

CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University

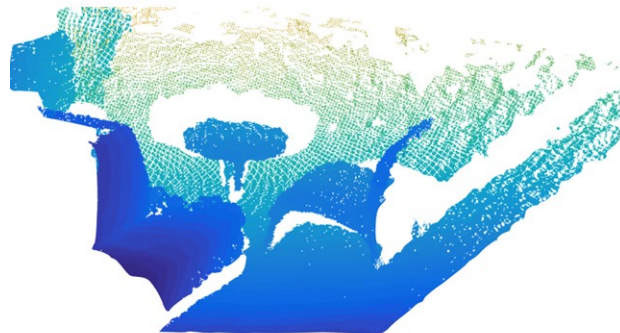


Lecture 15
18 May 2022

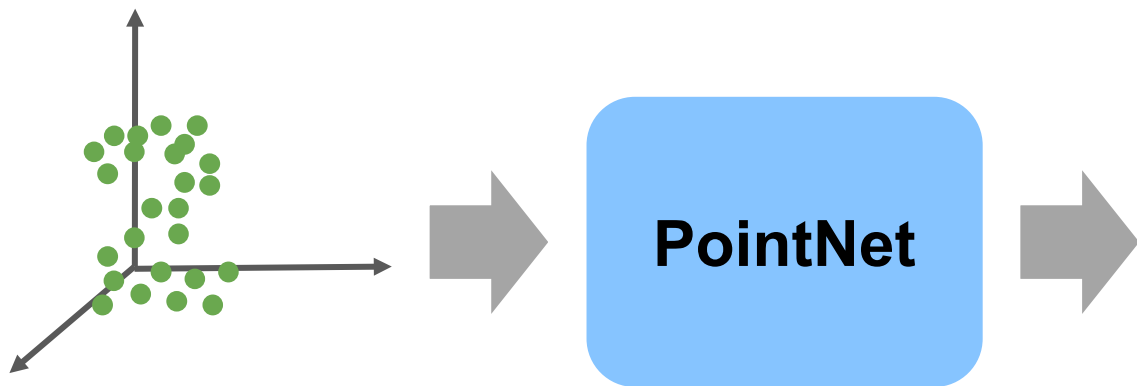


Last Time: Deep Learning on Point Cloud Data

Non-regular 3D data



Deep Nets for PCs: PointNet and PointNet++



Object Classification

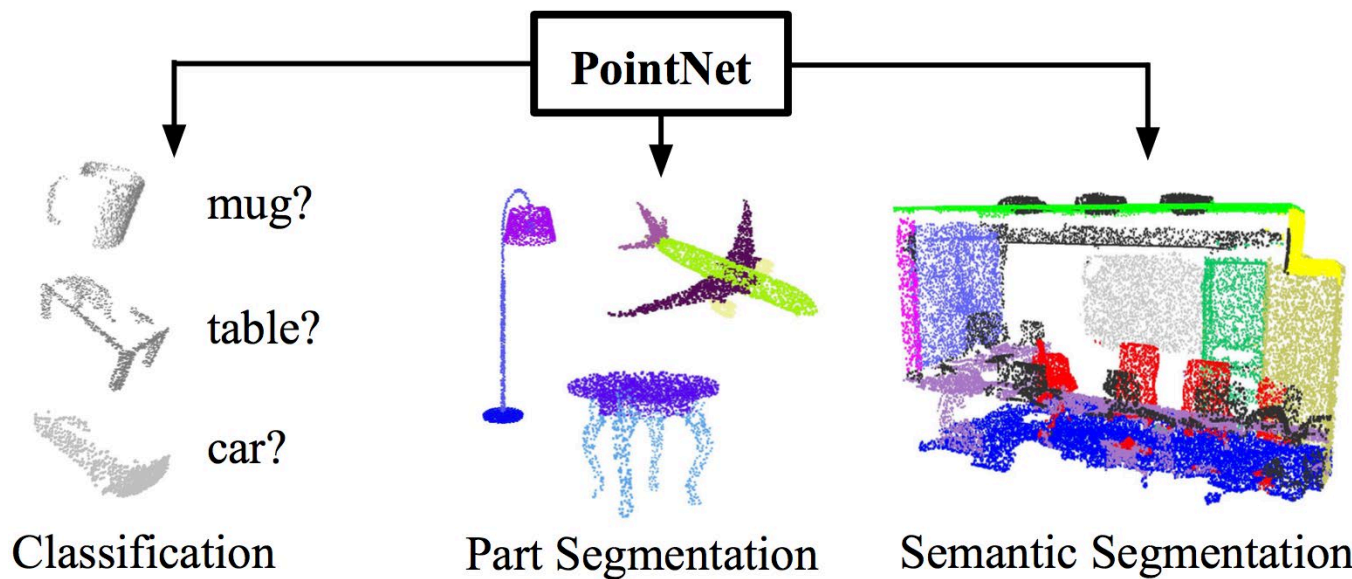
Object Part Segmentation

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

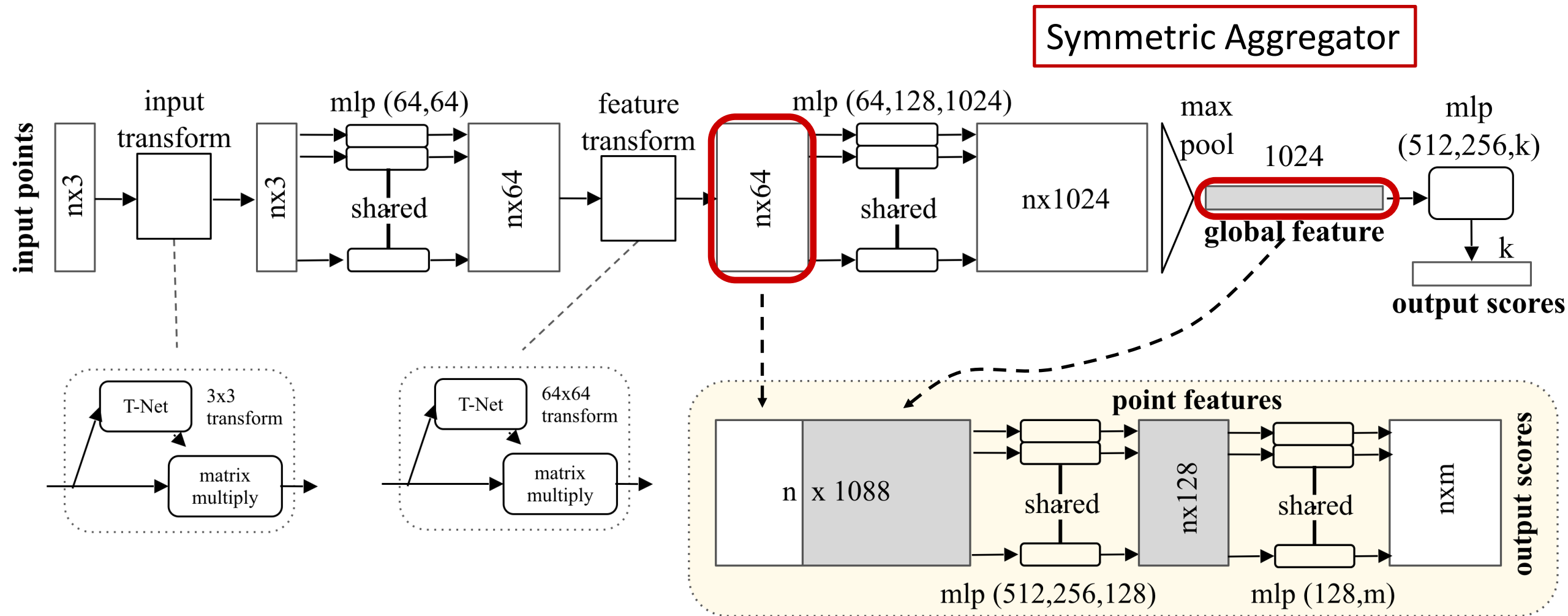
Spatial Transformation Invariance

Point cloud **rigid motions** should not alter classification results

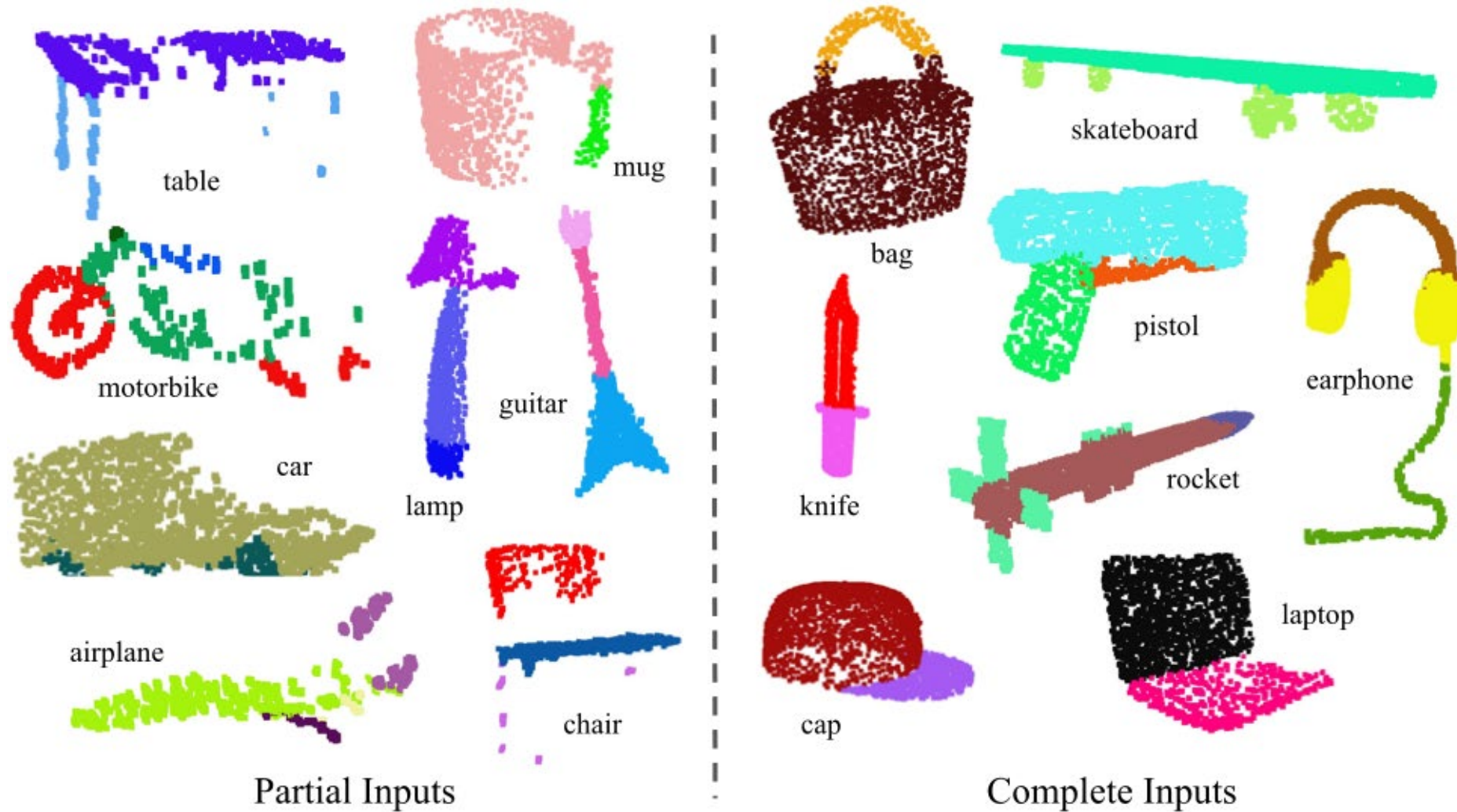
Sampling Invariance

Output a function of the underlying geometry and **not the sampling**

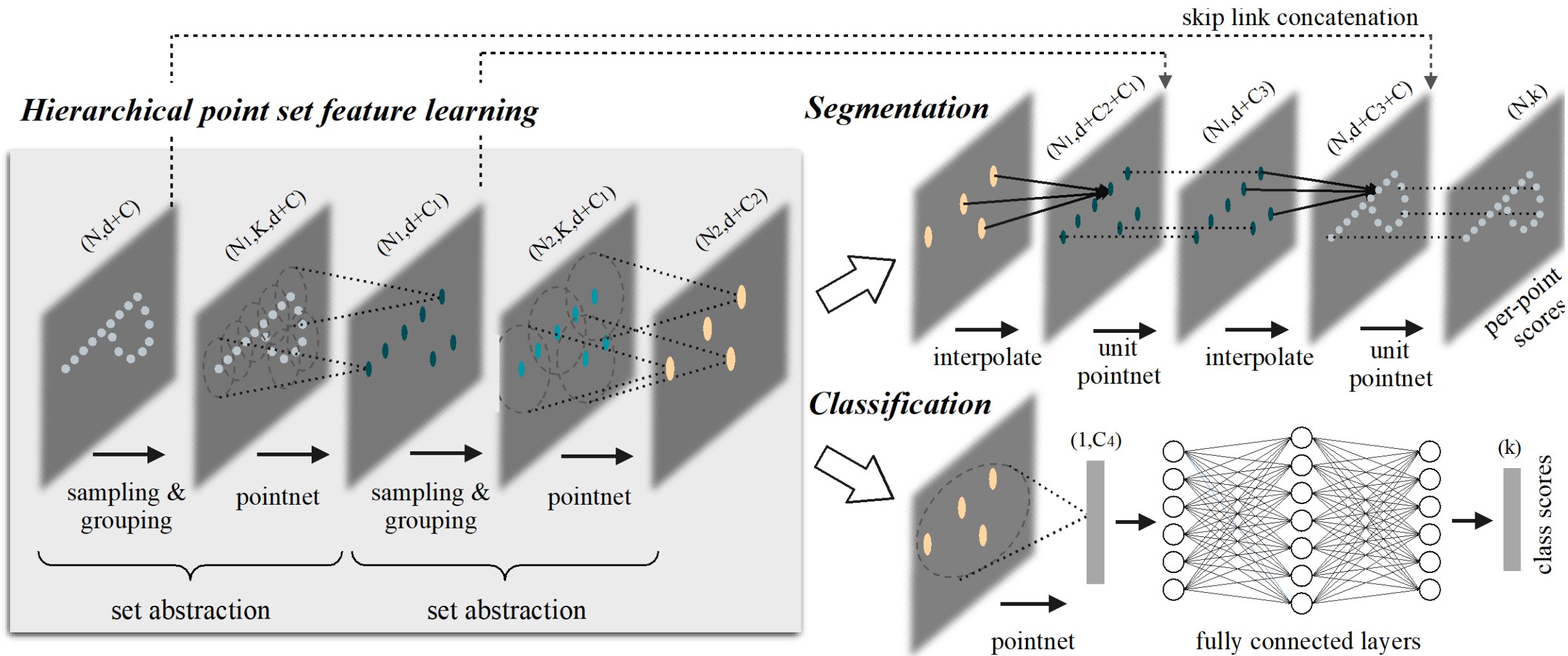
PointNet Classification and Segmentation



Results on Object Part Segmentation

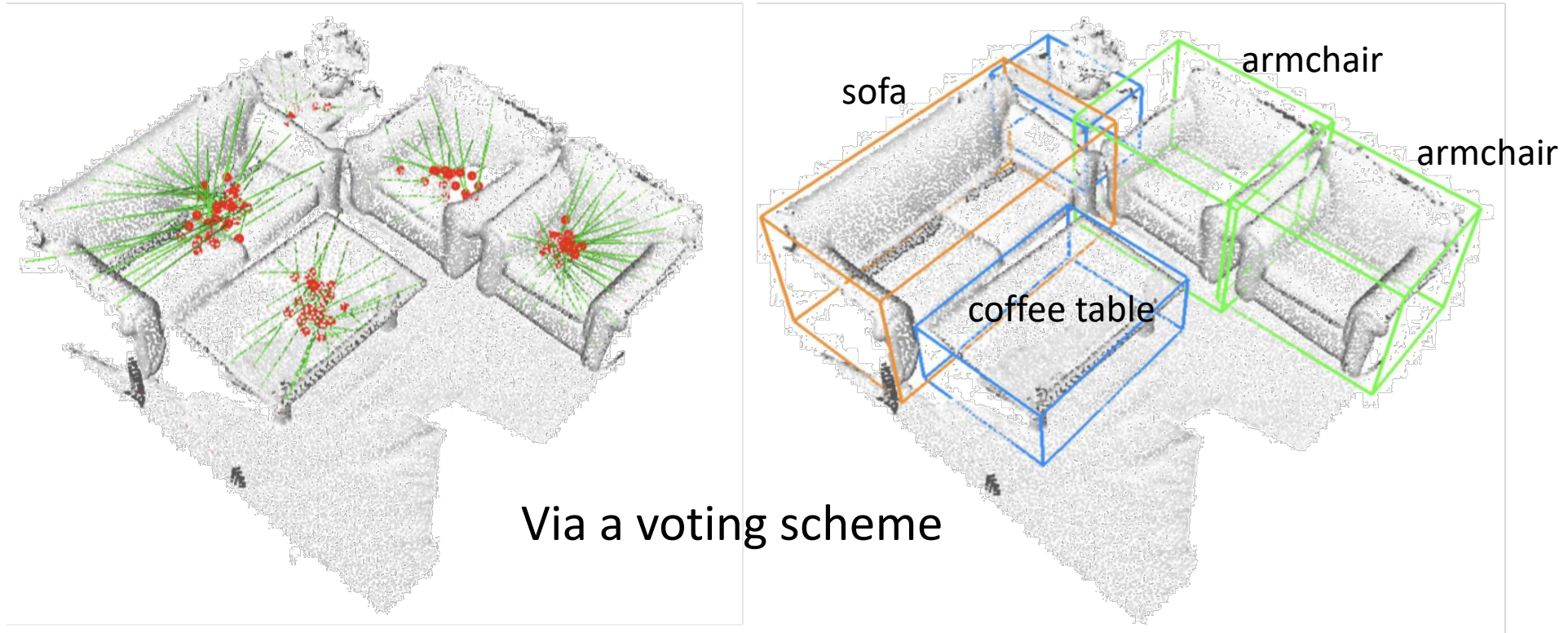


PointNet++ for Classification and Segmentation



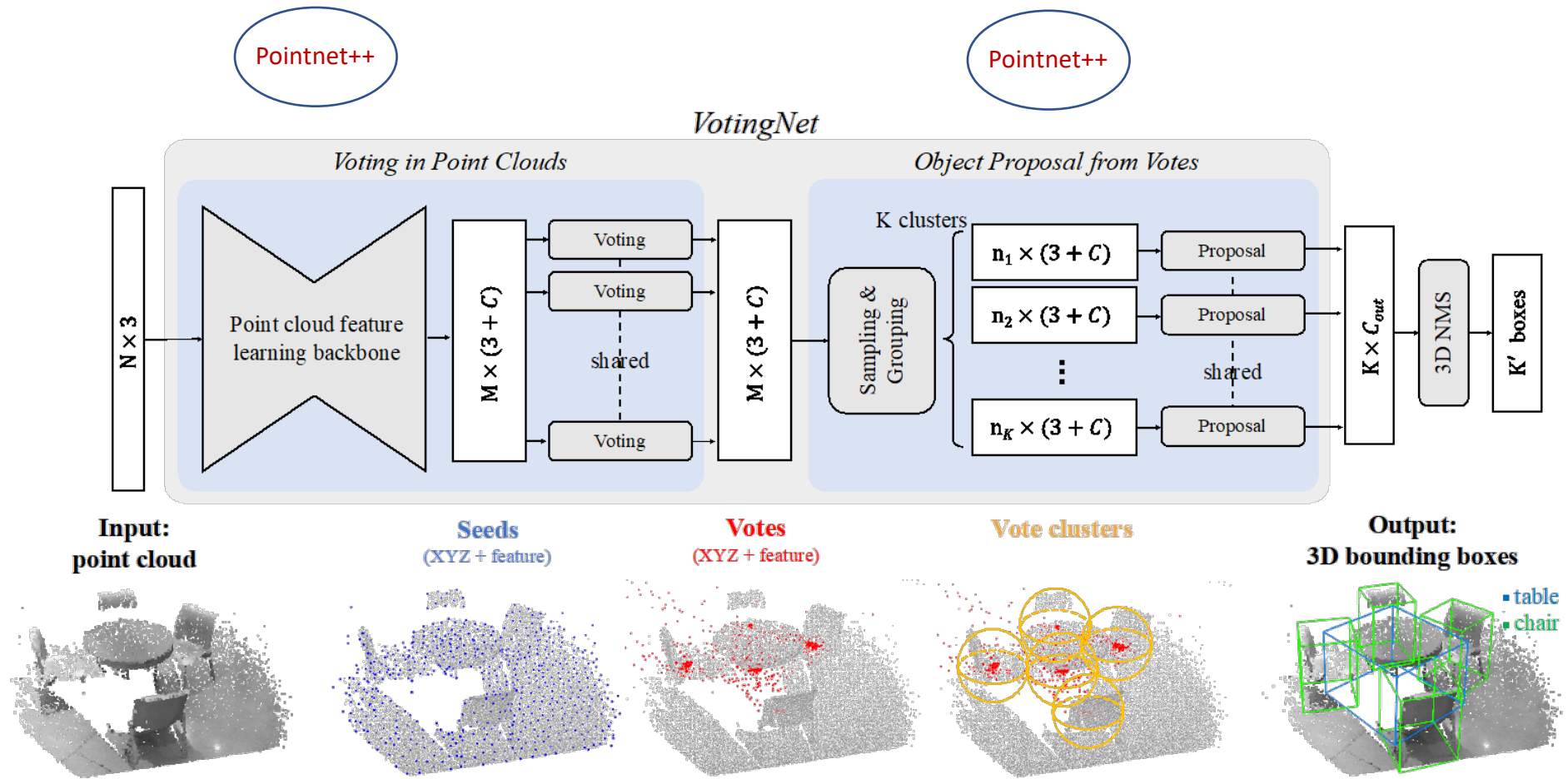
Aggregation pattern is only a function of the spatial locations of the points

Point Cloud Object Amodal Bounding Box Detection

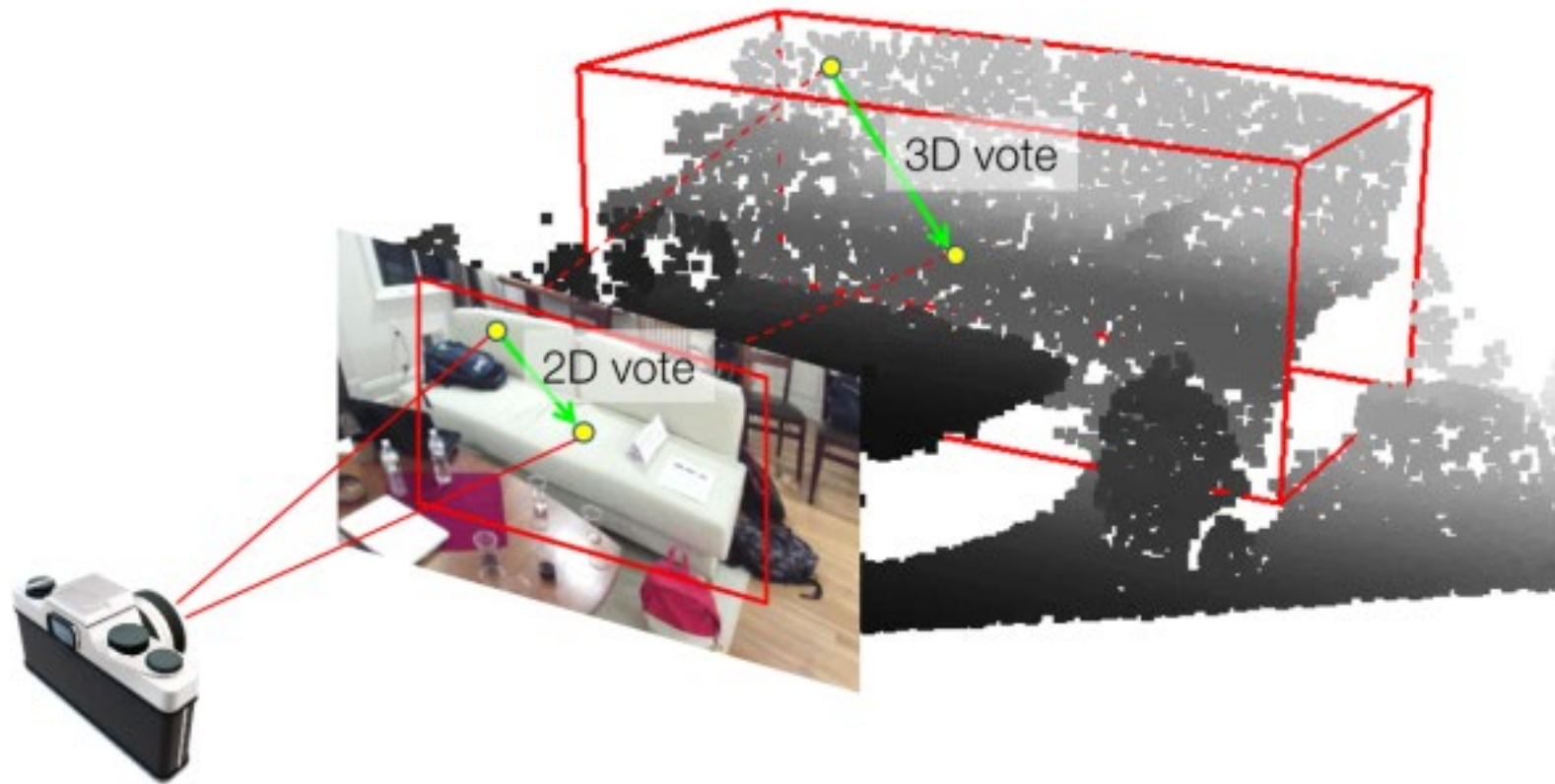


- Charles R. Qi, Or Litany, Kaiming He, Leonidas J. Guibas. *Deep Hough Voting for 3D Object Detection in Point Clouds*. ICCV 2019.
- Charles R. Qi, Xinlei Chen, Or Litany, Leonidas J. Guibas. *ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes*. CVPR 2020.

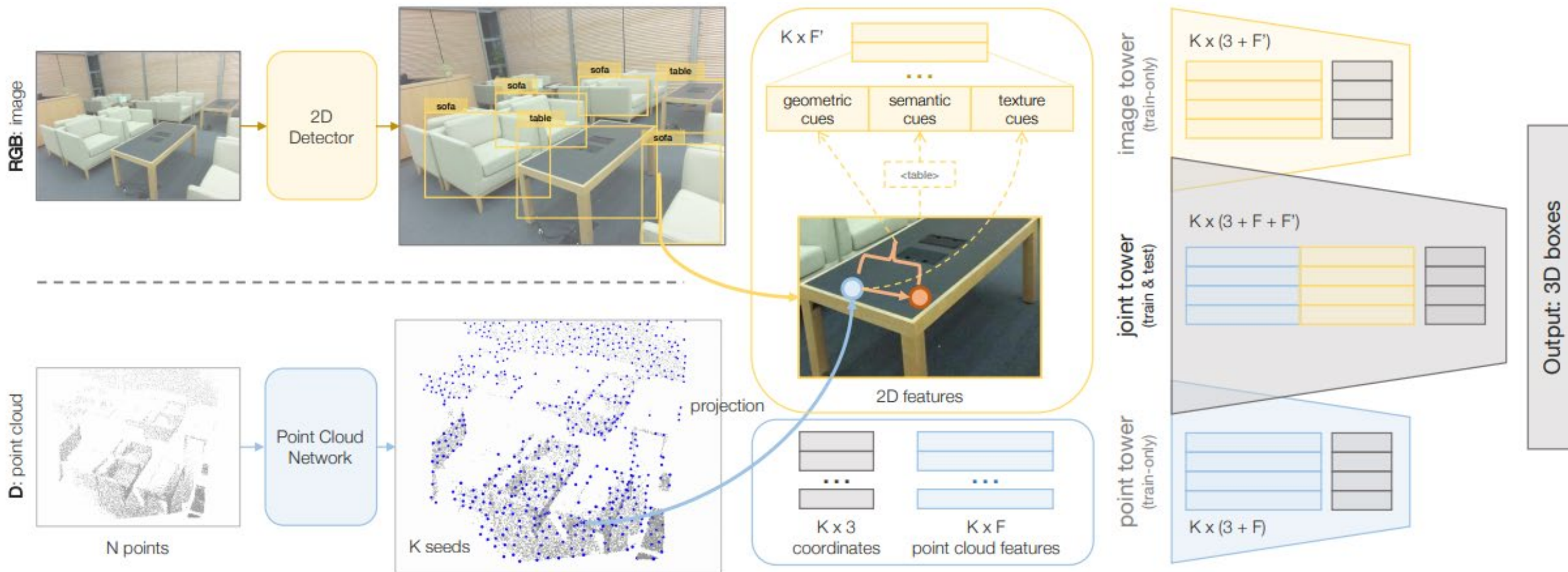
VoteNet – A Two-Stage Approach



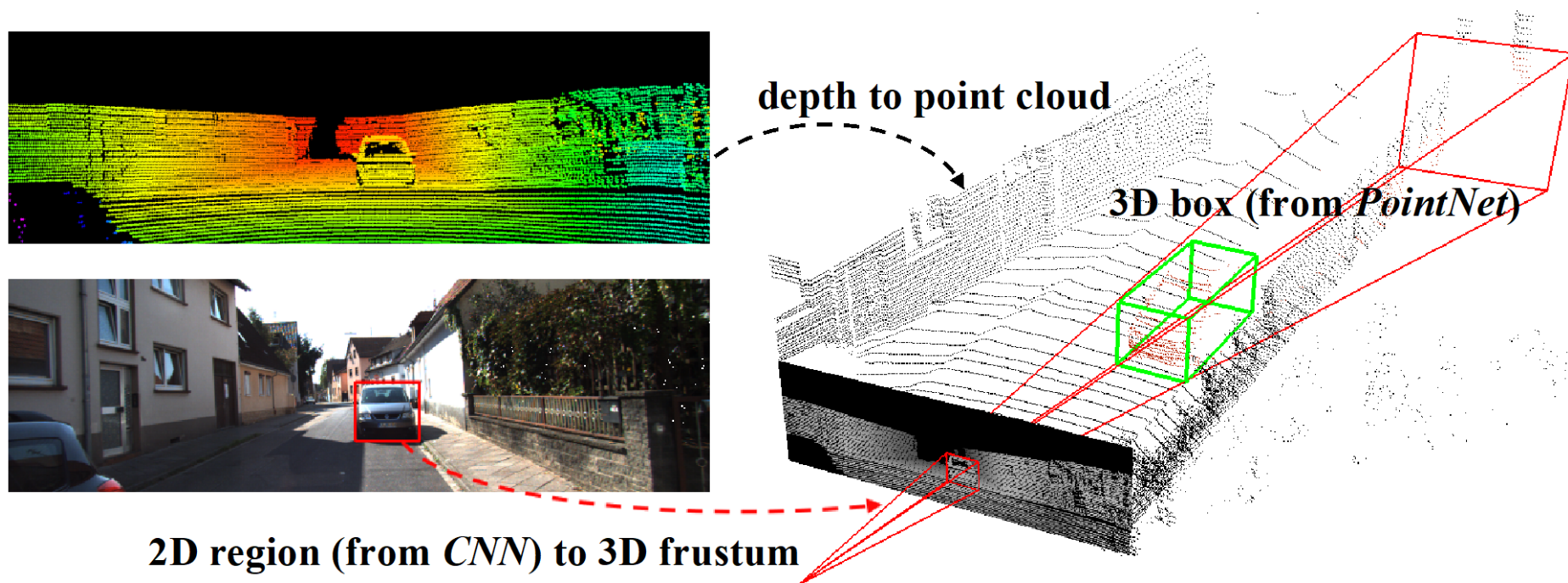
Basic idea: *ImVoteNet*



ImVoteNet Architecture

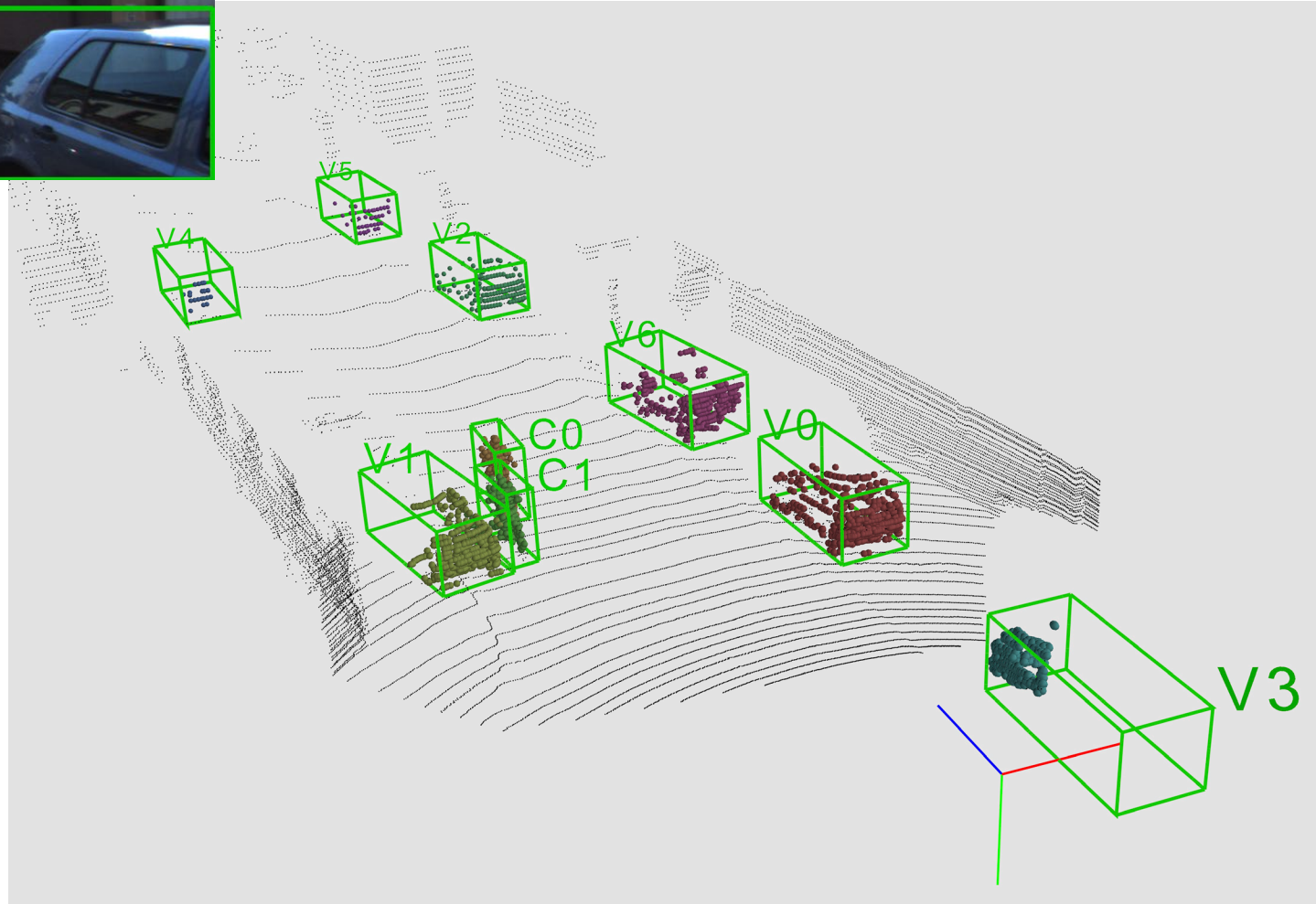
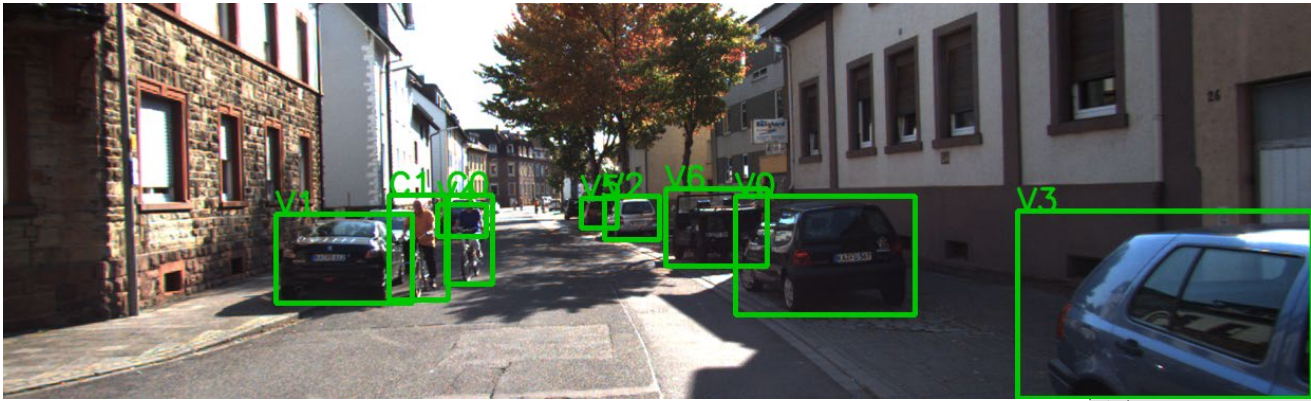


Frustum PointNets for 3D Object Detection



- + **Leveraging mature 2D detectors** for region proposal. greatly reducing 3D search space.
- + Solving 3D detection problem with **3D data and 3D deep learning**.

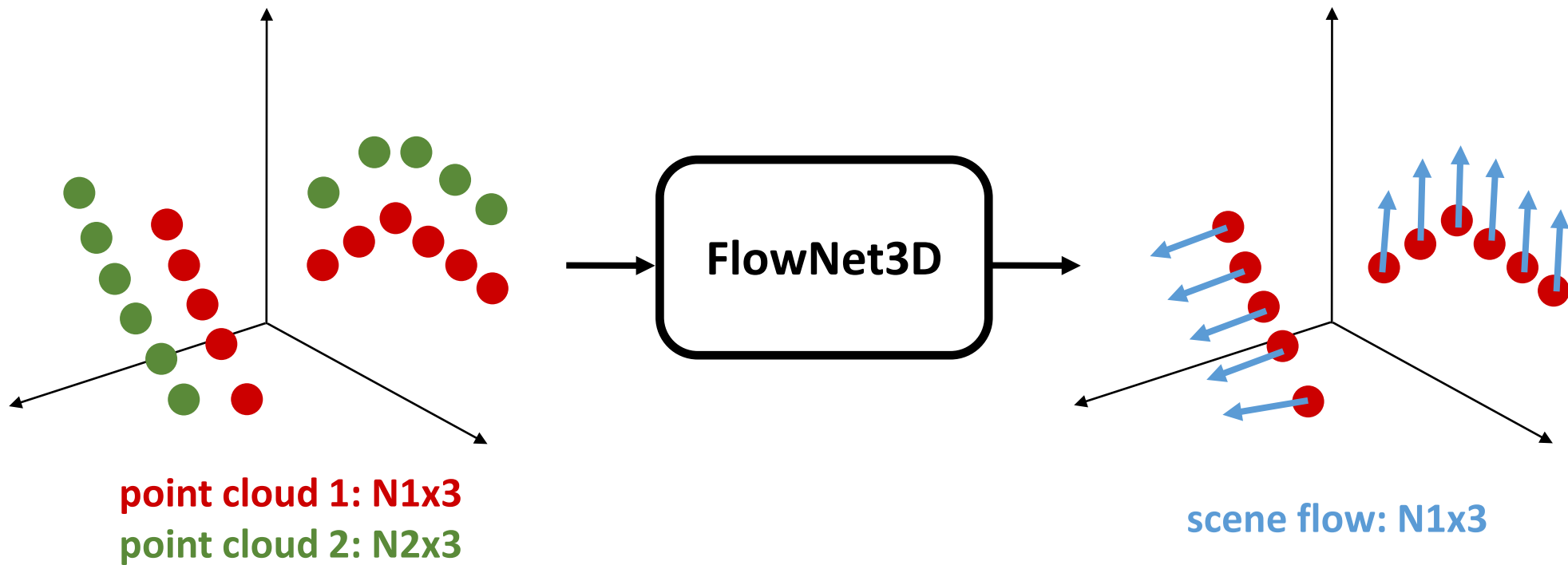
KITTI Results: Qualitative



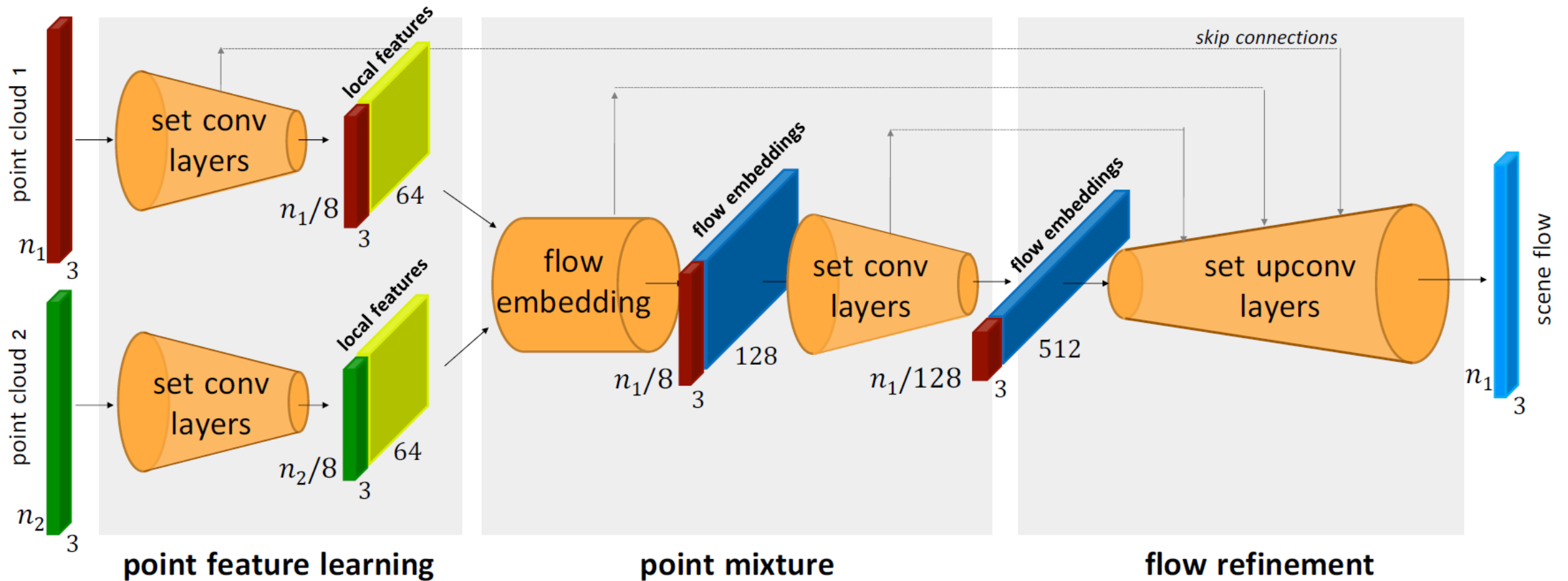
Remarkable box estimation accuracy even with a dozen of points or with very partial point clouds.

FlowNet3D – Point Clouds Over Time

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



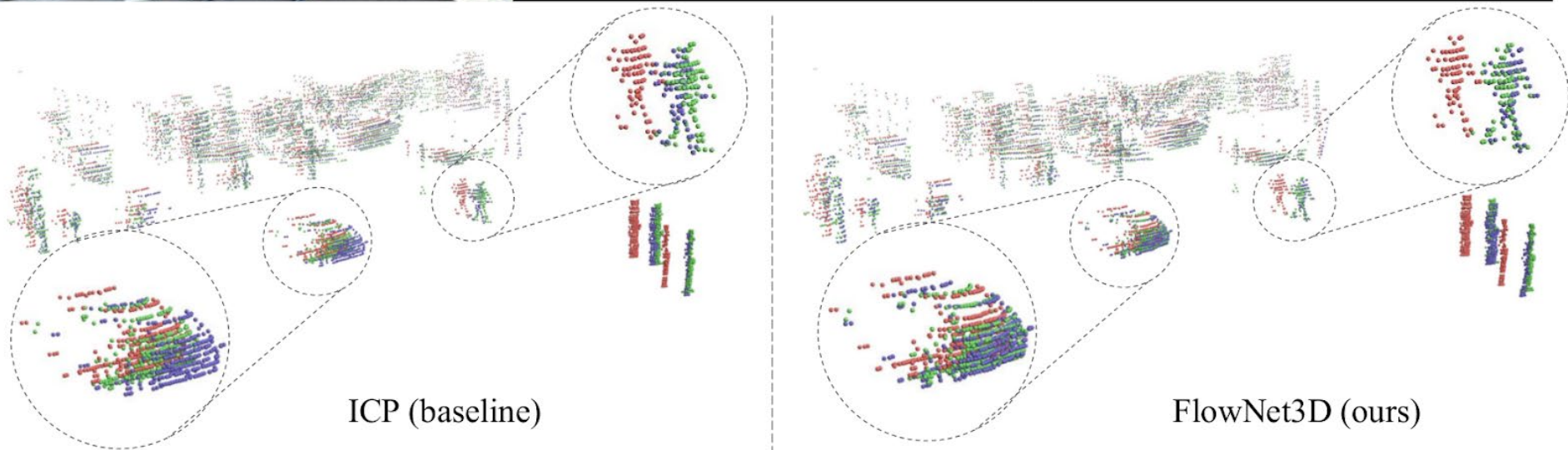
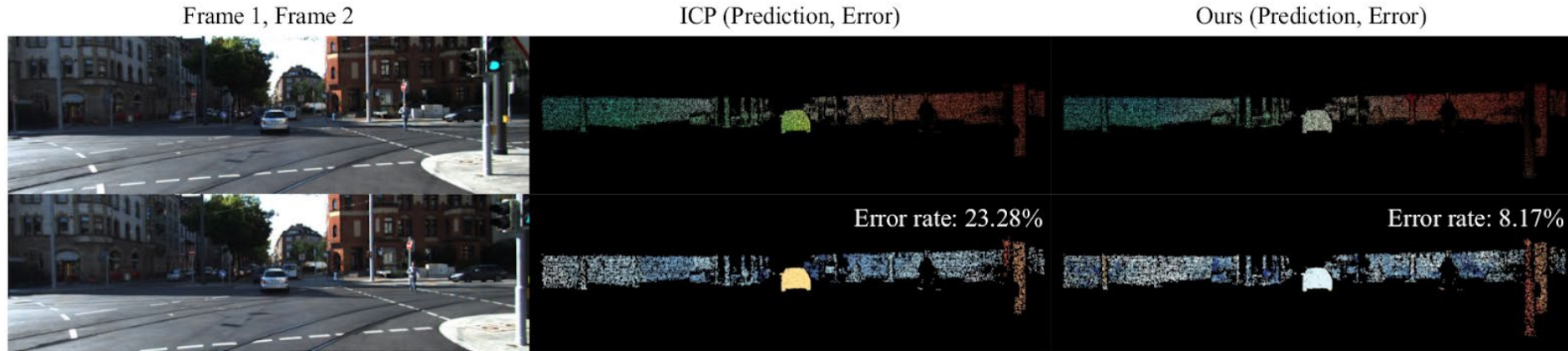
FlowNet3D – Point Clouds Over Time



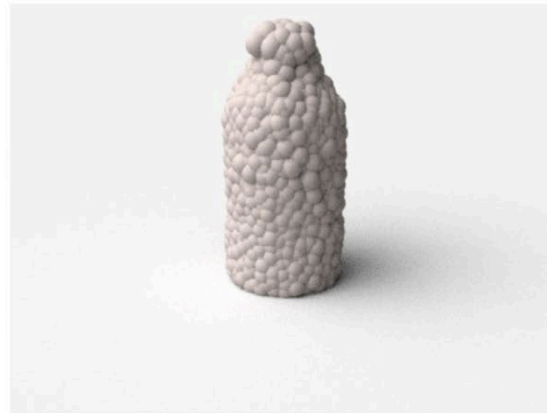
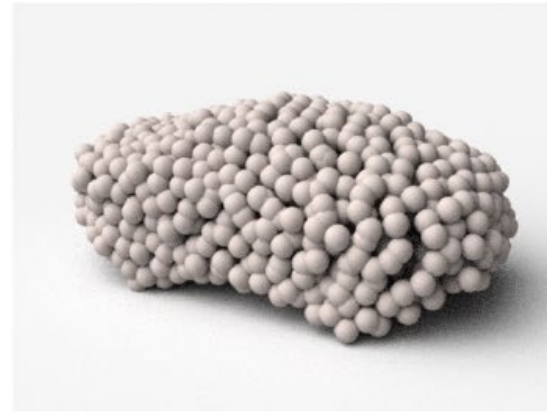
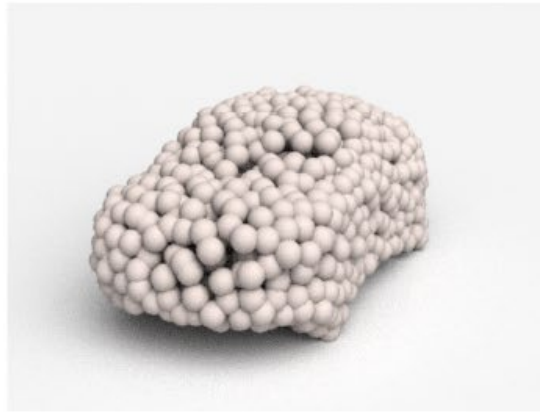
Composed of many many mini-pointnet++ modules ...

Pointnet++

KITTI Results



Point Cloud Synthesis from a Single Image

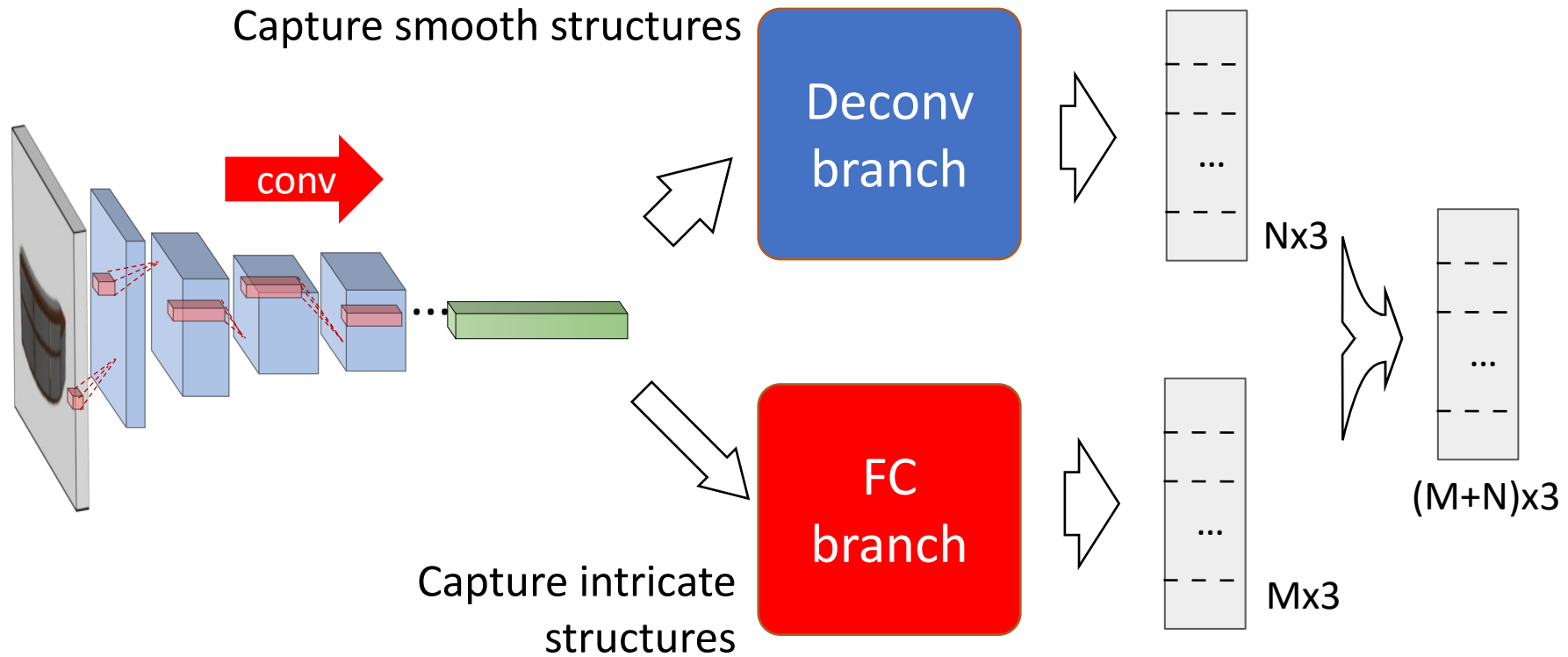


Input

Reconstructed 3D point cloud

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

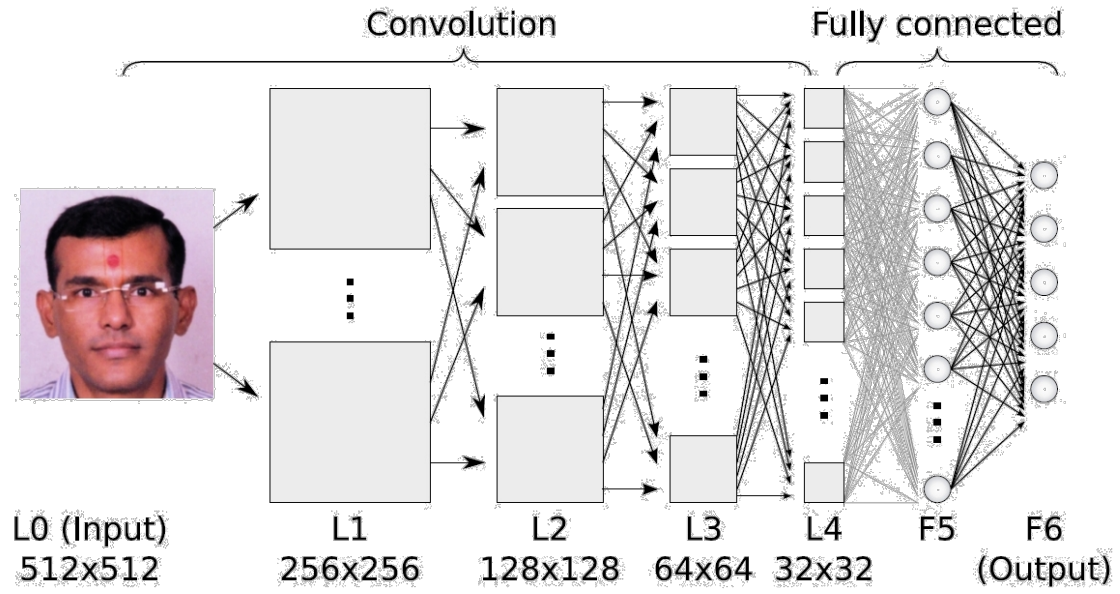
Two-Branch Architecture



Set union by array concatenation

Today: Functional Maps for Joint Data Analysis

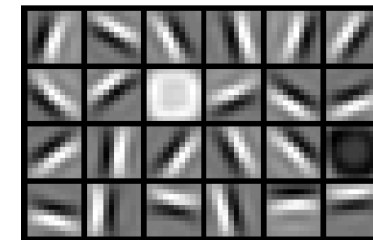
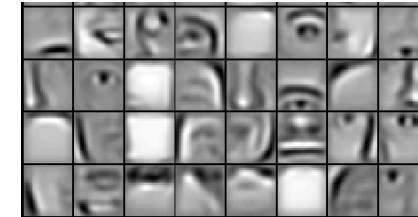
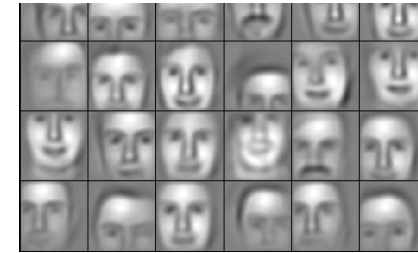
Vertical Learning Networks



[Makwana, 2016]

Data-driven feature learning at ascending abstraction layers

“Deep” nets



[Lee et al., 2009]

Horizontal Networks

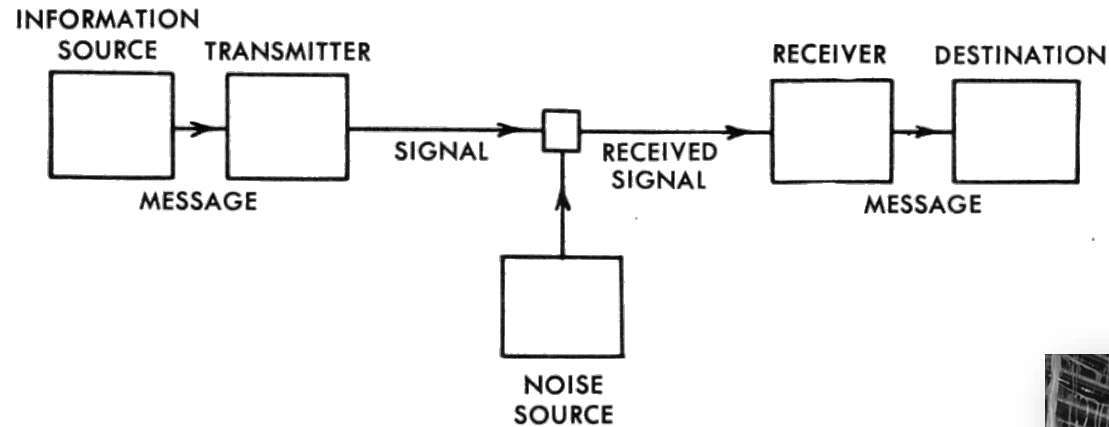


34

Similarity as a communications channel



The Mathematical Theory of Communication



Claude Shannon

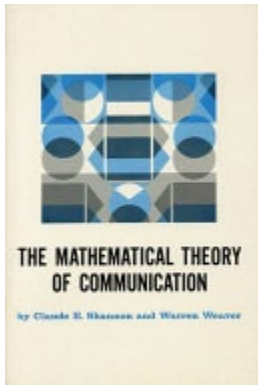
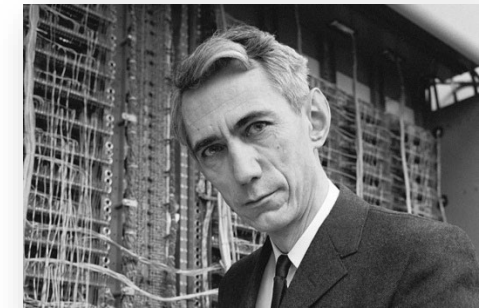


Fig. 1. — Schematic diagram of a general communication system.

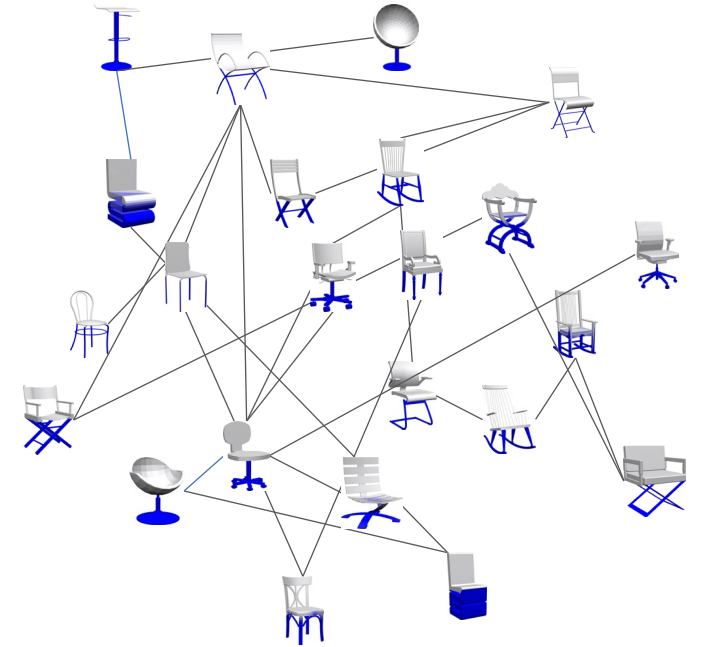


The Network View: Information Transport Between Visual Data

Networks of Images



Or of Shapes, Or of Both



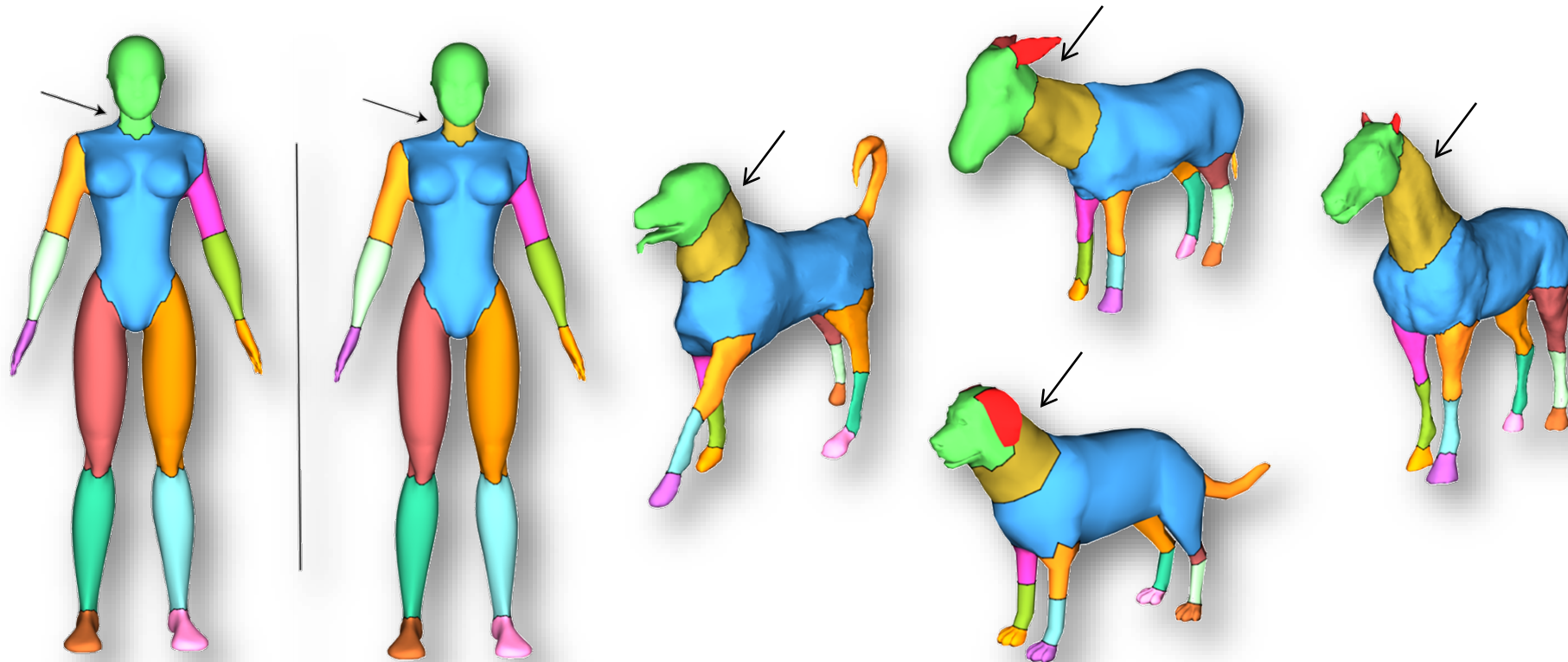
Relations Between Visual Data



Each Data Set Is Not Alone

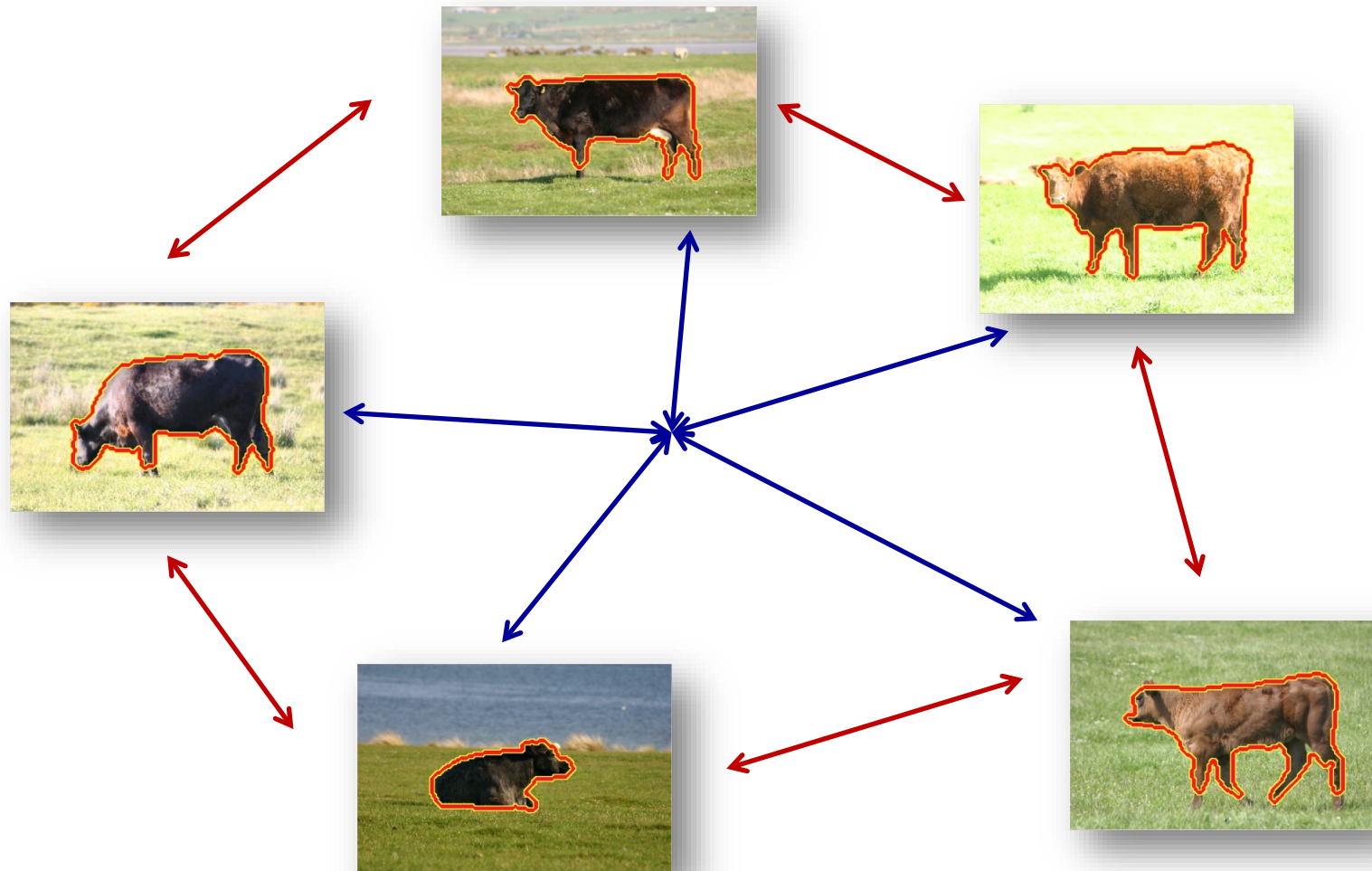
[Q. Huang, V. Koltun, L. Guibas, 2011]

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

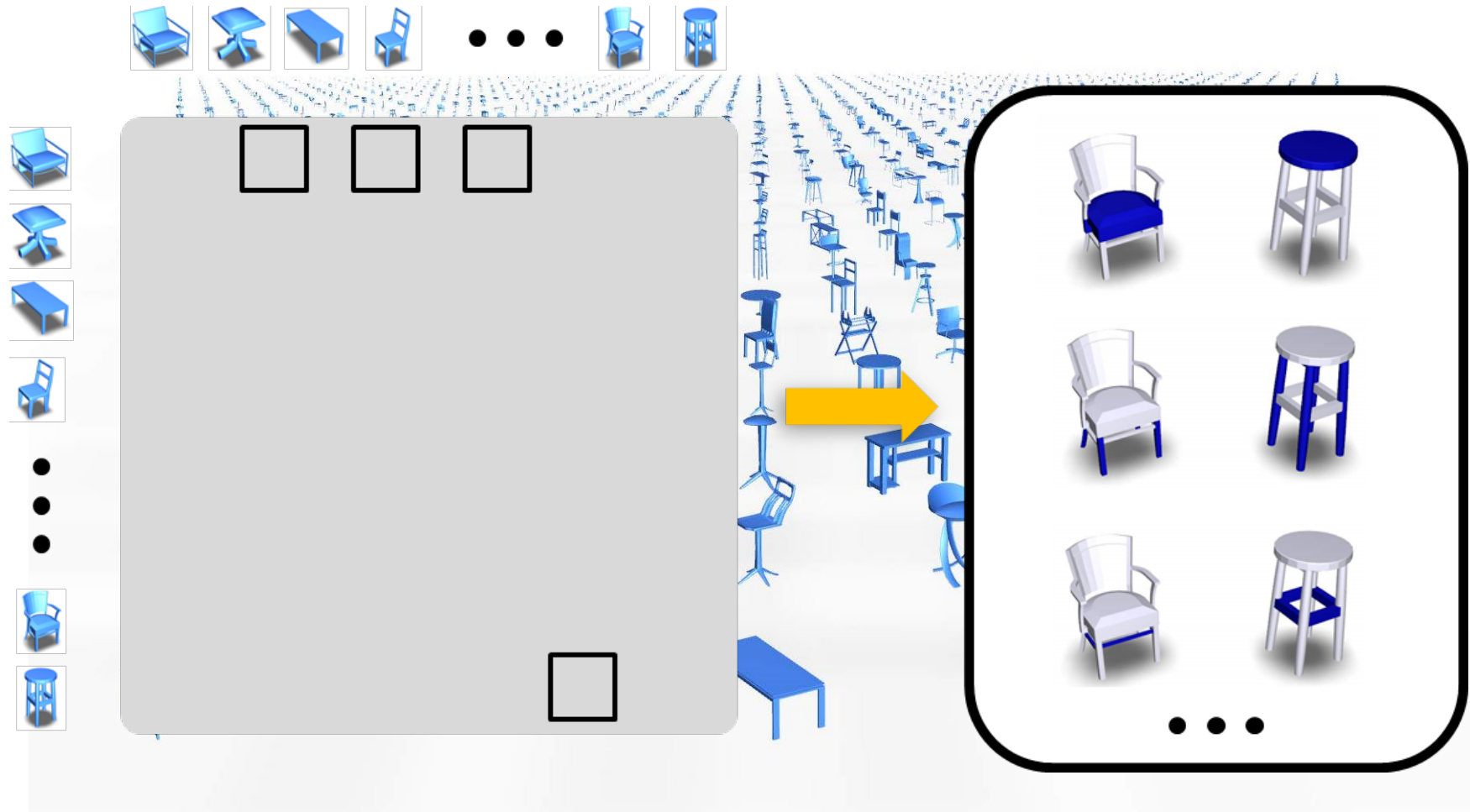


3D Segmentation

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.

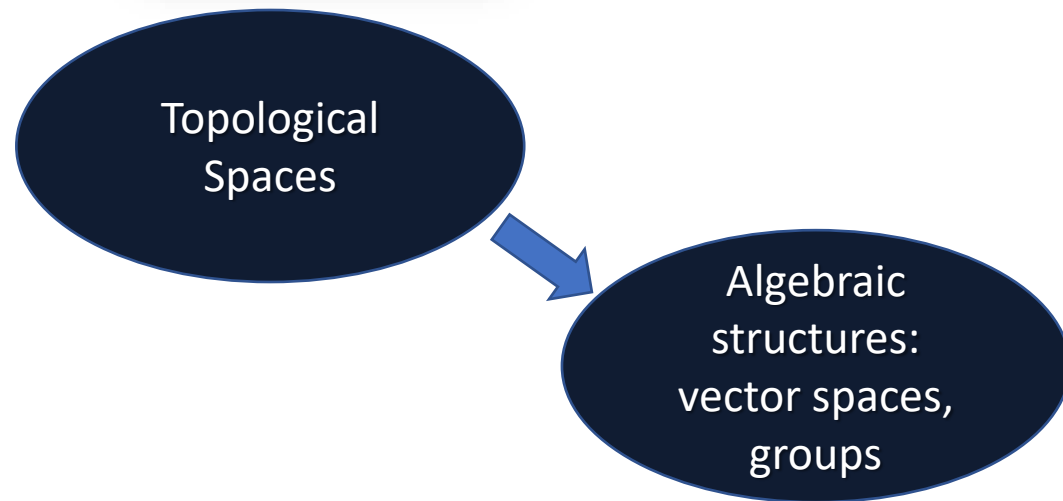
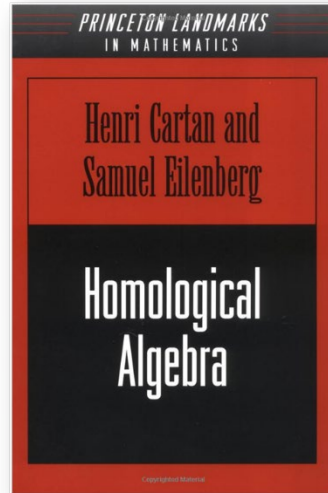
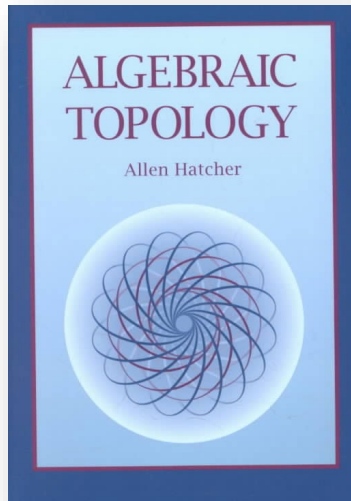
We can exploit the relational network to

- 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



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

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

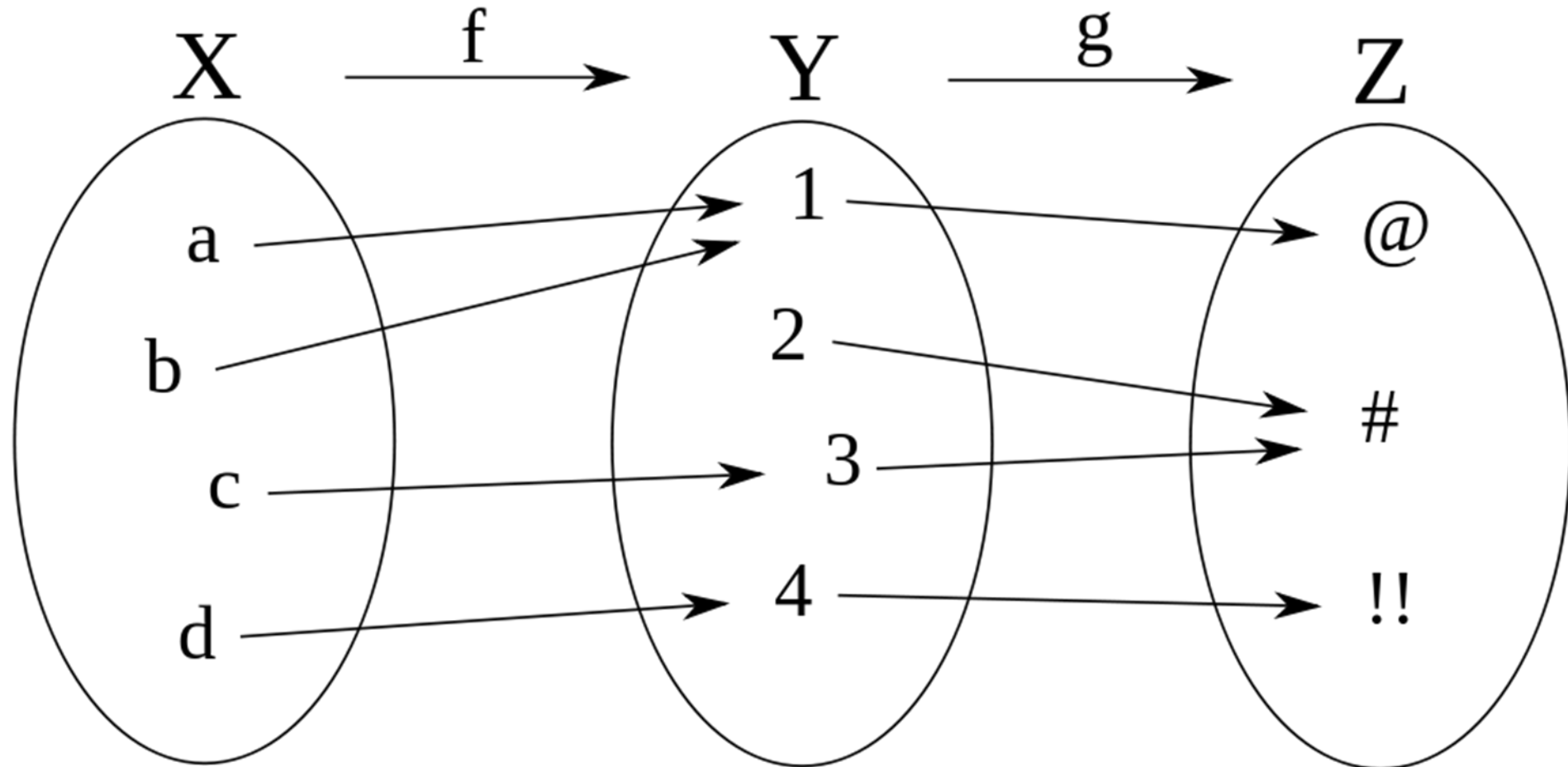
Maps as First-Class Citizens

Maps

$$\phi : X \rightarrow Y$$

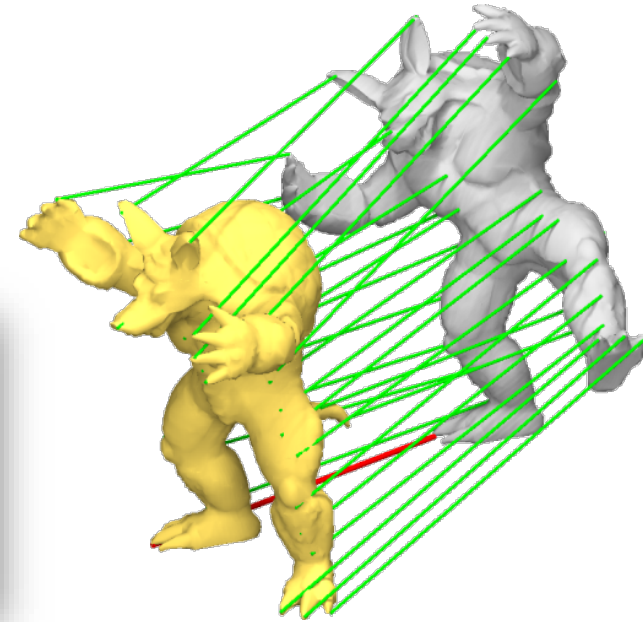
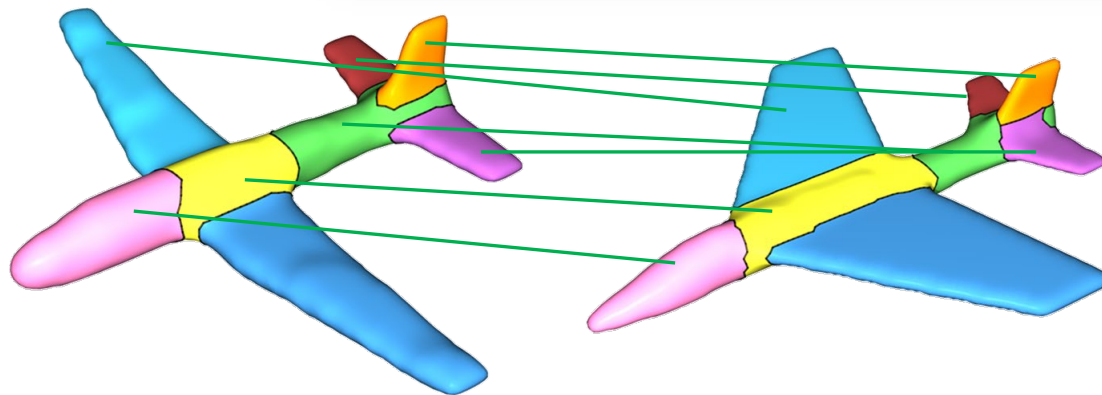
Map from X to Y

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

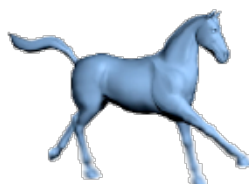
texture and
parametrization



segmentation
and labels



deformation



Matching Enables Information Transport

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 24, APRIL 2002

Shape Matching and Object Recognition Using Shape Contexts

Serge Belongie, Member, IEEE, Jitendra Malik, Member, IEEE, and Jan Puzicha

Abstract—We present a novel approach to measuring similarity between shapes and exploit it for object recognition. In our framework, the measurement of similarity is preceded by 1) solving for correspondences between points on the two shapes, 2) using the correspondences to estimate an aligning transform. In order to solve the correspondence problem, we attach a descriptor, the shape context, to each point. The shape context at a reference point captures the distribution of the remaining points relative to it, thus offering a globally discriminative characterization. Corresponding points on two similar shapes will have similar shape contexts, enabling us to solve for correspondences as an optimal assignment problem. Given the point correspondences, we estimate the transform that best aligns the two shapes; regularized thin-plate splines provide a flexible class of transformations maps for this purpose. The dissimilarity between the two shapes is computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform. We treat recognition in a nearest-neighbor classification framework as the problem of finding the stored prototype shape that is maximally similar to that in the image.

Index Terms—Shape, object recognition, digit recognition, correspondence problem, MPEG-7 templates.

1. INTRODUCTION

CONSIDER the two handwritten digits in Fig. 1. Regarded as vectors of pixel brightness values and compared using L_2 norms, they are very different. However, regarded as shapes, they appear rather similar to a human observer. Our objective in this paper is to operationalize this notion of shape similarity, with the ultimate goal of using it as a basis for category-level recognition. We approach this as a three-stage process:

1. solve the correspondence problem between the two shapes as a matching problem;
2. use the correspondences to estimate an aligning transform and compute the distance between the two shapes as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform;
3. use the aligned shapes to estimate an aligning transform that can be traced at least as far back as D'Arcy Thompson's work. On Growth and Form [5], Thompson observed that related but not identical shapes can often be deformed into alignment using simple coordinate transformations.

Our primary contribution is the development of another concept by means of which we can compare shapes. Another well-developed concept by means of which we can compare shapes is the Hausdorff distance. This vein was developed by Edelsz and Taubin [6]. Our primary contribution is the development of another concept by means of which we can compare shapes. Another well-developed concept by means of which we can compare shapes is the Hausdorff distance. This vein was developed by Edelsz and Taubin [6]. Our primary contribution is the development of another concept by means of which we can compare shapes. Another well-developed concept by means of which we can compare shapes is the Hausdorff distance. This vein was developed by Edelsz and Taubin [6].

At the heart of our approach is a tradition of matching shapes by deformation that can be traced at least as far back as D'Arcy Thompson's work. On Growth and Form [5], Thompson observed that related but not identical shapes can often be deformed into alignment using simple coordinate transformations.

1. Introduction

Correspondence estimation is one of the fundamental challenges in computer vision lying in the core of many of the most interesting and difficult problems in computer vision. A predominant paradigm in such cases has been to model the problem as a matching problem, where the goal is to find a set of interest points, whose power is in the ability to discriminate between different features. In many cases, however, the features are not well defined, and the matching problem is ill-posed. In this paper, we propose a new paradigm for matching, based on the idea of shape context.

Image Matching via Saliency Region Correspondences

Alexander Toshev, Jianbo Shi, and Kostas Daniilidis*
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA

Abstract
We introduce the notion of co-saliency for image matching. Our matching algorithm combines the discriminative power of feature correspondences with the descriptive power of saliency segments. Co-saliency matching score favors correspondences that are consistent with soft image segmentation as well as with local point feature matching. We express the matching model via a joint image inter-image relations. The dominant spectral components of this graph (JIG) whose edge weights represent intra- as well as inter-image relations. The dominant spectral alignment of the graph leads to simultaneous pixel-wise alignment of the images and saliency-based synchronization, which characterizes these spectral components, can be directly used as a similarity metric as well as a positive feedback for updating and establishing new point correspondences. We present experiments showing the extraction of matching regions and pointwise correspondences, and the utility of the global image similarity in the context of place recognition.

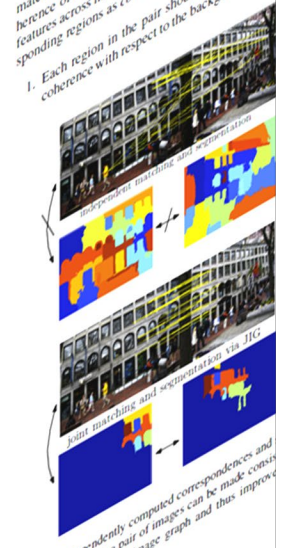


Figure 1. Independently computed correspondences and a bipartite graph for a pair of images can be made consistent via the joint image graph and thus improve

Abstract
As a fundamental problem in pattern recognition, graph matching has found a variety of applications in the field of computer vision. In graph matching, patterns are modeled as graphs and pattern recognition amounts to finding a correspondence between the nodes of different graphs. There are many ways in which the problem has been formulated, but most can be cast in general as a quadratic assignment problem, where a linear term in the objective function encodes node compatibility functions. The main research focus in this theme is about designing efficient algorithms for solving approximately the quadratic assignment problem, since it is NP-hard.

In this paper, we turn our attention to the complementary problem, how to estimate compatibility functions such that the solution of the resulting graph matching problem best matches the expected solution that a human would manually provide. We present a method for learning graph matching: the training examples are pairs of graphs and the labels are matchings between pairs of graphs. We present experimental results with real image data which give evidence that learning can improve the performance of standard graph matching algorithms. In particular, it turns out that linear assignment with such a learning scheme may improve over state-of-the-art quadratic assignment methods. This finding suggests that for a range of problems where quadratic assignment was thought to be essential for securing good results, linear assignment, which is far more efficient, could be just sufficient if learning is performed. This enables speed-ups of graph matching by up to 4 orders of magnitude while retaining state-of-the-art accuracies.

1. Introduction

Graphs are commonly used as abstract representations for complex scenes, and many computer vision problems can be formulated as an attributed graph matching problem, where the nodes of the graphs correspond to relational aspects between features (both nodes and edges can be attributed, i.e. they can encode feature vectors). Graph matching then consists in finding a correspondence between nodes of the graphs, such that they "look most similar" when the labels are taken into account. This problem is mathematically formulated as a

Learning Graph Matching

Tibério S. Caetano, Li Cheng, Quoc V. Le and Alex J. Smola
Statistical Machine Learning Program, NICTA and ANU
Canberra ACT 0200, Australia

Abstract
quadratic assignment problem, which consists in finding the assignment that maximizes an objective function encoding local compatibilities (a linear term) and structural compatibilities (a quadratic term). The main body of research in graph matching has then been focused on devising more accurate and/or faster algorithms to solve the problem approximately (since it is NP-hard). The compatibility functions used in graph matching are typically handcrafted.

An interesting question arises in this context: If we are given two attributed graphs, G and G' , should the optimal match be uniquely determined? For example, assume a surveillance camera in an airport's lounge. Now, assume the same G and G' come from two images in a photograph used in graph matching. Should the optimal match be the same in both situations? If the algorithm takes into account the same G and G' instead of the optimal solutions to the quadratic assignment problem, would the match be the same? We argue that, if we know the "conditions" under which a pair of graphs has been extracted, then we should take into account how graphs arising in those cases. This is how graph matching is approached today.

In this paper we address what we believe to be a limitation of this approach. We argue that, if we do not take into account the "conditions" explicitly into account, we should take into account how graphs arising in those cases. This is how graph matching is approached today.

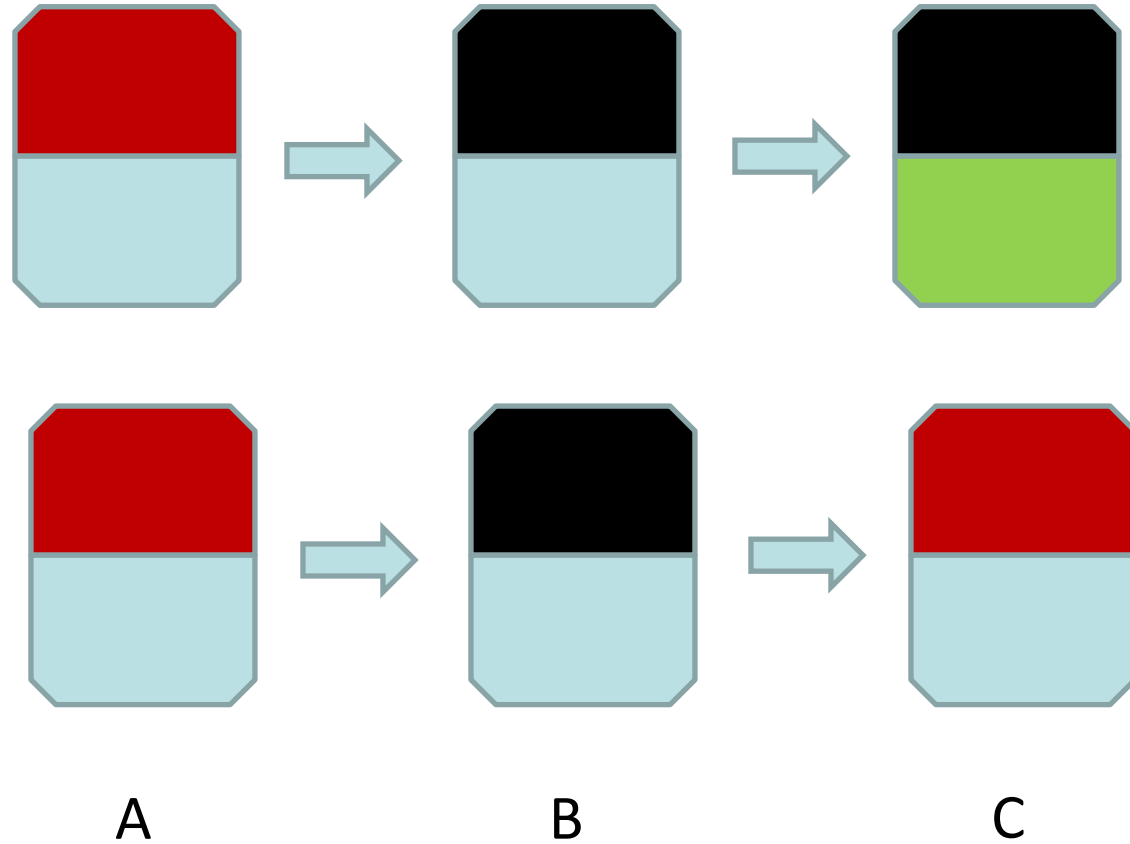
As a fundamental problem in pattern recognition, graph matching has found a variety of applications in the field of computer vision. In graph matching, patterns are modeled as graphs and pattern recognition amounts to finding a correspondence between the nodes of different graphs. There are many ways in which the problem has been formulated, but most can be cast in general as a quadratic assignment problem, where a linear term in the objective function encodes node compatibility functions. The main research focus in this theme is about designing efficient algorithms for solving approximately the quadratic assignment problem, since it is NP-hard.

In this paper, we turn our attention to the complementary problem, how to estimate compatibility functions such that the solution of the resulting graph matching problem best matches the expected solution that a human would manually provide. We present a method for learning graph matching: the training examples are pairs of graphs and the labels are matchings between pairs of graphs. We present experimental results with real image data which give evidence that learning can improve the performance of standard graph matching algorithms. In particular, it turns out that linear assignment with such a learning scheme may improve over state-of-the-art quadratic assignment methods. This finding suggests that for a range of problems where quadratic assignment was thought to be essential for securing good results, linear assignment, which is far more efficient, could be just sufficient if learning is performed. This enables speed-ups of graph matching by up to 4 orders of magnitude while retaining state-of-the-art accuracies.

1. Introduction

Graphs are commonly used as abstract representations for complex scenes, and many computer vision problems can be formulated as an attributed graph matching problem, where the nodes of the graphs correspond to relational aspects between features (both nodes and edges can be attributed, i.e. they can encode feature vectors). Graph matching then consists in finding a correspondence between nodes of the graphs, such that they "look most similar" when the labels are taken into account. This problem is mathematically formulated as a

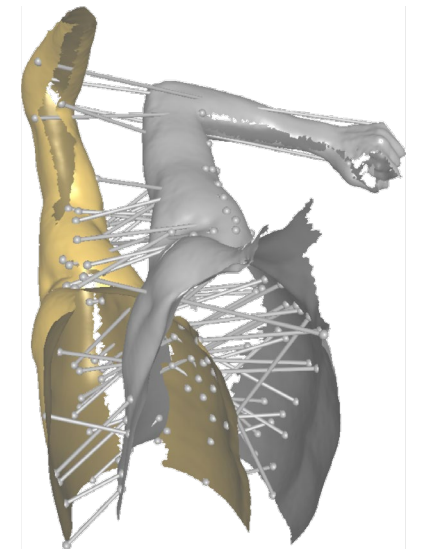
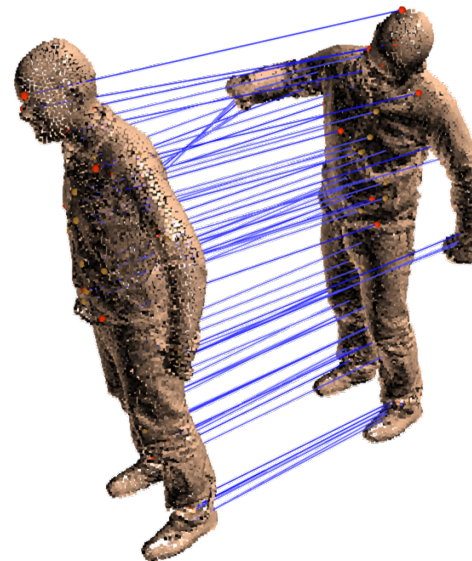
Maps vs. Distances/Similarities Networks vs. Graphs



Persistence of correspondences

What Makes a Map Good?

- 1st order descriptor or feature preservation (e.g., corners go to corners)
- 2nd order attribute preservation (e.g., Euclidean or geodesic distances)
- Smoothness or continuity
- Respect for “internal structure” (symmetries, etc)
- Semantic correctness may still be elusive



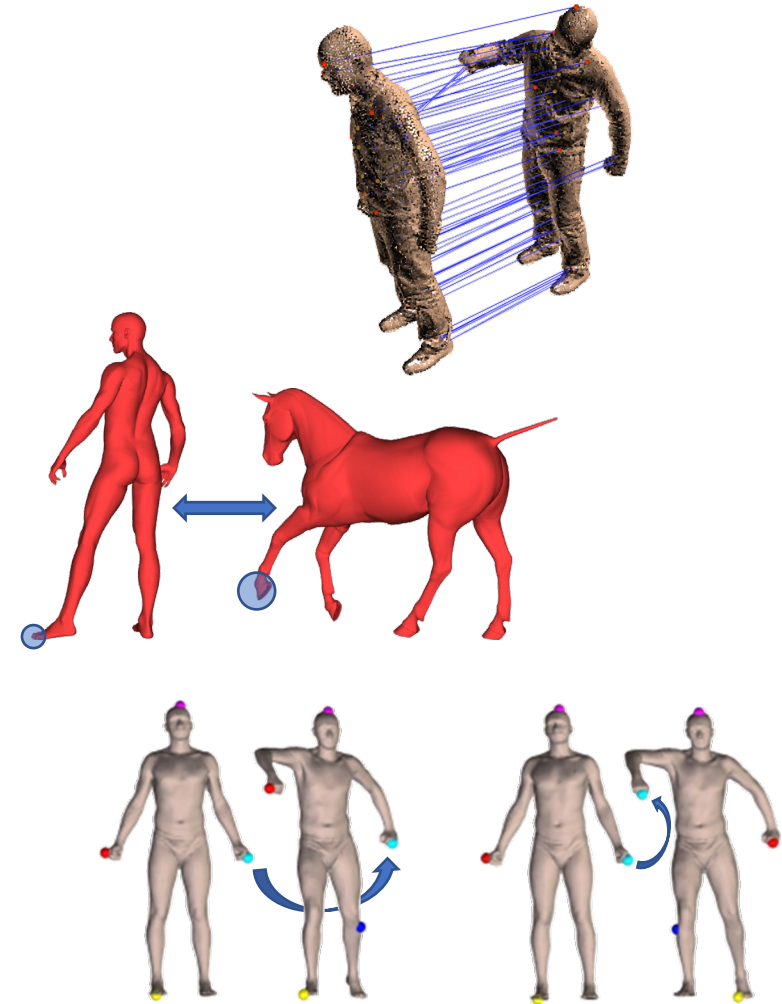
Problems and Issues



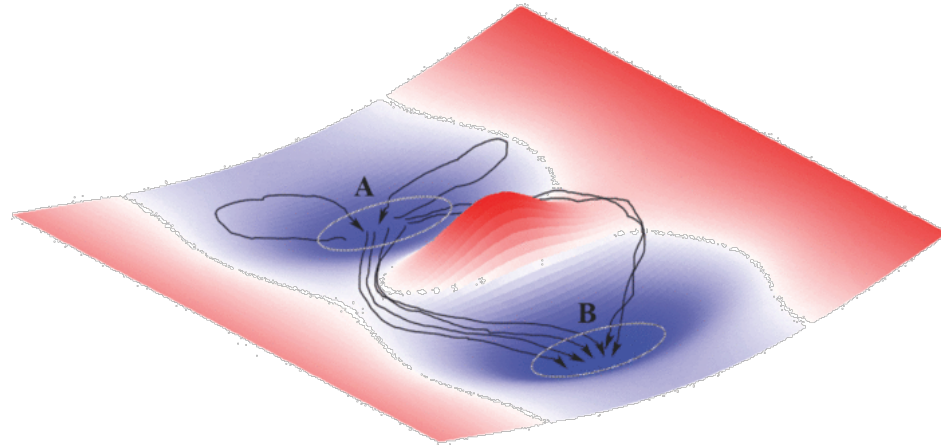
Symmetry, ambiguity, scale, bad data

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
 - 2nd order attribute preservation leads to NP-hard quadratic assignment problems

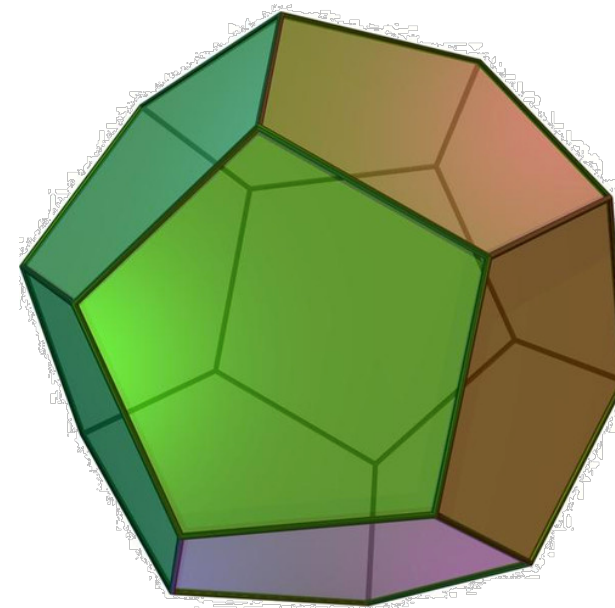


Non-Convex, Combinatorial Optimization



multiple minima

NP-hard quadratic assignments

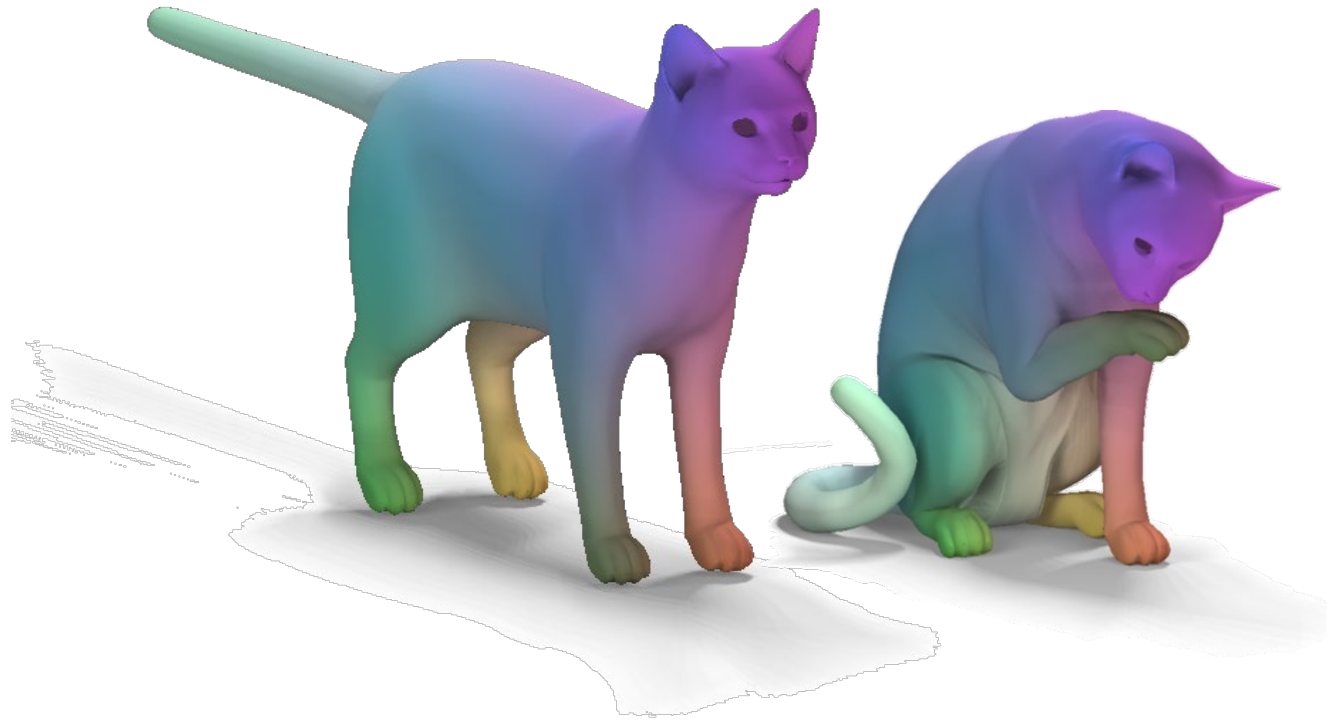


$n!$ permutations

Symmetry, ambiguity, scale, bad data

A Potential Way Out

Find alternative **representation** more amenable to optimization



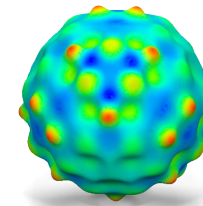
Redefine the notion of map

Technical Approach: Function Spaces and Functional Maps

Shapes: From a Particle View to a Wave View ...

A Dual View: Functions and Operators

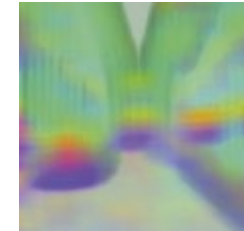
- Functions on data
 - Properties, attributes, descriptors, part indicators, etc.
 - But also beliefs, opinions, etc
- Operators on functions
 - Maps of functions to functions
 - Laplace-Beltrami operator on a manifold



Curvature



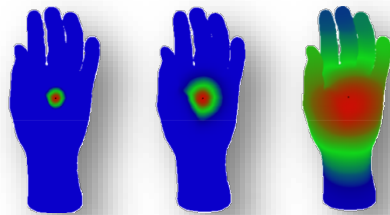
Parts



SIFT flow, C. Liu 2011

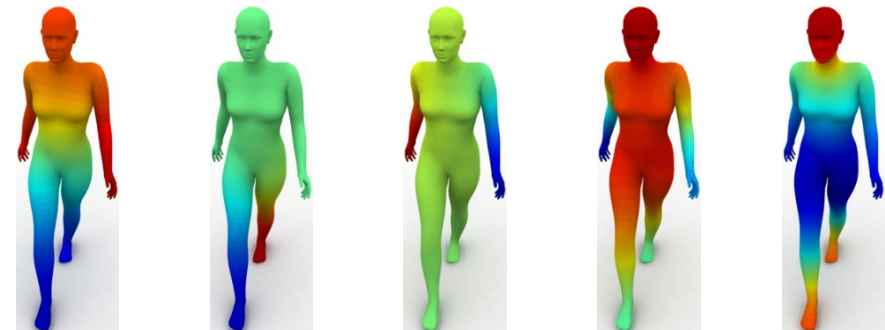
M

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



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

heat diffusion

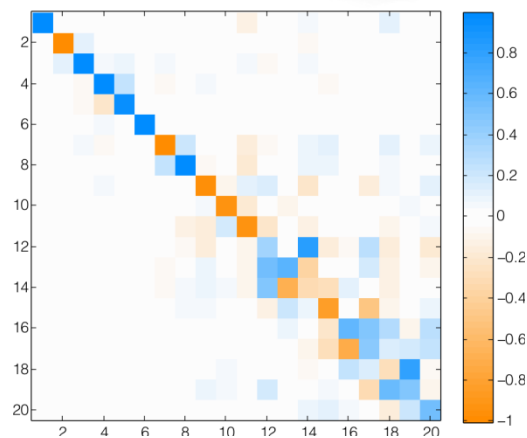


Laplace Beltrami eigenfunctions

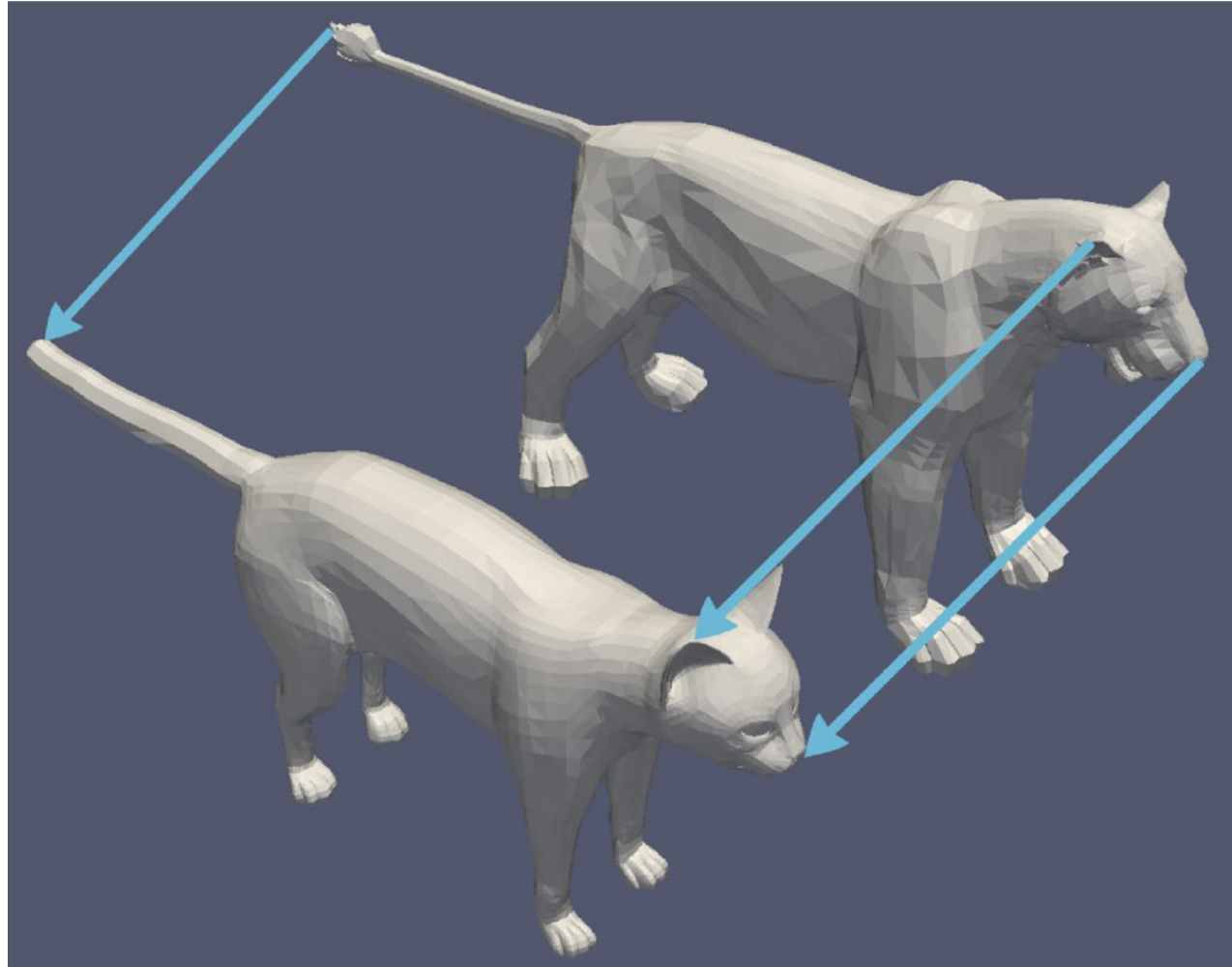


Functional Maps (a.k.a. Operators)

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

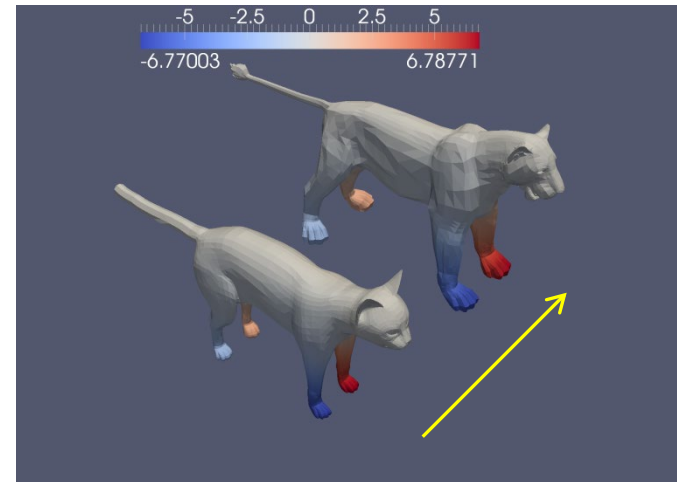
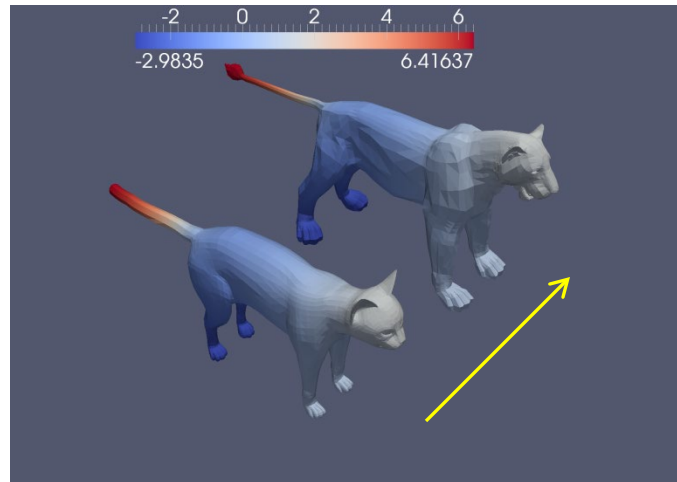
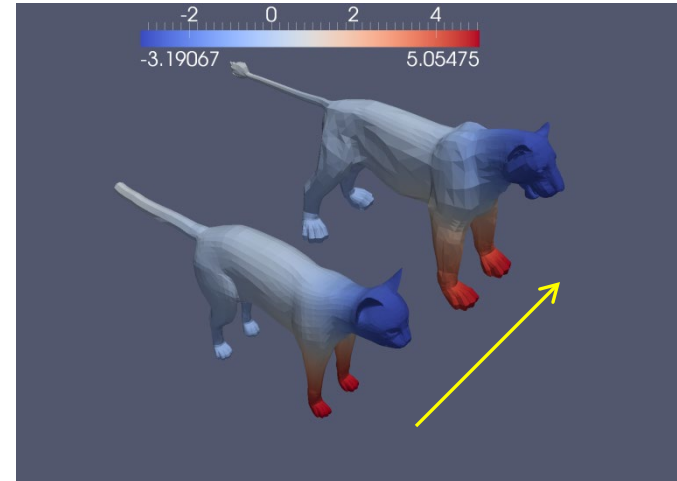
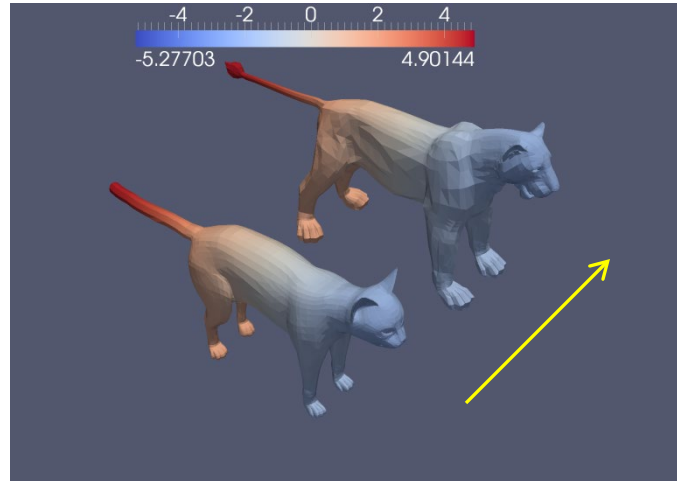


Starting from a Regular Map ϕ



$\phi: \text{lion} \rightarrow \text{cat}$

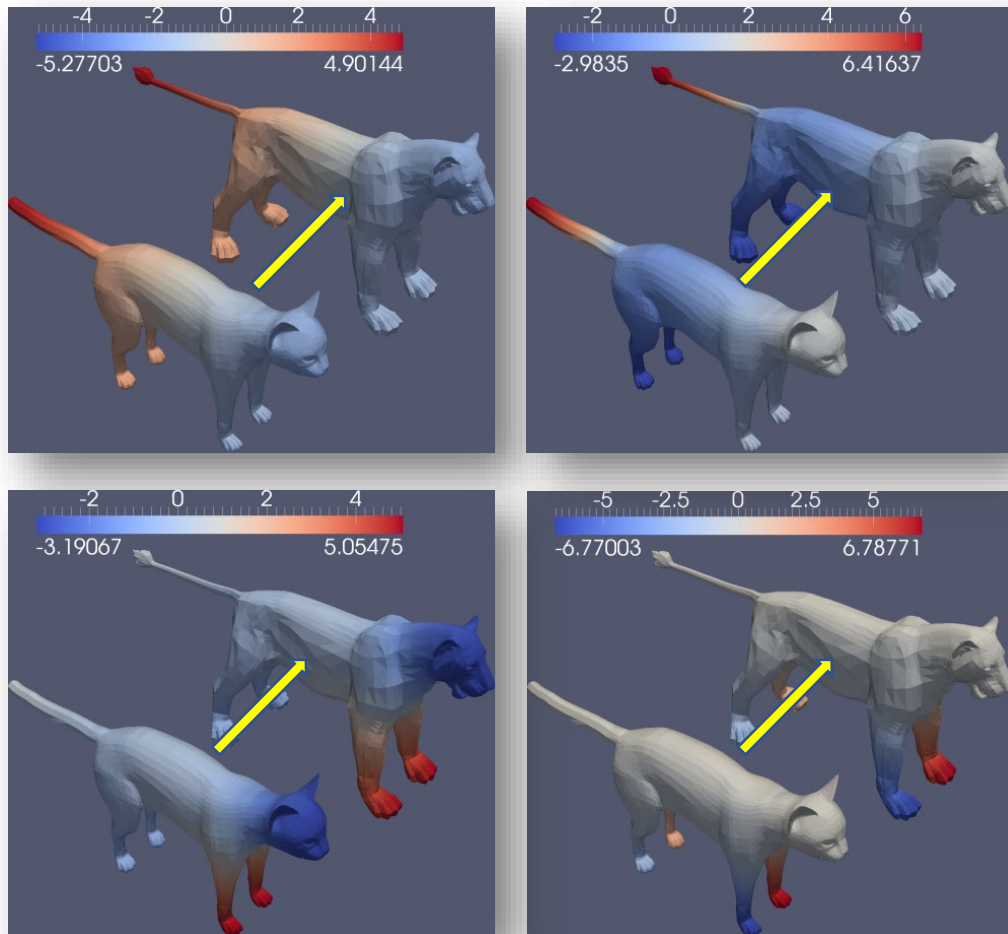
Attribute Transfer via Pull-Back



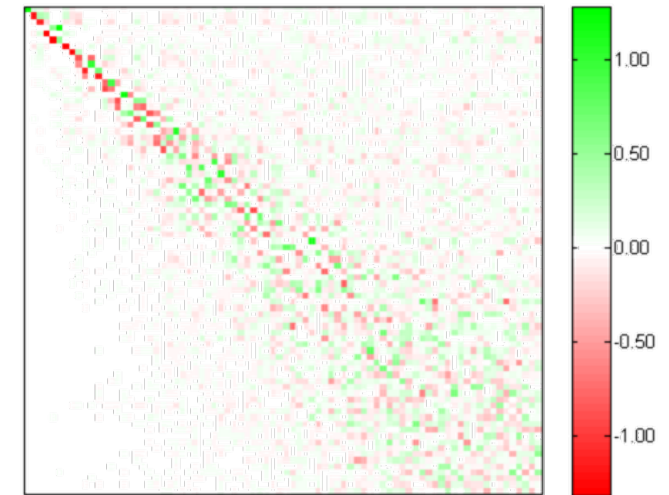
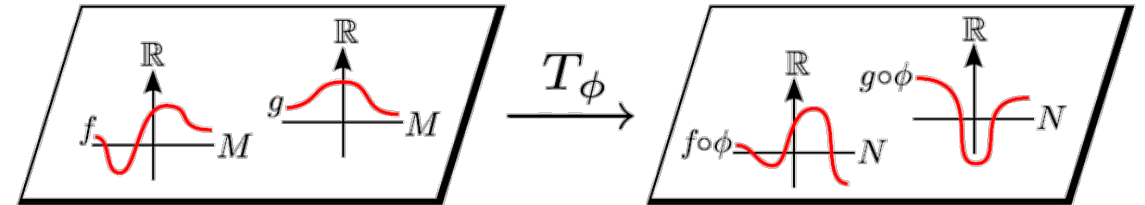
$T_\phi: \text{cat} \rightarrow \text{lion}$

A Contravariant Functor

from cat to lion



Functions on cat are transferred to lion using T_ϕ



T_ϕ is a linear operator (matrix)

$$T_\phi : L^2(\text{cat}) \rightarrow L^2(\text{lion})$$

Functional Map

$$\phi : M \rightarrow N$$

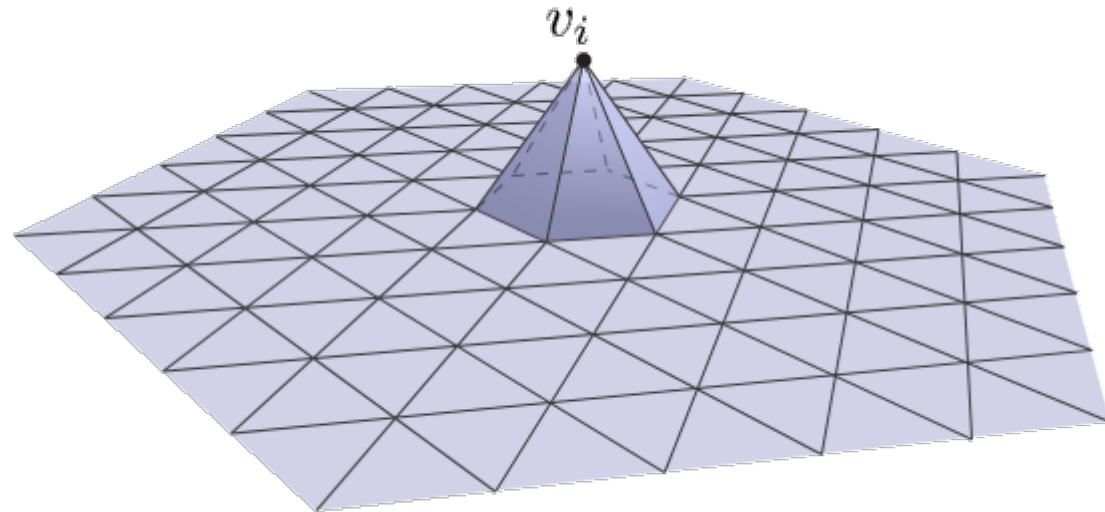
$$T_\phi : L^2(N) \rightarrow L^2(M)$$

Dual of a
point-to-point map

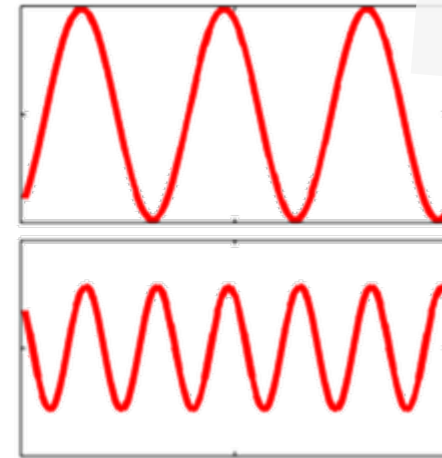
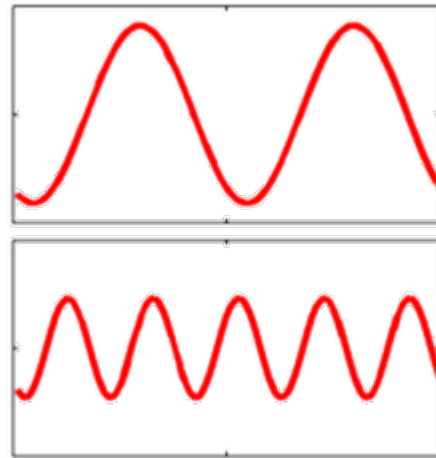
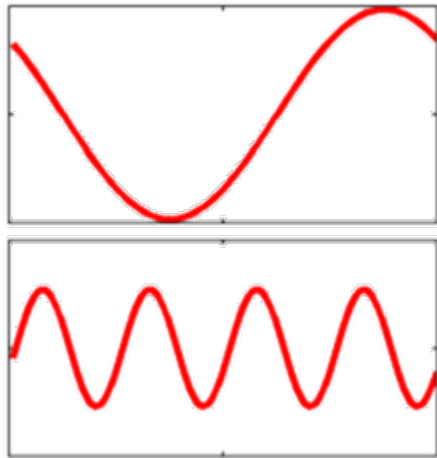
Encoding Functions: Bases for a Function Space

Point basis
Finite-element basis

Local bases



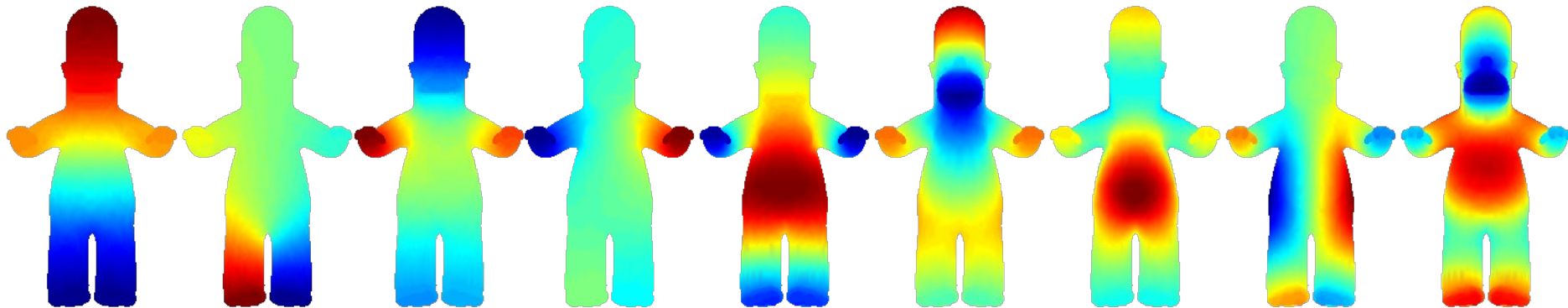
Hierarchical Bases for a Function Space



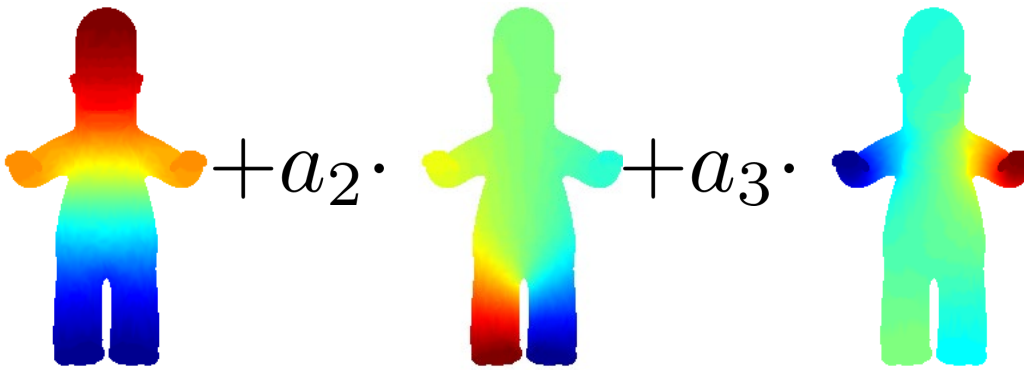
Fourier

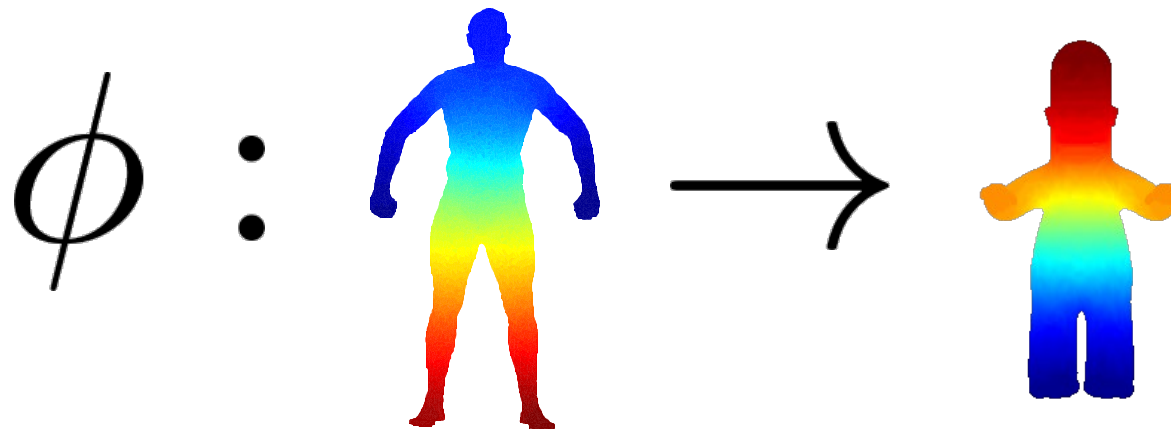
Laplace-Beltrami

global support



Application of Basis

$$f(x) = a_1 \cdot \text{Homer} + a_2 \cdot \text{Bart} + a_3 \cdot \text{Lisa} + \dots$$




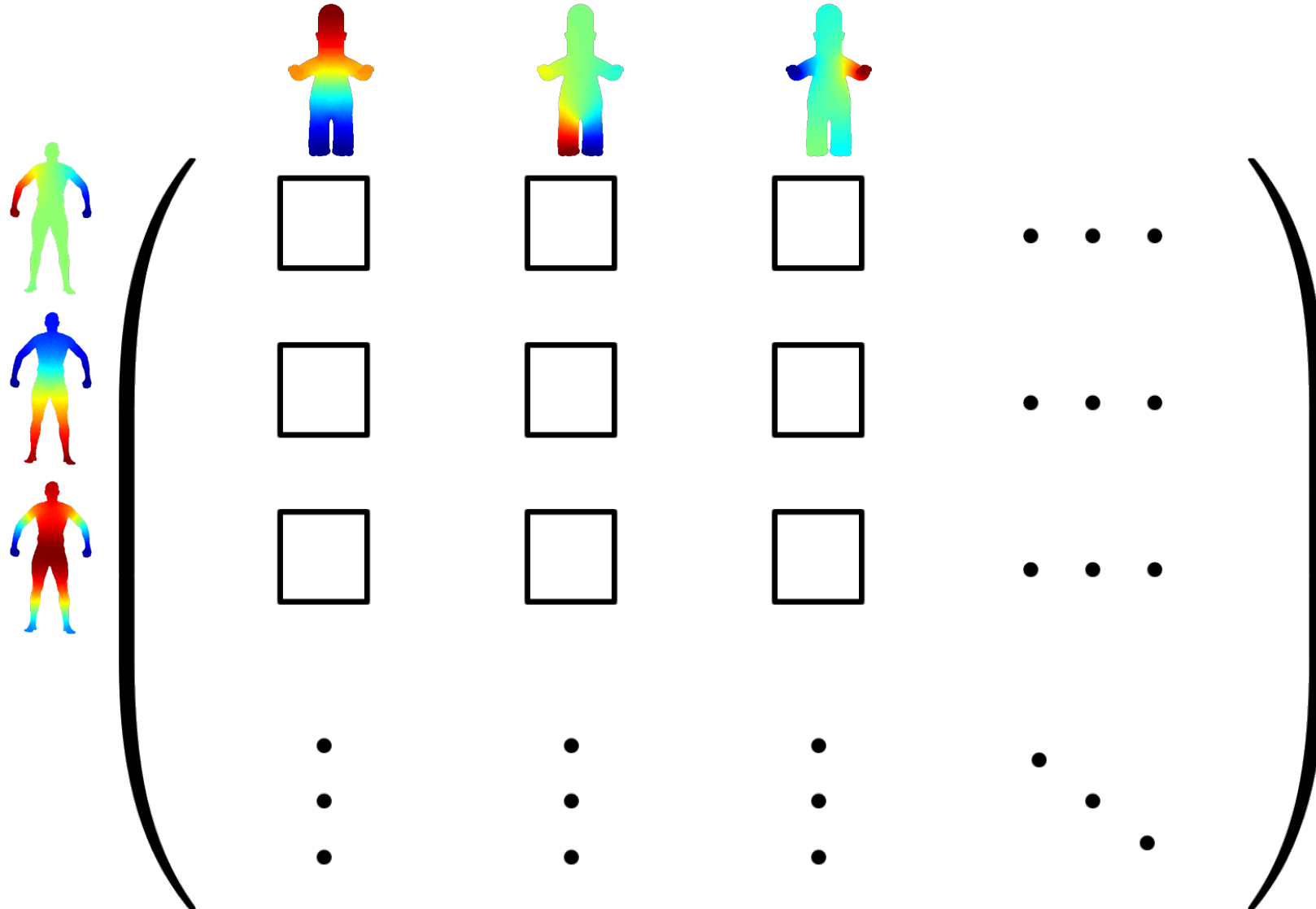
Application of Basis

$$T_\phi[f](x) = T_\phi[a_1 \cdot \text{img}_1 + a_2 \cdot \text{img}_2 + a_3 \cdot \text{img}_3 + \dots]$$

$$= a_1 T_\phi[\text{img}_1] + a_2 T_\phi[\text{img}_2] + a_3 T_\phi[\text{img}_3] + \dots$$

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 $T : M \rightarrow N$, 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:

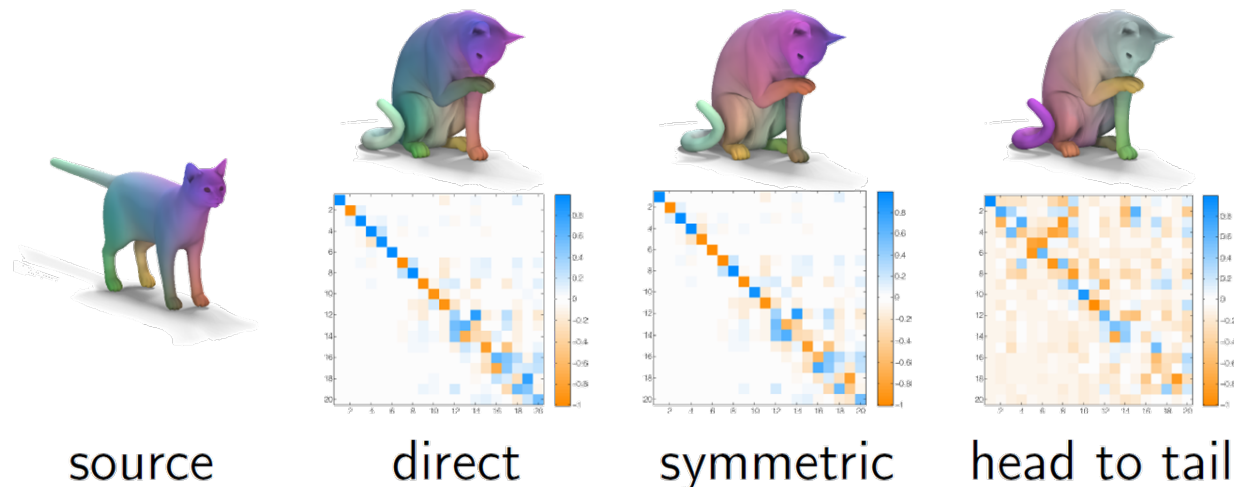
$$\mathbf{b} = C\mathbf{a}$$

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

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

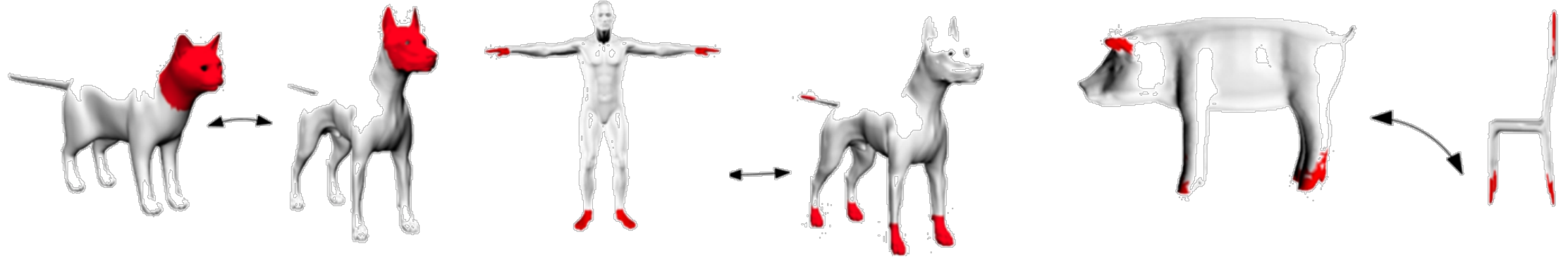
Maps as Linear Operators

- 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 $f : M \rightarrow \mathbb{R}$ and $g : N \rightarrow \mathbb{R}$ to correspond. Then, C must satisfy:

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

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



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 $f : M \rightarrow \mathbb{R}$ and $g : N \rightarrow \mathbb{R}$ to correspond. Then, C must be s.t.

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

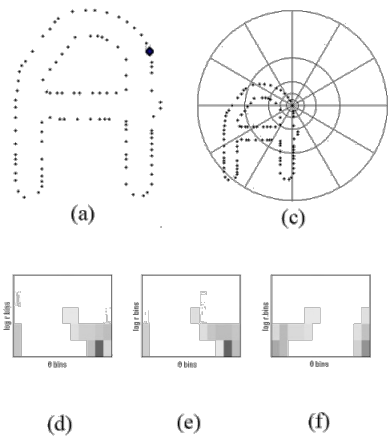
Function preservation constraint is quite general and includes:

- 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”

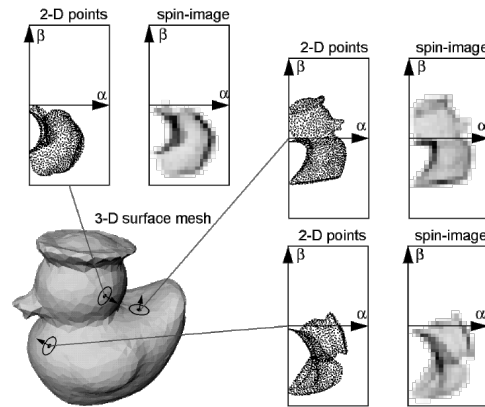
Plenty of Functions: Descriptors for Points and Parts

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



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

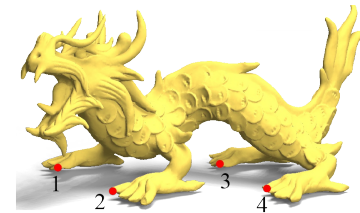
Rigid invariance
(extrinsic)



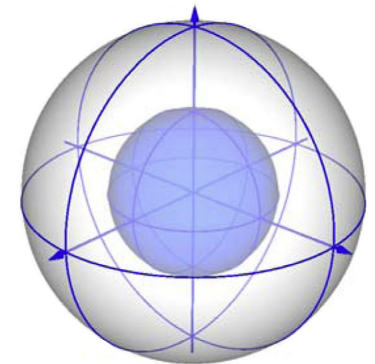
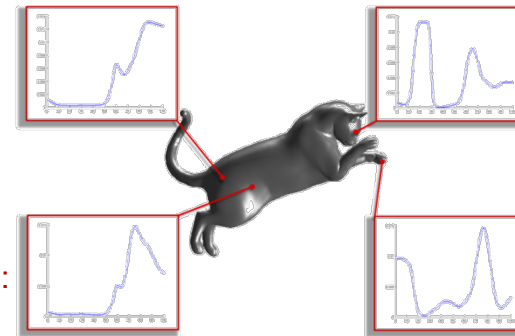
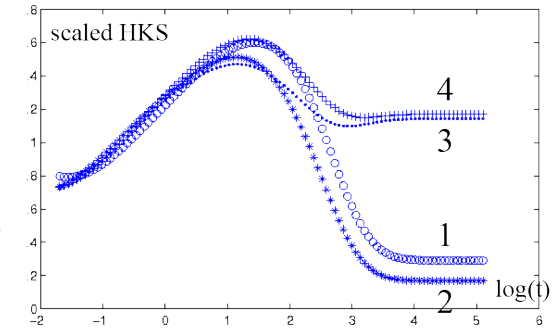
Spin Images:
[Johnson, Hebert '99]

Isometric invariance
(intrinsic)

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



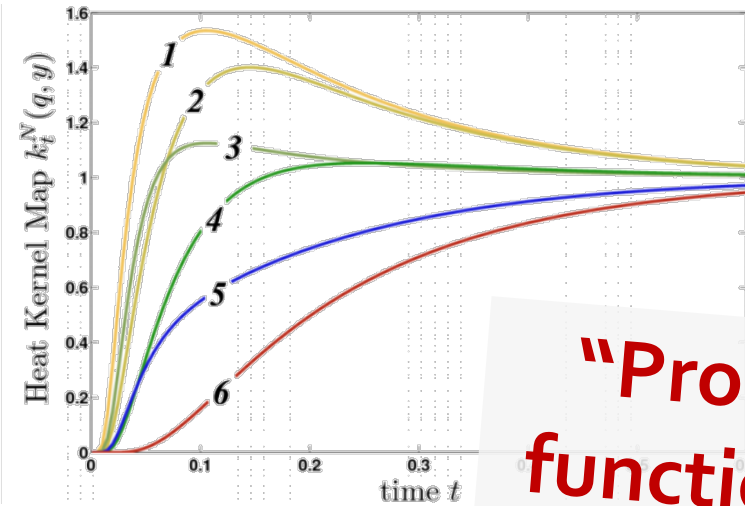
Heat Kernel Signatures (HKS):
[Sun, Ovsjanikov, G. '08]



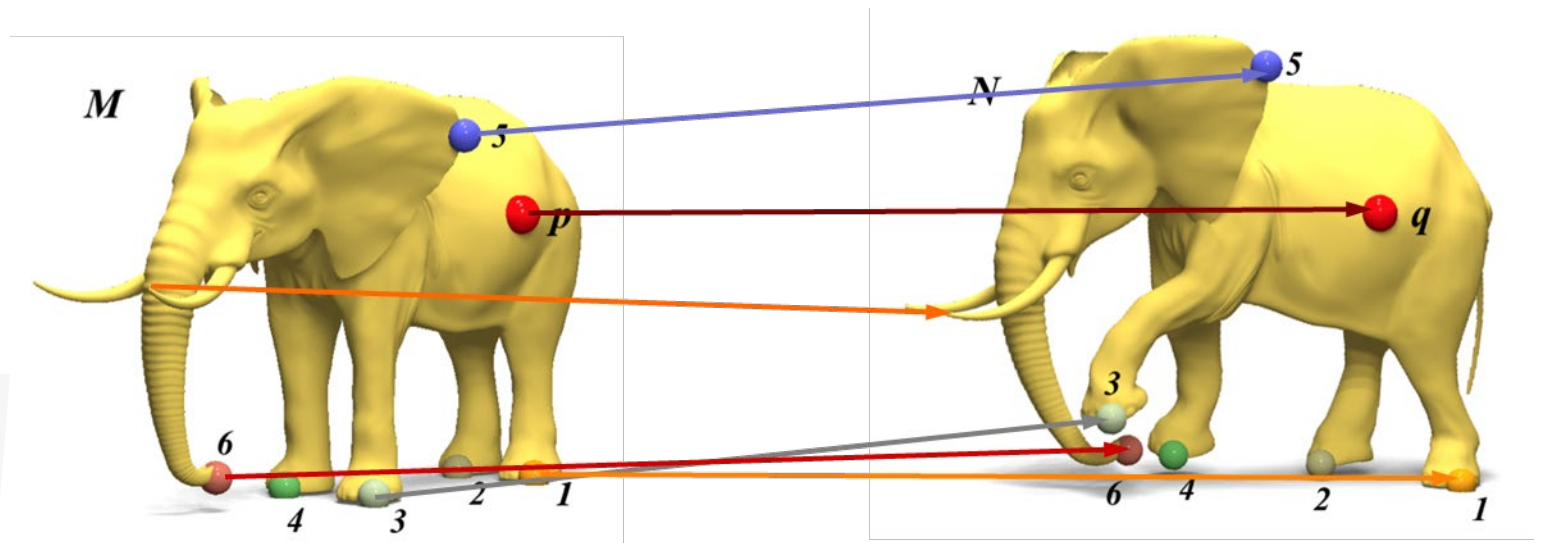
SHOT

Map Estimation

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



“Probe function”



Map from a linear solve

Commutativity Symmetry Regularization

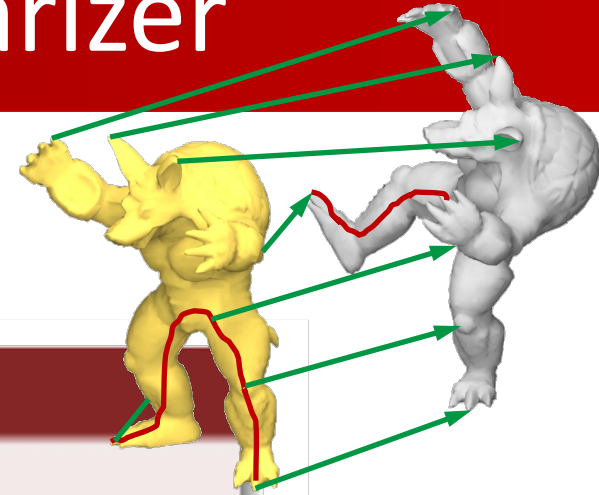
In addition, we can phrase an operator commutativity constraint: given two operators $S_1 : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(M, \mathbb{R})$ and $S_2 : \mathcal{F}(N, \mathbb{R}) \rightarrow \mathcal{F}(N, \mathbb{R})$

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

Thus: $CS_1 = S_2C$ or $\|CS_1 - S_2C\|$ 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

Δ_1 Laplacian on Shape 1
 Δ_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*:

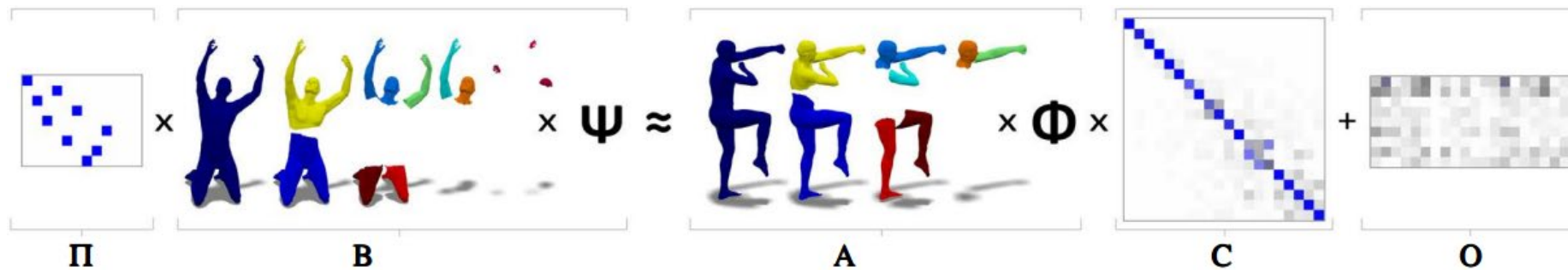
$$C^T C = I$$

Rotations/reflections in functions space

Sparcity in a Localized Basis

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

Sum of Euclidean norms of cols

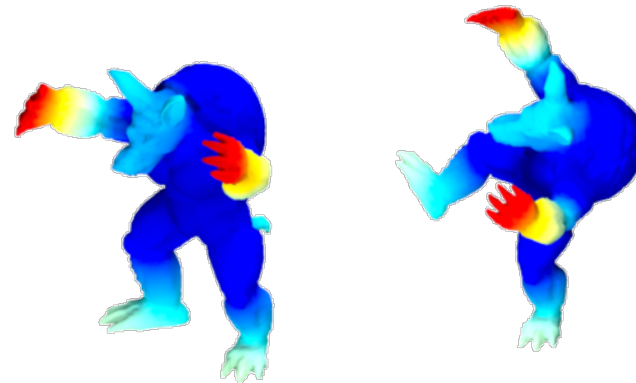


Sparse Modeling of Intrinsic Correspondences (Pokrass, Bronstein², Sprechmann, Sapiro)

Basic FMaps Pipeline

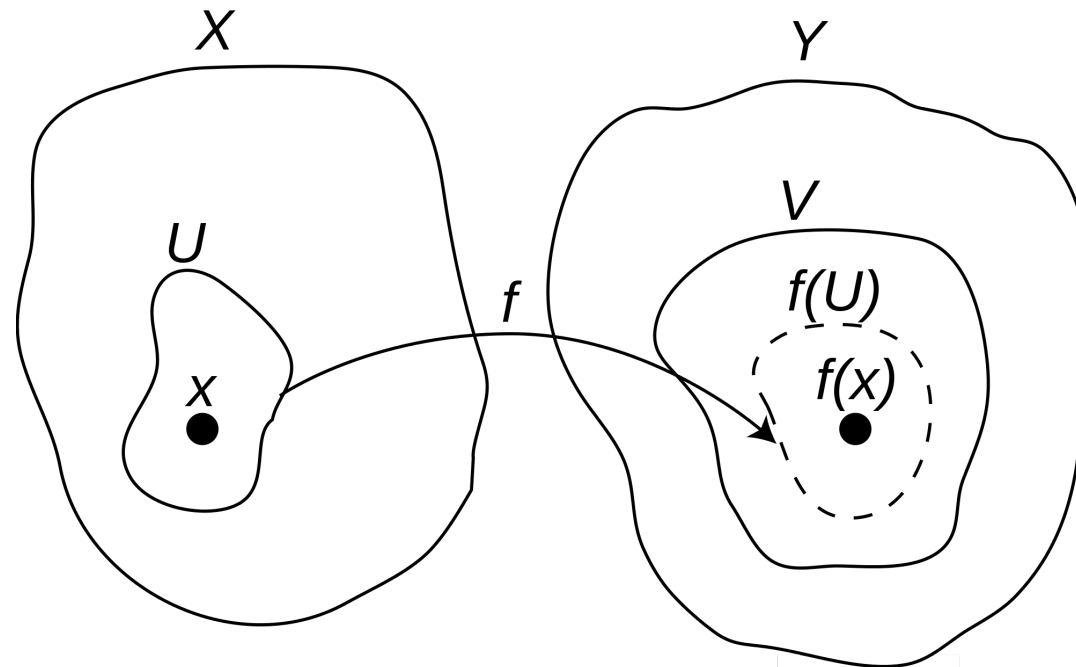
Given a pair of shapes: \mathcal{M}, \mathcal{N}

1. Compute the first k ($\sim 80-100$) 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 $C_{\text{opt}} = \arg \min_C \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2 + \dots$
 $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$: diagonal matrices of eigenvalues of LB operator
4. Convert the functional map C_{opt} to a point to point map T .



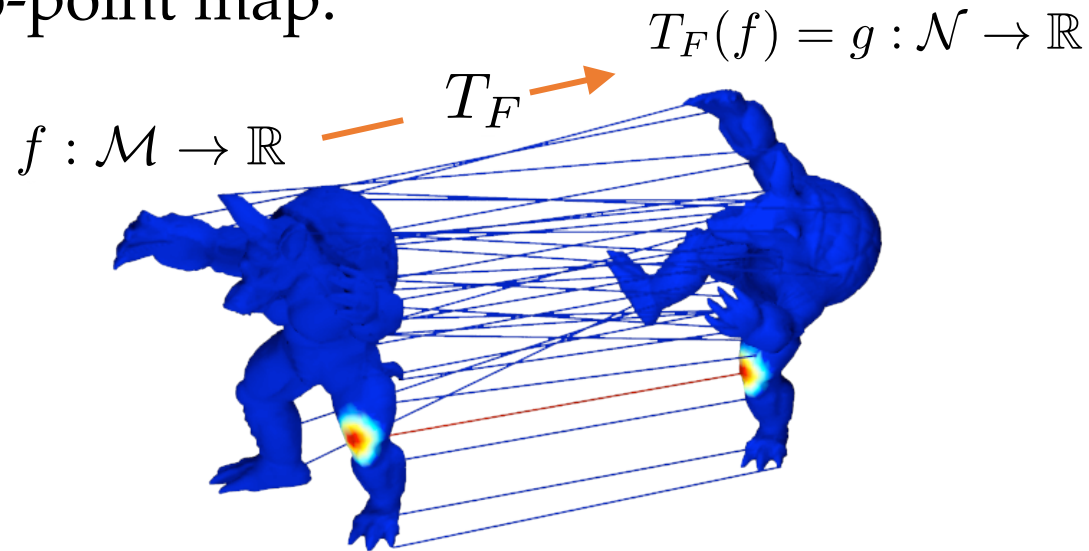
Side Comment: Map Continuity

- 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 a point-to-point map.

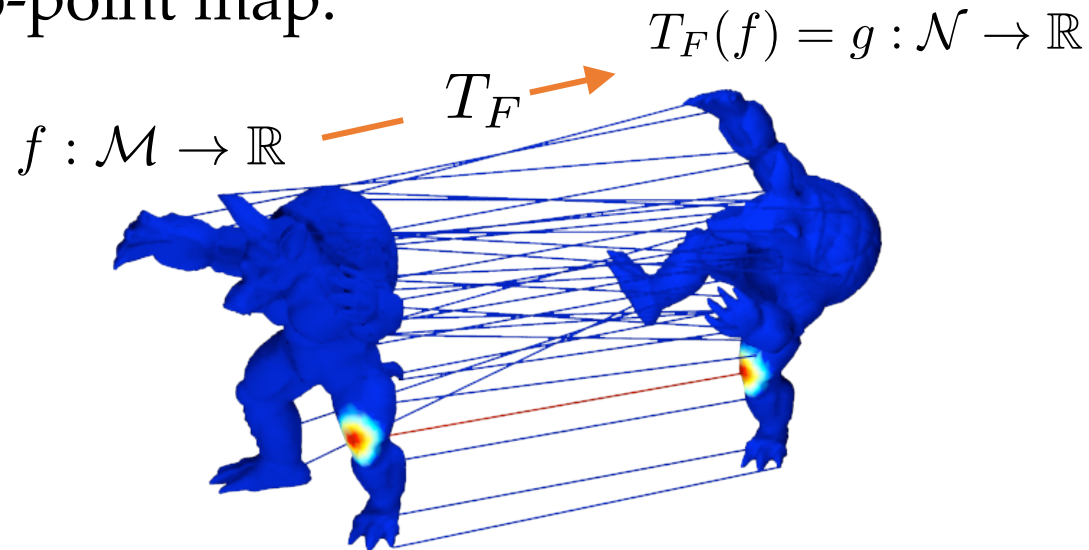


Option 1: declare $T(x) = \arg \max_y \Phi_{\mathcal{N}} C \delta_x$

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 a point-to-point map.

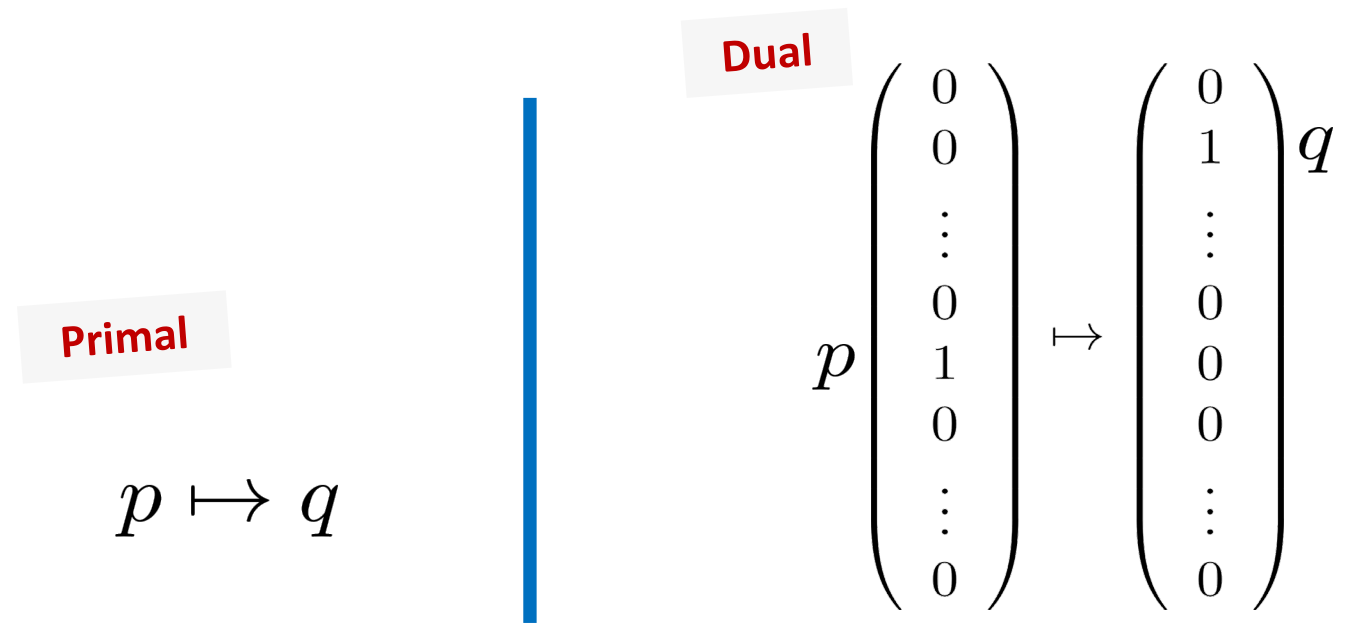


Option 2: declare $T(x) = \arg \min_y \|\delta_y - C\delta_x\|_2$

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



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

From Functional to Point-to-Point Maps

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

$$C\Phi_M^T \leftrightarrow \Phi_N$$

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:

$$C^T C = Id$$

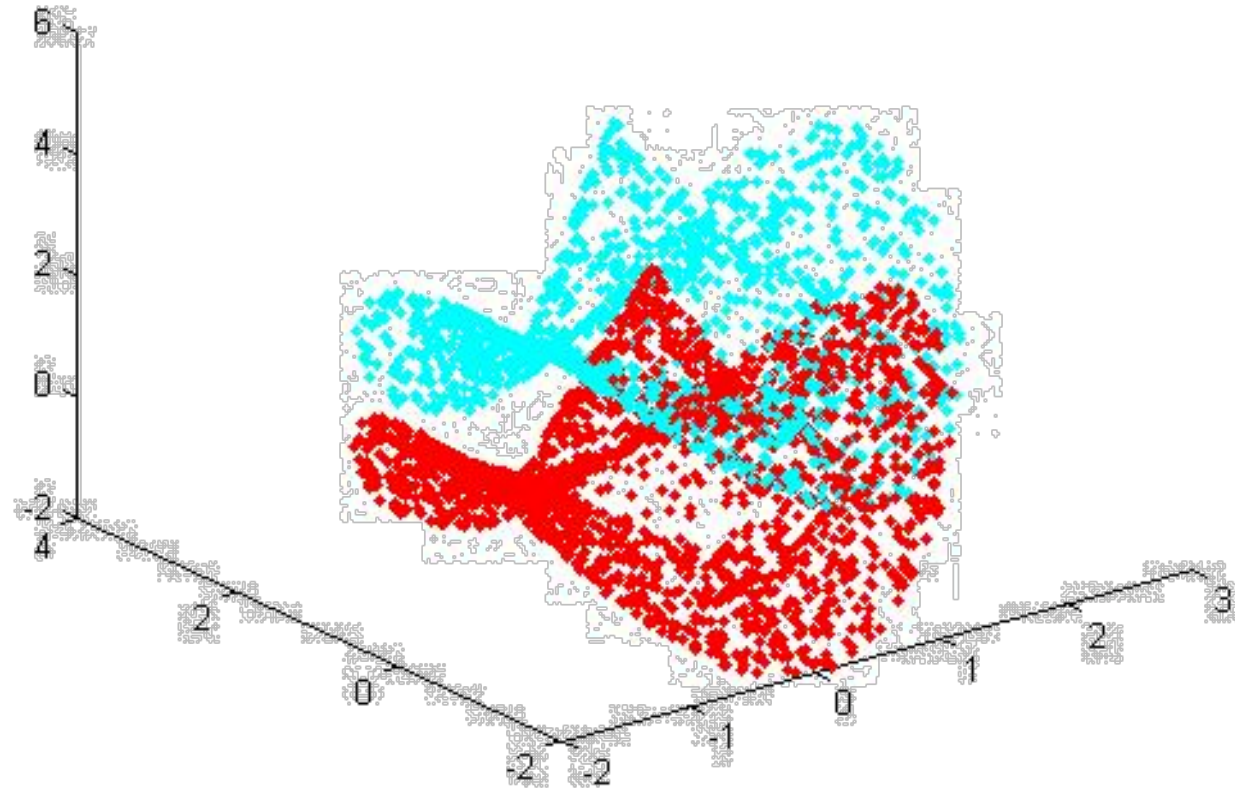
Option: use post-processing to enforce this constraint.

Iterate:

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

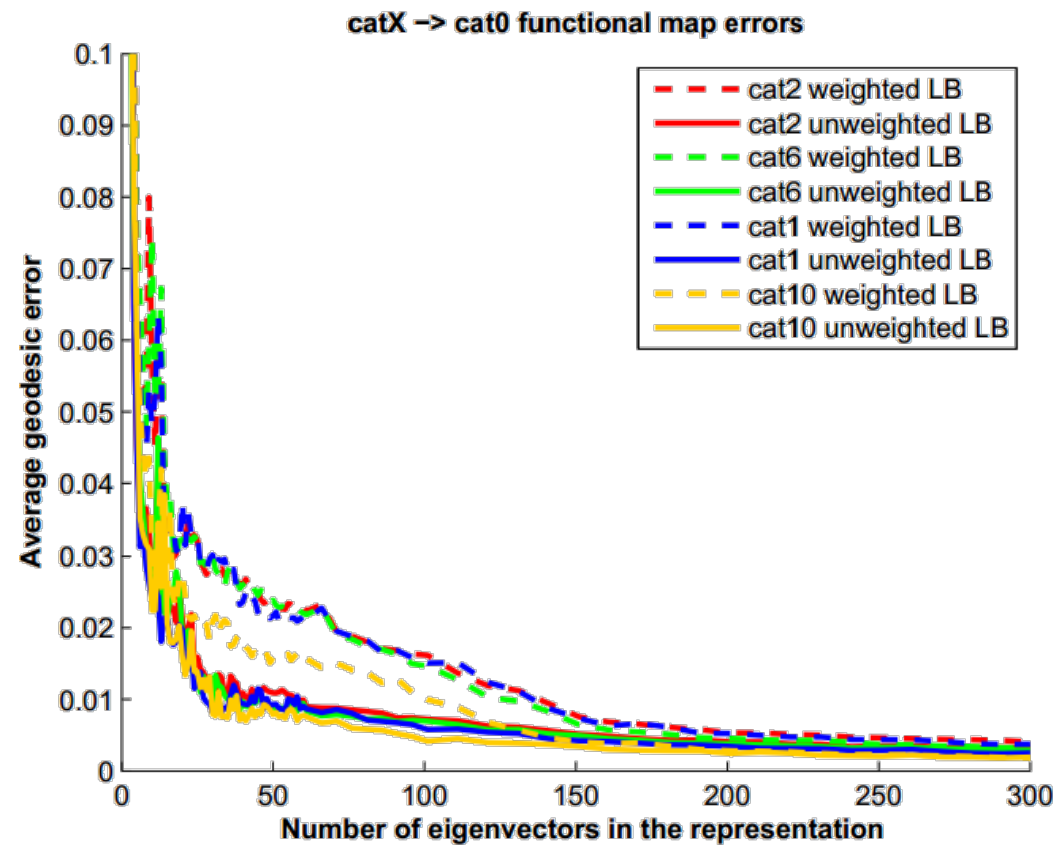
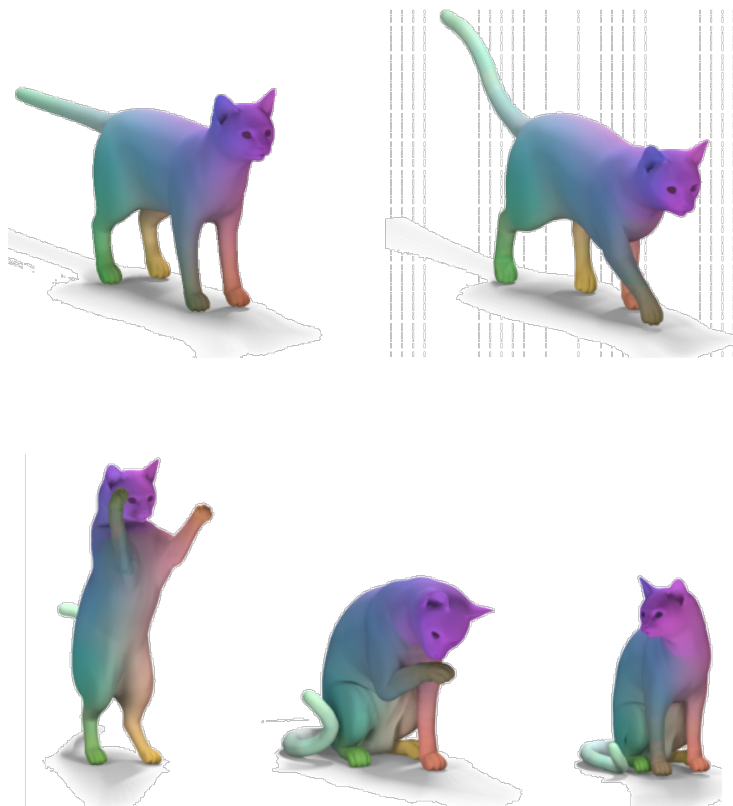
Exactly the same objective as ICP, but in higher dimension. Can use the same method!

From Functional to Point-to-Point Maps



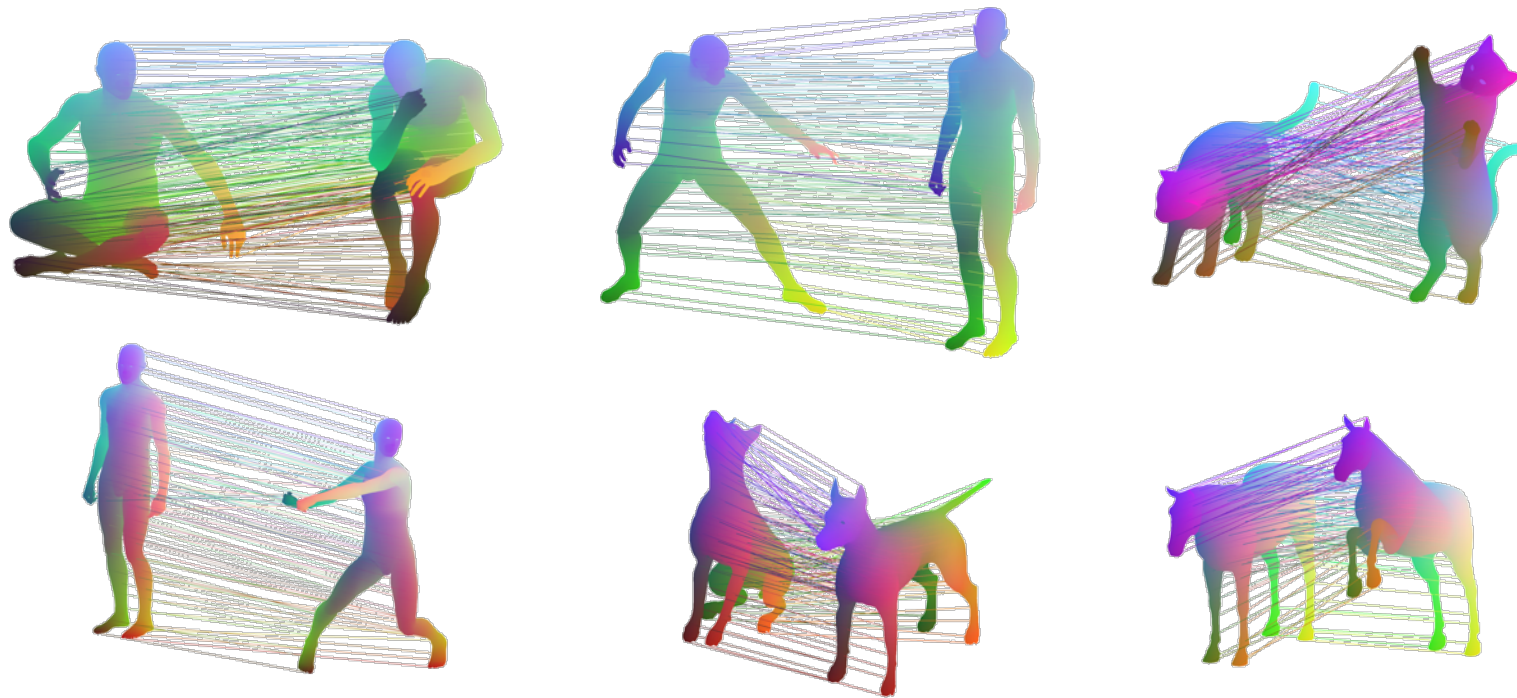
ICP in Function Space!

Ground Truth Comparison



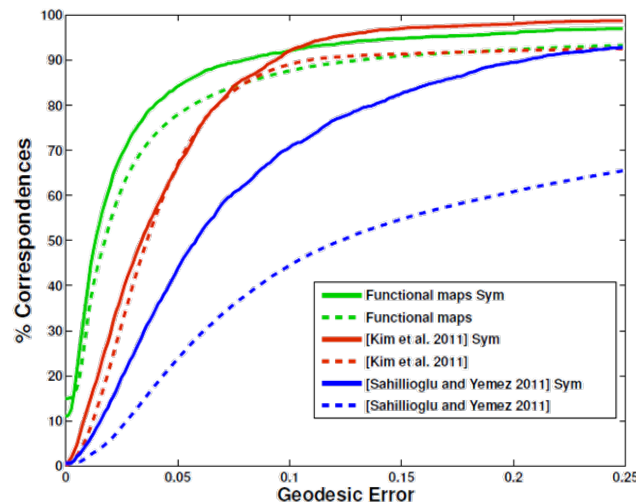
Results

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

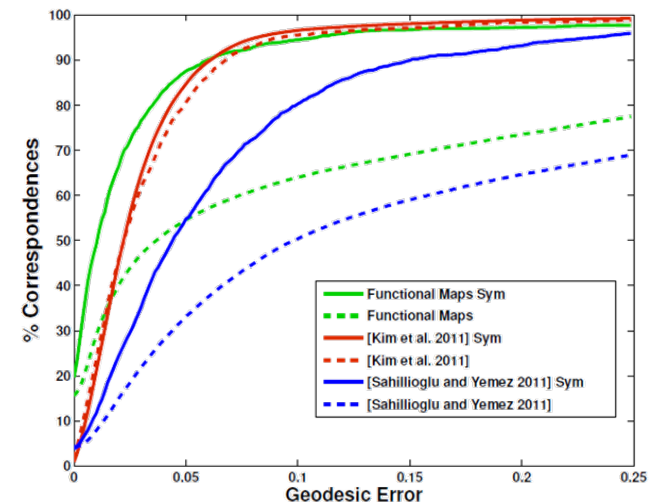


Map Estimation Quality

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



SCAPE



TOSCA

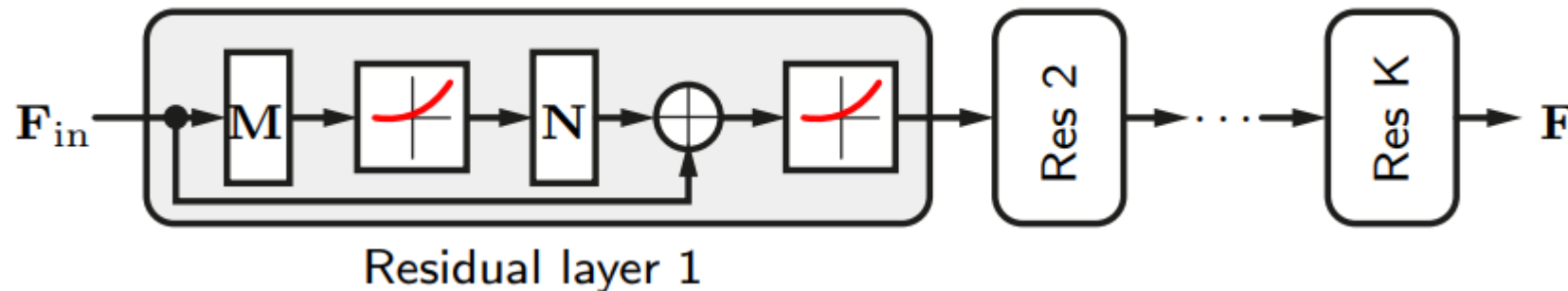
Roughly 10 probe functions + 1 part correspondence

Much Follow-On Work on FMaps

Deep Functional Maps

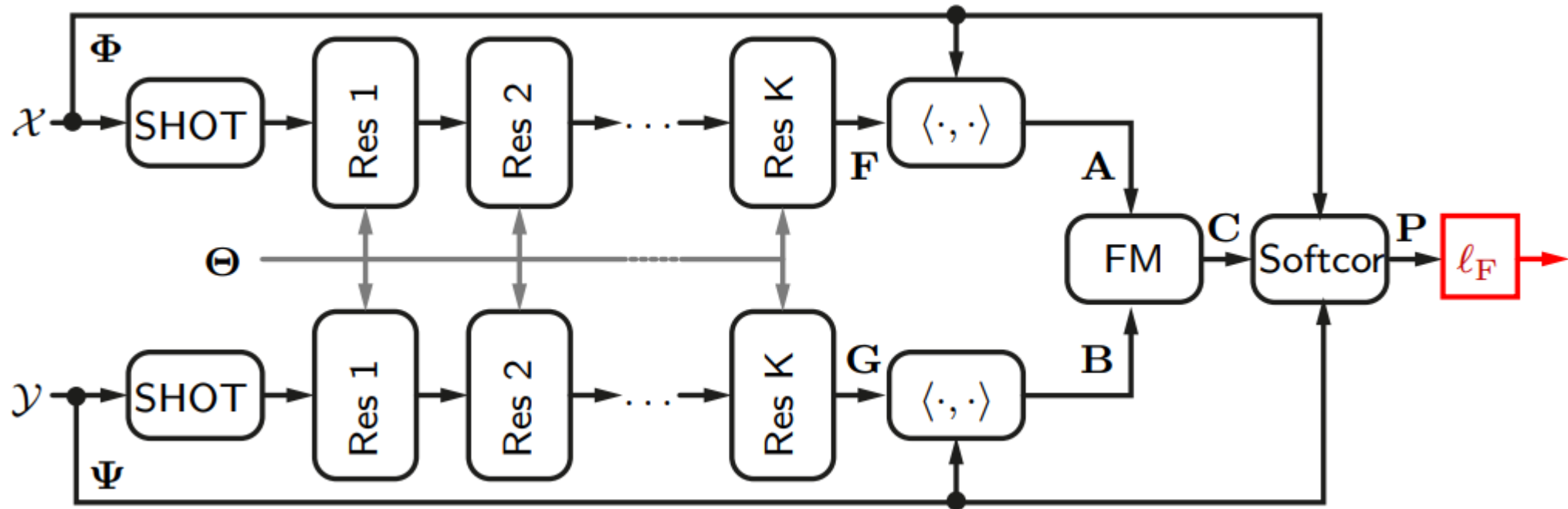
Deep Functional Maps

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



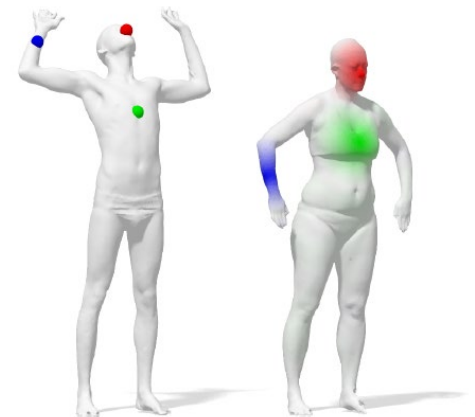
- **Input:** pointwise descriptor \mathbf{F}_{in} (e.g., 352-dimensional SHOT)
- N fully-connected residual layers with ELU activation parameterized by $\Theta = \{\mathbf{M}_1, \mathbf{N}_1, \dots, \mathbf{M}_K, \mathbf{N}_K\}$
- **Output:** transformed pointwise descriptor \mathbf{F}

FMNet



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

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



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:

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

Making Functional Maps Point-to-Point

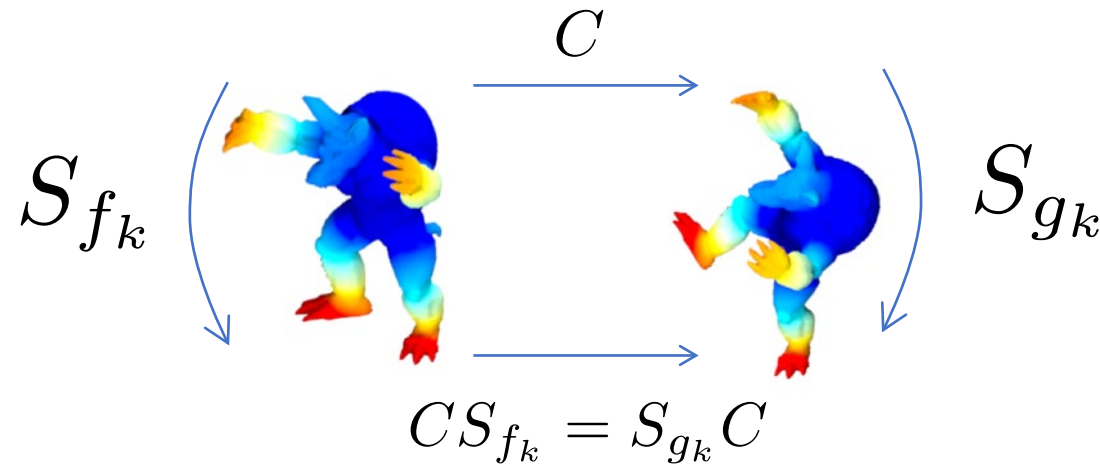
Approach

Represent descriptor functions via their action on functions through multiplication:

Functions as operators

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

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



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: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$

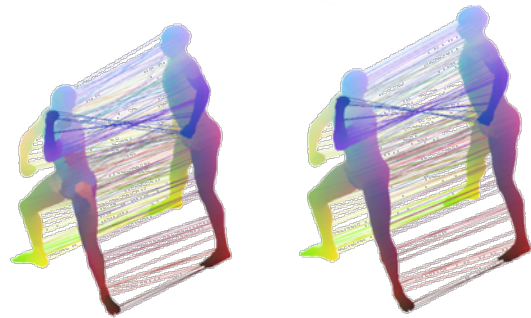
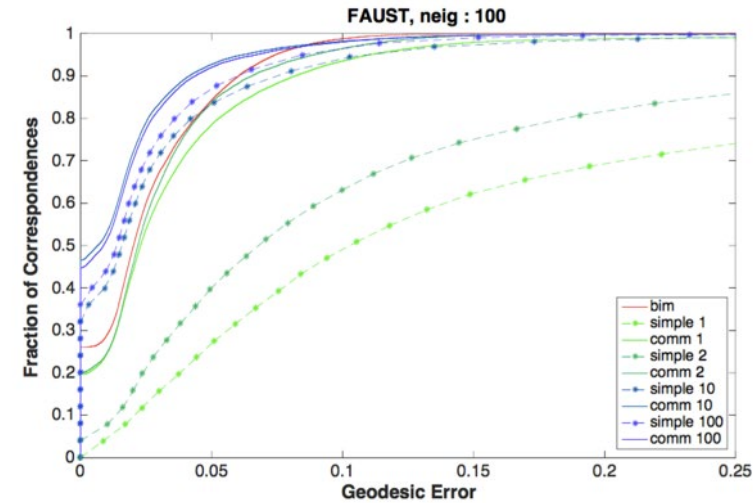
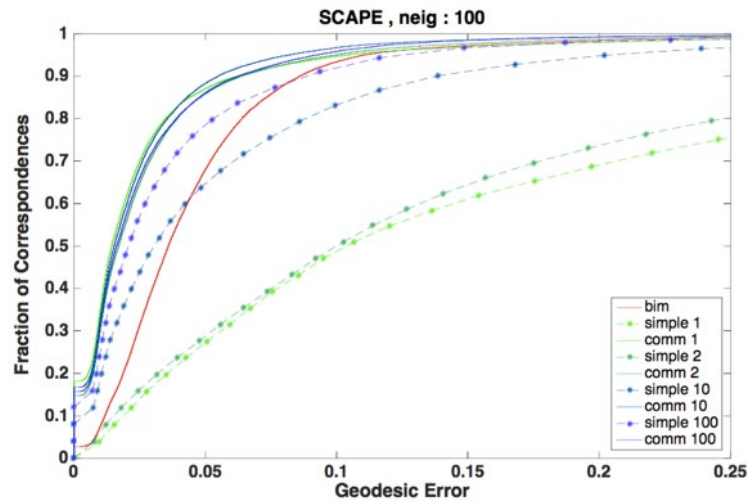
2. Compute descriptor functions (e.g., Gauss curvature) on
 \mathcal{M}, \mathcal{N} . Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$ as columns of : \mathbf{A}, \mathbf{B}

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

4. Convert the functional map C_{opt}
to a point to point map T .



Results with Extended Basic Pipeline



Incorporating multiplicative operators improves results significantly.

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

Segmentation Transport

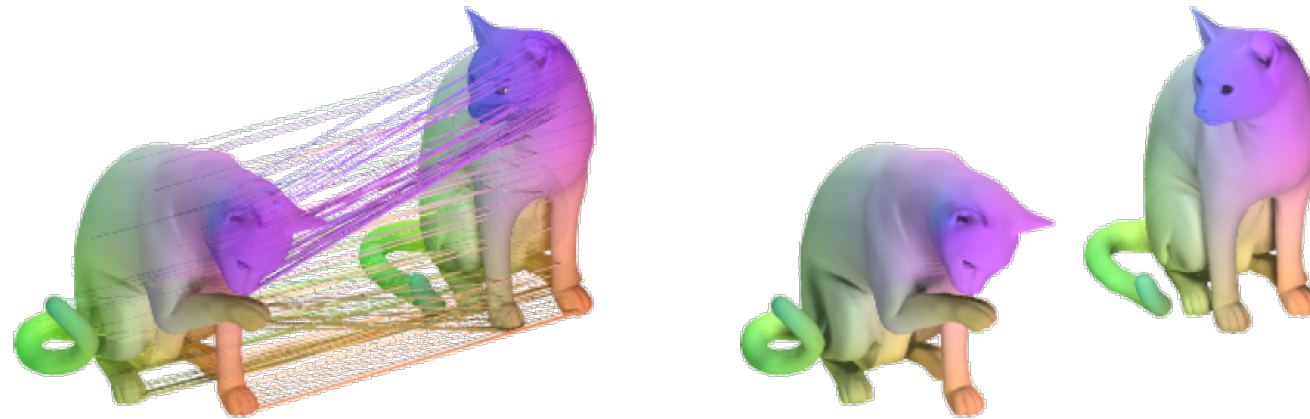
Application: Segmentation Transfer



Map Visualization

Map Visualization

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



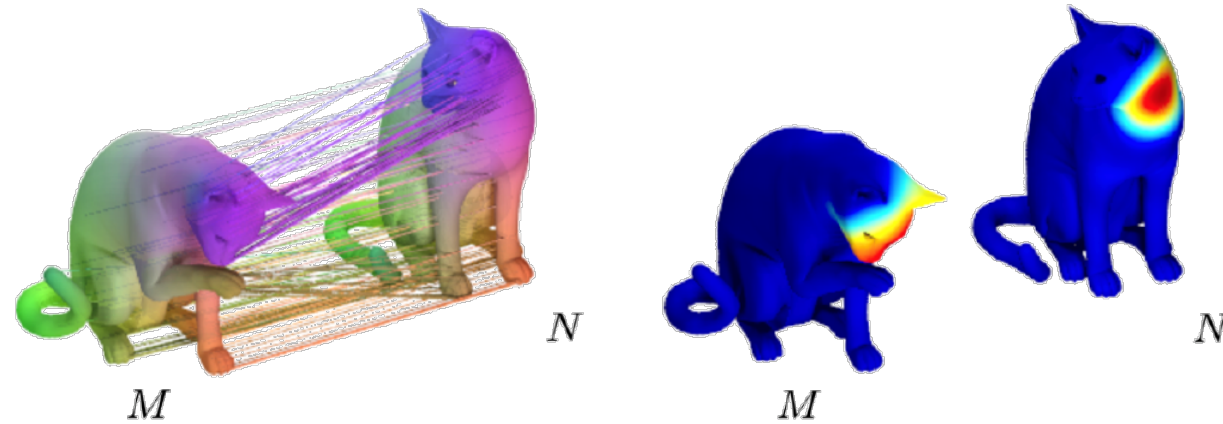
Common visualizations:

- 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 .

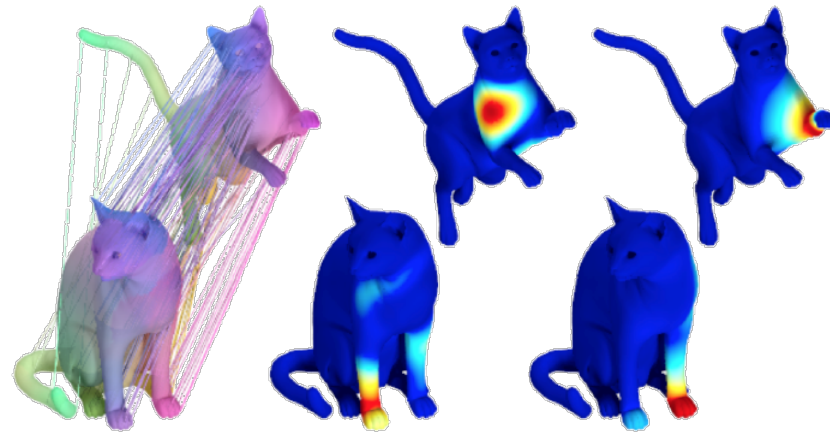
Map Visualization

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



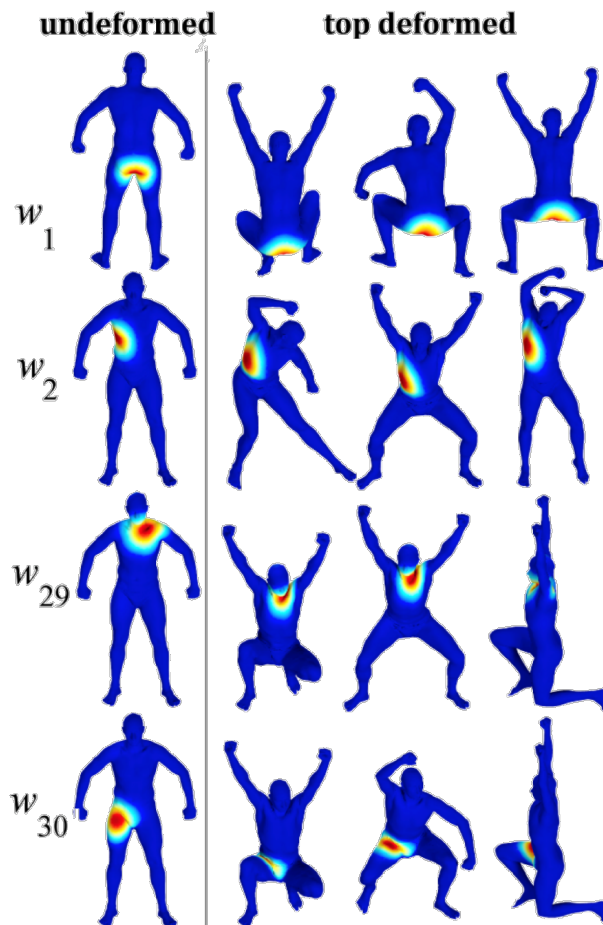
Map Visualization

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



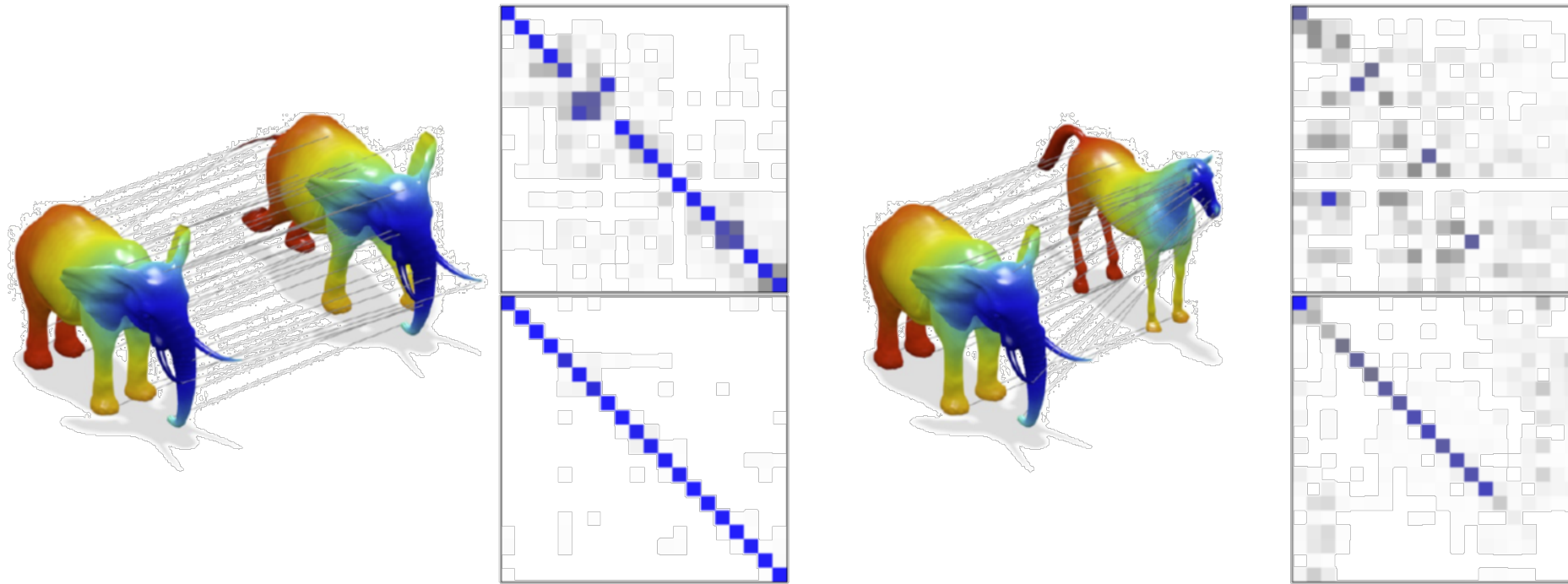
Multiple Shapes

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



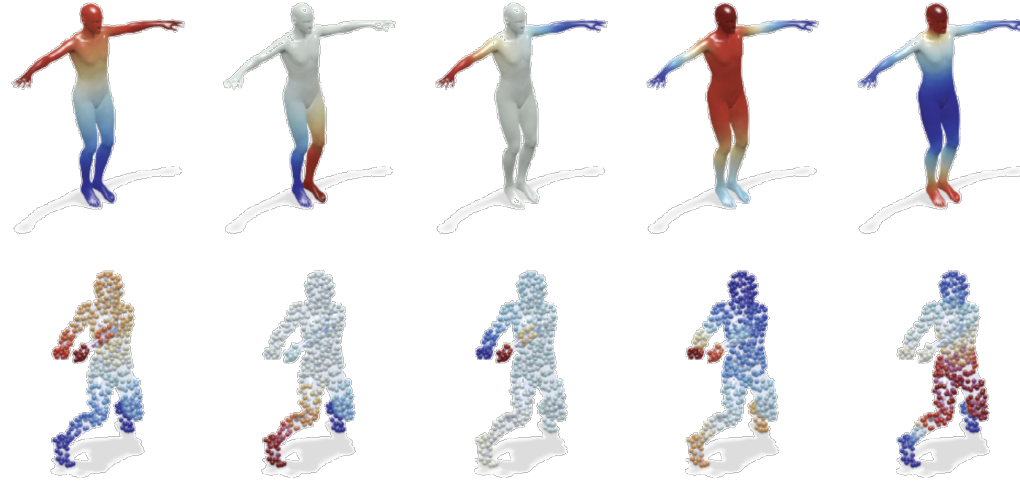
Diagonalizing the Map

Coupled Bases

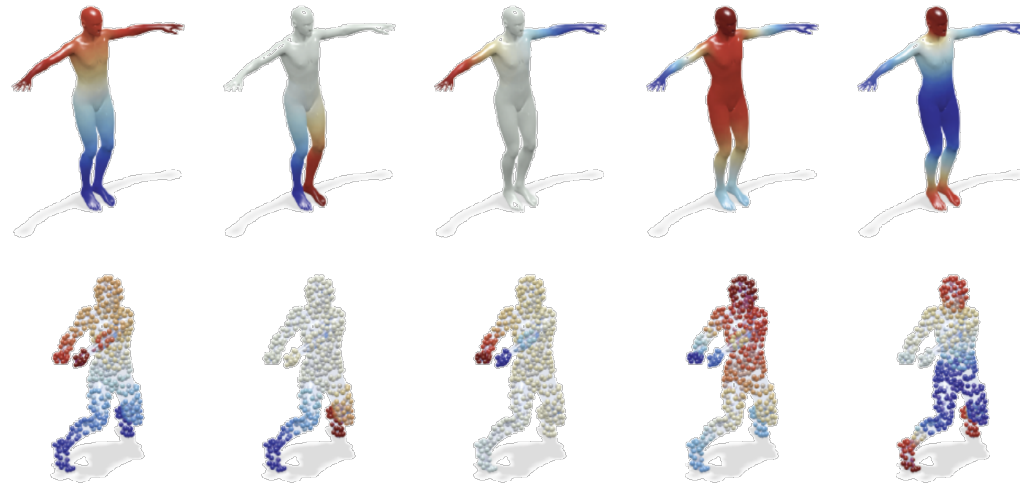


$$\begin{aligned} \min_{\mathbf{P}, \mathbf{Q}} \quad & \text{trace}(\mathbf{P}^\top \Lambda_X \mathbf{P}) + \text{trace}(\mathbf{Q}^\top \Lambda_Y \mathbf{Q}) + \mu \|\mathbf{P}^\top \mathbf{A} - \mathbf{Q}^\top \mathbf{B}\| \\ \text{s.t.} \quad & \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \mathbf{Q}^\top \mathbf{Q} = \mathbf{I} \end{aligned}$$

Coupled Bases



Laplacian eigenbases

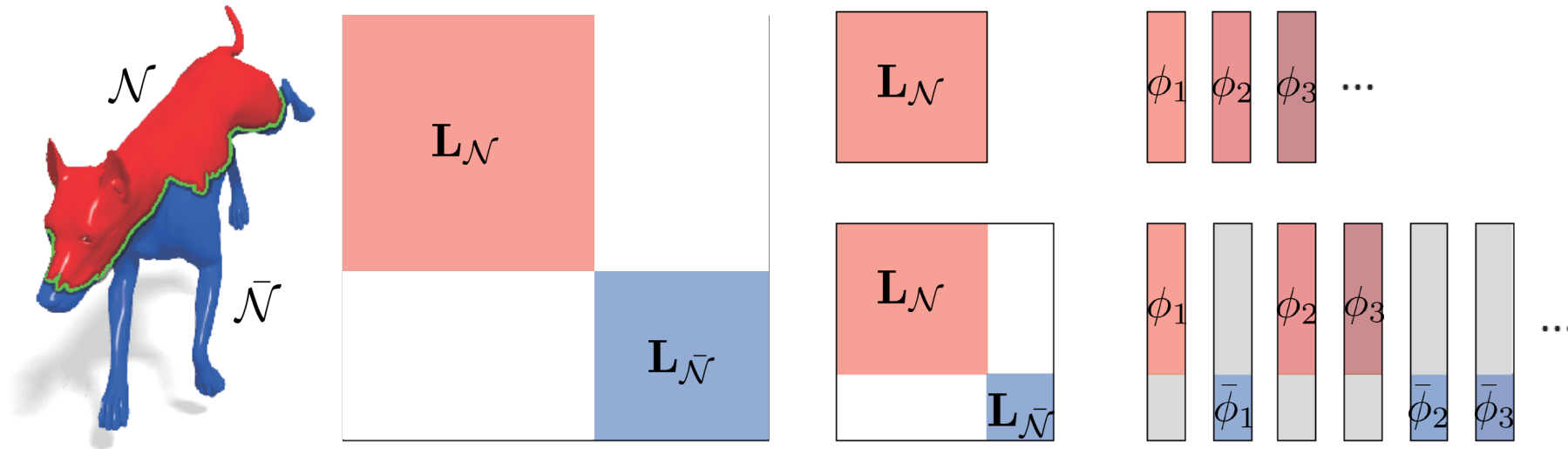


Coupled bases

Partial Functional Maps

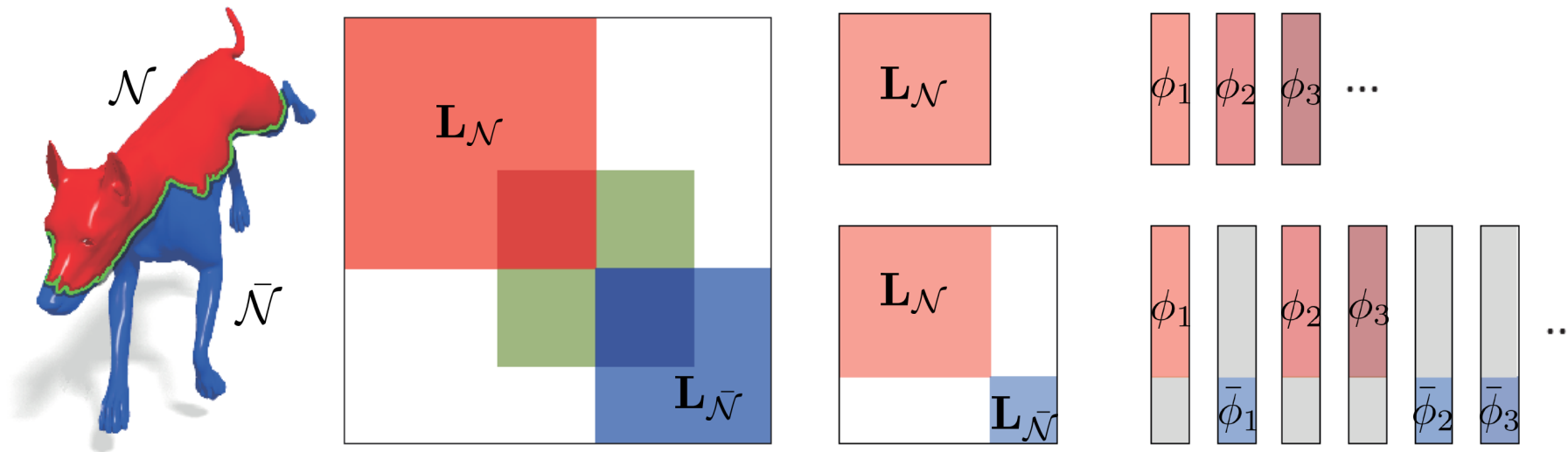
Partial Functional Maps

Block diagonal case

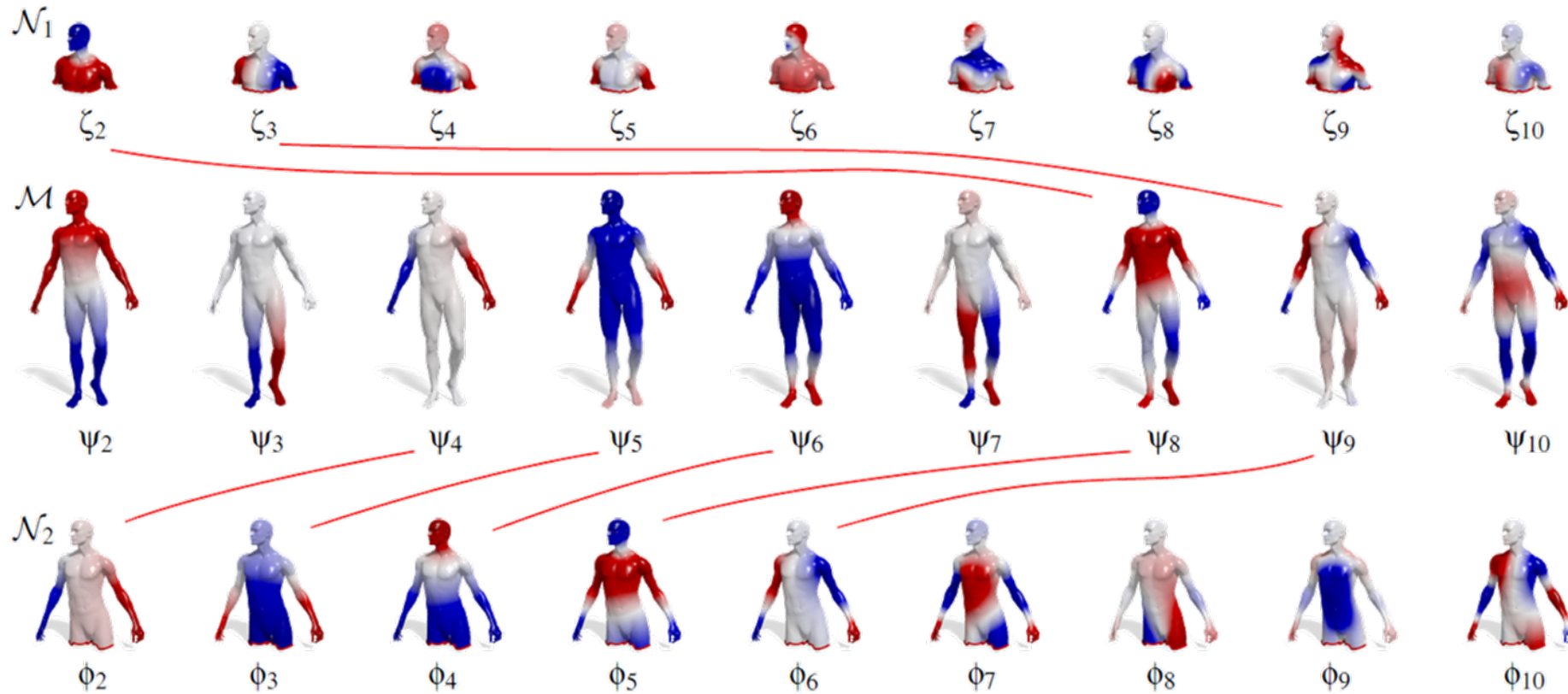


Partial Functional Maps

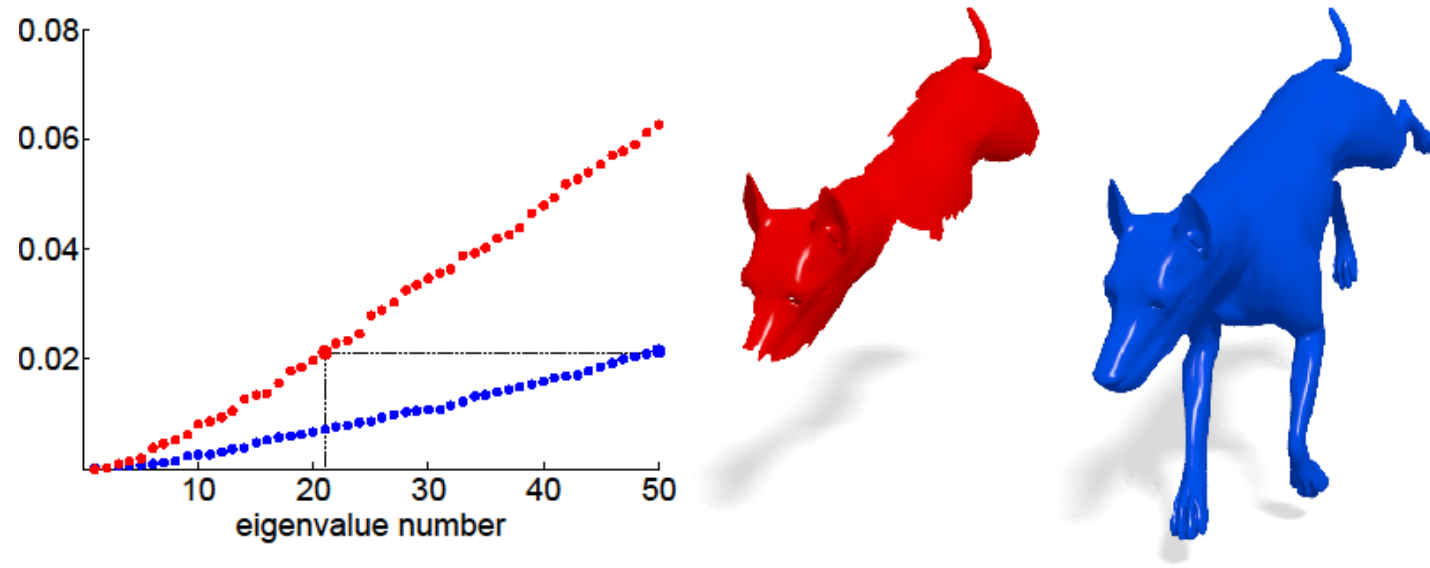
Weak coupling: eigenfunctions of the partial shape show up among those of the full shape, up to some bounded perturbation



Eigenfunction Preservation



The Slope Rule

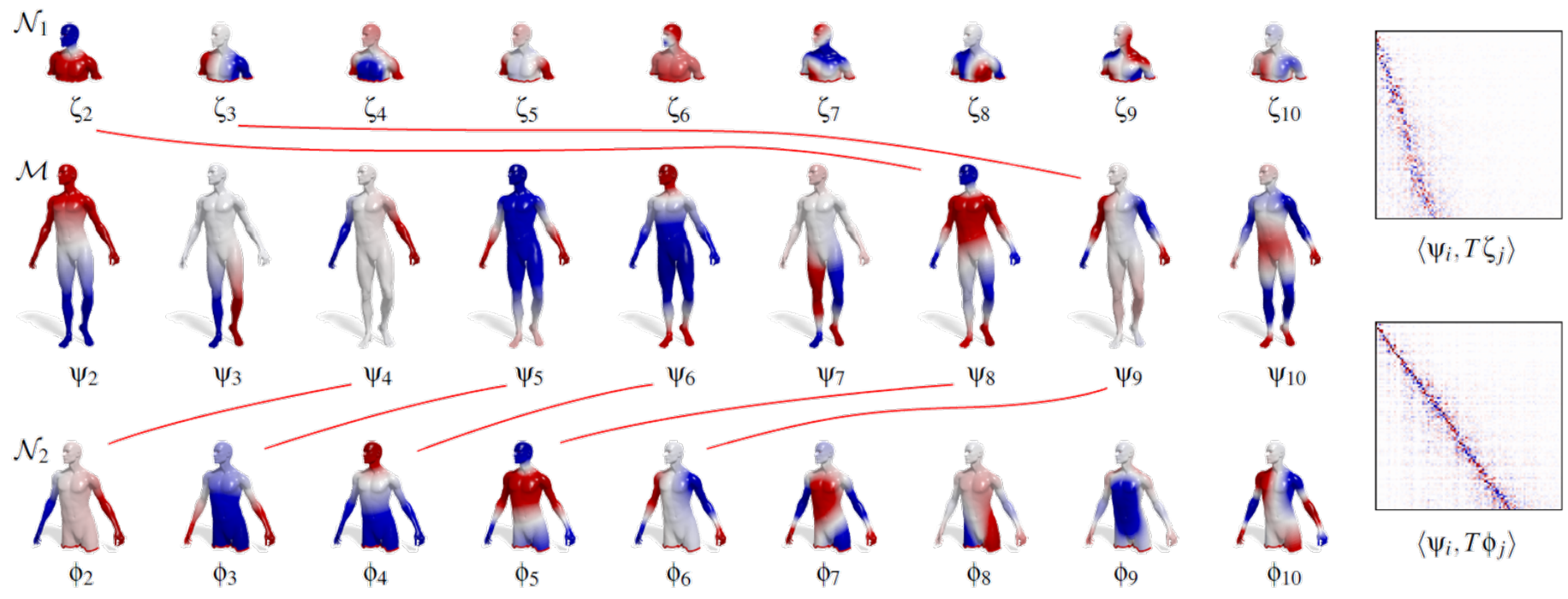


Weyl's law:

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

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

The Slope Rule



Partial Functional Maps



That's All

