组会学术分享

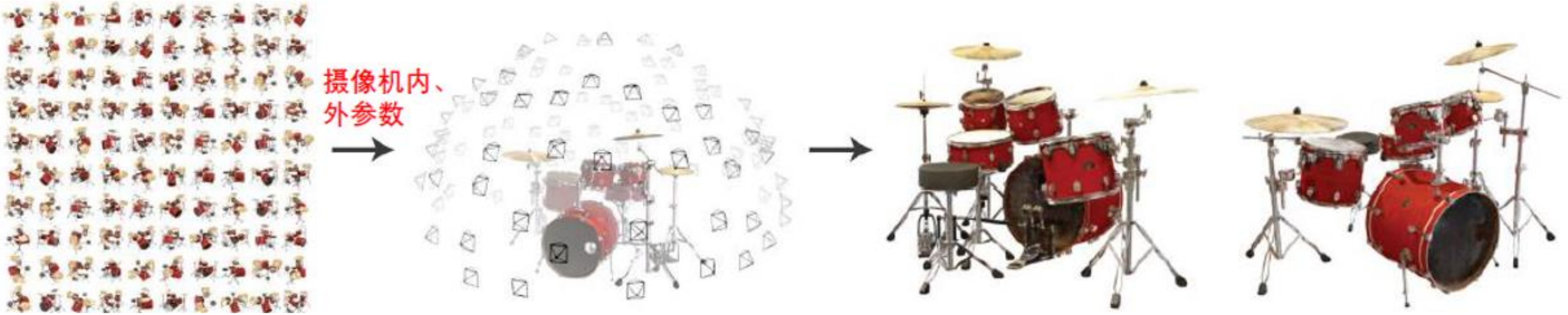
"Binary Opacity Grids Capturing Fine Geometric Detail for Mesh-Based View Synthesis"

https://creiser.github.io/binary_opacity_grid/

摆冬冬

20240909

任务：新视图合成（NVS）



Input
多视角图片

Neural Scene Representations
OccupancyField，DeepSDF，NeRF，3DGS…

Rendering
Volumetric，Point-based…

Mesh-Based View Synthesis
先重建出场景的Mesh：传统，Learning-based

Rendering
直接套现代图形渲染管线

Output
新视角图片

任重道远！

- **可视化**：输出逼真的**照片**
- **物理仿真**：把**现实环境数字化**，以便在**虚拟环境**训练**智能体**，如碰撞检测，动作生成，路径规划…

内容提纲

1 简介
2 场景表示
3 转三角形Mesh
4 渲染方法
5 实验

1 简介

Binary Opacity Grids: Capturing Fine Geometric Detail for Mesh-Based View Synthesis

SIGGRAPH 2024

Christian Reiser[1,2,3] Stephan Garbin[1] Pratul P. Srinivasan[1] Dor Verbin[1] Richard Szeliski[1] Ben Mildenhall[1] Jonathan T. Barron[1] Peter Hedman*[1]
Andreas Geiger*[2,3]

*equal advising Google Research[1] Tübingen AI Center[2] University of Tübingen[3]

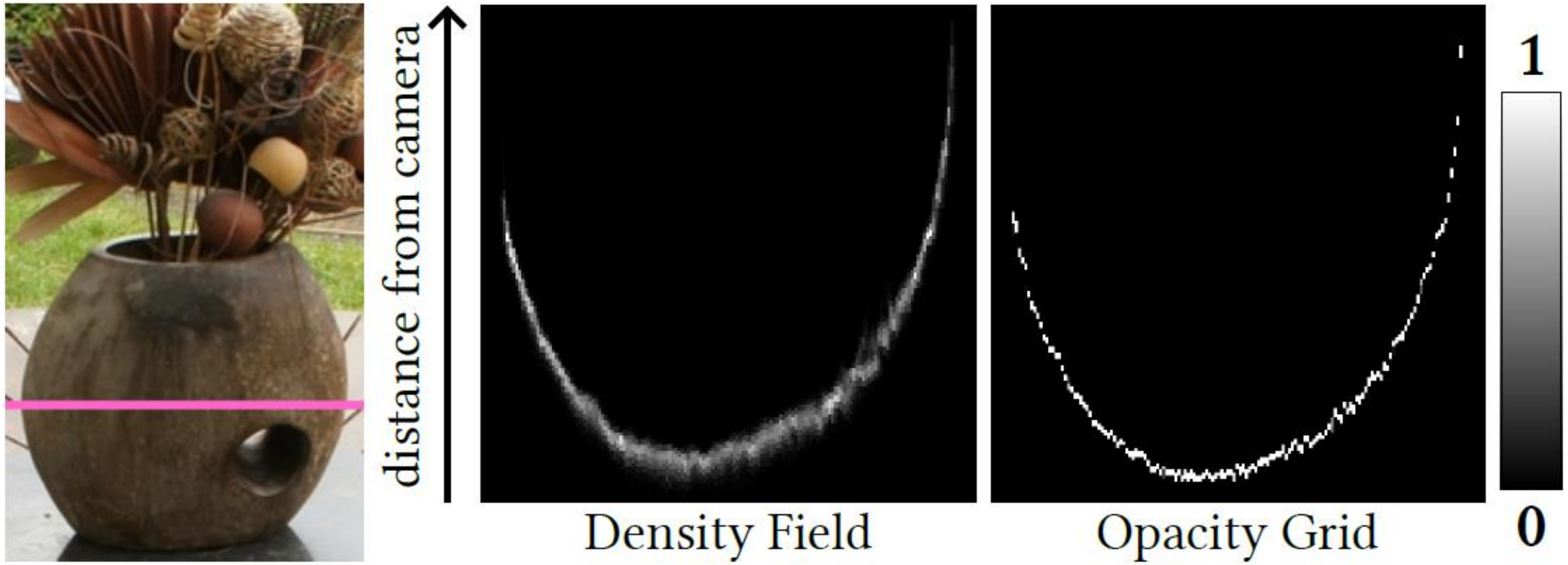
- Andreas Geiger：**图宾根大学**自主视觉组（AVG）负责人，计科系主任…

  2024年：CVPR*8，ECCV*6，SIGGRAPH*2，ICLR*2，TPAMI *1…

  有影响力的工作：**DVR**、**Occupancy Networks**、**TensoRF**、**2DGS**、**Mip-Splatting**…
- Christian Reiser ：1993.10生于德国，Andreas Geiger的在读博士生（since 2020），**KiloNeRF**、**MERF**等的一作
- 其余作者：**NeRF**及其各种变体的作者
- 作者大部分和**BakedSDF**的重叠！

➢ 动机

- **Surface-based** view synthesis algorithms (e.g. **MobileNeRF**, **BakedSDF**)
  - appealing due to low **computational requirements**
  - struggle to reproduce **thin structures**
- More expensive methods (dominant paradigm) model the scene as a **volumetric density field** (**NeRF类**)
  - excel at reconstructing **fine detail**
  - represent **geometry** in a "**fuzzy**" manner, which hinders **exact localization** of the **surface**



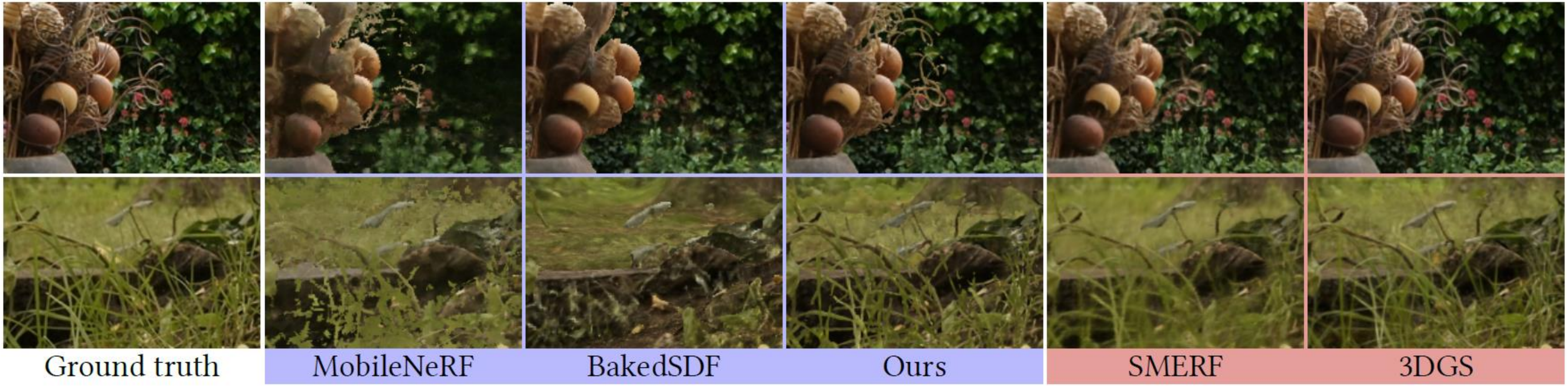
SDF类、NeRF类算法的共性问题！

Fig. 2. **Volume rendering** **weights** for rays along a row of pixels

- 本文方法
  - modify **density field** to encourage them to converge towards **surfaces**, without compromising their ability to reconstruct **thin structures**.
  - by applying the three **modifications** to an existing SOTA **Zip-NeRF**.
- 实现了
  - ✓ Produce compact **meshes**
  - ✓ Real-time rendering **mobile devices**
  - ✓ Higher view synthesis **quality** compared to existing mesh-based approaches

	Outdoor Scenes			Indoor Scenes		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Instant-NGP [2022]	22.90	0.566	0.371	29.15	0.880	0.216
MERF [2023]	23.19	0.616	0.343	27.80	0.855	0.271
3DGS [2023]	24.64	0.731	0.234	30.41	0.920	0.189
Zip-NeRF [2023]	**25.68**	**0.761**	**0.208**	**32.65**	**0.929**	**0.168**
Shells [2023b]	23.17	0.606	0.389	29.19	0.872	0.285
SMERF [2023]	25.32	0.739	0.232	31.32	0.917	0.186
Mobile-NeRF [2023]	21.95	0.470	0.470	–	–	–
BakedSDF [2023]	22.47	0.585	0.349	27.06	0.836	0.258
Ours (SSAA)	**23.94**	**0.680**	**0.263**	**27.71**	**0.873**	**0.227**

✓ narrows the **quality gap** between **surface-based** and **volume-based** methods when it comes to the reconstruction of **thin structures**.



2、场景表示：Opacity-based Voxel Grid

- Represent the scene with an R×R×R **voxel grid** (R＞2^13)
- Each **voxel**:
  - an **opacity** $\alpha \in [0, 1]$
  - a view-dependent **color** **c** ∈ [0, 1]^3
- To render a **pixel**, we cast a **ray**, it is then intersected with all of the **voxels** along its path.
  - For each intersected **voxels**, we query its **opacity** α_k and its **color** $\mathbf{c}_k$.
  - The final **pixel** value C is computed using **alpha compositing**:

$$\mathrm{C} = \sum_k \alpha_k \left(\prod_{j=1}^{k-1} \alpha_j \right) \mathbf{c}_k \, . \qquad (1)$$

回忆、对比**NeRF**和**3DGS**

- 优势：when all **opacity** are binary, the **surface** must be located at the first **voxels** along the **ray**
- predict the grid values using an **MLP** equipped with a **multi-resolution hash encoding** as **InstantNGP**
- train a **Zip-NeRF** to produce a converged **proposal MLP**: encodes the **coarse geometry** of the scene

2、场景表示：Binary Opacity

- To encourage binary **opacity** values
  - use an **entropy loss** that pulls **opacity** ＜0.5 towards 0 and ＞0.5 towards 1

$$\mathcal{L}_{\text{ent}} = \frac{1}{k}\sum_{k} \mathrm{H}(\alpha_k), \tag{2}$$

$$\mathrm{H}(p) = -p\log_2(p) - (1-p)\log_2(1-p). \tag{3}$$

  - 影响见实验-**Quality Loss Analysis**（论文第8页）
- Cast multiple **rays** per **pixel** during training (**Mip-NeRF**).
  - 16 sub-rays each pixel.
  - final pixel value is computed as the arithmetic mean of the subpixel values.
  - **supersampling** produces a significant improvement in **geometric quality**, especially regarding the **thin structures**, which often cover less than a single pixel.

3、转三角形Mesh：Volumetric Fusion

- During **training**, some **voxels** may only be sampled in a fraction of the **views** and thus have incorrect **opacity** values. This leads to **floating artifacts** in the resulting mesh (see Figure 3).

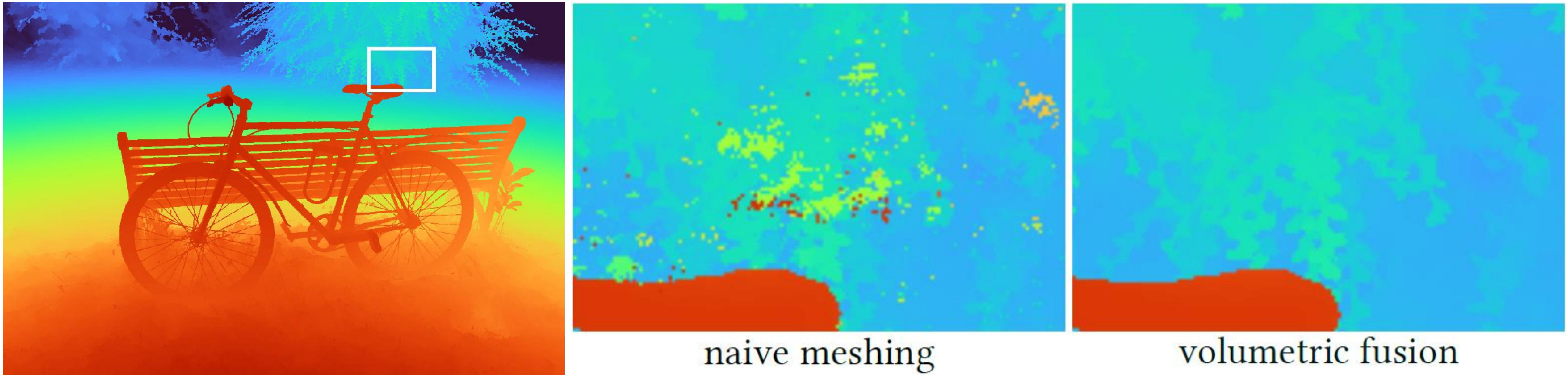
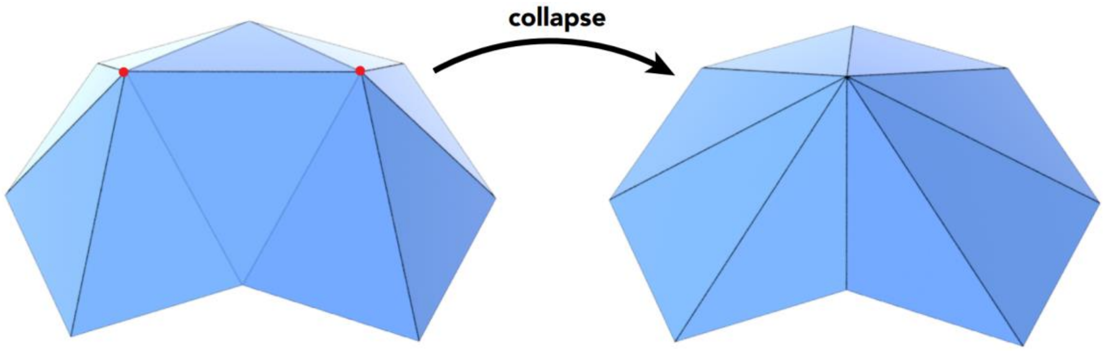


Fig. 3 **BakedSDF**（基于**MarchingCube**）vs. 本文使用**volumetric fusion**过滤后的

- Use **volumetric fusion** [Curless and Levoy 1996] to filter these underconstrained **voxels** and convert into a hole-free **mesh**.
  - This filtering step also fully preserves **thin structures**.

3、转三角形Mesh：Simplification and Culling

- To produce a more compact representation, we simplify the **mesh** with an off-the-shelf tool based on **quadric edge collapse decimation** [Garland and Heckbert 1997].



- **Cull triangles** that are not visible from any **training camera**, which leads to another significant reduction in the number of triangles.
  - Crucial to perform **culling** after **simplification**, as mesh simplification methods tend to not be robust to the numerous **small holes** introduced by **culling**.

4、渲染方法

- Explore parameterizations which efficiently map **positions** on **mesh** to **coefficients** that encode **appearance**:
  - **UV Mapping**
    - ✓ cannot deal well with the **complexity** of input **mesh**, which contains a lot of fine geometric detail
    - ✓ a viable path: [Nuvo: Neural UV Mapping for Unruly 3D Representations, 2023]
  - **Vertex Attributes: BakedSDF**
    - ✓ store appearance coefficients at vertex attributes on the mesh and interpolatie across each **face**.
    - ✓ requires the **vertex density** to be higher than the **desired texture density**, which results in prohibitively large and expensive **meshes**.
    - ✓ our meshes are drastically simplified, leading to large triangles in geometrically simple regions.
  - **Volume Textures**
    - ✓ directly associate a **color** value with each 3D **position**.
    - ✓ subdivide the **volume** into D^3 **voxels** and store only the nonempty ones.
    - ✓ **block size** D involves a trade-off: small - poor data locality, large - high memory consumption.

4、渲染方法

■ Triplanes and Low-resolution Voxel Grid

- ✓ **Volume Textures** encoded compactly with **triplanes** and a low-resolution **voxel grid** [**MERF**], both are cache-friendly, fast **random access**, **texture resolution** is not bounded by **vertex density**.
- ✓ fit our best-performing **appearance model** to the **meshes** from **BakedSDF**, call it **BakedSDF++**, leads to sharper **textures**, indicates that this representation might be a viable alternative to **Vertex Attributes** even for **dense meshes** as produced by **BakedSDF**.

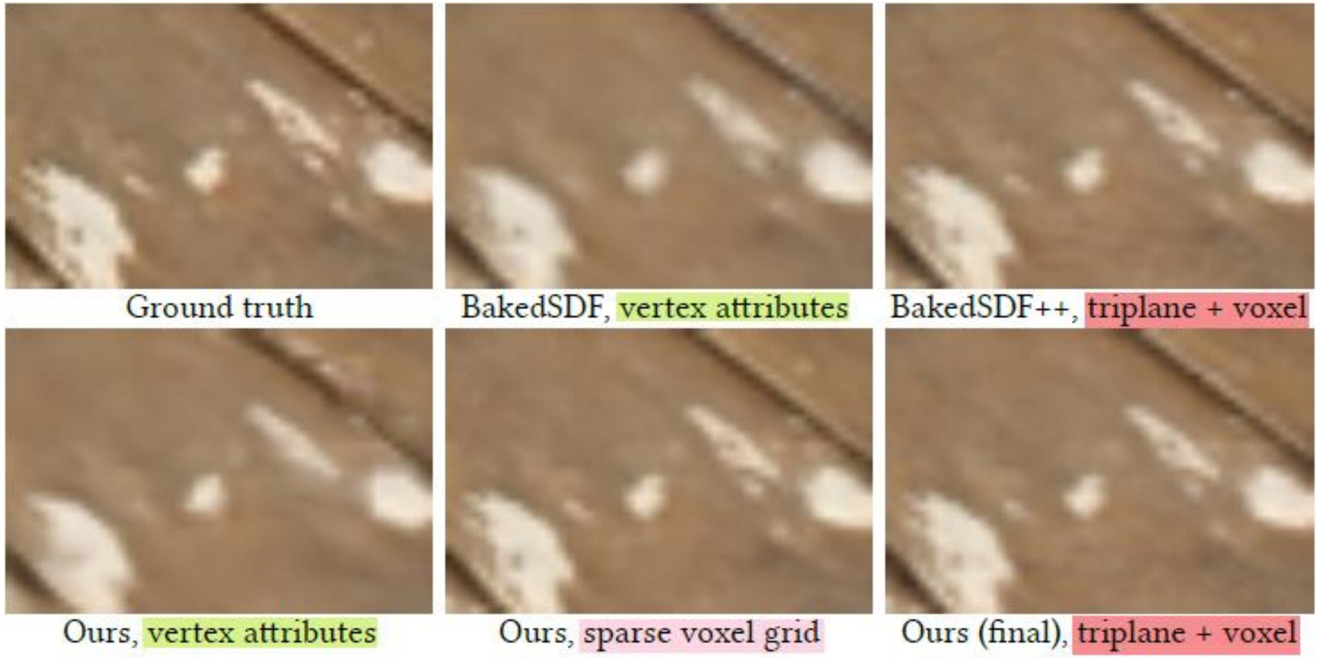


Fig. 4. **Different representations for mesh appearance.** Replacing ver-

Table 1. Different representations for mesh appearance in gardenvase. Replacing vertex attributes with a 3D grid leads to higher quality at the cost of higher memory consumption (VRAM). At a slight quality loss, the "triplane + voxel" option is more compact, while rendering faster than the alternatives. All rows except the last one use spherical Gaussians to model view-dependence. The last row (offline) is an upper bound on quality and uses an expensive appearance network.

	PSNR ↑	SSIM ↑	LPIPS ↓	VRAM ↓	FPS ↑
vertex attributes	25.58	0.771	0.211	**97**	261
volume textures	**26.25**	**0.820**	**0.143**	4513	169
triplane + voxel	26.02	0.807	0.157	629	**477**
offline	26.86	0.830	0.135	–	–

4、渲染方法

- Investigate several encodings for view-dependent **color**:
  - **spherical harmonics**
  - **spherical Gaussians**
  - **neural feature vectors**
    - ✓ get decoded to a view-dependent color with a small **MLP**.

Table 2. View-dependency encodings on gardenvase using our combination of triplanes and a low-resolution voxel grid.

	PSNR ↑	SSIM ↑	LPIPS ↓	bytes ↓
Spherical Gaussians	**26.02**	**0.807**	**0.157**	24
Spherical Harmonics	25.65	0.797	0.166	27
8-dim. Neural Feature	25.18	0.781	0.179	**8**
24-dim. Neural Feature	25.72	0.798	0.164	24

4、渲染方法

- Since our meshes contain many **tiny structures**, **antialiasing** (AA) is critical during **rendering**.
  - implement **temporal anti-aliasing** (**TAA**) [Computer Graphics Forum 2020]
  - **quality** of our TAA is on par with the significantly more expensive **SSAA**, even when capturing frames under **motion**. *见后面实验*

5、实验

- 训练耗时
  - **Table 10** shows processing times for each stage of our pipeline.
  - All stages use 8×**V100 GPUs**, except for **volumetric fusion** (single) and **simplification** (CPU only).
  - Our implementation of **volumetric fusion** is highly inefficient, offering potential for significant speed-ups with, e.g., a **CUDA** implementation.
  - 16x **supersampling** results in roughly 3x slower training than **Zip-NeRF** (6x **multisampling**).

Table 10. Processing times for the stages of our pipeline.

Stage	hours
Zip-NeRF Optimization	2.4
BOG Optimization	7.1
Volumetric Fusion	20.7
Simplification	3.7
Mesh Appearance Optimization	26.2
Total	60.1

5、实验

➢ Loss Analysis

- We use the same architecture as **Zip-NeRF**, any **loss** in rendering quality before **meshing** is from our **surface constraints (entropy regularization)**. Row 2 in Table 9: this leads to a 1.28 dB drop in **PSNR**.
- Converting opacity grid into a **triangle mesh** only decreases **PSNR** by 0.06 dB (row 3). To isolate the quality loss incurred by meshing, we equipped the mesh with the same offline **appearance model** that was used during **BOG optimization**.
- Comparing row 3 and 4, our lightweight **appearance model** incurs a quality loss of 0.39 dB.

Table 9. Quality loss incurred by real-time concessions. Results are averaged over the outdoor scenes from mip-NeRF 360 [Barron et al. 2022].

	PSNR ↑	SSIM ↑	LPIPS ↓
(1) Zip-NeRF	**25.68**	**0.761**	**0.208**
(2) BOG before meshing	24.40	0.698	0.263
(3) Our mesh: offline appearance	24.34	0.699	0.239
(4) Our mesh: real-time appearance	23.94	0.680	0.263

5、实验

➢ **Anti-Aliasing** for test-time rendering is crucial for high quality.

- always employ **16× supersampling strategy** during training
- only vary the **anti-aliasing** algorithm used for test-time rendering
- since **TAA** may introduce blur under **motion**, we also measure the quality of images that were captured after moving the camera over a fixed number of frames to the target pose.

Table 3. Test-time anti-aliasing algorithms on the outdoor scenes from the mip-NeRF 360 dataset [Barron et al. 2022]. TAA achieves nearly the same fidelity as significantly more expensive SSAA.

	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↓
SSAA	23.94	**0.680**	**0.263**	50
TAA, stationary	**24.00**	**0.680**	0.266	448
TAA, under motion	23.92	0.676	0.270	448
No AA	23.26	0.652	0.287	**477**

5、实验

➢ Comparison with **BakedSDF**, our method:

- significantly better at reconstructing **thin structures** (Figure 7)
- outperforms **BakedSDF++** across all key **metrics** (Table 6), our **meshes** are better for **NVS**

Ours Ground truth BakedSDF Ours

	PSNR ↑	SSIM ↑	LPIPS ↓	#faces ↓
BakedSDF	22.47	0.585	0.349	40M
BakedSDF++	22.50	0.612	0.315	40M
Ours (SSAA)	**23.94**	**0.680**	**0.263**	**13M**

5、实验

- Comparison with other **Baselines**: **rendering** speed
  - Google Pixel 8 Pro **smartphone**
  - MacBook M1 Pro (2022) **laptop**
  - **desktop** equipped with an **NVIDIA RTX 3090**

Table 5. Rendering speed comparison in frames per second. Our method is significantly faster than **volume-based** and **surface-based** baselines and is the only method capable of real-time rendering on our test smartphone.

Device Resolution	Smartphone 400 × 750	Laptop 1280 × 720	Desktop 1920 × 1080
MERF [2023]	10	21	113
3DGS [2023]	–	–	176
BakedSDF [2023]	19	81	412
Ours (TAA)	**67**	**448**	**927**

5、实验

- Comparison with other **Baselines**: **quality**

- still lags behind the most recent **volume-based** baselines (Table 4).
- **quality gap** between **surface-based** and **volume-based** methods is significantly reduced, especially **thin structures** (Figure 5).

	Outdoor Scenes			Indoor Scenes		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Instant-NGP [2022]	22.90	0.566	0.371	29.15	0.880	0.216
MERF [2023]	23.19	0.616	0.343	27.80	0.855	0.271
3DGS [2023]	24.64	0.731	0.234	30.41	0.920	0.189
Zip-NeRF [2023]	**25.68**	**0.761**	**0.208**	**32.65**	**0.929**	**0.168**
Shells [2023b]	23.17	0.606	0.389	29.19	0.872	0.285
SMERF [2023]	25.32	0.739	0.232	31.32	0.917	0.186
Mobile-NeRF [2023]	21.95	0.470	0.470	–	–	–
BakedSDF [2023]	22.47	0.585	0.349	27.06	0.836	0.258
Ours (SSAA)	**23.94**	**0.680**	**0.263**	**27.71**	**0.873**	**0.227**

Ground truth | MobileNeRF | BakedSDF | Ours | SMERF | 3DGS

5、实验

➢ Geometry Ablations

	PSNR ↑	SSIM ↑	LPIPS ↓
(a) No supersampling	23.38	0.645	0.292
(b) No entropy loss	23.21	0.635	0.293
(c) R = 2048 instead of R = 8192	22.44	0.582	0.343
Ours (SSAA)	**23.94**	**0.680**	**0.263**

a) disable **supersampling** during **training**, but still use for fitting the **appearance model** and for **computing quality metrics**. This isolates the effect on the quality of the obtained **mesh**. As shown in Figure 6, **thin structures** are hard to recover.

b) without the **entropy loss**: similar to (a), the effect is most pronounced for very **thin structures**.

c) decrease the **resolution** of the initial BOG, but use the same resolution during **appearance fitting**: a high resolution is crucial for reconstructing **thin structures**.

Ground truth (a) no supersampling (b) no entropy loss (c) low resolution Full model

5、实验

- **Storage Analysis**: how **mesh** and **appearance** contribute to **disk storage** and **memory consumption**.
  - using **simplification** and **culling**, the size of the **mesh** can be reduced by a factor of 100 to around 200 MiB.
  - the size of the **representation** being dominated by the **appearance model**, which occupies around 76% of the overall storage.
  - Table 8: results are averaged over all **scenes** from **mipNeRF-360**.

		(a) Dense Mesh	(b) + Simpl.	(c) + Culling
Mesh	#vertices	606M	9M	**7M**
Mesh	#faces	1208M	18M	**10M**
Mesh	VRAM	20.28 GiB	0.30 GiB	**0.19 GiB**
Mesh	DISK	21.40 GiB	0.32 GiB	**0.20 GiB**
Appearance	VRAM	0.75 GiB	0.75 GiB	**0.75 GiB**
Appearance	DISK	0.65 GiB	0.65 GiB	**0.65 GiB**
Total	VRAM	21.02 GiB	1.05 GiB	**0.94 GiB**
Total	DISK	22.05 GiB	0.97 GiB	**0.85 GiB**

总结

- 局限
  - Like other **mesh-based** view synthesis methods, ours does not handle **semi-transparent objects**.
  - While our meshes **render** quickly, the **processing time** is substantial巨大.
  - Reconstruction of the underconstrained **background** of the scene is often highly noisy, significantly increases the **size** of mesh, could potentially be mitigated with a **smoothness regularizer**.
  - Another consequence of the lack of **smoothness** is that mesh **normals** are too noisy for **relighting**.
  - **Quality gap** remains between our approach and **volume-based** methods. We hypothesize that **disturbances** during **capture**, like inaccurate poses, wind, motion blur, or depth of field, present a greater challenge for **surface-based** approaches since **volumetric** approaches can fuzzily resolve these disturbances.
- 作者众星云集，见解深刻
  - 揭示了使用了**体渲染**的场景表示方法，在**转换mesh**时存在的根源问题
  - NVS**研究范式**转变：mesh-based，实用性增加

谢谢大家！