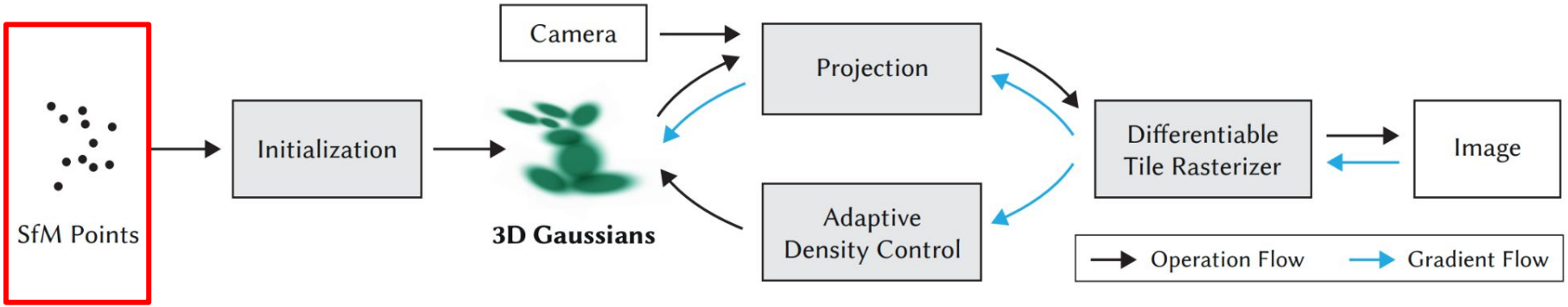
Studying the influence of initialisations on 3D Gaussian Splatting

(a) (b) (c) (d) (e) (f)

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.

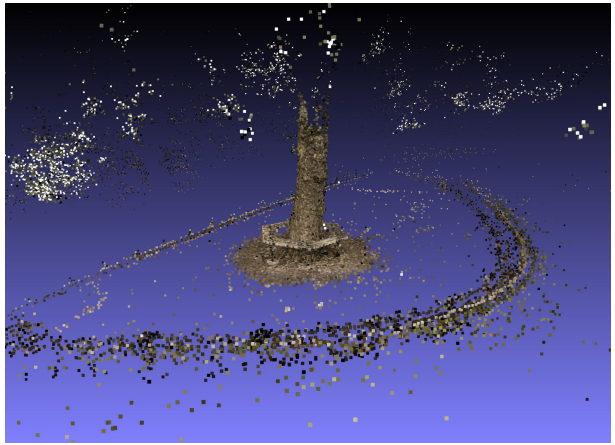
3D Gaussian Splatting



Kerbl et al. 2023

3D Gaussian Splatting - Initialisation

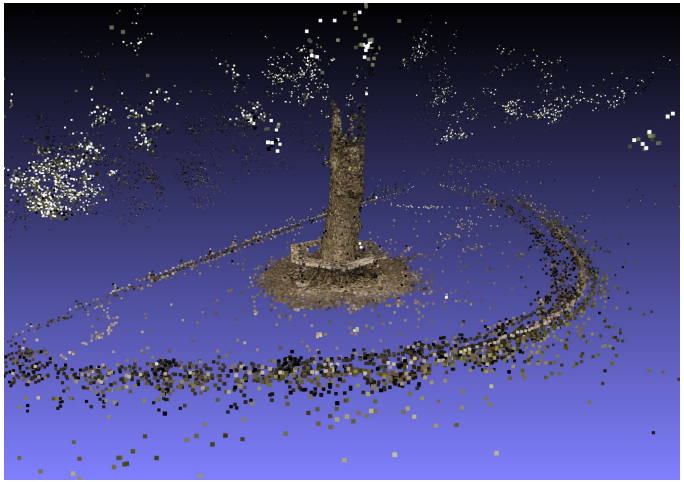
- Gaussians are initialised from sparse SfM points which largely ignore background regions
- Due to SfM point cloud sparsity, many gaussians are added (and also removed) through an *Adaptive Density Control* process
- Colour supervision alone is insufficient, and geometry deteriorates in background areas



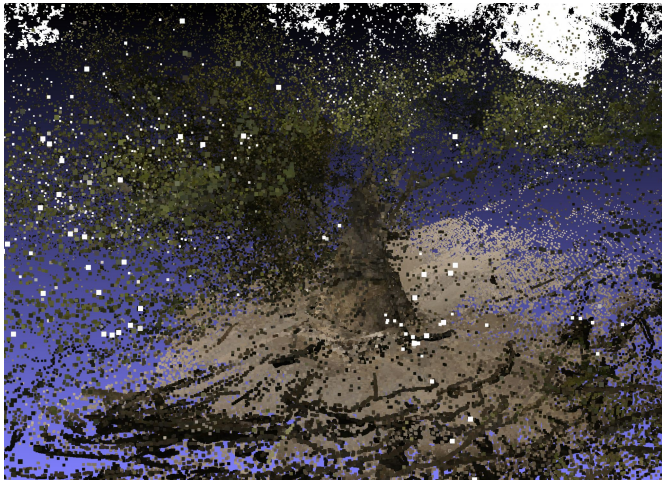
- Intuitively, initialising with a dense, accurate noise-free point cloud would improve both rendering quality and geometry compared to SfM initialisation

Can we initialise more points to provide more geometric priors?

- This can be done by **estimating a depth map for each image and fusing them into a single dense point cloud**
- This is a viable approach because monocular depth estimation models are now very powerful
- However, **scale ambiguity** remains an issue



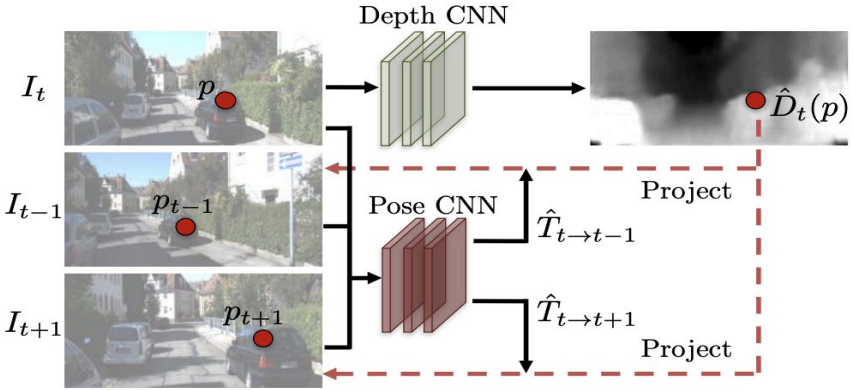
SfM point cloud with 52 363 points



Fused point cloud with 1 million points

Background: Differentiable image warping process

- In order to fuse monocular depth maps, it is necessary to first disambiguate the scale
- The method for doing this is based on *Unsupervised Learning of Depth and Ego-Motion from Video (Zhou et al., 2017)*



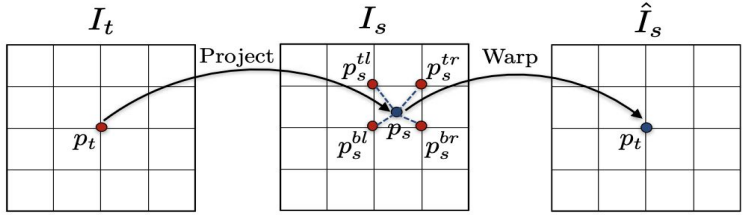


Figure 3. Illustration of the differentiable image warping process. For each point p_t in the target view, we first project it onto the source view based on the predicted depth and camera pose, and then use bilinear interpolation to obtain the value of the warped image $\hat{I}_s$ at location p_t.

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t$$

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

Zhou et al. 2017

Scale ambiguity

- Under this framework, the scale ambiguity issue still persists due to unknown camera poses:

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

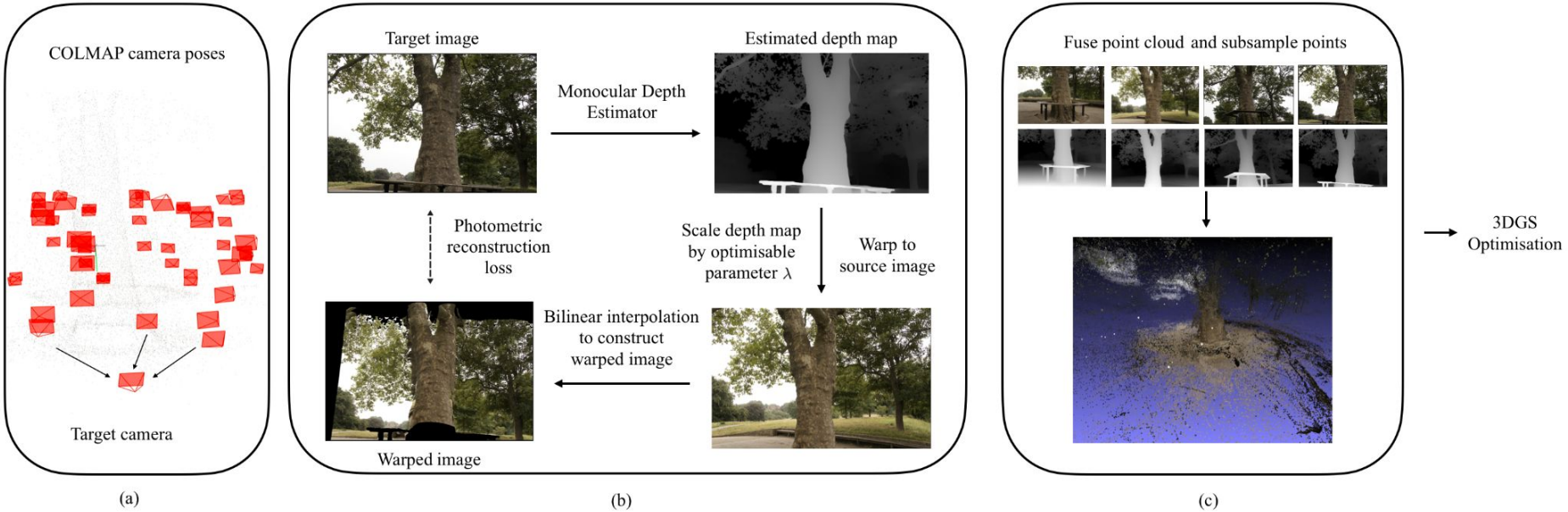
where scaling both sides by $\lambda \in \mathbb{R}$ gives

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

- That is, scaling the predicted depth and translation by the same value gives the same pixel in homogeneous coordinates

  In this project we are given **fixed camera poses** from SfM, and this procedure can be reformulated to optimise the scale.

Method Overview



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 image warp process introduced by *Zhou et al.* c) Finally, the scaled depth maps are fused and a subsampling procedure obtains a point cloud for 3DGS initialisation.

Method: Depth scale optimisation

- The image warping process is reformulated as

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

  Where λ is the unknown scale. The warped image is obtained by bilinear interpolation.

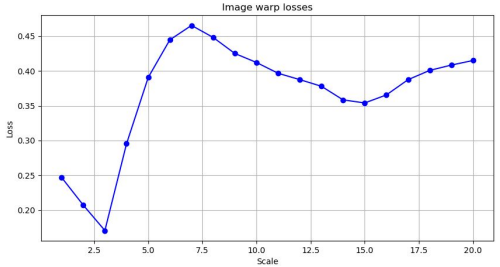
- The optimal scale λ minimises the image reconstruction error

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

  Where *M* is a binary mask that selects *non-blank* pixels.

Practical Details

- In practice, even when the loss is averaged over non-blank pixels there are cases where the loss is minimised with an incorrect scale due to **low-texture regions**



Target image | Source image | Warped image λ=3.0 | Warped image λ=15.0

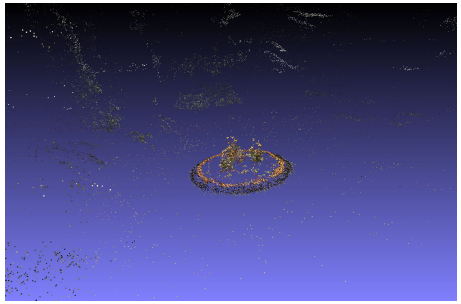
- Experiments found the optimised scale to lie between 10-20 for most images across all datasets
- Therefore, to obtain a good initialisation and to avoid low scales being selected, λ is initialised by searching from 10-20 and selecting the scale that gives the lowest loss while having over 50% non-blank pixels
- λ is then optimised with SGD for 100 iterations

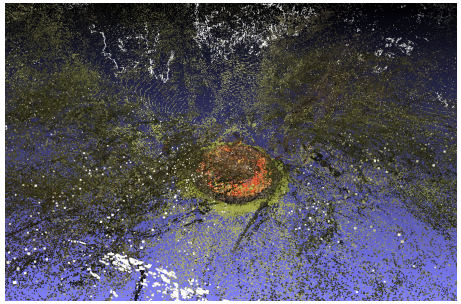
Point Cloud Fusion

- Taking every pixel in every image will result in too many points
- A simple strategy would be to *randomly sample* N=1 000 000 points, but this results in a lot of noise due to inaccuracies in the estimated depth map and scale:
- Reduce noise with *warp-sampling*: project points to a nearby held-out test view, and compute loss pixel-wise. Then select the pixels that give the lowest loss.

Dataset image - Flowers

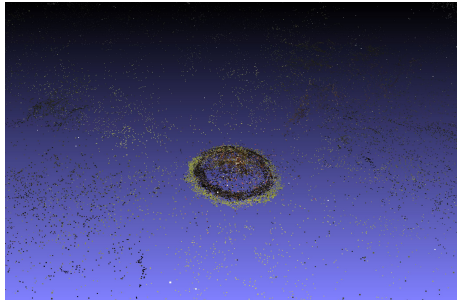
SfM

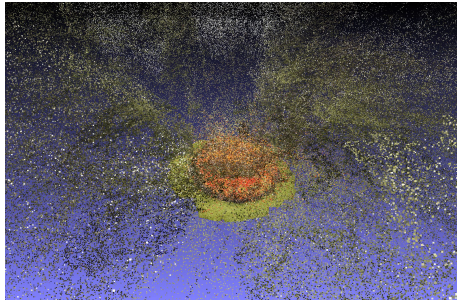




Warp-sample
1M pts

Warp-sample
100K pts

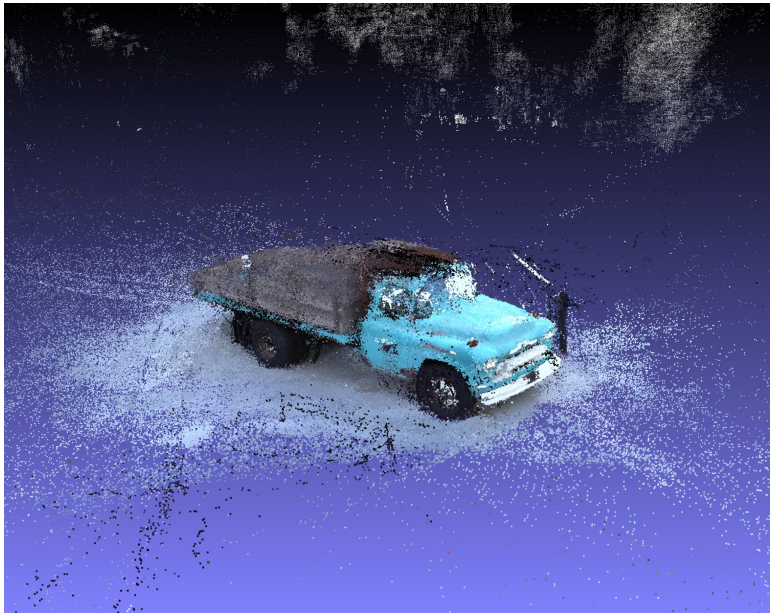




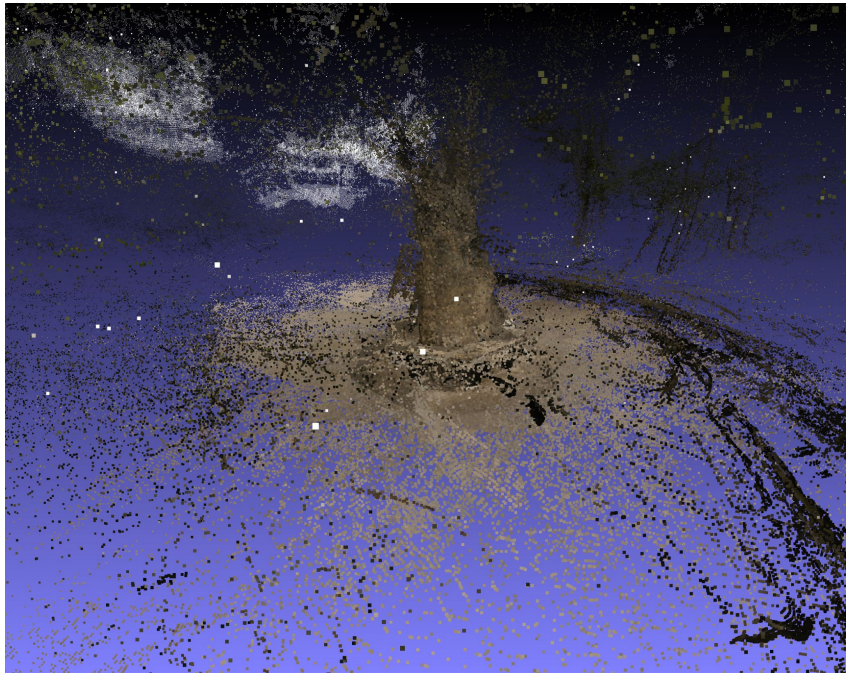
Random sample
1M pts

Are the optimised scales correct?

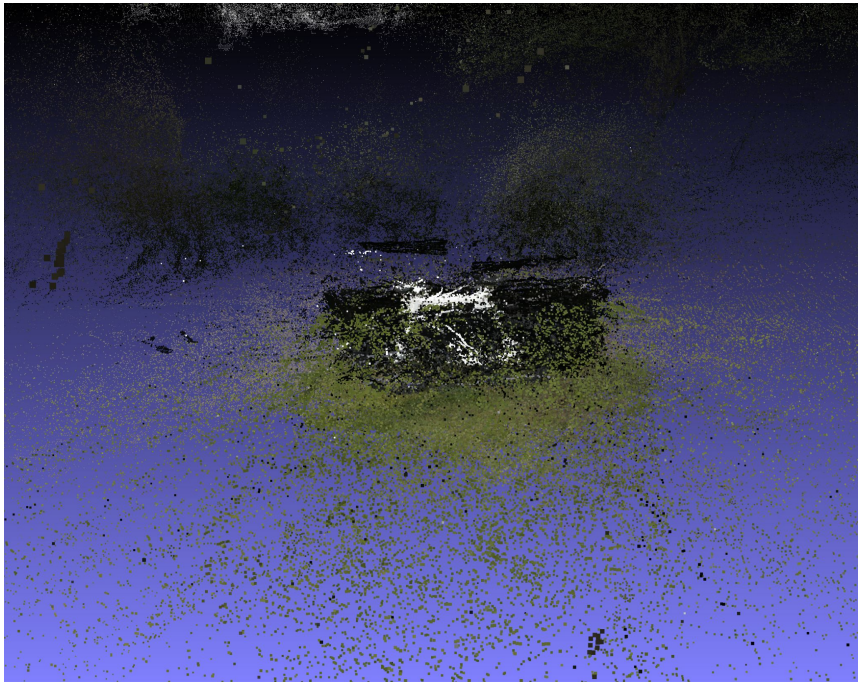
It is difficult to tell without ground truth depth.



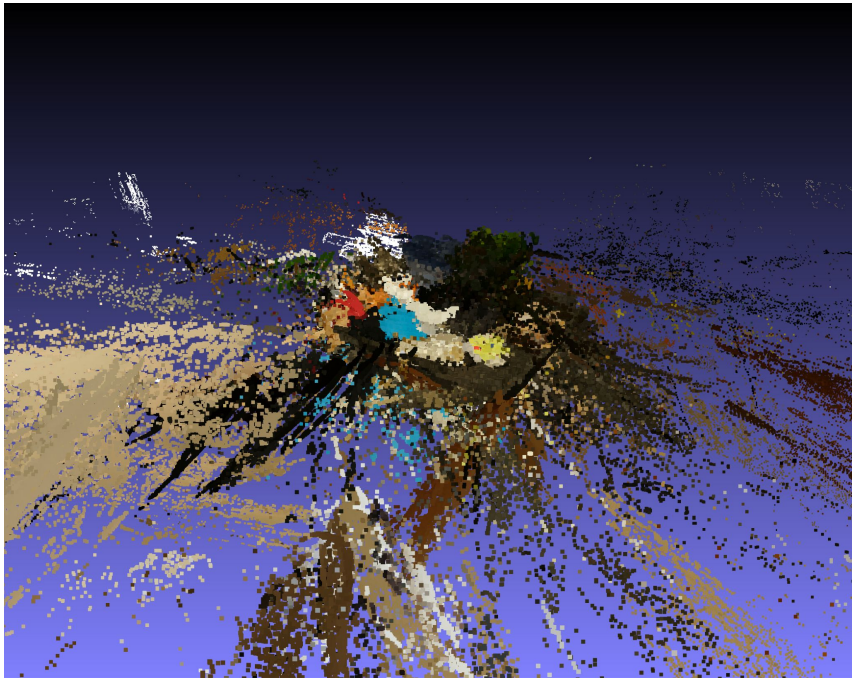
Truck



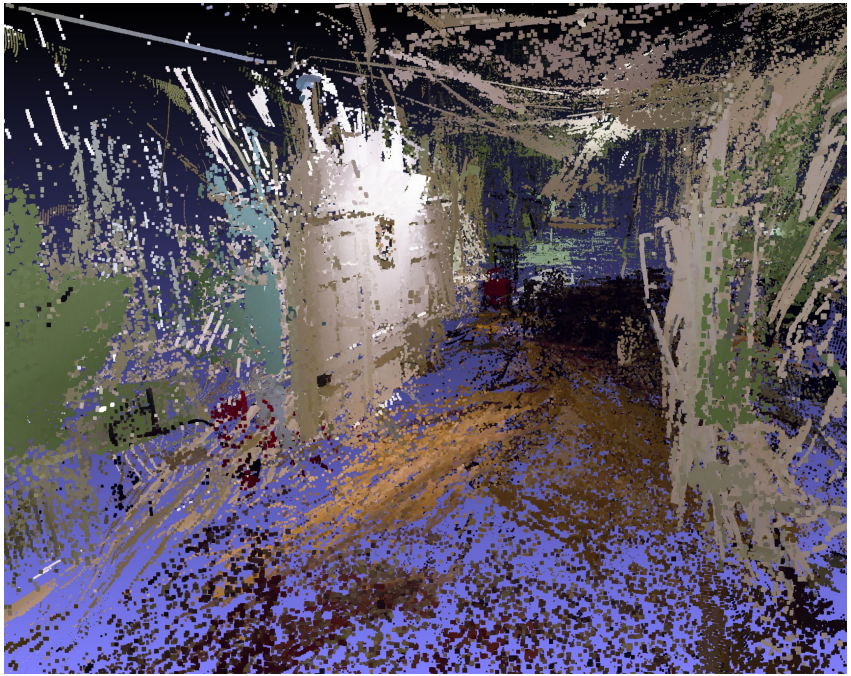
Treehill



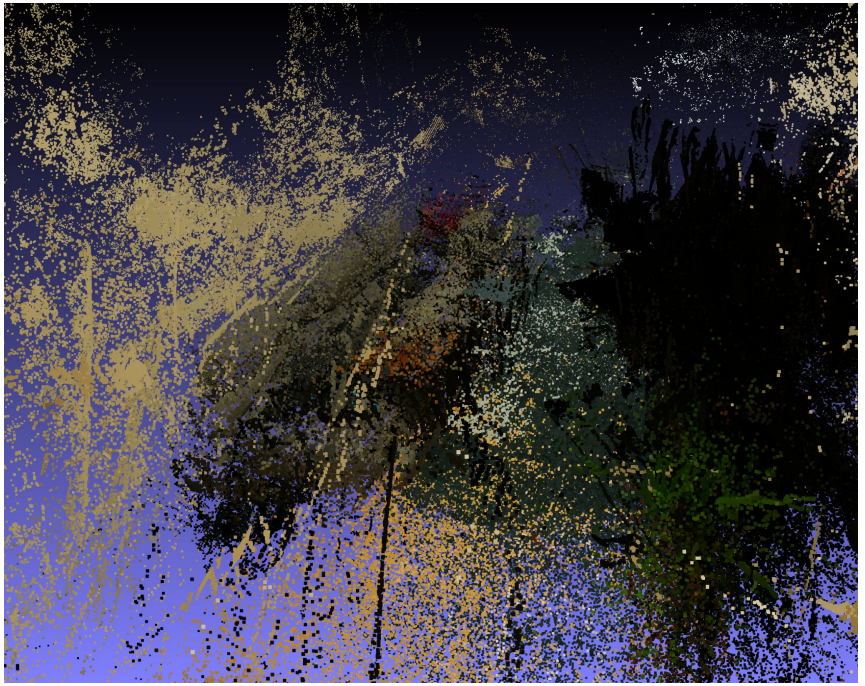
Bicycle



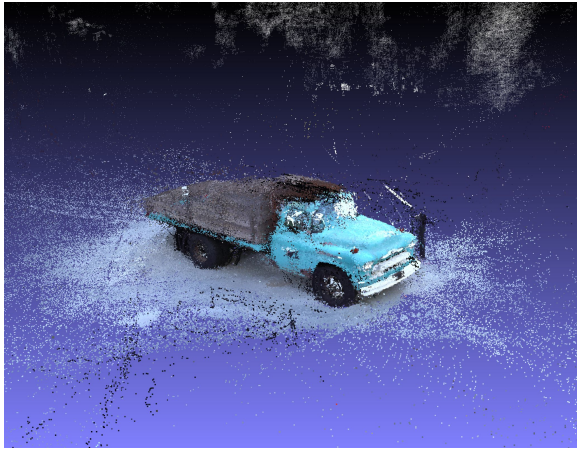
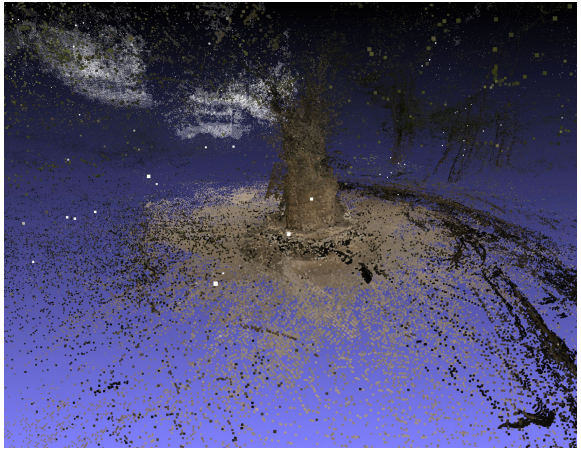
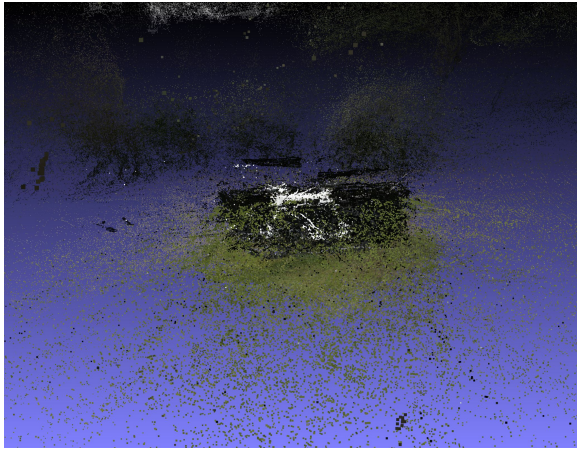
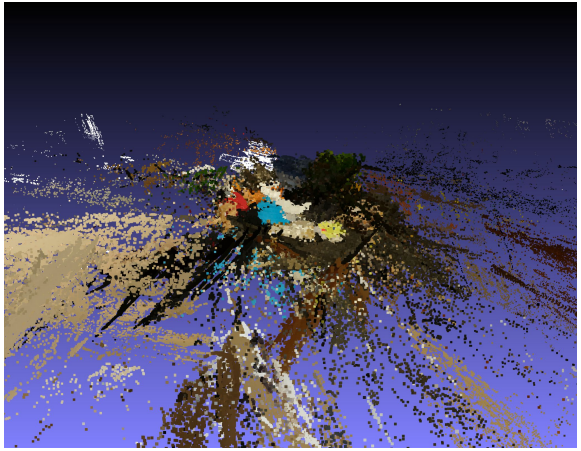
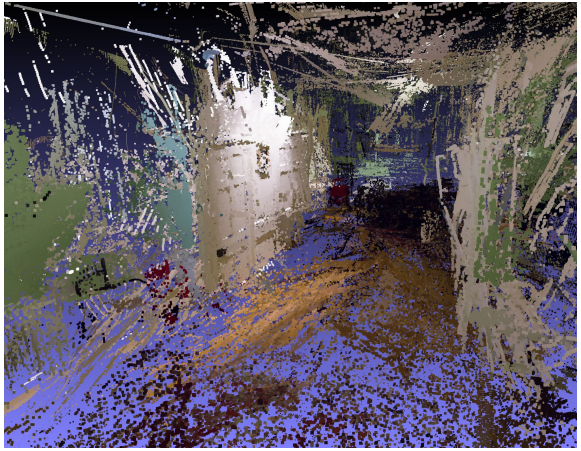
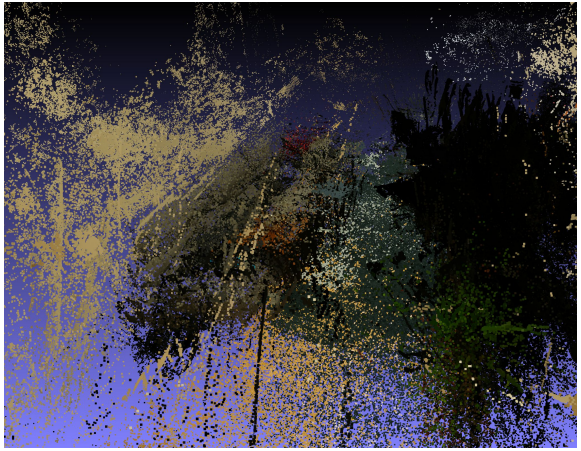
Counter



Dr Johnson



Room

Experiments

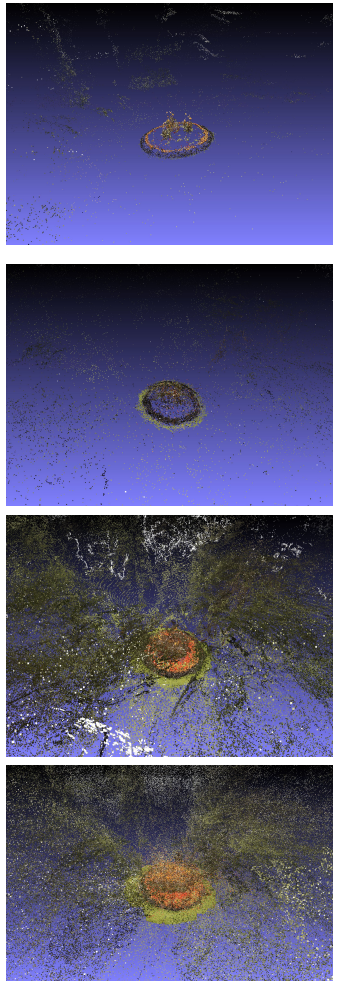
Optimise scenes with the following initialisations with default hyperparameter setting and SSIM, PSNR, LPIPS recorded at 7K and 30K iterations.

1) SfM

2) Warp-sample 100K pts

3) Warp-sample 1M pts

4) Random sample 1M pts



Results

- SfM initialisation scores the best overall, but the difference is small
- The difference between results at 7K and 30K iterations is similar for all initialisation methods.

Notes:

- Initialising with even more points did not improve performance
- Increasing gaussian sizes led to notable decrease in performance

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

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

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

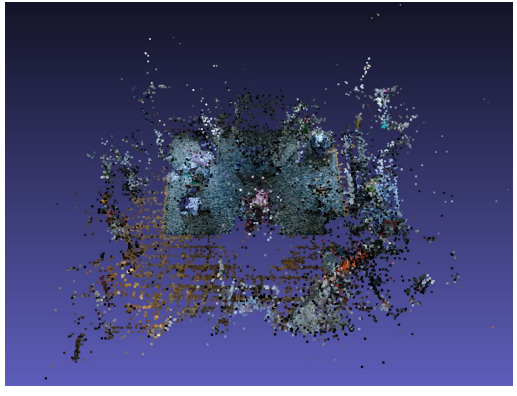
SfM initialisation - optimised 30K iterations

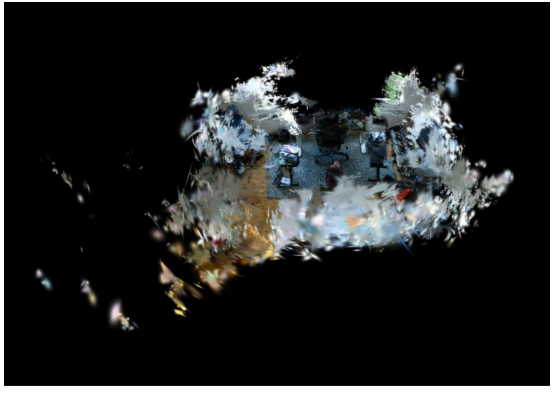
Warp-sampled 1M pts - optimised 30K iterations

Results

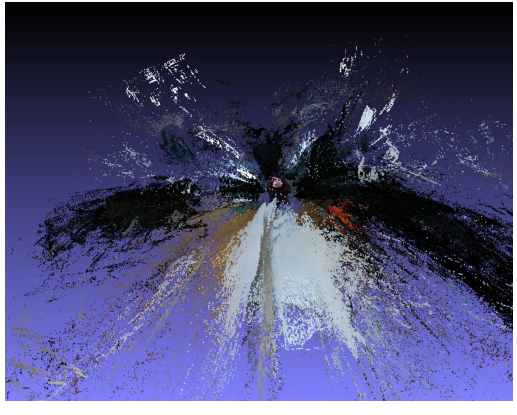
Our initialisation method mainly affects geometry in background regions

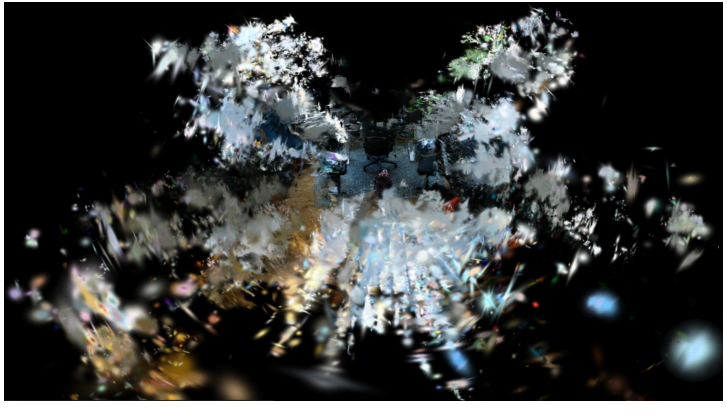
SfM initialisation





Warp-sampled 1M points





Geometry

SfM initialisation - optimised 30K iterations

Warp-sampled 1M pts - optimised 30K iterations

Red: regions further away are better reconstructed in warp-sampled initialisation
Blue: regions closer are better reconstructed by SfM initialisation

Geometry

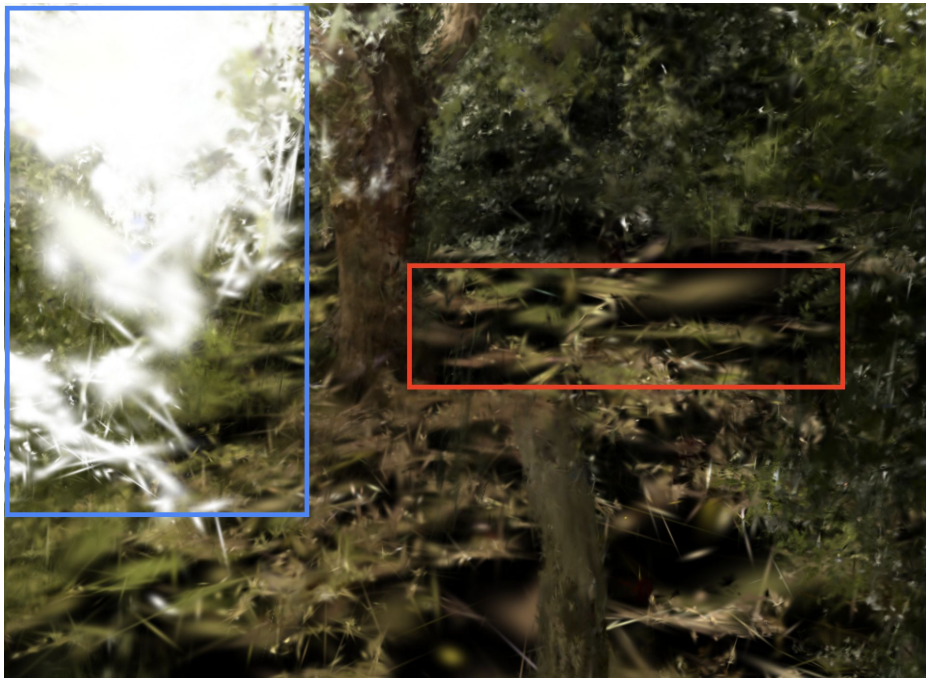
SfM initialisation - optimised 30K iterations

Warp-sampled 1M pts - optimised 30K iterations

Red: regions further away are better reconstructed in warp-sampled initialisation
Blue: regions closer are better reconstructed by SfM initialisation

Geometry

SfM initialisation - optimised 30K iterations



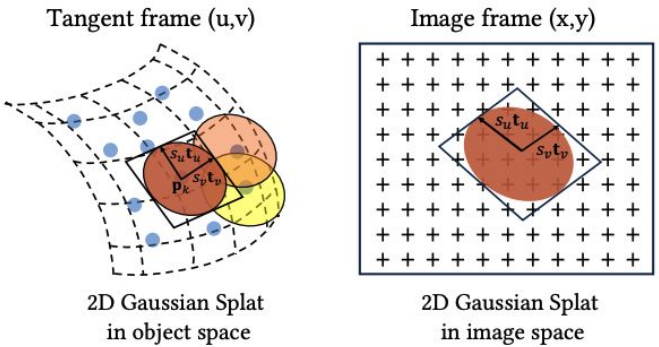
Warp-sampled 1M pts - optimised 30K iterations

Red: regions further away are better reconstructed in warp-sampled initialisation
Blue: regions closer are better reconstructed by SfM initialisation

The rendering quality - geometry trade-off

2DGS represents gaussians as discs to better model surfaces and employs depth and normal regularisation, while

SuGaR extracts a mesh and restricts gaussians to lie on the surface of the mesh.



2DGS

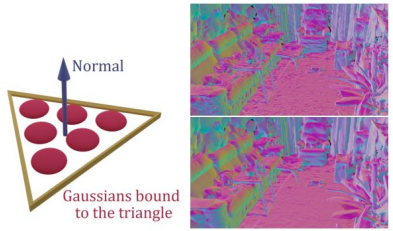


Figure 7. **Joint refinement of mesh and Gaussians. Left:** We bind Gaussians to the triangles of the mesh. Depending on the number of triangles in the scene, we bind a different number of Gaussians per triangle, with predefined barycentric coordinates. **Right:** Mesh before and after joint refinement.

SuGaR

Both works observe **improved geometry** but **decrease in rendering quality** compared to vanilla 3DGS due to increased regularisation. This indicates that there is an inherent **trade-off between rendering quality and geometry.**

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. **2d gaussian splatting for geometrically accurate radiance fields.** 2024.
Antoine Guedon and Vincent Lepetit. **Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering.** 2024.

Limitations & Future Work

1) The fused point clouds were too noisy.
   - Because SfM points are sparse, Adaptive Density Control focuses on **creating new gaussians**
   - Can we modify it to focus on **removing incorrect gaussians?**

2) It was difficult to tell whether optimised scales were correct.
   - Use datasets with ground truth depth or point clouds

3) No depth regularisation during 3DGS optimisation.
   - Use depth regularisation techniques with scaled depth maps