Skoltech
Skolkovo Institute of Science and Technology

Learning with implicit functions

Geometric Computer Vision

GCV v2021.1, Module 6

Alexey Artemov, Spring 2021

Skoltech
Skolkovo Institute of Science and Technology

Lecture Outline

Skoltech
Skolkovo Institute of Science and Technology

§1. Implicit functions: recap

Skoltech
Skolkovo Institute of Science and Technology

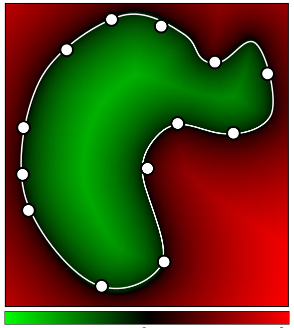
Implicit vs. other 3D representations

1.1. Implicit function representation

Skolkovo Institute of Science and Technology

§1. Implicit functions: recap

- Implicit representation: signed distance function (SDF); extract 0-level set



- Assumes the existence of a function

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

with value > 0 outside the shape and < 0 inside

- Extract zero-level set

$$\{\mathbf{x} : f(\mathbf{x}) = 0\}$$

1.1. Implicit function representation

Skolkovo Institute of Science and Technology

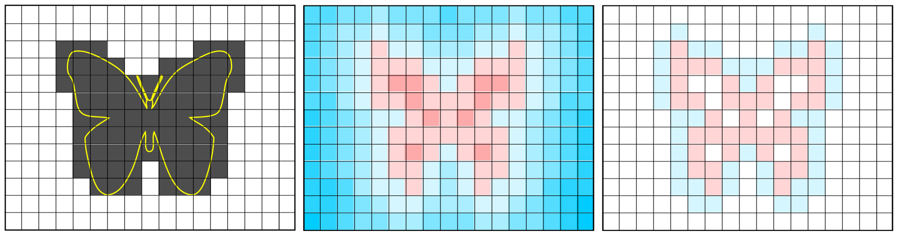
§1. Implicit functions: recap

- **Occupancy map**: a binary function $\text{OCC}(\mathbf{x}) : \Omega \rightarrow \{0,1\}$
- **Signed distance function** (SDF): an array of signed distance values $\text{SDF}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$
  - Positive values outsize the shape (free space), negative inside (occupied space)
- **Truncated SDF** (TSDF): $\text{TSDF}(\mathbf{x}) : \Omega \rightarrow [-\sigma, \sigma]$
  - Set max value to a fixed value $\text{TSDF}(\mathbf{x}) = \min(\max(\text{SDF}(\mathbf{x}), -\sigma), \sigma)$

1.1. Implicit function: defined on a grid

Skolkovo Institute of Science and Technology

§1. Implicit functions: recap



Occupancy map SDF TSDF

1.1. Implicit function: defined on a point set

Skolkovo Institute of Science and Technology

§1. Implicit functions: recap

- Recall point set representations from week 4
- We can define SDF/TSDF values in XYZ points only
  - Non-uniform support compared to volumetric grids

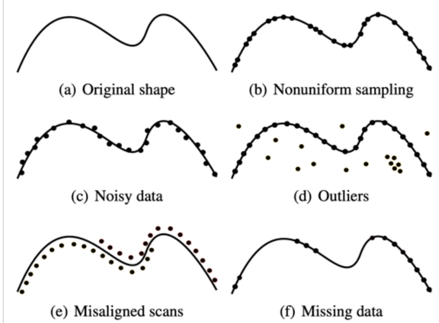


Figure 2: *Different forms of point cloud artifacts, shown here in the case of a curve in 2D.*

Figure credit: Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A. and Silva, C.T., 2017, January. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum* (Vol. 36, No. 1, pp. 301-329).

Skoltech
Skolkovo Institute of Science and Technology

1.1. Implicit function representation

§1. Implicit functions: recap

- **Surface reconstruction:** create a surface representation from sparse input points
  - This week: create an **implicit surface representation** (fit an implicit function): recover a signed distance function (SDF) with values < 0 inside the shape and > 0 outside
  - Some reconstruction algorithms include:
    - Radial Basis Functions
    - Moving Least Squares
    - Poisson Surface Reconstruction

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

Skoltech
Skolkovo Institute of Science and Technology

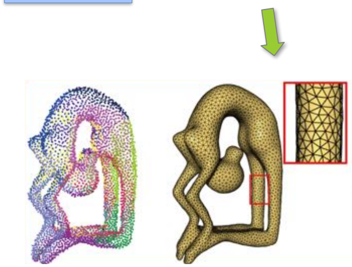
Reconstruction: problem setting

Skoltech
Skolkovo Institute of Science and Technology

Geometry Acquisition Pipeline

§2. Reconstruction: estimating implicit functions

Scanning: results in range images ⇨ **Registration:** bring all range images to one coordinate system ⇨ **Stitching/ reconstruction:** Integration of scans into a single mesh ⇨ **Postprocess:**
- Topological and geometric filtering
- Remeshing
- Compression



Slide Credit: Daniele Panozzo

Digital Michelangelo Project

§2. Reconstruction: estimating implicit functions

1G sample points → 8M triangles

4G sample points → 8M triangles

Input to Reconstruction Process

§2. Reconstruction: estimating implicit functions

- Input option 1: just a set of 3D points, irregularly spaced
  - Need to estimate normals ➔ next class

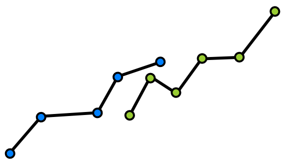
- Input option 2:
  normals come from the range scans

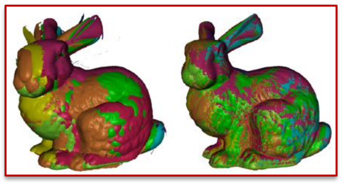
set of raw scans → reconstructed model

How to Connect the Dots?

§2. Reconstruction: estimating implicit functions

- **Explicit reconstruction:**
  stitch the range scans together



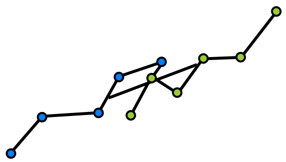


“Zippered Polygon Meshes from Range Images”, Greg Turk and Marc Levoy, ACM SIGGRAPH 1994

How to Connect the Dots?

§2. Reconstruction: estimating implicit functions

- **Explicit reconstruction:**
  stitch the range scans together



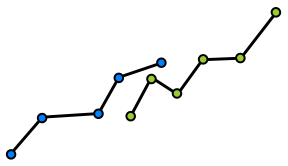
- Connect sample points by triangles
- Exact interpolation of sample points
- Bad for noisy or misaligned data
- Can lead to holes or non-manifold situations

How to Connect the Dots?

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- **Implicit reconstruction:** estimate a signed distance function (SDF); extract 0-level set mesh using Marching Cubes

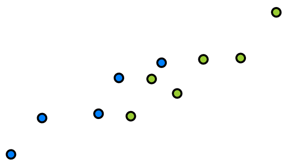


How to Connect the Dots?

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- **Implicit reconstruction:** estimate a signed distance function (SDF); extract 0-level set mesh using Marching Cubes

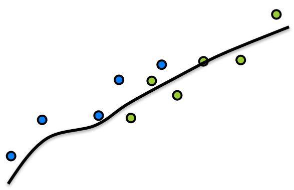


How to Connect the Dots?

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- **Implicit reconstruction:** estimate a signed distance function (SDF); extract 0-level set mesh using Marching Cubes



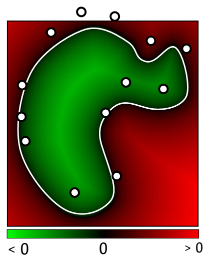
- Approximation of input points
- Watertight manifold results by construction

How to Connect the Dots?

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

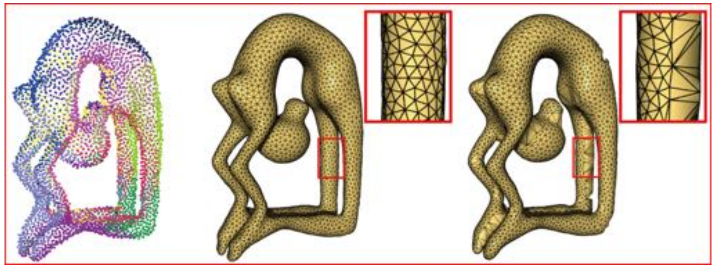
- **Implicit reconstruction:** estimate a signed distance function (SDF); extract 0-level set mesh using Marching Cubes



- Approximation of input points
- Watertight manifold results by construction

Implicit vs. Explicit

§2. Reconstruction: estimating implicit functions

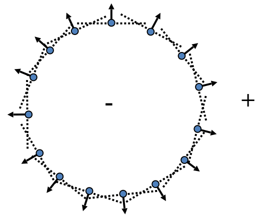


Input Implicit Explicit

SDF from Points and Normals

§2. Reconstruction: estimating implicit functions

- Compute signed distance to the tangent plane of the closest point
- Normals help to distinguish between inside and outside

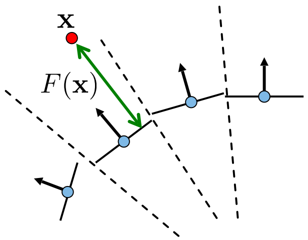


"Surface reconstruction from unorganized points", Hoppe et al., ACM SIGGRAPH 1992
http://research.microsoft.com/en-us/um/people/hoppe/proj/recon/

SDF from Points and Normals

§2. Reconstruction: estimating implicit functions

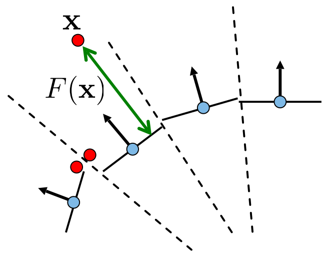
- Compute signed distance to the tangent plane of the closest point
- Problem??



SDF from Points and Normals

§2. Reconstruction: estimating implicit functions

- Compute signed distance to the tangent plane* of the closest point
- The function will be discontinuous

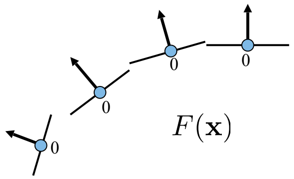


* The Hoppe92 paper computes the tangent planes slightly differently (by PCA on k-nearest-neighbors of each data point, see next class), but the consequences are still the same.

Smooth SDF

§2. Reconstruction: estimating implicit functions

- Instead find a smooth formulation for F.
- Scattered data interpolation:
  - $F(\mathbf{p}_i) = 0$
  - F is smooth
  - Avoid trivial $F \equiv 0$

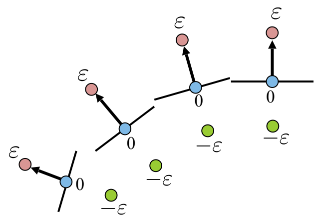


"Reconstruction and representation of 3D objects with radial basis functions", Carr et al., ACM SIGGRAPH 2001

Smooth SDF

§2. Reconstruction: estimating implicit functions

- Scattered data interpolation:
  - $F(\mathbf{p}_i) = 0$
  - F is smooth
  - Avoid trivial $F \equiv 0$
- Add off-surface constraints



$F(\mathbf{p}_i + \varepsilon \mathbf{n}_i) = \varepsilon$

$F(\mathbf{p}_i - \varepsilon \mathbf{n}_i) = -\varepsilon$

"Reconstruction and representation of 3D objects with radial basis functions", Carr et al., ACM SIGGRAPH 2001

Radial Basis Function Interpolation

§2. Reconstruction: estimating implicit functions

- **RBF**: Weighted sum of shifted, smooth kernels

$$F(\mathbf{x}) = \sum_{i=0}^{N-1} w_i \, \varphi(\|\mathbf{x} - \mathbf{c}_i\|)$$

Scalar weights
Unknowns

Smooth kernels (basis functions) centered at constrained points.
For example:
$\varphi(r) = r^3$

$N = 3n$

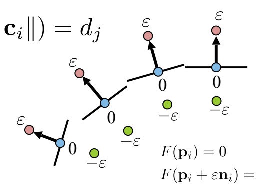
Radial Basis Function Interpolation

§2. Reconstruction: estimating implicit functions

- Interpolate the constraints:

$$\{\mathbf{c}_{3i}, \mathbf{c}_{3i+1}, \mathbf{c}_{3i+2}\} = \{\mathbf{p}_i,\ \mathbf{p}_i + \varepsilon \mathbf{n}_i,\ \mathbf{p}_i - \varepsilon \mathbf{n}_i\}$$

$$\forall j = 0, \ldots, N-1, \quad \sum_{i=0}^{N-1} w_i \, \varphi(\|\mathbf{c}_j - \mathbf{c}_i\|) = d_j$$



$F(\mathbf{p}_i) = 0$

$F(\mathbf{p}_i + \varepsilon \mathbf{n}_i) = \varepsilon$

$F(\mathbf{p}_i - \varepsilon \mathbf{n}_i) = -\varepsilon$

Radial Basis Function Interpolation

§2. Reconstruction: estimating implicit functions

- Interpolate the constraints:

$$\{\mathbf{c}_{3i}, \mathbf{c}_{3i+1}, \mathbf{c}_{3i+2}\} = \{\mathbf{p}_i,\ \mathbf{p}_i + \varepsilon \mathbf{n}_i,\ \mathbf{p}_i - \varepsilon \mathbf{n}_i\}$$

- Symmetric linear system to get the weights:

$$\begin{pmatrix} \varphi(\|\mathbf{c}_0 - \mathbf{c}_0\|) & \dots & \varphi(\|\mathbf{c}_0 - \mathbf{c}_{N-1}\| \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{c}_{N-1} - \mathbf{c}_0\|) & \dots & \varphi(\|\mathbf{c}_{N-1} - \mathbf{c}_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_{N-1} \end{pmatrix}$$

RBF Kernels

§2. Reconstruction: estimating implicit functions

$$\varphi(r) = r^3$$

- Triharmonic:
  - Globally supported
  - Leads to dense symmetric linear system
  - **C^2 smoothness**
  - Works well for highly irregular sampling

RBF Kernels

§2. Reconstruction: estimating implicit functions

- Polyharmonic spline

$$\varphi(r) = r^k \log(r),\ \ k = 2, 4, 6 \dots$$
$$\varphi(r) = r^k,\ \ k = 1, 3, 5 \dots$$

- Multiquadratic

$$\varphi(r) = \sqrt{r^2 + \beta^2}$$

- Gaussian

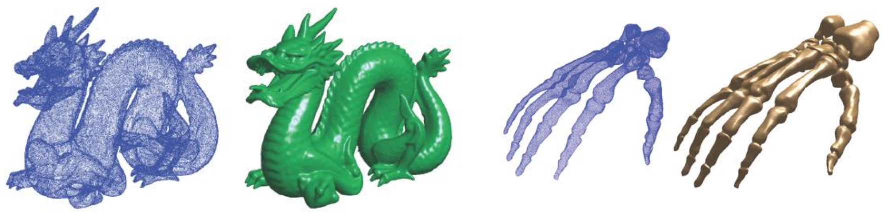
$$\varphi(r) = e^{-\beta r^2}$$

- B-Spline (compact support)

$$\varphi(r) = \text{piecewise-polynomial}(r)$$

RBF Reconstruction Examples

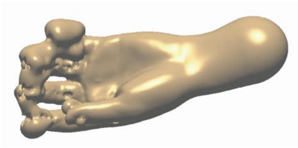
§2. Reconstruction: estimating implicit functions



"Reconstruction and representation of 3D objects with radial basis functions", Carr et al., ACM SIGGRAPH 2001

Off-Surface Points

§2. Reconstruction: estimating implicit functions



Insufficient number/
badly placed off-surface points

Properly chosen off-surface points

"Reconstruction and representation of 3D objects with radial basis functions", Carr et al., ACM SIGGRAPH 2001

Comparison of the various SDFs so far

§2. Reconstruction: estimating implicit functions

Distance
to plane

Compact RBF

Global RBF
Triharmonic

RBF – Discussion

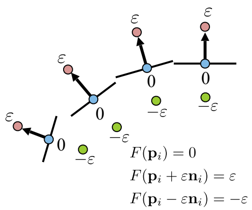
§2. Reconstruction: estimating implicit functions

- Global definition!

$$F(\mathbf{x}) = \sum_{i=0}^{N-1} w_i\,\varphi(\|\mathbf{x} - \mathbf{c}_i\|)$$

$$\{\mathbf{c}_{3i}, \mathbf{c}_{3i+1}, \mathbf{c}_{3i+2}\} = \{\mathbf{p}_i,\ \mathbf{p}_i + \varepsilon\mathbf{n}_i,\ \mathbf{p}_i - \varepsilon\mathbf{n}_i\}$$

$$\begin{pmatrix} \varphi(\|\mathbf{c}_0 - \mathbf{c}_0\|) & \dots & \varphi(\|\mathbf{c}_0 - \mathbf{c}_{N-1}\| \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{c}_{N-1} - \mathbf{c}_0\|) & \dots & \varphi(\|\mathbf{c}_{N-1} - \mathbf{c}_{N-1}\|) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{N-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_{N-1} \end{pmatrix}$$



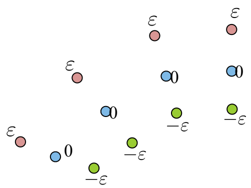
- Global optimization of the weights, even if the basis functions are local

Moving Least Squares

Moving Least Squares (MLS)

§2. Reconstruction: estimating implicit functions

- Do purely **local** approximation of the SDF
- Weights change depending on where we are evaluating
- The beauty: the "stitching" of all local approximations, seen as one function $F(\mathbf{x})$, is smooth everywhere!
  - We get a **globally** smooth function but only do **local** computation

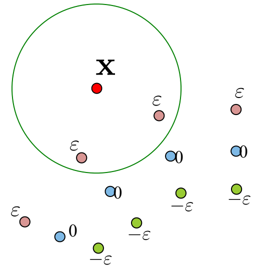


"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004
http://graphics.berkeley.edu/papers/Shen-IAI-2004-08/index.html

Moving Least Squares (MLS)

§2. Reconstruction: estimating implicit functions

- Do purely **local** approximation of the SDF
- Weights change depending on where we are evaluating
- The beauty: the "stitching" of all local approximations, seen as one function $F(\mathbf{x})$, is smooth everywhere!
  - We get a **globally** smooth function but only do **local** computation

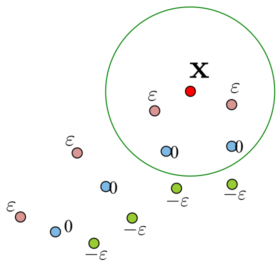


"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004
http://graphics.berkeley.edu/papers/Shen-IAI-2004-08/index.html

Moving Least Squares (MLS)

§2. Reconstruction: estimating implicit functions

- Do purely **local** approximation of the SDF
- Weights change depending on where we are evaluating
- The beauty: the "stitching" of all local approximations, seen as one function $F(\mathbf{x})$, is smooth everywhere!
  - We get a **globally** smooth function but only do **local** computation

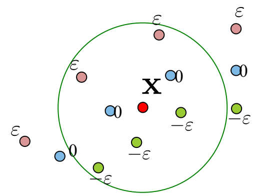


"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004
http://graphics.berkeley.edu/papers/Shen-IAI-2004-08/index.html

Moving Least Squares (MLS)

§2. Reconstruction: estimating implicit functions

- Do purely **local** approximation of the SDF
- Weights change depending on where we are evaluating
- The beauty: the "stitching" of all local approximations, seen as one function $F(\mathbf{x})$, is smooth everywhere!
  - We get a **globally** smooth function but only do **local** computation



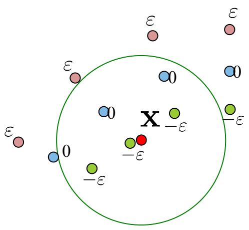
"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004
http://graphics.berkeley.edu/papers/Shen-IAI-2004-08/index.html

Moving Least Squares (MLS)

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- Do purely **local** approximation of the SDF
- Weights change depending on where we are evaluating
- The beauty: the "stitching" of all local approximations, seen as one function $F(\mathbf{x})$, is smooth everywhere!
  - We get a **globally** smooth function but only do **local** computation



"Interpolating and Approximating Implicit Surfaces from Polygon Soup", Shen et al., ACM SIGGRAPH 2004

http://graphics.berkeley.edu/papers/Shen-IAI-2004-08/index.html

Least-Squares Approximation

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- Polygonial least-squares approximation
  - Choose degree, k

$$f \in \Pi_k^3 : f(x,y,z) = a_0 + a_1x + a_2y + a_3z + a_4x^2 + a_5xy + \ldots + a_*z^k$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\,\mathbf{a}$$

$$\mathbf{a} = (a_1, a_2, \ldots, a_*)^T,\ \ \mathbf{b}(\mathbf{x})^T = (1, x, y, z, x^2, xy, \ldots, z^k)$$

  - Find $\mathbf{a}$ that minimizes sum of squared differences

$$\underset{f\in\Pi_k^3}{\operatorname{argmin}} \sum_{i=0}^{N-1} (f(\mathbf{c}_i) - d_i)^2 \quad \text{or:} \quad \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=0}^{N-1} (\mathbf{b}(\mathbf{c}_i)^T\mathbf{a} - d_i)^2$$

MOVING Least-Squares Approximation

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- Polynomial least-squares approximation
  - Choose degree, k

$$f \in \Pi_k^3 : f(x,y,z) = a_0 + a_1x + a_2y + a_3z + a_4x^2 + a_5xy + \ldots + a_*z^k$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\,\mathbf{a}$$

$$\mathbf{a} = (a_1, a_2, \ldots, a_*)^T,\ \ \mathbf{b}(\mathbf{x})^T = (1, x, y, z, x^2, xy, \ldots, z^k)$$

  - Find $\mathbf{a_x}$ that minimizes WEIGHTED sum of squared differences

$$f_{\mathbf{x}} = \underset{f\in\Pi_k^3}{\operatorname{argmin}} \sum_{i=0}^{N-1} \theta(\|\mathbf{x} - \mathbf{c}_i\|)\,(f(\mathbf{c}_i) - d_i)^2 \quad \text{or:}$$

$$\mathbf{a_x} = \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=0}^{N-1} \theta(\|\mathbf{x} - \mathbf{c}_i\|)\,(\mathbf{b}(\mathbf{c}_i)^T\mathbf{a} - d_i)^2$$

MOVING Least-Squares Approximation

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

- Polynomial least-squares approximation
  - Choose degree, k

$$f \in \Pi_k^3 : f(x,y,z) = a_0 + a_1x + a_2y + a_3z + a_4x^2 + a_5xy + \ldots + a_*z^k$$

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\,\mathbf{a}$$

$$\mathbf{a} = (a_1, a_2, \ldots, a_*)^T,\ \ \mathbf{b}(\mathbf{x})^T = (1, x, y, z, x^2, xy, \ldots, z^k)$$

  - Find $\mathbf{a_x}$ that minimizes WEIGHTED sum of squared differences
  - The value of the SDF is the obtained approximation evaluated at $\mathbf{x}$:

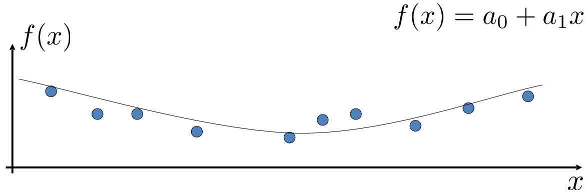
$$F(\mathbf{x}) = f_{\mathbf{x}}(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\mathbf{a_x}$$

MLS – 1D Example

§2. Reconstruction: estimating implicit functions

- Global approximation in $\prod_2^1$

$$f(x) = a_0 + a_1 x + a_2 x^2$$

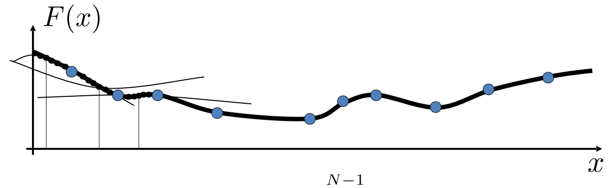


$$f = \underset{f \in \Pi_2^1}{\operatorname{argmin}} \sum_{i=0}^{N-1} \left(f(c_i) - d_i\right)^2$$

MLS – 1D Example

§2. Reconstruction: estimating implicit functions

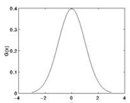
- MLS approximation using functions in $\prod_2^1$



$$F(x) = f_x(x), \quad f_x = \underset{f \in \Pi_2^1}{\operatorname{argmin}} \sum_{i=0}^{N-1} \theta(\|c_i - x\|) \left(f(c_i) - d_i\right)^2$$

Weight Functions

§2. Reconstruction: estimating implicit functions



- Gaussian
  - h is a smoothing parameter

$$\theta(r) = e^{-\frac{r^2}{h^2}}$$

$$\theta(r) = (1 - r/h)^4 (4r/h + 1)$$

- Wendland function
  - Defined in $[0, h]$ and

$$\theta(0) = 1, \ \theta(h) = 0, \ \theta'(h) = 0, \ \theta''(h) = 0$$

- Singular function

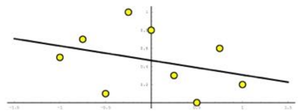
$$\theta(r) = \frac{1}{r^2 + \varepsilon^2}$$

  - For small ε, weights large near r=0 (interpolation)

Dependence on Weight Function

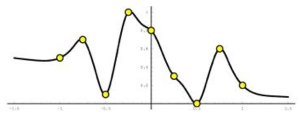
§2. Reconstruction: estimating implicit functions

- Global least squares with linear basis



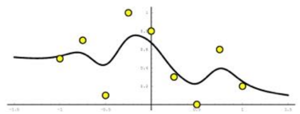
- MLS with (nearly) singular weight function

$$\theta(r) = \frac{1}{r^2 + \varepsilon^2}$$



- MLS with approximating weight function

$$\theta(r) = e^{-\frac{r^2}{h^2}}$$



Dependence on Weight Function

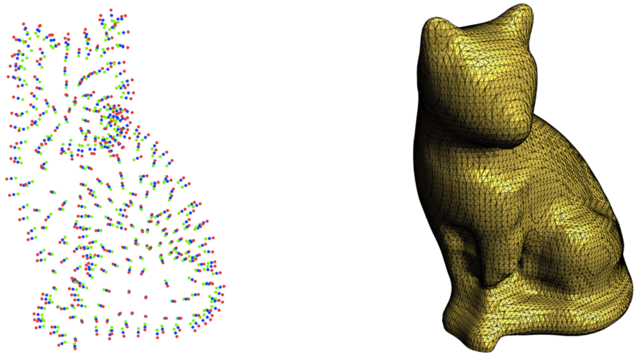
§2. Reconstruction: estimating implicit functions

- The MLS function F is continuously differentiable if and only if the weight function θ is continuously differentiable
- In general, F is as smooth as θ

$$F(\mathbf{x}) = f_{\mathbf{x}}(\mathbf{x}), \quad f_{\mathbf{x}} = \underset{f \in \Pi_k^d}{\operatorname{argmin}} \sum_{i=0}^{N-1} \theta(\|\mathbf{c}_i - \mathbf{x}\|) \left(f(\mathbf{c}_i) - d_i\right)^2$$

Example: Reconstruction

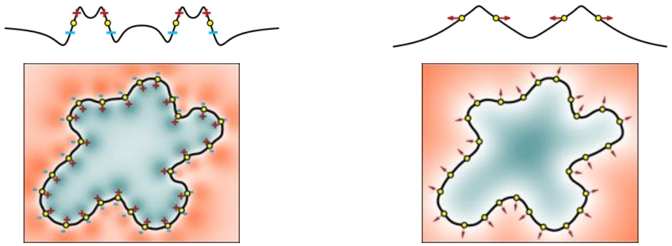
§2. Reconstruction: estimating implicit functions



MLS SDF – Possible Improvement

§2. Reconstruction: estimating implicit functions

- Point constraints vs. true normal constraints



-

Global RBF vs. Local MLS

§2. Reconstruction: estimating implicit functions

- RBF:
  - sees the whole data set, can make for very smooth surfaces
  - global (dense) system to solve – expensive
- MLS:
  - sees only a small part of the dataset, can get confused by noise
  - local linear solves – cheap

Poisson Surface Reconstruction

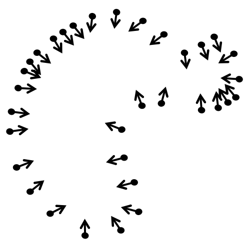
Poisson Surface Reconstruction

§2. Reconstruction: estimating implicit functions

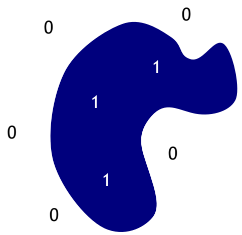
- Very popular modern method, code available: M. Kazhdan, M. Bolitho and H. Hoppe, Symposium on Geometry Processing 2006 http://www.cs.jhu.edu/~misha/Code/PoissonRecon/
- Global fitting of an *indicator function* using PDE
  - Robust to noise, sparse, computationally tractable

Poisson Surface Reconstruction

§2. Reconstruction: estimating implicit functions



Oriented points

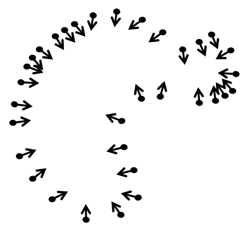


Indicator function

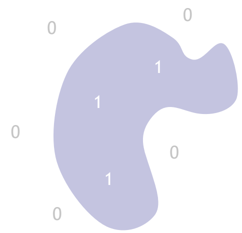
$\chi_{\mathcal{M}}$

Poisson Surface Reconstruction

§2. Reconstruction: estimating implicit functions



Oriented points



Indicator function

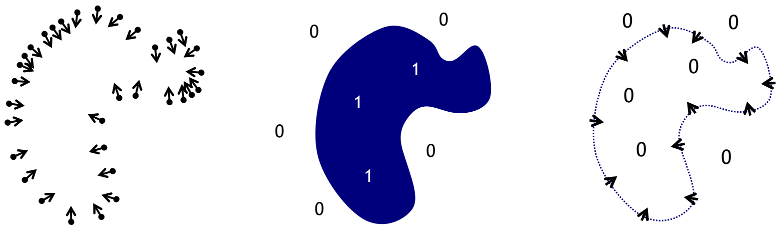
$\chi_{\mathcal{M}}$

We don't know the indicator function ☹

Poisson Surface Reconstruction

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions



Oriented points

Indicator function
$\chi_{\mathcal{M}}$

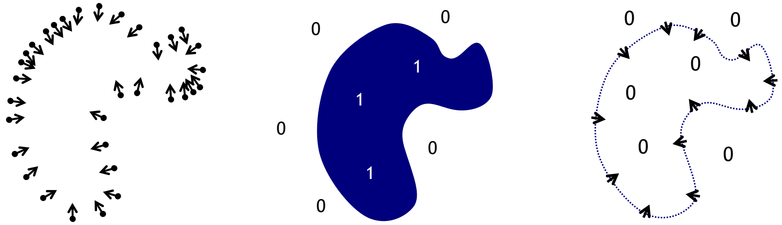
Indicator gradient
$\nabla\chi_{\mathcal{M}}$

But we can estimate its gradient! ☺

Poisson Surface Reconstruction

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions



Oriented points

Indicator function
$\chi_{\mathcal{M}}$

Indicator gradient
$\nabla\chi_{\mathcal{M}}$

Reconstruct χ by solving the Poisson equation

$$\Delta\chi_{\mathcal{M}} = \mathrm{div}\nabla\chi_{\mathcal{M}}$$

Michelangelo's David

Skoltech
Skolkovo Institute of Science and Technology

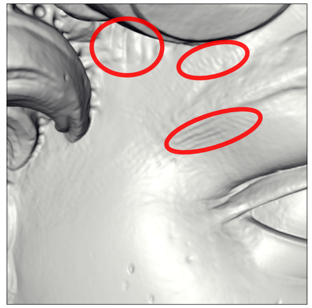
§2. Reconstruction: estimating implicit functions

- 215 million data points from 1000 scans
- 22 million triangle reconstruction
- Compute Time: 2.1 hours
  (this was in year 2006)
- Peak Memory: 6600MB

David – Chisel marks

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions



Skoltech
Skolkovo Institute of Science and Technology

David – Drill Marks

§2. Reconstruction: estimating implicit functions

Skoltech
Skolkovo Institute of Science and Technology

David – Eye

§2. Reconstruction: estimating implicit functions

Skoltech
Skolkovo Institute of Science and Technology

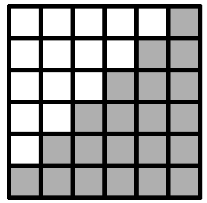
Neural approximators for implicit functions

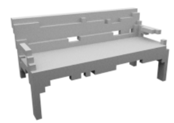
Skoltech
Skolkovo Institute of Science and Technology

2.3. 3D Representations

§2. Reconstruction: estimating implicit functions

- **Voxels:**
  - Discretization of 3D space into grid
  - Easy to process with neural networks
  - Cubic memory $O(N^3) \Rightarrow$ limited resolution
  - Manhattan world bias



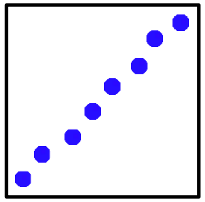


- Maturana, D., & Scherer, S. (2015, September). Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 922-928). IEEE.

2.3. 3D Representations

§2. Reconstruction: estimating implicit functions

- **Points:**
  - Discretization of surface into 3D points
  - Does not model connectivity / topology
  - Limited number of points
  - Global shape description

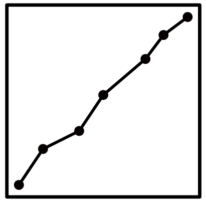


- Fan, H., Su, H., & Guibas, L. J. (2017). A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 605-613).

2.3. 3D Representations

§2. Reconstruction: estimating implicit functions

- **Meshes:**
  - Discretization into vertices and faces
  - Limited number of vertices / granularity
  - Requires class-specific template
    – or –
  - Leads to self-intersections

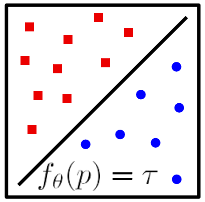


- Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). A papier-mâché approach to learning 3d surface generation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 216-224).

2.3. 3D Representations

§2. Reconstruction: estimating implicit functions

- Implicit representations:
  - Implicit representation ⇒ No discretization
  - Arbitrary topology & resolution
  - Low memory footprint
  - Not restricted to specific class



- Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 165-174).
- Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4460-4470).

2.3. DeepSDF

§2. Reconstruction: estimating implicit functions

- Represent the shape's surface using a continuous volumetric field (SDF)
  - Point magnitude: distance to the surface boundary
  - Point sign: inside (–) or outside (+) of the shape
- Implicitly encode shape's boundary as the zero-level-set of the learned function

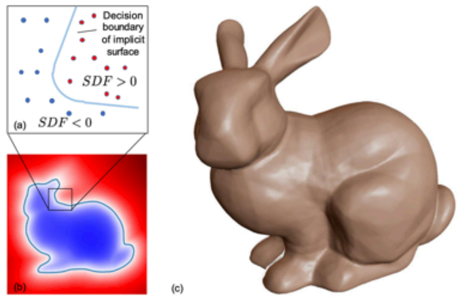


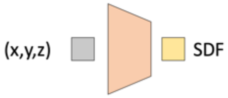
Figure credit: DeepSDF

2.3. DeepSDF: formulation

§2. Reconstruction: estimating implicit functions

- A conventional definition of SDF:

  $\text{SDF}(\,\cdot\,) : \mathbb{R}^3 \to \mathbb{R}, \qquad \text{SDF}(\mathbf{x}) = s$
- Represent the SDF using a neural network $f_\theta(\mathbf{x})$ (e.g. an MLP)
- Direct application for fitting SDF:
  - Use a training dataset $\mathbf{X}^\ell = \{(\mathbf{x}, s) : \text{SDF}(\mathbf{x}) = s\}$
  - Select a neural network approximator $f_\theta(\mathbf{x}) \approx \text{SDF}(\mathbf{x}), \forall \mathbf{x} \in \Omega$
  - Minimize a fitting loss: $\mathscr{L}(f_\theta(\mathbf{x}), s) = |\,\text{clamp}(f_\theta(\mathbf{x}), \delta) - \text{clamp}(s, \delta)\,| \to \min_\theta$
  - δ controls the distance from the surface over which we expect to maintain a metric SDF



(a) Single Shape DeepSDF

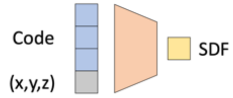
Figure credit: DeepSDF

2.3. DeepSDF: formulation

§2. Reconstruction: estimating implicit functions

- We want a model to represent many shapes!
- Introduce a latent vector $\mathbf{z}_i$ (encoding of desired shape i)
- Now approximate $f_\theta(\mathbf{z}_i, \mathbf{x}) \approx \text{SDF}_i(\mathbf{x})$
  - Model multiple SDFs with a single function $f_\theta(\mathbf{z}, \mathbf{x})$
- **During training** (probabilistic formulation) find shape codes $\{\mathbf{z}_i\}_{i=1}^N$ and network parameters θ by minimizing (assuming zero-mean multivariate-Gaussian prior with a spherical cov. $\sigma^2 I$)

$$\sum_{i=1}^{N} \left(\sum_{j=1}^{K} \mathscr{L}(f_\theta(\mathbf{z}_i, \mathbf{x}_j), s_j) + \frac{1}{\sigma^2} \|\mathbf{z}_i\|_2^2 \right) \to \min_{\theta, \{\mathbf{z}_i\}_{i=1}^N}$$



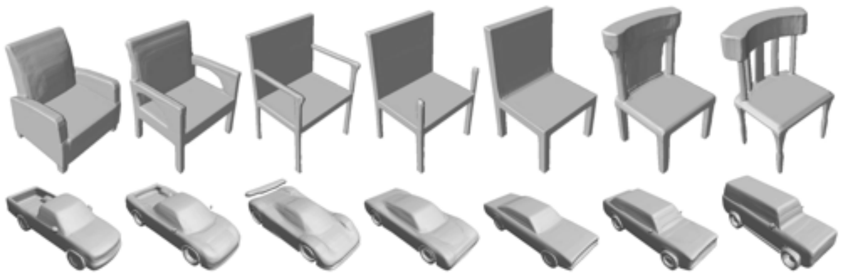
(b) Coded Shape DeepSDF

Figure credit: DeepSDF

2.3. DeepSDF: results

§2. Reconstruction: estimating implicit functions

- Renderings of DeepSDF interpolating between two shapes in the latent space of $\mathbf{z}_i$s

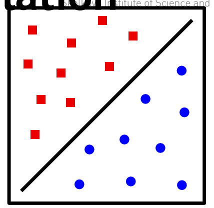


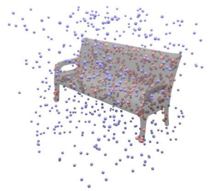
2.3. Occupancy networks: formulation

§2. Reconstruction: estimating implicit functions

- Key idea:
  - Do not represent 3D shape explicitly
  - Instead, consider surface implicitly
  - as decision boundary of a non-linear classifier:

$$\underset{\substack{\uparrow \\ \text{3D Location}}}{f_\theta : \mathbb{R}^3} \times \underset{\substack{\uparrow \\ \text{Condition (eg, Image)}}}{\mathcal{X}} \to \underset{\substack{\uparrow \\ \text{Occupancy Probability}}}{[0, 1]}$$





Mescheder, Oechsle, Niemeyer, Nowozin and Geiger: Occupancy Networks: Learning 3D Reconstruction in Function Space. CVPR, 2019.

2.3. Occupancy networks: results

Skoltech
Skolkovo Institute of Science and Technology

§2. Reconstruction: estimating implicit functions

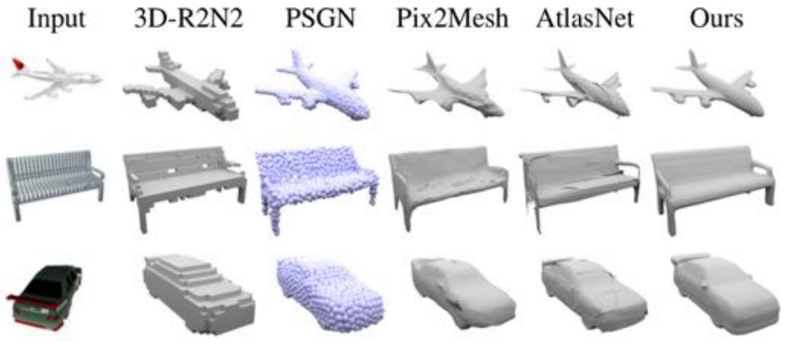


Figure credit: DeepSDF