Implicit Representation Is All ~~You Need~~ I Know

Introduction to Differentiable Implicit Surface

Binbin Huang

ShanghaiTech University

2021

Overviews

Introduction to implicit geometry (definition, discretization, ray marching, ...)

Properties of implicit function (normals, sdf, ...)

Preliminary (mathematics, neural networks, ...)

Differentiale Rendering (volume, surface, regularization, ...)

Overviews

Introduction to implicit geometry (definition, discretization, ray marching, ...)

Properties of implicit function (normals, sdf, ...)

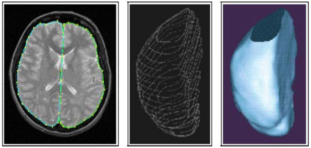
Preliminary (mathematics, neural networks, ...)

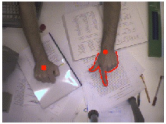
Differentiale Rendering (volume, surface, regularization, ...)

Surface

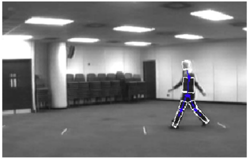
Tracking boundaries is an important problem.

Image segmentation

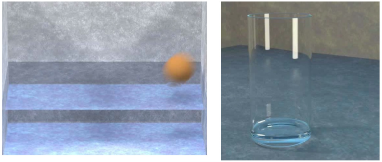




Human Computer Interaction



Machine Vision



Physical Simulation

The problem is how it is possible to efficiently and accurately represent the boundary?

http://faculty.missouri.edu/duanye/course/cs8620-spring-2017/lecture-notes/3b-signed-distance-function.pdf

Explicit Representation

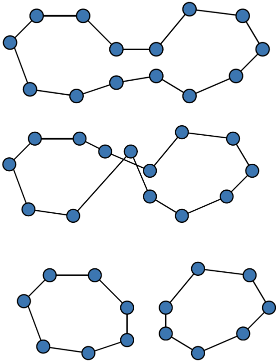
Explicit Representation or Discrete representation.

Marker

- The idea is to represent boundaries with particles.
- more particles should mean greater accuracy.

Issue (when deforming boundaries)

- markers may potentially cross over one another.
- the boundary may split or merge.

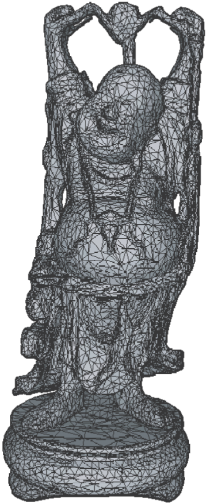


Explicit Representation

Explicit Representation or Discrete representation.

Marker (mesh for example.)

- For complicated surfaces in 3D, the explicit representation can be quite difficiult to discretize. One needs to choose a number of points on the 2d surface and record their connectivity.
- Connectivity can change for dynamic/deformable surfaces.

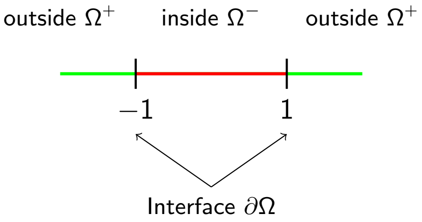


Explicit Representation

Explicit Functions

- Explicit functions directly tell you where is the boundary.

Considering a 1-D example

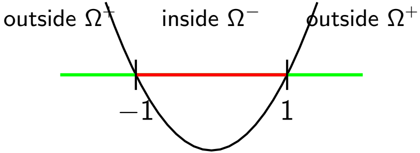


Explicit Representation:
$\Omega^- = (-1, 1)$
$\Omega^+ = (-\infty, -1) \cap (1, \infty)$
$\partial\Omega = \{-1, 1\}$

Implicit Representation

Implicit Functions

- Implicit divide a 1-d regions Ω with a function.



Implicit Representation
$\phi(x) = x^2 - 1$
$\Omega^- : \phi(x) < 0$
$\Omega^+ : \phi(x) > 0$
$\partial\Omega : \phi(x) = 0$

Implicit Representation

Impicit Functions

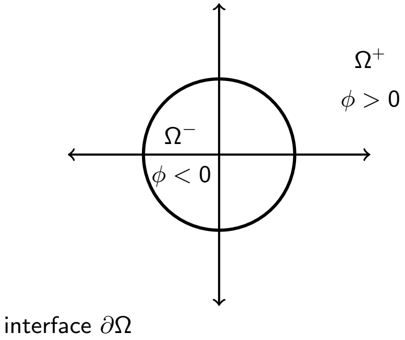
Points

- An implicit representation provides a simple, numerical method to determine the interior, exterior and interface of a region Ω.
- The set of points where $\phi(x) = 0$ is known as the **zero level set** or zero isocontour and will always give the interface of the region.
- In regions R^n, subdomains are n-dimensional, while the interface has dimension n-1.

Implicit Functions

Examples of Implicit Functions.

- Curves
  In 2 spatial dimmensions a closed curve separates Ω into two sub-regions.

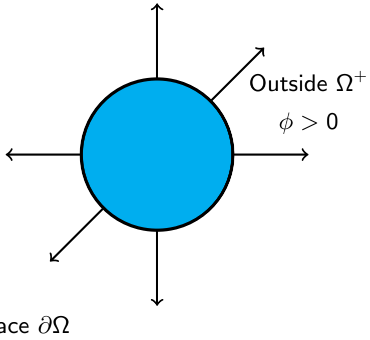


$$\phi((\vec{x})) = x^2 + y^2 - 1 = 0$$

Implicit Functions

Examples of Implicit Functions.

- Surfaces
  In 3+ spatial dimensions higher dimensional surfaces are used to represent the zero level set of ϕ.



$$\phi((\vec{x})) = x^2 + y^2 + z^2 - 1 = 0$$

Implicit Functions

Explicit Representation VS. Implicit Representation

Explicit:

- Pros: easier display.
- Cons: maintain connectivity, difficult to handle topology change.

Implicit:

- Pros: No connectivity, easier to handle topology change.
- Cons: Need extra step to display.

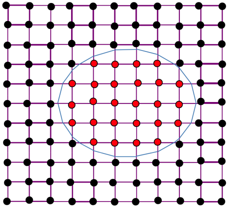
Remark

Explict representation as Lagrangian Mesh.
Implicit representation as Eulerian Mesh.
Not detail here (Not necessary and IDK); But The concepts may help when you do something like fluid simulation.

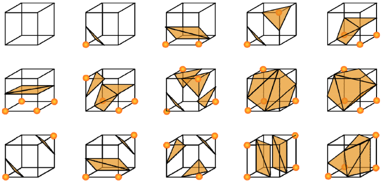
Implicit Functions

Discrete Representations of Implicit Function.

- Iso-contour extraction

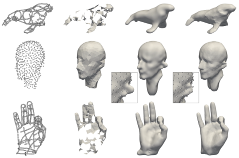


- Marching Cubes Algorithm



Lorensen & Cline: *Marching Cubes: A high resolution 3D surface construction algorithm.* In SIGGRAPH 1987.

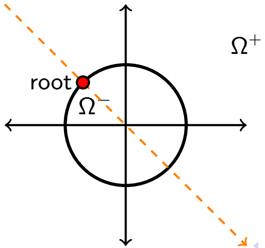
How Discrete representation to Implicit Function?



SAL: Sign Agnostic Learning of Shapes from Raw Data. CVPR2020.

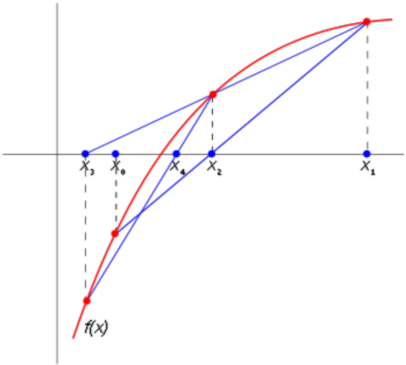
Ray marching in implicit surfaces

- The surface (boundary) is $\{x : f(x) \equiv 0\}$.
  A ray can be parameterized with $x = c + \vec{v}t(x)$, where c, v, t is camera, view direction, depth respectively.
- We aim to find the first intersection of the ray and the surface, i.e $min_t f(c + \vec{v}t) \equiv 0$.
- There are many numerical methods for root-finding algorithms. Here mainly introduce two:
  - Bisection (binary search)
  - Newton's method
  - Secant
  - Sphere Tracing (specific for SDF)



Ray marching in implicit surfaces

- secant
  $x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}$

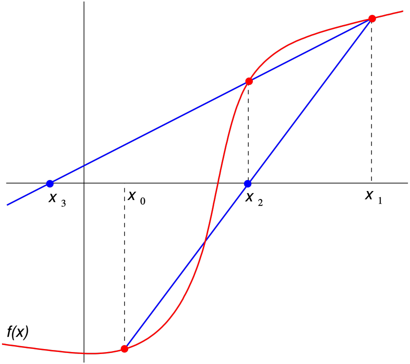


Remark

Broyden's method for k-variables equation. (may useful for tracing intersections with high-dimension lights, i.g. wavefront.)

Ray marching in implicit surfaces

- secant
  $x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}$



Remark

Not gurantee convergency. Usually require good initial guesses.

Ray marching in implicit surface

- Sphere Tracing

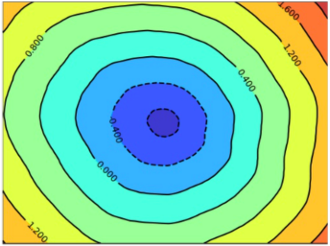


Figure: illustration of sdf.

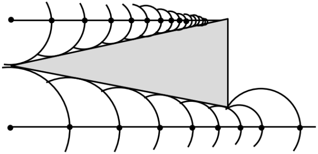


Figure: Sphere tracing.

Sphere tracing:
$t = t + f(x_{n-1})$
$x_n = x_0 + t \cdot \vec{v}$
until $\|f(x_n)\| \leq \epsilon$

Overviews

Introduction to implicit geometry (definition, discretization, ray marching, ...)

Properties of implicit function (normals, sdf, ...)

Preliminary (mathematics, neural networks, ...)

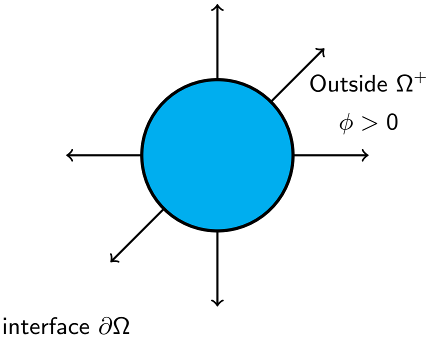
Differentiale Rendering (volume, surface, regularization, ...)

Properties

- Inside / outside function (occpuancy function)
- Signed distance function
- Boolean operations
- The normal of implicit function
- Mean curvature of the implicit function

Properties - Signed

- Inside / Outside function
  Easier to determine whether a point is inside/outside the surface.



$$\phi(\vec{(x)}) = x^2 + y^2 + z^2 - 1 = 0$$

Properties - SDF

Signed Distance Functions

- Singed distance functions are subset of implicit function defined to be positive on the exterior, negative on the interior with $\|\nabla\phi(\vec{x})\| = 1$.
- A distance function is defined as: $d(\vec{x}) = min(\|\vec{x} - \vec{x}_s\|)$ where $\vec{x}_s \in \partial\Omega$
- Given a point $\vec{x}$, the closest point x_s on the surface is $\vec{x}_s = \vec{x} - \phi(x)\vec{N}(x)$ (not restrictly)

Remark

$\phi(\vec{x})$ is not differentiable everywhere when x has multiple closest points, so we usually discuss the properties of x in a small region M.

Properties - Boolean

Boolean Operations of Implicit Functions

- $\phi(x) = min(\phi_1(x), \phi_2(x)$ is the union of the interior regions of $\phi_1(x)$ and $\phi_2(x)$.
- $\phi(x) = max(\phi_1(x), \phi_2(x))$ is the intersection of the interior regions of $\phi_1(x)$ and $\phi_2(x)$.
- $\phi(x) = -\phi_1(x)$ is the complement of $\phi_1(x)$.
- $\phi(x) = max(\phi_1(x), -\phi_2(x))$ represents the subtracton of the interior regions of $\phi_1(x)$ by the interior regions of $\phi_2(x)$.

Properties - Normal

Proposition. the normal $\vec{n}(x) = \frac{\nabla\phi(x)}{\|\nabla\phi(x)\|}$.
Proof sketch:
For any curve $\vec{x}(t)$ in the surface $\phi(\vec{x}) = 0$. Assume that partial derivatives continuous and exist, we apply total differentiation.

$$(\frac{\partial\phi(\vec{x})}{\partial\vec{x}})^T \frac{\partial\vec{x}(t)}{\partial t} \equiv 0 \tag{1}$$

Notice that $\partial\vec{x}(t)/\partial t$ is the tagent vector of the curve in x. All tagent vectors of curves pass x form a tagent plane, and $\frac{\partial f(\vec{x})}{\partial\vec{x}}$ is orthogonal to the tagent plane.

Properties - Mean Curvature

The mean curvature of the interface is
$\Delta\phi(x) = \nabla \cdot \nabla\phi(x) = \phi_{xx} + \phi_{yy} + \phi_{zz}$.
This is a Laplacian operator.

Remark

We usually describe scalar fields with gradient operator ∇ and describe vector fields with divergence operator $(\nabla\cdot)$.

Overviews

- Introduction to implicit geometry (definition, discretization, ray marching, ...)
- Properties of implicit function (normals, sdf, ...)
- Preliminary (mathematics, neural networks, ...)
- Differentiale Rendering (volume, surface, regularization, ...)

Preliminary

Lipschitz Continuous

Definition 1. f is *Lipschitz continuous* on Q with constant L, if for all $x, y \in Q$, we have

$$\|f(x) - f(y)\| \leq L\|x - y\| \tag{2}$$

Proposition 2, $L = sup(f'(x))$. For all $x, y \in Q$, there exist $\eta \in [x, y]$, we have

$$\|f(x) - f(y)\| = \|f'(\eta)(x - y)\| = \|f'(\eta)\|\|x - y\| \leq L\|x - y\| \tag{3}$$

Preliminary

MLP is Lipschitz Continuous

We only need to proof one-layer MLP is Lipschitz continuous.

$$\|f(Wx_1 + b) - f(Wx_2 + b)\| \leq sup(\|f'(x)\|)\|Wx_1 - Wx_2\| \tag{4}$$

$$\|Wx_1 - Wx_2\| = \|W(x_1 - x_2)\| \leq \|W\|\|x_1 - x_2\| \tag{5}$$

$$\|W\| = max_{x \neq 0} \frac{\|Wx\|}{\|x\|} \tag{6}$$

spectural norm equals to the largest eigen value

Since the gradients of activation functions such as ReLU, Sigmoid, Sine, Cosine are bounded, so MLP is Lipschitz continuous.

Overviews

- Introduction to implicit geometry (definition, discretization, ray marching, ...)
- Properties of implicit function (normals, sdf, ...)
- Preliminary (mathematics, neural networks, ...)
- Differentiale Rendering (volume, surface, regularization, ...)

Differentiable Rendering

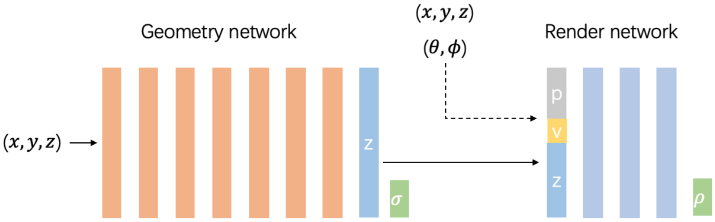


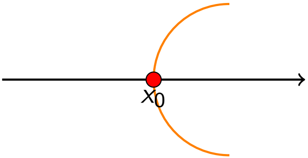
Figure: A common structure for implicit representation.

The problem is how it is possible to learn the shape accurately and fast?

Surface Rendering

- Differentiation only on the surface samples, *i.e.* $\sigma(x;\theta) = 0$

1. Use root finding to find the first intersection x_0.
2. Render the color $\rho(x_0;\gamma)$ and apply gradient descent.
3. The color (render) network should also drive geometry network $\sigma(x;\theta)$ to back propagation.



Surface Rendering

- Differentiation only on the surface samples, *i.e.* $\sigma(x;\theta) = 0$
- We must let x_0 differentiable to θ, *i.e* $x_0(\theta) = c_0 + \vec{v}_0 t(\theta)$.

$$\left(\frac{\partial \sigma(x_0;\theta)}{\partial x_0}\right)^T \vec{v_0} \cdot \frac{\partial t(\theta)}{\partial \theta} + \frac{\partial \sigma(x_0;\theta)}{\partial \theta} = 0 \tag{7}$$

$$\frac{\partial t(\theta)}{\partial \theta} = -\frac{1}{\langle \frac{\partial \sigma(x_0;\theta)}{\partial x_0}, \vec{v}_0 \rangle} \frac{\partial \sigma(x_0;\theta)}{\partial \theta} \tag{8}$$

- Forward implementation (Trick).
$x_0(\theta) = c_0 + \vec{v}_0 t(\theta_0) - \frac{1}{\langle \frac{\partial \sigma(x_0;\theta_0)}{\partial x_0}, \vec{v}_0 \rangle} \sigma(x_0;\theta)$

Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. CVPR 2020.

Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. NeurIPS 2020

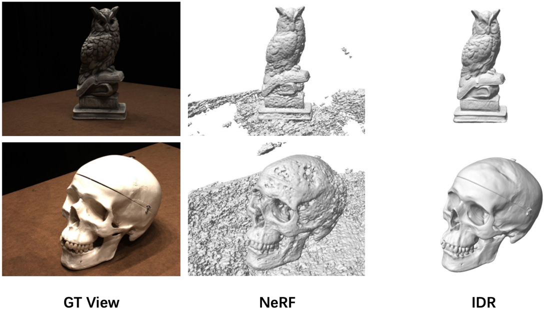
Surface Rendering

Pros:

- accurately.

Cons:

- slow to convergency.
- require a good initialization or regularization (object mask).



UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction. arXiv 2021

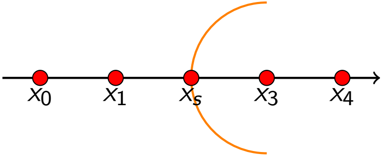
Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. NeurIPS 2020

Volume Rendering

Differentiation with many samples along the light. Alpha blending the samples to one pixel.

$$C(r) = \sum_{i=1}^{N} T_i(1 - exp(-\sigma_\theta(x_i)\delta_i))\rho(x_i, d) \tag{9}$$

$$T_i = exp(-\sum_{j<i} \sigma_\theta(x_j)\delta_j) \tag{10}$$



The question is when $C(r) \to \rho(x_s)$?

Volume Rendering

let $\sigma(x_i) = (1 - exp(-\sigma_\theta(x_i)\delta_i))$ (density $\rightarrow$ occpuany)
$T_i = \prod_{i<j}(1 - \sigma(x_j))$

Remark

Assume the Lipschitz constant of $\rho(x)$ is γ, $\sigma(x)$ is ϵ. Δ is max length from x_i to x_s.

We analyse :

$$
\begin{aligned}
\|C(r) - \rho(x_s)\| &= \|\sum_{i=1}^{N}(\prod_{j<i}(1 - \sigma(x_j))\sigma(x_i) - \frac{1}{N})\|\|(\rho(x_i) - \rho(x_s))\| \\
&\leq \|\gamma\Delta\|\|\sum_{i=1}^{N}(\prod_{j<i}(1 - \sigma(x_j))\sigma(x_i) - \frac{1}{N})\| \\
&= \gamma\Delta\|\sum_{i=1}^{N}(\prod_{j<i}(1 - \sigma(x_j))\sigma(x_i)) - 1\|
\end{aligned}
$$

Volume Rendering

$$\|\sigma(x_i) - \sigma(x_s)\| = \|\sigma(x_i) - 0.5\| \leq \epsilon\|x_i - x_s\| \leq \epsilon\Delta \tag{11}$$

$$\begin{aligned}
\sum_{i=1}^{N}(\prod_{j<i}(1-\sigma(x_j))\sigma(x_i)) &\leq \sum_{i=1}^{N}(\prod_{j\leq i}(0.5+\epsilon\Delta)(0.5+\epsilon\Delta) \\
&= \sum_{i=1}^{N}(0.5+\epsilon\Delta)^i \text{sum of geometric sequence} \\
&= \frac{(0.5+\epsilon\Delta)-(0.5+\epsilon\Delta)^N}{0.5-\epsilon\Delta}
\end{aligned}$$

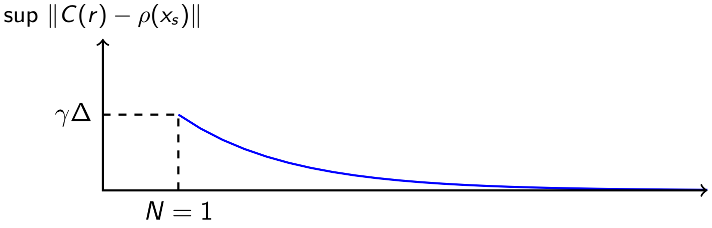
$$\|C(r) - \rho(x_s)\| \leq \gamma\Delta\|\frac{2\epsilon\Delta-(0.5+\epsilon\Delta)^N}{0.5-\epsilon\Delta}\| \tag{12}$$

$$\lim_{\Delta\to 0, N\to\infty}\|C(r) - \rho(x_s)\| \to 0 \tag{13}$$

Volume Rendering

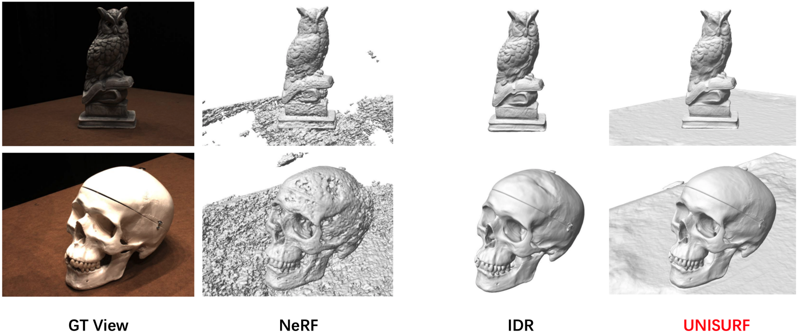
$$\|C(r) - \rho(x_s)\| \leq \gamma\Delta \| \frac{2\epsilon\Delta - (0.5 + \epsilon\Delta)^N}{0.5 - \epsilon\Delta} \| \tag{14}$$

$$\lim_{\Delta \to 0, N \to \infty} \|C(r) - \rho(x_s)\| \to 0 \tag{15}$$



This says that when sampling densely around the surface, volume rendering $\rightarrow$ surface rendering.

Volume Rendering



The original NeRF is not good at geometry; however, by sampling densely around the surface, NeRF convergences faster and does not require mask and good initialization compared to surface rendering techniques.

Other rendering techniques

- Differentiable Sphere Tracing

$$x_n(\theta) = x_{n-1} + \vec{d}\phi(x_{n-1};\theta) \quad (16)$$

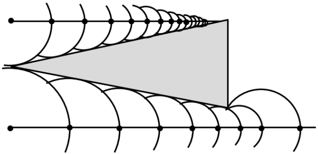


Figure: Sphere tracing.

- Differentiable Secant?
- Differentiable Root-finding?

An open question

Take-Home Message

Implicit Geometry

- Implicit function accurately represents a shape.
- Continous almost everywhere, and properties such as normal and mean curvature are well defined.
- Volume rendering is quick but not accurate (NeRF), while surface rendering is complicated.

However,

- Post-processing step is required when rendering novel views or extracting geometry.
- The achitecture designs are not fully investigated.