Volume Rendering of Neural Implicit Surfaces

Aqeel Alshakhori

Informatics - Technische Universität München

Abstract

Up until recently, a generic density function was used to model the geometry learnt by neural volume rendering approaches. The volume density function is described in this paper, Volume Rendering of Neural Implicit Surfaces, as Laplace's cumulative distribution function (CDF) applied to a representation of a signed distance function (SDF). There are three advantages to using this straightforward density representation: it gives the geometry a helpful inductive bias; it makes it easier to set a bound on the opacity approximation error, which results in an accurate sampling of the viewing ray. Effective unsupervised disentanglement of shape and appearance in volume rendering is made possible. Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman introduced this paper in December 2021 [5].

1 Introduction

Volume rendering is a group of methods for displaying volume density in radiance fields using what is known as the volume rendering integral. Recently, it has been demonstrated that modeling the density and radiance fields as neural networks may produce excellent predictions. However, this method of visualizing neural volumes has few drawbacks.

NeRF [3] has opened up a field of study integrating volume rendering by introducing neural implicit functions to provide photo-realistic rendering outcomes and it has low memory foot-print. This method, which was improved upon by its follow-ups, roughly represents the integral as alpha-composition. However, finding the right threshold to separate surfaces from the anticipated density is difficult, and the recovered geometry is not good enough.

Multi-view 3D surface reconstruction is another related field of work. Traditionally, it is either a depth-based or voxel-based. Depth-based suffers from complex pipeline which could accumelate errors through each stage. Voxel-basaed is limited to low resolution because its use of memory and requires accurate object masking.

A new model for the density in neural volume rendering is introduced by VolSDF [5]. The main concept is to show density as a function of signed distance to scene surface. Such a density function offers a clearly defined surface that creates density, among other advantages. Additionally, VolSDF enables bounding the opacity along ray approximation error.

2 Method Description

2.1 A New Density Function

Moving from neural implicit function in volume rendering to geometry-based function requires parameterization of the volume density. The parameterization is defined as a modified signed distance function. This specification starts by introducing a new density function σ that uses LaPlace distribution. Laplace distribution is useful where heterogeneity in the population is suspected, and the observations might show large errors which in our case sampling points in tracing ray from different poses.

$$\sigma(\boldsymbol{x}) = \alpha \Psi_\beta(-d_\Omega(\boldsymbol{x})) \tag{1}$$

The first part is the laplace distribution

$$\Psi_\beta(s) = \begin{cases} \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s \leq 0 \\ 1 - \frac{1}{2}\exp\left(-\frac{s}{\beta}\right) & \text{if } s > 0 \end{cases} \tag{2}$$

where α and β are learnable parameters, as the SDF get nears the object both parameters get decreased and the density converges to the object volume. The Signed Distance Function (SDF) is

$$d_\Omega(\boldsymbol{x}) = (-1)^{1_\Omega(\boldsymbol{x})} \min_{\boldsymbol{y} \in \mathcal{M}} \|\boldsymbol{x} - \boldsymbol{y}\| \tag{3}$$

where $1_\Omega(\boldsymbol{x})$ is a binary indicator $[1, 0]$ if the point $\boldsymbol{x}$ is inside the volume object. $\mathcal{M}$ is the surface boundary of the object and $\boldsymbol{y}$ is a point on that surface. This approach will pave the way to reconstruct the surface in a well-defined process.

2.2 Density-based Volume Rendering

Second step in the rendering process is rendering the volume of the density. A crucial method in rendering volume involves using ray tracing, where the light radiance from the ray is integerated to render the volume of the object. In calculating this step, two quantities are involved; namely the volume opacity and the radinace field.

Transperancy is defined as the propability that the ray will travel without hitting a surface along a certain segment.

$$T(t) = \exp\left(-\int_0^t \sigma(\boldsymbol{x}(s))ds\right) \tag{4}$$

where $\boldsymbol{x}$ is the ray, t is the segment it traveled without bouncing off, and s is the sampled points along the ray $\boldsymbol{x}$. Since we are more interested in the opacity (the complement of T) of the object, we could consider the opacity as a CDF and derive the PDF as follows and name it τ:

$$\frac{dO}{dt}(t) = \frac{d}{dt}\left(1 - T(t)\right) = \sigma(\boldsymbol{x}(t))T(t) = \tau(t) \tag{5}$$

Now we can integerate the light along the radiance field L through this formula:

$$I(\boldsymbol{c}, \boldsymbol{v}) = \int_0^\infty L(\boldsymbol{x}(t), \boldsymbol{n}(t), \boldsymbol{v}) \tau(t)\, dt \tag{6}$$

where $\boldsymbol{c}$ is the position of the camera eminating light from point $\boldsymbol{x}$ in the direction of unit vector $\boldsymbol{v}$, and $\boldsymbol{n}(t)$ is the surface normal since the problem is of bidirectional reflectance nature.

Since we are using samples of points along the ray, the paper uses rectangle rule approximation of the integral in equation 6:

$$I(\boldsymbol{c}, \boldsymbol{v}) = \hat{I}_{\mathcal{S}}(\boldsymbol{c}, \boldsymbol{v}) = \sum_{i=1}^{m-1} \hat{\tau}_i L_i \tag{7}$$

$\mathcal{S}$ in this equation is the set of discrete sample points where $s_1 = 0 < s_2 < ... < s_m = M$ and M is a large constant.

2.3 Bounding the Error on Opacity

Using approximation to calculate the integral will have a margin of error, and this error should be bound if we want to have a more accurate representation of the object. Using left Riemann summ to approximate the opacity $\hat{O}$:

$$\begin{aligned} \hat{O}(t) &= 1 - T(t) = 1 - \exp(-\hat{R}(t)), \\ \text{where } \hat{R}(t) &= \sum_{i=1}^{k-1} \delta_i \sigma_i + (t - t_k)\sigma_k \end{aligned} \tag{8}$$

Since this is rectangle rule, $\delta_i = t_{i+1} - t_i$ is the difference between two intervals over the ray $\boldsymbol{x}$. The density introduced in equation 1 provide a good basis to conclude the opacity approximation error. Deriving the Ψ_β function and bound it with Lipschitz constant will yeilds:

$$\left| \frac{d}{ds} \sigma(\boldsymbol{x}(s)) \right| \leq \frac{\alpha}{2\beta} \exp\left(-\frac{d_i^*}{\beta} \right), \text{ where } d_i^* = \min_{\substack{s \in [t_i, t_{i+1}] \\ \boldsymbol{y} \notin B_i \cup B_{i+1}}} \|\boldsymbol{x}(s) - \boldsymbol{y}\| \tag{9}$$

As illustrated in figure 1 [5], where B is the ball of the ray $\boldsymbol{x}$ at interval t, and d is the distance between the point and the surface. From here, the paper based the error E on the unsigned distance at the interval's two end points and density parameters α and β:

$$|E(t)| \leq \hat{E}(t) = \frac{\alpha}{4\beta} \left(\sum_{i=1}^{k-1} \delta_i^2 e^{-\frac{d_i^*}{\beta}} + (t - k_k)^2 e^{-\frac{d_k^*}{\beta}} \right) \tag{10}$$

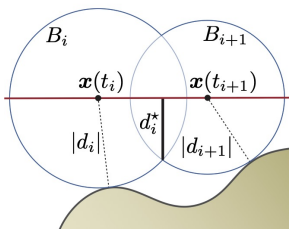

Figure 1: d_i^* as the minimal distance between two sets.

Now we have the approxiamtion for the integral $\hat{R}$ and the bounded opacity approximation error as follows for a period t:

$$|O(t) - \hat{O}(t)| \leq \exp\left(-\hat{R}(t_k)\right) \left(\exp(\hat{E}(t_{k+1})) - 1 \right) \tag{11}$$

However, we want to take into account all the intervals $\mathcal{T}$ and set the maximum as our bound B as a function of $\mathcal{T}$ and β:

$$\max_{t\in[0,M]} \left|O(t) - \hat{O}(t)\right| \leq B_{\mathcal{T},\beta} = \max_{k\in[n-1]} \left\{\exp\left(-\hat{R}(t_k)\right)\left(\exp(\hat{E}(t_{k+1})) - 1\right)\right\} \quad (12)$$

2.4 Sampling Algorithm

From previous equation, we can provide an ϵ for the opacity. However the choice of samples used in $\mathcal{T}$ plays a critical role in the quality of the integral equation 7. The paper suggests the adaptive sampling, i.e. by using the inverse CDF O^{-1}. The algorithm starts with uniform sampling $\mathcal{T}_0$ and two β, one above threshold and one below. Then using bisection method iteratively until β_* is found. This value is then used to esitimate $\hat{O}$. A fresh samples ($m = 64$) using the inverse sampling is returned to $\mathcal{S}$.

Algorithm 1: Sampling algorithm.

Input: error threshold $\epsilon > 0$; β

Initialize $\mathcal{T} = \mathcal{T}_0$
Initialize β_+ such that $B_{\mathcal{T},\beta_+} \leq \epsilon$
while $B_{\mathcal{T},\beta} > \epsilon$ *and not max_iter* **do**
    upsample $\mathcal{T}$
    if $B_{\mathcal{T},\beta_+} < \epsilon$ **then**
        Find $\beta_\star \in (\beta, \beta_+)$ so that $B_{\mathcal{T},\beta_\star} = \epsilon$
        Update $\beta_+ \leftarrow \beta_\star$
    end
end
Estimate $\hat{O}$ using $\mathcal{T}$ and β_+
$\mathcal{S} \leftarrow$ get fresh m samples using $\hat{O}^{-1}$
return $\mathcal{S}$

2.5 Training

Training setting used two MLP; one is for approximating SDF of learned geometry and global geometry feature z with 8 layer of width 256: $f_\varphi(x) = (d(x), z(x)) \in \mathbb{R}^{1+256}$. The second MLP is deployed to present the secene's radiance field: $L_\psi(x, n, y, z) \in \mathbb{R}^3$ with learnable parameter ψ. Two scalar learnable parameters $\alpha, \beta \in \mathbb{R}$, with $\alpha = \beta^{-1}$. And the positional encoding for x and v, was the same as NeRF.

The data were images collected from camera at different positions. Each pixel in these images has three values: $(I_p, \boldsymbol{c}_p, \boldsymbol{v}_p)$, the first one is the RGB color intensity, the second the position of the camera and the third is the viewing direction. The loss consist of two parts; one for RGB color loss and the second is Eikonal loss, which showed from previous work that it results in a solution that is close to a signed distance function[1]:

$$\mathcal{L}(\theta) = \mathcal{L}_{RGB}(\theta) + \lambda\mathcal{L}_{SDF}(\varphi) \quad (13)$$

where θ is the set of all learnalble pararmeters $\theta = (\varphi, \psi, \beta)$, and λ is a hyper-parameter set to 0.1.

3 Experiments and Results

Two datasets were used for testing: DTU [2] and BlendedMVS [4]. DTU is a multi-view image of different objects with fixed camera and lighting parameters. BlendedMVS has a large collection of scenery that can be used as a high quality ground truth, however 9 scenes were selected.

	Scan	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
Chamfer Distance	**IDR**	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.90	0.42	0.51	0.53	0.90
	colmap$_7$	0.45	0.91	0.37	0.37	0.90	1.00	0.54	1.22	1.08	0.64	0.48	0.59	0.32	0.45	0.43	0.65
	colmap$_0$	**0.81**	2.05	**0.73**	1.22	1.79	1.58	1.02	3.05	1.40	2.05	1.00	1.32	0.49	0.78	1.17	1.36
	NeRF	1.92	1.73	1.92	0.80	3.41	1.39	1.51	5.44	2.04	1.10	1.01	2.88	0.91	1.00	0.79	1.89
	VolSDF	1.14	**1.26**	0.81	**0.49**	**1.25**	**0.70**	**0.72**	**1.29**	**1.18**	**0.70**	**0.66**	**1.08**	**0.42**	**0.61**	**0.55**	**0.86**
PSNR	**NeRF**	26.24	25.74	26.79	27.57	31.96	31.50	29.58	32.78	28.35	32.08	33.49	31.54	31.0	35.59	35.51	30.65
	VolSDF	26.28	25.61	26.55	26.76	31.57	31.5	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.9	34.75	30.38

Table 1: Quantitative results for the DTU dataset.

The upper part of the table 1 [5] is the surface accuracy measured using the Chamfer l1 loss (measured in mm). COLMAP0 is a watertight reconstruction pipeline with wide selection of features. IDR [6] is the state of the art 3D surface reconstruction method using implicit representation. We can see that VolSDF performs on par with IDR and outperforms NeRF and COLMAP in terms of reconstruction accuracy. The lower part is PSNR (peak signal-to-noise ratio) which is used to systematically compare different algorithms, shows that VolSDF is comparable with NeRF.

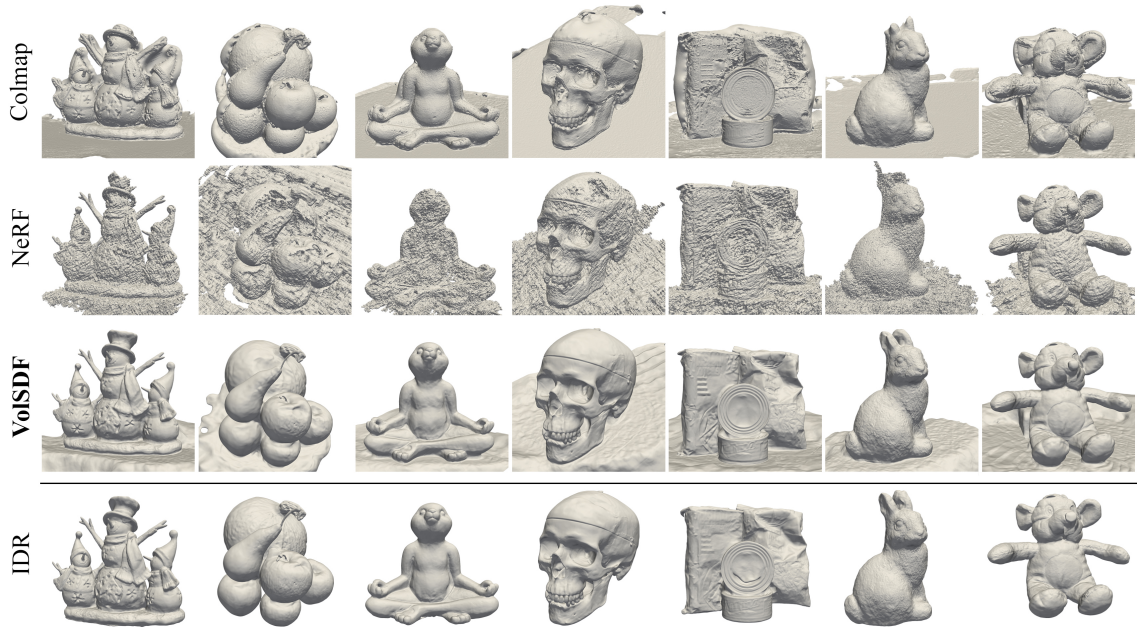

Figure 2: Qualitative results for the DTU dataset.

Qualitative results for reconstructed geometries of objects from the DTU dataset in Figure 2 [5], where we see that NeRF struggle with the details while VolSDF is yet comparable with the STOA IDR.

	Scene	Doll	Egg	Head	Angel	Bull	Robot	Dog	Bread	Camera	**Mean**
Chamfer l_1	Our Improvement (%)	54.0	91.2	24.3	75.1	60.7	27.2	47.7	34.6	51.8	51.8
PSNR	**NeRF++**	26.95	27.34	27.23	30.06	26.65	26.73	27.90	31.68	23.44	27.55
	VolSDF	25.49	27.18	26.36	29.79	26.01	26.03	28.65	31.24	22.97	27.08

Table 2: Quantitative results for the BlendedMVS dataset.

In BlendedMVS, the paper compared VolSDF with NeRF++ [7] since the dataset has more complex backgrounds. In table 2 [5], VolSDF performs on par with NeRF++ and the improvement of Chamfer distance is compared to NeRF. In figure 3 [5], NeRF++ shows artifacts and grains on the surfaces while VolSDF shows improved reconstructions and more faithful results.

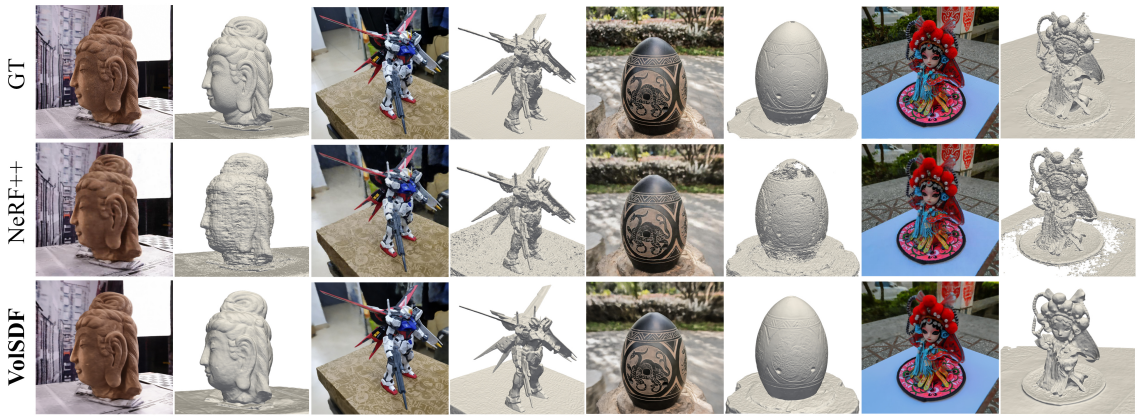

Figure 3: Qualitative results sampled from the BlendedMVS dataset.

4 Discussion / Conclusion

NeRF [3] introduced a neural volume rendering approach, which showed a significant potential. The paper was followed up with several related works. However, in their approach, the density component generally yields noisy, imprecise geometry approximations since it is less adept at accurately anticipating the scene's true geometry. VolSDF [5] introduced a better approach by taking into account the geomatry side of the problem. This results in a quality of view synthesis. As shown in the qualitative and quantitative results, implicit function approach struggles with the details of the object's surface, and in some cases it loses the details. VolSDF provides a promosing approach in volume rendering area. However, VolSDF assumes homogeneous density of the object which limits the classes of geometry that can modeled. Secondly, homogeneous texture-less areas are hard to reconstruct faithfully, which can be avoided by adding an extra assumption before the reconstruction.

References

[1] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*,

2020.

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

[3] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[4] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.

[5] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

[6] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

[7] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.