Implicit neural representations

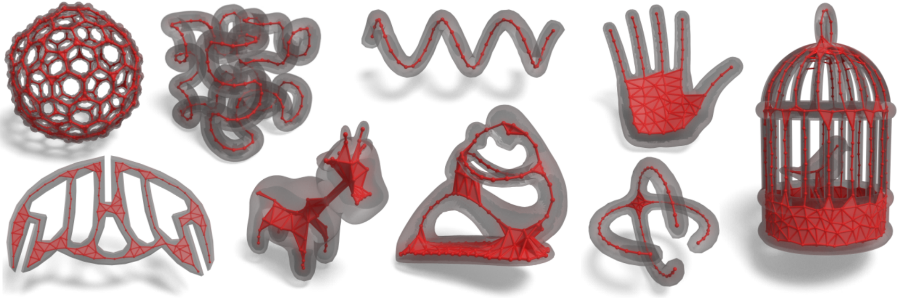
Julie Digne

Master ID3D
LIRIS - CNRS
Équipe Origami

14/10/2025

Outline

Regularizing INR away from the surface

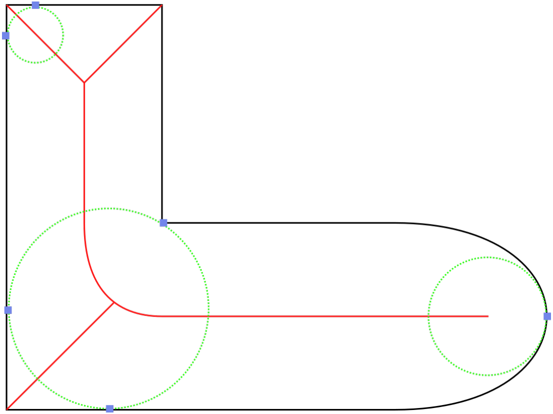


[Clémot, Digne 2023]

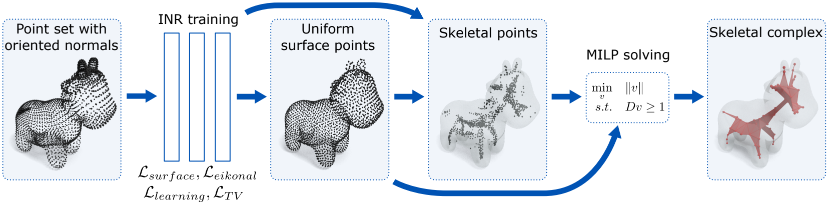
Medial Axis

Definition

A point p belongs to the medial axis of a compact shape if it has at least two distinct nearest neighbors on the shape surface.



Overview



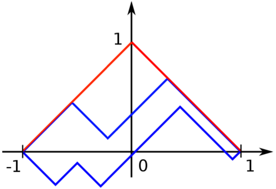
Point set with oriented normals
INR training
$\mathcal{L}_{surface}, \mathcal{L}_{eikonal}$
$\mathcal{L}_{learning}, \mathcal{L}_{TV}$
Uniform surface points
Skeletal points
MILP solving
$\min_v \; \|v\|$
$s.t. \; Dv \geq 1$
Skeletal complex

Eikonal Equation

- Infinite number of solutions
- Viscosity solution theory: allows to select the right solution
- Use smooth eikonal equation (not practical [Lipman 2019])

$$\|\nabla u\| - \varepsilon \Delta u = 1$$

- Consequence: blobs appear



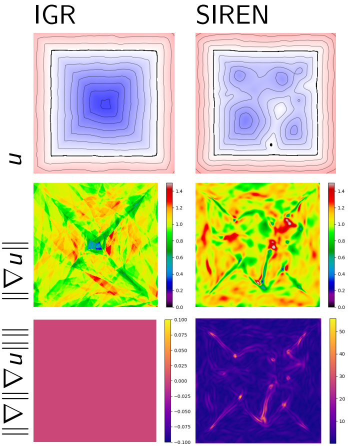
[Camilli 2014]

Infinite nber of solutions

Not an issue close to the surface – but far away?

Which neural network?

- MLP (6 layers, 128-256 neurons/layer) with ReLU activation functions
- ReLU yields a function in $W^{1,p}$ [Lipman 2019]
- But: not always easy to train
- Sitzman (2021) replaces ReLU with sine activation function: smooth function



TV regularization - small analysis

- Look for a smooth surrogate for the signed distance function
- Medial axis: zeros of the gradient
- The TV term favors that u has no second order differential content along the gradient lines

Since $\nabla u = (u_x, u_y, u_z)$, it follows:

$$\begin{aligned}
\nabla \|\nabla u\| &= \nabla \sqrt{u_x^2 + u_y^2 + u_z^2} \\
&= \frac{1}{2\|\nabla u\|} \begin{pmatrix} 2u_x u_{xx} + 2u_y u_{xy} + 2u_z u_{xz} \\ 2u_x u_{xy} + 2u_y u_{yy} + 2u_z u_{yz} \\ 2u_x u_{zx} + 2u_y u_{zy} + 2u_z u_{zz} \end{pmatrix} \\
&= H_u \frac{\nabla u}{\|\nabla u\|}
\end{aligned}$$

Total loss

- Eikonal loss:

$$\mathcal{L}_{eikonal} = \int_{\mathbb{R}^3} (1 - \|\nabla u(p)\|)^2 \, dp \tag{1}$$

- Surface loss:

$$\mathcal{L}_{\text{surface}} = \int_{\partial\Omega} u(p)^2 dp + \int_{\partial\Omega} 1 - \frac{\mathrm{n}(p) \cdot \nabla u(p)}{\|\mathrm{n}(p)\| \, \|\nabla u(p)\|} dp \tag{2}$$

- Learning point loss

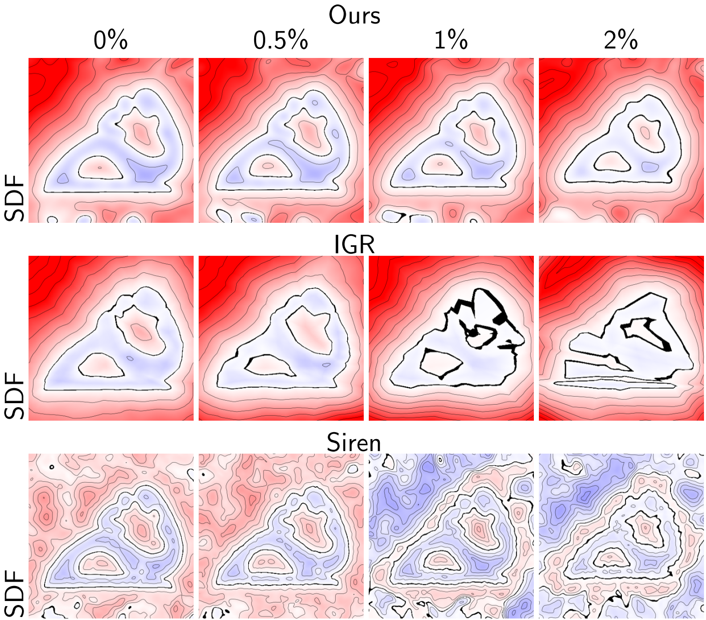
$$\mathcal{L}_{\text{learning}} = \sum_{p \in \mathcal{P}} (u(p) - d(p))^2 + \sum_{p \in \mathcal{P}} 1 - \frac{\nabla u(p) \cdot \nabla d(p)}{\|\nabla u(p)\| \, \|\nabla d(p)\|} \tag{3}$$

- + TV loss

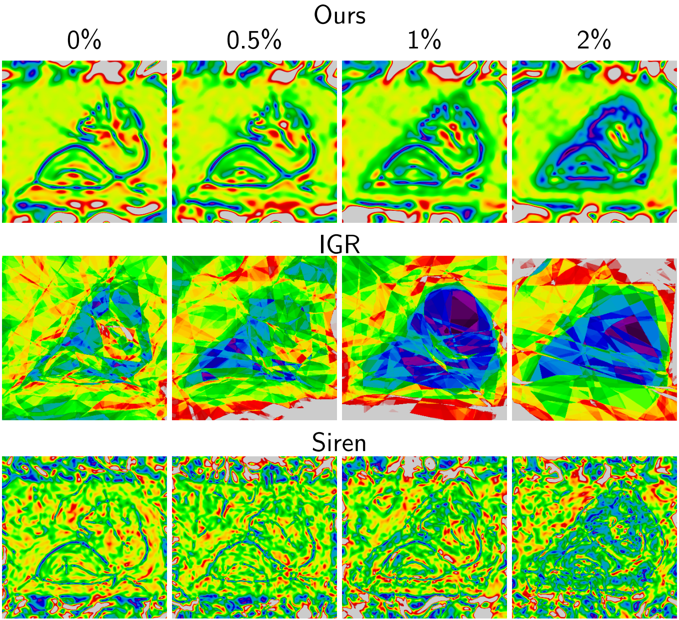
Loss

$$\mathcal{L} = \lambda_e \mathcal{L}_{eikonal} + \lambda_s \mathcal{L}_{surface} + \lambda_l \mathcal{L}_{learning} + \lambda_{TV} \mathcal{L}_{TV} \tag{4}$$

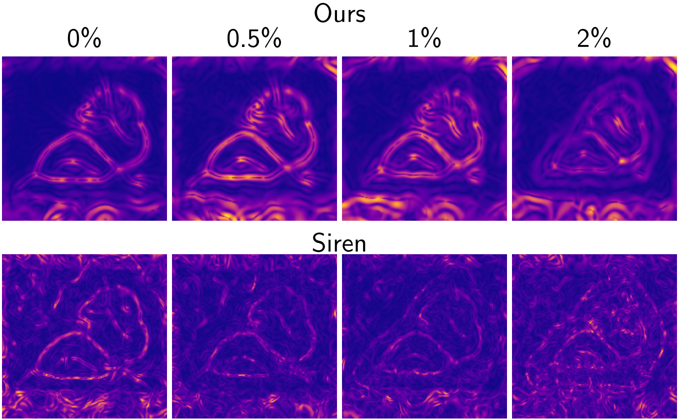
Resulting fields with increasing noise



$\|\nabla u\|$



$\nabla \|\nabla u\|$



then...

- GPU skeleton tracing to extract points on the skeleton
- Select a subset based on the Coverage Axis method [Dou 2022]
  - N points x_i, M skeletal points s_i with distance r_i to the surface.
  - Coverage matrix: D $(N \times M)$

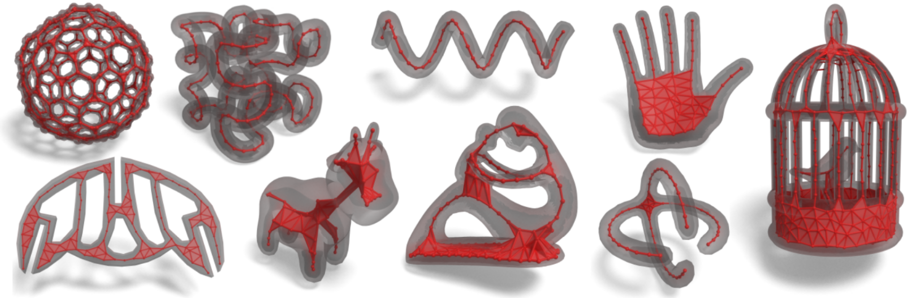
$$D_{ij} = 1 \text{ if } \|p_i - s_j\| - r_j \leq \delta \text{ and } 0 \text{ otherwise}$$

  - Mixed Integer Linear Problem:

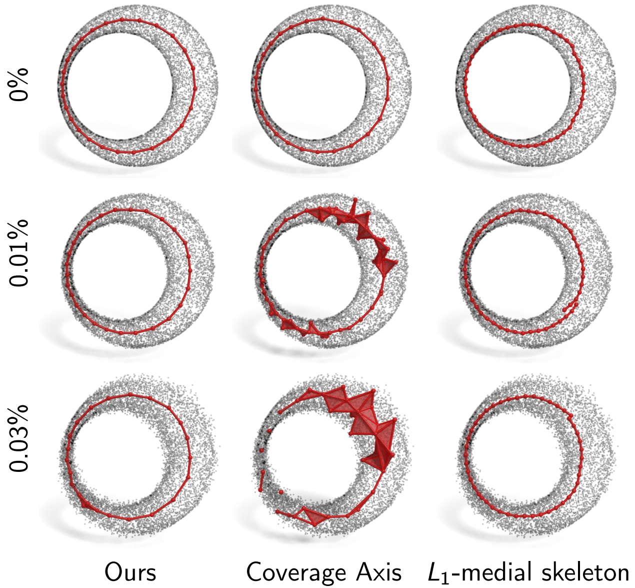
$$\begin{aligned} \min \quad & \|v\|_2 \\ \text{s.t.} \quad & Dv \succeq 1 \end{aligned} \tag{5}$$

- Link the selected points by computing the regular triangulation of weighted skeletal points and surface points + keep simplices between skeletal points

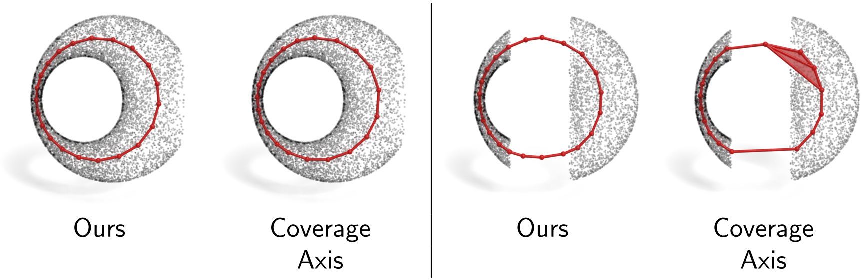
Results



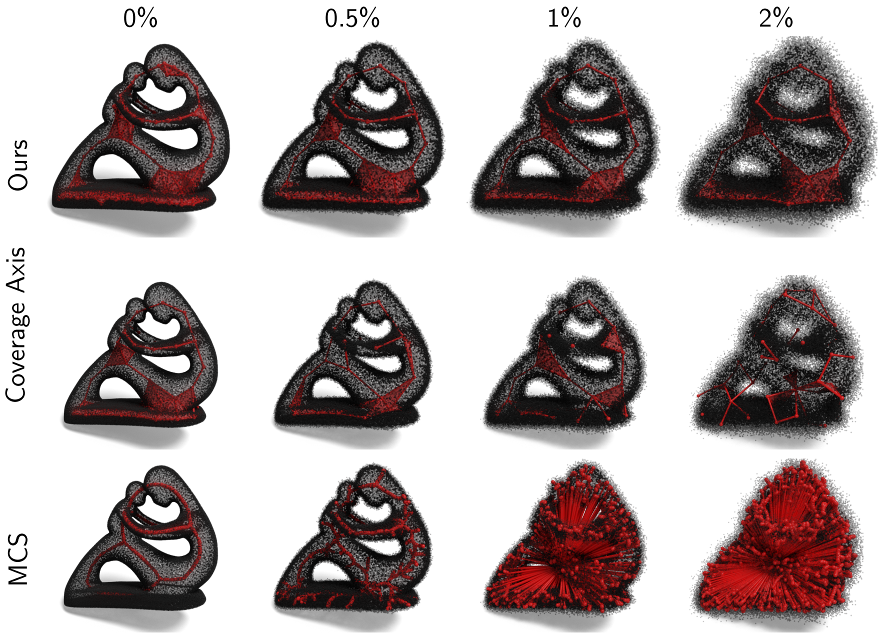
Results



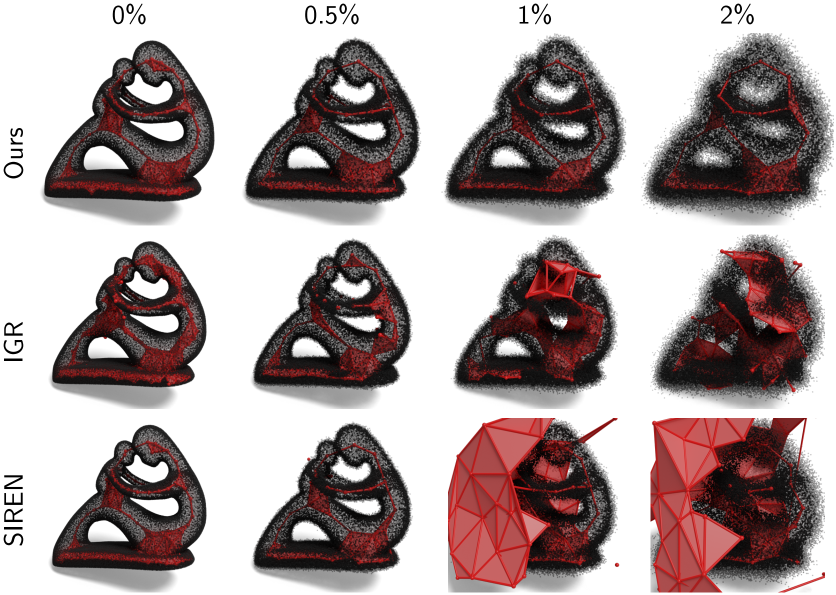
Results



With noise



With noise



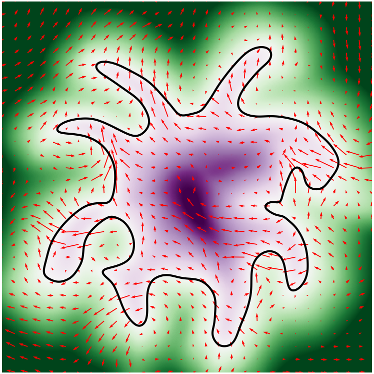
Outline

Shape Interpolation [Buonomo et al. 2025]

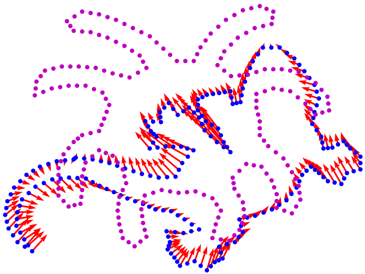
Approach

Interpolation with a neural implicit function in space and time:

- As SDF as possible at each time
- Volume preserving neural deformation field



Deformation Vector Field



Given a vector field V deforming a shape $\mathcal{S}_0$ into $\mathcal{S}_1$
Motion of point p (ODE):

$$\dot{p}(t) = \mathsf{V}(p(t), t)$$

Level-set equation

A function $f(x, t)$ encodes $\mathcal{S}_t$ ($\forall t \in [0, 1]$) if :

$$f(p(t), t) = 0$$

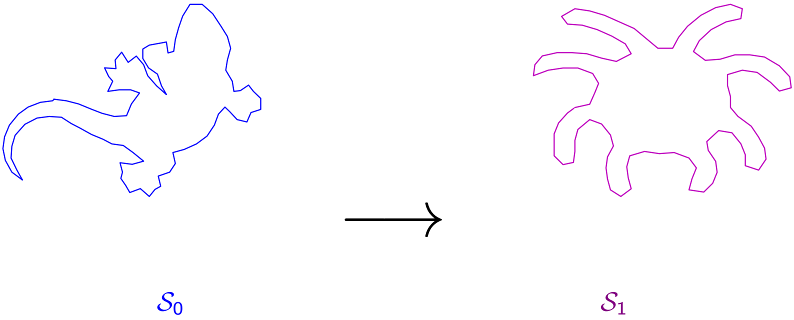
Differentiation :

$$\frac{df(p(t), t)}{dt} = \frac{\partial f}{\partial t}(p(t), t) + \langle \nabla f(p(t), t), \dot{p}(t) \rangle = 0$$

$$\Rightarrow \frac{\partial f}{\partial t}(x, t) + \langle \nabla f(x, t), V(x, t) \rangle = 0$$

See e.g. [Osher 1988]

Level-set equation

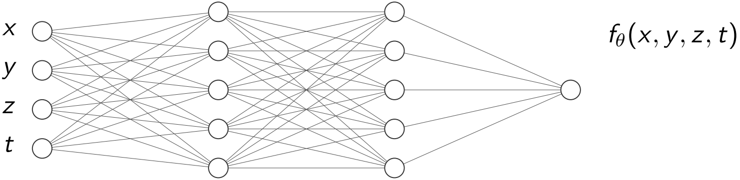


$$\begin{cases} \frac{\partial f}{\partial t} + \langle \nabla f, V \rangle & = 0 \text{ on } \mathbb{R}^d \times [0,1] \\ f(x,0) & = g_0(x) \ \forall x \in \mathbb{R}^d \\ f(x,1) & = g_1(x) \ \forall x \in \mathbb{R}^d \end{cases}$$

Neural solution of the level-set equation

Neural network f_θ:

- Siren network ([Sitzmann 2022])
  - Infinitely differentiable
  - High representative power
  - Light network (128×6)
- Derivatives by auto-grad



Training

LSE loss:

$$l_{LSE} = \int_{\mathbb{R}^d \times [0,1]} \mid \frac{\partial f_\theta}{\partial t} - \langle \nabla f_\theta, V \rangle \mid \mathrm{d}p\mathrm{d}t$$

Eikonal loss :

$$l_{Eik} = \int_{\mathbb{R}^d \times [0,1]} \mid 1 - \mid \nabla f_\theta(p,t) \mid\mid \mathrm{d}p\mathrm{d}t$$

Boundary conditions :

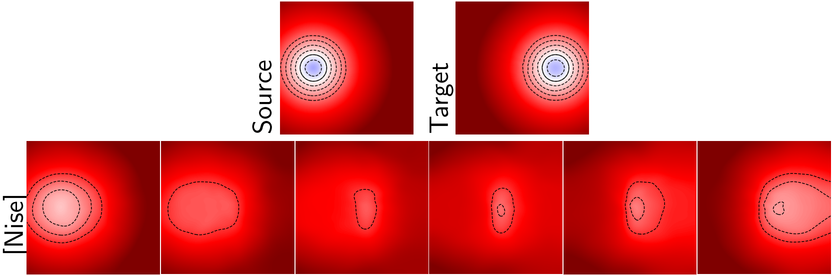
$$l_{Dirichlet} = \sum_{i=0,1} \int_{\mathbb{R}^d} \mid g_i - f_\theta(.,i) \mid \mathrm{d}p + \sum_{i=0,1} \int_{x \in \partial \mathcal{S}_i} |f_\theta(x,i)| dx$$

$$l_{Neumann} = \sum_{i=0,1} \int_{\mathcal{S}_i} \mid 1 - \langle \nabla f_\theta(.,i), \vec{N}_{\mathcal{S}_i} \rangle \mid \mathrm{d}p$$

LSE with handcrafted V (Nise) [Novello 2023]

Ad hoc expression :

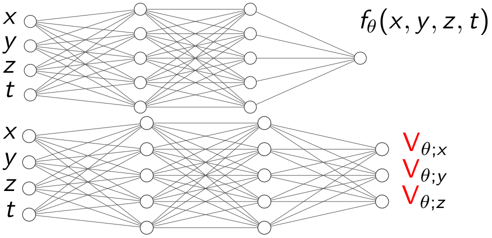
$$V(p,t) = -(g_1(p) - f_\theta(p,t))\frac{\nabla f_\theta(t,p)}{\|\nabla f_\theta(t,p)\|}$$



Joint learning of f_θ and V_θ

2 Siren networks trained jointly :

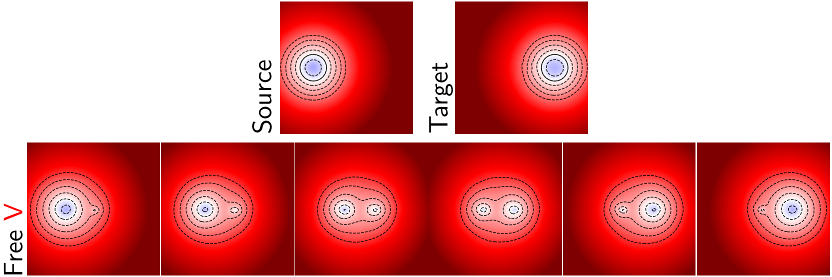
- Estimate f_θ and V_θ with end-to-end training



Updated LSE loss:

$$l_{LSE} = \int_{\mathbb{R}^d \times [0,1]} \mid \frac{\partial f_\theta}{\partial t} - \langle \nabla f_\theta, V_\theta \rangle \mid \, \mathrm{d}p\mathrm{d}t$$

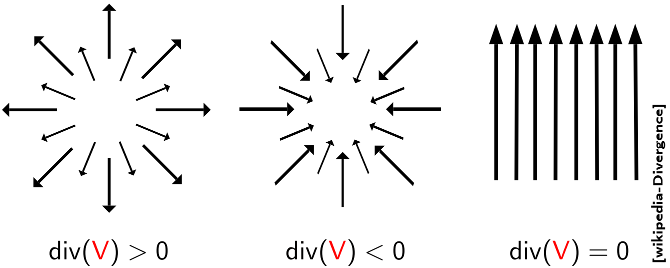
Joint learning of f_θ and V_θ



Volume preservation

Sufficient condition

If div $\textcolor{red}{V} = 0$, then the shape advected by $\textcolor{red}{V}$ has constant volume.



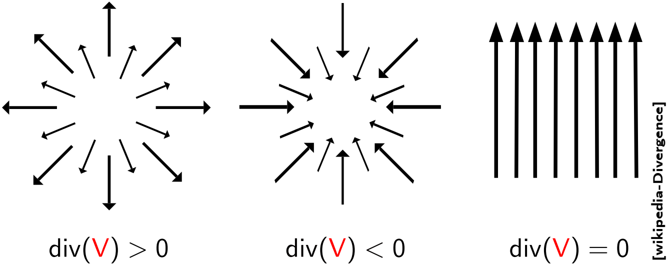
Practical construction

For any smooth vector field $D \in \mathbb{R}^3$, $\operatorname{div} \operatorname{curl} D = 0$

Volume preservation

Sufficient condition

If $\operatorname{div} V = 0$, then the shape advected by V has constant volume.



Practical construction

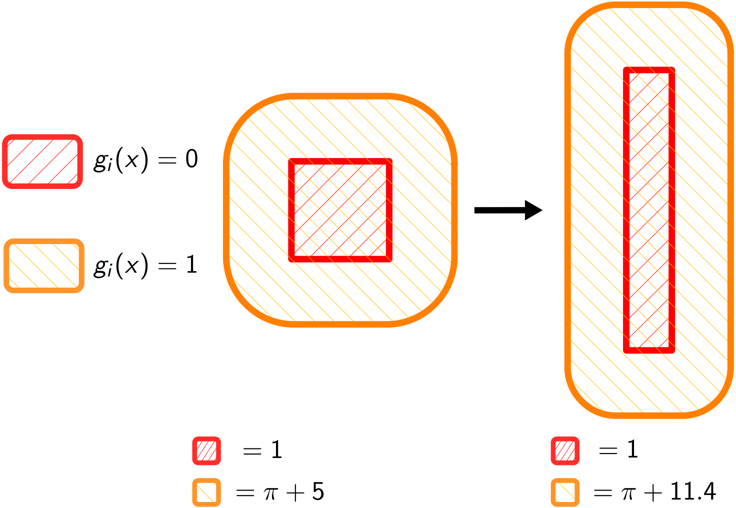
For any smooth vector field $D \in \mathbb{R}^3$, $\operatorname{div} \operatorname{curl} D = 0$

- Estimate D and set $V = \operatorname{curl} D$.

See also [Richter-Powell 2022].

Incompatibility Eikonal/Divergence free field

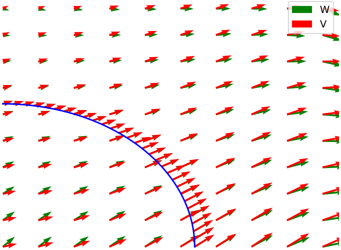
If div(V) = 0, V preserves the volume of every level-sets



Only the preservation of the 0-levelset is necessary.

Adaptive divergence : intuitive definition

Idea : Augment a divergence free vector field W by a quantity that vanishes along $\partial\mathcal{S}_t$.



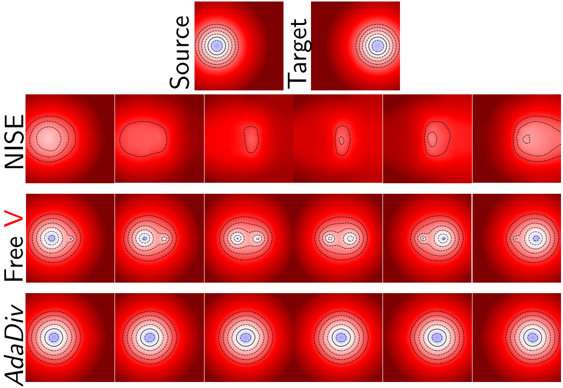
See [Buonomo et al. 2025] for the full mathematical construction

Volume preservation theorem

Theorem (Volume Preservation)

If $\color{red}{V}$ *has adaptive divergence w.r.t.* $\mathcal{S}$ *and corresponding flow* $\phi_{\color{red}{V}}$*:*

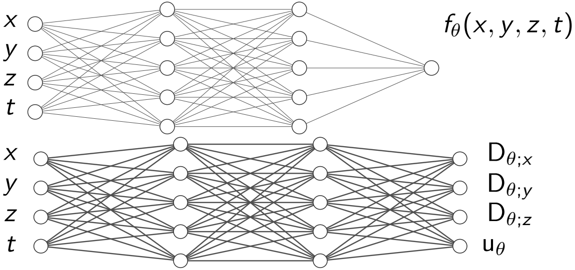
$$\mathrm{d}_t \mathit{Vol}(\phi_{\color{red}{V}}(\mathcal{S}, t)) = 0$$



Model Architecture

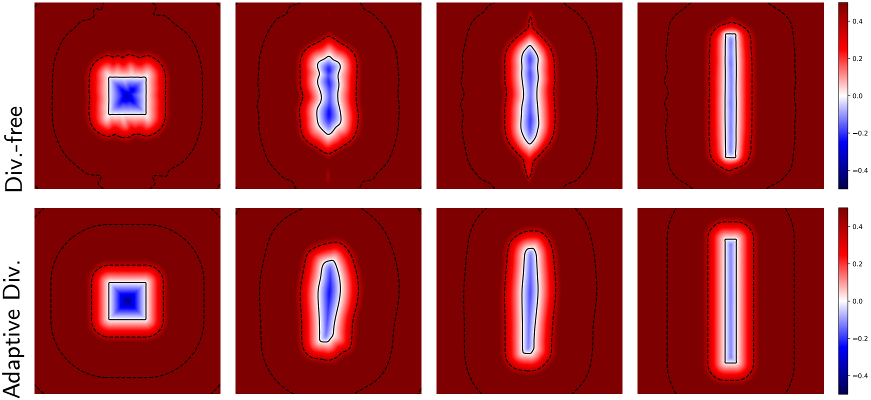
2 Siren networks trained jointly end-to-end.

- Solution continuous in both time and space.
- Volume preservation via the architecture of V_θ

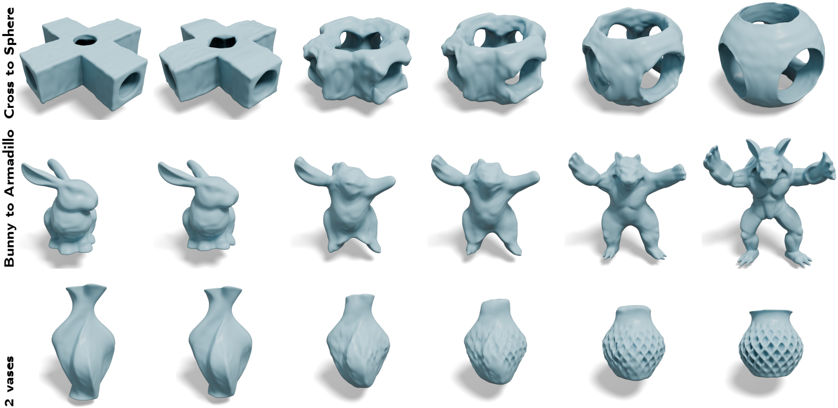


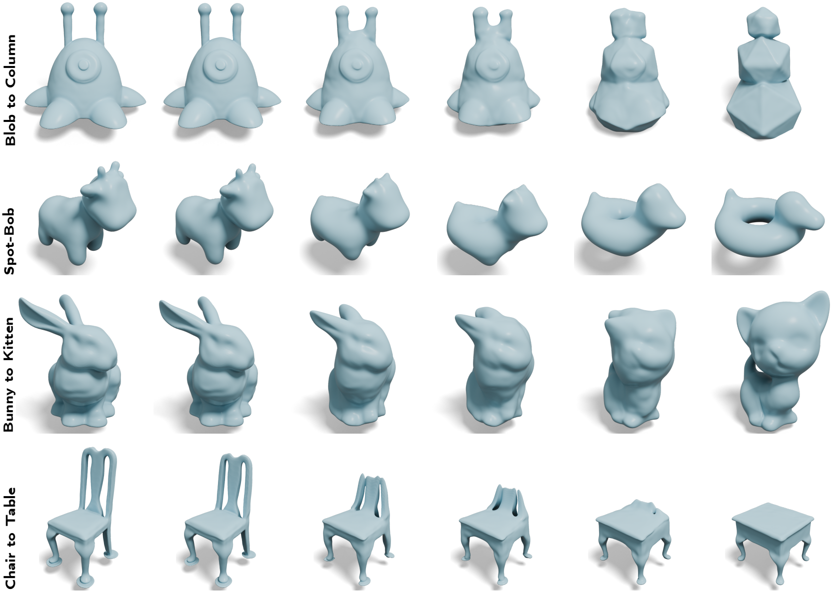
$$\mathrm{V}_\theta = \mathrm{curl}(\mathrm{D}_\theta) + \beta(f_\theta)\nabla u_\theta$$

Result in 2D



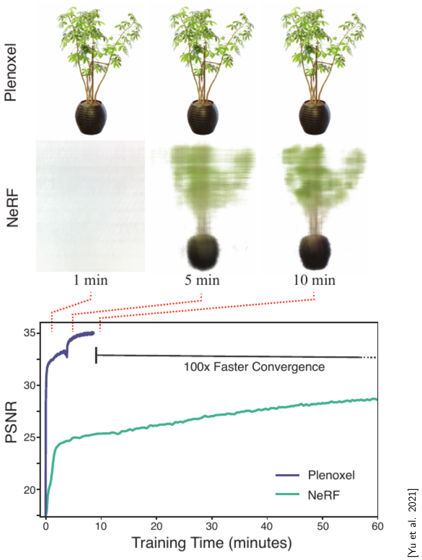
Results




Blob to Column
Spot-Bob
Bunny to Kitten
Chair to Table

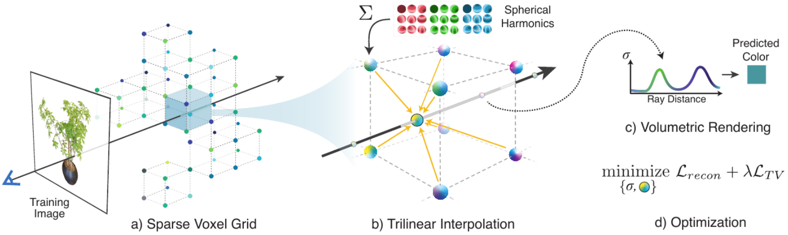
Outline

After Nerf... Plenoxels [Yu et al. 2021]



- No neural net
- (way) faster than nerf

Method



[Yu et al. 2021]

Spherical harmonics

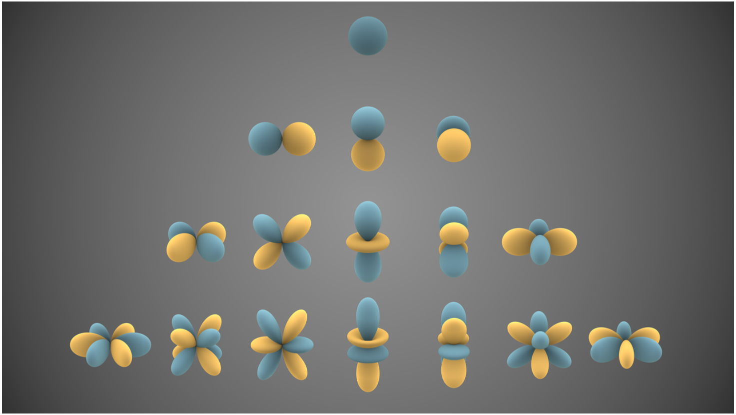


Image by Inigo Quilez (Wikipedia)

$$Y_l^m(\theta, \varphi) = e^{im\varphi} P_l^m(\cos(\theta))$$

- P_l^m Associated Legendre polynomial

$$P_l^m(x) = (-1)^m (1 - x^2)^{m/2} \sum_{k=m}^{l} \frac{k!}{(k-m)!} x^{k-m} \binom{l}{k} \binom{(l+k-1)/2}{l}$$

Color and spherical harmonics

- Spherical harmonics of degree 2 $\rightarrow$ 9 coefficients per color channel
- Color $C(r)$ = sum of the spherical harmonics evaluated in the ray direction
- Estimation on the vertices of a sparse grid and linear interpolation per grid cell.

Losses

- Optimization on SH coefficients and density minimizing the Loss:

$$\mathcal{L}_{recon} + \lambda \mathcal{L}_{TV}$$

- Reconstruction Loss:

$$\mathcal{L}_{recon} = \sum_{r \in \mathcal{R}} \| C(r) - \hat{C}(r) \|_2^2$$

- TV Loss:

$$\mathcal{L}_{TV} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}, d \in \mathcal{D}} \sum_i \| \nabla_x SH_i \|_2 + \| \nabla_x \sigma \|_2$$

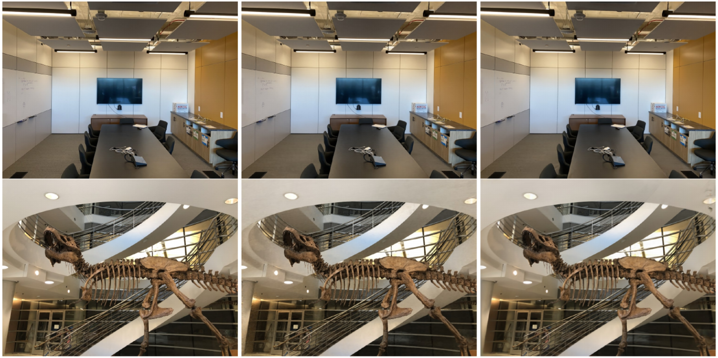
($\mathcal{V}$ and $\mathcal{R}$ stochastic samplings of the grid vertices and rays)

Results

Ground Truth NeRF++ [57] Plenoxels

[Yu et al. 2021]

Results



(d) Ground Truth (e) JAXNeRF (f) Plenoxels

[Yu et al. 2021]

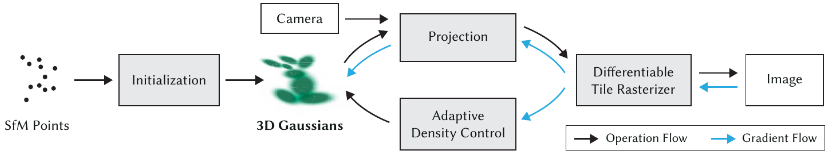
- Insight: What makes nerf work is not the neural net but *Differentiable* rendering.

Outline

Gaussian Splatting

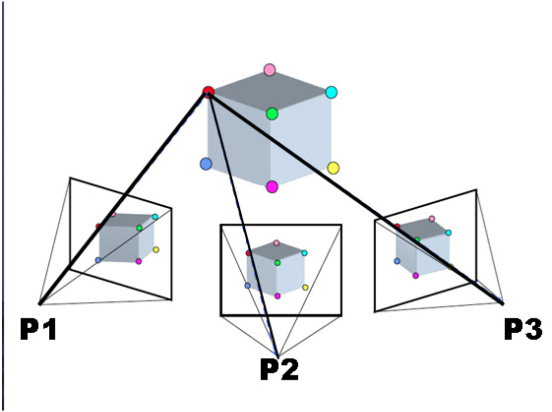
- Build on point set Splatting [Zwicker 2001]
- Each point is the center of a small 3D Gaussian on it,
- Each 3D Gaussian is represented by a quaternion and 3 scaling factors.
- Gaussian splat = gaussian parameters + opacity + Spherical harmonics

Overview



[Kerble 2023]

Structure from Motion (SfM)



[Maiteng - Wikipedia]

- Cameras calibrated by Structure from Motion [Snavely 2006]

Rendering a Gaussian splat scene

- Projective space Gaussian giving the color.

$$G(x) = \exp -x^T \Sigma^{-1} x \to G'(x) = \exp -x^T \Sigma'^{-1} x$$

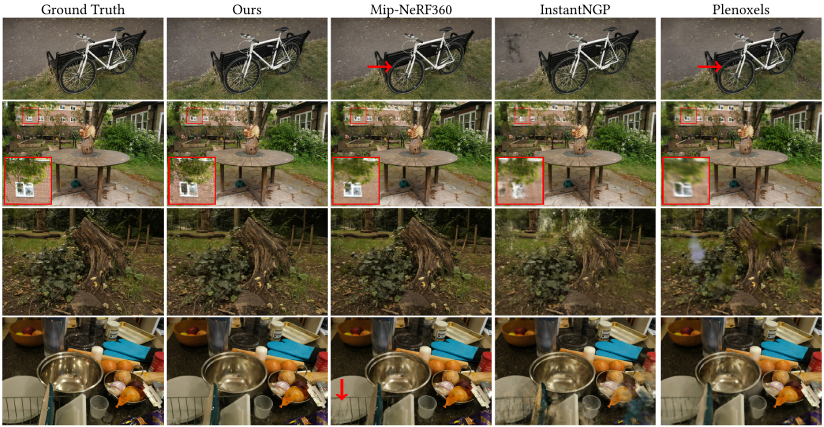
- Viewing direction W $\Sigma' = JW\Sigma W^T$
- J jacobian of the affine approx of the projective transformation:

$$J = \begin{pmatrix} f_x/z & 0 & -f_x t_x/z^2 \\ 0 & f_y/z & -f_y t_y/z^2 \\ 0 & 0 & 0 \end{pmatrix}$$

Rasterizer

- Split screen in tiles
- Cull 3d Gaussians against view frustrum
- Each tile = depth sorted Gaussians
- When saturation level is reached: stop

Creating or Destroying Geometry



[Kerble 2023]

Number of iterations

[Kerble 2023]

Conclusion

- Overview of *Single shape* implicit representation techniques
- Signed distance field or occupancy function or ??
- Nerf/Gaussian Splat: do we need to compute the geometry or only render?
- Multi-resolution, levels of details for neural implicits.