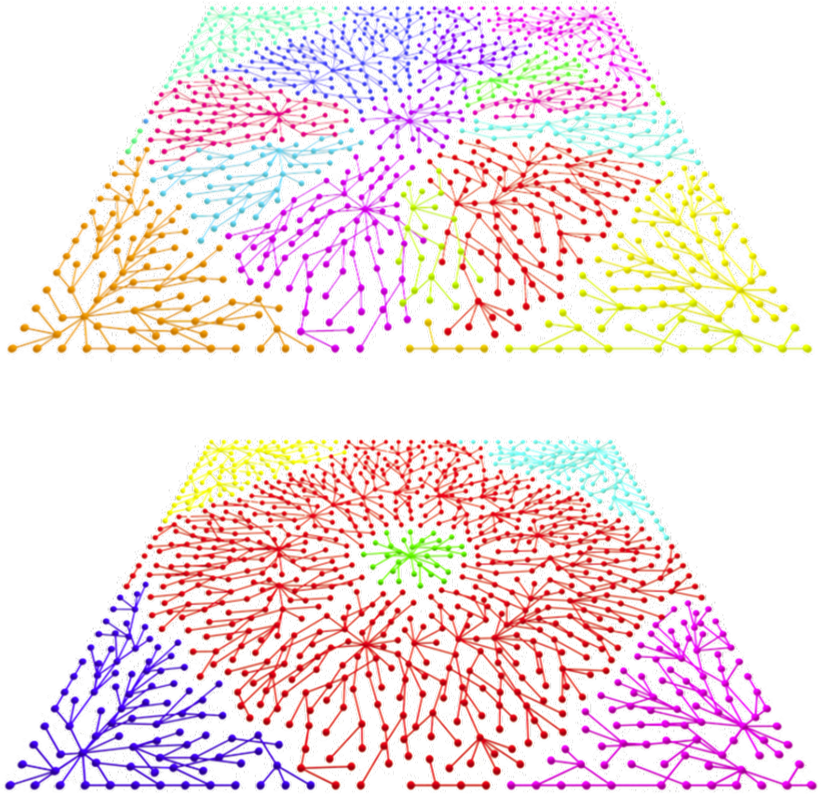
# CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University

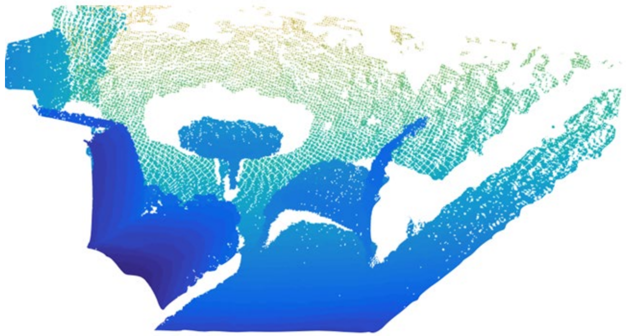


Lecture 15
18 May 2022

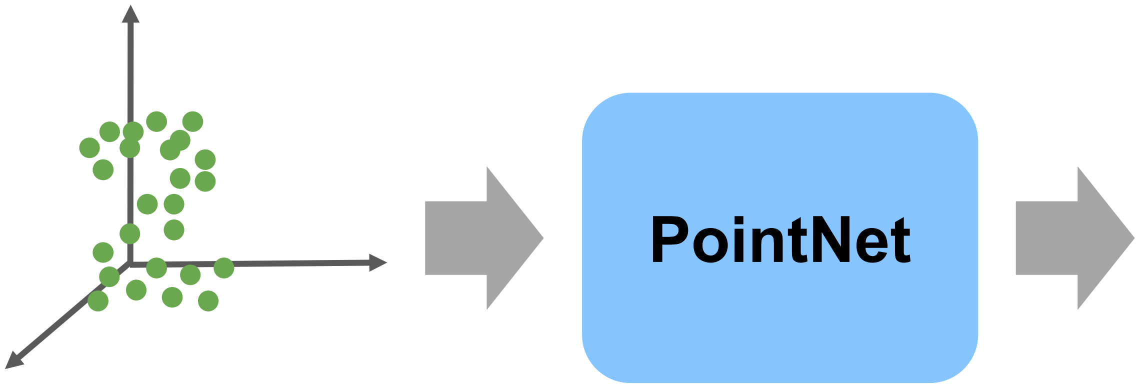
Leonidas Guibas Laboratory
Geometric Computing

# Last Time: Deep Learning on Point Cloud Data

Non-regular 3D data



# Deep Nets for PCs: PointNet and PointNet++




*Object Classification*

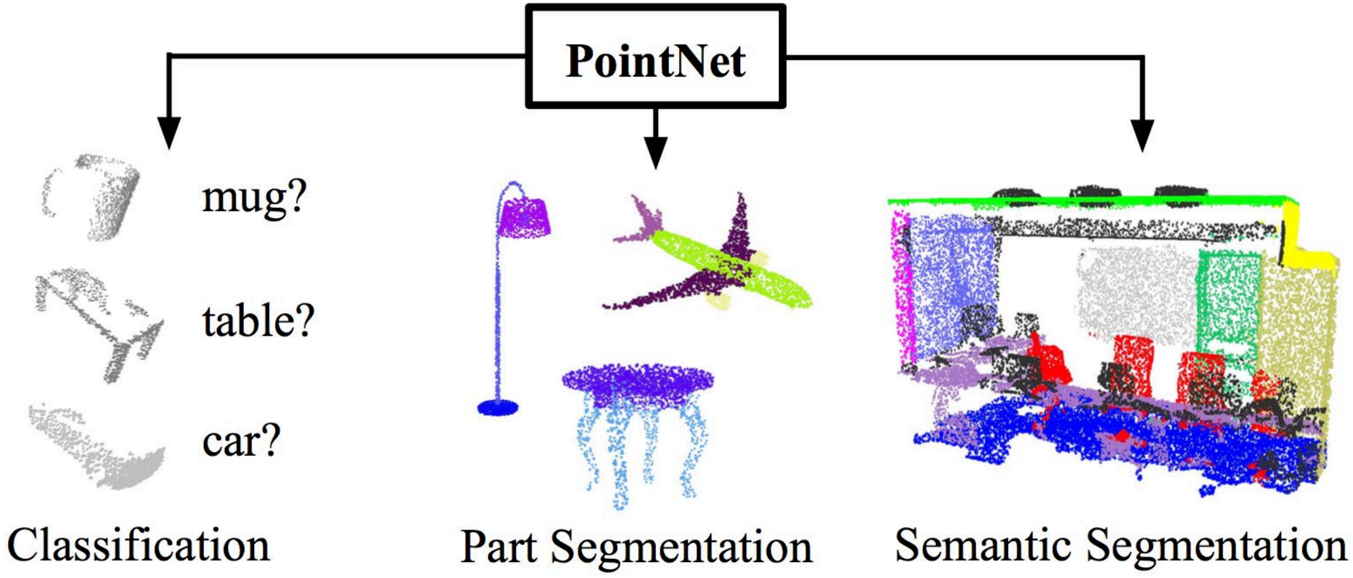
*Object Part Segmentation*

*Semantic Scene Parsing*

*...*

**End-to-end learning** for irregular point data

**Unified** framework for various tasks




Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. (CVPR'17)

# Invariances

*The model has to respect key desiderata for point clouds:*

**Point Permutation Invariance**

Point cloud is a set of unordered points

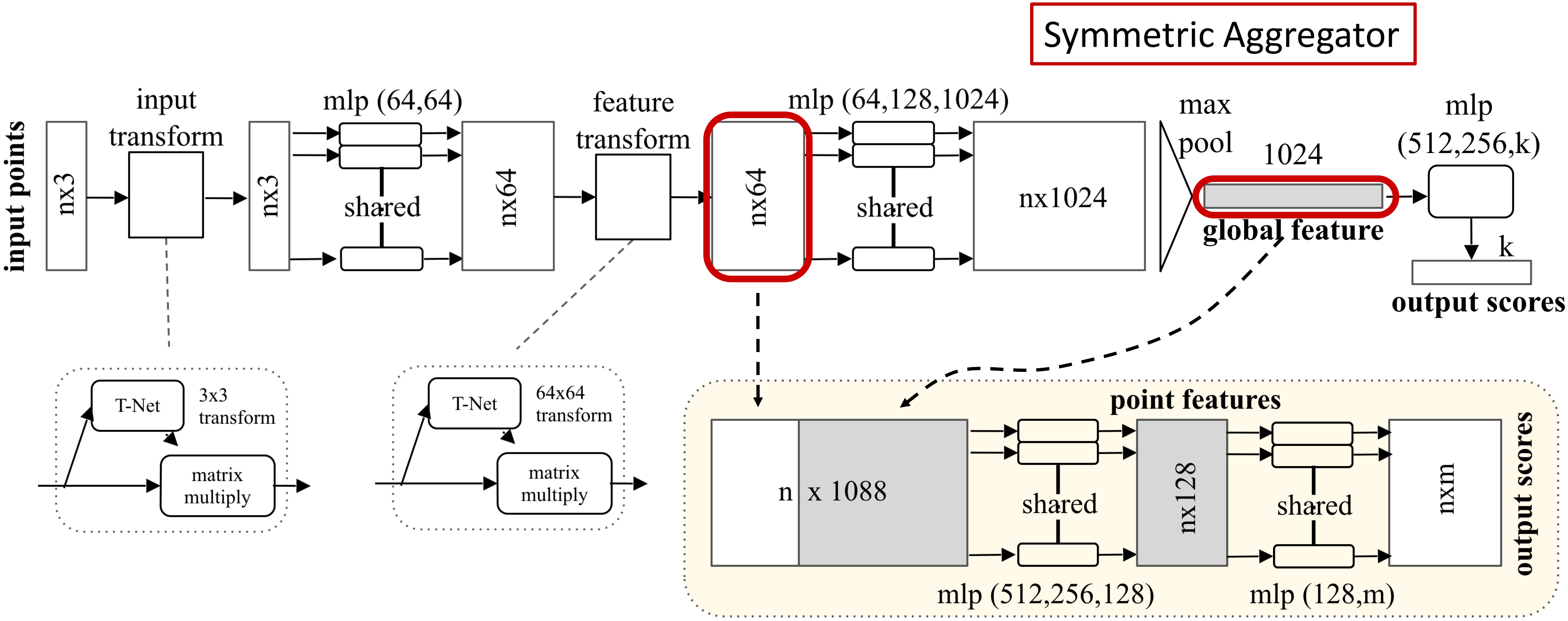
**Spatial Transformation Invariance**

Point cloud rigid motions should not alter classification results

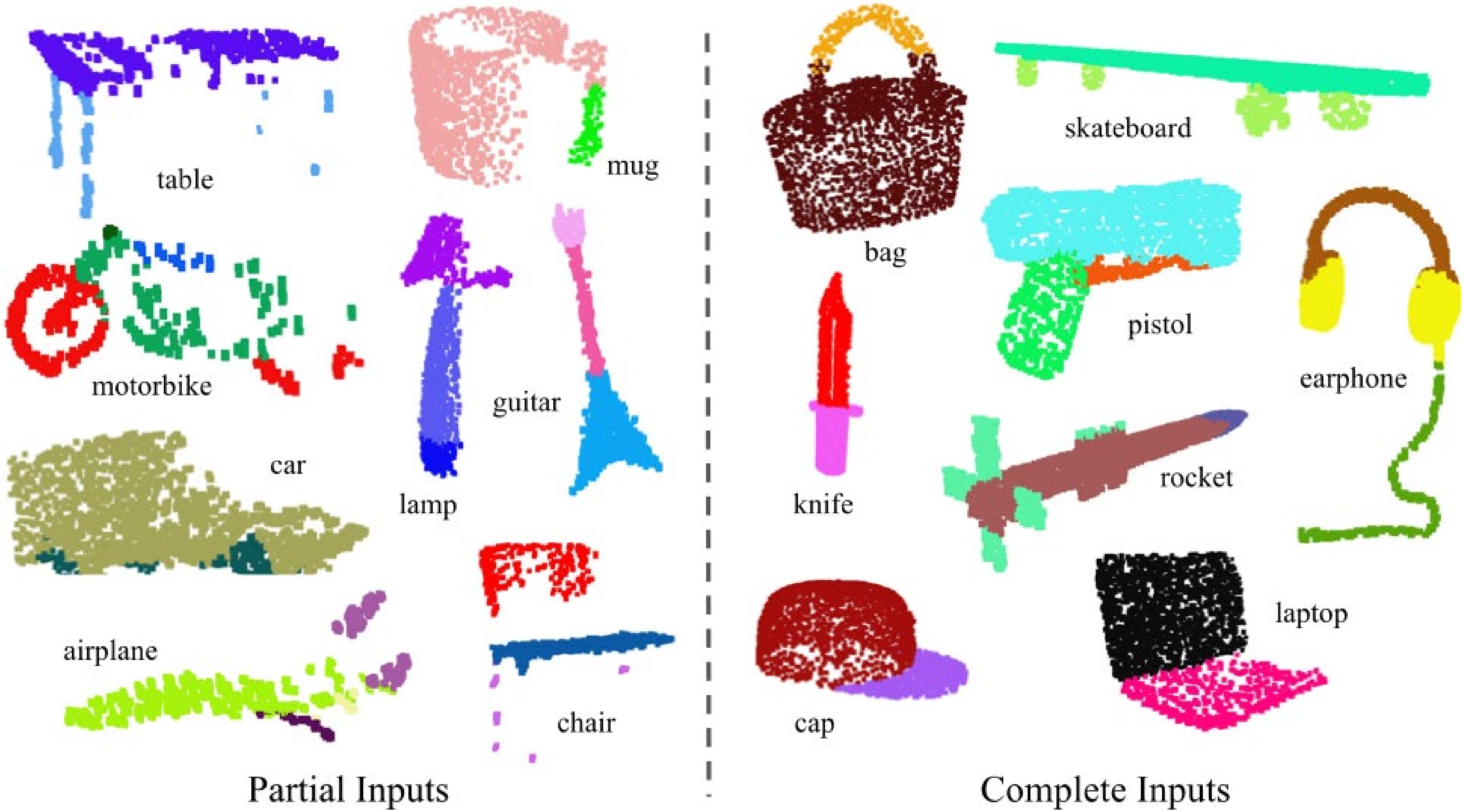
**Sampling Invariance**

Output a function of the underlying geometry and not the sampling

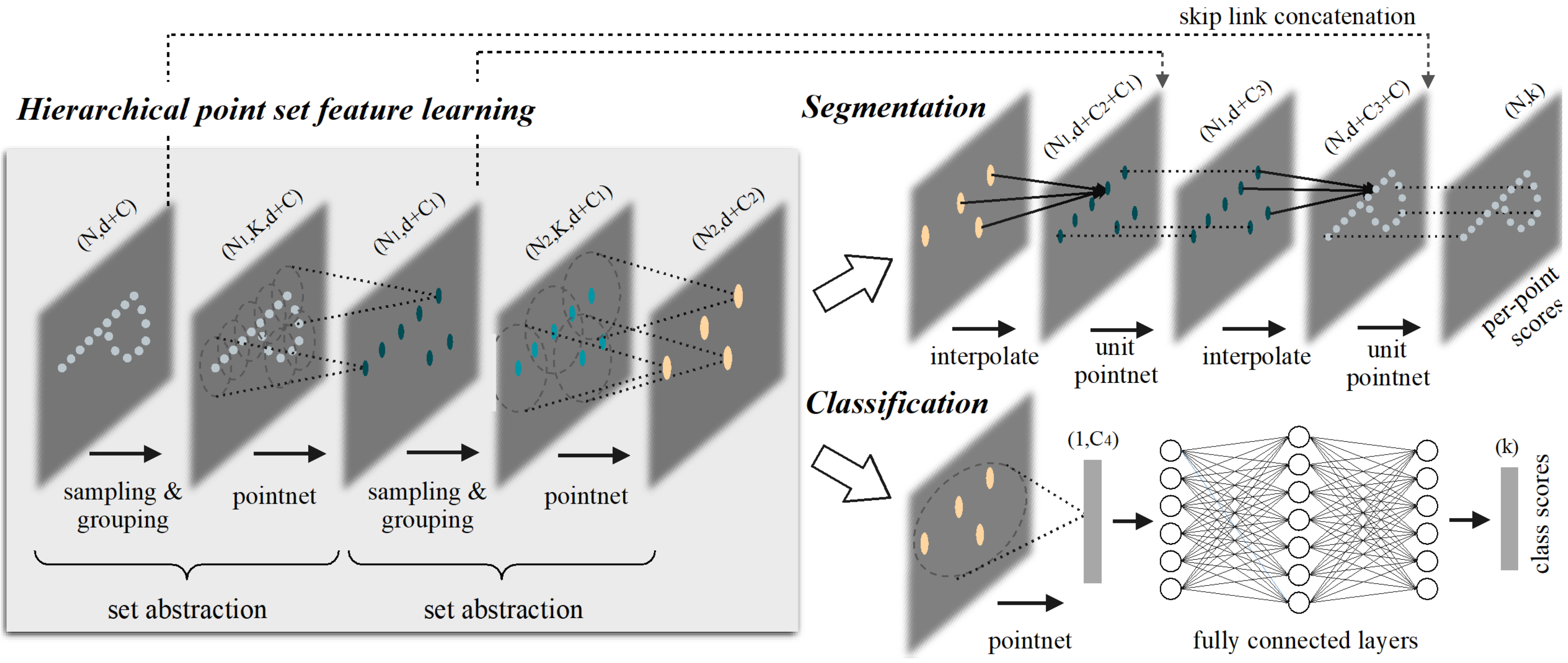
# PointNet Classification and Segmentation



# Results on Object Part Segmentation

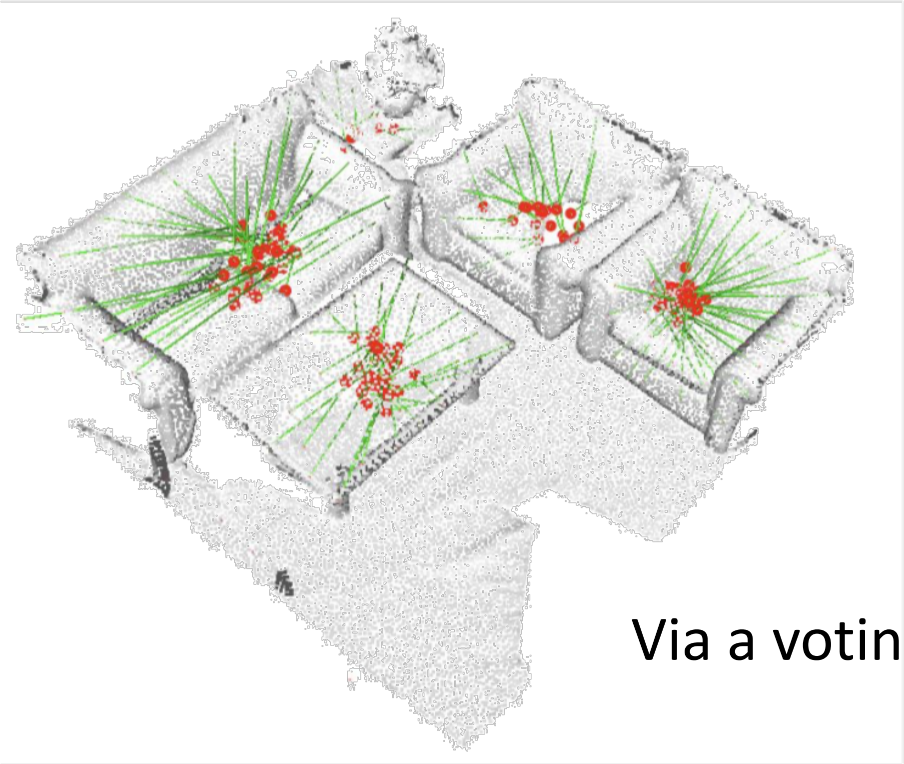


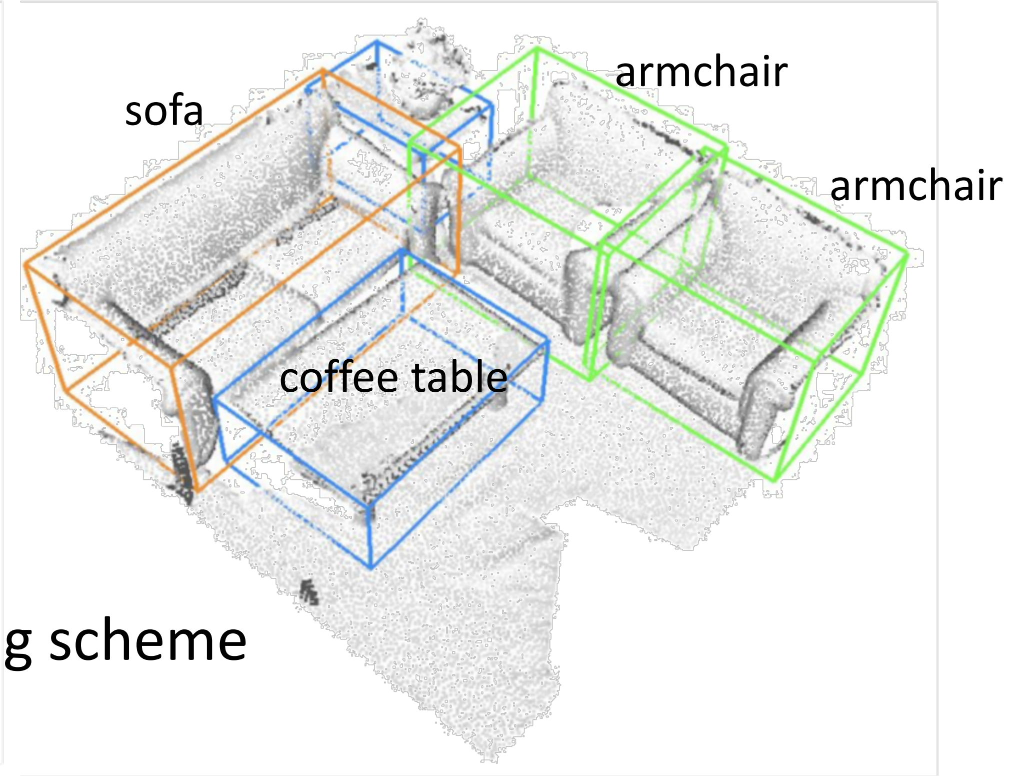
# PointNet++ for Classification and Segmentation




Aggregation pattern is only a function of the spatial locations of the points

# Point Cloud Object Amodal Bounding Box Detection

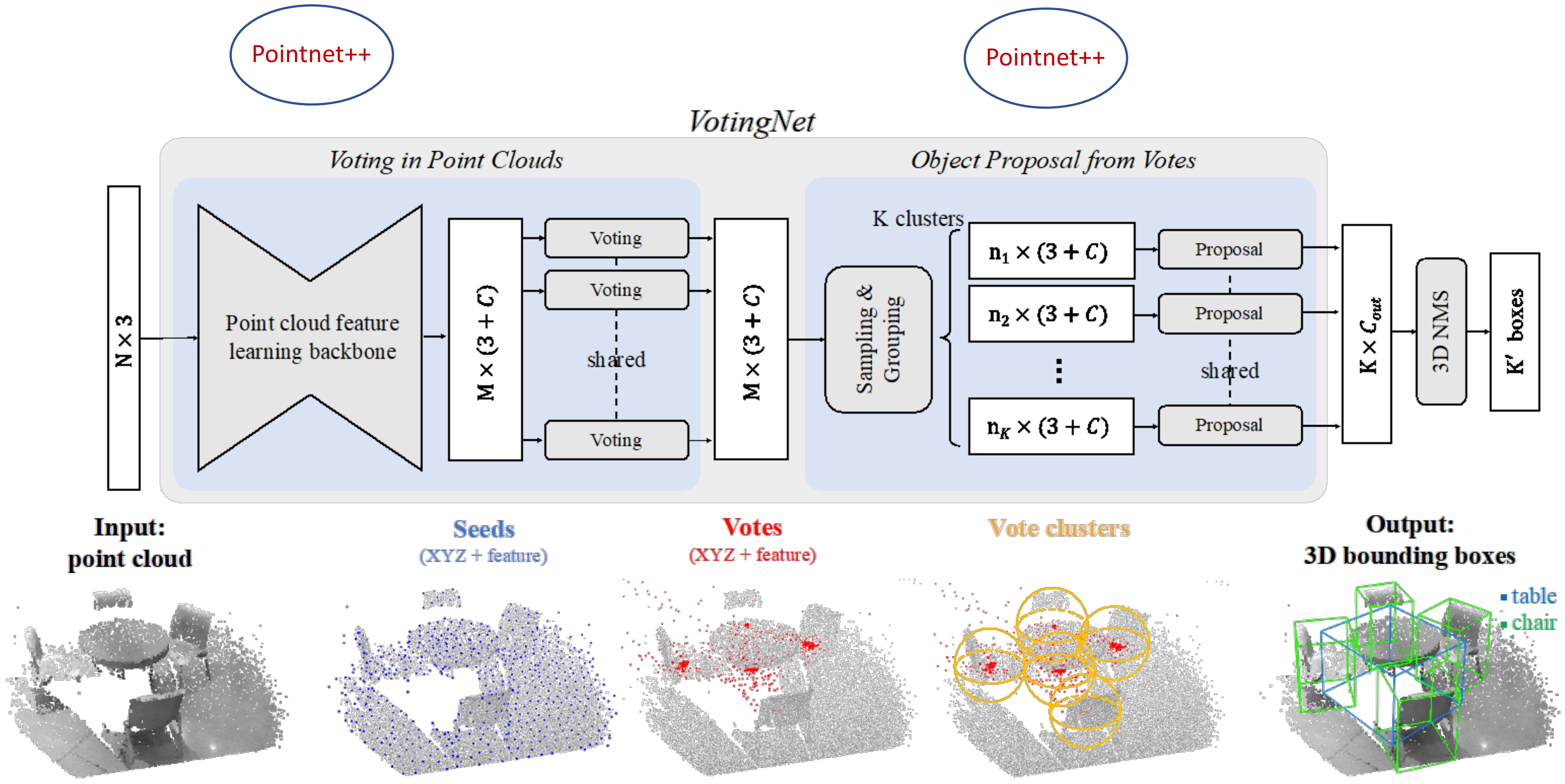




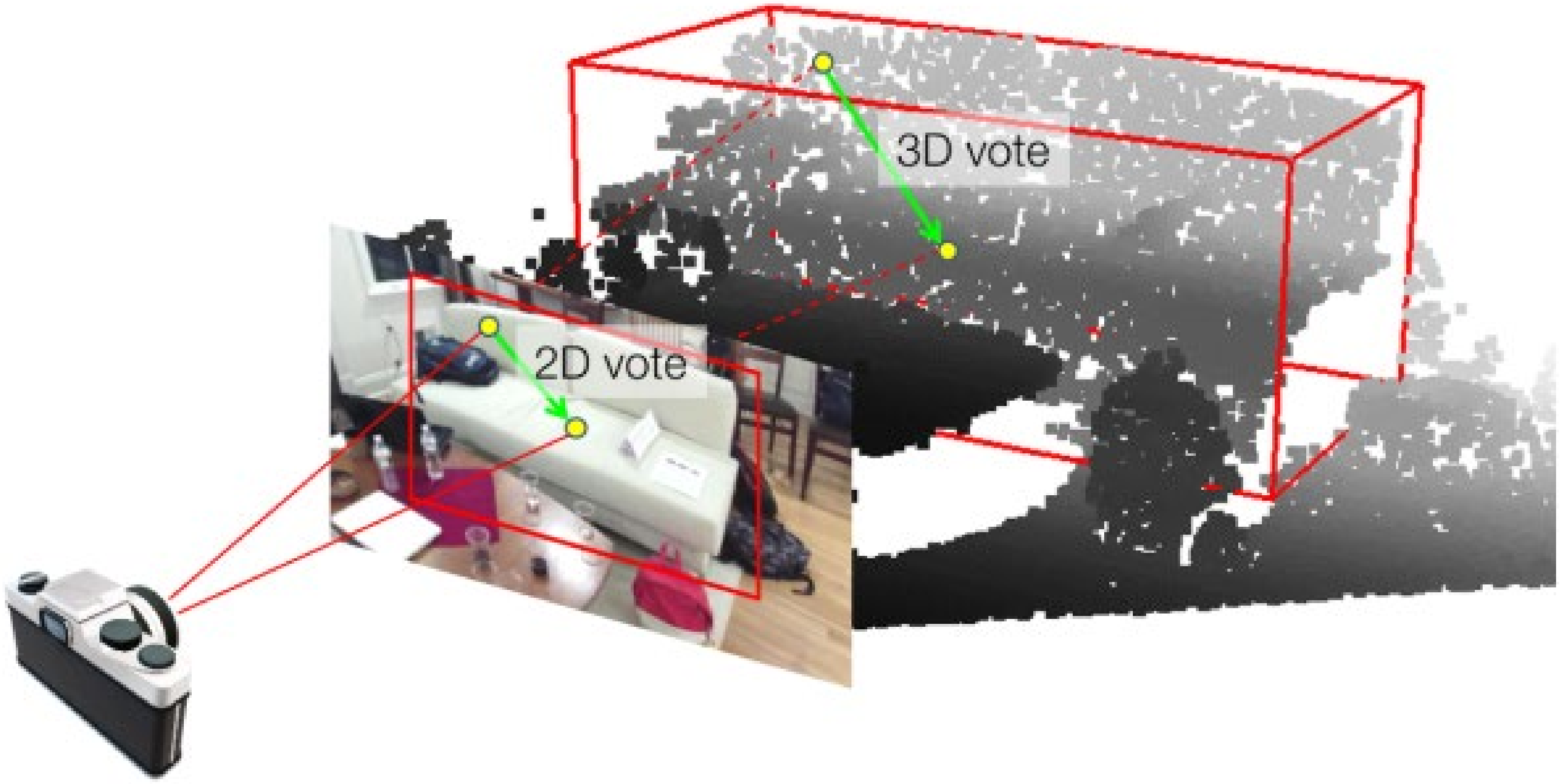

Via a voting scheme

- Charles R. Qi, Or Litany, Kaiming He, Leonidas J. Guibas. *Deep Hough Voting for 3D Object Detection in Point Clouds*. ICCV 2019.
- Charles R. Qi, Xinlei Chen, Or Litany, Leonidas J. Guibas. *ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes*. CVPR 2020.

# VoteNet – A Two-Stage Approach

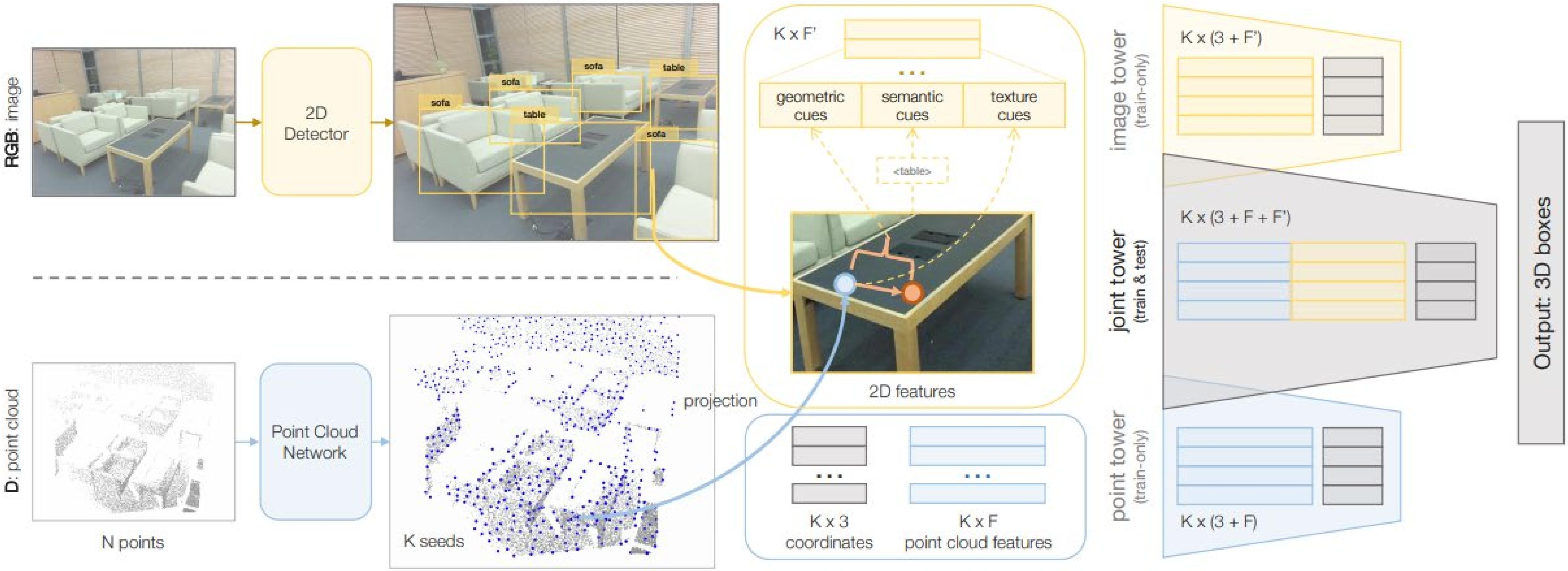


# Basic idea: *ImVoteNet*

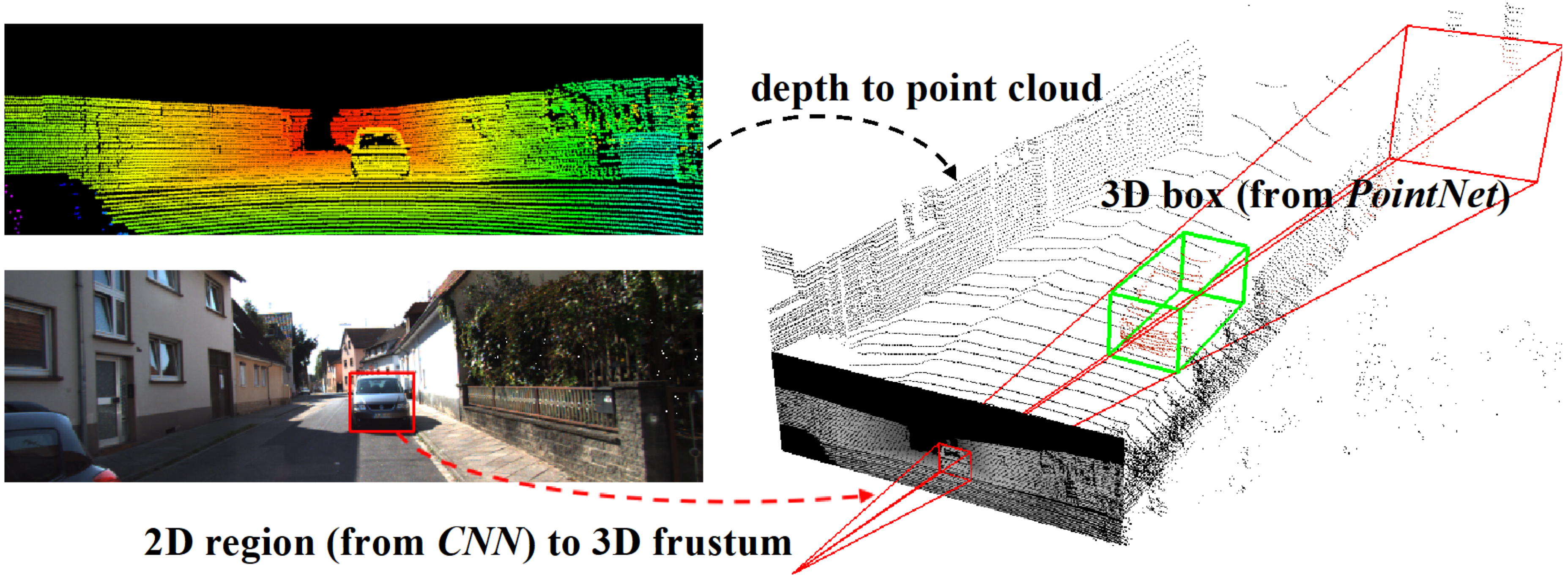



C. Qi, X. Chen, O. Litany, L.J. Guibas CVPR 2020

# *ImVoteNet* Architecture



# Frustum PointNets for 3D Object Detection

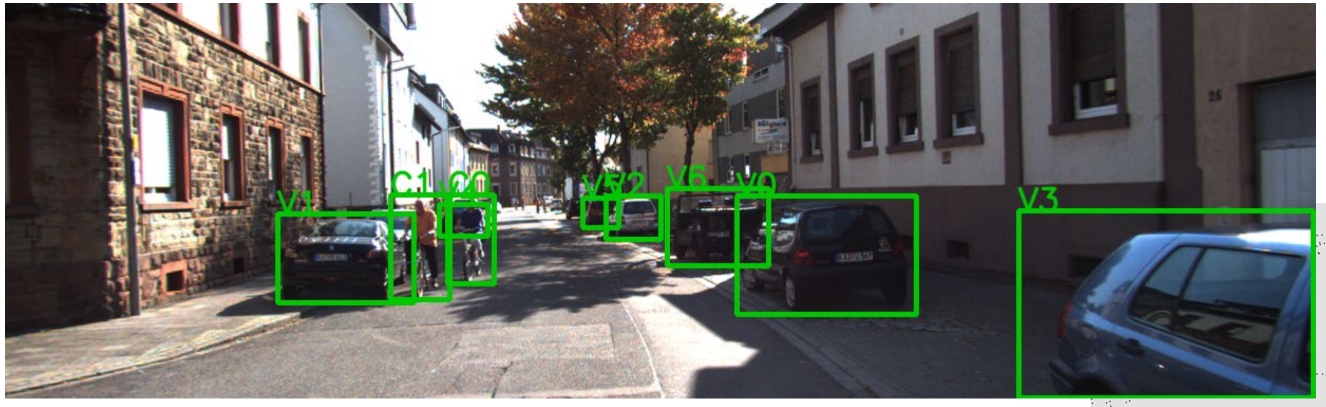



***+ Leveraging mature 2D detectors for region proposal. greatly reducing 3D search space.***

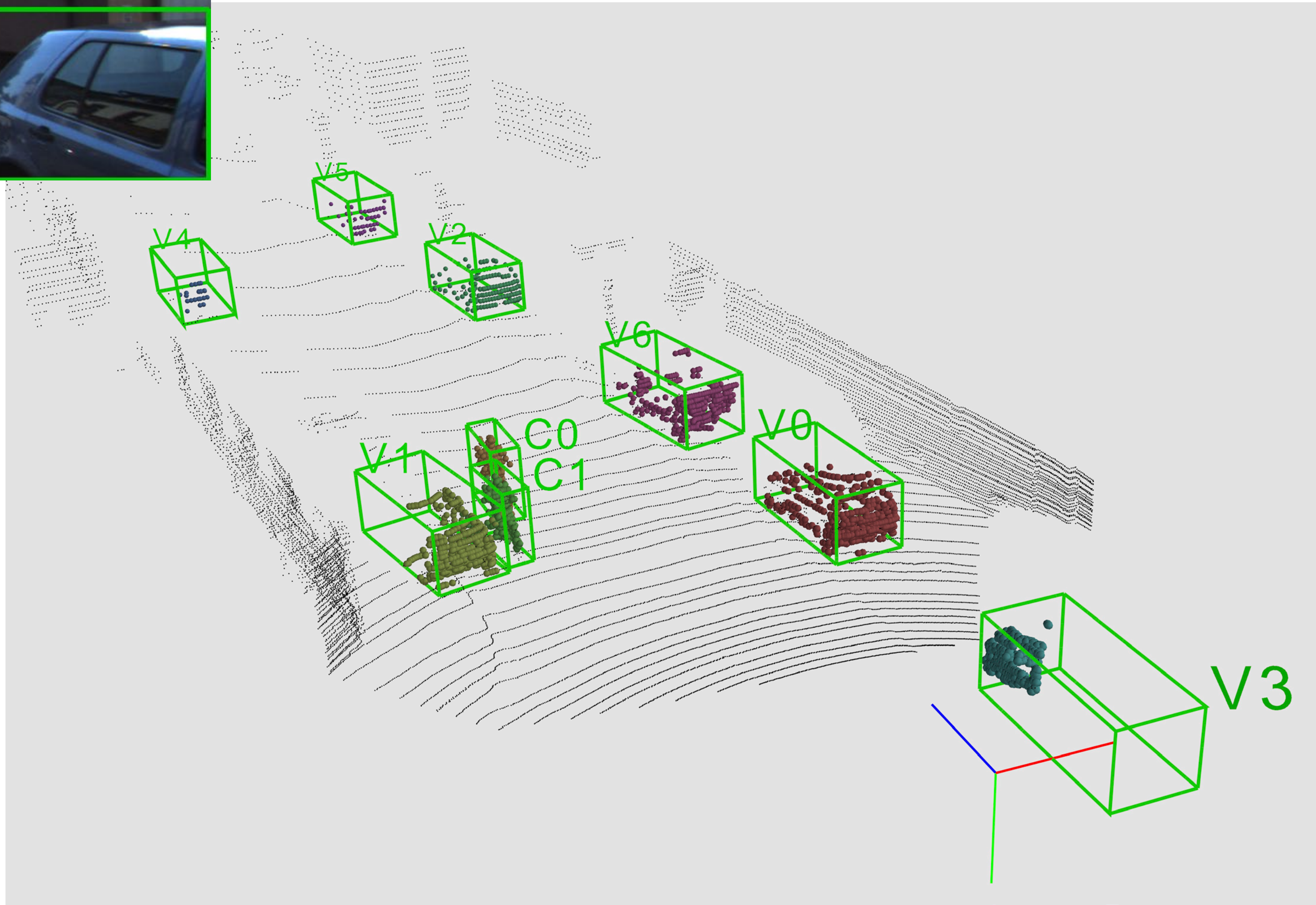
***+ Solving 3D detection problem with 3D data and 3D deep learning.***

*Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, Leonidas Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data (CVPR 2018)*

# KITTI Results: Qualitative

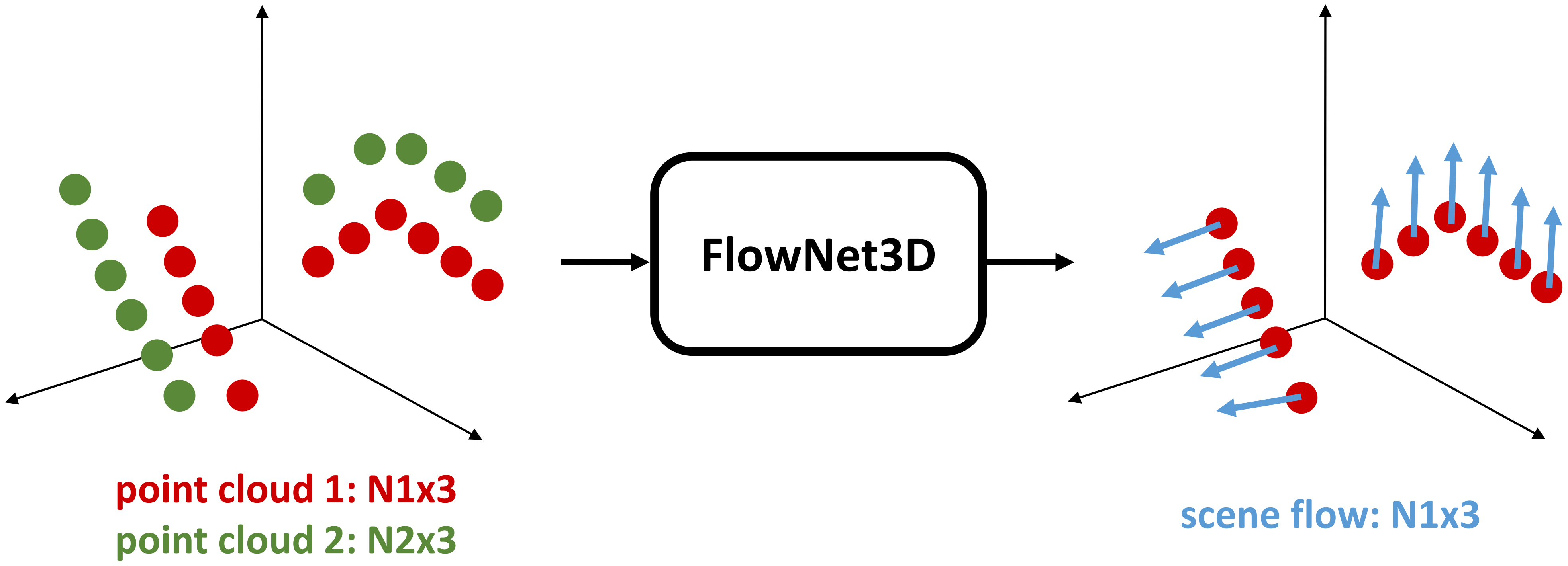



Remarkable box estimation accuracy even with a dozen of points or with very partial point clouds.



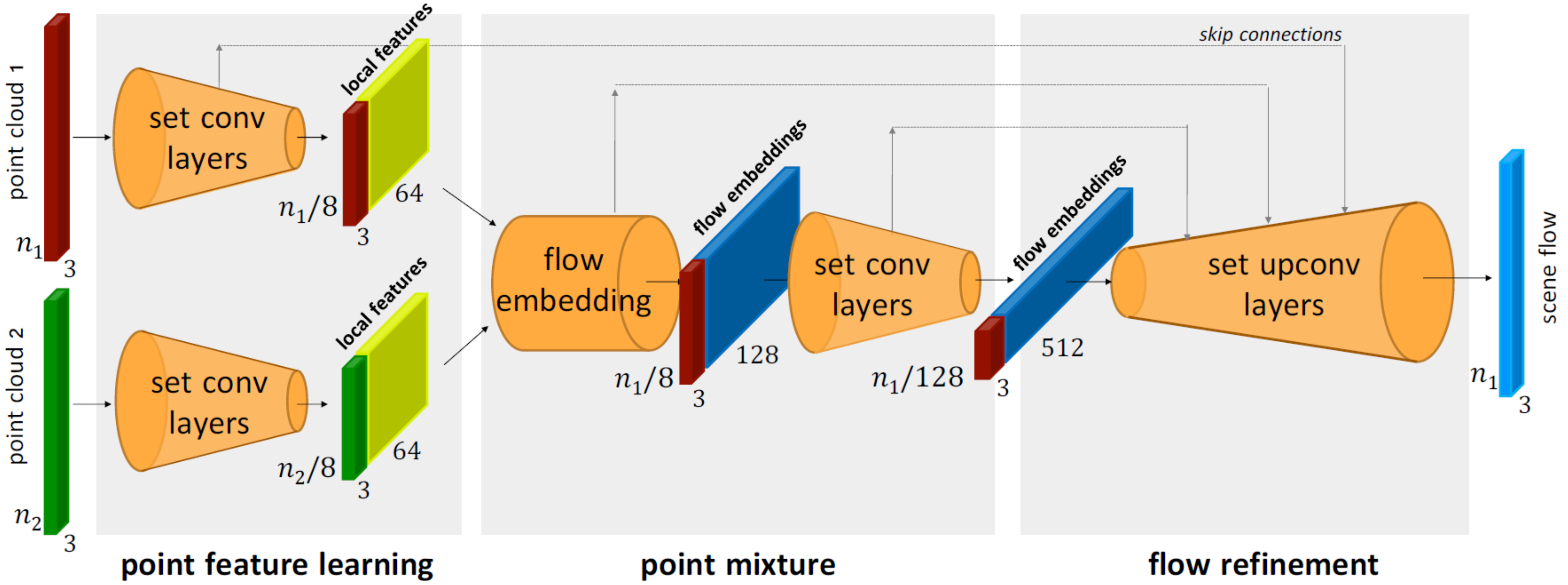
# FlowNet3D – Point Clouds Over Time

- Directly learning scene flow in 3D point clouds, with 3D deep learning architectures.




*Xingyu Liu, Charles R. Qi, Leonidas Guibas. Learning Scene Flow in 3D Point Clouds, (CVPR 2019).*

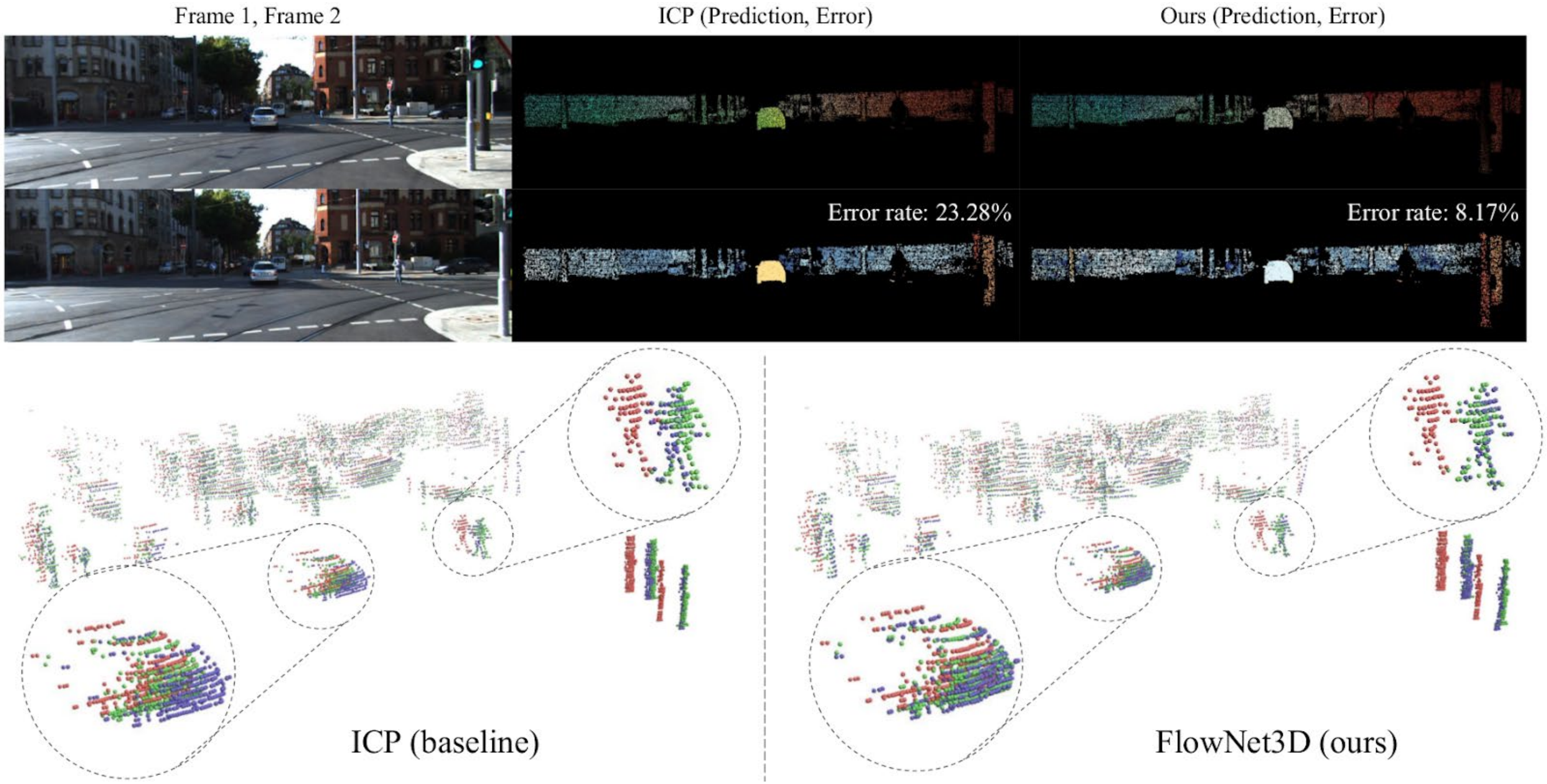
# FlowNet3D – Point Clouds Over Time



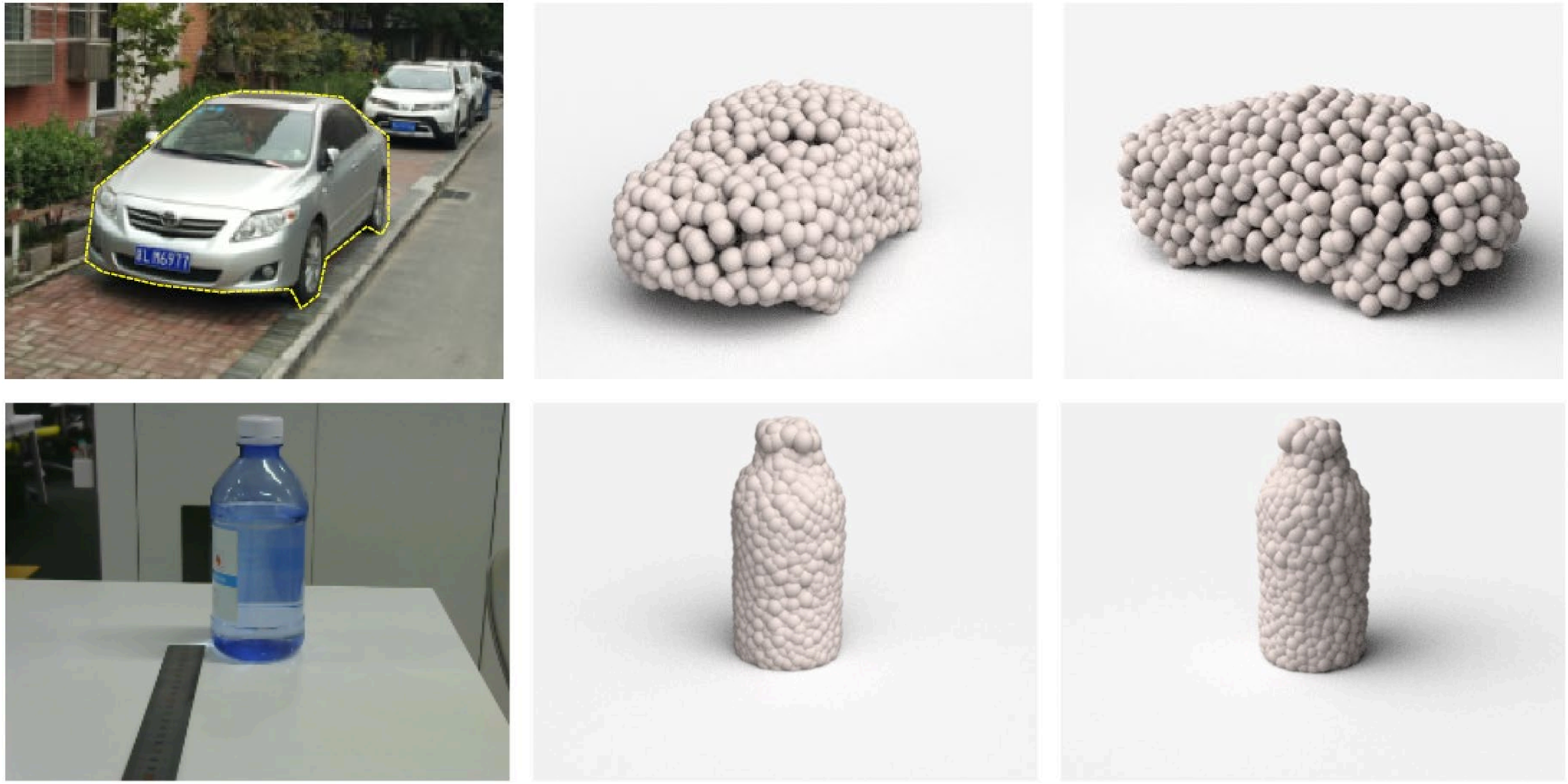

Composed of many many mini-pointnet++ modules …

Pointnet++

# KITTI Results



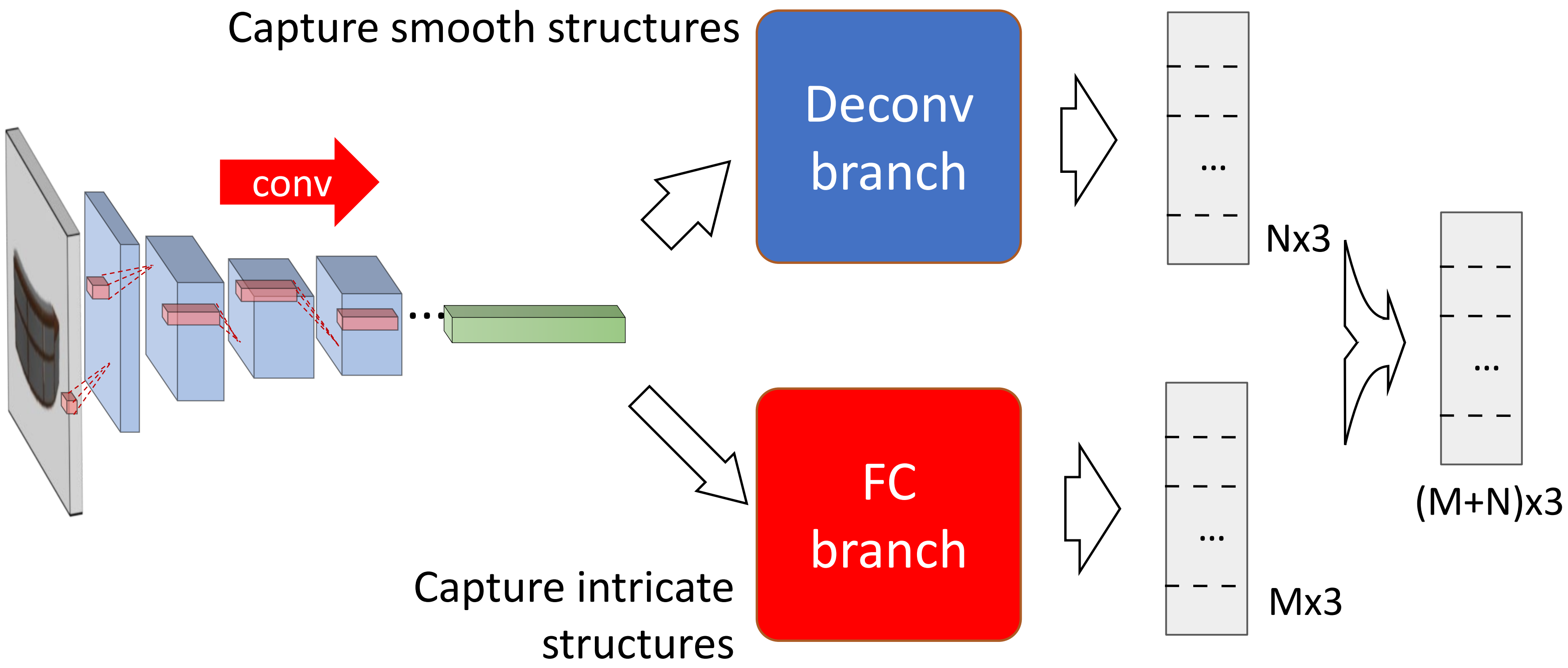
# Point Cloud Synthesis from a Single Image



Input Reconstructed 3D point cloud

[H. Su, H. Fan, LG, 2017]

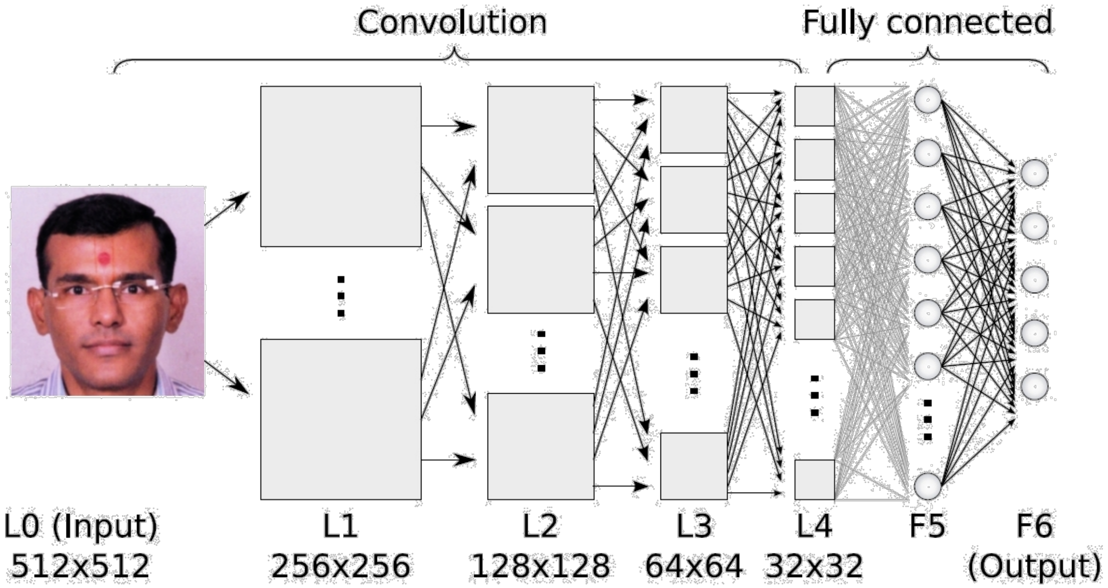
# Two-Branch Architecture



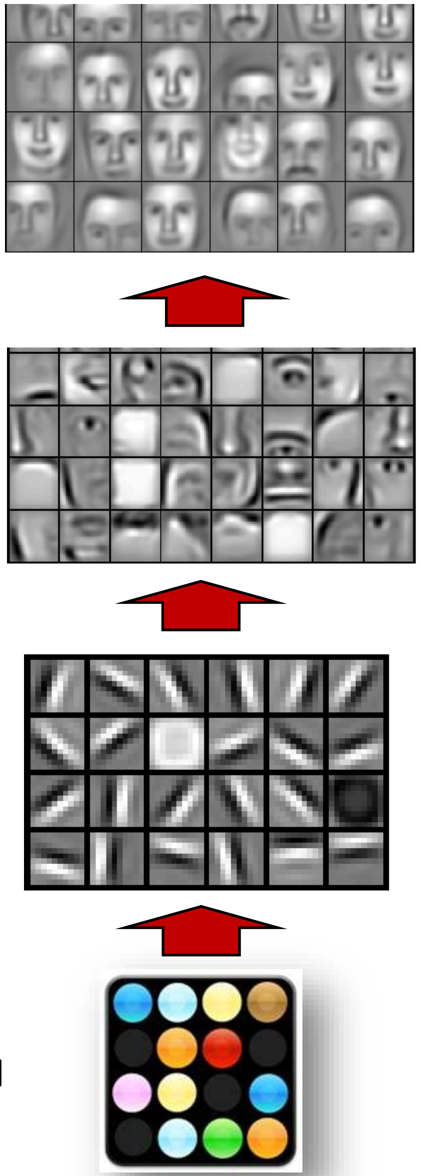

**Set union by array concatenation**

# Today: Functional Maps for Joint Data Analysis

# Vertical Learning Networks




[Makwana, 2016]



[Lee et al., 2009]

Data-driven feature learning at ascending abstraction layers

“Deep” nets

# Horizontal Networks

Similarity as a communications channel

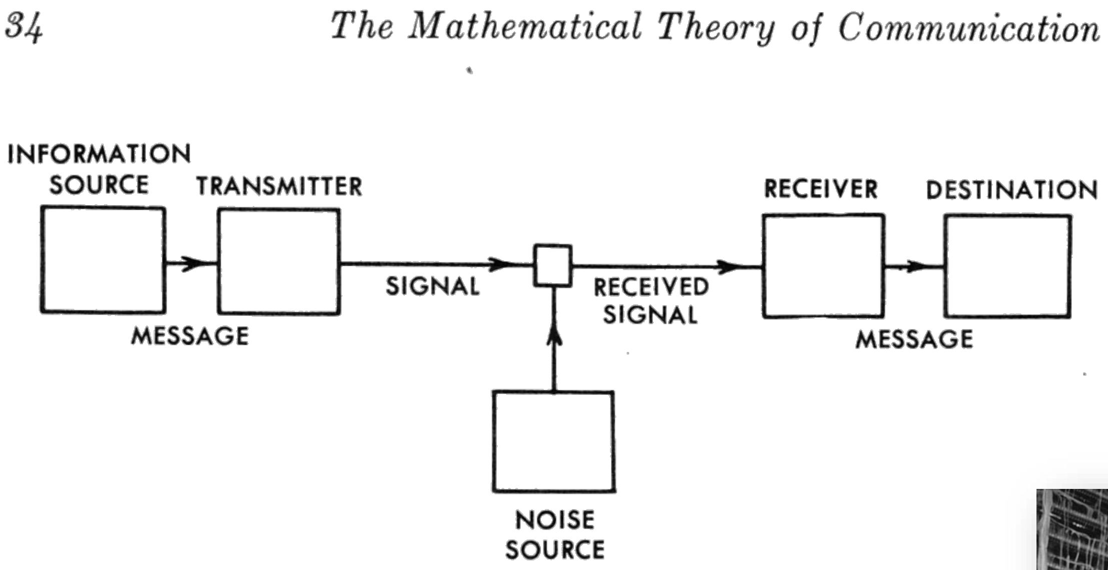
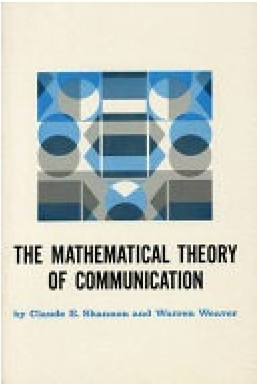
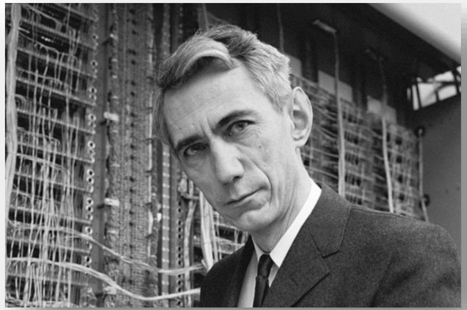



Fig. 1. — Schematic diagram of a general communication system.

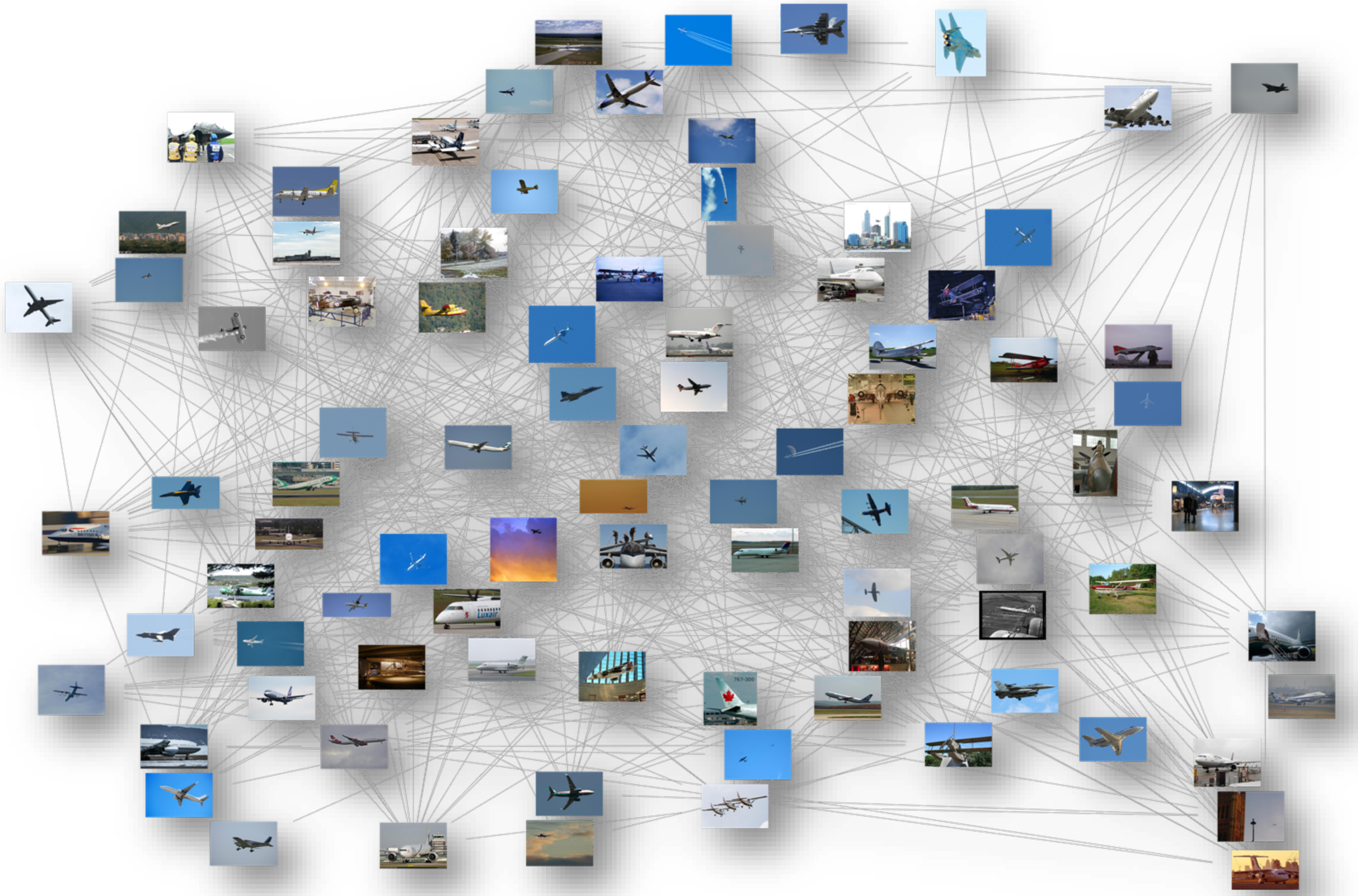



Claude Shannon

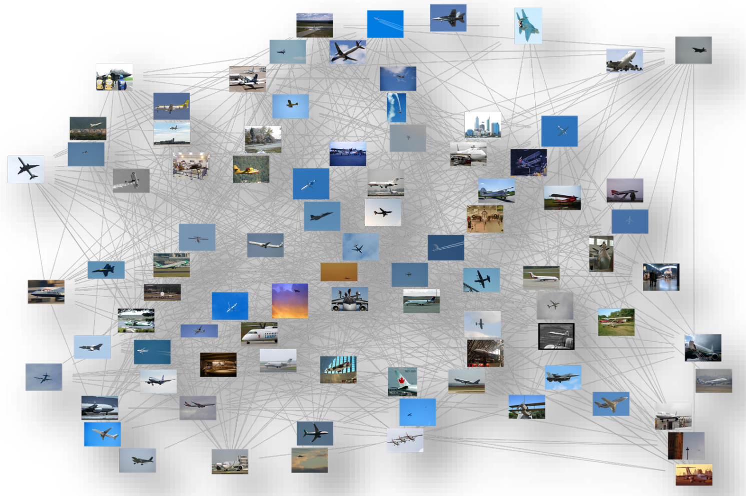


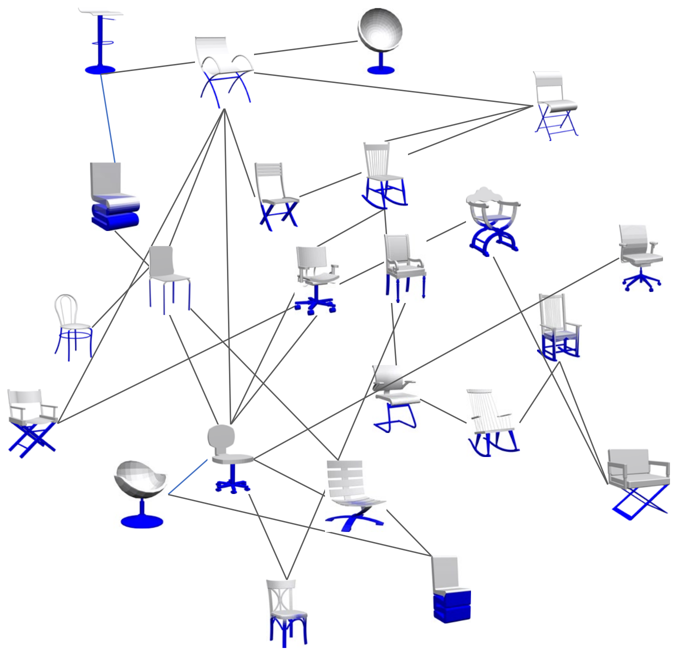
# The Network View: Information Transport Between Visual Data

# Networks of Images



# Or of Shapes, Or of Both



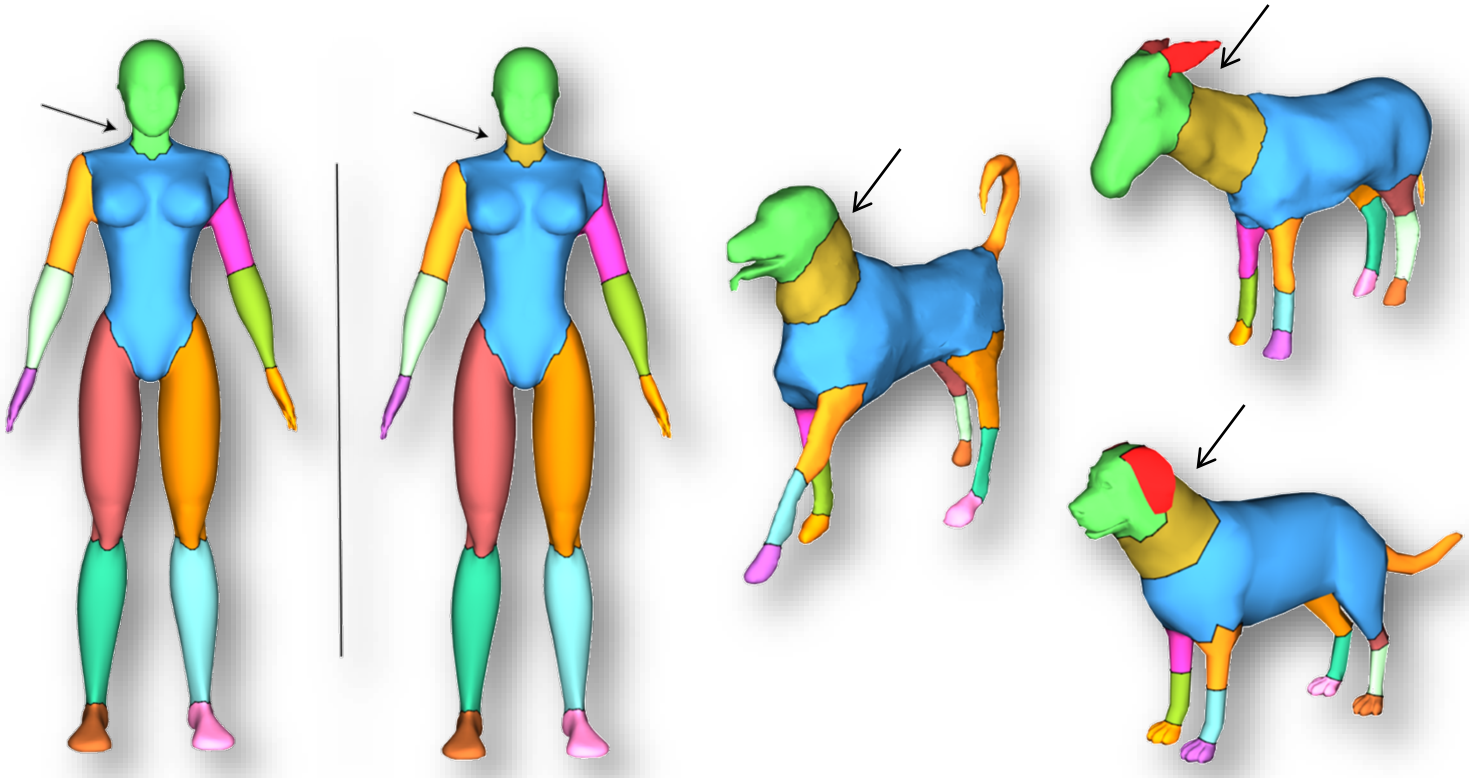


# Relations Between Visual Data

# Each Data Set Is Not Alone

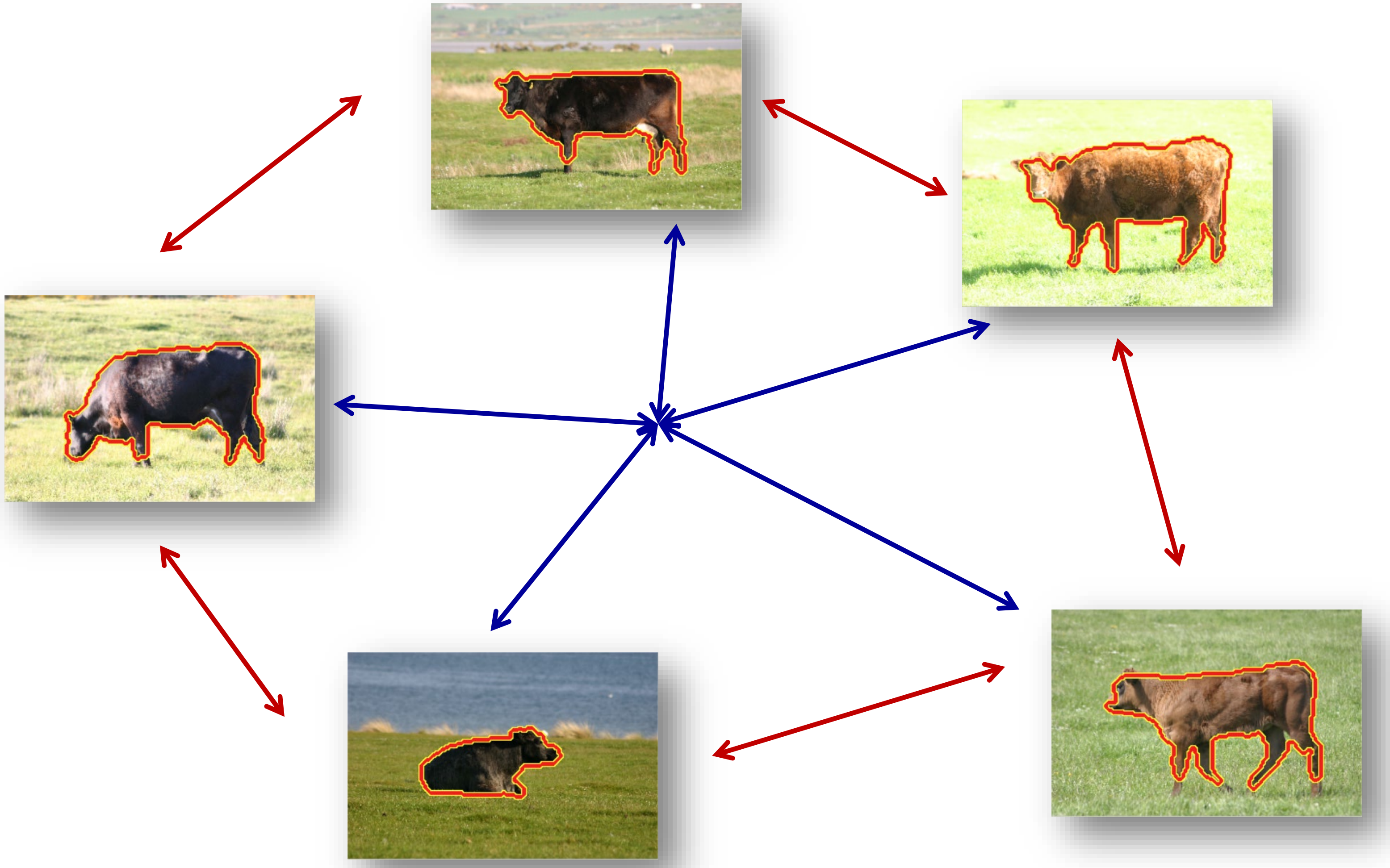
[Q. Huang, V. Koltun, L. Guibas, 2011]

- The interpretation of a particular piece of geometric data is deeply influenced by our interpretation of other related data

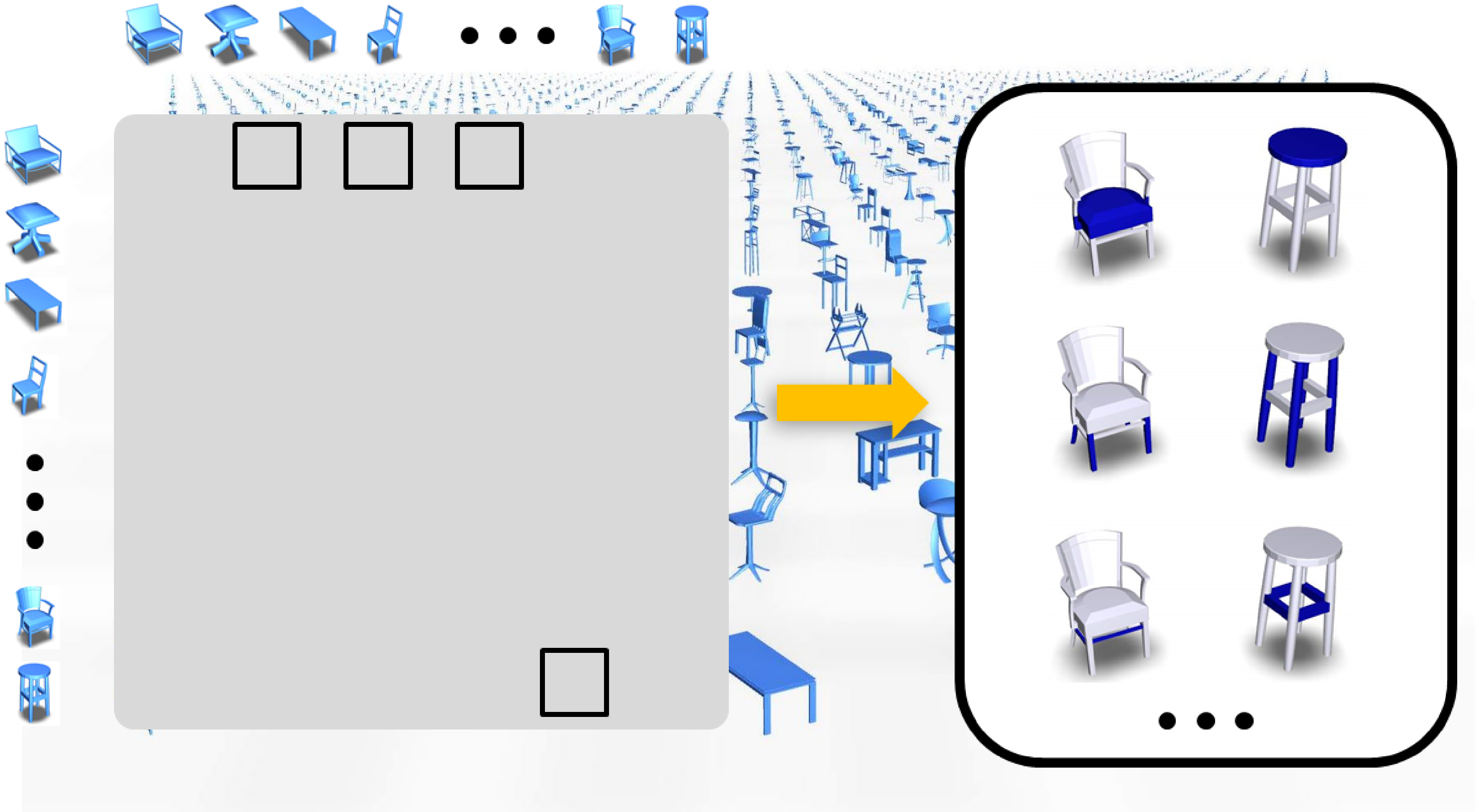


3D Segmentation

# Co-Segmentation

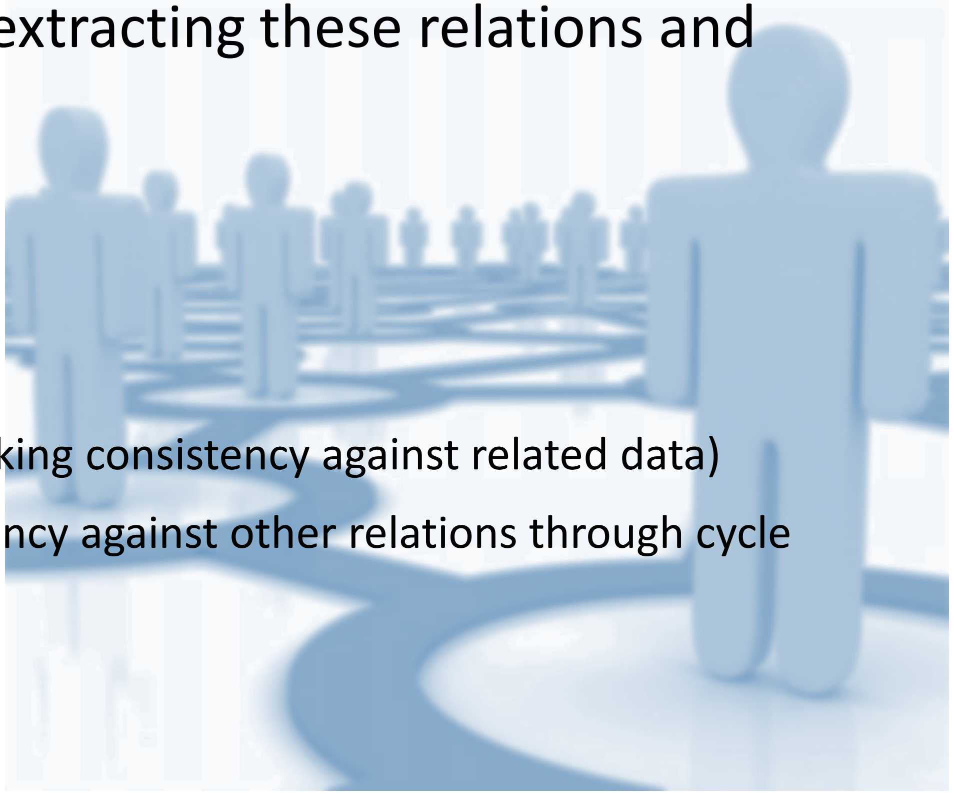


# Semantic Structure Emerges from the Network



[Q. Huang, F. Wang, L. Guibas, ’14]

# Societies, or Social Networks of Data Sets



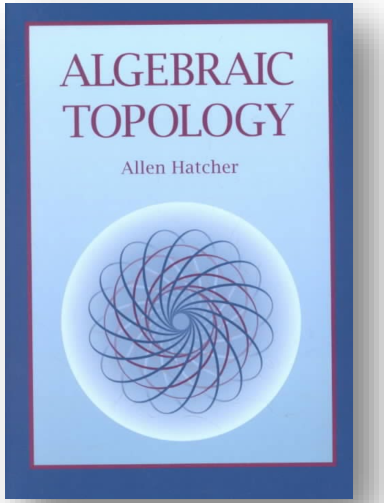
Our understanding of data can greatly benefit from extracting these relations and building relational networks.

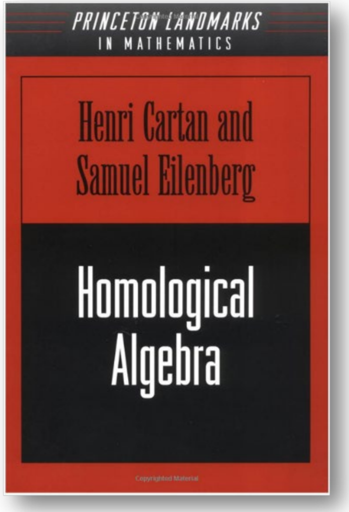
We can exploit the relational network to

- transport information around the network
- assess the validity of operations or interpretations of data (by checking consistency against related data)
- assess the quality of the relations themselves (by checking consistency against other relations through cycle closure, etc.)
- extract shared structure among the data

Thus the network becomes the great regularizer in joint data analysis.

# V+H: Functorial Data Analysis




$$\begin{array}{ccc} H_*(X) & \xrightarrow{\phi} & H_*(Y) \\ \uparrow H_* & & \uparrow H_* \\ X & \xrightarrow[f]{} & Y \end{array}$$

Topological Spaces

Algebraic structures: vector spaces, groups

$$\begin{array}{ccc} L(X) & \xrightarrow{\phi} & L(Y) \\ DN \uparrow & & \uparrow DN \\ X & \xrightarrow[f]{} & Y \end{array}$$

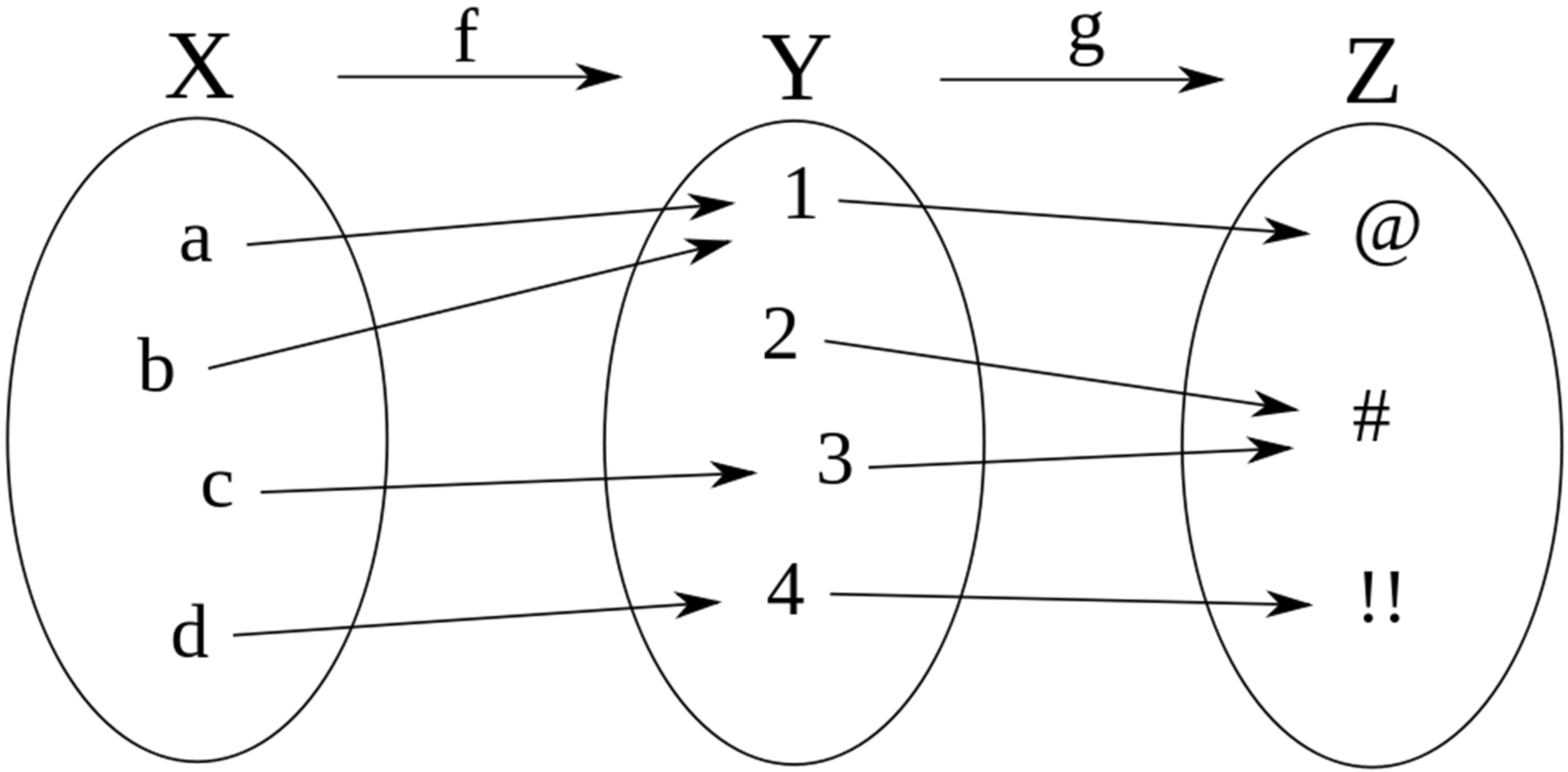
# Maps as First-Class Citizens

# Maps

$$\phi : X \rightarrow Y$$

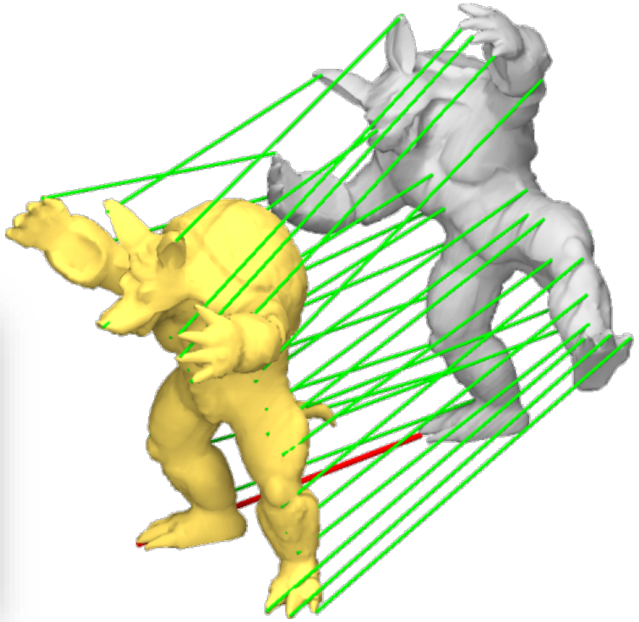
**Map from *X* to *Y***

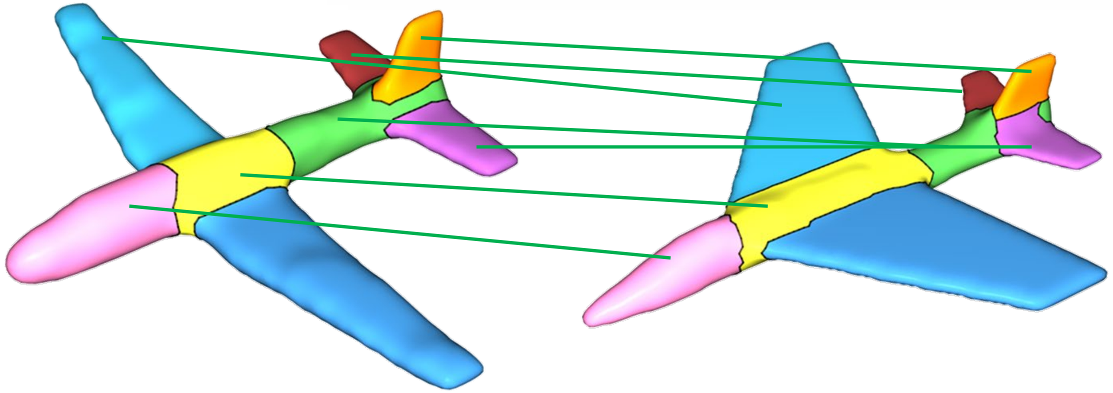
# Algebraic Structure: Map Composition



# Relationships as Correspondences or Maps

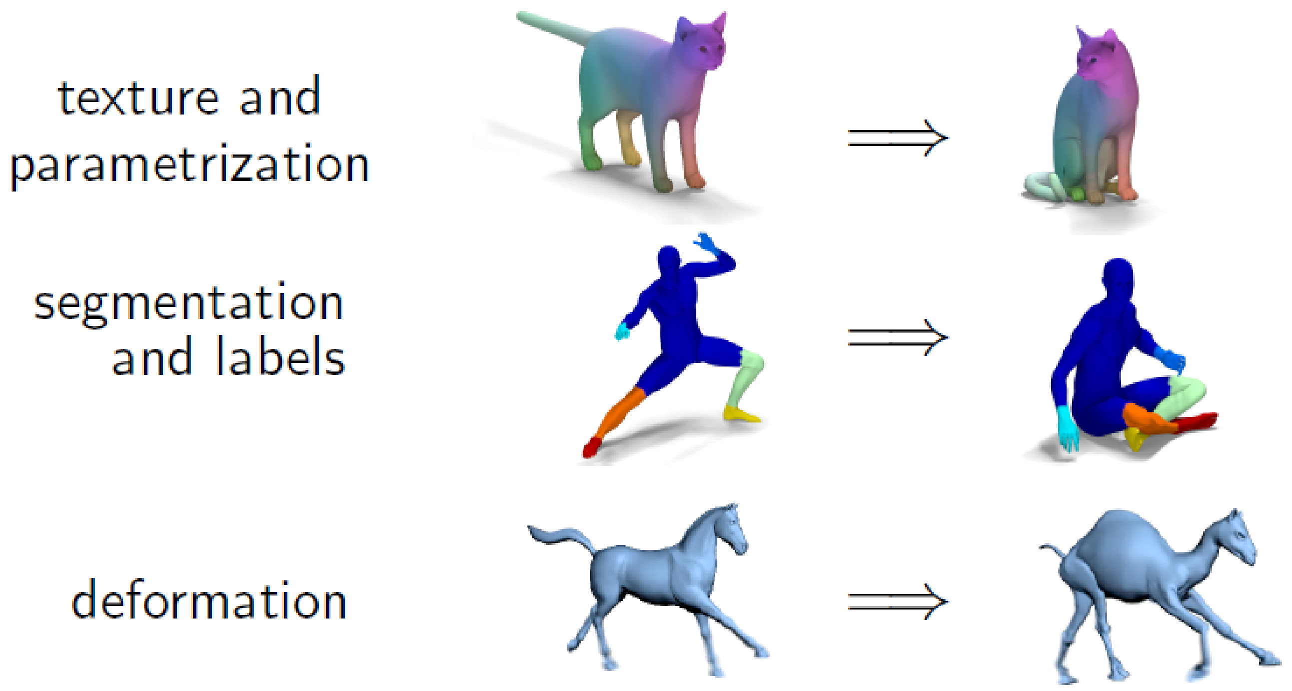
- Multiscale mappings
  - Point/pixel level
  - part level





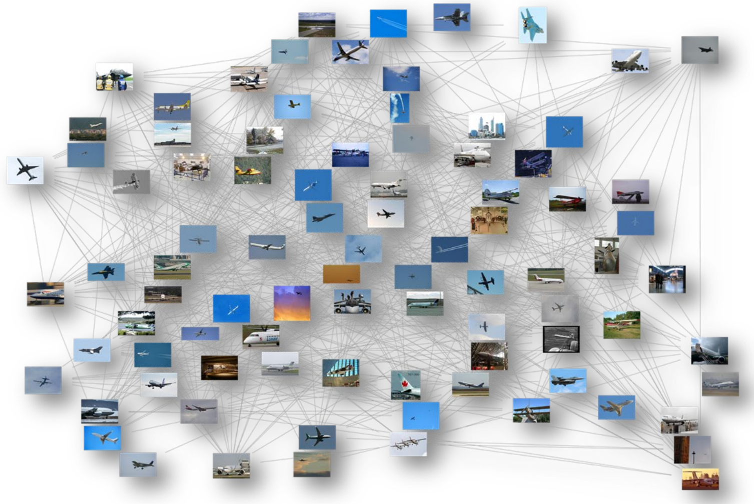
Maps capture what is the same or similar across two data sets

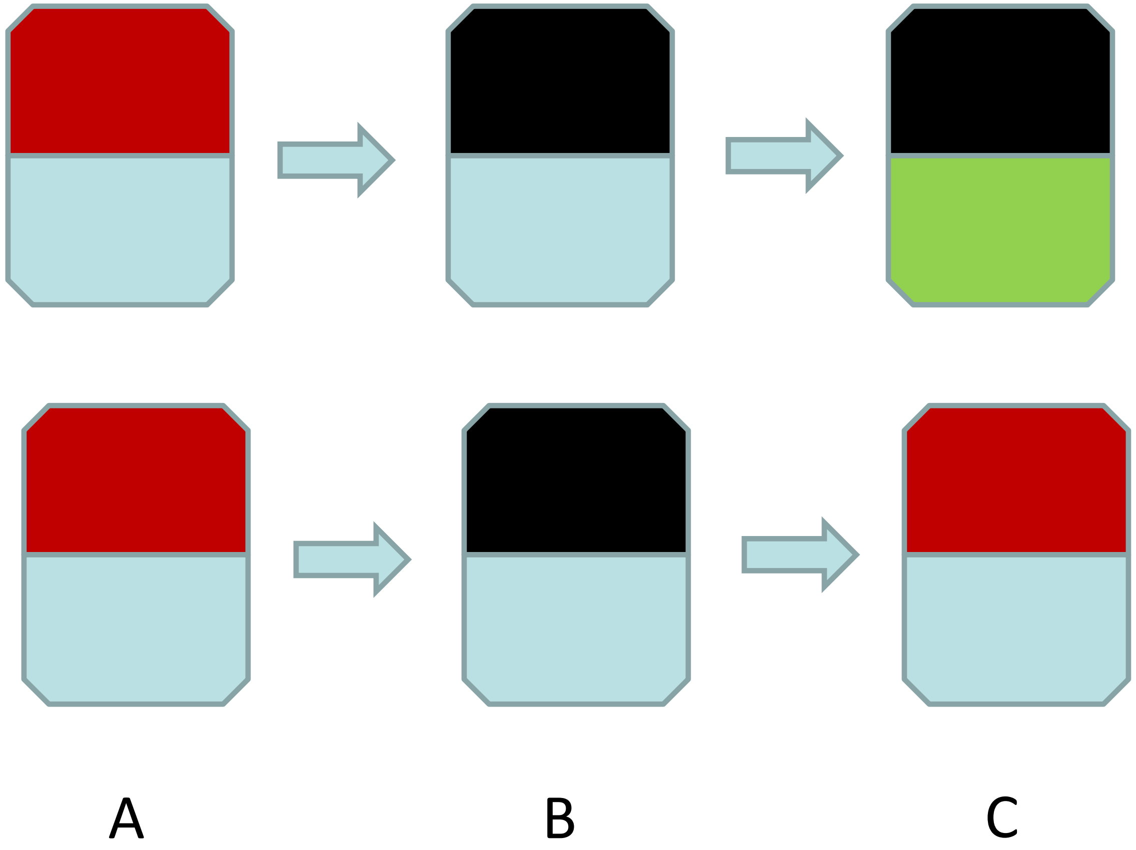
# Correspondences or Maps are Information Transporters



# Matching Enables Information Transport

# Maps vs. Distances/Similarities Networks vs. Graphs

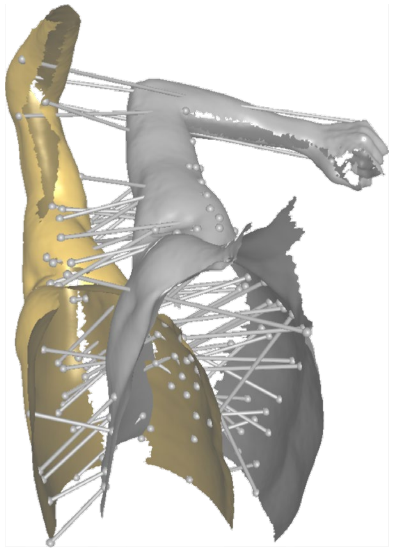





Persistence of correspondences

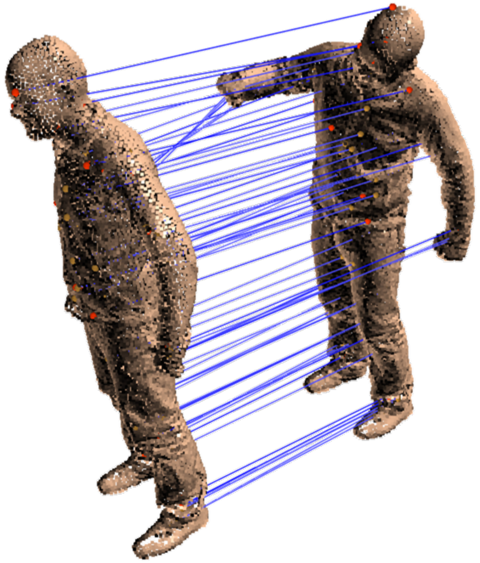
# What Makes a Map Good?

- $1^{st}$ order descriptor or feature preservation (e.g., corners go to corners)
- $2^{nd}$ order attribute preservation (e.g., Euclidean or geodesic distances)



- Smoothness or continuity
- Respect for “internal structure” (symmetries, etc)

- Semantic correctness may still be elusive

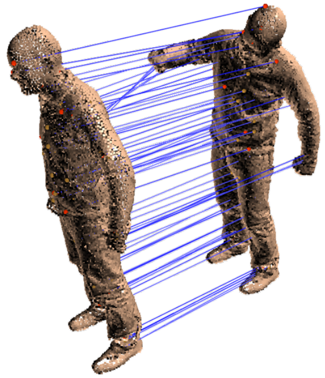


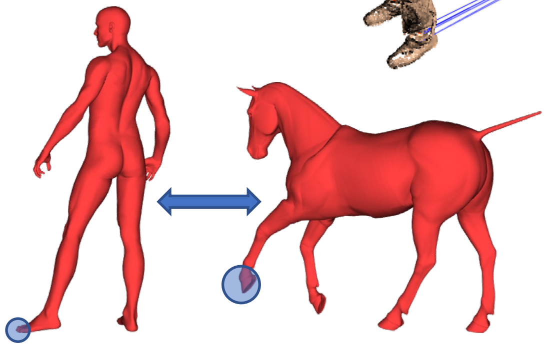
# Problems and Issues

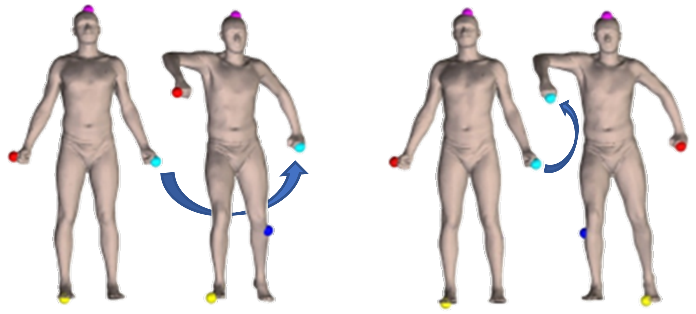

**Symmetry, ambiguity, scale, bad data**

# Maps Challenges: Representation and Computation

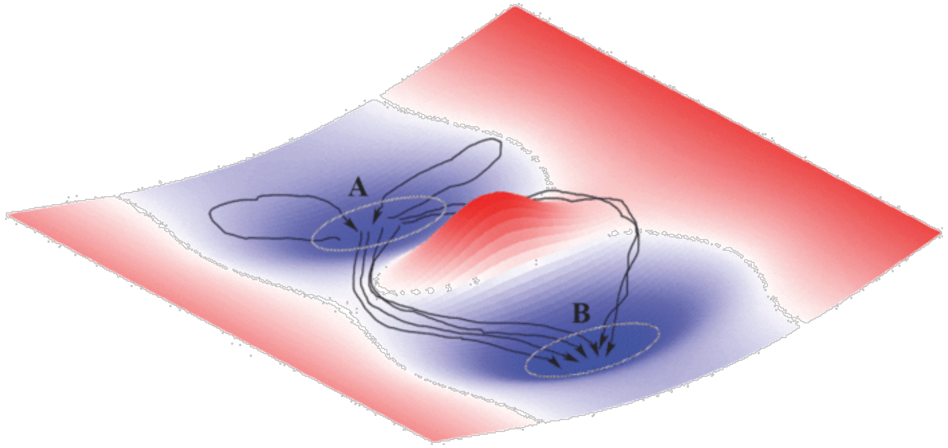
- Map representation as a data structuring problem
  - hard to encode compactly
  - hard to select correct scale
  - hard to enforce consistency across abstraction levels
- Also, difficult to compute
  - typically based on matching features/descriptors
  - symmetries, discrete and continuous, lead to ambiguities
  - $2^{nd}$ order attribute preservation leads to NP-hard quadratic assignment problems





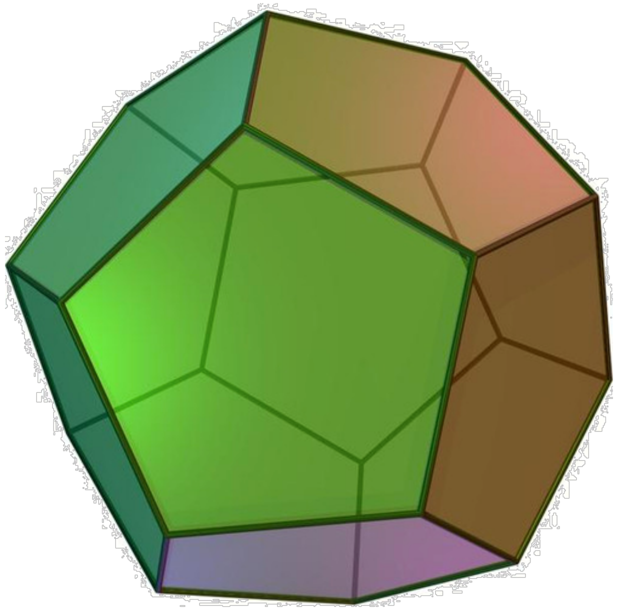


# Non-Convex, Combinatorial Optimization




multiple minima

NP-hard quadratic assignments

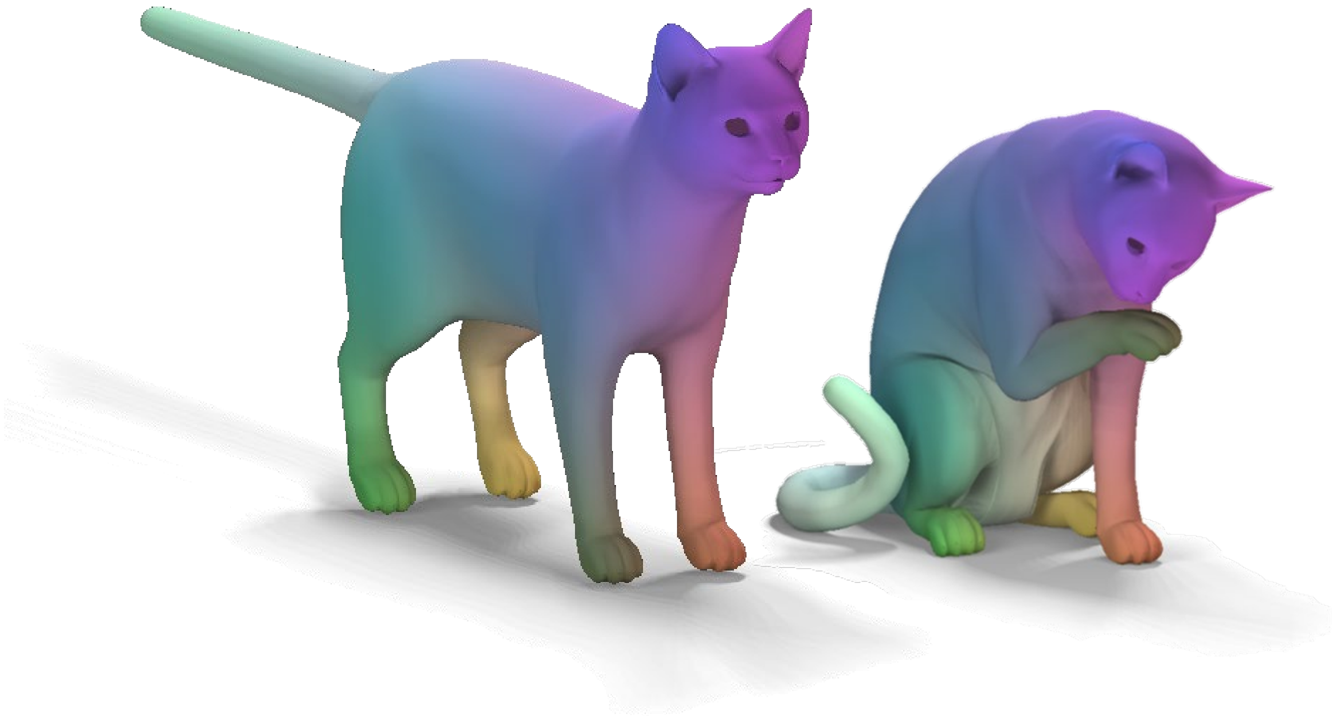


*n*! permutations

**Symmetry, ambiguity, scale, bad data**

# A Potential Way Out

Find alternative representation more amenable to optimization



**Redefine the notion of map**

# Technical Approach: Function Spaces and Functional Maps

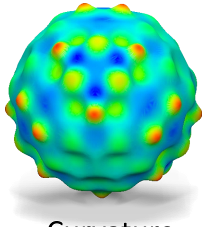
Shapes: From a Particle View to a Wave View …

# A Dual View: Functions and Operators

- Functions on data
  - Properties, attributes, descriptors, part indicators, etc.
  - But also beliefs, opinions, etc
- Operators on functions
  - Maps of functions to functions
    - Laplace-Beltrami operator on a manifold

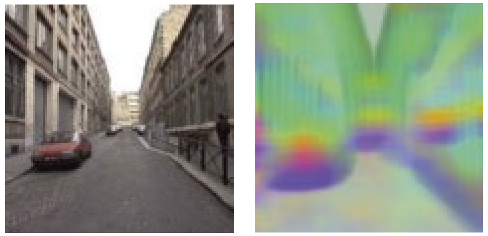
$$M$$

$$\Delta : C^{\infty}(M) \to C^{\infty}(M), \Delta f = \text{div } \nabla f$$

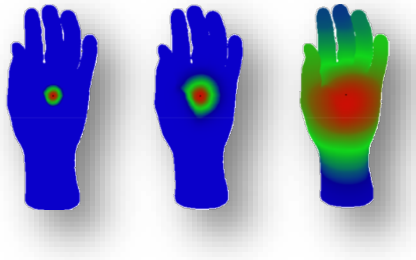


Curvature

Parts

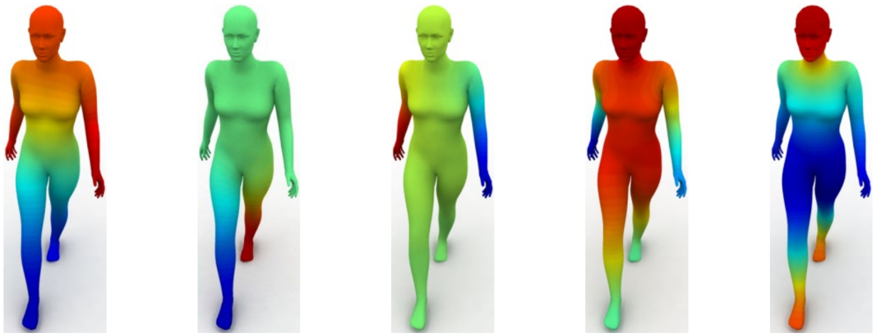


SIFT flow, C. Liu 2011

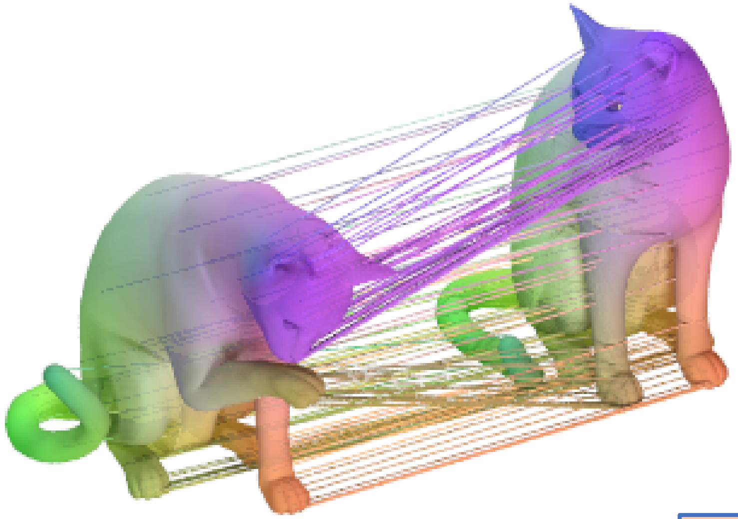


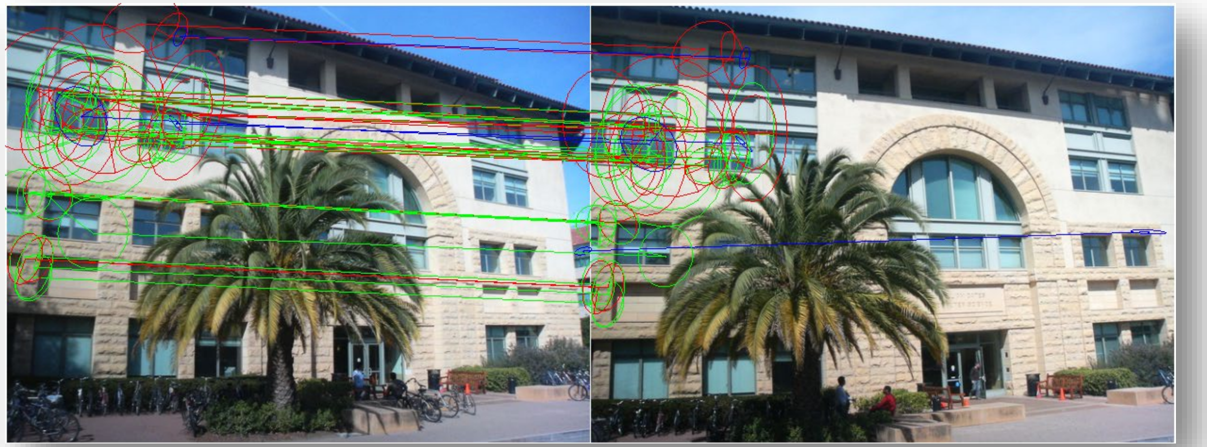
$$\frac{\partial u}{\partial t} = -\Delta u$$

heat diffusion



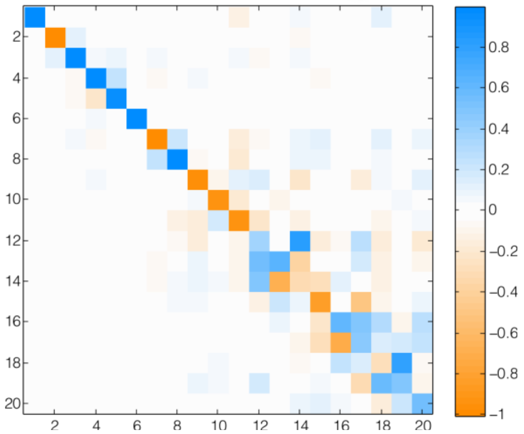
Laplace Beltrami eigenfunctions



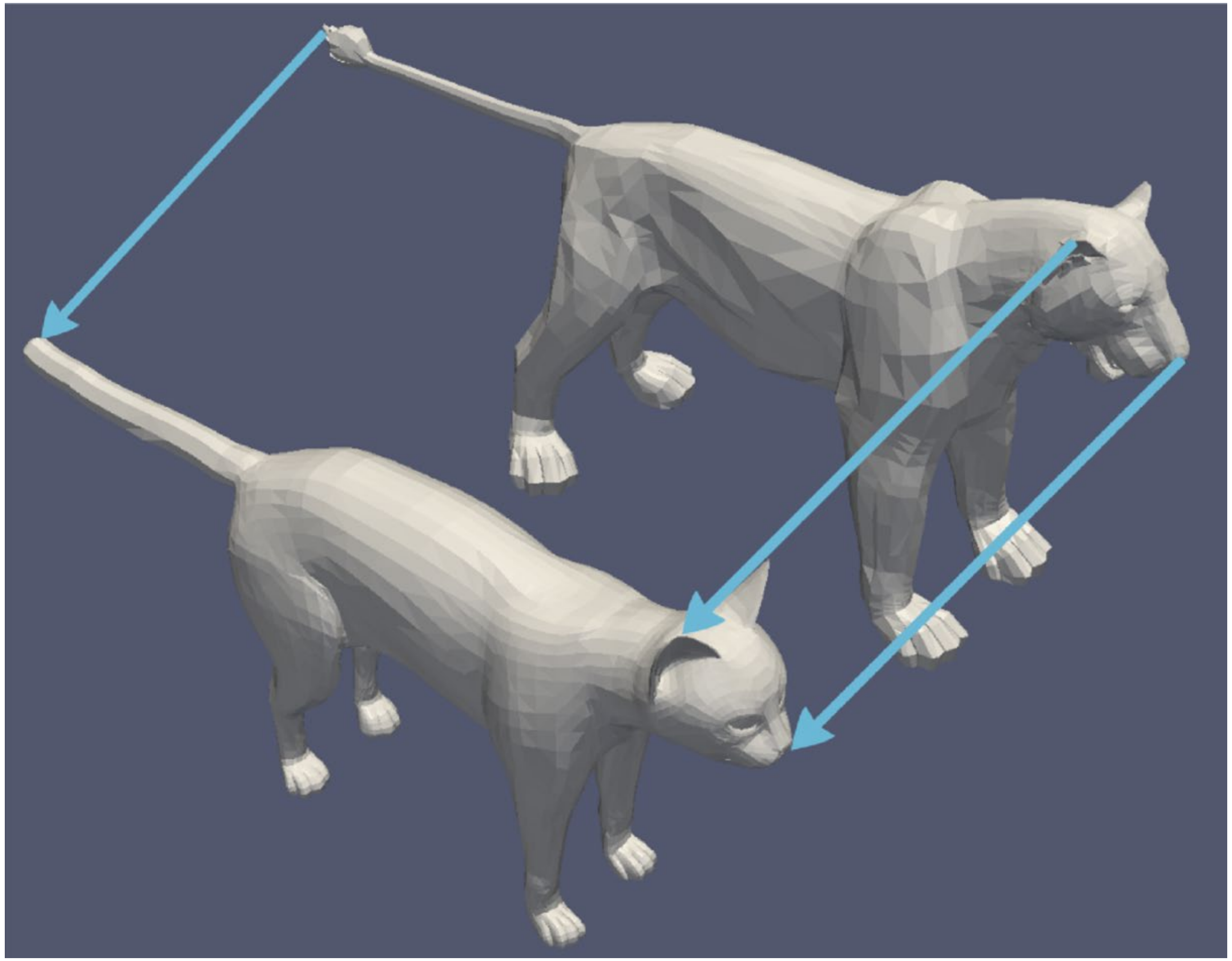


# Functional Maps (a.k.a. Operators)

[M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, L. G., Siggraph ’12]

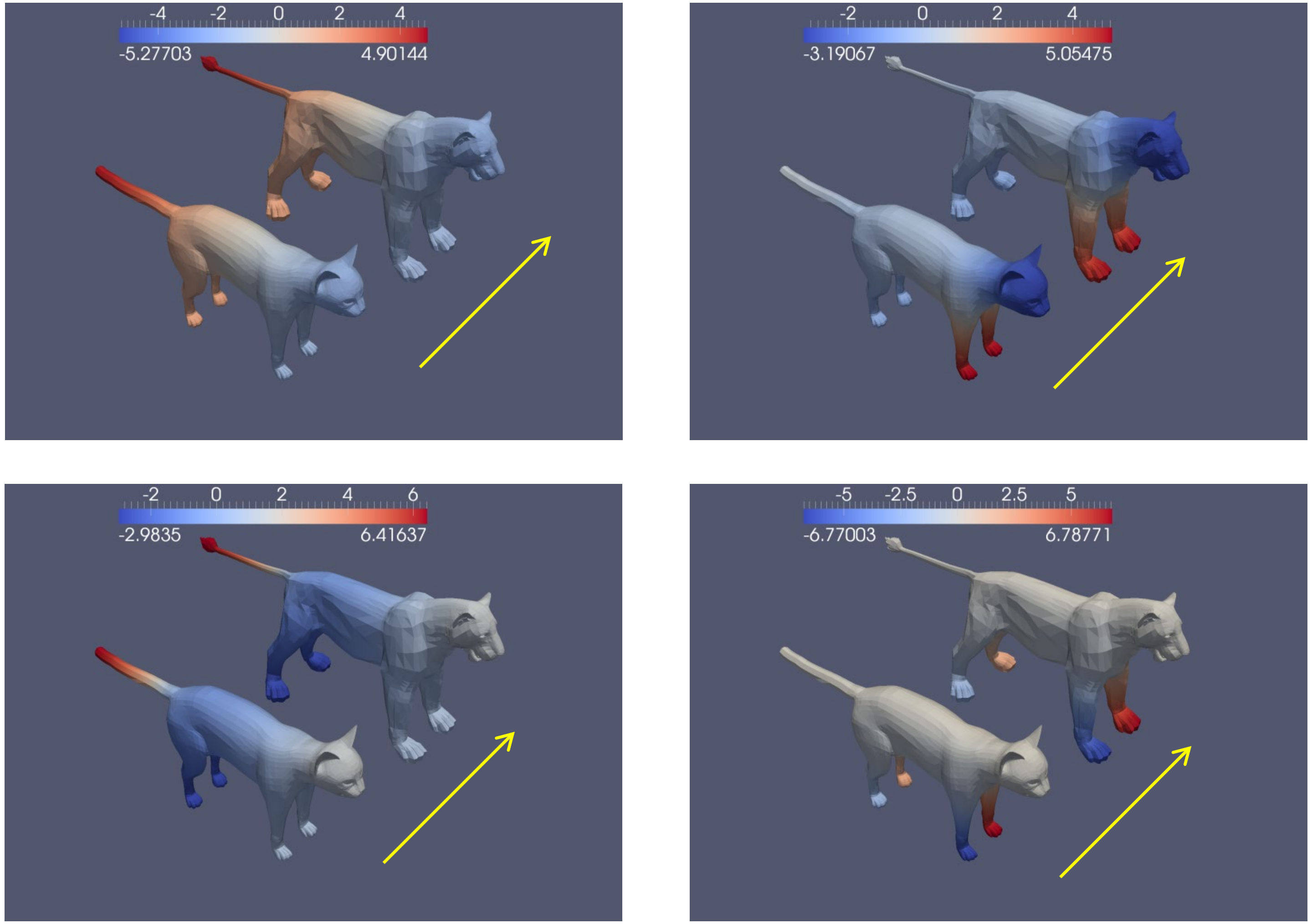


# Starting from a Regular Map ϕ



ϕ: lion → cat

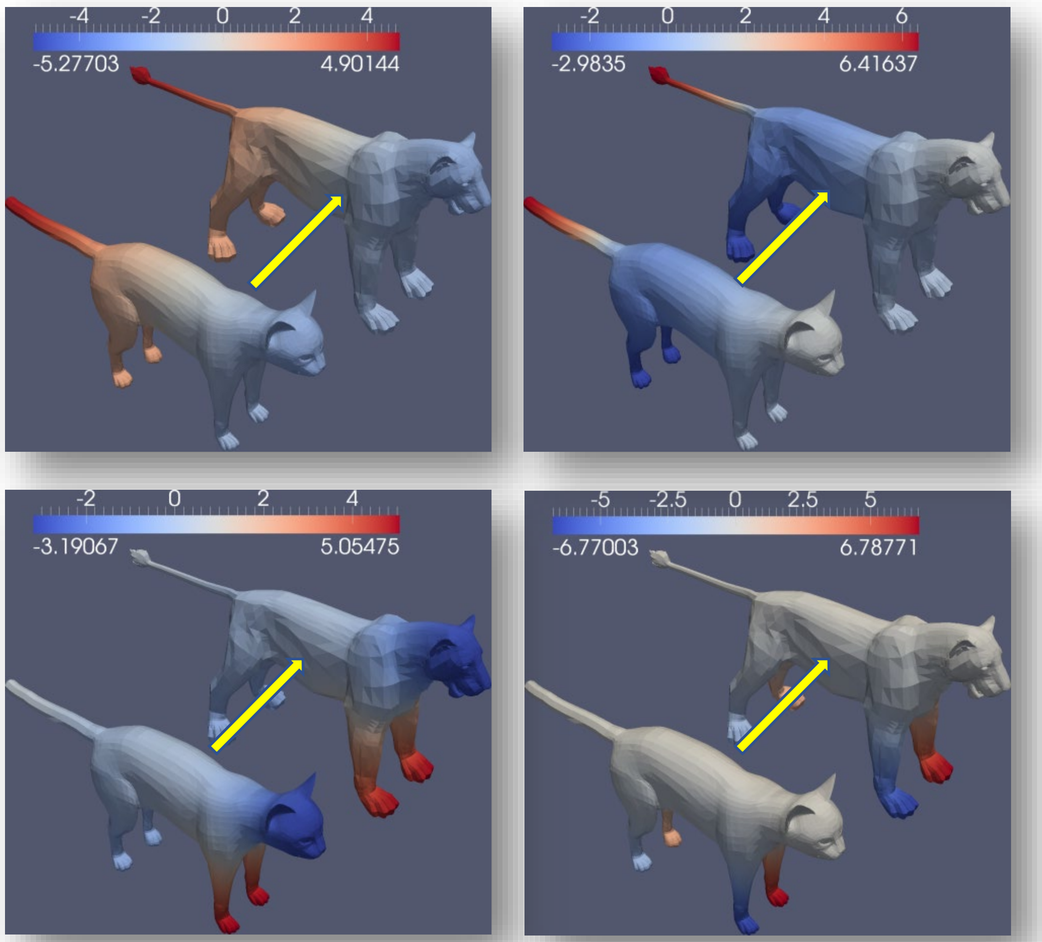
# Attribute Transfer via Pull-Back



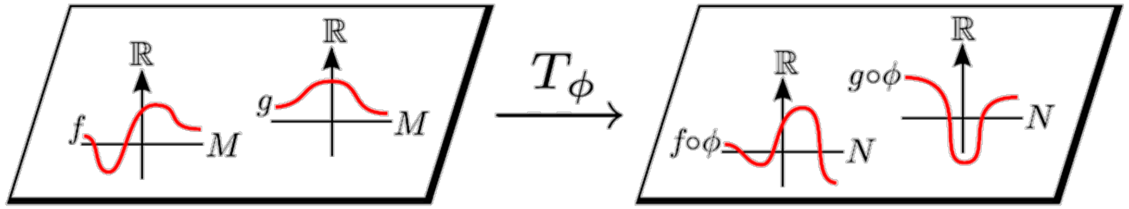

$T_\phi$: cat → lion

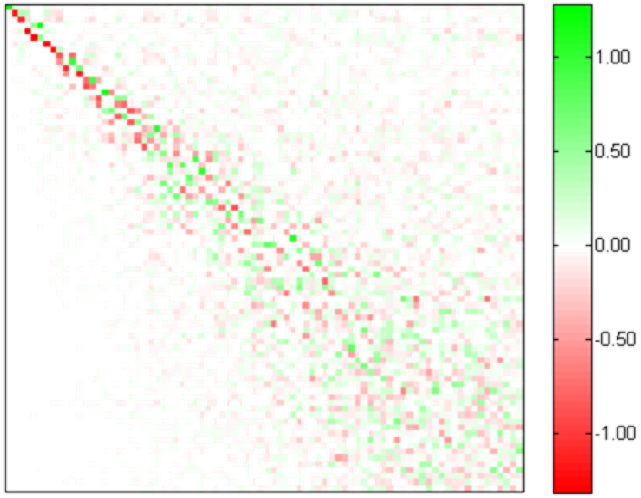
# A Contravariant Functor

from cat to lion




Functions on cat are transferred to lion using $T_\phi$




**$T_\phi$ is a linear operator (matrix)**

$$T_\phi : L^2(cat) \to L^2(lion)$$

# Functional Map

$$\phi : M \to N$$

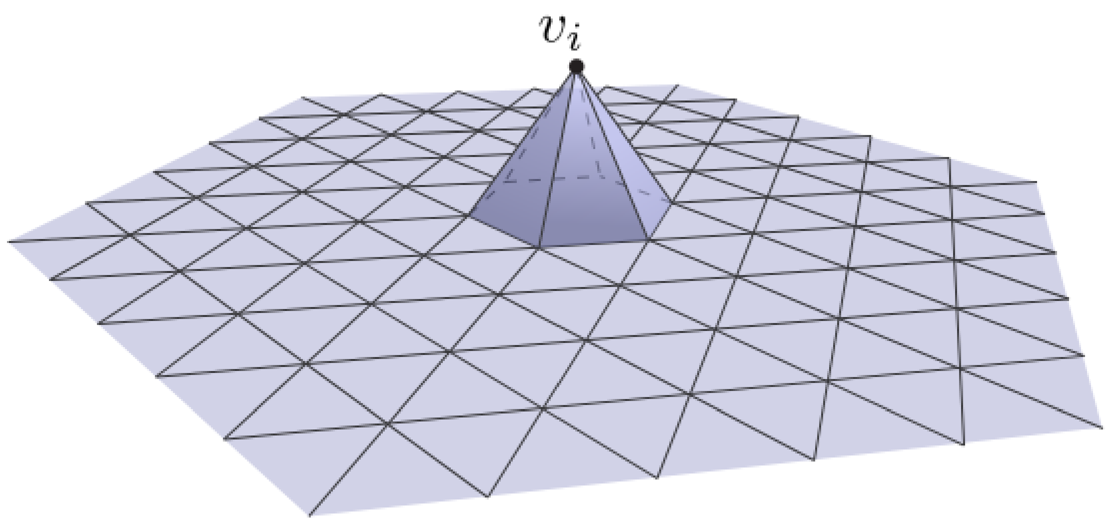
$$T_\phi : L^2(N) \to L^2(M)$$

**Dual of a point-to-point map**

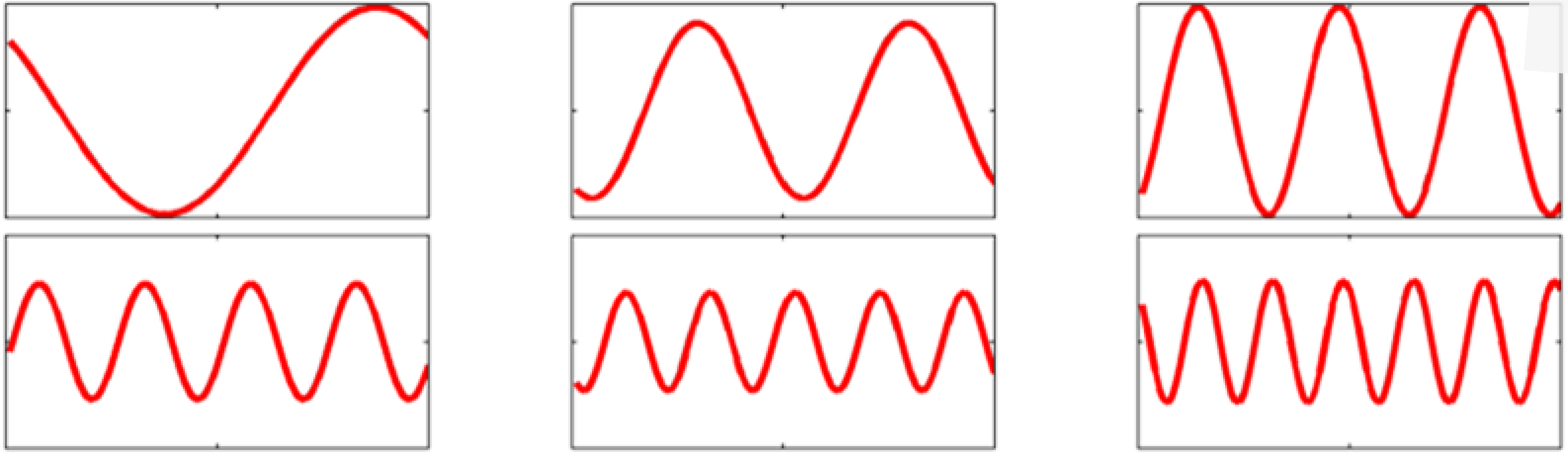
# Encoding Functions: Bases for a Function Space

Point basis
Finite-element basis

Local bases



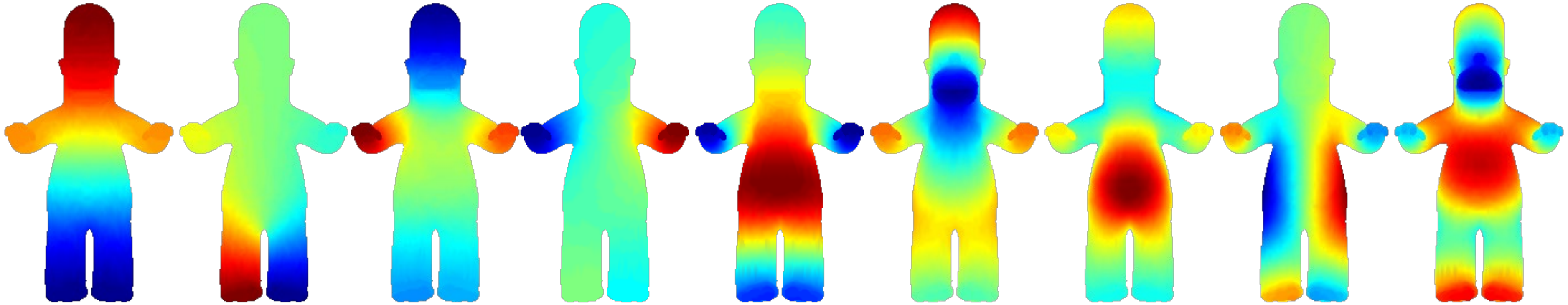
# Hierarchical Bases for a Function Space



**Fourier**

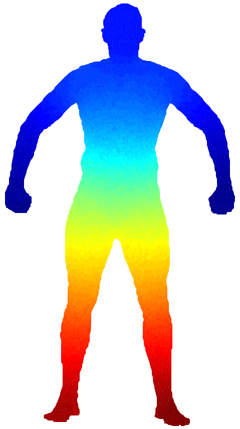
**Laplace-Beltrami**

global support



# Application of Basis

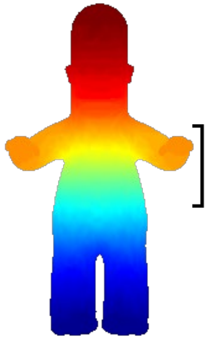
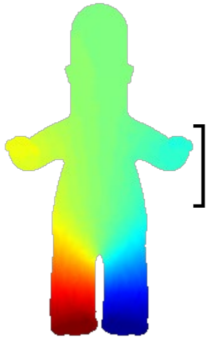
$f(x) = a_1 \cdot$  $+ a_2 \cdot$  $+ a_3 \cdot$  $+ \cdots$

$\phi :$  $\longrightarrow$ 

Homer Simpson

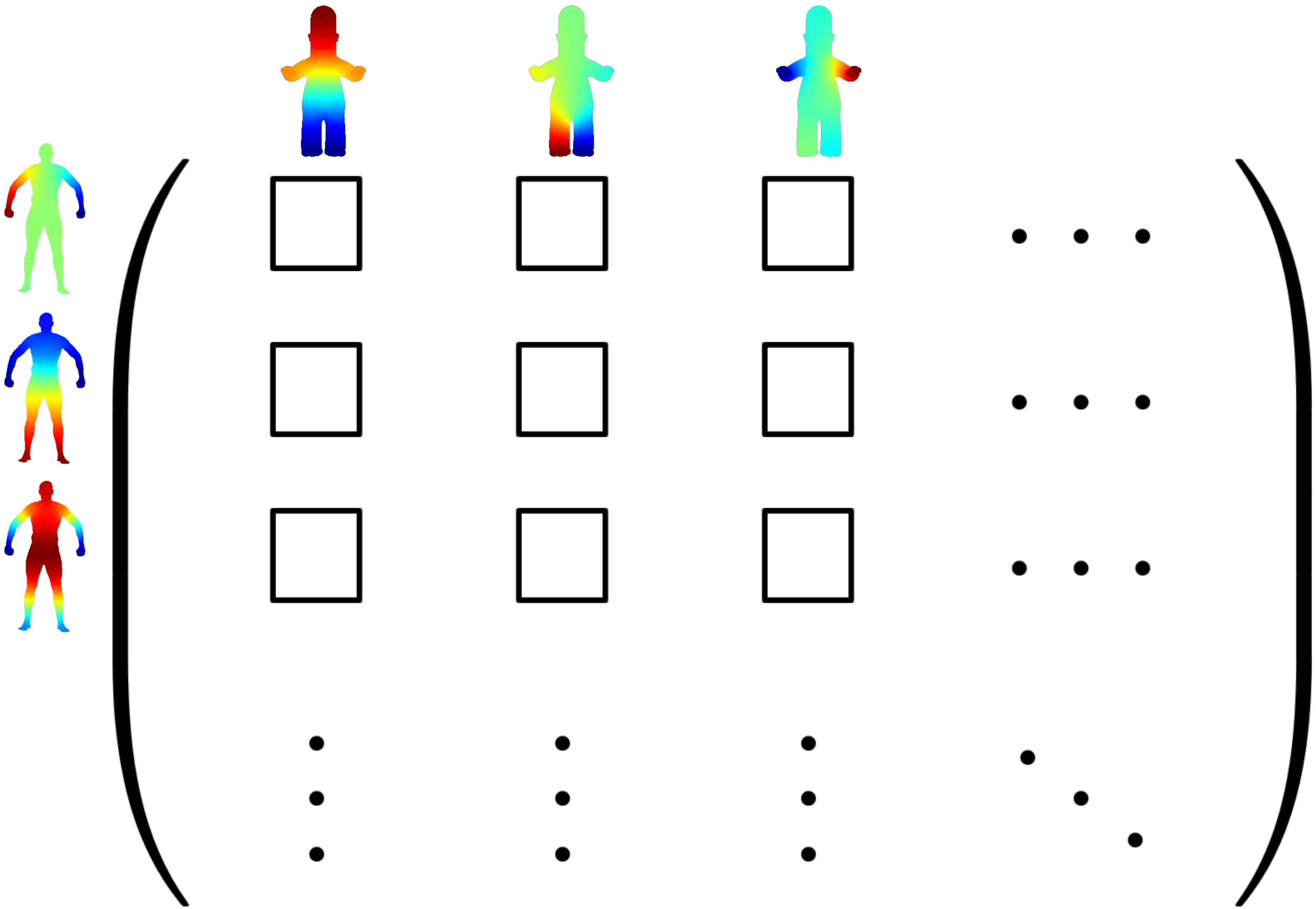
# Application of Basis

$T_\phi[f](x) = T_\phi[a_1 \cdot$  $+a_2 \cdot$  $+a_3 \cdot$  $+ \cdots]$

$= a_1 T_\phi[$  $] + a_2 T_\phi[$  $] + a_3 T_\phi[$  $] + \cdots$

**Enough to know these**

# Functional Map Matrix



# Functional Map Representation

**Definition**

For a fixed choice of basis functions $\{\phi^M\}$ and $\{\phi^N\}$, and a bijection $T : M \to N$, define its **functional representation** as a matrix $C$, s.t. for all $f = \sum_i a_i \phi_i^M$, if $T_F(f) = \sum_i b_i \phi_i^N$ then:

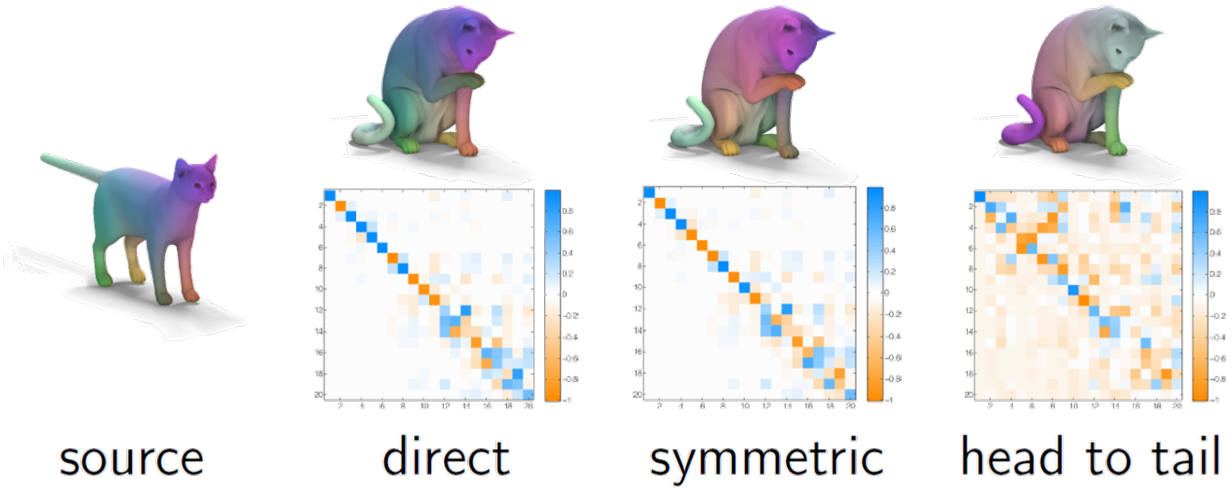
$$\mathbf{b} = C\mathbf{a}$$

If $\{\phi^M\}$ and $\{\phi^N\}$ are both orthonormal w.r.t. some inner product, then

$$C_{ij} = \left\langle T_F(\phi_i^M), \phi_j^N \right\rangle .$$

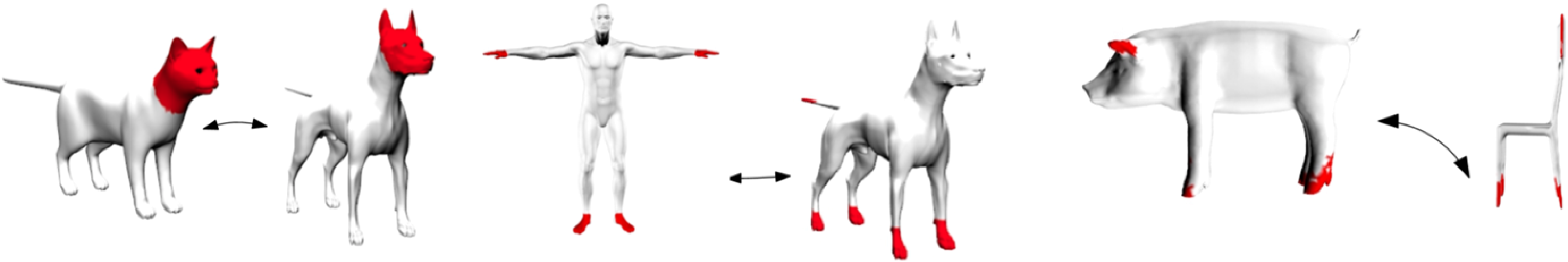
# Maps as Linear Operators

- An ordinary shape map lifts to a linear operator mapping the function spaces
- With a truncated hierarchical basis, compact representations of functional maps are possible as ordinary matrices
- Map composition becomes ordinary matrix multiplication
- Functional maps can express many-to-many associations, generalizing classical 1-1 maps




Using truncated Laplace-Beltrami basis

# FMaps Can Represent Broader Classes of Correspondences



FMaps include point-to-point maps, but are much more general

# Estimating the Mapping Matrix Directly

Suppose we don't know *C*. However, we expect a pair of functions $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$ to correspond. Then, *C* must satisfy:

$$C\mathbf{a} \approx \mathbf{b}$$

where $f = \sum_i \mathbf{a_i} \phi_i^M, \quad g = \sum_i \mathbf{b}_i \phi_i^N$

Functional Correspondences

Given enough $\{\mathbf{a_i}, \mathbf{b}_i\}$ pairs in correspondence, we can recover C through a linear least squares system.

# Function Preservation Constraints

Suppose we don't know *C*. However, we expect a pair of functions $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$ to correspond. Then, *C* must be s.t.

$$C\mathbf{a} \approx \mathbf{b}$$

Function preservation constraint is quite general and includes:

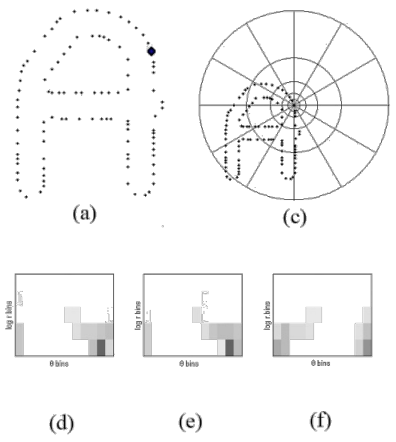
- Descriptor preservation (e.g. Gaussian curvature, spin images, HKS, WKS).
- Landmark correspondences (e.g. distance to the point).
- Part correspondences (e.g. indicator function).
- Texture preservation

**"Probe functions"**

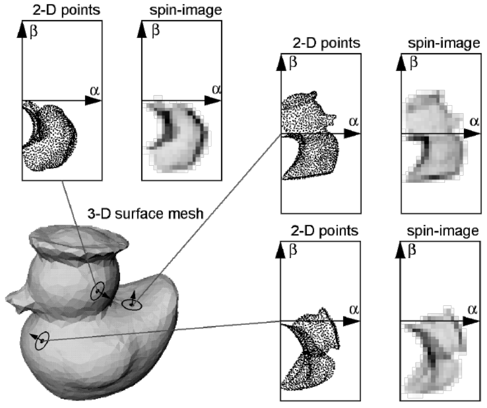
injection of low-level knowledge or supervision

# Plenty of Functions: Descriptors for Points and Parts

- For shapes, there are many descriptors with various types of invariances

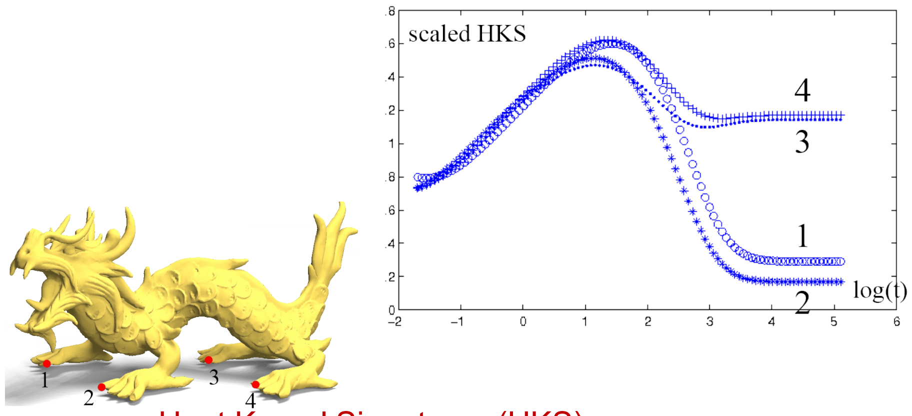



Shape Contexts:
[Belongie et al. ’00, Frome et al. ’04]




Spin Images:
[Johnson, Hebert ’99]

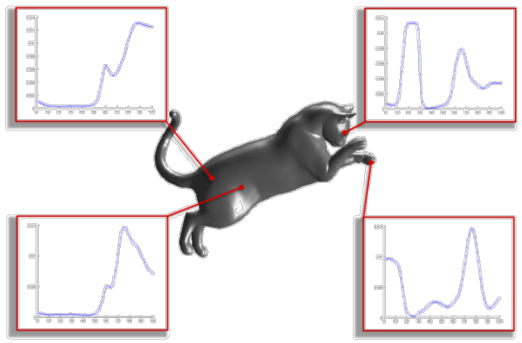
Rigid invariance (extrinsic)

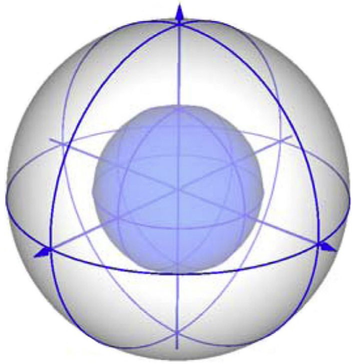



Heat Kernel Signatures (HKS):
[Sun, Ovsjanikov, G. ’08]]

Isometric invariance (intrinsic)

Wave Kernel Signatures (WKS):
[Aubry et. al. ‘11]

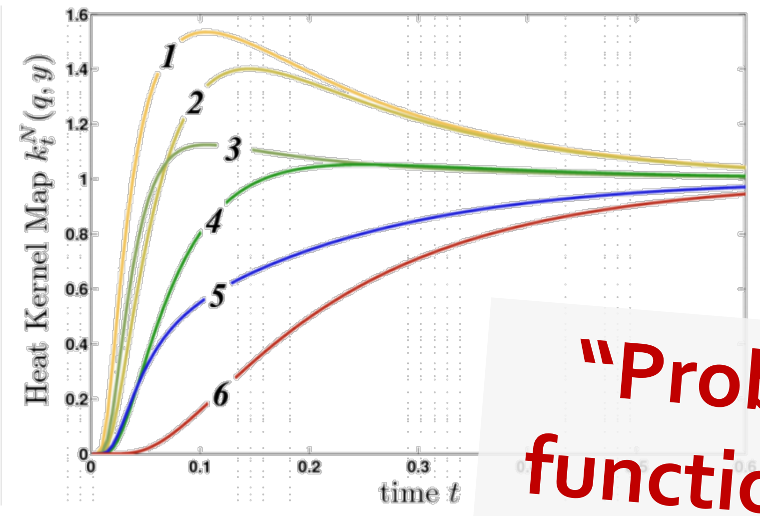




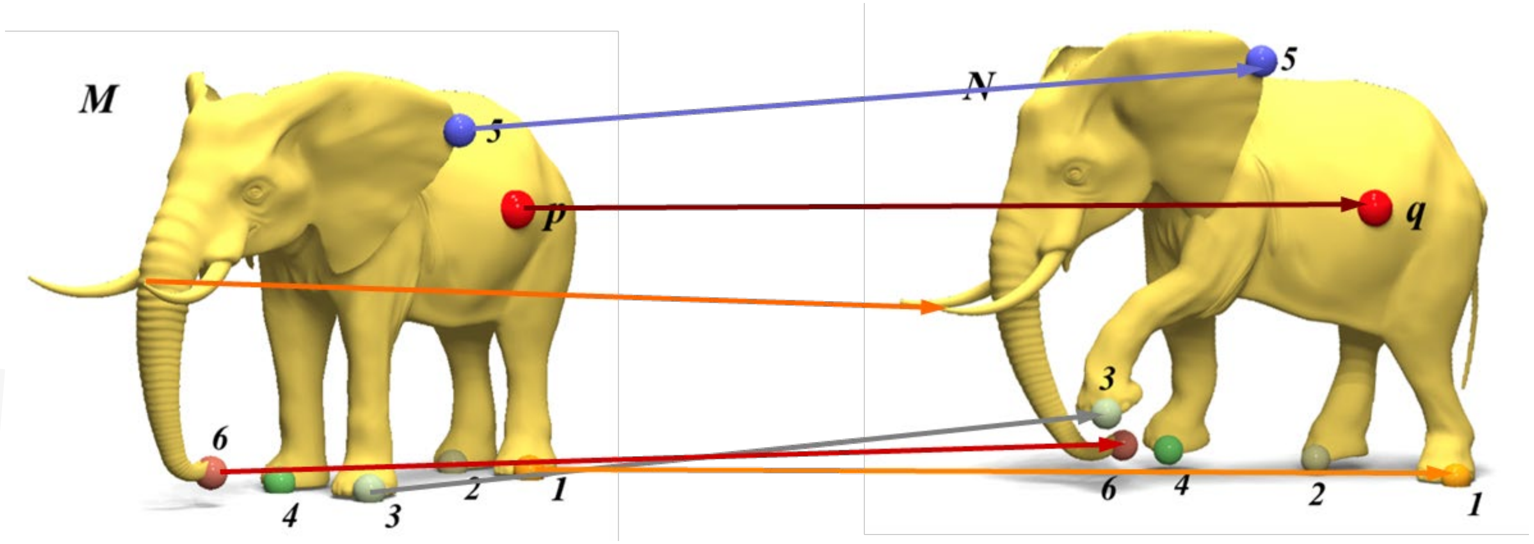
SHOT

# Map Estimation

$$CD_1 = D_2 \implies C = D_2 D_1^{-1}$$




"Probe function"




**Map from a linear solve**

# Commutativity Symmetry Regularization

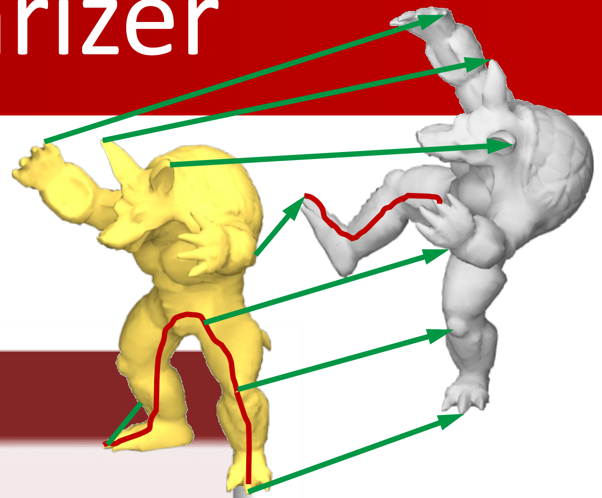
In addition, we can phrase an operator commutativity constraint: given two operators $S_1 : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(M, \mathbb{R})$ and $S_2 : \mathcal{F}(N, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$

$$\begin{array}{ccc} \mathcal{F}(M,\mathbb{R}) & \xrightarrow{C} & \mathcal{F}(N,\mathbb{R}) \\ {\scriptstyle S_1}\downarrow & & \downarrow{\scriptstyle S_2} \\ \mathcal{F}(M,\mathbb{R}) & \xrightarrow{C} & \mathcal{F}(N,\mathbb{R}) \end{array}$$

Thus: $CS_1 = S_2 C$ or $\|CS_1 - S_2 C\|$ should be minimized

Note: this is a linear constraint on $C$. $S_1$ and $S_2$ could be symmetry operators or e.g. Laplace-Beltrami or heat operators.

# Isometry (Length Preservation) Regularizer



**Lemma 1:**

The mapping is *isometric*, if and only if the functional map matrix commutes with the Laplacian:

$$C\Delta_1 = \Delta_2 C$$

**Differentiate and then transport**

**Transport and then differentiate**

$\Delta_1$ Laplacian on Shape 1
$\Delta_2$ Laplacian on Shape 2

# Conformal (Angle Preservation) Regularization

**Lemma 3:**

If the mapping is *conformal* if and only if:

$$C^T \Delta_1 C = \Delta_2$$

Using these regularizations, we get a very efficient shape matching method.

# Volume Preservation Regularizer

**Lemma 2:**

The mapping is *locally volume preserving*, if and only if the functional map matrix is *orthonormal*:

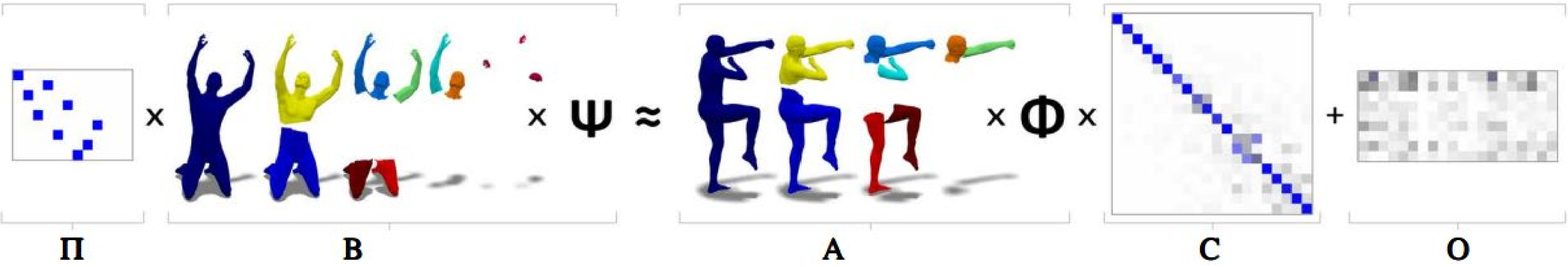
$$C^T C = \mathrm{I}$$

Rotations/reflections in functions space

# Sparcity in a Localized Basis

$$\min \|C\|_{2,1}$$

Sum of Euclidean norms of cols




**Sparse Modeling of Intrinsic Correspondences (Pokrass, Bronstein[2], Sprechmann, Sapiro)**

# Basic FMaps Pipeline

Given a pair of shapes: $\mathcal{M}, \mathcal{N}$

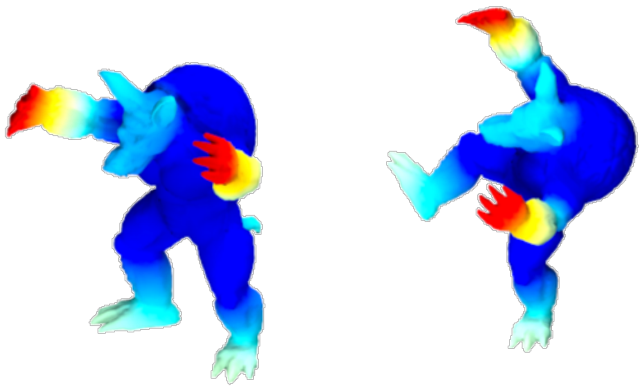
1. Compute the first *k* (~80-100) eigenfunctions of the Laplace-Beltrami operator. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

2. Compute descriptor functions (e.g., Heat Kernel Signatures) on $\mathcal{M}, \mathcal{N}$. Express them in $\mathbf{A}, \mathbf{B}$, as columns of $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

3. Solve $C_{\text{opt}} = \arg\min_{C} \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2 + \ \ldots$

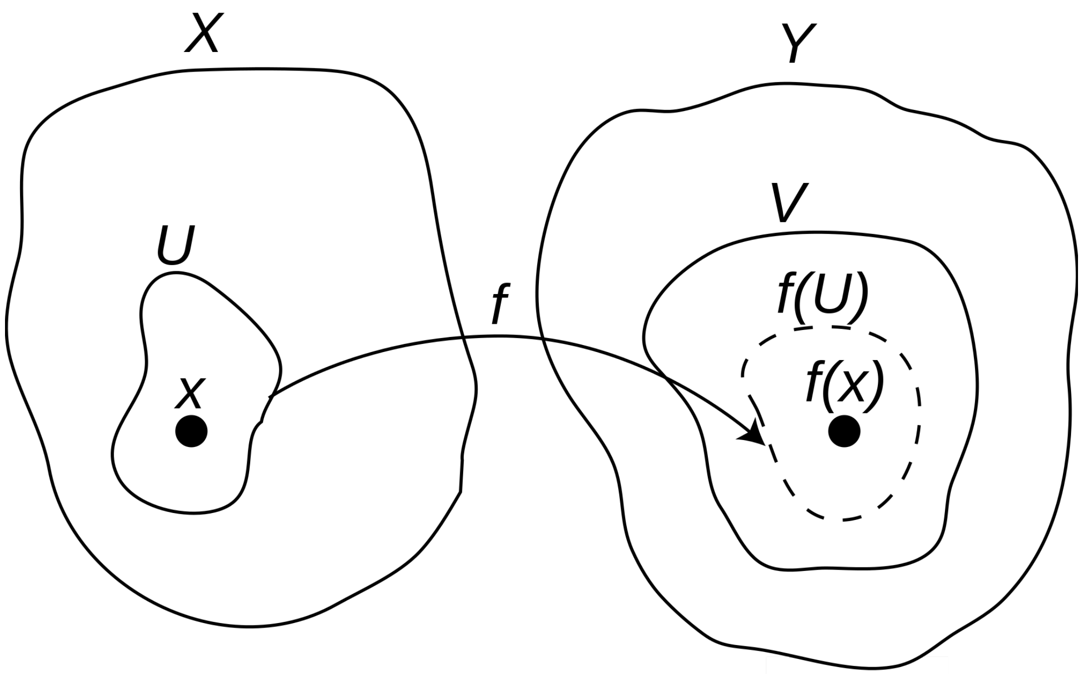
   $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$ : diagonal matrices of eigenvalues of LB operator

4. Convert the functional map $C_{\text{opt}}$ to a point to point map *T*.



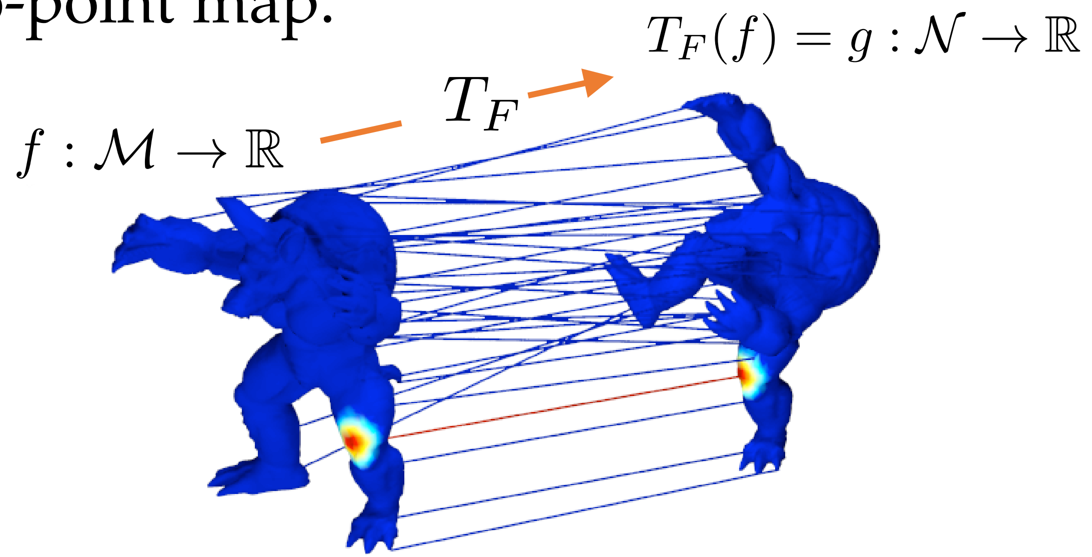
# Side Comment: Map Continuity

- Not explicitly enforced
- Implicit in the choice of basis



# Conversion to Point-to-Point

Given a functional map $C$, we would like to convert to to a point-to-point map.

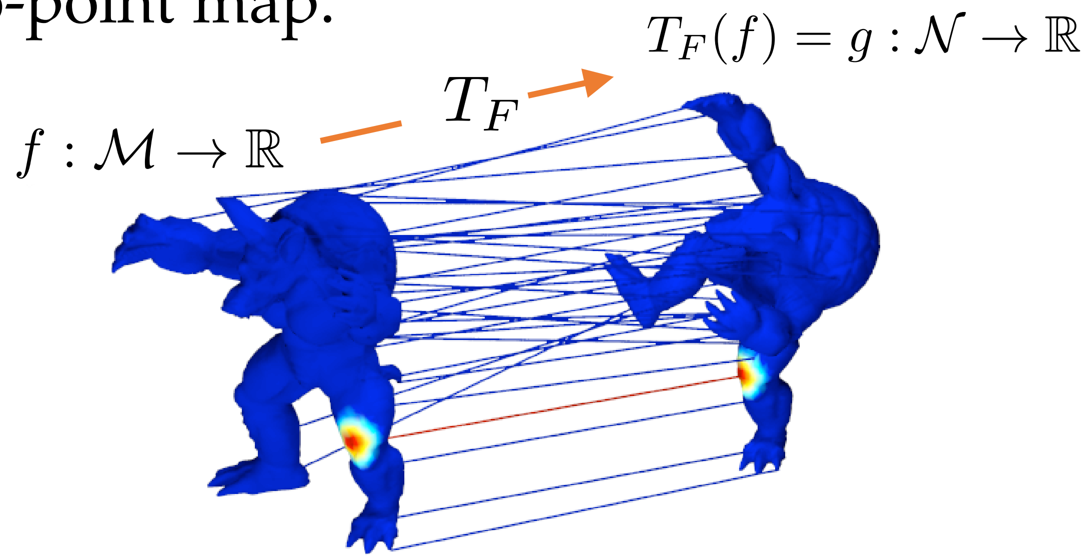



Option 1: declare $T(x) = \arg\max_y \Phi_{\mathcal{N}} C \delta_x$

Problems: high computational complexity $O(n_{\mathcal{M}} n_{\mathcal{N}})$, low accuracy.

# Conversion to Point-to-Point

Given a functional map $C$, we would like to convert to to a point-to-point map.




Option 2: declare $T(x) = \arg\min_y \|\delta_y - C\delta_x\|_2$

Advantages: computational complexity $O(n_{\mathcal{M}} \log n_{\mathcal{N}})$, higher accuracy (e.g., works with the identity map).

# From Functional to Point-to-Point Maps

- Can try transporting delta functions individually -- expensive

**Primal**

$$p \mapsto q$$

**Dual**

$$p \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} q$$

$$\delta_x = (\phi_1^M(x), \phi_2^M(x), \phi_3^M(x), \ldots)$$

# From Functional to Point-to-Point Maps

$$T(x) = \arg\min_y \|\delta_y - C\delta_x\|_2$$

$$C\Phi_M^\top \leftrightarrow \Phi_N$$

Image of each point on surface M

Each point on surface N in LB basis

So transport, and then use nearest neighbor search

# Incorporating Orthonormality

In many practical situations we would expect a volume-preserving map, which implies:

$$C^T C = Id$$

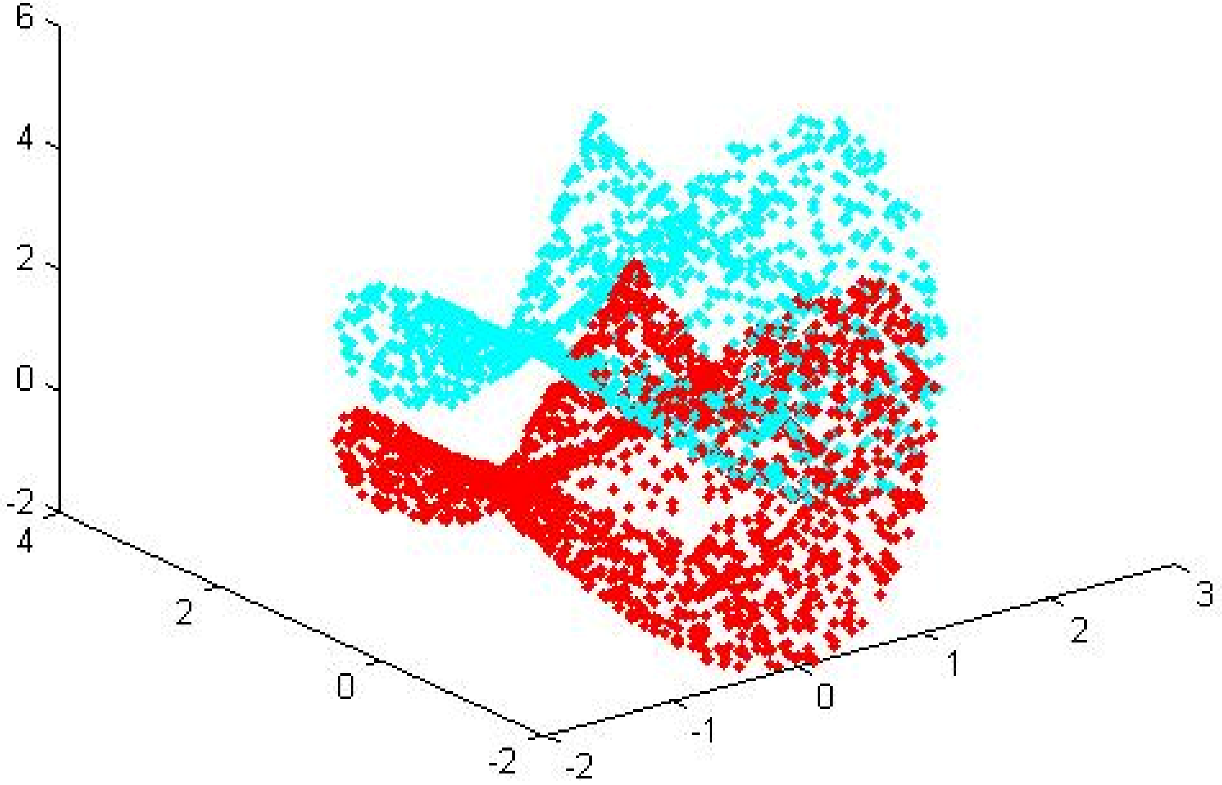
Option: use post-processing to enforce this constraint.

Iterate:

1. Compute the point-to-point map $T$.
2. Solve for the functional map: $\underset{C,\text{ s.t. } C^T C = Id}{\arg\min} \sum_{x \in \mathcal{M}} \| C\delta_x - \delta_{T(x)} \|_2^2$

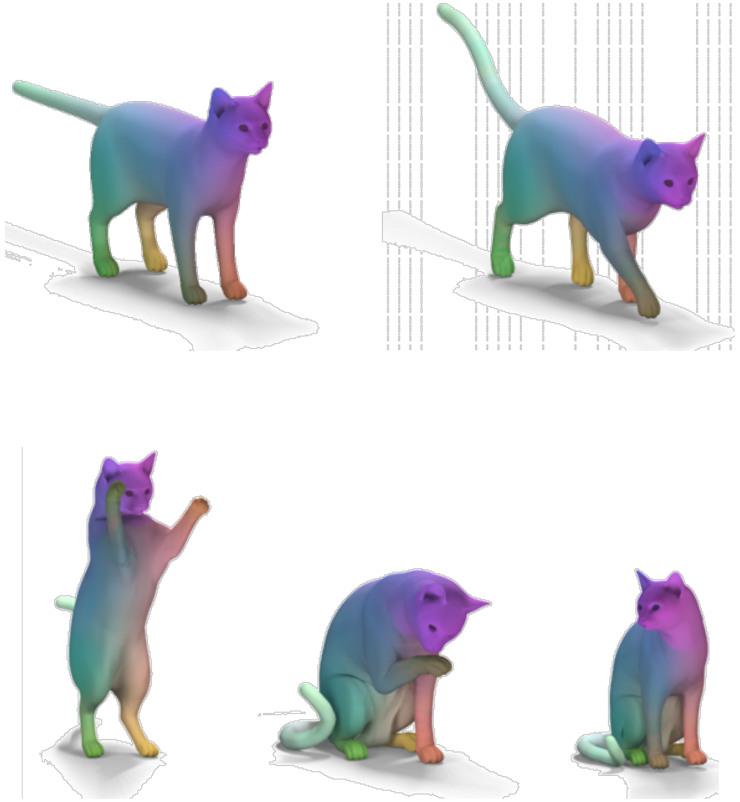
Exactly the same objective as ICP, but in higher dimension. Can use the same method!

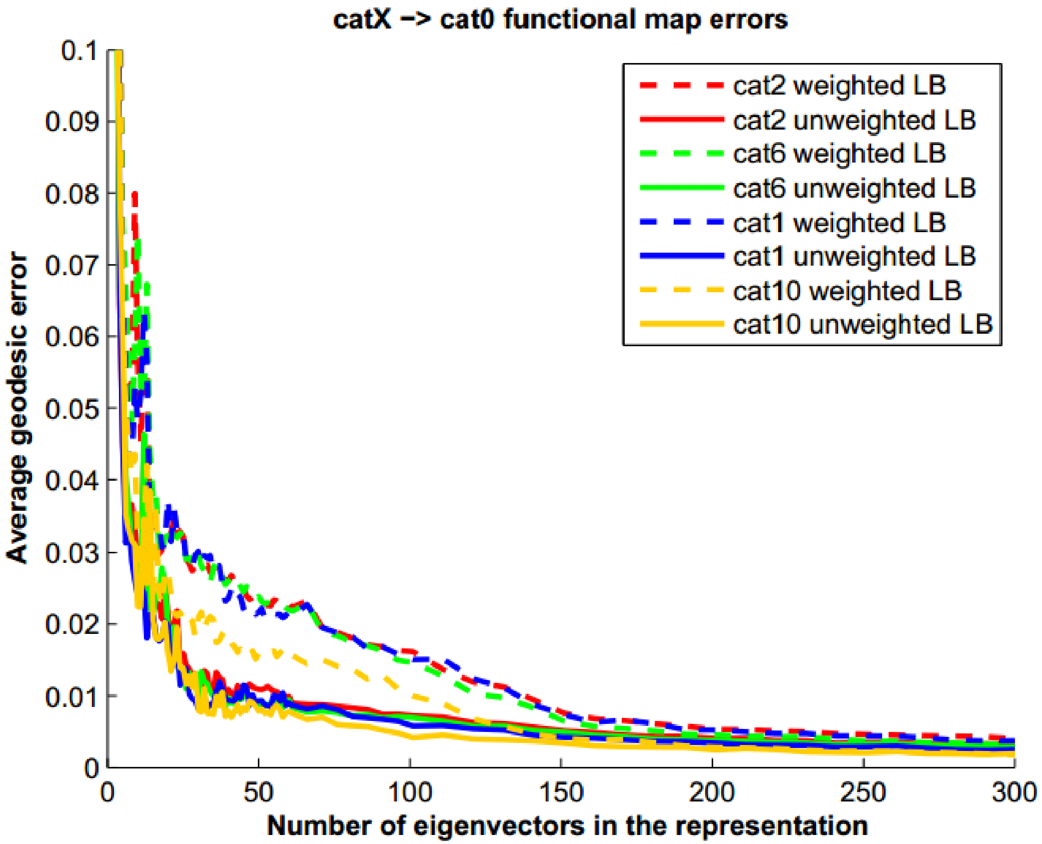
# From Functional to Point-to-Point Maps




**ICP in Function Space!**

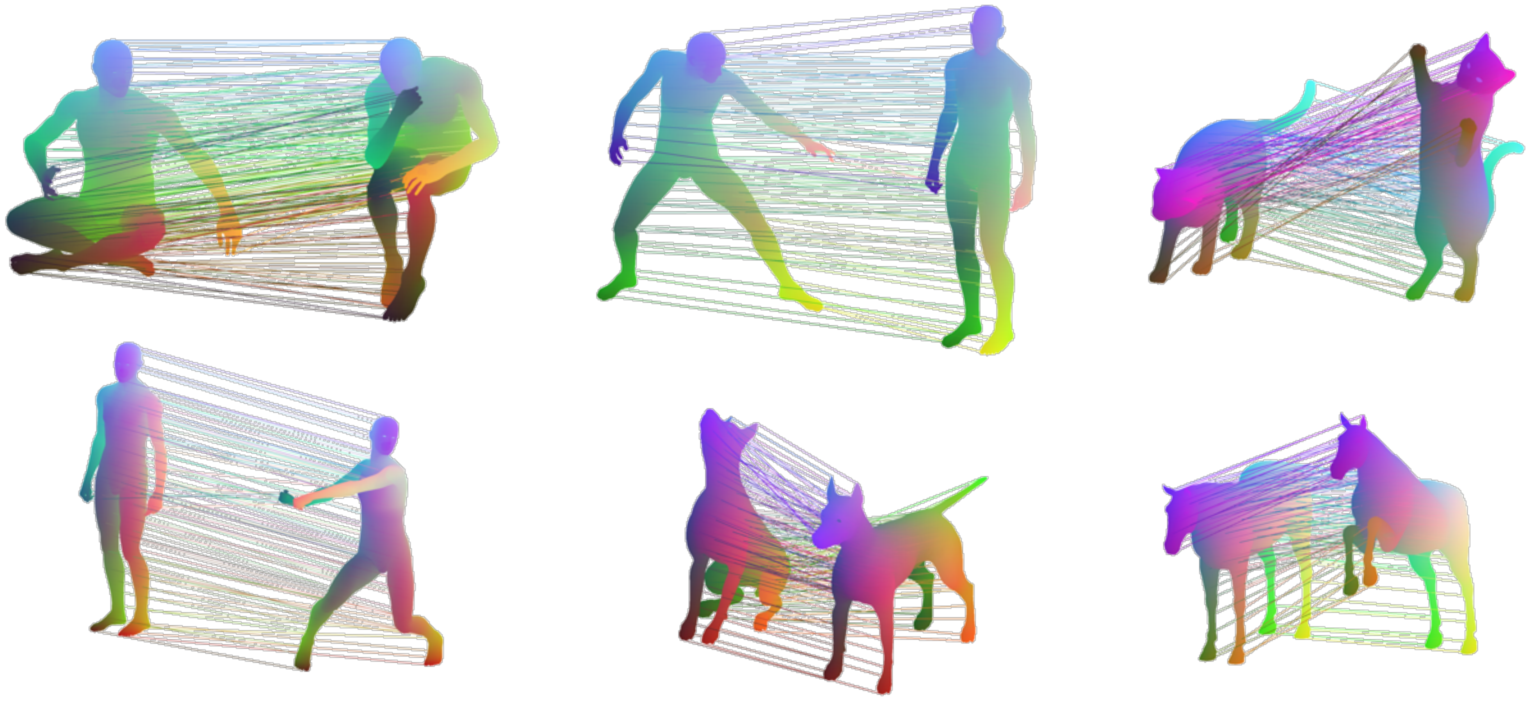
# Ground Truth Comparison





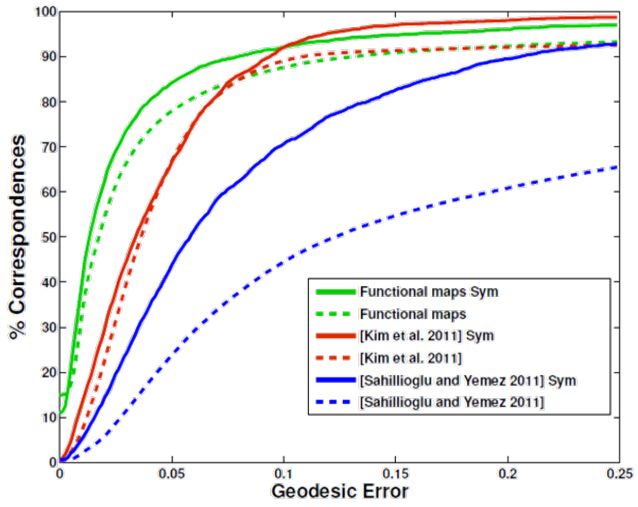
# Results

A very simple method that puts together many constraints and uses 100 basis functions gives reasonable results:

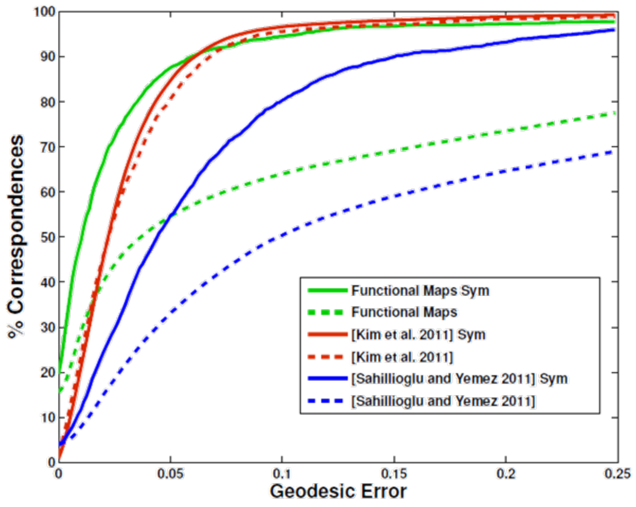


# Map Estimation Quality

A very simple method that puts together a modest set of constraints and uses 100 basis functions outperforms state-of-the-art:




SCAPE




TOSCA

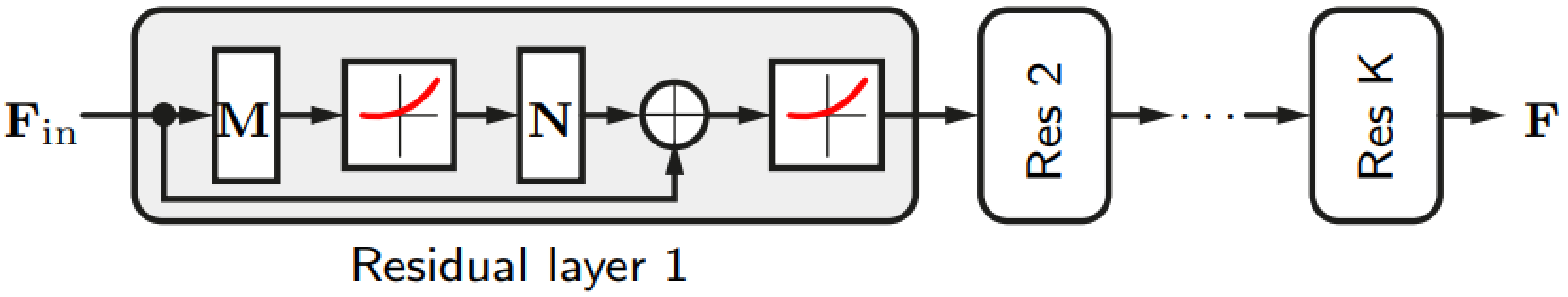
Roughly 10 probe functions + 1 part correspondence

# Much Follow-On Work on FMaps

# Deep Functional Maps

# Deep Functional Maps

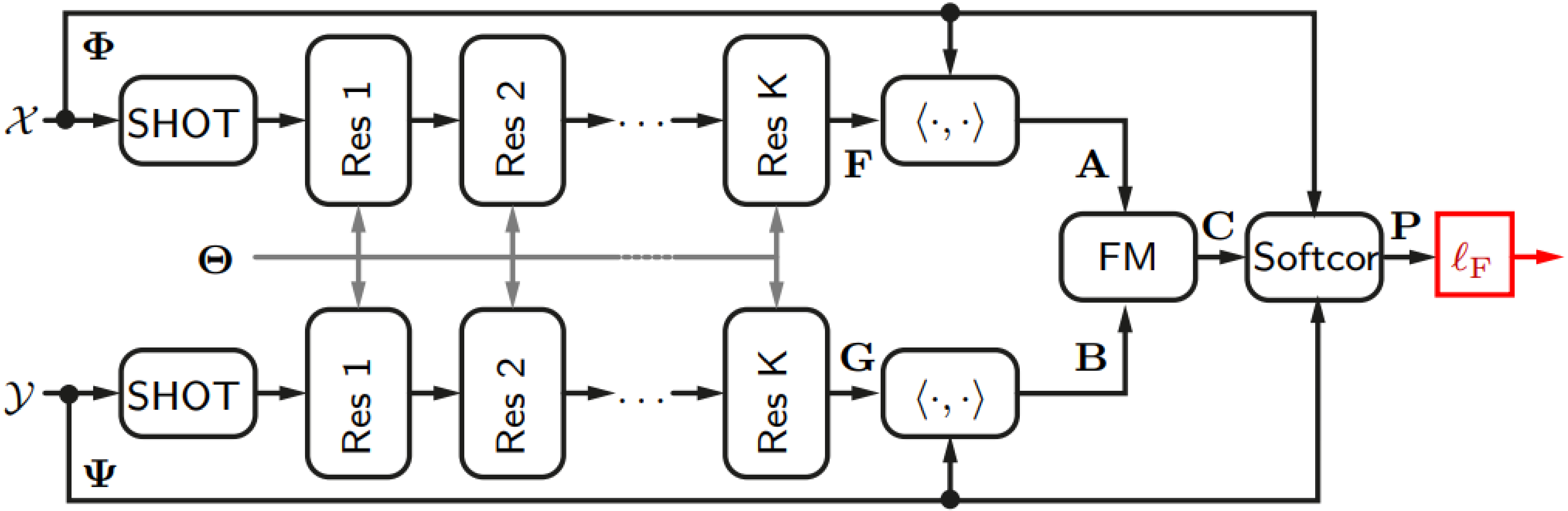
- Main idea: Learn pointwise descriptors resulting in *best functional maps*
- Residual network (ResNet) for descriptor learning:




- **Input:** pointwise descriptor $\mathbf{F}_{\text{in}}$ (e.g., 352-dimensional SHOT)
- $N$ fully-connected residual layers with ELU activation parameterized by $\mathbf{\Theta} = \{\mathbf{M}_1, \mathbf{N}_1, \ldots, \mathbf{M}_K, \mathbf{N}_K\}$
- **Output:** transformed pointwise descriptor $\mathbf{F}$

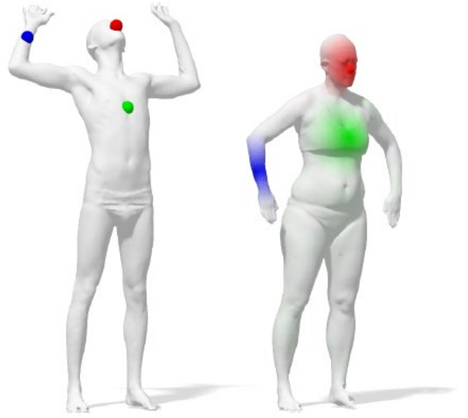
Litany, Remez, Rodola, Bronstein (ICCV’17)

# FMNet




- **Functional map layer:** $\mathbf{C} = \arg\min \|\mathbf{CA} - \mathbf{B}\|_F^2$
- **Soft correspondence layer:** $\mathbf{P} = |\mathbf{\Psi C \Phi}^\top|_{\|\cdot\|}$
- Functional map loss:

$$\ell_F = \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} P(x,y)\, d_{\mathcal{Y}}(y, \pi^*(x)) = \|\mathbf{P} \odot \mathbf{D}_{\mathcal{Y}}\|_F$$



# Making Functional Maps More Point-to-Point

# Making Functional Maps Point-to-Point

**Question:**

When does a linear functional mapping correspond to a pull-back by a point-to-point map?

**(Known) Theoretical result:**

A functional map is point-to-point iff it preserves pointwise products of functions:

$$C(fh) = C(f)C(h) \ \forall \ f, h \quad (fh)(x) = f(x)h(x)$$

J. von Neumann, *Zur operatoren methode in der klassichen Mechanik*, Ann. of Math.(2) 33 (1932)

# Making Functional Maps Point-to-Point

**Approach**

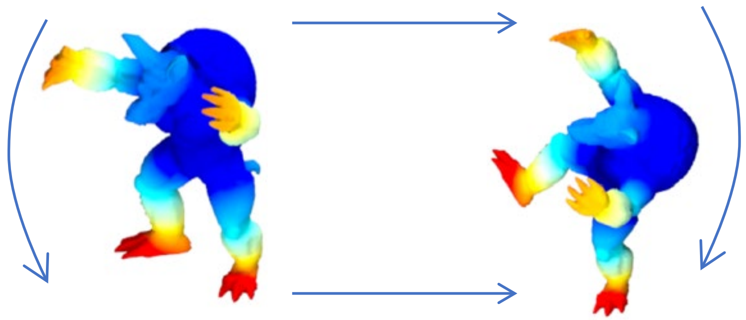
Represent descriptor functions via their action on functions through multiplication:

Functions as operators

$$S_{f_k} = \Phi_{\mathcal{M}}^{+} Diag(f_k) \Phi_{\mathcal{M}}$$
$$S_{g_k} = \Phi_{\mathcal{N}}^{+} Diag(g_k) \Phi_{\mathcal{N}}$$

$C$

$S_{f_k}$



$S_{g_k}$

$$C S_{f_k} = S_{g_k} C$$

*Informative Descriptor Preservation via Commutativity for Shape Matching*, Nogneng, O., Eurographics 2017

# Extended Basic Pipeline

Given a pair of shapes $\mathcal{M}, \mathcal{N}$ :

1. Compute the multi-scale bases for functions on the two shapes. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

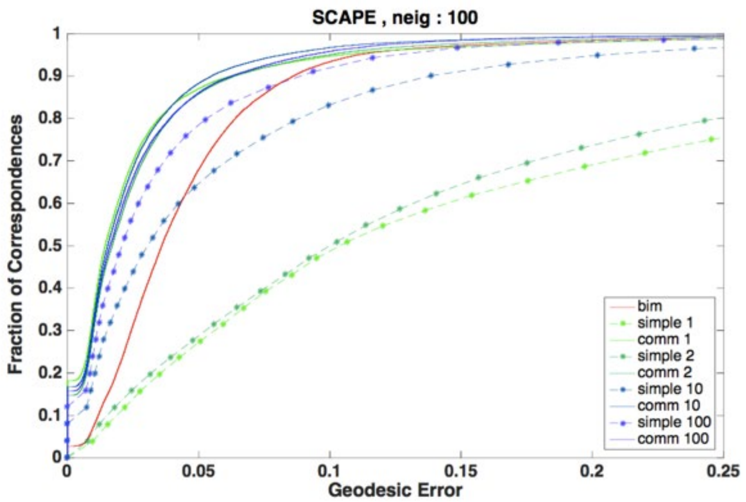
2. Compute descriptor functions (e.g., Gauss curvature) on $\mathcal{M}, \mathcal{N}$ . Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$ as columns of : $\mathbf{A}, \mathbf{B}$

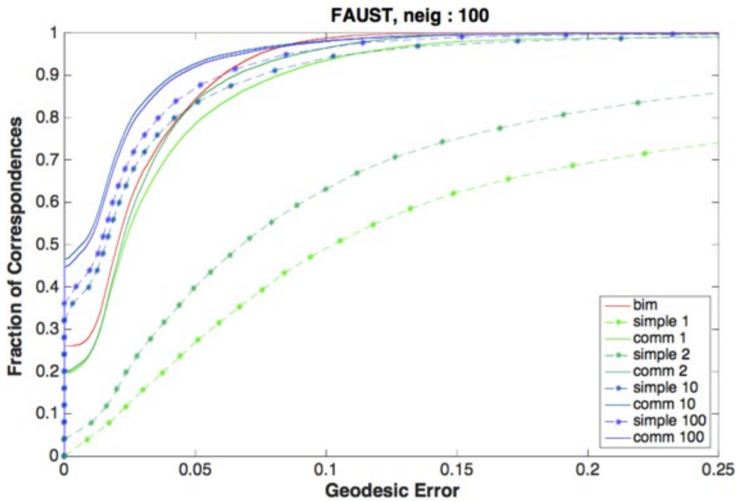
3. Solve $$C_{\text{opt}} = \arg\min_{C} \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2 + \sum_k \|CS_{f_k} - S_{g_k}C\|^2$$

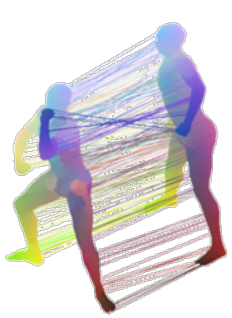
4. Convert the functional map $C_{\text{opt}}$ to a point to point map *T*.

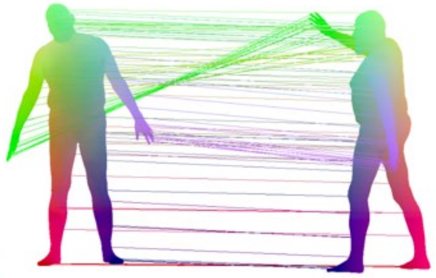
*Informative Descriptor Preservation via Commutativity for Shape Matching*, Nogneng, O., Eurographics 2017

# Results with Extended Basic Pipeline





before after before after

Incorporating multiplicative operators improves results significantly.

*Informative Descriptor Preservation via Commutativity for Shape Matching*, Nogneng, O., Eurographics 2017

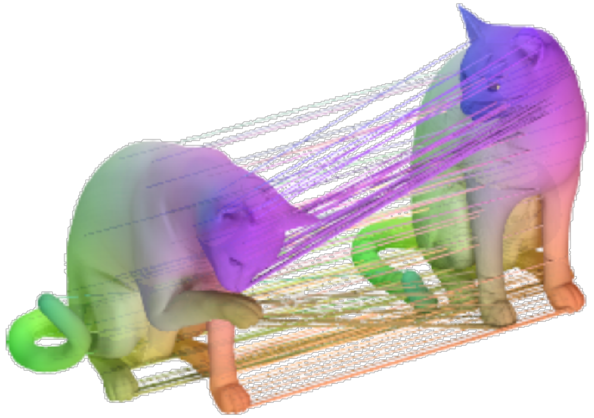
# Segmentation Transport

# Application: Segmentation Transfer

# Map Visualization

# Map Visualization

Even given a map $T : M \to N$, it is often hard to visualize it.



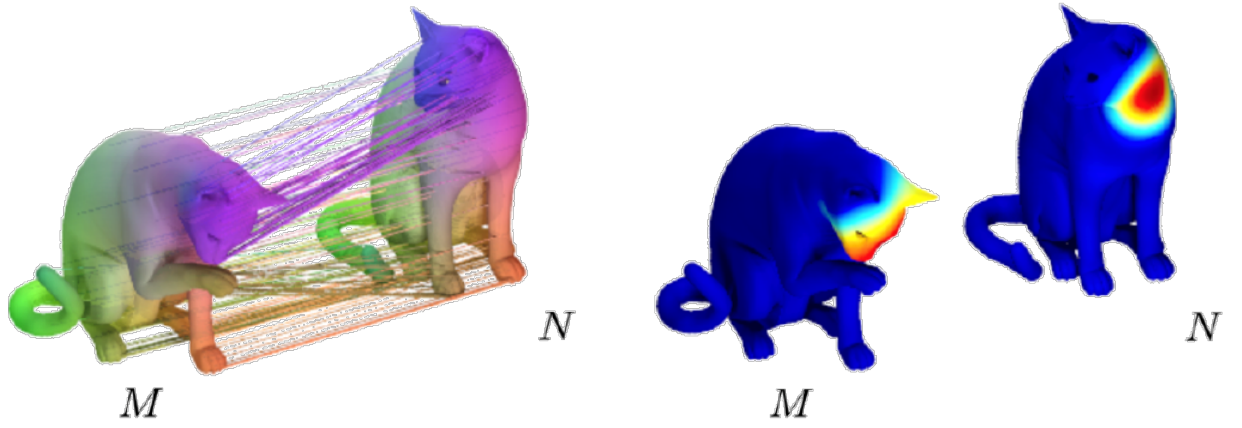
Common visualizations:

- Connecting (some) points by lines
- Plotting a function $f$ on $N$ and $f \circ T$ on $M$.

Question: how to pick a “good” function $f$.

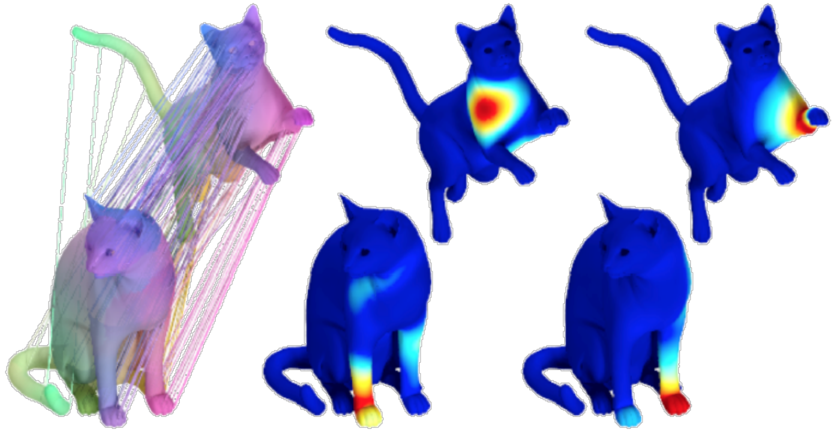
# Map Visualization

Singular vectors of the functional representation $C$ of $T$ identify most distorted regions in a multi-scale way.



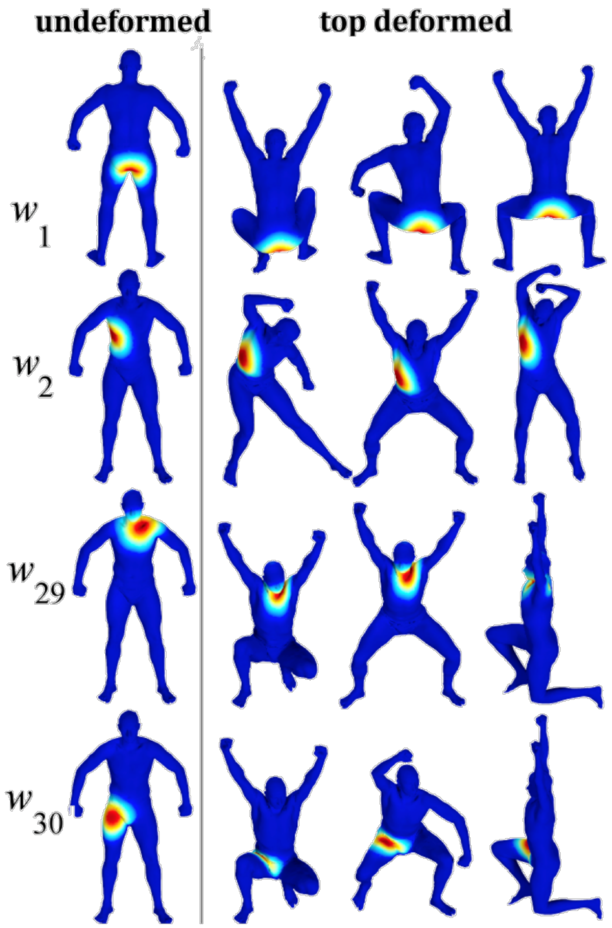
# Map Visualization

Can show that singular vectors of the functional representation $C$ of $T$ identify most distorted regions in a multi-scale way.



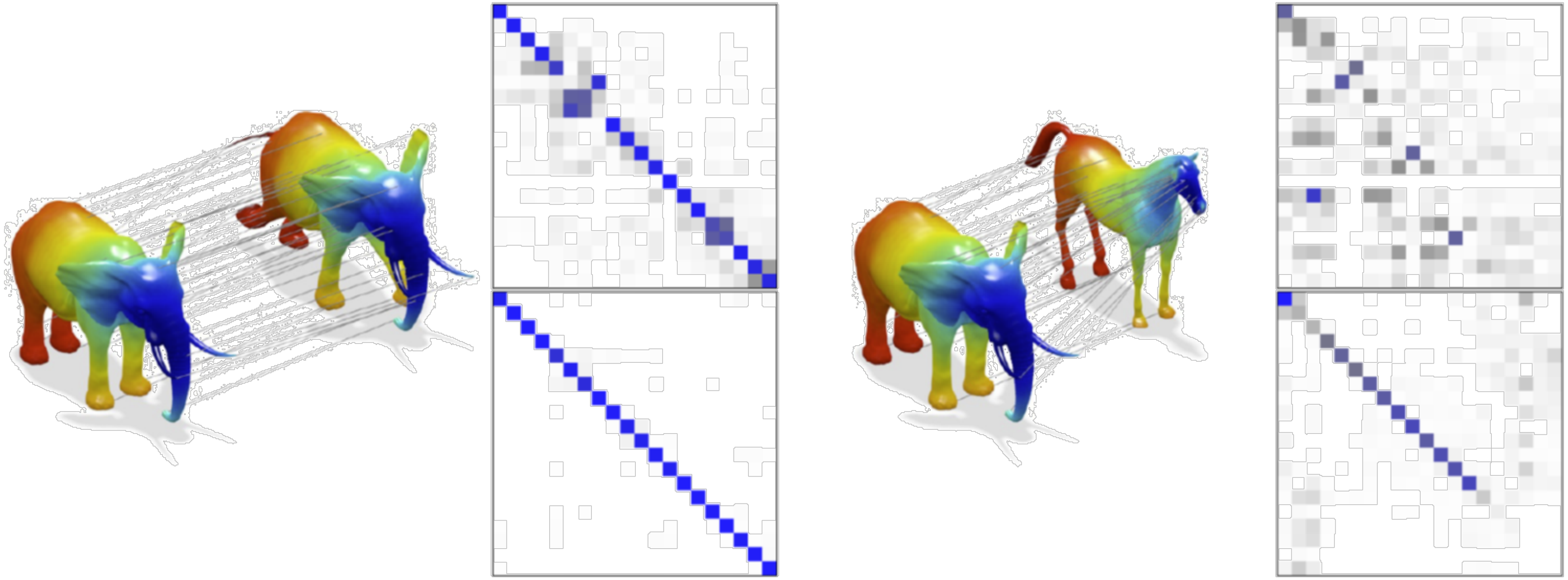
# Multiple Shapes

With same method, can visualize maps to *multiple* shapes.



# Diagonalizing the Map

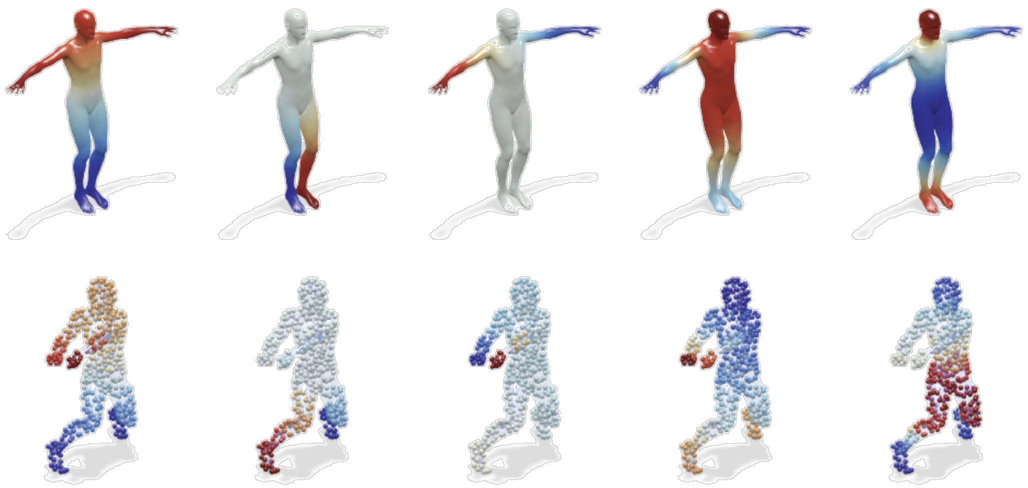
# Coupled Bases



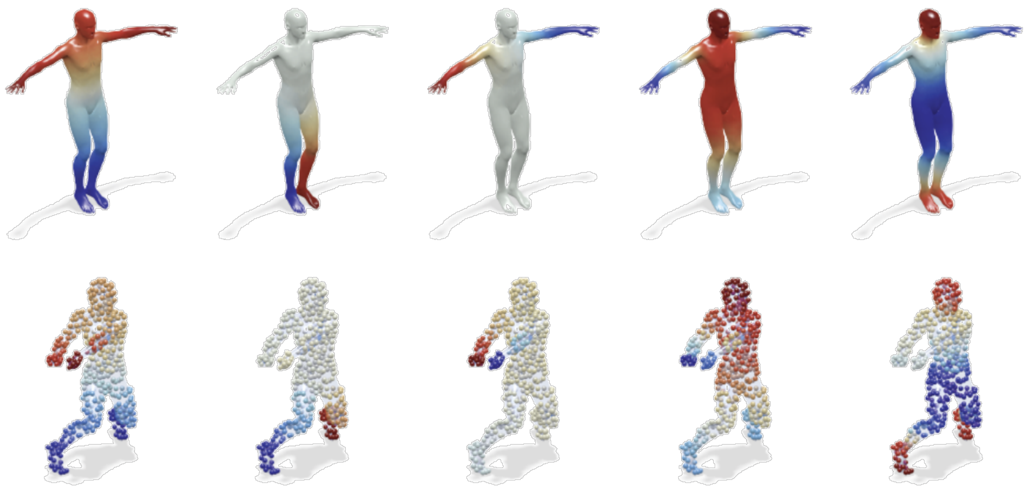
$$\min_{\mathbf{P},\mathbf{Q}} \quad \mathrm{trace}(\mathbf{P}^\top \mathbf{\Lambda}_X \mathbf{P}) + \mathrm{trace}(\mathbf{Q}^\top \mathbf{\Lambda}_Y \mathbf{Q}) + \mu \|\mathbf{P}^\top \mathbf{A} - \mathbf{Q}^\top \mathbf{B}\|$$

$$\text{s.t.} \quad \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$$

# Coupled Bases



Laplacian eigenbases

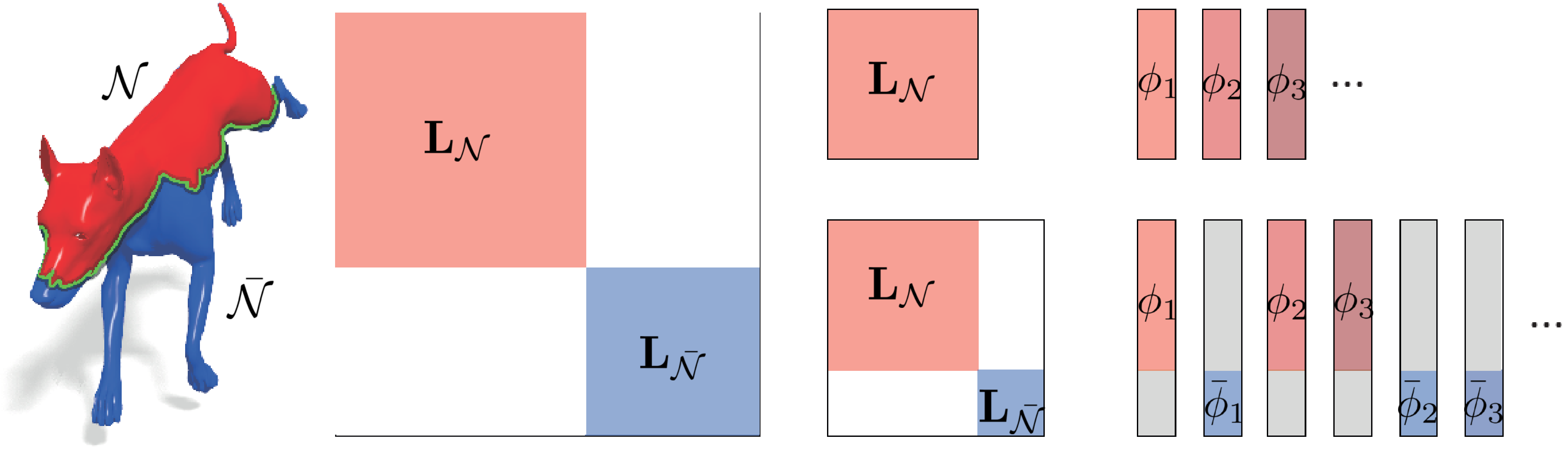


Coupled bases

# Partial Functional Maps

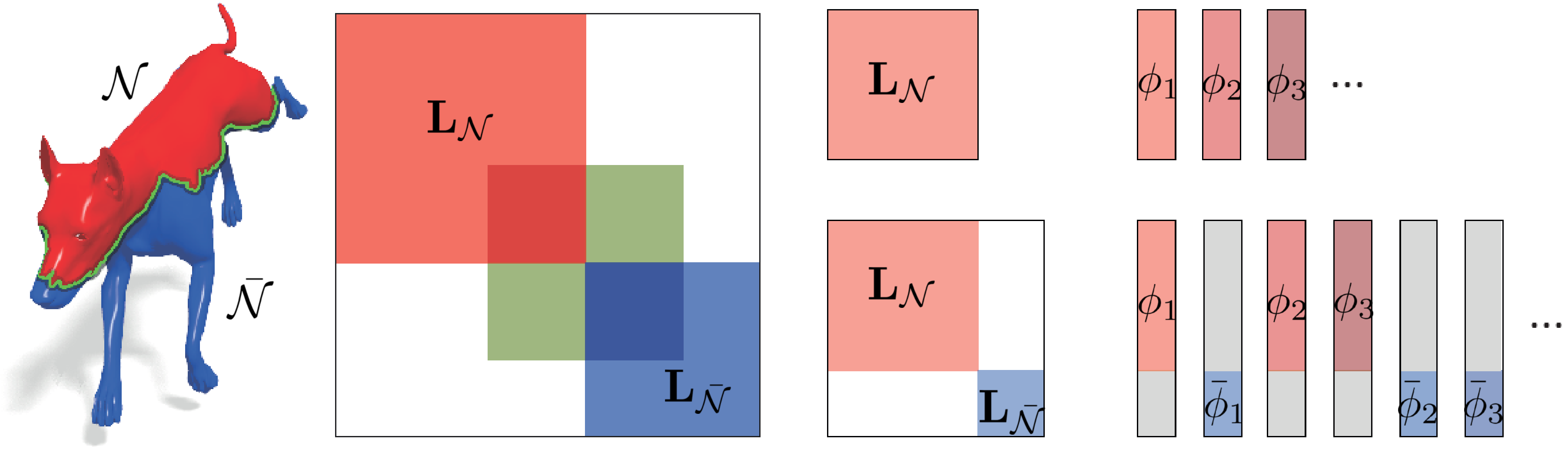
# Partial Functional Maps

Block diagonal case

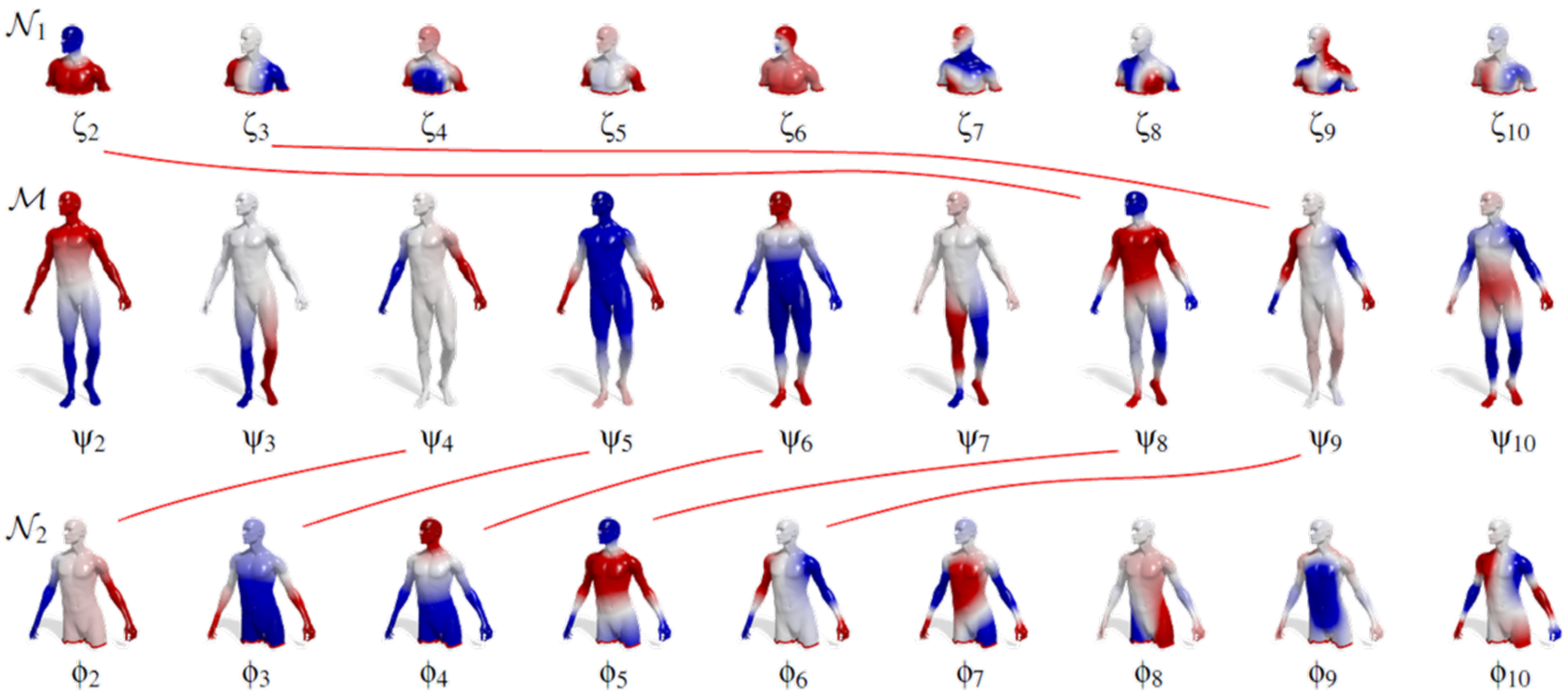


# Partial Functional Maps

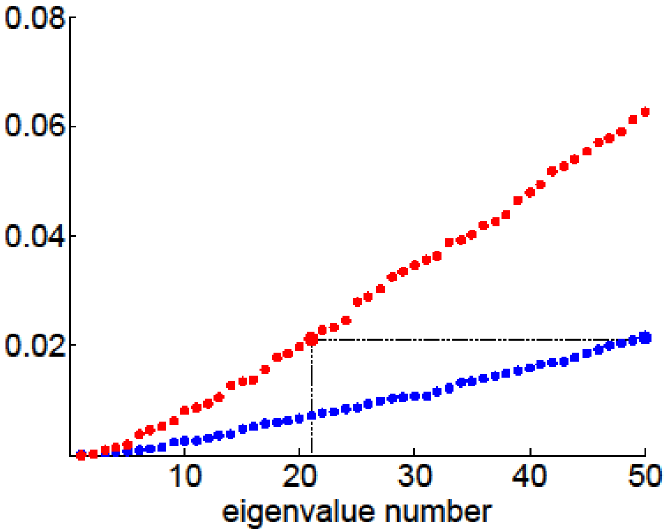
Weak coupling: eigenfunctions of the partial shape show up among those of the full shape, up to some bounded perturbation

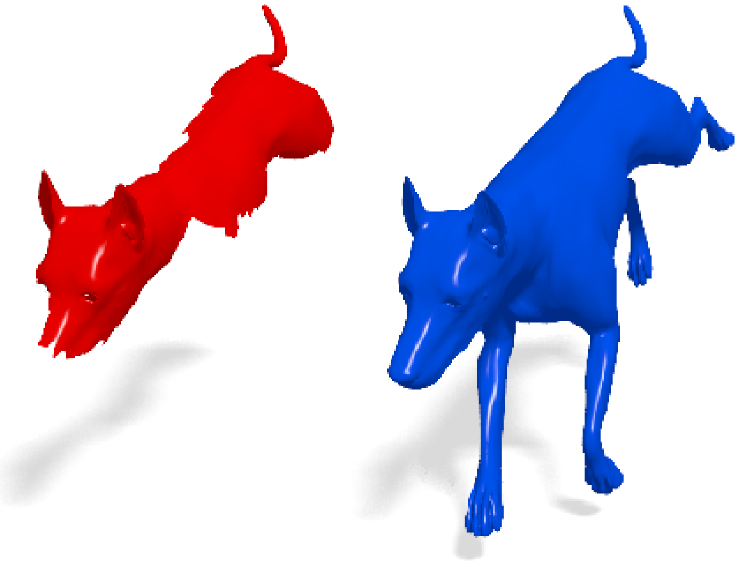


# Eigenfunction Preservation



# The Slope Rule



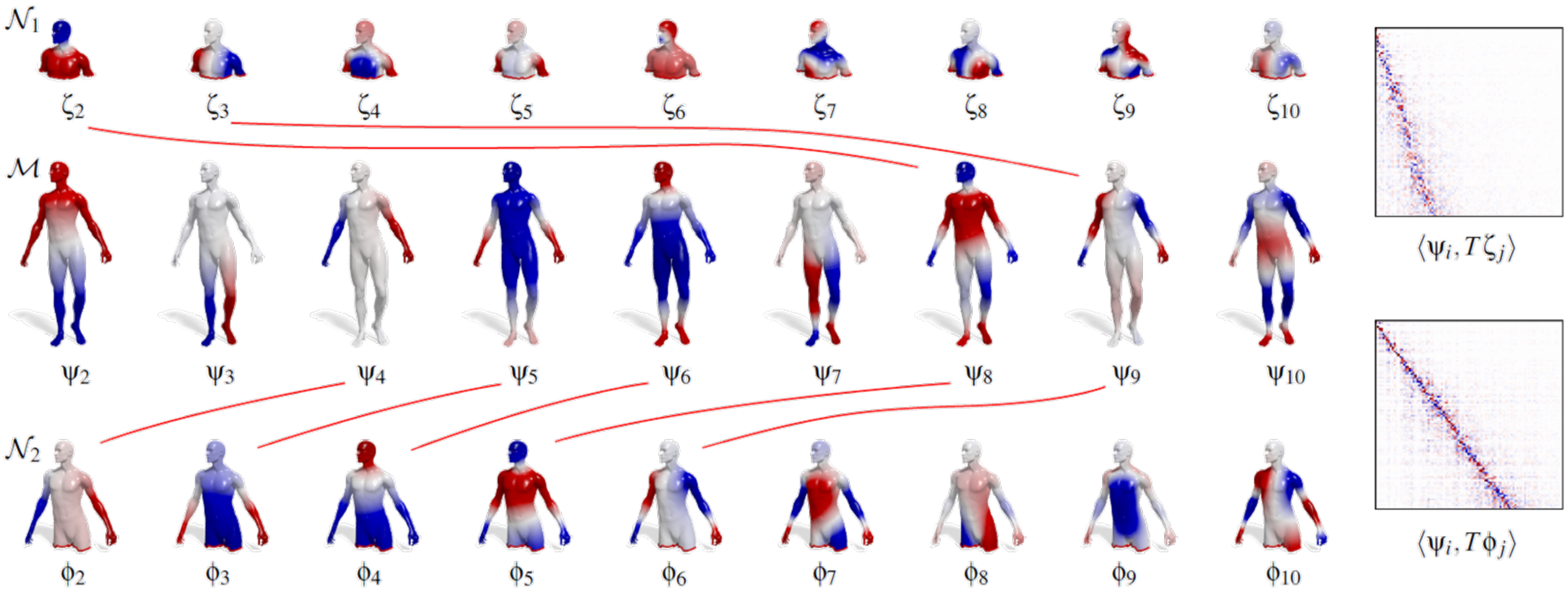



**Weyl's law**:

$$\lambda_j \approx \frac{1}{|S|} j$$

The Laplacian spectrum has slope inversely proportional to the surface area.

# The Slope Rule



# Partial Functional Maps

# That's All