NeurIPS | 2018

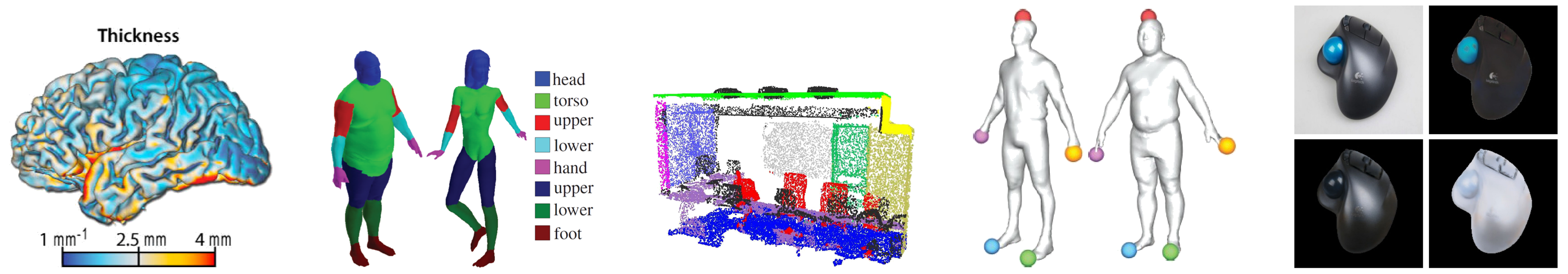
Deep Functional Dictionaries: Learning Consistent Semantic Structures on 3D Models from Functions

Minhyuk Sung [1] Hao Su [2] Ronald Yu [2] Leonidas Guibas [1]

[1]Stanford University [2]University of California San Diego

Introduction

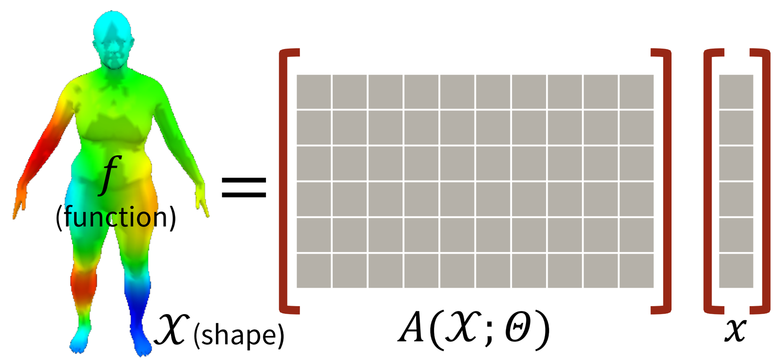
Various 3D semantic attributes can be encoded as per-point probe functions on 3D geometries.



Clinical annotations Parts Instances Keypoints Reflectance

Given a collection of related 3D shapes $\{\mathcal{X}_i\}$, we consider how to jointly analyze such probe functions over different shapes $\{f_i\}$, and how to discover common latent structures using a neural network — even in the absence of any correspondence information. Our network produces a small dictionary of basis functions for each shape $A(\mathcal{X}_i;\Theta)$, a dictionary whose linear span includes the semantic functions provided for that shape. Even though our shapes have independent discretizations and no functional correspondences are provided, the network is able to generate latent bases, in a consistent order, that reflect the shared semantic structure among the shapes.

- Each shape $\mathcal{X}$ is given as n points sampled on its surface.
- A function f is represented with a vector in $\mathbb{R}^n$ (a scalar per point).
- The atoms of the dictionary $A(\mathcal{X};\Theta) \in \mathbb{R}^{n\times k}$ are represented as columns of a matrix. k is a large number for the maximum size of the dictionary.
- The columns of $A(\mathcal{X};\Theta)$ are enforced to encode atomic semantics in applications (e.g. atomic instances in segmentation) by adding appropriate constraints.



Deep Functional Dictionary Learning Framework

Our neural network takes shape-function pairs $(\mathcal{X}, f)$ as inputs in training, and outputs a functional dictionary $A(\mathcal{X};\Theta)$ for each shape. Loss function $L(A(\mathcal{X};\Theta); f)$ is designed to minimize 1) the projection error from f to the vector space $A(\mathcal{X};\Theta)$, and 2) the number of atoms in the dictionary matrix:

$$
\begin{aligned}
L(A(\mathcal{X};\Theta); f) &= \min_{\boldsymbol{x}} F(A(\mathcal{X};\Theta), \boldsymbol{x}; f) + \gamma \|A(\mathcal{X};\Theta)\|_{2,1} \\
\text{s.t.} \quad & F(A(\mathcal{X};\Theta), \boldsymbol{x}; f) = \|A(\mathcal{X};\Theta)\boldsymbol{x} - f\|_2^2 \\
& C(A(\mathcal{X};\Theta), \boldsymbol{x})
\end{aligned}
$$

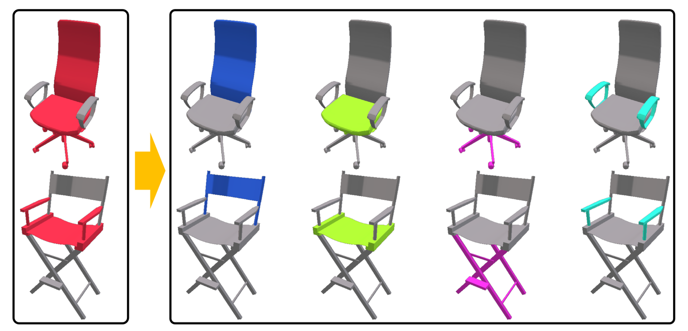
$\boldsymbol{x} \in \mathbb{R}^k$ is a linear combination weights, and $C(A(\mathcal{X};\Theta), \boldsymbol{x})$ is a set of constraints on both $A(\mathcal{X};\Theta)$ and $\boldsymbol{x}$ determined in each application.

Since the nested minimization $F(A(\mathcal{X};\Theta), \boldsymbol{x}; f)$ is generally not solved analytically due to $C(A(\mathcal{X};\Theta), \boldsymbol{x})$, we use an alternating minimization scheme:

function Single Step Gradient Iteration($\mathcal{X}, f, \Theta^t, \eta$)
  Compute: $A^t = A(\mathcal{X};\Theta^t)$.
  Solve: $\boldsymbol{x}^t = \arg\min_{\boldsymbol{x}} \|A^t\boldsymbol{x} - f\|_2^2$ s.t. $C(\boldsymbol{x})$.
  Update: $\Theta^{t+1} = \Theta^t - \eta \nabla L(A(\mathcal{X};\Theta^t); f, \boldsymbol{x}^t)$.
end function

Adaptation in Applications

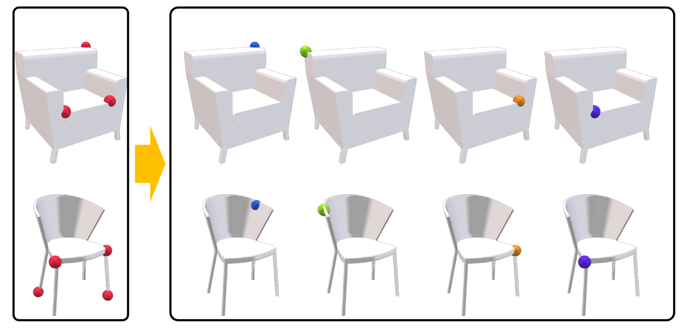
Weakly-supervised Co-segmentation



$[0,1]$ range and partitioning constraints:

$$C_{\text{seg}}(A(\mathcal{X};\Theta), \boldsymbol{x}) = \left\{ \begin{array}{l} \mathbf{0} \le \boldsymbol{x} \le \mathbb{1} \\ \mathbf{0} \le A(\mathcal{X};\Theta) \le \mathbb{1} \\ \Sigma_j A(\mathcal{X};\Theta)_{i,j} = 1 \text{ for all } i \end{array} \right\}$$

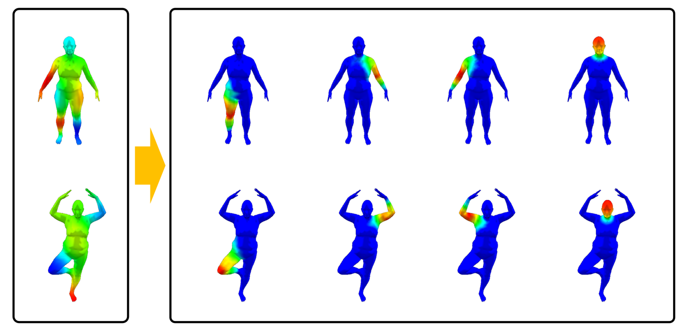
Weakly-supervised Keypoint Correspondence Estimation



$[0,1]$ range and single point constraints:

$$C_{\text{key}}(A(\mathcal{X};\Theta), \boldsymbol{x}) = \left\{ \begin{array}{l} \mathbf{0} \le \boldsymbol{x} \le \mathbb{1} \\ \mathbf{0} \le A(\mathcal{X};\Theta) \le \mathbb{1} \\ \Sigma_i A(\mathcal{X};\Theta)_{i,j} = 1 \text{ for all } j \end{array} \right\}$$

Smooth Function Apprixmation and Mapping



Unit vector constraint:

$$C_{\text{map}}(A(\mathcal{X};\Theta), \boldsymbol{x}) = \left\{ \sum_i A(\mathcal{X};\Theta)_{i,j}^2 = 1 \text{ for all } j \right\}$$

Neural Network Architecture

In all experiments, we used PointNet [1] segmentation architecture without any modification, but any other architecture processing 3D geometry can be employed.

ShapeNet Semantic Part Segmentation

Ours outperforms vanilla PointNet [1] on average mean IoU metric when finding correspondences between ground truth and predicted part segments in each shape (left). Note that PointNet uses additional part label supervision. The average Mean IoU is still comparable when finding correspondences between part labels and atom indices in each category (right).

	PointNet [1]	Ours
avg. mIoU	82.4	**84.6**

	Ours (per shape)	Ours (per cat.)
avg. mIoU	**84.6**	77.3

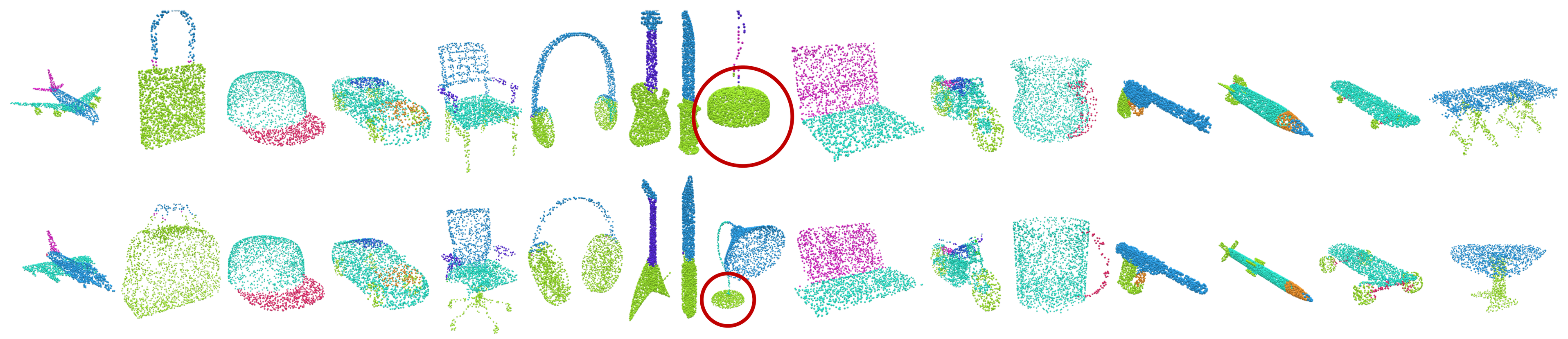
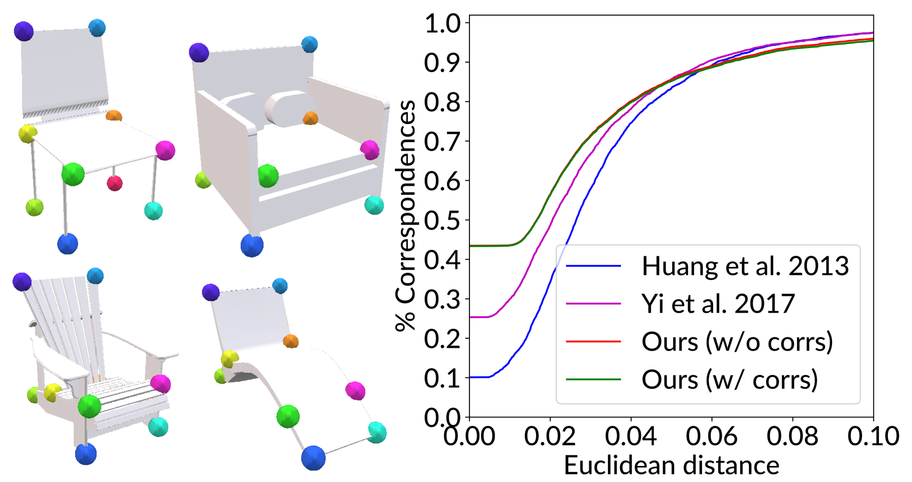


Figure 1: Colors of part segments indicate indices of atoms, which are consistent unless part geometries are not distinguishable: e.g. shade of ceiling lamp and base of standing lamp (in circles).

ShapeNet Keypoint Correspondences



The percentage of correct keypoints (PCK) is measured both when finding correspondences between ground truth and predicted keypoints in each shape (red line) and finding between ground truth labels and atom indices for all shapes (green line). The PCK curves are identical, meaning that the order of predicted keypoints is consistent across shapes.

S3DIS Instance Segmentation

The performance of instance proposal prediction in 3D scenes is evaluated with proposal recall metric. The class confusion (middle) is calculated as frequencies being included in the same atoms.

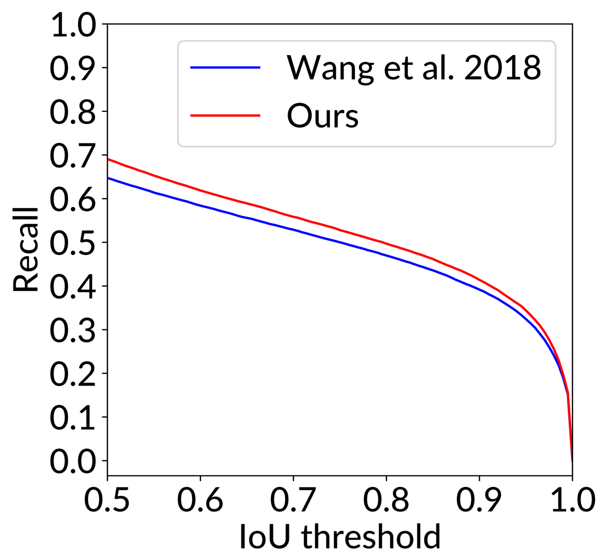


Figure 2: Proposal recall.

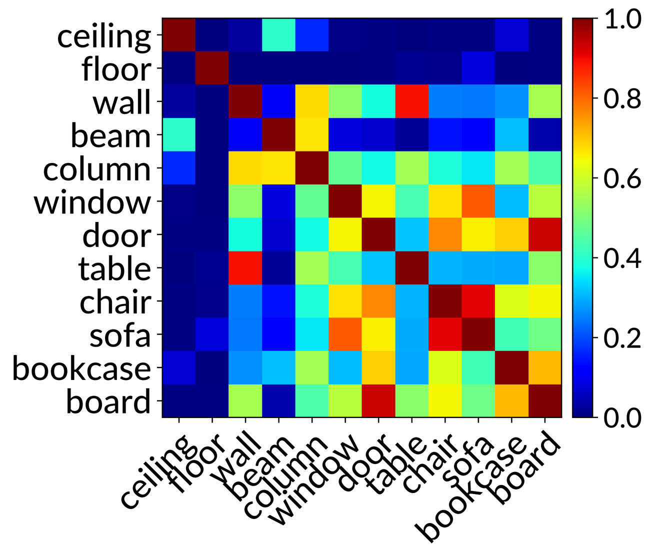


Figure 3: Class confusion matrix.

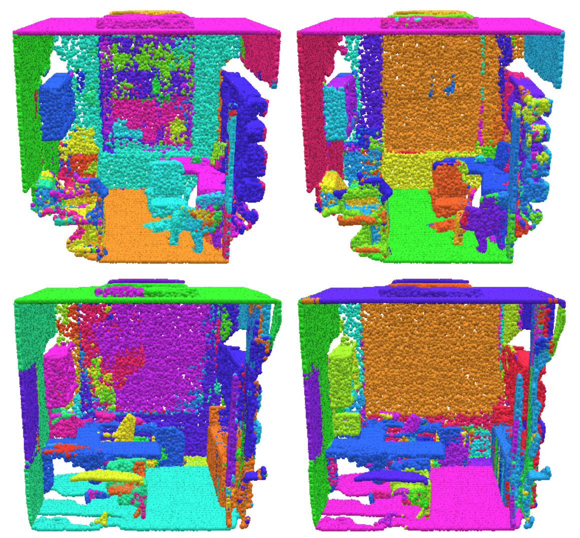
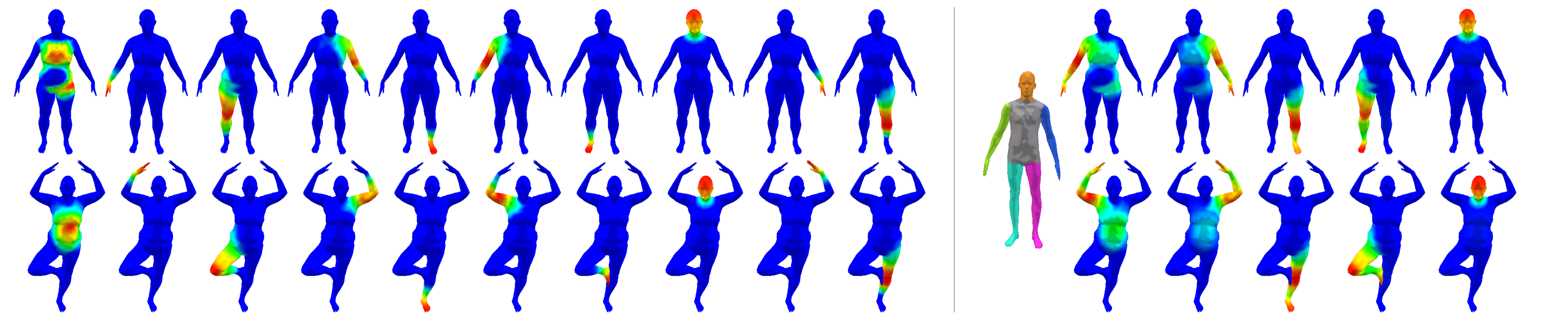


Figure 4: Wang et al. [2] (L) / Ours (R).

MPI-FAUST Human Shape Bases Synchronization

Our framework can find synchronized atomic functions for which linear combination can approximate any random continuous function (left). Any information in one shape can be transferred to the other shapes through the synchronized bases: e.g. parts (right).



[Code] https://github.com/mhsung/deep-functional-dictionaries

Acknowledgements

This project was supported by a DoD Vannevar Bush Faculty Fellowship, NSF grants CHS-1528025 and IIS-1763268, and an Amazon AWS AI Research gift.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[2] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018.