

# A Comparative Study of Neural Surface Reconstruction for Scientific Visualization

Siyuan Yao\*

University of Notre Dame

Weixi Song†

Wuhan University

Chaoli Wang‡

University of Notre Dame

## ABSTRACT

This comparative study evaluates various neural surface reconstruction methods, particularly focusing on their implications for scientific visualization through reconstructing 3D surfaces via multi-view rendering images. We categorize ten methods into neural radiance fields and neural implicit surfaces, uncovering the benefits of leveraging distance functions (i.e., SDFs and UDFs) to enhance the accuracy and smoothness of the reconstructed surfaces. Our findings highlight the efficiency and quality of NeuS2 for reconstructing closed surfaces and identify NeUDF as a promising candidate for reconstructing open surfaces despite some limitations. By sharing our benchmark dataset, we invite researchers to test the performance of their methods, contributing to the advancement of surface reconstruction solutions for scientific visualization.

## 1 INTRODUCTION

Multi-view 3D surface reconstruction represents a pivotal progress in computer vision, offering a methodology for synthesizing 3D geometric models from multiple 2D images captured from varying camera views. This research involves sophisticated algorithms integrating these disparate views to form a cohesive 3D representation of the observed object or scene.

In scientific visualization, this research direction is advantageous when only surface rendering images, rather than the original surface or volume data, are available. Such a scenario is common, as original volume data and extracted surfaces are not always publicly disclosed. With a small set of rendering images of the surface captured from different viewpoints, we can faithfully reconstruct the 3D surface, enabling examination from arbitrary angles. Additionally, with the reconstructed surface, we can adjust lighting and rendering parameters to enhance surface details or present the surface in a preferred manner. For large volumetric datasets, surface rendering images can be remotely produced by high-performance computing (HPC) clusters and efficiently transferred to local users. The corresponding surfaces can be locally reconstructed in minutes using the resulting rendering images instead of the volume data.

Yet, a significant gap remains in our understanding of how these reconstruction methods perform when applied to scientific datasets. In scientific visualization, various surface types present unique challenges for 3D surface reconstruction, and no single method is guaranteed to work for every dataset. Special care must be taken to account for data characteristics and make necessary tradeoffs to meet the demands of professional users or the general public.

For example, isosurfaces can be complex and exhibit occlusions, with inner surfaces visible only from certain viewpoints. In such a case, a method that excels in occlusion tolerance or maintains good surface consistency may be preferable over efficiency. Conversely,

for remote reconstruction from HPC clusters, users might prioritize more efficient methods to quickly obtain the surface.

Different surface characteristics also play a crucial role. Although isosurfaces usually include open and closed surfaces, applying closed surface reconstruction methods can still produce reasonable results. However, this approach is not effective for stream surfaces. Using closed surface reconstruction methods on stream surfaces, which are often open, can lead to extra faces or incorrect face connections, making the reconstructed surface very different from the rendered appearance. Even with the existing open surface methods, we must balance accuracy, smoothness, and speed.

To bridge this gap, we introduce a benchmark dataset to thoroughly compare ten state-of-the-art surface reconstruction techniques. Our dataset encompasses nine isosurfaces and stream surfaces, ranging from simple to complex, including closed and open surfaces, each with distinct characteristics. Through this comprehensive study, we shed light on the performance of these methods, offering valuable insights and recommendations.

## 2 RELATED WORK

**Data reconstruction via deep learning.** Deep learning has accomplished many scientific visualization tasks, including data reconstruction [31], which focuses on restoring data from visual or incomplete sources. Examples include [10, 13, 15, 35, 36]. Nevertheless, in scientific visualization, no reconstruction technique has been developed that learns surface representations from a collection of 2D rendered images taken from various angles, as opposed to utilizing volumetric data [12] or surface meshes [14].

**Explicit surface reconstruction.** Reconstructing 3D surfaces from multi-view 2D images is common in computer vision and computer graphics. Typically, 3D surfaces can be represented explicitly as *voxels* [6, 17, 38], *point clouds* [1, 7, 19, 23], or *meshes* [9, 32, 37], and surface reconstruction is often performed using techniques like marching cubes [22], ball pivoting [4], and Poisson reconstruction [16]. Due to the discrete nature of these representations, surface reconstruction from these fully explicit representations usually yields noisy and incomplete results.

**Implicit surface reconstruction.** The recent surge of neural rendering frameworks [26, 39, 40] and implicit surface representations [11, 25, 29, 30] has propelled advances in multi-view 3D surface reconstruction with considerable improvements in accuracy, consistency, and flexibility. Although neural rendering frameworks like NeRF [26] can extract surfaces from learned density fields, the results tend to be rough and irregular due to the lack of geometric constraints. Multi-view surface reconstruction methods, such as VolSDF [39] and NeuS [33], employ the *signed distance function* (SDF) for implicit surface representation. NeuS2 [34] and Neuralangelo [18] significantly improve the training speed of NeuS by integrating *multiresolution hash encoding* (MHE). NeAT [24] introduces a validation network for open surface reconstruction. In contrast, NeUDF [20] and NeuralUDF [21] employ an *unsigned distance field* (UDF) to represent open surfaces and achieve high-quality reconstruction. We assess and benchmark various neural surface reconstruction methods, discussing their adaptation to isosurfaces and stream surfaces commonly found in scientific visualization, offering recommendations, and outlining remaining challenges.

\*e-mail: syao2@nd.edu

†e-mail: song.wx@whu.edu.cn

‡e-mail: chaoli.wang@nd.edu

### 3 RADIANCE FIELDS AND IMPLICIT SURFACES

**Neural radiance fields.** Neural radiance field (NeRF) [26], a seminal work on view synthesis, uses a deep, fully connected network that ingests a continuous 5D coordinate (including 3D spatial location and 2D viewing angle) and outputs the corresponding volume density and view-dependent emitted radiance. This method synthesizes views by iterating 5D coordinates along camera rays and employs the standard volume rendering process to produce the novel view. One of its most striking features is NeRF’s simplicity, i.e., using a *multilayer perceptron* (MLP) to process these coordinates and output density and color. NeRF can represent detailed scene geometry with complex occlusions and even convert the radiance field to a mesh using marching cubes. However, the vanilla NeRF has limitations such as slow training and rendering speeds, inability to represent dynamic scenes, and “baking in” lighting.

TensorRF [5] pushes the boundaries of 3D scene reconstruction regarding efficiency and fidelity. In contrast to NeRF (which depends entirely on the use of MLPs for scene rendering) and Plenoxels [8] (which relies on a fully explicit sparse voxel grid for volume representation), TensorRF conceptualizes the entire volume of a scene as a 4D tensor, including the feature channel as one dimension. Specifically, it involves decomposing the scene’s tensor into several compact, low-rank tensor components, enabling a more refined and efficient model representation using a significantly smaller MLP than NeRF. Compared with Plenoxels, TensorRF offers a substantial reduction in memory usage due to its efficient tensor factorization for handling the feature grid, which enhances scene modeling and allows for more efficient and accurate rendering.

Using MHE, Instant-NGP [27] replaces many parameters in the NeRF network with a smaller one, supplemented by a set of trainable encoding parameters. These encoding parameters are stored at the vertices of multiple grid layers, facilitating the learning of scene details at various resolutions. A key feature of Instant-NGP is its ability to achieve almost-instant training of neural graphics primitives using a single GPU. Moreover, Instant-NGP’s architecture allows real-time training progress on various datasets, supporting complex scenes. Its implementation in a single CUDA kernel, referred to as “fully-fused MLP” results in a tenfold efficiency improvement compared to the original NeRF implementation.

**Neural implicit surfaces.** Differentiable volumetric renderer (DVR) [28] and implicit differentiable renderer (IDR) [40] significantly advance neural implicit surface reconstruction from multi-view 3D images, laying the groundwork for subsequent development. NeuS [33] further eliminates the need for mask supervision in DVR and IDR. Using an SDF and a new volume rendering method for training, NeuS allows for more accurate surface reconstruction, especially in complex structures and self-occlusion cases. NeuS2 [34] and Neuralangelo [18], both implementations of NeuS within the Instant-NGP framework, permit instantaneous training. However, this comes at the cost of noise artifacts in both approaches.

NeUDF [20] and NeuralUDF [21] focus on reconstructing surfaces with open boundaries and arbitrary topologies, utilizing UDFs. In NeUDF, a differentiable volume rendering framework is introduced to predict UDFs from input images, emphasizing the need for a specialized rendering procedure and a point sampling strategy tailored for UDFs. NeuralUDF overcomes the challenges of locating the zero level set for a UDF field, which is crucial for accurately defining open surfaces. Leveraging a modified visibility indicator function and optimizing the UDF fields, NeuralUDF can accurately recover objects with open surfaces from 2D images.

### 4 COMPARISON AND DISCUSSION

For a more nuanced comparative analysis, we categorize ten methods into three distinct groups: NeRF (including NeRF [26], TensorRF [5], and Instant-NGP [27]), neural implicit closed surface (featuring IDR [40], NeuS [33], NeuS2 [34], and Neuralangelo [18]), and neu-

ral implicit open surface (comprising NeAT [24], NeUDF [20], and NeuralUDF [21]). We include NeuS in all comparisons as it stands out as the most representative work. NeuS is closely related to the methods in the three groups and effectively highlights their distinctions. This section compares neural renderings and reconstructed surfaces in static images. The accompanying video compares the results across the full 360-degree view.

#### 4.1 Datasets, Training, and Metrics

We used nine datasets in our experiments, with each group comparison using three datasets to showcase the differences. Table 1 lists the volumetric datasets, spatial resolutions, and surface information. The five critical points and solar plume datasets use stream surfaces, while all others use isosurfaces. For the isosurface, we selected an isovalue leading to a salient isosurface covering the domain well. Note that for the supernova dataset, the isosurface exhibits a complex, multi-layered structure, including inner parts hidden from any viewing angle. We use this challenging case to evaluate the robustness of surface reconstruction methods and the consistency of their reconstructed surfaces. For the stream surface, we picked a seeding curve to yield a prominent stream surface. For each dataset, we used 42 rendering images from evenly-placed sample views for training and synthesized 181 new views during inference. The image resolution is  $1024 \times 1024$  for both training and inference. We also reconstructed the underlying surface through the learned density field, SDF, or UDF. All methods were trained on a single NVIDIA V100 GPU using the default configurations suggested in the respective works. Then, we rendered the reconstructed isosurface or stream surface as usual using the same 181 views. The details of training parameters are presented in Table 2.

Table 1: Datasets, spatial resolutions, and surface information.

	dataset	dimension ( $x \times y \times z$ )	surface type	# original vertices
neural radiance field	Nyx	$256 \times 256 \times 256$	isosurface	211,910
	Tangaroa	$300 \times 180 \times 120$	isosurface	162,187
	vortex	$128 \times 128 \times 128$	isosurface	153,574
neural closed surface	aorta	$512 \times 666 \times 251$	isosurface	949,999
	combustion	$480 \times 720 \times 120$	isosurface	1,945,531
	supernova	$432 \times 432 \times 432$	isosurface	2,396,278
neural open surface	five critical points	$51 \times 51 \times 51$	stream surface	2,072
	Marschner-Lobb	$256 \times 256 \times 256$	isosurface	497,108
	solar plume	$126 \times 126 \times 512$	stream surface	85,863

Table 2: Training details of the three groups of methods.

method	neural radiance field			
	NeRF	TensorRF	Instant-NGP	NeuS
framework	PyTorch	PyTorch	CUDA	PyTorch
learning rate	$5 \times 10^{-4}$	$1 \times 10^{-3}$	$1 \times 10^{-3}$	$5 \times 10^{-4}$
# iterations	100k	30k	35k	300k
method	neural closed surface			
	IDR	NeuS	NeuS2	Neuralangelo
framework	PyTorch	PyTorch	CUDA	PyTorch
learning rate	$1 \times 10^{-4}$	$5 \times 10^{-4}$	$1 \times 10^{-2}$	$1 \times 10^{-3}$
# iterations	1000k	300k	15k	500k
method	neural open surface			
	NeuS	NeAT	NeUDF	NeuralUDF
framework	PyTorch	PyTorch	PyTorch	PyTorch
learning rate	$5 \times 10^{-4}$	$1 \times 10^{-4}$	$2 \times 10^{-4}$	$5 \times 10^{-4}$
# iterations	300k	1000k	400k	300k

For quantitative comparison, we used peak signal-to-noise ratio (PSNR) and learned perceptual image patch similarity (LPIPS) [41] to evaluate the quality of inferred neural rendering images. We utilized the Chamfer distance (CD) [3] to compute the difference between reconstructed surface  $\hat{S}$  and ground truth (GT) surface  $S$

$$d_{CD}(S, \hat{S}) = \sum_{v \in S} \min_{\hat{v} \in \hat{S}} \|v - \hat{v}\|_2^2 + \sum_{\hat{v} \in \hat{S}} \min_{v \in S} \|\hat{v} - v\|_2^2. \quad (1)$$

We reported CD in the above two terms rather than their summation. The first term measures the distance from GT surface vertices  $v$  to the nearest vertices on the reconstructed surface  $\hat{S}$ , and the second

term from reconstructed surface vertices  $\hat{v}$  to the nearest vertices on the GT surface  $S$ . This separation provides a clearer assessment of reconstruction accuracy.

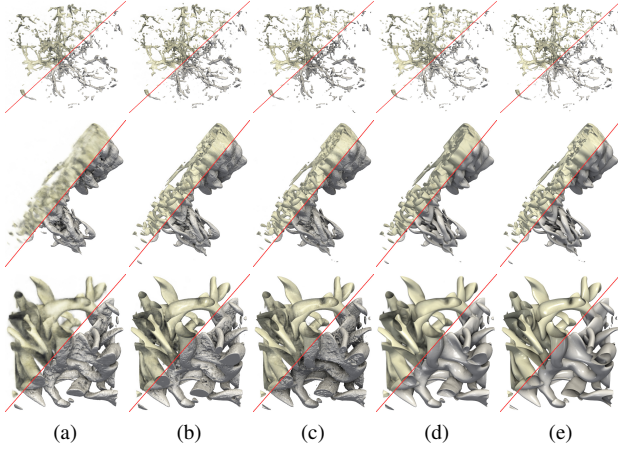


Figure 1: Inferred neural rendering images (upper-left) and rendering images of reconstructed surfaces (lower-right) of Nyx, Tangaroa, and vortex generated by (a) NeRF, (b) TensorRF, (c) Instant-NGP, and (d) NeuS. (e) shows the GT results.

Table 3: Comparison of NeuS vs. NeRF methods. Average PSNR (dB) and LPIPS across all synthesized views, CD between the reconstructed and GT surfaces, training time (TT, in hours), and model size (MS, in MB). The best ones are highlighted in bold.

dataset	method	PSNR $\uparrow$	LPIPS $\downarrow$	CD $\downarrow$	TT $\downarrow$	MS $\downarrow$
Nyx	NeRF	24.20	0.224	11.32 + <b>0.49</b>	11.67	13.69
	TensorRF	26.89	0.090	<b>4.19</b> + 8.14	0.68	66.90
	Instant-NGP	<b>27.82</b>	<b>0.075</b>	7.09 + 788.12	<b>0.05</b>	23.80
	NeuS	23.74	0.157	5.43 + 0.58	11.01	<b>11.55</b>
	NeRF	25.83	0.145	3.38 + 1.31	11.31	13.69
Tangaroa	TensorRF	<b>32.08</b>	<b>0.029</b>	<b>0.75</b> + 1.00	0.69	69.16
	Instant-NGP	31.73	0.034	0.78 + 90.54	<b>0.06</b>	23.80
	NeuS	30.13	0.039	4.20 + <b>0.55</b>	10.98	<b>11.55</b>
	NeRF	23.21	0.220	1.40 + 2.37	11.72	13.69
	TensorRF	<b>26.55</b>	0.101	<b>0.26</b> + 3.84	0.73	67.13
vortex	Instant-NGP	26.54	<b>0.096</b>	0.30 + 427.88	<b>0.06</b>	24.00
	NeuS	24.65	0.146	0.29 + <b>1.14</b>	10.30	<b>11.55</b>

## 4.2 Comparison Results

**NeuS vs. NeRF methods.** Figure 1 shows the qualitative results. For space-saving, we display the inferred neural rendering images on the upper-left side and the rendering images of reconstructed surfaces on the lower-right side. For neural rendering, the results indicate that except for the vanilla NeRF, which leads to blurred results, all other methods can generate inferred neural rendering images with reasonably good quality. NeRF methods are capable of generating reconstructed surfaces of reasonable quality. Nonetheless, reconstructing the underlying surfaces is more challenging than inferring novel views. We can see that NeRF-generated surfaces exhibit noticeable noise, especially for datasets with larger areas of even surfaces like Tangaroa and vortex. In contrast, NeuS delivers surfaces with a significantly higher degree of smoothness, owing to the integration of Eikonal regularization. This technique effectively aligns the normals across the surface, enhancing the smoothness and coherence of the reconstructed surface.

The quantitative results presented in Table 3 reveal that NeRF and NeuS do not yield neural renderings as detailed as the other two methods due to their fully implicit design. Nonetheless, NeuS, benefiting from its SDF representation, consistently maintains a low value of the second CD term across all datasets. This indicates that while some parts of the original surface may be missing, the reconstructed surface does not generate excessive extraneous elements. Conversely, for Instant-NGP, MHE is responsible for introducing

outliers significantly apart from the actual surface, resulting in an exceptionally high second CD term.

Regarding the training speed, the hybrid architectures of TensorRF and Instant-NGP demonstrate a considerable acceleration over the fully implicit models of NeRF and NeuS. This advantage, however, incurs an increased model size for storing explicit representations. Between Instant-NGP and TensorRF, Instant-NGP is more than  $10\times$  faster to train than TensorRF.

In summary, this group analysis suggests that while occupancy-based surface representations employed by NeRF are somewhat effective, the SDF-based representation showcases superior efficacy in reconstructing smooth and consistent surfaces.

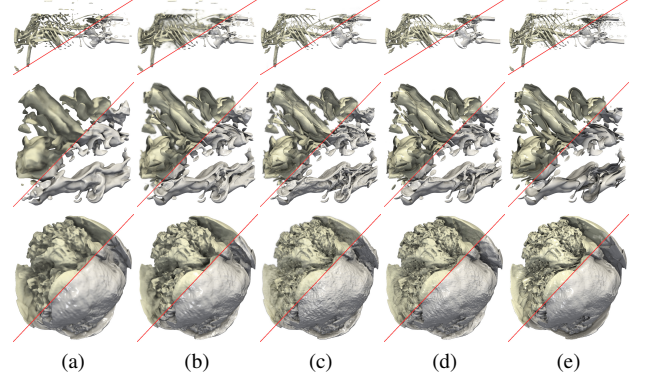


Figure 2: Inferred neural rendering images (upper-left) and rendering images of reconstructed surfaces (lower-right) of aorta, combustion, and supernova generated by (a) IDR, (b) NeuS, (c) NeuS2, and (d) Neuralangelo. (e) shows the GT results.

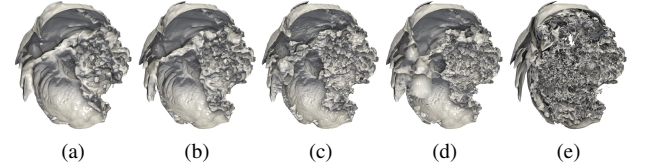


Figure 3: Cut-through rendering of reconstructed supernova surfaces generated by (a) IDR, (b) NeuS, (c) NeuS2, and (d) Neuralangelo. (e) shows the GT results.

Table 4: Comparison of neural implicit closed surface methods.

dataset	method	PSNR $\uparrow$	LPIPS $\downarrow$	CD $\downarrow$	TT $\downarrow$	MS $\downarrow$
aorta	IDR	20.06	<b>0.100</b>	<b>24.70</b> + 19.80	4.75	<b>11.13</b>
	NeuS	20.48	0.138	35.25 + 12.18	10.89	11.55
	NeuS2	<b>23.22</b>	<b>0.100</b>	127.03 + <b>1.76</b>	<b>0.13</b>	24.15
	Neuralangelo	22.27	0.117	433.59 + 2.07	23.87	1396.81
	IDR	20.19	0.210	74.61 + 29.20	5.03	<b>11.13</b>
combustion	NeuS	22.70	0.192	32.34 + 18.37	11.01	11.55
	NeuS2	24.56	0.161	<b>21.52</b> + 12.37	<b>0.12</b>	24.15
	Neuralangelo	<b>25.05</b>	<b>0.114</b>	24.21 + <b>12.35</b>	23.81	1396.81
	IDR	25.12	0.198	692.26 + 9.54	5.61	<b>11.13</b>
supernova	NeuS	26.08	0.179	729.48 + <b>9.17</b>	10.58	11.55
	NeuS2	26.63	0.162	334.14 + 9.44	<b>0.13</b>	24.15
	Neuralangelo	<b>27.84</b>	<b>0.112</b>	<b>281.26</b> + 10.87	23.73	1396.81

**Neural implicit closed surface methods.** Analyzing the results in Figure 2 reveals that IDR and NeuS tend to produce blurry images, in contrast to NeuS2 and Neuralangelo, which, utilizing MHE, can generate more detailed neural renderings. Compared to NeuS2, the neural renderings produced by Neuralangelo are characterized by improved smoothness and enhanced specular highlights. However, within the aorta dataset, NeuS2 and Neuralangelo exhibit missing components in the reconstructed surfaces, stemming from omitted isolated parts in voxels at the lower-resolution hash grids.

Table 4 indicates that Neuralangelo achieves superior overall neural rendering quality. However, it is outperformed by NeuS2 for the aorta dataset due to the absence of the prominent collarbone

and other details. IDR records the lowest overall CD for the aorta dataset, but its high second CD term implies the generation of inaccurate pieces. NeuS2 and Neuralangelo significantly surpass the other methods in neural rendering and surface reconstruction of the combustion and supernova datasets, which predominantly consist of dense surfaces. Upon closer examination, we find that the supernova surface exhibits a complex inner structure yet is heavily occluded from most camera views. To illustrate this, we slice the surface in half to expose its interior, as shown in Figure 3. The results indicate that all methods fail to reconstruct the intricate hidden details, leading to a notably high first CD term as shown in Table 4. Among these methods, Neuralangelo handles this scenario slightly better than its counterparts.

Although IDR and NeuS share similar model architectures, IDR trains faster due to its use of surface instead of volume rendering. NeuS2, utilizing tiny MLPs optimized in the CUDA framework, achieves a speed increase of approximately  $100\times$  compared to NeuS. Neuralangelo, with enormous hash grids and more training iterations, takes twice as long as NeuS.

The model sizes of both NeuS2 and Neuralangelo largely depend on their MHE configurations. Neuralangelo incorporates 16 levels of MHE, with resolution from  $2^5$  to  $2^{11}$ , and each resolution uses  $2^{22}$  hash entries. In contrast, NeuS2 employs 14 levels, with resolution from  $2^4$  to  $2^{11}$ , and allocates  $2^{19}$  hash entries per level.

Given the above findings, we recommend using NeuS2 for reconstructing closed surfaces due to its extraordinary efficiency and high-quality reconstruction.

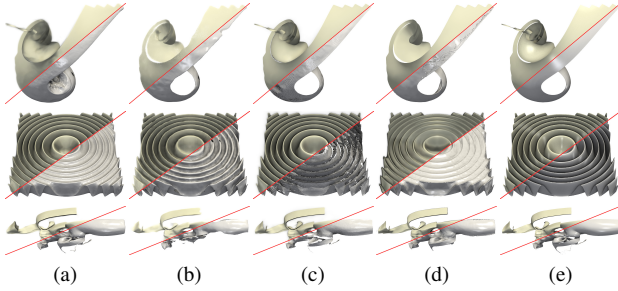


Figure 4: Inferred neural rendering images (upper-left) and rendering images of reconstructed surfaces (lower-right) of five critical points, Marschner-Lobb, and solar plume generated by (a) NeuS, (b) NeAT, (c) NeUDF, and (d) NeuralUDF. (e) shows the GT results.

Table 5: Comparison of neural implicit open surface methods.

dataset	method	PSNR $\uparrow$	LPIPS $\downarrow$	CD $\downarrow$	TT $\downarrow$	MS $\downarrow$
five critical points	NeuS	23.60	0.087	0.11 + 9.52	<b>11.12</b>	11.55
	NeAT	<b>26.99</b>	0.102	0.64 + 0.41	33.10	<b>9.50</b>
	NeUDF	26.89	<b>0.040</b>	<b>0.05 + 0.31</b>	16.92	11.55
	NeuralUDF	26.30	0.049	1.57 + 0.57	20.64	10.23
		<b>25.51</b>	0.084	<b>0.35 + 1.03</b>	<b>10.77</b>	11.55
Marschner-Lobb	NeAT	24.36	<b>0.074</b>	0.40 + 0.91	33.99	<b>9.50</b>
	NeUDF	25.41	0.084	1.33 + <b>0.87</b>	17.41	11.55
	NeuralUDF	21.69	0.104	0.40 + 1.08	20.04	10.23
		26.04	0.079	0.26 + 1.24	<b>10.68</b>	11.55
solar plume	NeAT	22.50	0.113	<b>0.16 + 0.36</b>	33.22	<b>9.50</b>
	NeUDF	<b>26.92</b>	<b>0.063</b>	<b>0.18 + 0.21</b>	16.70	11.55
	NeuralUDF	25.39	0.079	0.25 + 0.58	19.36	10.23

**NeuS vs. neural implicit open surface methods.** Table 5 shows that methods based on SDF and UDF exhibit comparable neural rendering performance. However, NeAT and NeuralUDF demonstrate less consistency in their results than NeuS and NeUDF. According to Figure 4, NeAT and NeuralUDF miss the noticeable spiral’s tip in their neural renderings for the five critical points dataset.

For stream surfaces of the five critical points and solar plume datasets, NeUDF achieves the lowest overall CD, indicating its most accurately reconstructed surfaces. The rendering results further validate its ability to reconstruct stream surfaces with high fidelity.

For the isosurface of the Marschner-Lobb dataset, NeUDF stands out as the only method capable of accurately reconstructing the

isosurface as a single open surface despite numerous artifacts. Contrarily, other methods often yield a watertight surface reconstruction. While NeUDF effectively learns the correct surface representation using UDF, improvements in the surface extraction technique are necessary to achieve optimal open surface reconstruction.

### 4.3 Summary

Across these ten methods compared, We see significant advantages in using SDF to depict surfaces instead of relying on density fields. This approach markedly enhances the precision and smoothness of the reconstructed surfaces. Additionally, numerous follow-up works have shown progress over the initial implementation of SDF techniques. Although Neuralangelo stands out for its substantial improvements in reconstructing complex surfaces, the increment of training time and model size is fairly substantial. In contrast, NeuS2 presents a more feasible option, offering efficient, high-quality surface reconstruction. Therefore, for the reconstruction of closed surfaces, we advocate for the adoption of NeuS2.

Regarding the reconstruction of open surfaces, the challenge intensifies. NeUDF emerges as a promising candidate, although the way it extracts surfaces still falls short of expectations. Other techniques for open surfaces often overlook fine details or inadvertently create a closed, watertight surface with unintended thickness, deviating from the goal of reproducing an open surface. Furthermore, the lack of acceleration strategies among these methods for open surfaces leaves room for innovation and improvement.

## 5 CONCLUDING REMARKS

Our comparative study shows that current surface reconstruction methods are efficient and can produce high-quality results within minutes when applied to isosurfaces. This success is attributed mainly to SDF surface representations and MHE. Furthermore, we recognize the promise of UDF for representing isosurfaces and stream surfaces and reconstructing them as open surfaces, as evidenced by the results of NeUDF. Despite these encouraging findings, surface reconstruction faces considerable challenges, especially in tackling occlusion from limited camera views and reconstructing minute surface details. We outline three potential future research and development directions that could benefit the scientific visualization community. **(1) Detail enhancement:** To improve the fidelity and minimize errors in reconstructed surfaces, we should develop strategies to capture the fine details more effectively without significantly impacting training time or inflating model size like Neuralangelo. This could involve utilizing new sampling methods (such as cone-casting in Mip-NeRF [2]) or optimizing existing ones (especially MHE) to balance detail fidelity with computational efficiency better. **(2) Efficient UDF representation:** This direction involves optimizing UDF computations or exploring novel architectures, like MHE or CUDA framework implementation, to efficiently leverage UDF for open and closed surface reconstruction. **(3) Surface extraction enhancement:** To obtain a high-quality open surface that is reliable for simulation or modeling, it is necessary to refine surface extraction techniques for more robust reconstruction of open surface results with enhanced flexibility. This may include developing new algorithms or modifying existing ones, such as marching cubes, to better accommodate the complexities of open surface reconstruction.

Finally, we invite researchers to utilize our benchmark dataset, publicly available at <https://www.kaggle.com/datasets/syaond/scivis-surface-dataset/>, to evaluate the performance of their methods. We hope this will promote advances in the field by providing a standardized basis for comparison, enabling rigorous evaluation of refined surface reconstruction techniques.

## ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation through grants IIS-1955395, IIS-2101696, OAC-2104158, and IIS-2401144, and the U.S. Department of Energy



through grant DE-SC0023145. The authors would like to thank the anonymous reviewers for their insightful comments.

## REFERENCES

- [1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3D point clouds. In *Proc. ICML*, pp. 40–49, 2018.
- [2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. ICCV*, 2021. doi: 10.1109/ICCV48922.2021.00580
- [3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. IJCAI*, pp. 659–663, 1977.
- [4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE TVCG*, 5(4):349–359, 1999. doi: 10.1109/2945.817351
- [5] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. TensorRF: Tensorial radiance fields. In *Proc. ECCV*, pp. 333–350, 2022. doi: 10.1007/978-3-031-19824-3\_20
- [6] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *Proc. ECCV*, pp. 628–644, 2016. doi: 10.1007/978-3-319-46484-8\_38
- [7] H. Fan, H. Su, and L. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proc. CVPR*, pp. 2463–2471, 2017. doi: 10.1109/CVPR.2017.264
- [8] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. CVPR*, pp. 5491–5500, 2022. doi: 10.1109/CVPR52688.2022.00542
- [9] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. A Papier-Mâché approach to learning 3D surface generation. In *Proc. CVPR*, pp. 216–224, 2018. doi: 10.1109/CVPR.2018.00030
- [10] P. Gu, J. Han, D. Z. Chen, and C. Wang. Reconstructing unsteady flow data from representative streamlines via diffusion and deep-learning-based denoising. *IEEE CG&A*, 41(6):111–121, 2021. doi: 10.1109/MCG.2021.3089627
- [11] B. Guillard, F. Stella, and P. Fua. MeshUDF: Fast and differentiable meshing of unsigned distance field networks. In *Proc. ECCV*, pp. 576–592, 2022. doi: 10.1007/978-3-031-20062-5\_33
- [12] J. Han, J. Tao, and C. Wang. FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE TVCG*, 26(4):1732–1744, 2020. doi: 10.1109/TVCG.2018.2880207
- [13] J. Han, J. Tao, H. Zheng, H. Guo, D. Z. Chen, and C. Wang. Flow field reduction via reconstructing vector data from 3-D streamlines using deep learning. *IEEE CG&A*, 39(4):54–67, 2019. doi: 10.1109/MCG.2018.2881523
- [14] J. Han and C. Wang. SurfNet: Learning surface representations via graph convolutional network. *CGF*, 41(3):109–120, 2022. doi: 10.1111/cgf.14526
- [15] J. Han and C. Wang. VCNet: A generative model for volume completion. *Vis. Informatics*, 6(2):62–73, 2022. doi: 10.1016/j.visinf.2022.04.004
- [16] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. SGP*, pp. 61–70, 2006. doi: 10.2312/SGP/SGP06/061-070
- [17] H. Li, X. Yang, H. Zhai, Y. Liu, H. Bao, and G. Zhang. Vox-Surf: Voxel-based implicit surface representation. *IEEE TVCG*, 30(3):1743–1755, 2024. doi: 10.1109/TVCG.2022.3225844
- [18] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proc. CVPR*, pp. 8456–8465, 2023. doi: 10.1109/CVPR52729.2023.00817
- [19] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *Proc. AAAI*, pp. 7114–7121, 2018. doi: 10.1609/AAAI.V32I1.12278
- [20] Y.-T. Liu, L. Wang, J. Yang, W. Chen, X. Meng, B. Yang, and L. Gao. NeUDF: Learning neural unsigned distance fields with volume rendering. In *Proc. CVPR*, pp. 237–247, 2023. doi: 10.1109/CVPR52729.2023.00031
- [21] X. Long, C. Lin, L. Liu, Y. Liu, P. Wang, C. Theobalt, T. Komura, and W. Wang. NeuralUDF: Learning unsigned distance fields for multi-view reconstruction of surfaces with arbitrary topologies. In *Proc. CVPR*, pp. 20834–20843, 2023. doi: 10.1109/CVPR52729.2023.01996
- [22] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH*, pp. 163–169, 1987. doi: 10.1145/37401.37422
- [23] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. In *Proc. BMVC*, 2018.
- [24] X. Meng, W. Chen, and B. Yang. NeAT: Learning neural implicit surfaces with arbitrary topologies from multi-view images. *Proc. CVPR*, pp. 248–258, 2023. doi: 10.1109/CVPR52729.2023.00032
- [25] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. CVPR*, pp. 4460–4470, 2019. doi: 10.1109/CVPR.2019.00459
- [26] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, pp. 405–421, 2020. doi: 10.1007/978-3-030-58452-8\_24
- [27] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127
- [28] M. Niemeyer, L. M. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. CVPR*, pp. 3501–3512, 2020. doi: 10.1109/CVPR42600.2020.00356
- [29] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. CVPR*, pp. 165–174, 2019. doi: 10.1109/CVPR.2019.00025
- [30] E. Remelli, A. Lukoianov, S. Richter, B. Guillard, T. Bagautdinov, P. Bague, and P. Fua. MeshSDF: Differentiable iso-surface extraction. In *Proc. NeurIPS*, pp. 22468–22478, 2020.
- [31] C. Wang and J. Han. DL4SciVis: A state-of-the-art survey on deep learning for scientific visualization. *IEEE TVCG*, 29(8):3714–3733, 2023. doi: 10.1109/TVCG.2022.3167896
- [32] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proc. ECCV*, pp. 55–71, 2018. doi: 10.1007/978-3-030-01252-6\_4
- [33] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proc. NeurIPS*, pp. 27171–27183, 2021.
- [34] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu. NeuS2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proc. ICCV*, pp. 3272–3283, 2023. doi: 10.1109/ICCV51070.2023.00305
- [35] S. Weiss, M. Chu, N. Thuerey, and R. Westermann. Volumetric isosurface rendering with deep learning-based super-resolution. *IEEE TVCG*, 27(6):3064–3078, 2021. doi: 10.1109/TVCG.2019.2956697
- [36] S. Weiss, M. Işık, J. Thies, and R. Westermann. Learning adaptive sampling and reconstruction for volume visualization. *IEEE TVCG*, 28(7):2654–2667, 2022. doi: 10.1109/TVCG.2020.3039340
- [37] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In *Proc. ICCV*, pp. 1042–1051, 2019. doi: 10.1109/ICCV.2019.00113
- [38] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. Pix2Vox: Context-aware 3D reconstruction from single and multi-view images. In *Proc. ICCV*, pp. 2690–2698, 2019. doi: 10.1109/ICCV.2019.00278
- [39] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. In *Proc. NeurIPS*, pp. 4805–4815, 2021.
- [40] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proc. NeurIPS*, pp. 2492–2502, 2020.
- [41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068