ASC Final Report: Studying the Influence of Initialisations on Gaussian Splatting for Novel View Synthesis

Daniel Wang
u7918232

Figure 1. **Influence of initialisations on Treehill (top) and Bonsai (bottom) scenes**. (a) SfM point cloud (b) fused point cloud from depth maps (c) Renders optimised from SfM (d) Renders optimised from fused depth map point cloud (e) zoomed-out view of optimised scene from SfM initialisation (f) zoomed-out view of optimised scene from fused point cloud. Renders from held-out test views are similar for both initisation methods whereas the geometry of background regions is heavily influenced by the initial point cloud.

Abstract

3D Gaussian Splatting has emerged as a popular 3D novel view synthesis method due to its high efficiency and rendering quality. 3DGS primarily relies on sparse Structure From Motion point clouds as initialisation which is insufficient to cover entire scenes, leading to inaccurate geometry in background regions. This project explores an alternative method of initialising 3D Gaussian scenes by leveraging monocular depth estimation models. Specifically, scales for estimated depth maps are disambiguated using a differentiable image warping procedure and then scaled depth maps are used to unproject images into a point cloud for initialisation. Experiments show that this initialisation method achieves slightly lower performance on rendering quality metrics compared to the original SfM approach while also creating better geometry in background regions.

1. Introduction

Novel view synthesis is an important problem in computer vision that has seen rapid progress with the introduction of 3D Gaussian Splatting (3DGS) [8] due to a significant increase in performance while maintaining high rendering quality compared to previous state-of-the-art NeRFs [13]. Despite near-photorealistic results, rendering quality drastically degrades when the scene is observed from viewpoints that are far from training views. This is because without sufficient prior geometric information as is typically the case with Structure from Motion initialisation, supervision from photometric loss alone is unable to enforce correct geometry in background regions of scenes. 3DGS typically obtains camera poses from COLMAP [15] which produces a point cloud that can conveniently be used to initialise 3D gaussians. However, the initial point cloud is sparse and contains points that are extracted from matching features across multiple images, resulting in severe under-reconstruction in background or low-texture regions. It is therefore desirable to leverage geometric information in order to initialise scenes with dense, accurate point clouds that cover all visible regions. Recent 3DGS works have largely focused on utilising geometric information in the optimisation stage rather than initialisation, where a popular technique is to use depth priors to regularise scene geometry. This project explores how initialising scenes by fusing depth maps into a dense point cloud affects the rendering quality and geometry of the final optimised scene.

Monocular depth estimation has also recently undergone significant advancements by following the large-scale pre-training then fine tuning paradigm [7, 14, 17, 18], relying on vision foundation models fine tuned on the downstream task

of monocular depth estimation. Due to the generalisation capabilities of large foundation models, monocular depth estimation models trained in this way perform extremely well in a diverse range of settings and have been successfully applied to 3DGS pipelines. However, the inherent scale ambiguity in monocular depth estimation constrains its usefulness in 3DGS where geometric consistency across multiple views is required. In order to navigate this issue, numerous works [2, 3, 9, 11, 12, 16, 19, 21] have proposed scale-invariant losses as well as techniques to disambiguate the unknown scale. In this project, depth maps obtained from an off-the-shelf monocular depth estimation model are disambiguated using the image warp procedure introduced by Zhou et al. [20], from which a point cloud can be fused for 3DGS initialisation.

In summary, this project explores an alternative initialisation to SfM points by fusing disambiguated depth maps into a point cloud. Experiments are conducted to evaluate the effect of this initialisation on rendering quality and scene geometry in comparison to the original 3DGS initialisation method.

2. Related Work

2.1. 3D Gaussian Splatting

Kerbl et al. [8] introduced 3D Gaussian Splatting, a novel view synthesis method where 3D scenes are represented as a set of 3D gaussians. Given a set of images along with camera poses, 3DGS optimises the parameters of gaussians via SGD under an image reconstruction loss. Each gaussian is parameterised by mean μ, opacity o, positive semi-definite covariance matrix $\Sigma = RSS^TR^T$ with scale S and rotation R computed from a unit Quaternion q, and spherical harmonics to represent colour. 3D gaussians are projected to 2D screen space with the colour of a pixel x_p computed by α-blending N gaussians overlapping x_p,

$$\mathcal{C}(x_p) = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{1}$$

where c_i is the colour of the ith gaussian and α_i is obtained by evaluating the projected 2D gaussian at x_p multiplied by opacity o_i. This is done by an efficient differentiable gaussian rasterizer which enables low training times and high framerates.

At the start of optimisation, a set of 3D gaussians is initialised from the sparse point cloud obtained from SfM. However, sparse SfM point clouds do not contain enough gaussians to cover entire scenes. Scenes are typically initialised with around 100K-200K SfM points and densified to millions of points at the end of optimisation. This is done through a crucial part of the optimisation process, termed *adaptive density control*, which densifies gaussians in under-reconstructed regions and removes them in over-reconstructed regions according to opacity and positional gradient magnitudes. However, colour supervision alone is insufficient to accurately reconstruct scenes as geometry in background regions is oftern poor despite high rendering quality from viewpoints close to ones used in training. The authors experimented with initialising scenes with a random set of gaussians and observed a decrease in quality, showing that accurate initialisation of gaussians using SfM point clouds leads to better results. Intuitively, initialising a larger number of gaussians at correct positions will improve the optimised scene.

2.2. Depth Supervision for 3D Gaussian Splatting

Depth supervision is a common technique employed by recent 3DGS and NeRF based works to incorporate additional geometric information, particularly in sparse-view 3D reconstruction. In this setting where only a small number of training images are available, colour supervision alone leads to overfitting of training viewpoints, poor geometry and with scene appearance deteriorating when observed away from training viewpoints. Numerous works have proposed to solve his problem by leveraging estimated monocular depth maps for regularisation to improve scene geometry [2, 11, 12, 16, 21]. A common approach is to render depth maps by swapping out colour features with the depth with respect to the camera origin, where various techniques have been proposed to mitigate the scale ambiguity issue in order to effectively utilise monocular depth maps for regularisation.

Without disambiguating the depth scale, [3, 21] regularise using the Pearson correlation between the two depth maps while [12] uses a scale and shift invariant loss. Alternatively, [2, 9] disambiguate the scale and shift by aligning the monocular depth with SfM points and [16, 19] estimate the scale and shift with a least-squares method, and then minimise the L1/L2 norms between the scaled and rendered depth maps.

It is desirable to disambiguate the depth scale as it provides stronger regularisation during optimisation and in the case of this project enables multiple depth maps to be fused into a single point cloud for initialisation. Among the aforementioned works that disambiguate depth scales, the sparsity of SfM points and error in least squares estimation cause inaccuracies in the disambiguated scale. This project explores an alternative method for disambiguating depth scales by reformulating the image warp loss procedure described in the following section.

2.3. Monocular Depth Estimation

Monocular depth estimation suffers from the fundamental problem of scale ambiguity which compromises the usefulness of monocular depth estimation in tasks such

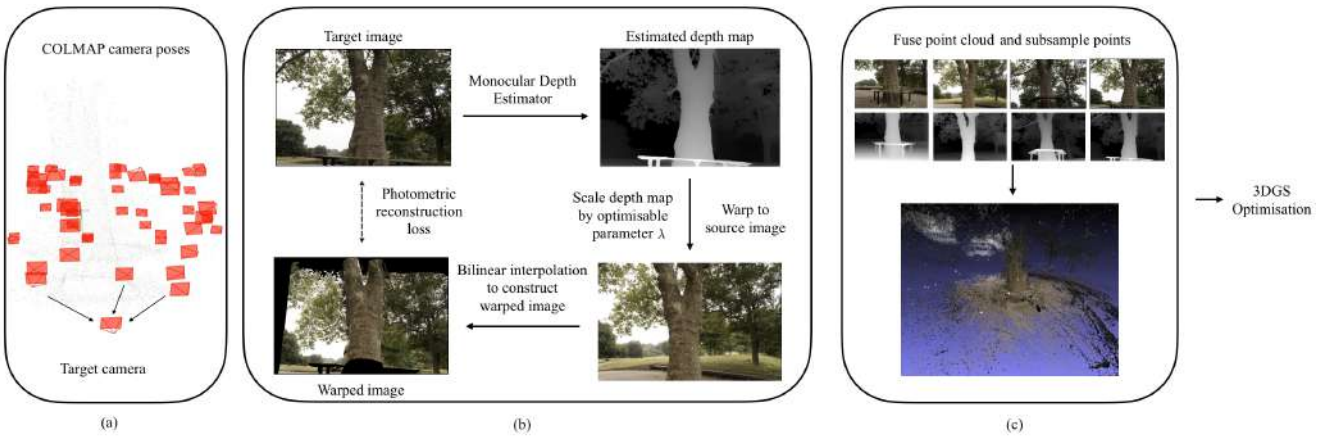


Figure 2. **Overview of initialisation method.** The method consists of three stages. a) First, k nearby source cameras are selected for every camera. b) Then, a scale for each depth map is optimised using the differentiable image warp process. c) Finally, the scaled depth maps are fused and a subsampling procedure obtains a point cloud for 3DGS initialisation.

as 3DGS that require consistent geometry across multiple views. Zhou et al. [20] introduced an unsupervised framework that relies on monocular video sequences to simultaneously train motion and monocular depth prediction networks. Given a sequence of images, a target image I_t and adjacent source images I_s are selected. The objective is to reconstruct I_t with a warped image $\hat{I}_s$ that is obtained by projecting pixels in I_t to I_s and sampling the corresponding colours from I_s. Specifically, the depth and pose networks predict a depth map $\hat{D}_t$ corresponding to I_t and relative pose $\hat{T}_{t\to s}$ respectively. Then, pixels p_t in I_t represented by homogeneous coordinates can be mapped to homogeneous coordinates p_s in I_s by

$$p_s = K_s \hat{T}_{t\to s} \hat{D}_t(p_t) K_t^{-1} p_t \tag{2}$$

where K denotes the intrinsic matrix. For each pixel p_t in the warped image $\hat{I}_s$, the colour $\hat{I}_s(p_t)$ at p_t is obtained by bilinear sampling from $I_s(p_s)$. Supervision for the depth and pose networks is given by the image reconstruction loss between the warped image and target image

$$\mathcal{L} = \sum_s \sum_p |I_t(p) - \hat{I}_s(p)| \tag{3}$$

where s and p index over source images and pixel coordinates respectively. For this procedure to be effective, it is essential that the scene appearance is consistent across multiple views which requires that the scene is static, Lambertian and that there is no occlusion between views. This method also struggles with low-texture regions, as small changes in the estimated position of the projected pixel will have minimal influence on the sampled colour so the gradient will not be meaningful. Under this framework, the scale ambiguity issue still persists due to the unknown relative pose. Rewriting the relative pose $\hat{T}_{t\to s}$ as a rotation $\hat{R}$ and translation $\hat{T}$, we have

$$p_s = K_s(\hat{R}\hat{D}_t(p_t)K_t^{-1}p_t + \hat{T}) \tag{4}$$

where scaling both sides by $\lambda \in \mathbb{R}$ gives

$$\lambda p_s = K_s(\hat{R}\lambda\hat{D}_t(p_t)K_t^{-1}p_t + \lambda\hat{T}) \tag{5}$$

. That is, scaling both the predicted depth and translation vector by an arbitrary scale λ gives the same warped point in homogeneous coordinates. However, with fixed camera poses this will not be an issue, and the image warp procedure can be reformulated to disambiguate the scale λ given an estimated depth map.

3. Method

Scenes are initialised by first estimating a depth map for each image using an off-the-shelf monocular depth estimation model. Then, each depth map is scaled by optimising a scalar using the image warping procedure. Finally, a point cloud is fused from the scaled depth maps and used as initialisation for Gaussian Splatting. The method is illustrated in Figure 2.

3.1. Depth Scale Optimisation

The key component of this method is the optimisation of depth scales corresponding to each depth map which is done by reformulating the differentiable image warp procedure described in Section 2.3. Starting from known camera poses given by COLMAP [15], a set of k closest *source* cameras in terms of position and viewing angle need to be selected for each *target* camera. This is done by adding the normalised absolute value difference between the target and source translation vectors with the normalised absolute value difference between the target and source rotation matrices, and taking the k source cameras corresponding to the k lowest values.

Given intrinsics K_t, K_s and extrinsics $[R|t]_t, [R|t]_s$ for cameras corresponding to target and source images I_t and I_s, estimated depth map $\hat{D}$ and homogeneous coordinates p_t of a target image pixel, Eq. 2 can be rewritten as

$$p_s = K_s[R|t]_s{[R|t]_t}^{-1}\lambda\hat{D}_t(p_t)K_t^{-1}p_t \tag{6}$$

where λ is the unknown scale. As before, the colour $\hat{I}_s(p_t)$ at p_t in the warped image $\hat{I}_s$ is obtained by bilinear interpolation from $I_s(p_s)$. If there are points in the target image that cannot be projected to the source image because they are unprojected to a location outside the view frustum of the source camera, there will be blank pixels in the warped image. The correct scale λ is the one that mimimises the reconstruction error

$$\mathcal{L} = \frac{1}{\text{sum}(M)}\sum_s\sum_p |M \odot (I_t(p) - \hat{I}_s(p)| \tag{7}$$

where M is a binary mask that selects non-blank pixels in the warped image and s and p index over source images and pixel coordinates respectively. The $\frac{1}{\text{sum}(M)}$ factor is crucial to average the loss over non-blank pixels because otherwise the loss can be trivially minimised by setting λ small enough such that none of the unprojected target points lie within the view frustum of the source camera, resulting in a blank image with zero loss. However, even when the loss is averaged over non-blank pixels, there are some cases where the image warp loss is minimised when the depth scales are very low and the warped image contains very few pixels. An example of this is shown in Figure 3. These cases are often caused by a high proportion of low-texture regions where photometric loss does not reliably indicate geometric accuracy.

It is desirable to obtain a good initialisation for λ in order to speed up optimisation and to avoid the scenario where the loss is minimised at an incorrect scale. Experiments found the optimised λ be between 10 and 20 for almost all images in every scene when using the monocular depth model Depth Anything V2-Small. Therefore, λ is initialised by linearly sampling values from 10 to 20 in increments of 2 and selecting the value that gives the lowest loss. Furthermore, an initial value for λ is only selected if at least 50% of the warped image is non-blank. Then, λ is optimised with gradient descent to obtain the final scale.

3.2. Point Cloud Fusion

With the optimised scale, points from each input image can now unprojected using the scaled depth map. Depending on the image resolution and number of cameras, this can result in a total of tens or hundreds of million points, whereas the SfM point clouds used typically in 3DGS typically only contain one to two hundred thousand points with final converged scenes containing up to millions points for

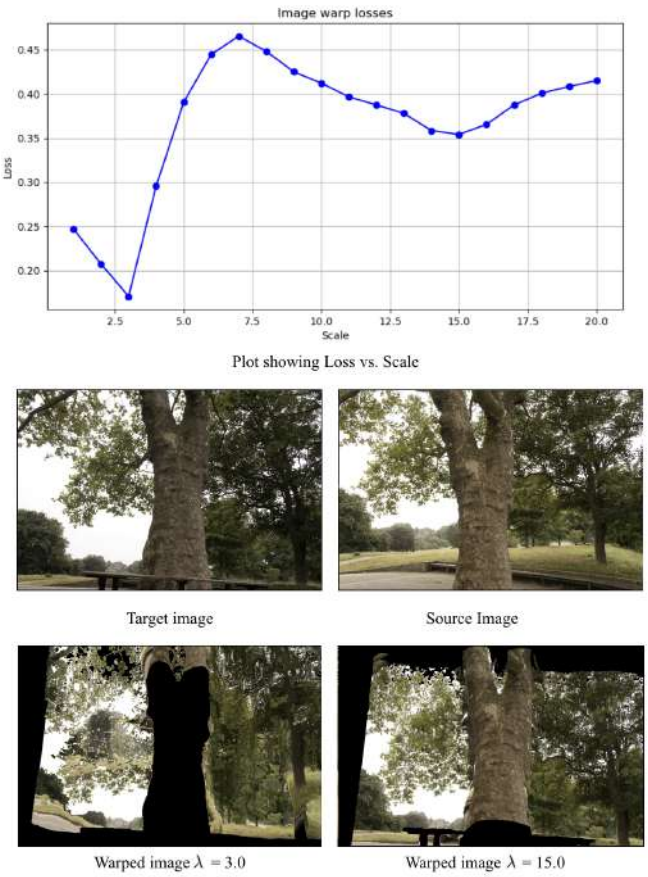
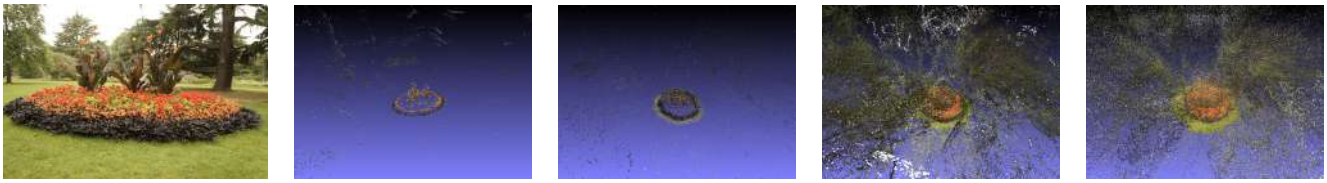


Figure 3. **Incorrect scale**. An example of the image warp loss being minimised by an incorrect scale due to the low-texture grass.

the scenes used in this project. To strike a balance between sufficiently covering all areas of the scene and avoiding having too many unnecessary points that will slow down the optimisation process, a subset of N points need to be subsampled from the fused point cloud where N is a hyperparameter. A simple approach is to randomly sample N points from the fused point cloud as initialisation for the scene. However, this fused point cloud is very noisy due to inaccuracies in the estimated depth and scale. To balance between sparse SfM point cloud initialisation and the millions of gaussians in converged scenes, N is set to 1 million for experiments.

Warp-sampling. The amount of noise can be slightly reduced by filtering points using the image warp loss again with a camera not used during optimisation. Specifically, for each target camera, the points are projected to the $k+1$th nearest source camera with the scaled depth map and the image reconstruction loss is evaluated pixel-wise. Among the non-blank pixels, the N/(number of cameras) pixels with the lowest loss are selected to be fused into the point cloud. This technique will be referred to as *warp-*



GT image | SfM point cloud | 100K Warp-sampled pts | 1M warp-sampled pts | 1M random sampled pts

Figure 4. Visualisations of different initialisation methods in MeshLab.

SSIM	SfM		Random Sample 1M		Warp Sample 100K		Warp Sample 1M	
	7K	30K	7K	30K	7K	30K	7K	30K
Bicycle	0.793	**0.842**	0.794	0.84	0.752	0.828	0.797	0.841
Bonsai	0.928	**0.952**	0.921	0.951	0.879	0.927	0.919	0.944
Counter	0.885	**0.914**	0.877	0.91	0.851	0.896	0.878	0.91
Dr Johnson	0.86	**0.898**	0.853	0.894	0.848	0.896	0.855	0.894
Flowers	0.509	0.586	0.541	**0.603**	0.48	0.568	0.544	**0.603**
Garden	0.813	**0.855**	0.763	0.833	0.729	0.822	0.754	0.829
Kitchen	0.91	**0.932**	0.898	0.928	0.871	0.916	0.883	0.926
Playroom	0.893	**0.901**	0.875	0.895	0.887	0.899	0.891	0.898
Room	0.903	**0.925**	0.898	0.924	0.872	0.906	0.893	0.92
Stump	0.717	**0.769**	0.707	0.746	0.665	0.731	0.709	0.744
Train	0.6	**0.649**	0.596	0.641	0.58	0.632	0.59	0.639
Treehill	0.596	**0.639**	0.608	0.633	0.554	0.618	0.59	0.626
Truck	0.689	**0.715**	0.683	0.705	0.667	0.699	0.682	0.707
Average	0.777	**0.814**	0.77	0.807	0.741	0.795	0.768	0.806

Table 1. Comparison of SSIM scores for different initialisation methods.

PSNR	SfM		Random Sample 1M		Warp Sample 100K		Warp Sample 1M	
	7K	30K	7K	30K	7K	30K	7K	30K
Bicycle	25.49	**26.737**	25.469	26.636	25.047	26.407	25.586	26.673
Bonsai	30.069	**33.153**	29.9	33.107	28.145	31.166	29.424	32.16
Counter	27.253	**29.094**	26.968	28.834	26.242	28.584	26.888	28.812
Drjohnson	27.08	29.096	26.76	28.831	26.752	**29.164**	27.019	29.038
Flowers	20.314	**21.367**	20.688	21.213	19.963	21.042	20.692	21.255
Garden	26.072	**27.25**	24.892	26.366	24.279	26.197	24.576	26.206
Kitchen	29.342	**31.536**	28.989	31.33	27.593	30.175	28.18	31.048
Playroom	29.365	29.854	27.906	29.645	29.187	29.887	29.37	**29.889**
Room	29.507	31.506	29.513	**31.742**	28.287	30.493	29.161	31.246
Stump	25.548	**26.633**	24.931	25.58	24.332	25.523	25.12	25.625
Train	18.026	**19.516**	17.84	19.204	17.66	19.251	17.85	19.356
Treehill	22.162	**22.538**	22.266	22.241	21.822	22.285	21.817	22.031
Truck	20.736	**21.467**	20.531	21.098	20.24	20.968	20.47	21.161
Average	25.459	**26.904**	25.127	26.602	24.581	26.242	25.089	26.5

Table 2. Comparison of PSNR scores for different initialisation methods.

sampling. A comparison of the different initialisation methods is shown in Figure 4.

Note that for outdoor scenes, image regions corresponding to the sky are unprojected very far away. While this is geometrically accurate, it is unnecessary for 3DGS scenes which occupy a relatively small volume. Hence, the pre-scaled estimated disparity map is clamped to an empirically determined minimum value of 0.4 to roughly align the sky with objects in the far background.

LPIPS	SfM		Random Sample 1M		Warp Sample 100K		Warp Sample 1M	
	7K	30K	7K	30K	7K	30K	7K	30K
Bicycle	0.202	**0.127**	0.2	**0.127**	0.254	0.143	0.197	**0.127**
Bonsai	0.21	0.175	0.215	**0.171**	0.272	0.205	0.215	0.209
Counter	0.229	0.184	0.238	0.185	0.278	0.214	0.233	**0.182**
Drjohnson	0.335	**0.247**	0.342	0.25	0.357	0.253	0.34	0.249
Flowers	0.441	0.36	0.406	**0.321**	0.467	0.38	0.4	0.324
Garden	0.186	**0.123**	0.228	0.139	0.27	0.152	0.232	0.14
Kitchen	0.149	**0.117**	0.17	0.121	0.205	0.134	0.195	0.126
Playroom	0.29	0.246	0.315	0.252	0.297	0.248	0.285	0.241
Room	0.24	**0.198**	0.249	0.199	0.286	0.234	0.257	0.207
Stump	0.327	**0.244**	0.32	0.251	0.377	0.282	0.313	0.252
Train	0.455	**0.359**	0.461	0.361	0.49	0.387	0.469	0.37
Treehill	0.423	0.338	0.396	**0.32**	0.472	0.376	0.423	0.334
Truck	0.392	0.321	0.396	**0.313**	0.423	0.343	0.397	0.313
Average	0.298	0.234	0.303	**0.232**	0.342	0.258	0.304	0.236

Table 3. Comparison of LPIPS scores for different initialisation methods.

4. Experiments

4.1. Experiment setup

Datasets. The datasets used in this project are the same datasets used in the original 3DGS paper which include scenes from the Mip-NeRF360 [1], Deep Blending [5] and Tanks&Temples [10] datasets.

Implementation details. This project was implemented in PyTorch around the original 3DGS codebase. To increase efficiency, images were downsampled by factors of 2 to a resolution of around 800x500. For each target camera, only the 2 nearest cameras were selected as source cameras as selecting more cameras slowed down optimisation without leading to a more accurate scale. Monocular depth maps were obtained using the small 25M parameter Depth Anything V2 model [18]. These settings aimed to reduce computational cost and had negligible effect on the appearance of the fused point cloud. After the scales λ are initialised, they are optimised for 100 iterations using the Adam optimiser with learning rate set to 0.01 and no learning rate decay. Note that the initialisation process does not require the CUDA-based gaussian rasterizer. With these hyperparameter settings it takes approximately 15 minutes to initialise a given scene from the above datasets on an Apple M1 Pro 14-core GPU, though it would likely take considerably less time on a GPU where the PyTorch *grid_sample* function used for bilinear interpolation is supported. The fused point

cloud only provides the 3D positions of the gaussians and their colour. The remaining features are initialised as is in the original 3DGS method, and then optimised with the default hyperparameter settings.

4.2. Results

Verifying the optimised depth scales. The datasets used do not provide ground-truth depth or point clouds. An alternative would be to treat SfM point clouds as ground truth and compute the Chamfer distance between fused points cloud in order to evaluate subsampling strategies and the correctness of the optimised depth scales. However, SfM point clouds have a far lower number points and leave the background severely under-reconstructed. It was observed that the Chamfer distance varied based on the number of subsampled points from the fused point cloud and the strategy used to sample them. Hence, this was not a reliable method for verifying the optimised scale. Instead, visualisation of fused point clouds in MeshLab (Figure 5) showed that most scenes are well-reconstructed, indicating that the scales are mostly correct although there is also a lot of noise present. Some scenes, particularly indoor ones appear to be much noisier. This is possibly due to a higher proportion of low-texture regions such as walls and other uniform surfaces where the image warp loss struggles to obtain the correct scale.

Rendering quality. To evaluate the effect of various initialisations on rendering quality, each scene was initialised from (1) SfM, (2) warp-sampling 100K points (3) warp-sampling 1M points, and (4) randomly sampling 1M points from the fused point cloud. Optimisation was done with the default hyperparameter settings in the original 3DGS. SSIM, PSNR, LPIPS metrics are reported at 7K and 30K iterations of optimisation in Tables 1, 2 and 3 with the train/test dataset split obtained by taking every 8th image following Kerbl et al. [8].

The vanilla SfM initialisation consistently performs the best on SSIM and PSNR while randomly sampled 1M points performs the best on LPIPS, although the difference between each initialisation method is small. This decrease in rendering quality is very likely due to noise in the fused point clouds. Given that initialisataion by random sampling and warp sampling produces 1M points which is around 5-10 times more points than SfM initialisations and much closer to the number of points in fully optimised scenes, one might anticipate that initialising with more points will lead to convergence in fewer iterations. However, Tables 1, 2 and 3 show that this is not the case. The difference between metrics recorded at 7K and 30K iterations is similar across all initialisation methods, suggesting that the density of gaussians at initialisation has little effect on the number of iterations is required to converge. One possible reason for this is that because gaussisan covariance scales are initialised to equal the mean of the distance to the closest three

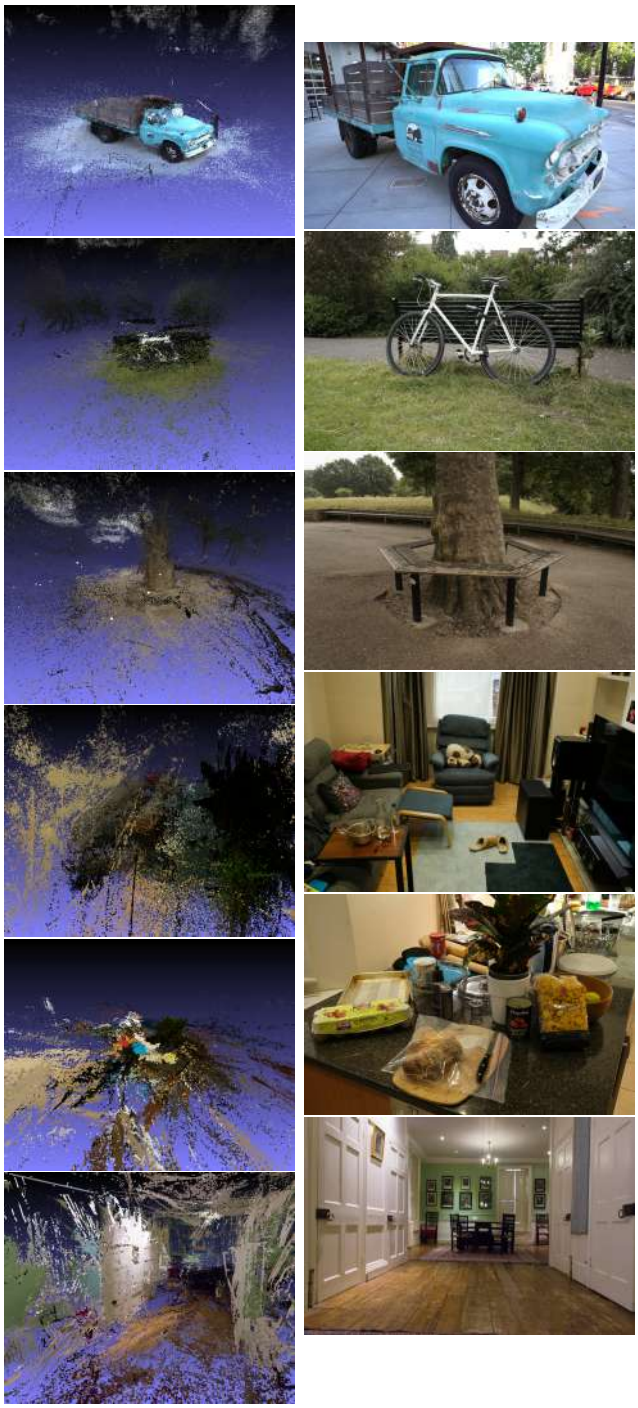


Figure 5. Left: MeshLab visualisations of fused point clouds with 1M warp-sampled points from slightly zoomed-out views. Right: images from respective datasets. Notably, outdoor scenes (top 3 rows) appear better reconstructed compared to indoor scenes (bottom 3 rows).

Images from datasets | SfM initialisation | Warp-sampled 1M points

Figure 6. Optimised scenes *Garden, Room* and *Stump* at 30K iterations. Red boxes highlight regions in the background where initialisation with 1 million Warp-sampled points achieves better reconstruction while blue boxes highlight regions closer to the centre where SfM initialisation performs better.

points, dense point clouds will cause gaussians to be smaller and thus receive less gradient signal. An intuitive solution would be to increase the gaussian scales in initialisations with dense point clouds. However, experiments found that this led to a decrease in performance again likely due to increased noise.

Scene geometry. Figure **??** shows that the influence of different initialisations is mainly seen in the background regions. With a much higher number of points initialised in the background, even despite noise and incorrect geometry it can be observed that the fused point cloud initialisation method generally achieves better reconstruction in background regions where SfM does not provide any geometry as shown in Figure 6. However, in regions closer to the centre that are modelled by SfM points, fused point cloud initialisation performs worse which can be explained by increased noise.

Depth maps from scene gaussians can be rendered by swapping out colour features for depth values during α-blending. Figure 7 shows a comparison between estimated depth maps from Depth Anything V2 and rendered depth maps from held-out test views. The rendered depth maps are noisier than Depth Anything V2 but are similar for different initialisation methods, which is to be expected given that rendering quality is comparable. When observed from viewpoints futher away, a larger discrepancy between the depth maps can be anticipated due to irregularites in background geometry but the datasets do not provide images from such viewpoints.

Rendering quality/geometry trade-off. The results of this project can be explained by a trade-off between rendering quality and geometry that has been observed in the literature. In order to introduce addtional geometric priors, *2D Gaussian Splatting* [6] models gaussians as two-

Depth Anything V2 | SfM initialisation | Warp-sampled 1M points

Figure 7. Depth renderings of optimised scenes *Treehill* and *Counter* at 30K iterations compared to the depth map obtained from Depth Anything V2. There is minimal difference between the renders from the two initialisation methods.

dimensional discs to better represent surfaces and employs depth and normal regularisation, while *SuGaR* [4] extracts a mesh and restricts gaussians to lie on the surface of the mesh. Both works report improved geometry but at the expense of slightly lower rendering quality compared to vanilla 3DGS. By restricting geometric attributes of gaussians, they become less flexible and are less able to model scene appearane effectively. Similarly, this project introduces geometric priors by initialising a large number of gaussians in background regions albeit with a considerable amount of noise, with results showing improved geometry and slightly lower rendering quality. These results indicate a trade-off between rendering quality and geometry that is difficult to navigate.

5. Limitations and future work

Intuitively, obtaining a dense, accurate noise-free point cloud would improve both rendering quality and geometry compared to SfM initialisation. The method presented in this project has not achieved this due to excessive noise caused by inaccuracies in the optimised scale, depth map and subsampling strategy. The vanilla 3DGS method expects sparse point clouds as initalisation, and as such the *Adaptive Density Control* strategy is set up to add a large number new gaussians. However, this is undesired given a fused point cloud that already contains excess incorrect geometry. Modifying the densification strategy to focus on removing noisy gaussians may lead to better results.

Furthermore, the accuracy of optimised depth scales could not be confidently verified since the datasets used do not provide ground truth depth or point clouds. Utilising synthetic datasets where this information can be obtained would allow for better evaluation of the proposed method.

6. Conclusion

This project explored an alternative method for initialising 3D Gaussian Splatting by disambiguating and fusing estimated monocular depth maps into a point cloud. Experiments showed improved geometry in background regions due to increased geometric priors but lower rendering quality due to excessive noise.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 5

[2] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. *arXiv preprint arXiv:2311.13398*, 2023. 2

[3] Congyue Deng, Chiyu "Max" Jiang, Charles R. Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, and Dragomir Anguelov. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20637–20647, June 2023. 2

[4] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 8

[5] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. 37(6):257:1–257:15, 2018. 5

[6] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 7

[7] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1, 2, 6

[9] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), July 2024. 2

[10] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 5

[11] Raja Kumar and Vanshika Vats. Few-shot novel view synthesis using depth aware 3d gaussian splatting. *arXiv preprint arXiv:2410.11080*, 2024. 2

[12] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. *arXiv preprint arXiv:2403.06912*, 2024. 2

[13] 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

[14] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 1

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

[16] Jiuhn Song, Seonghoon Park, Honggyu An, Seokju Cho, Min-Seop Kwak, Sungjin Cho, and Seungryong Kim. D\" arf: Boosting radiance fields from sparse inputs with monocular depth adaptation. *arXiv preprint arXiv:2305.19201*, 2023. 2

[17] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1

[18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 1, 5

[19] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[20] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017. 2, 3

[21] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting, 2023. 2