DGS, (following [Kerbl et al. 2024]), the intrinsics matrix $\mathbf{K}$ and extrinsics $\mathbf{M}_{\text{cam}}$. We obtain the corresponding 3D point matches $P_{3D} = \{p_1 \dots p_N\}$ and $Q_{3D} = \{q_1 \dots q_N\}$ expressed in world coordinates:

$$p_i = D_i \mathbf{M}_{\text{cam}} \mathbf{K}^{-1} (u_i \; v_i \; 1)^T \tag{2}$$

Back-projected depth tends to be closer to the geometric surface than the Gaussian primitives themselves because depth is computed as a weighted average of the Gaussian centers, some of which are in front of the surface and some of which are behind.

Given one of the k pairs of views, we find the rigid transformation between the source and the target Gaussians using Perspective-n-Point (PnP) on the 2D-3D correspondences between the source 3D points P_{3D} and the target 2D keypoints Q_{2D}. The PnP-RANSAC solver outputs the camera-to-world transformation corresponding to the camera that sees the source 3D points P_{3D} from the target camera viewpoint. We run this robust PnP-RANSAC solver for each pair and keep the transformation with the largest number of inliers.

Our pose refinement consist of two steps: first we run ICP [Zhang 1994] on the Gaussian centers, initialized with the coarse initialization of the pose, then we further refine the pose by adding its parameters to the subsequent 3DGS optimization.

For symmetric objects (e.g. the pawns in the Chessboard scene), the matching procedure finds correspondences based on appearance. For both symmetric and textureless objects, multiple rigid transformations are valid depending on the type of symmetry. In this case, finding one of them during registration is enough to finetune the shared representation.

4.3 Shared Representation and Optimization

Now that we have the transformations of all the instances into a single template space, we can create a shared representation for the repetitive scene elements. Our goal for this representation is twofold. First, we want a geometry representation that shares the information provided by each instance, thus significantly improving the 3DGS reconstruction quality for all instances. In particular, we want to complete instances with occluded parts using the information from the visible instances, and improve low-quality regions by propagating fine-grain details from high-quality parts. Second, we want our representation to handle the differences in appearance mainly due to different illumination.

Once we have all the transformations of instances $\mathcal{I}_i$ to the template $\mathcal{I}_T$, we create the initial shared representation by placing all the instances in a common coordinate system and taking their union.

For each Gaussian primitive, all parameters are shared across instances of the same object, effectively constraining the geometry of all instances with multi-view information from all instances. To handle the differences in appearance, we decompose the SH coefficients of each primitive into a shared component and an offset term for each instance:

$$\mathbf{c}^{\ell m} = \lambda \mathbf{c}^{\ell m}_{\text{shared}} + (1 - \lambda) \mathbf{c}^{\ell m}_{\text{offset}} \tag{3}$$

where ℓ is the degree, m the order of the coefficient, and λ a mixing weight set to 0.8. The offset parameters $\mathbf{c}^{\ell m}_{\text{offset}}$ represent the individual differences between instances, typically due to illumination, and we encourage them to be small with an ℓ_1 penalty.

Optimization. We replace each instance with a *reference* to the shared representation that undergoes the per-instance geometric transform and SH offset. As a result, we can propagate gradients from all instances to the shared representation. We optimize the shared geometry and SHs with gradient descent. For example, if $\mathcal{I}_1$ and $\mathcal{I}_2$ are seen from opposite sides in template space, the shared representation will benefit from the information each instance provides and be well reconstructed from both sides. We jointly optimize over the pose to refine its estimation from the previous step.

5 DATA & EVALUATION

We implemented our method building on the original codebase of 3DGS [Kerbl et al. 2023]. Our source code and data are available at https://repo-sam.inria.fr/nerphys/splat-and-replace.

5.1 Synthetic & Real Scenes

We evaluate our method on four synthetic scenes and four real scenes, the former allowing more precise quantitative evaluation. For each scene, we optimize the reconstruction over the training sequence and evaluate over a held-out test sequence. For synthetic scene, Office, Temple, Chessboard and Classroom, we render 200 images for training and 50 for testing. Our test views are far from the training views, and are not used in training. The image size is 1370×912, rendered using Blender's Cycles [Community 2018], along with ground truth masks and depth maps, that we use in our quantitative evaluation. We use the following repetitive elements for each scene: 3 tables and 9 chairs for Office, 13 columns for Temple, 5 desks for Classroom, 2 rooks, 2 knights, 2 bishops, and 8 pawns for Chessboard.

For each real scene, House, MeetingRoom, Pillars and Facade, we run COLMAP [Schonberger and Frahm 2016] on both the train and test sequences together to obtain a coherent calibration, then

Table 1. **Quantitative evaluation.** We report average PSNR, SSIM, LPIPS, and KID across test views on our synthetic and real dataset. We compute the metrics both in the complete images and in the masked regions where replacement of instances took place to isolate the effect of our method. For easier readability, we color-code results as a linear gradient between worst and best

	Synthetic				Synthetic (masked)				Real				Real (masked)			
	PSNR↑	SSIM↑	LPIPS ↓	KID↓	PSNR↑	SSIM↑	LPIPS ↓	KID↓	PSNR↑	SSIM↑	LPIPS ↓	KID↓	PSNR↑	SSIM↑	LPIPS ↓	KID↓
Nerfbusters	19.48	0.690	0.464	0.4185	17.97	0.605	0.219	0.2098	19.26	0.757	0.422	0.3017	17.62	0.687	0.138	0.1827
BayesRays	19.50	0.655	0.418	0.3018	18.20	0.573	0.223	0.2242	22.05	0.779	0.385	0.1215	21.97	0.743	0.123	0.0837
Nerfacto	20.18	0.683	0.405	0.2288	19.37	0.607	0.215	0.1694	22.43	0.803	0.368	0.1303	21.54	0.771	0.114	0.0765
3DGS	23.37	0.800	0.266	0.1808	22.78	0.761	0.139	0.1607	22.57	0.849	0.273	0.0974	21.87	0.798	0.089	0.0766
3DGS*	24.41	0.843	0.236	0.1389	23.02	0.778	0.119	0.1076	22.59	0.842	0.273	0.0994	21.91	0.799	0.090	0.0749
Ours	27.62	0.897	0.163	0.0316	27.54	0.887	0.063	0.0097	24.18	0.868	0.248	0.0483	24.63	0.847	0.068	0.0348

exclude the SfM points for which the test views were used for triangulation from the 3DGS initialization. We compute monocular depth using Depth-Anything-v2 [Yang et al. 2024] for depth supervision, and the segmentation masks using GroundingDINO [Liu et al. 2024] and SAM-HQ [Ke et al. 2023]. We use the following repetitive elements for each scene: 2 windows for House, 4 windows and 4 railings for Facade, 6 chairs and 3 tables for MeetingRoom, and 4 pillars for Pillars.

We also include one scene from ScanNet++ [Yeshwanth et al. 2023] and two scenes from DL3DV [Ling et al. 2024] with repetitive elements (chairs).

5.2 Evaluation

Baselines. We compare with the vanilla 3DGS [Kerbl et al. 2023], and an improved version, denoted 3DGS*, using monocular depth regularization, exposure adjustment from [Kerbl et al. 2024] and our proposed opacity and scale regularization for the segmentation (see Sec. 4.1). We additionally compare with Nerfbusters [Warburg et al. 2023] and Bayes' Rays [Goli et al. 2024], which both aim at better generalization by measuring uncertainty in novel views and using it to remove floaters. Finally, Nerfacto [Tancik et al. 2023] is a strong NeRF-based baseline that trades off speed and quality, and is also the underlying model that Nerfbusters and Bayes' Rays use.

Qualitative Evaluation. We show different unseen test views in Fig. 4 for real scenes, Fig. 3 for synthetic scenes, and Fig 8 for ScanNet++/DL3DV. Our reconstruction, based on a shared representation for repetitive elements, shows better overall appearance for the repetitive objects. For instance, in Office scene of Fig. 3, the chairs and table have severe occlusion in the training views, leading to poor reconstruction by the baselines. In contrast, our shared representation incorporates the information from visible instances to improve the reconstruction.

Zooming In. Our method can also render detailed zoomed-in novel view of objects that are seen from afar in the training sequence, if one of its repetition is seen from up close. We show this in Fig. 5, where the reconstruction of the bust in the background is improved using the bust in the front. Our method removes artifacts and adds the fine-grain details that were missing in the 3DGS* reconstruction, due to this instance being captured from far away.

Quantitative Evaluation. Our quantitative results are summarized in Table 1, on the synthetic and real scenes. We report the standard reconstruction error metrics PSNR and SSIM [Wang et al. 2004], and the perceptual metric LPIPS [Zhang et al. 2018]. In addition, we compute KID [Bińkowski et al. 2021] to compare the distribution of rendered and ground truth views, which is typically used to evaluate the overall realism of rendered images. We compute metrics on the full rendered test views, as well as on the masked regions corresponding to repetitions to isolate the impact of our method on the repetitive elements. Our visual improvement translates directly to the quantitative results, where we improve on all metrics by a large margin. Most importantly, on the real scenes, our approach improves by 1.59 dB the PSNR compared to the second-best performing method and 2.72 dB if only considering the masked region with repetitions. For the ScanNet++/DL3DV scenes we obtain an average improvement of 1.28 dB in PSNR in the masked regions.

5.3 Analysis

Segmentation. In Fig. 6, we qualitatively ablate our opacity and scale regularization (see Eq. 1) and our space carving post-processing. As shown on the left, without the opacity and scale regularization, many Gaussian primitives fail to meet the contrastive features threshold, and remain where the object was. Our regularization reduces the number of trailing Gaussians (middle) but some remain at the border of the object. We hypothesize that they do not pass the contrastive features threshold because they can contribute both to the appearance of the object and to the background, being at the border. Our space carving approach successfully removes them (right), achieving a clean segmentation.

Registration. We compare our registration procedure with a global registration approach based on 3D features, FPFH [Rusu et al. 2009] and a RANSAC procedure to register the two point clouds based on matching the features. Fig. 7 showcases the limitations of this approach in the context of 3DGS, where the point cloud data is very noisy and not concentrated only on surfaces. By replacing each instance with the union of its registered repetitions, we observe that the registered instances for the chair and tables are not aligned. This is confirmed by our quantitative analysis in Table 3, where we

Fig. 3. **Qualitative evaluation on synthetic scenes.** Each column corresponds to a different scene (Temple, Classroom, Chessboard, and Office), and each row shows results from different methods: Nerfbusters [Warburg et al. 2023], Bayes' Rays [Goli et al. 2024], 3DGS* [Kerbl et al. 2023], Ours, and the Ground Truth.

measure the angular error of the alignment on the synthetic scenes. Our method averages a half-degree error, while [Rusu et al. 2009] fails with 50 degrees of error. We also ablate the main components of our registration approach. Using the train views instead of sampling new views in our virtual sphere degrades the performance by 4 degrees. Indeed, the train views cannot ensure finding a pair of views from a similar perspective and roughly from the same distance, which makes 2D matching harder for MASt3R. We also ablate the refinement step showing that performances degrades by 5 degrees without ICP. We also ablate the dense matching with MASt3R using SIFT [Lowe 2004] and SuperPoint [DeTone et al. 2018] as alternatives. SuperPoint failed in the Temple scene, we report the average on the other synthetic scenes. On real scenes, SuperPoint also failed in Pillars, while SIFT failed in Facade and MeetingRoom. These failures emphasize the need for a robust matcher like MASt3R.

Shared Representation. A key element of the shared representation is its ability to adapt to the specific appearance of each instance via the SH offsets while aggregating information with a shared SH component. We report our ablations in Table 2 on synthetic scenes. If no SH offsets are used, the shared representation cannot adapt to each instance and learns a suboptimal appearance. On the other hand, when no shared SHs are used, the representation does not

Fig. 4. **Qualitative evaluation on real scenes.** Each column corresponds to a different scene (MeetingRoom, Pillars, and Facade), and each row shows results from different methods: Nerfbusters [Warburg et al. 2023], Bayes' Rays [Goli et al. 2024], 3DGS* [Kerbl et al. 2023], Ours, and the Ground Truth.

Fig. 5. **Zoom-in.** Our method improves the reconstruction of the bust in the background, allowing for a successful close-up shot (*middle and right*). This happens because a repetition of the bust is seen up-close in the training views, benefiting the shared representation (*left*).

Fig. 6. **Segmentation ablation.** Without opacity and scale regularization or post-processing (*left*) more gaussians are left over. Using the regularization (without post-processing) shows a reduced number of remaining gaussians (*middle*). Our complete solution with both regularization and post-processing produces the cleanest results (*right)* .

Fig. 7. **Registration comparison.** We compare our registration scheme (*right*) with FPFH [Rusu et al. 2009] on the Gaussian primitives (*left*). We replace each instance by the union of all its registered repetitions. Note that the table and chair both have two modes for their pose for FPFH, while with our registration they are aligned. This is an intermediary visualization of our method, after which the shared representation is optimized with gradients flowing from all the repetitions.

aggregate the information from all instances effectively, resulting in degrading quality, and lower PSNR.

We also experimented with a coordinate-based MLP shared across instances, that takes as input a Gaussian primitive position in the shared template, the rigid transformation of the instance, the viewing direction and the SHs in the base 3DGS reconstruction for this primitive, and outputs the RGB color of the gaussian primitive. We observed similar quality but found it more than one and a half times slower. Additionally, in this representation, the reflectance's shared component and the instance-specific components are not disentangled.

Computational Cost. Our method extends a base 3DGS representation, requiring additional training time for contrastive features and shared representation finetuning, similar to prior 3DGS extensions. We provide a time breakdown for each step with our default setup: (i) 3DGS: 22min, (ii) Contrastive features: 21min, (iii) 3D Segmentation: 19sec, (iv) Registration: 123sec, (v) Finetuning: 51min.

Table 2. **Shared representation ablation.** We report average PSNR across test views on two synthetic scenes. We also include average training iterations per second.

	PSNR ↑	it/s ↑
SHs	27.33	**2.6**
w/o offset	26.63	**2.6**
w/o shared	25.19	**2.6**
MLP	**27.34**	1.5

Table 3. **Registration comparison and ablation.** On the synthetic dataset, we report mean absolute error (MAE) for the predicted rotation **R** (geodesic distance in degrees) and the translation vector **t**. † Both ablations fail in some real scenes. SIFT also fails in Temple, which we exclude to compute the metrics.

	MAE(**R**) ↓	MAE(**t**) ↓
Ours	0.490	0.065
w/o Virtual view	4.367	0.204
w/o ICP	5.455	0.114
w/ SIFT†	0.379	0.059
w/ SuperPoint†	0.458	0.075
FPFH [Rusu et al. 2009]	50.58	2.547
w/o ICP	66.94	2.606

5.4 Limitations & Future Work

A natural limitation of our method is given by the number of repetitions in the scene and the variability they show. Our method benefits the most when several objects are seen from different points of view: this provides a stronger signal for our shared representation. However, our method cannot improve the background, and some artifacts remain in the test views as a result. Also, the identification of the repetitive elements in a scene require user interaction.

Our method is limited by strong changes in illumination in the scene (for example, strong highlights in specular surfaces), which produce pronounced differences in appearance among instances of the same object. Scenes with uniform illumination are easier to handle with our method. Strong differences in appearance are still challenging to model and we leave their specialized treatment for future work.

As discussed in Sec 5.3, our method requires additional training for the contrastive features and the shared representation. Contrastive features can be trained in 4 min (1k iterations) instead of 21 min (5k), with minimal performance difference (0.969 vs. 0.966 mAcc, 0.965 vs. 0.963 mIoU). Furthermore, we can reduce finetuning iterations from 7k to 4k (29min) and keep the performance (PSNR=27.62@7k vs PSNR=27.63@4k in synthetic scenes). Recent advances like Taming 3DGS [Mallick et al. 2024] should reduce the cost of contrastive training and finetuning.

An interesting direction for future work is to leverage the shared representation to reduce the memory requirements of 3DGS: if N instances can be represented with a single and compact shared representation, the total number of Gaussians required for all instances is reduced. Here, the major challenge is how to encode the appearance of each instance in a compact representation.

Another promising direction is improving inverse rendering with our shared representation. A direct extension would be to also share

Fig. 8. **Qualitative evaluation on real scenes from ScanNet++ and DL3DV** Each column corresponds to a different scene (fist column from ScanNet++ [Yeshwanth et al. 2023], second and third from DL3DV [Ling et al. 2024]), and each row shows results from different methods: 3DGS* [Kerbl et al. 2023], Ours, and the Ground Truth.

the same material parameters of a BRDF. The shared representation would receive multi-illumination information for all instances, helping material and illumination disentanglement.

6 CONCLUSIONS

We show that repetitions in 3D scenes can be leveraged to improve reconstruction and novel view synthesis. After an initial 3DGS reconstruction, our method detects and fuses the multiple occurrences of a given object into a shared representation with common geometry and base appearance, while individual appearance is modeled as an offset for each instance. Then, each instance is replaced with this representation. Our key insight is that this shared representation and the offsets can be jointly optimized using all information available in the scene for that object, which improves the geometry and appearance for all instances.

ACKNOWLEDGMENTS

This work was funded by the European Research Council (ERC) Advanced Grant NERPHYS, number 101141721 https://project.inria.fr/nerphys/. The authors are grateful to the OPAL infrastructure of the Université Côte d'Azur for providing resources and support, as well as Adobe and NVIDIA for software and hardware donations. The authors thank the anonymous reviewers for their valuable feedback.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR* (2022).

Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. 2023. Contrastive Lift: 3D Object Instance Segmentation by Slow-Fast Contrastive Fusion. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=bbbbbov4Xu

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 294–311.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2021. Demystifying MMD GANs. http://arxiv.org/abs/1801.01401 arXiv:1801.01401 [cs, stat].

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. 2024. Segment Any 3D Gaussians. https://doi.org/10.48550/arXiv.2312.00860 arXiv:2312.00860.

Guikun Chen and Wenguan Wang. 2024. A Survey on 3D Gaussian Splatting. arXiv:2401.03890 [cs.CV] https://arxiv.org/abs/2401.03890

Tianhang Cheng, Wei-Chiu Ma, Kaiyu Guan, Antonio Torralba, and Shenlong Wang. 2023. Structure from Duplicates: Neural Inverse Graphics from a Pile of Objects. https://openreview.net/forum?id=7irm2VJARb

Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. 2024. Click-Gaussian: Interactive Segmentation to Any 3D Gaussians. In *ECCV*.

Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. 2025. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*. Springer, 289–305.

Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 145–156.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 337–33712. https://doi.org/10.1109/CVPRW.2018.00060 ISSN: 2160-7516.

Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. 2023. NeRF-SOS: Any-View Self-supervised Object Segmentation on Complex Scenes. In *The Eleventh International Conference on Learning Representations*.

Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 2024. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics* (2024).

Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Commun. ACM*, Vol. 24. Association for Computing Machinery, New York, NY, USA, 381–395. https://doi.org/10.1145/358669.358692

Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. 2024. CAT3D: Create Anything in 3D with Multi-View Diffusion Models. *Advances in Neural Information Processing Systems* (2024).

Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P.J. Narayanan. 2023. Interactive Segmentation of Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. 2024. Bayes' Rays: Uncertainty Quantification in Neural Radiance Fields. *CVPR* (2024).

Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. 2024. EgoLifter: Open-world 3D Segmentation for Egocentric Perception. http://arxiv.org/abs/2403.18118 arXiv:2403.18118 [cs].

Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. 2021. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690* (2021).

Jihyeon Je, Jiayi Liu, Guandao Yang, Boyang Deng, Shengqu Cai, Gordon Wetzstein, Or Litany, and Leonidas Guibas. 2024. Robust Symmetry Detection via Riemannian Langevin Dynamics. https://doi.org/10.1145/3680528.3687682 arXiv:2410.02786 [cs].

Andrew E Johnson and Martial Hebert. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on pattern analysis and machine intelligence* 21, 5, 433–449.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. 2023. Segment Anything in High Quality. In *NeurIPS*.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4. https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. 2024. A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets. In *ACM Transactions on Graphics*, Vol. 43. https://repo-sam.inria.fr/fungraph/hierarchical-3d-gaussians/

Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. 2024. GARField: Group Anything with Radiance Fields. https://doi.org/10.48550/arXiv.2401.09419 arXiv:2401.09419 [cs].

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, Vol. 35. 23311–23330.

Hyunjee Lee, Youngsik Yun, Jeongmin Bae, Seoha Kim, and Youngjung Uh. 2024. Rethinking Open-Vocabulary Segmentation of Radiance Fields in 3D Space. http://arxiv.org/abs/2408.07416 arXiv:2408.07416 [cs].

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. EPnP: An accurate O(n) solution to the PnP problem. In *International Journal of Computer Vision*, Vol. 81. https://doi.org/10.1007/s11263-008-0152-6

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. 2024. Grounding Image Matching in 3D with MASt3R. http://arxiv.org/abs/2406.09756 arXiv:2406.09756 [cs].

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. 2024. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22160–22169.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*.

David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (Nov. 2004), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. 2024. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. 2006. Partial and approximate symmetry detection for 3D geometry. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 560–568. https://doi.org/10.1145/1179352.1141924

Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. 2013. Symmetry in 3D Geometry: Extraction and Applications. *Computer Graphics Forum* 32, 6 (2013), 1–23. https://doi.org/10.1111/cgf.12010 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12010.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* (2022).

Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. 2024. GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9903–9913.

Joshua Podolak, Aleksey Golovinskiy, and Szymon Rusinkiewicz. 2007. Symmetry-enhanced remeshing of surfaces. In *Proceedings of the fifth Eurographics symposium on Geometry processing*. 235–242.

Yohan Poirier-Ginter, Alban Gauthier, Julien Phillip, J-F Lalonde, and George Drettakis. 2024. A Diffusion Approach to Radiance Field Relighting using Multi-Illumination Synthesis. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e15147.

Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento. 2024. XFeat: Accelerated Features for Lightweight Image Matching. 2682–2691. https://openaccess.thecvf.com/content/CVPR2024/html/Potje_XFeat_Accelerated_Features_for_Lightweight_Image_Matching_CVPR_2024_paper.html

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA, USA.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. 2023. LangSplat: 3D Language Gaussian Splatting. https://doi.org/10.48550/arXiv.2312.16084 arXiv:2312.16084 [cs].

Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. 2024. Language-Driven Physics-Based Scene Synthesis and Editing via Feature Splatting. In *European Conference on Computer Vision (ECCV)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714* (2024). https://arxiv.org/abs/2408.00714

Simon Rodriguez, Adrien Bousseau, Fredo Durand, and George Drettakis. 2018. Exploiting Repetitions for Image-Based Rendering of Facades. *Computer Graphics Forum* 37, 4 (2018), 119–131. https://doi.org/10.1111/cgf.13480 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13480.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. 2009. Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*. 3212–3217. https://doi.org/10.1109/ROBOT.2009.5152473

Johannes L. Schonberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 4104–4113. https://doi.org/10.1109/CVPR.2016.445

Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. 2023. Panoptic Lifting for 3D Scene Understanding with Neural Fields. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Vancouver, BC, Canada, 9043–9052. https://doi.org/10.1109/CVPR52729.2023.00873

Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. 2023. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–12.

S. Thrun and B. Wegbreit. 2005. Shape from symmetry. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. 1824–1831 Vol. 2. https://doi.org/10.1109/ICCV.2005.221

Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. 2022. Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representations. In *2022 International Conference on 3D Vision (3DV)*. 443–453. https://doi.org/10.1109/3DV57658.2022.00056

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. 2023. DUSt3R: Geometric 3D Vision Made Easy. https://doi.org/10.48550/arXiv.2312.14132 arXiv:2312.14132 [cs].

Yue Wang and Justin M Solomon. 2019. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3523–3532.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4, 600–612. https://doi.org/10.1109/TIP.2003.819861

Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. 2023. Nerfbusters: Removing Ghostly Artifacts from Casually Captured NeRFs. http://arxiv.org/abs/2304.10532 arXiv:2304.10532 [cs].

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. 2024. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21551–21561.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything V2. In *NeurIPS*.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. 2023. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. https://doi.org/10.48550/arXiv.2312.00732 arXiv:2312.00732 [cs].

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. 2023. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12–22.

Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. 2024. OmniSeg3D: Omniversal 3D Segmentation via Hierarchical Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 586–595. https://doi.org/10.1109/CVPR.2018.00068

Yunzhi Zhang, Shangzhe Wu, Noah Snavely, and Jiajun Wu. 2023. Seeing a rose in five thousand ways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 962–971.

Zhengyou Zhang. 1994. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision* 13, 2 (1994), 119–152.

Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J Mitra, Daniel Cohen-Or, and Baoquan Chen. 2010. Non-local scan consolidation for 3D urban scenes. In *ACM Trans. Graph.*, Vol. 29. 94–1.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. 2021. In-Place Scene Labelling and Understanding With Implicit Scene Representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 15838–15847.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. 2024. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21676–21685.

Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. 2024. Long-LRM: Long-sequence Large Reconstruction Model for Wide-coverage Gaussian Splats. *arXiv preprint 2410.12781* (2024).