GAMES
Geometric Deep Learning III

Qixing Huang

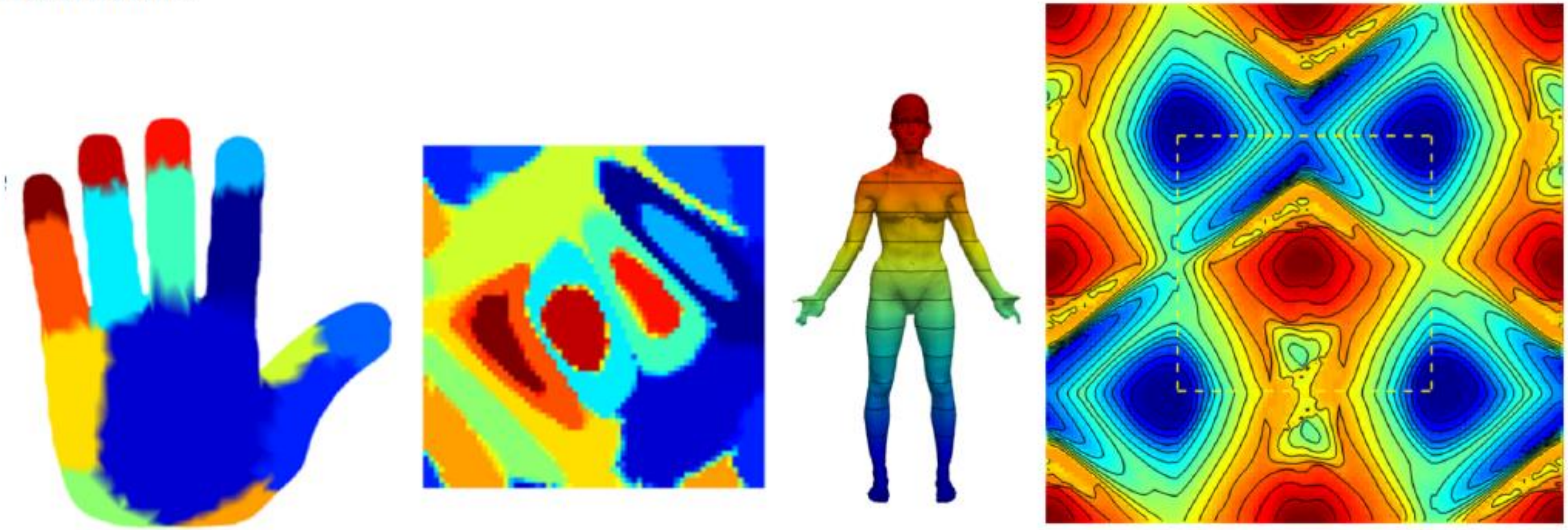
Oct. 14th 2021

Slide credit: Michael Bronstein

Parametric domain geometric deep learning methods

Global parametrization

Map the input surface to some parametric domain with shift-invariant structure



- ☺ Allows to use standard CNNs (pull back convolution from the parametric space)
- ☺ Guaranteed invariance to some classes of transformations
- ☹ Parametrization may not be unique
- ☹ Embedding may introduce distortion

Sinha et al. 2016; Maron et al. 2017; Ezuz et al. 2017

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation = flow along a non-vanishing vector field

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation = flow along a non-vanishing vector field

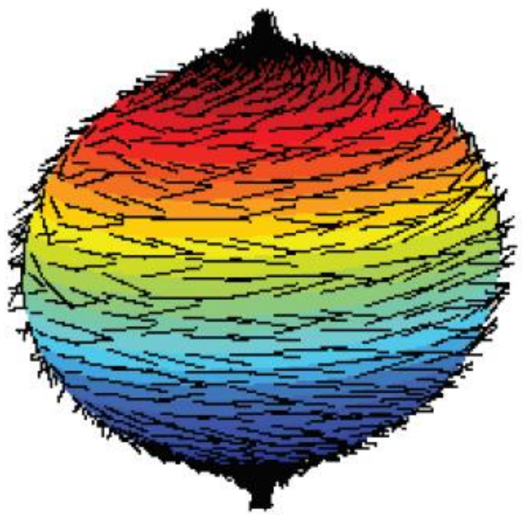
Poincaré-Hopf Theorem Non-vanishing vector field on a closed orientable compact 2-manifold implies manifold of genus 1 (torus)

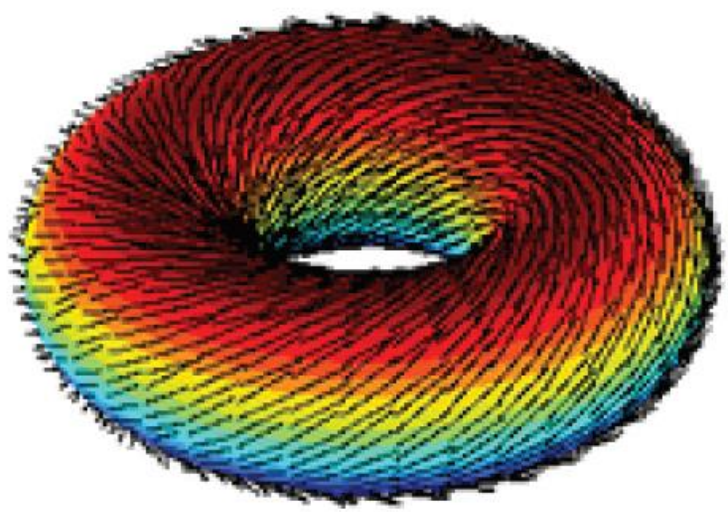
Poincaré 1881; Hopf 1926

Translation invariance on manifolds

Translation on manifold = locally Euclidean translation = flow along a non-vanishing vector field

Poincaré-Hopf Theorem Non-vanishing vector field on a closed orientable compact 2-manifold implies manifold of genus 1 (torus)

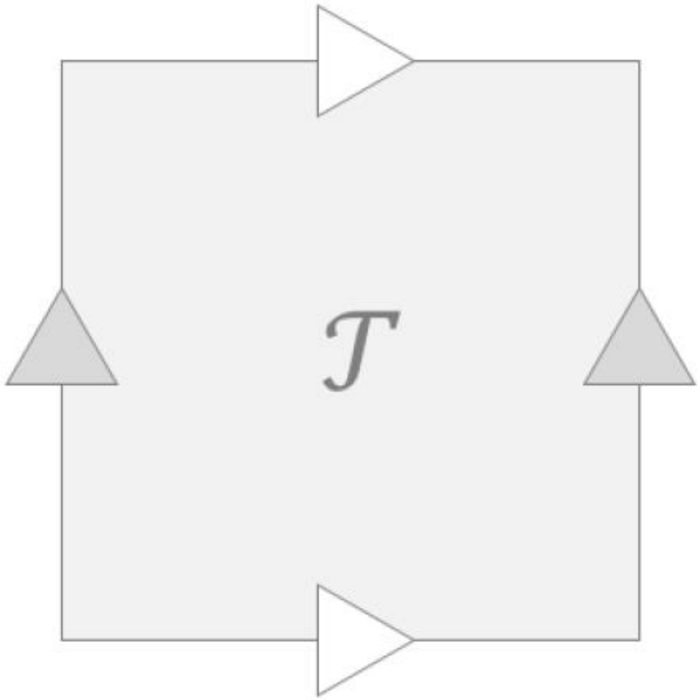




'Hairy Ball Theorem' states that a sphere cannot be combed

Poincaré 1881; Hopf 1926

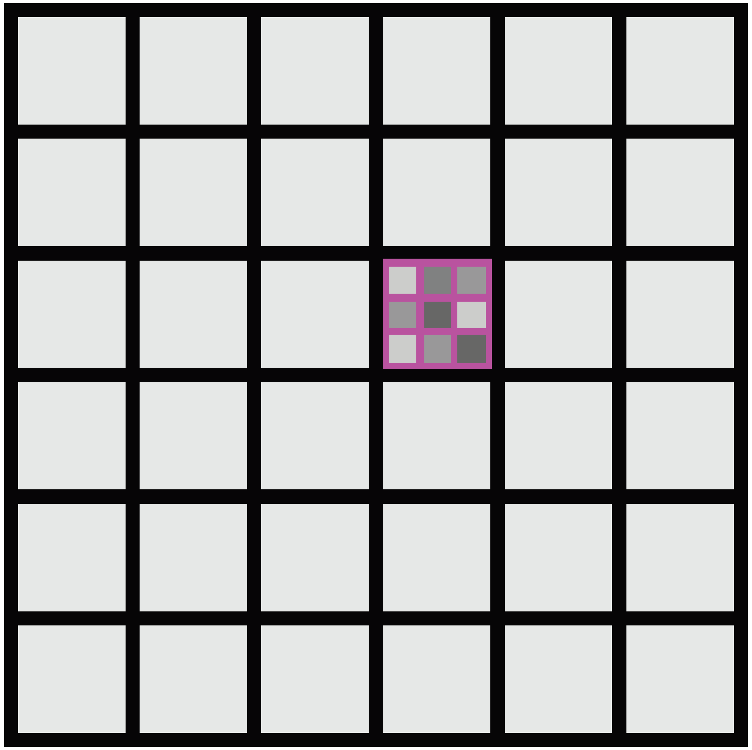
Translation invariance on the torus



Torus is the only closed orientable surface admitting a translation group

Convolution on torus

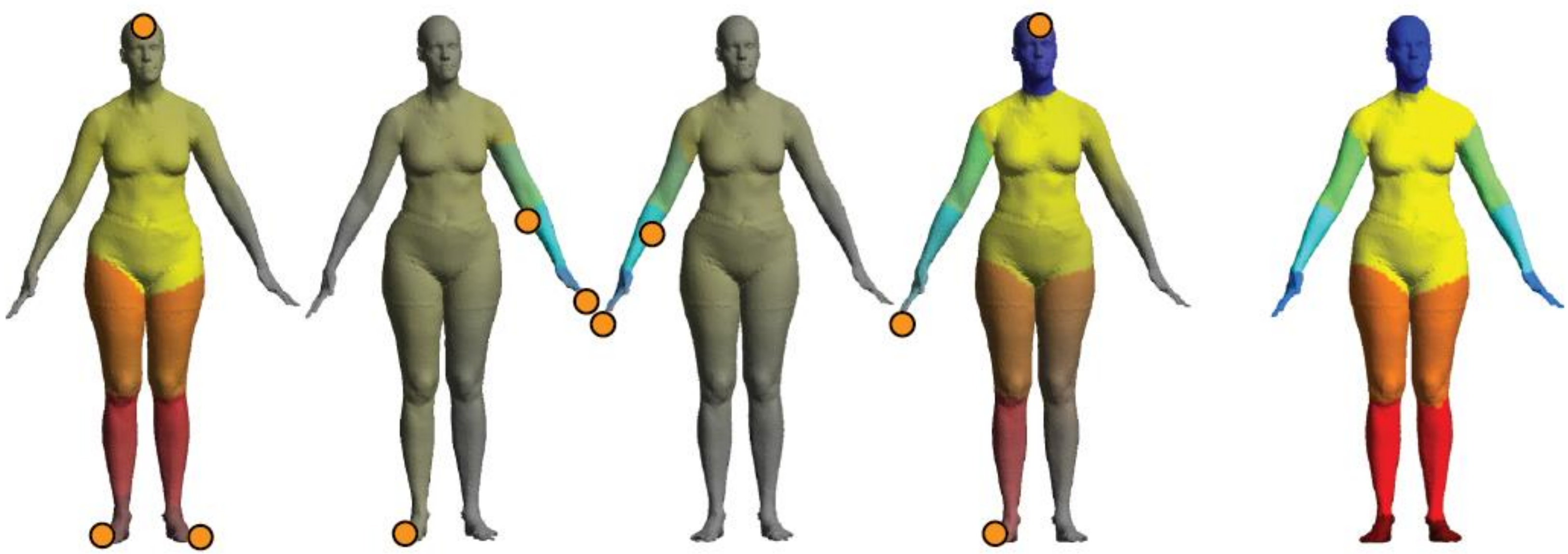
For any triplet of points on $\mathcal{X}$, construct conformal homeomorphism from the 4-cover $\mathcal{X}^4$ to $\mathcal{T}$ using orbifold-Tutte method



Maron et al. 2017

Conformal zoom

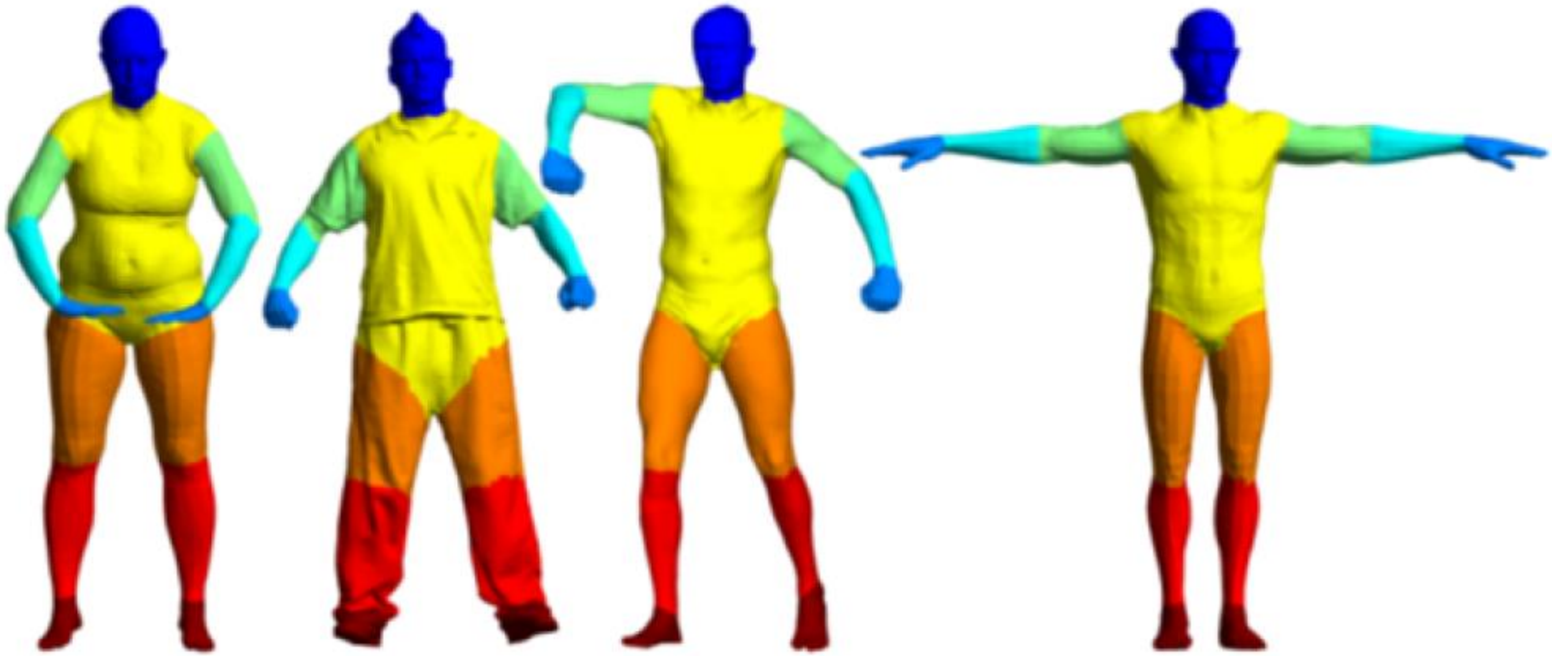
- Embedding depends on the choice of the triplets of points
- 'Conformal zoom' effect
- Choose multiple triples and aggregate results in training / test phase



different triplets aggregated result

Maron et al. 2017

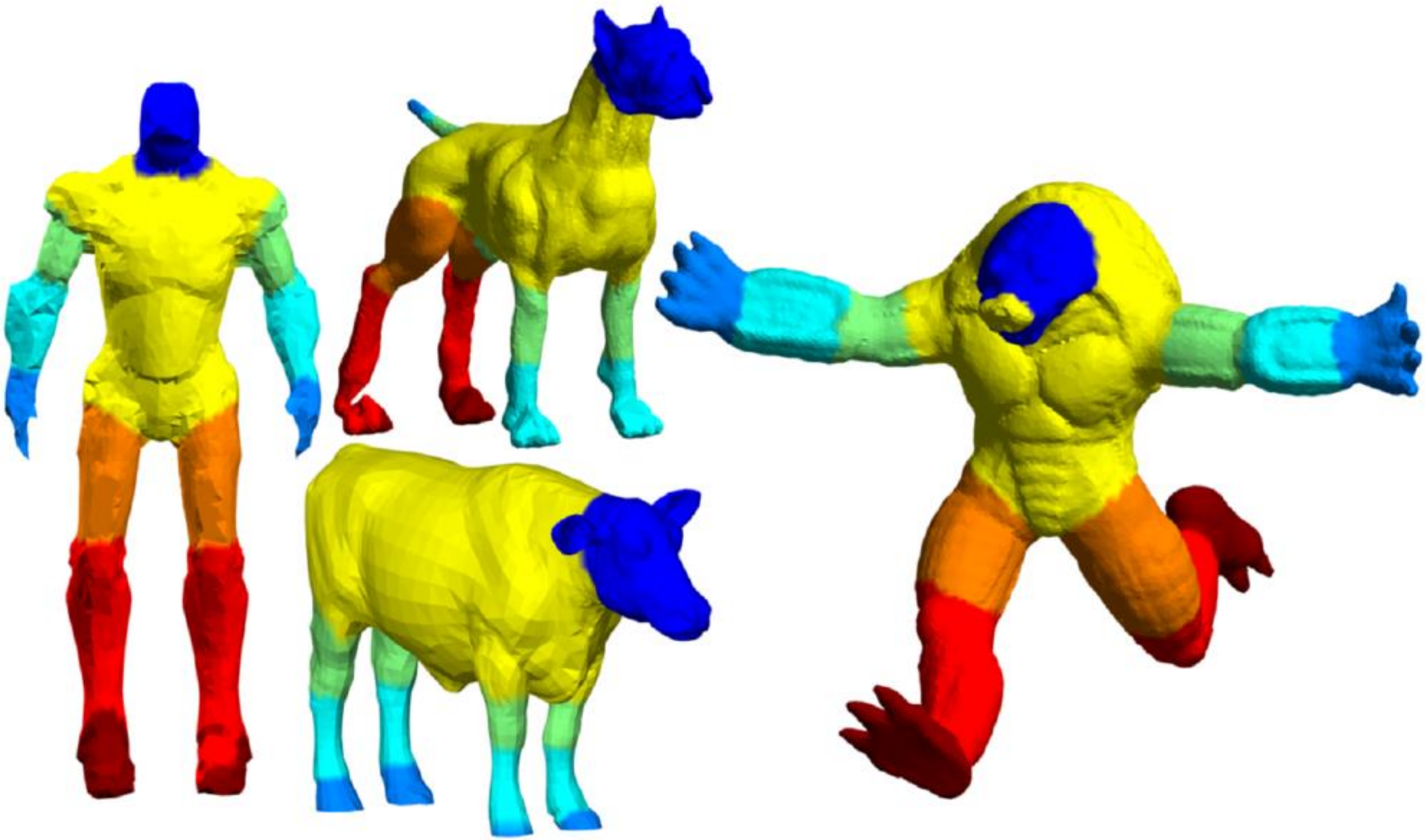
Example: shape segmentation with Toric CNN



Examples of shape segmentation obtained with Toric CNN

Maron et al. 2017

Example: shape segmentation with Toric CNN



Examples of shape segmentation obtained with Toric CNN

Maron et al. 2017

Application in
Computer Graphics and 3D Vision

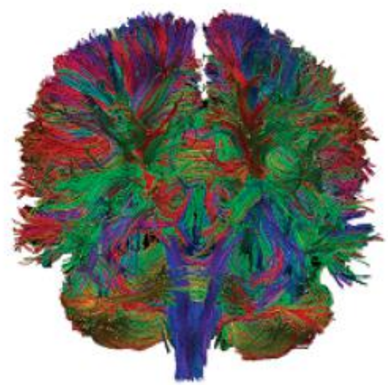
Application dealing with 3D data

Computer graphics

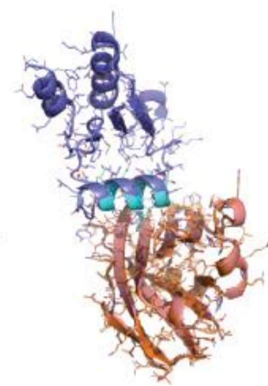
Virtual/augmented reality

Robotics

Autonomous driving

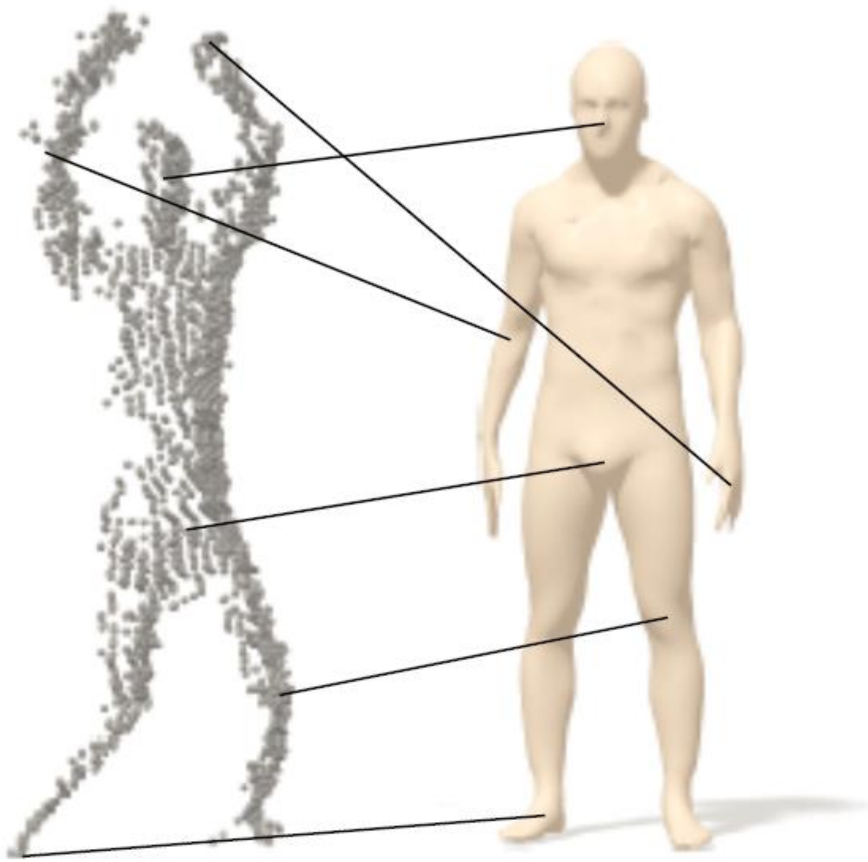


Medicine



Drug design

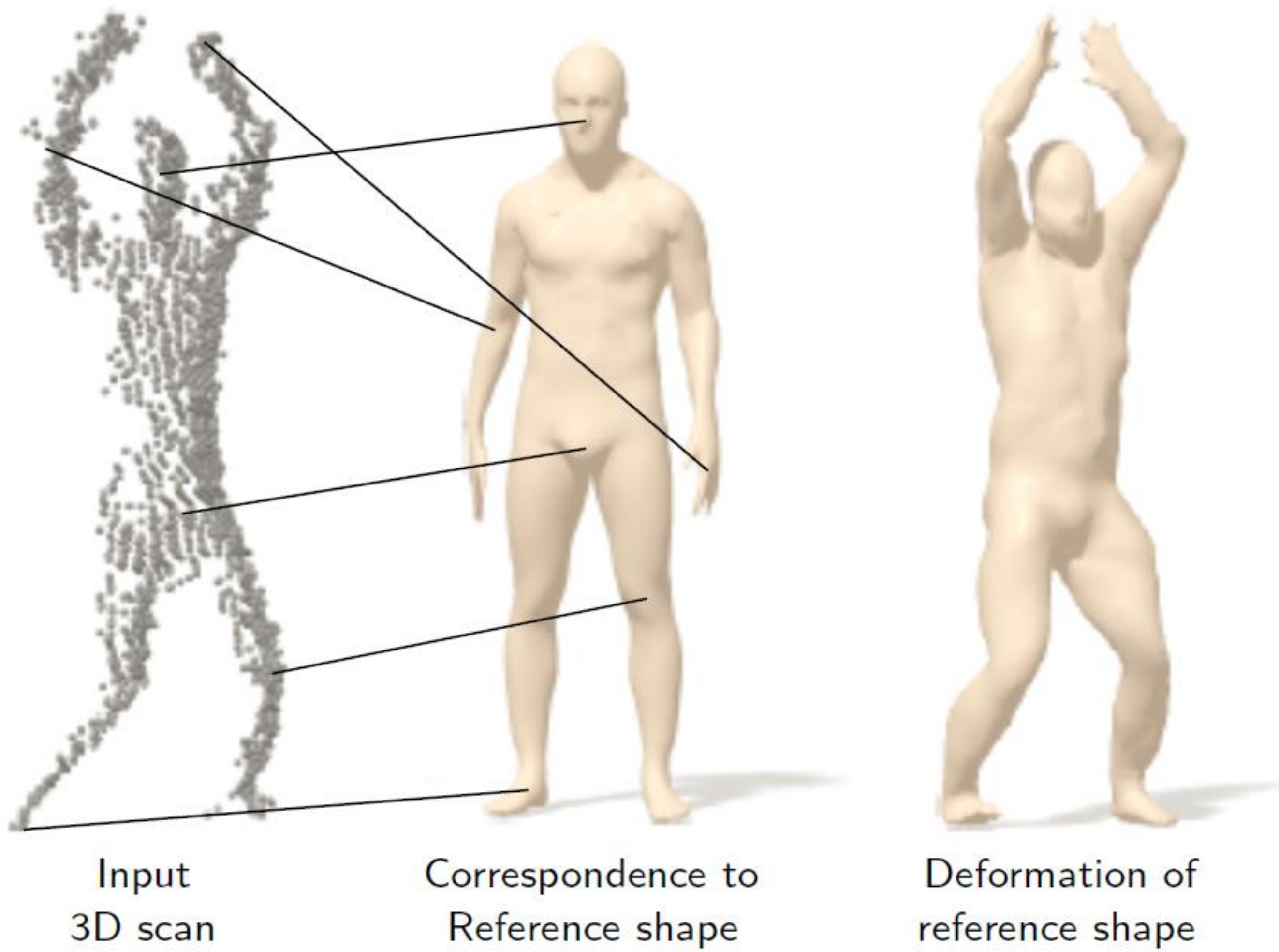
Analysis and synthesis



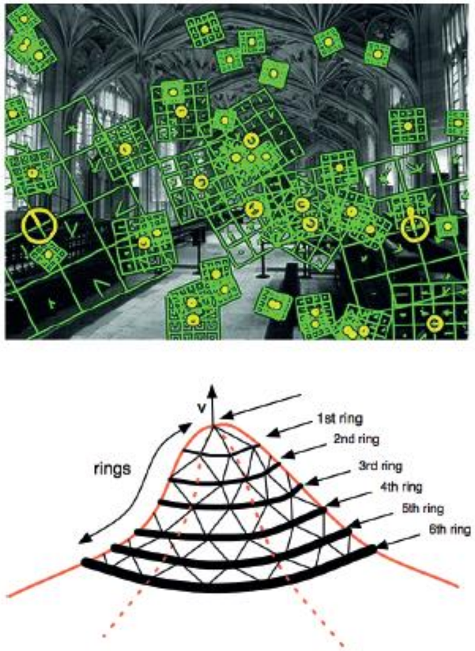
Input
3D scan

Correspondence to
Reference shape

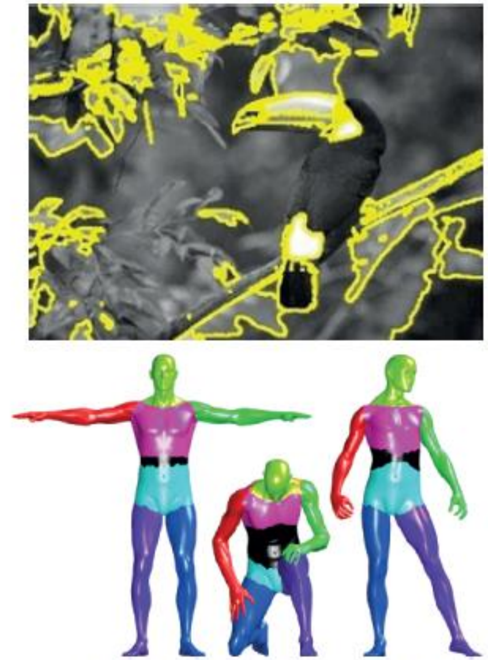
Analysis and synthesis



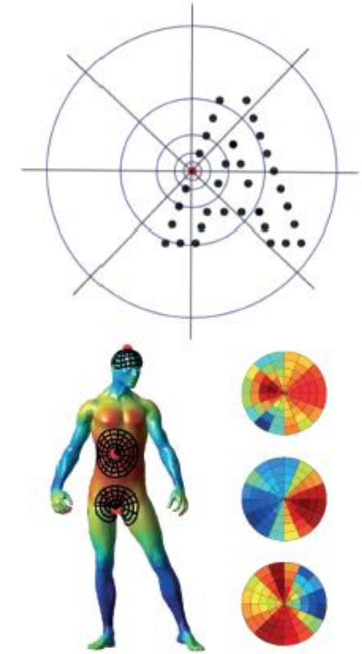
3D feature descriptors



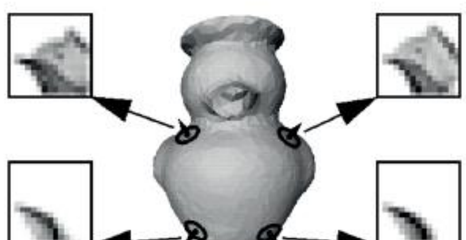
SIFT[1] / MeshHOG[2]



MSER[3] / ShapeMSER[4]



(Intrinsic[6]) Shape context[5]



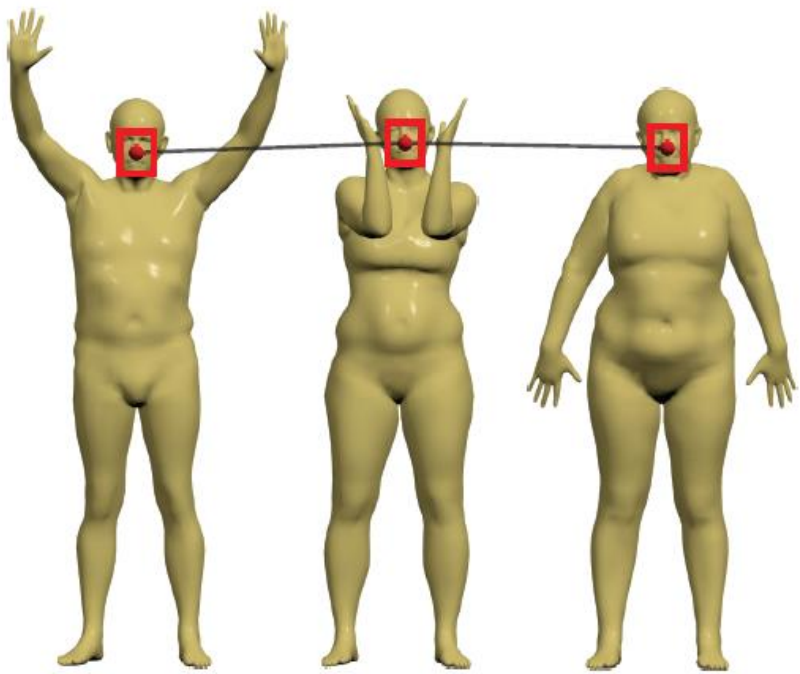
Spin image[7]

Heat kernel signature[8]

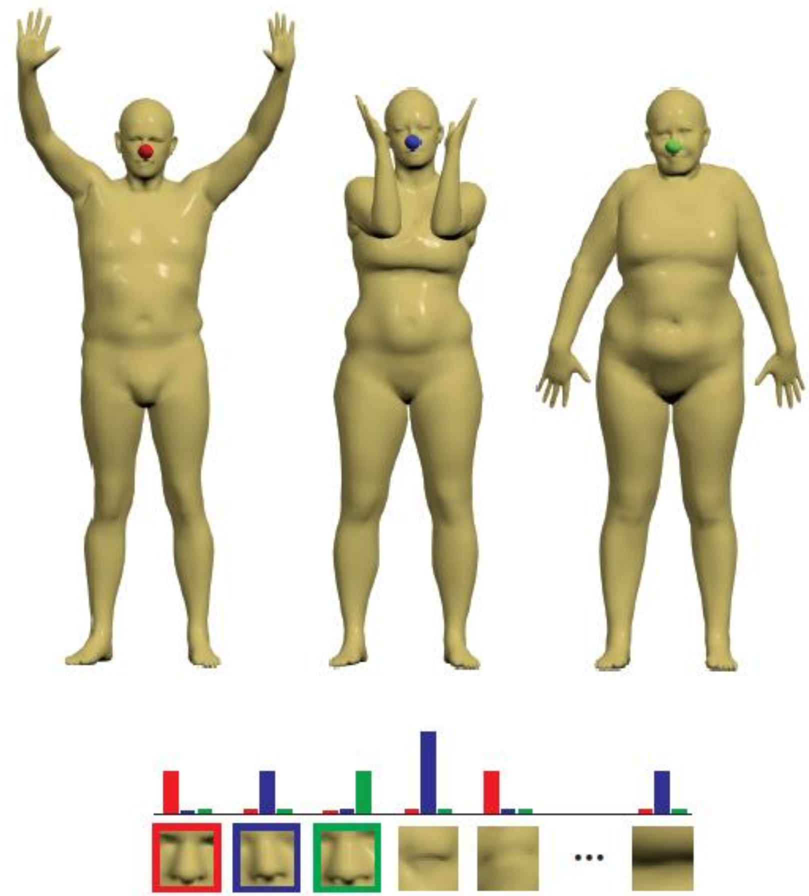
[1]Lowe 2004; [2]Zaharescu et al. 2009; [3]Matas et al. 2002; [4]Litman et al. 2010;
[5]Belongie et al. 2000; [6]Kokkinos et al. 2012; [7]Johnson et al. 1999; [8]Sun et al. 2009

Task-specific features

Correspondence



Similarity



Shape representation

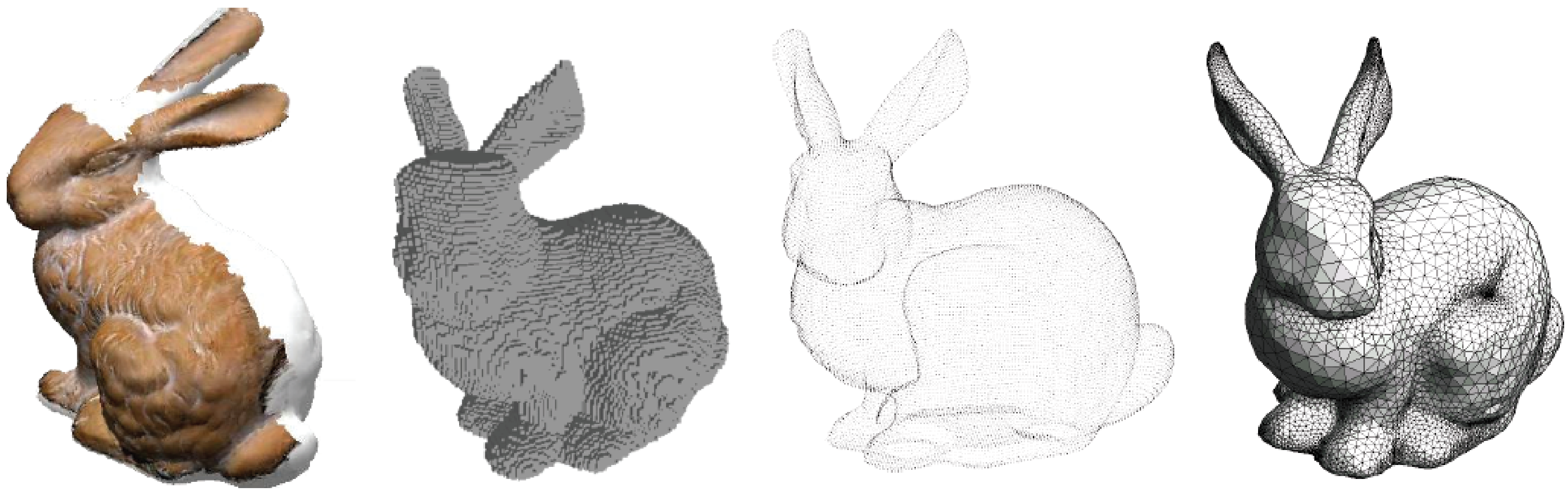


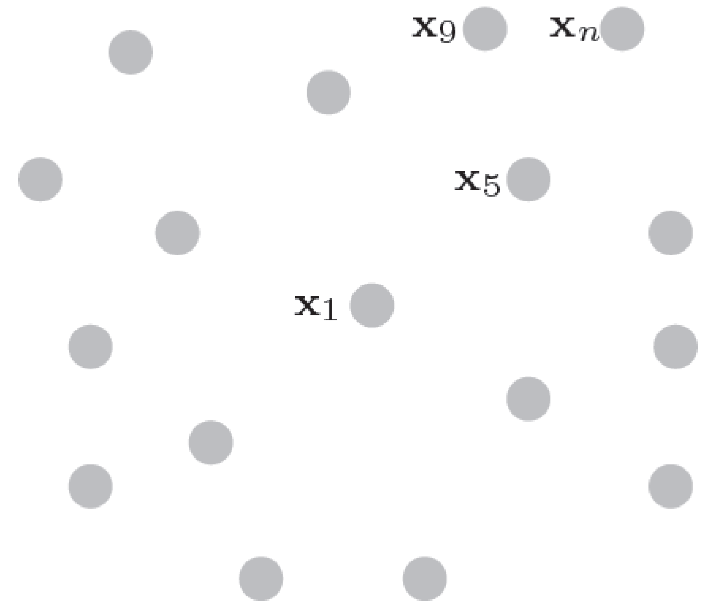
Image-based Volumetric Point-based Surface-based

PointNet: learning on sets

- Permutation-invariant function

  $$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = f(\mathbf{x}_{\pi_1}, \ldots, \mathbf{x}_{\pi_n})$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i



(Vinyals, Bengio, Kudlur 2015); Qi et al. 2017

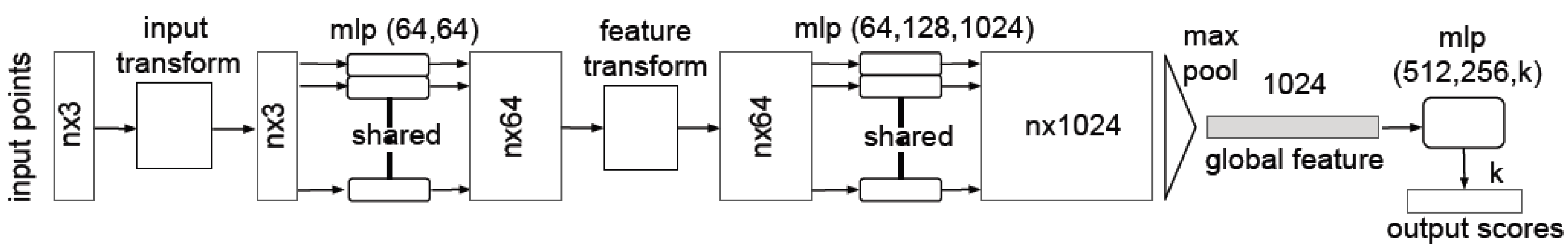
PointNet: learning on sets

- Permutation-invariant function

  $$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = f(\mathbf{x}_{\pi_1}, \ldots, \mathbf{x}_{\pi_n})$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i

- Shared function $h_{\mathbf{\Theta}}(\cdot)$ applied to each point + permutation-invariant aggregation ($\max$ or $\sum$)



(Vinyals, Bengio, Kudlur 2015); Qi et al. 2017

PointNet: learning on sets

- Permutation-invariant function

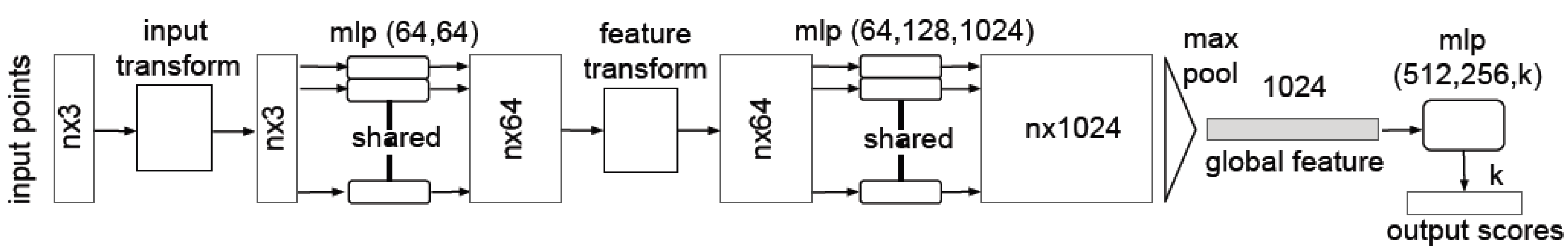
  $$f(\mathbf{x}_1, \dots, \mathbf{x}_n) = f(\mathbf{x}_{\pi_1}, \dots, \mathbf{x}_{\pi_n})$$

  where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i

- Shared function $h_{\boldsymbol{\Theta}}(\cdot)$ applied to each point + permutation-invariant aggregation ($\max$ or $\sum$)

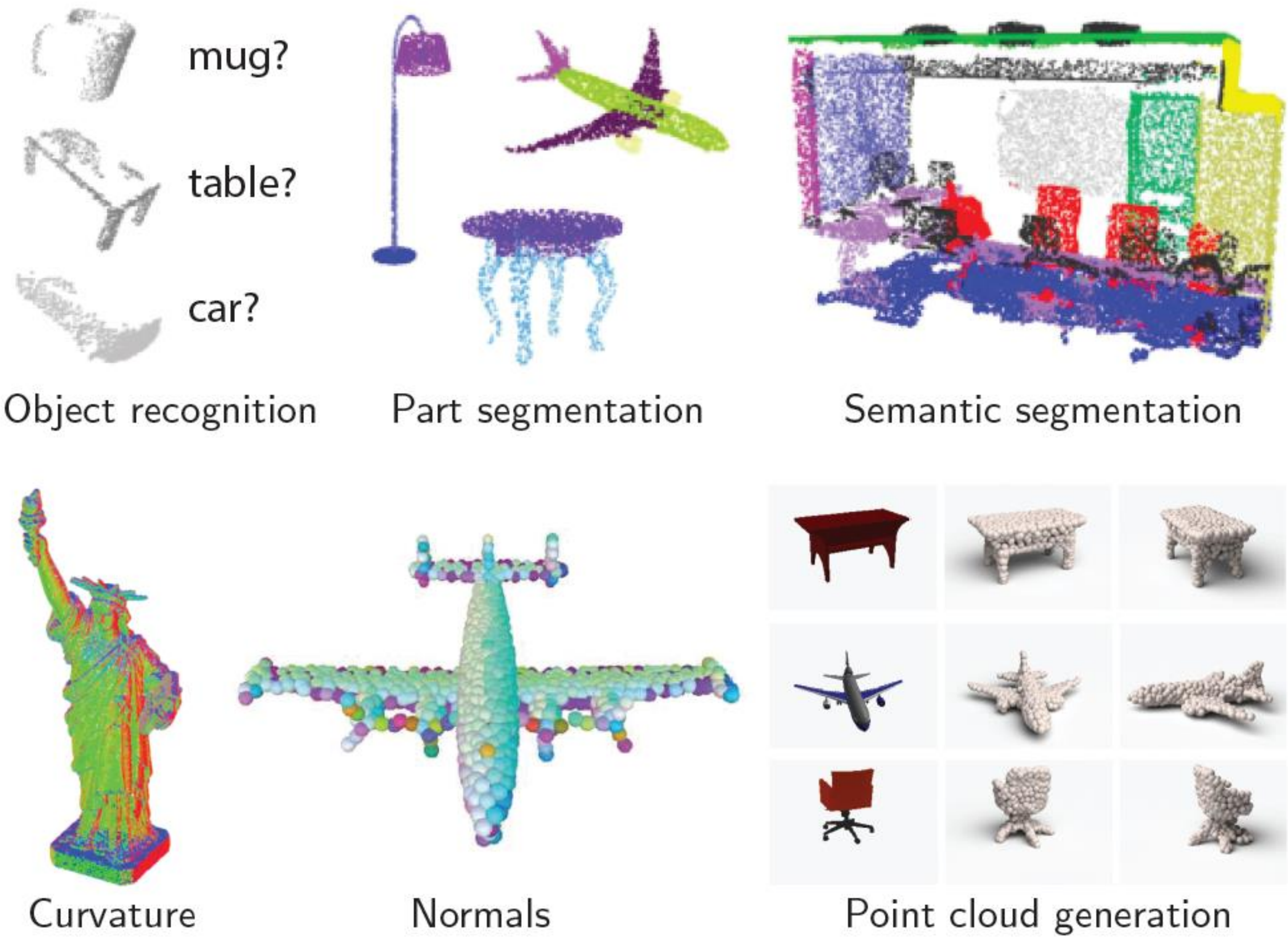
- Spatial transformer units

- Local grouping (PointNet++, PCPNet)



(Vinyals, Bengio, Kudlur 2015); Qi et al. 2017; Qi, Yi et al. 2017; Guerrero et al. 2018

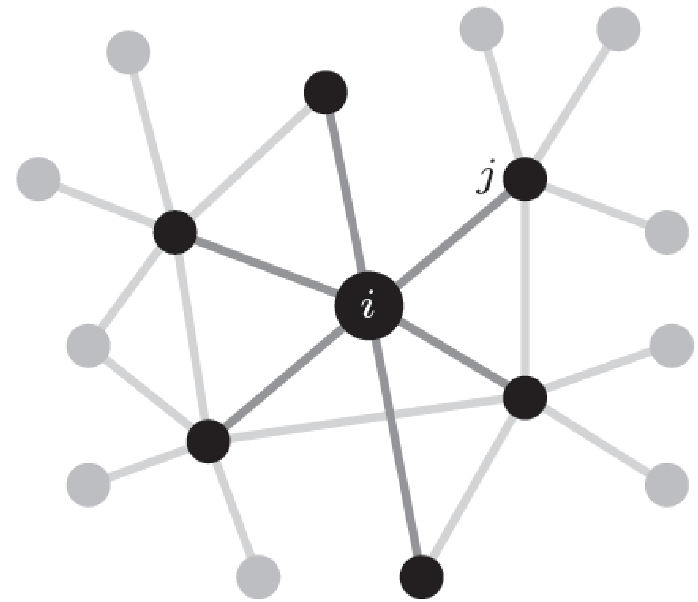
PointNet applications



Graph-based edge convolution

- Local neighborhood structure modeled as a graph
- Edge feature function $h_{\mathbf{\Theta}}(\cdot,\cdot)$ parametrized by $\mathbf{\Theta}$
- Permutation-invariant aggregation operator $\square$ (e.g. $\sum$ or $\max$) on the neighborhood of i
- Edge convolution (EdgeConv)

$$\mathbf{x}'_i = \underset{j}{\square}\, h_{\mathbf{\Theta}}(\mathbf{x}_i, \mathbf{x}_j)$$



Learnable local (nonlinear) operator

Wang et al. 2018

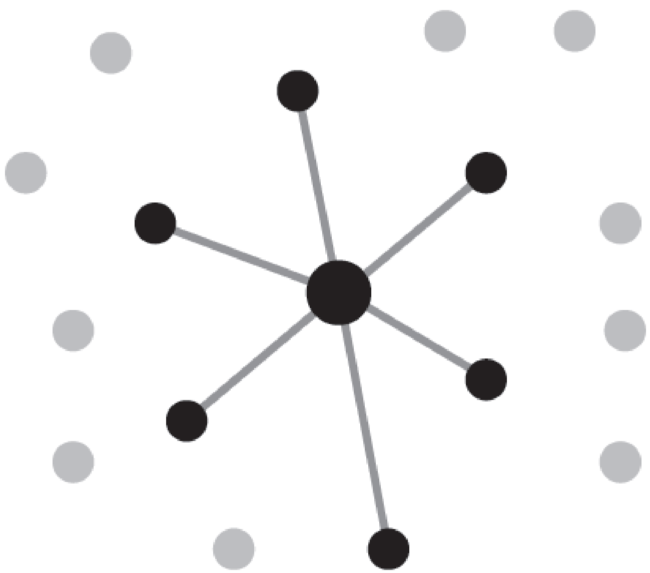
Particular cases

Method	**Aggregation** $\square$	**Edge feature** $h(\mathbf{x}_i, \mathbf{x}_j)$
Laplacian	$\sum$	$w_{ij}(\mathbf{x}_j - \mathbf{x}_i)$
PointNet[1]	–	$h(\mathbf{x}_i)$
PointNet++[2]	$\max$	$h(\mathbf{x}_i)$
MoNet[3]	$\sum$	$\sum_\ell g_\ell w_\ell(\mathbf{u}_{ij})\mathbf{x}_j$
PCNN[4]	$\sum$	$\sum_{\ell m} c(\mathbf{x}_i \cdot \mathbf{k}_{\ell m}) w_i q_{\mathbf{\Theta}_\ell}(\mathbf{x}_i, \mathbf{x}_j)$

Wang et al. 2018; [1]Qi et al. 2017; [2]Qi, Su et al. 2017; [3]Monti et al. 2017; [4]Atzmon et al. 2018

Dynamic Graph CNN (DynGCNN)

Construct k-NN graph in feature space and update it after each layer

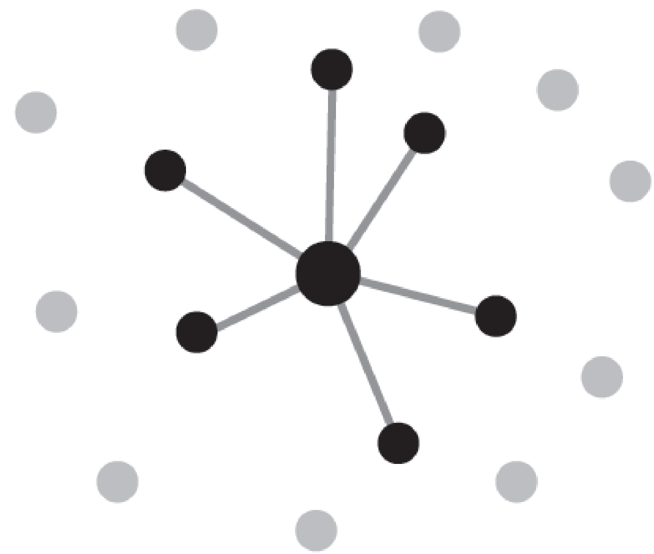


Layer l

Features $\mathbf{x}_1^{(l)}, \dots, \mathbf{x}_n^{(l)} \in \mathbb{R}^{d_l}$

k-NN graph $\mathcal{G}^{(l)}$

$h^{(l)} : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathbb{R}^{d_{l+1}}$



Layer $l+1$

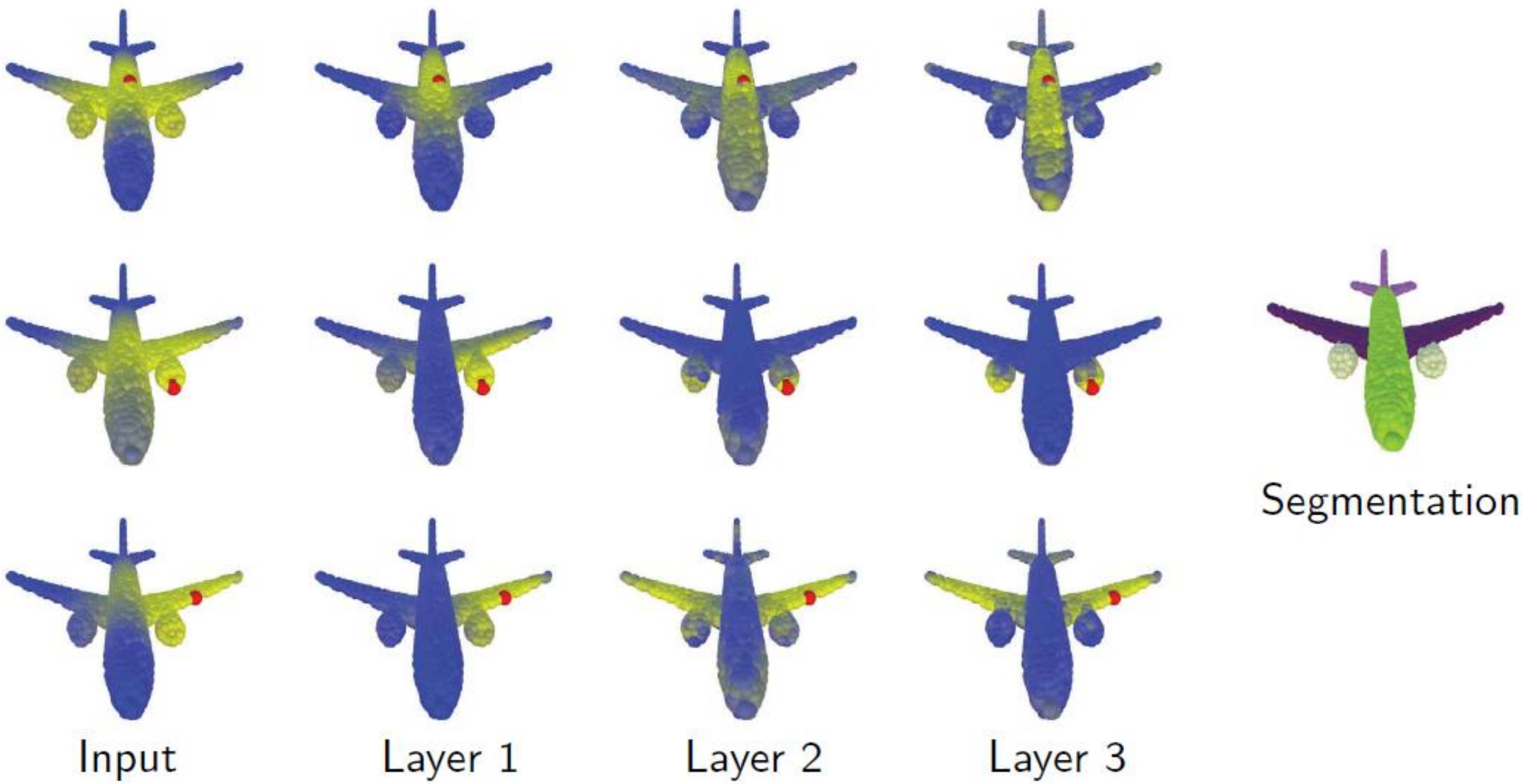
Features $\mathbf{x}_1^{(l+1)}, \dots, \mathbf{x}_n^{(l+1)} \in \mathbb{R}^{d_{l+1}}$

k-NN graph $\mathcal{G}^{(l+1)}$

$h^{(l+1)} : \mathbb{R}^{d_{l+1}} \times \mathbb{R}^{d_{l+1}} \to \mathbb{R}^{d_{l+2}}$

Wang et al. 2018

Learning semantic features



Left: Distance from red point in the feature space of different DynGCNN layers
Right: semantic segmentation results

Wang et al. 2018

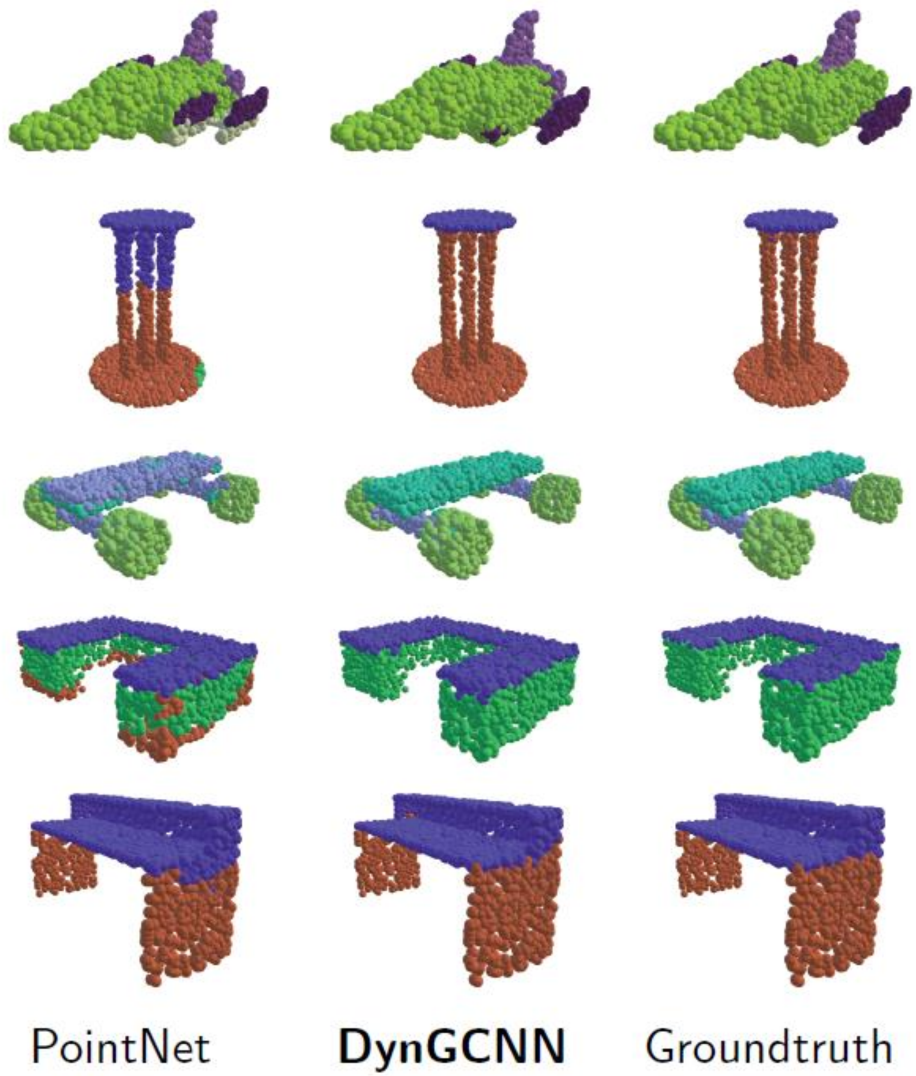
Shape classification (ModelNet40)

Method	Mean class accuracy	Overall accuracy
3DShapeNet[1]	77.3%	84.7%
VoxNet[2]	83.0%	85.9%
Subvolume[3]	86.0%	89.2%
ECC[4]	83.2%	87.4%
PointNet[5]	86.0%	89.2%
PointNet++[6]	–	90.7%
Kd-Net[7]	–	91.8%
DynGCNN (baseline) [8]	88.8%	91.2%
DynGCNN[8]	**90.2%**	**92.2%**

Classification accuracy of different methods on ModelNet40

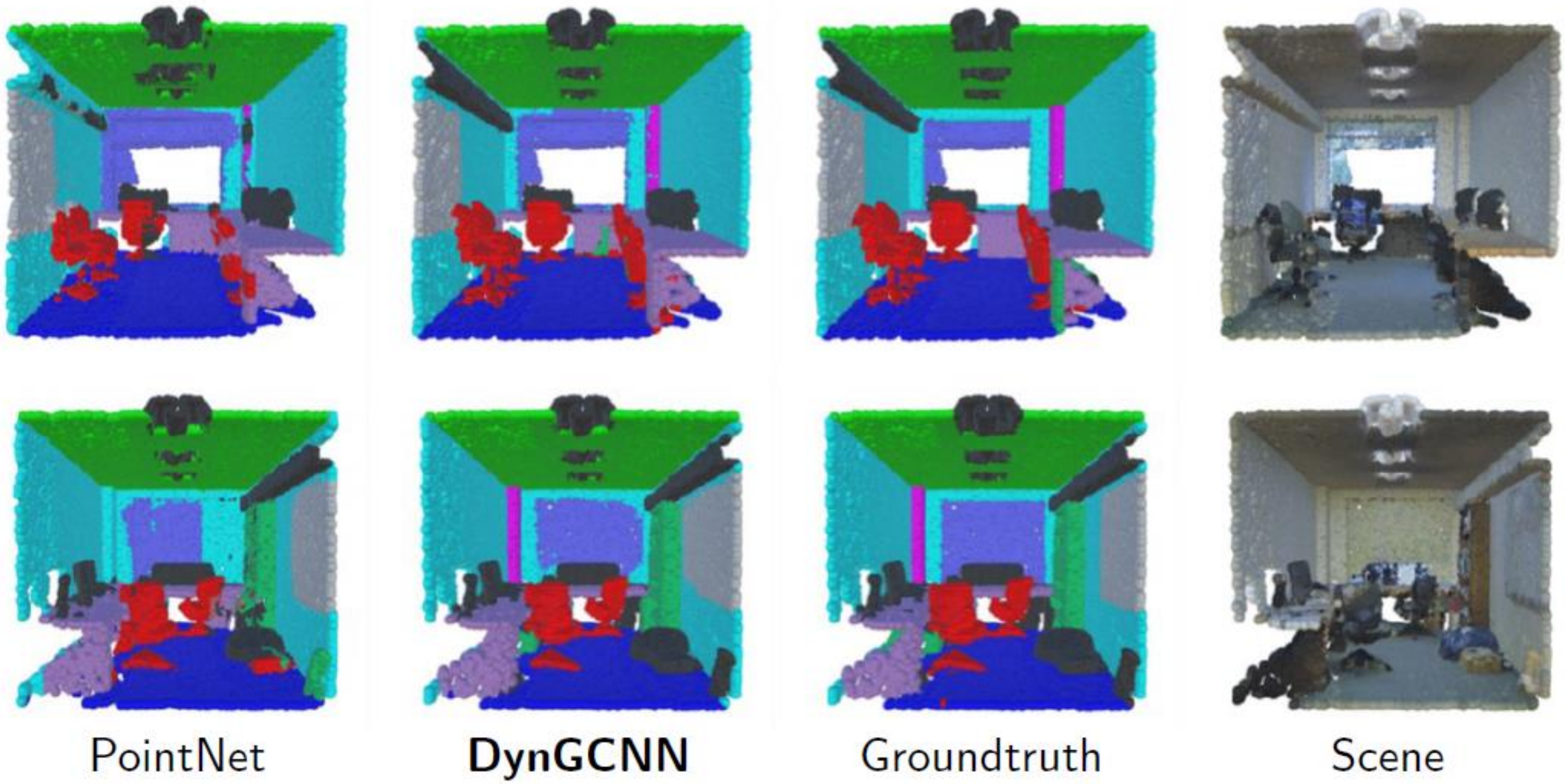
Methods: [1]Wu et al. 2015; [2]Maturana et al. 2015; Qi et al. 2016; [4]Simonovsky, Komodakis 2017; [5]Qi et al. 2017; [6]Qi, Su et al. 2017; [7]Klokov, Lempitsky 2017; [8]Wang et al. 2018; data: Wu et al. 2015 (ModelNet)

Semantic segmentation: synthetic (ShapeNet)



Methods: Qi et al. 2017 (PointNet); Wang et al. 2018 (DynGCNN); data: Yi et al. 2016 (ShapeNet)

Semantic segmentation: indoor scans (S3DIS)



Results of semantic segmentation of point cloud+RGB data using different architectures

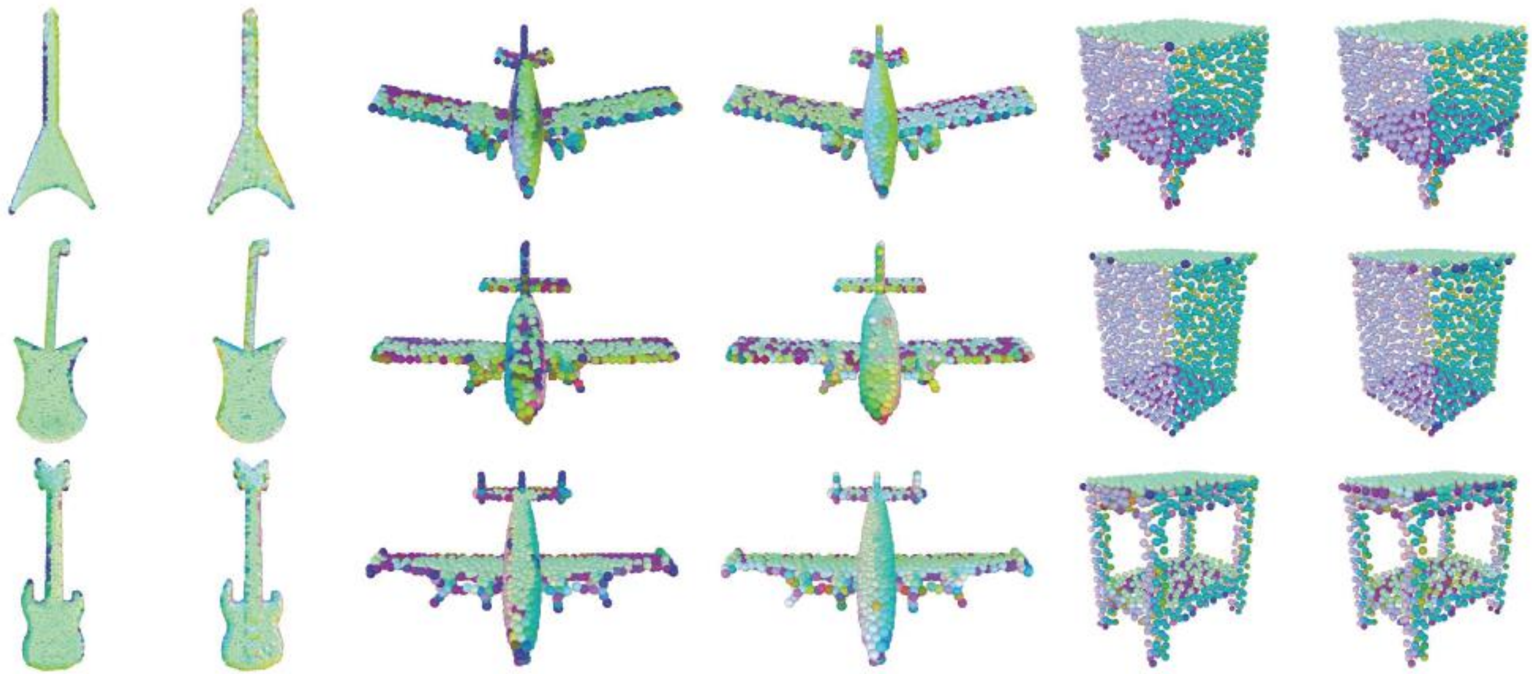
Methods: Qi et al. 2017 (PointNet); Wang et al. 2018 (DynGCNN); data: Armeni et al. 2016 (S3DIS)

Method	Mean IoU	Overall accuracy
PointNet (Baseline)[1]	20.1%	53.2%
PointNet[1]	47.6%	78.5%
MS + CU(2)[2]	47.8%	79.2%
G + RCU[2]	49.7%	81.1%
DynGCNN[3]	**56.1%**	**84.1%**

S3DIS indoor scene semantic segmentation accuracy

Methods: [1]Qi et al. 2017; [2]Engelmann et al. 2017 [3]Wang et al. 2018; data: Armeni et al. 2016 (S3DIS)

Surface normal prediction

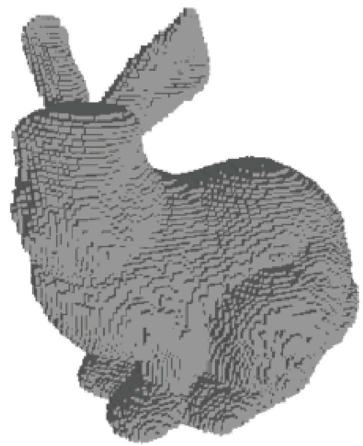


Surface normal predicted using DynGCNN (odd columns) and groundtruth (even columns). Normal direction is color-coded

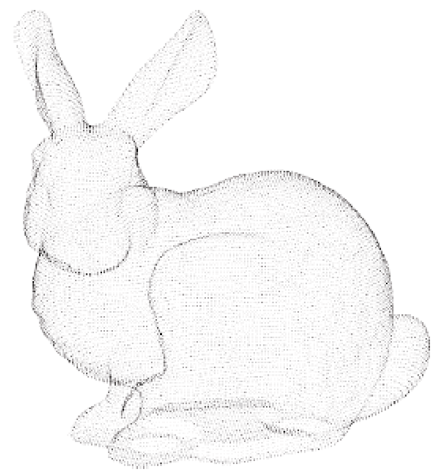
Wang et al. 2018; data: Wu et al. 2015 (ModelNet)

Shape representation

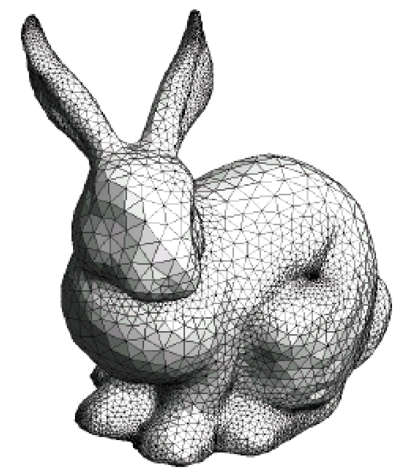
Image-based



Volumetric



Point-based

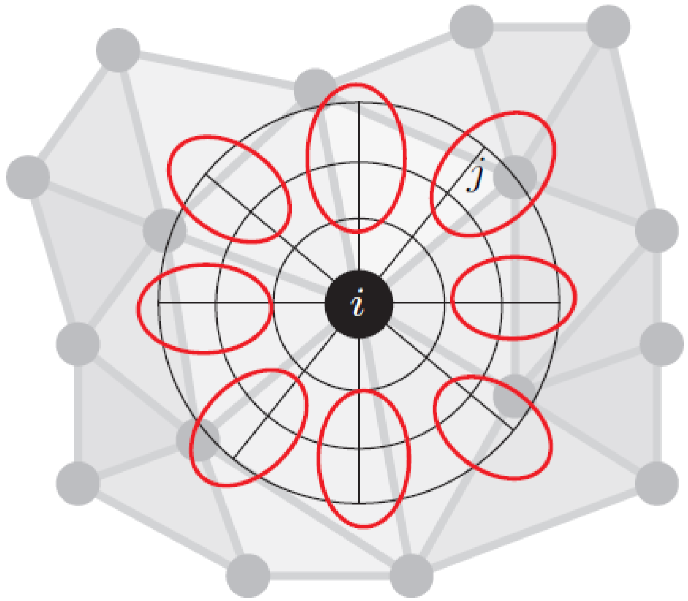


Surface-based

Convolution on meshes

- Local system of coordinates $\mathbf{u}_{ij}$ around i (e.g. geodesic polar)
- Local weights $w_1(\mathbf{u}), \ldots, w_L(\mathbf{u})$ w.r.t. $\mathbf{u}$, e.g. Gaussians

  $$w_\ell = \exp\left(-(\mathbf{u}-\boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1}(\mathbf{u}-\boldsymbol{\mu}_\ell)\right)$$



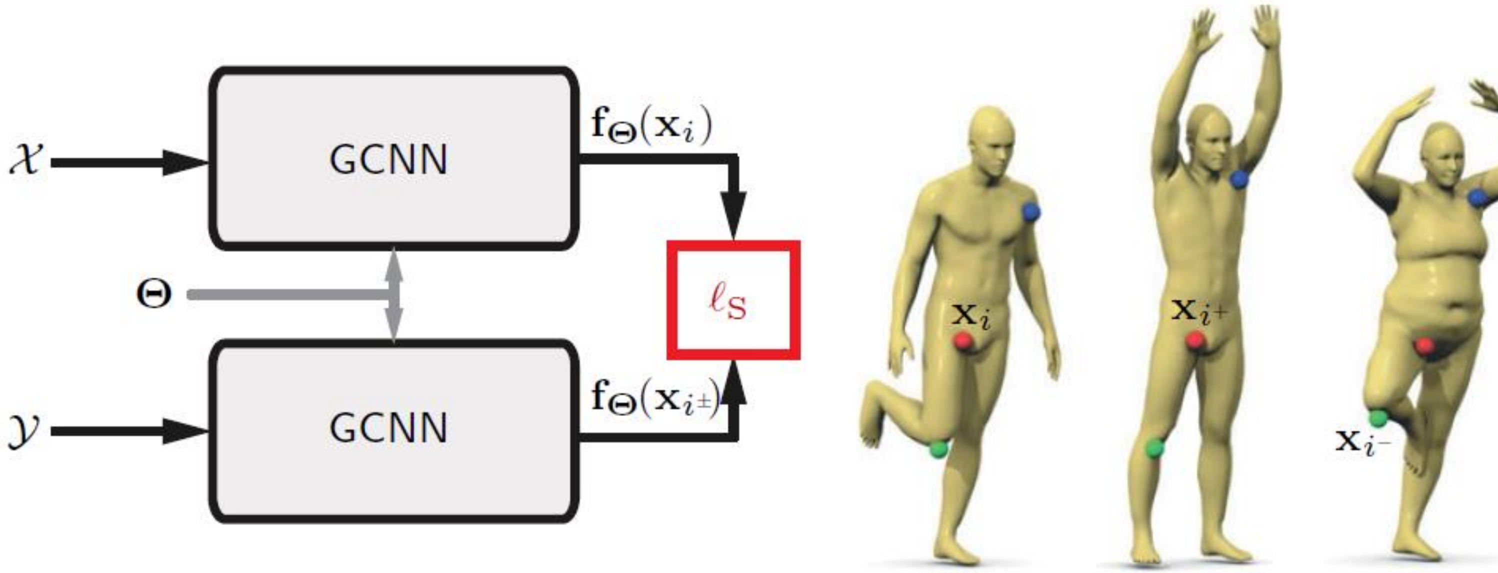
- Spatial convolution with filter g

$$\mathbf{x}'_i \propto \sum_{\ell=1}^{L} g_\ell \underbrace{\sum_{j=1}^{n} w_\ell(\mathbf{u}_{ij})\mathbf{x}_j}_{\text{patch operator}}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is feature at vertex i

Masci et al. 2015; Boscaini et al. 2016; Monti et al. 2017

Learning local descriptors with intrinsic CNN



Training set	positive (i, i^+) and negative (i, i^-) pairs of points
Siamese net	two net instances with shared parameters $\boldsymbol{\Theta}$
Poitwise feature cost	$\ell_{\mathrm{S}}(\boldsymbol{\Theta}) = \gamma \sum_{i,i^+} \|\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{i^+})\|_2^2 + (1-\gamma) \sum_{i,i^-} \left[\mu - \|\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_{i^-})\|_2^2\right]_+$

Masci et al. 2015

HKS descriptor

Distance in the space of local Heat Kernel Signature (HKS) features
(shown is distance from a point on the shoulder marked in white)

Descriptor: Sun, Ovsjanikov, Guibas 2009 (HKS); data: Bronstein, Bronstein, Kimmel 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

WKS descriptor

Distance in the space of local Wave Kernel Signature (WKS) features
(shown is distance from a point on the shoulder marked in white)

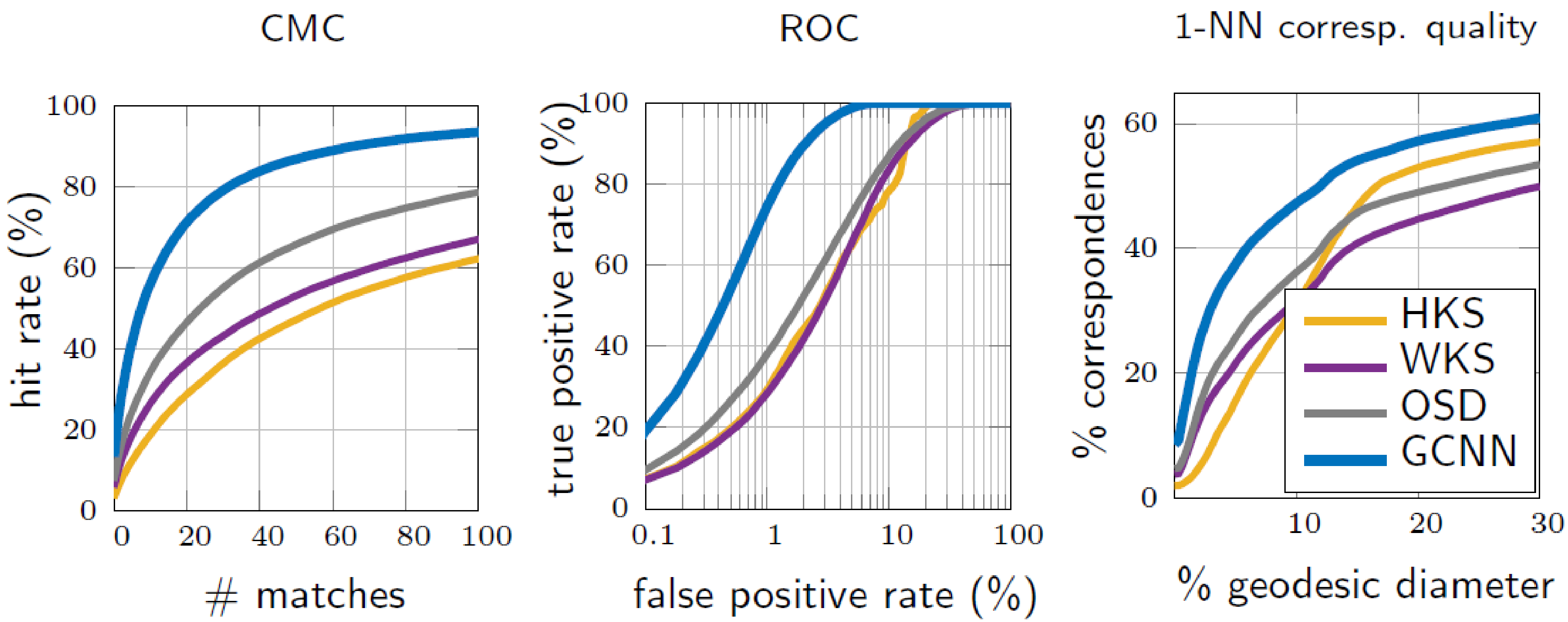
Descriptor: Aubry, Schlickewei, Cremers 2011 (WKS); data: Bronstein, Bronstein, Kimmel 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

Descriptor learned with GCNN

Distance in the space of local GCNN features
(shown is distance from a point on the shoulder marked in white)

Descriptor: Masci et al. 2015 (GCNN); data: Bronstein, Bronstein, Kimmel 2008 (TOSCA); Anguelov et al. 2005 (SCAPE); Bogo et al. 2014 (FAUST)

Descriptor quality comparison



Descriptor performance using symmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

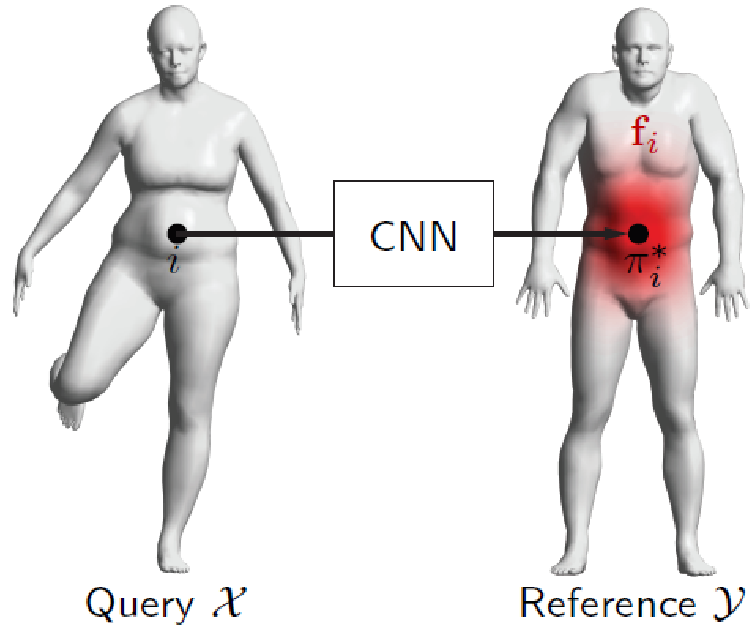
Methods: Sun et al. 2009 (HKS); Aubry et al. 2011 (WKS); Litman, B 2014 (OSD); Masci et al. 2015 (GCNN); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Learning deformation-invariant correspondence

- Groundtruth correspondence $\pi^* : \mathcal{X} \to \mathcal{Y}$ from query shape $\mathcal{X}$ to some reference shape $\mathcal{Y}$
- Correspondence = label each query vertex $i \in \{1, \ldots, n\}$ as reference vertex $\pi_i \in \{1, \ldots, m\}$
- Net output at i after softmax layer

$$\mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_i) = (f_{i1}, \ldots, f_{im})$$

$=$ probability distribution on $\mathcal{Y}$



Minimize on training set the cross entropy between groundtruth correspondence and output probability distribution w.r.t. net parameters $\mathbf{\Theta}$

$$\min_{\mathbf{\Theta}} \sum_{i=1}^{n} H(\boldsymbol{\delta}_{\pi_i^*}, \mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_i))$$

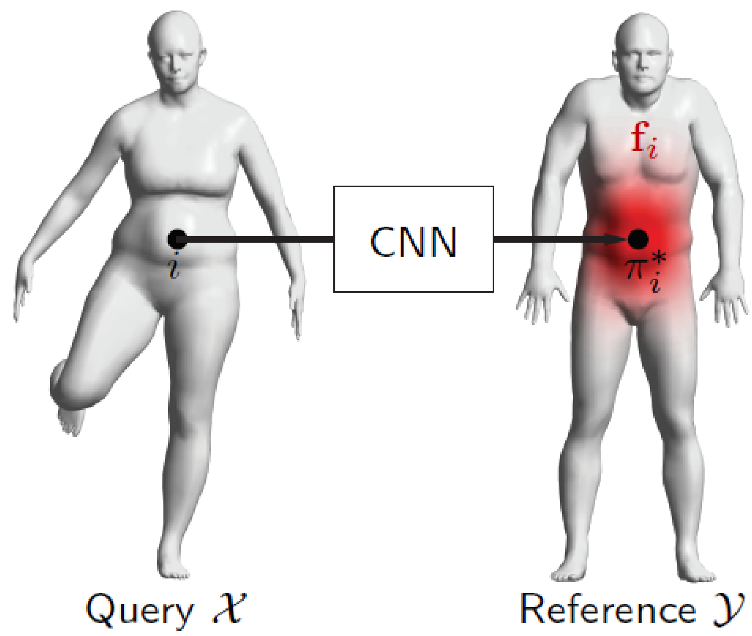
Rodolà et al. 2014; Masci et al. 2015

Learning deformation-invariant correspondence

- Groundtruth correspondence $\pi^* : \mathcal{X} \rightarrow \mathcal{Y}$ from query shape $\mathcal{X}$ to some reference shape $\mathcal{Y}$
- Correspondence = label each query vertex $i \in \{1, \ldots, n\}$ as reference vertex $\pi_i \in \{1, \ldots, m\}$
- Net output at i after softmax layer

$$\mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i) = (f_{i1}, \ldots, f_{im})$$

$=$ probability distribution on $\mathcal{Y}$

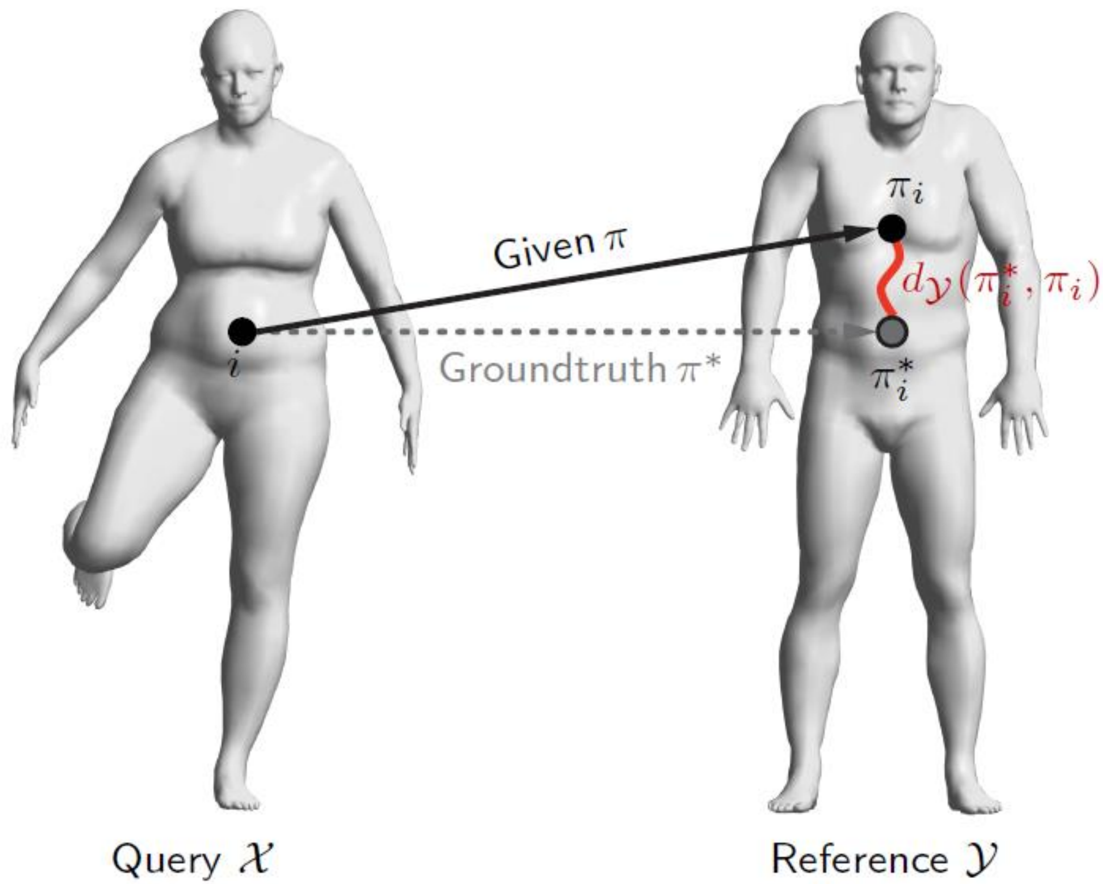


Minimize on training set the cross entropy between groundtruth correspondence and output probability distribution w.r.t. net parameters $\boldsymbol{\Theta}$

$$\min_{\boldsymbol{\Theta}} \sum_{i=1}^{n} H(\boldsymbol{\delta}_{\pi_i^*}, \mathbf{f}_{\boldsymbol{\Theta}}(\mathbf{x}_i))$$

Rodolà et al. 2014; Masci et al. 2015

Correspondence evaluation: Princeton benchmark

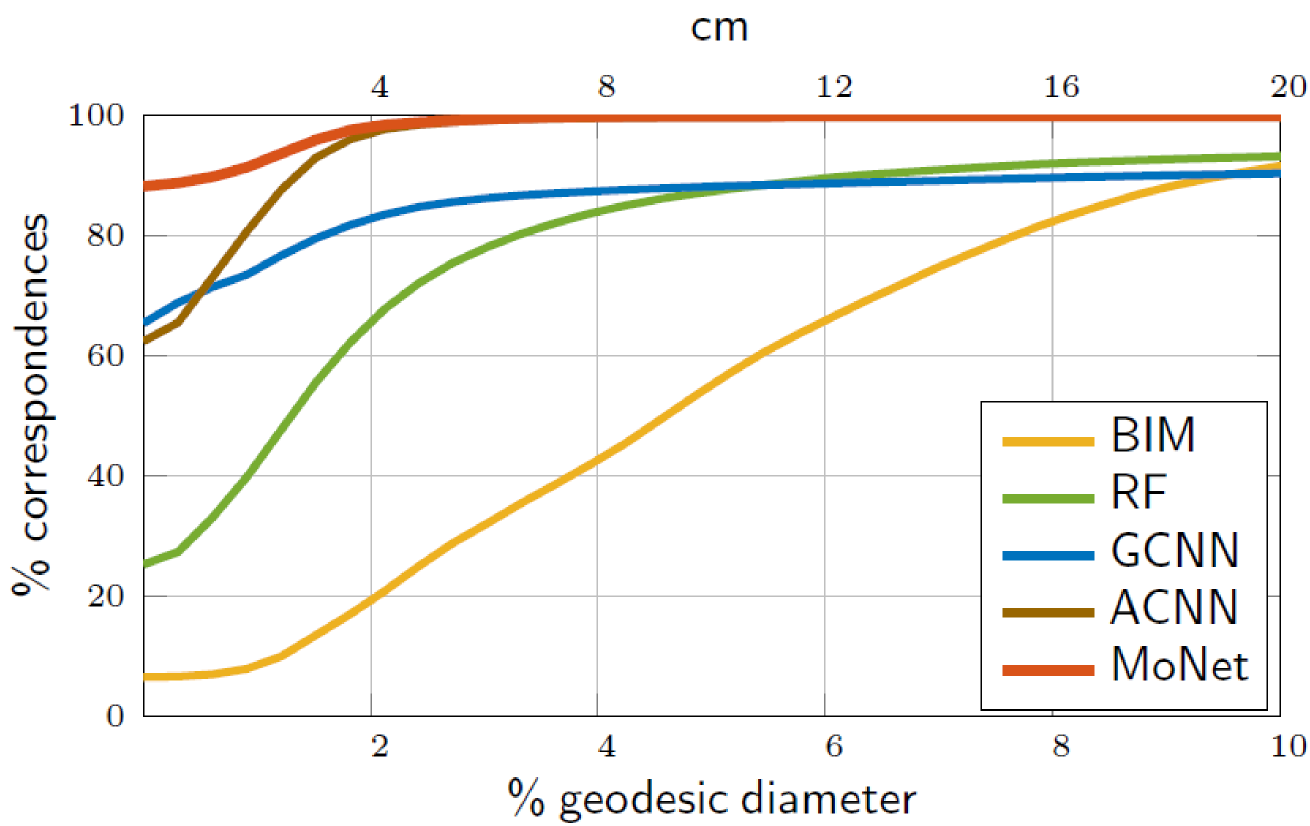


Pointwise correspondence error = geodesic distance from the groundtruth

$$\epsilon_i = d_{\mathcal{Y}}(\pi_i^*, \pi_i)$$

Kim et al. 2011

Correspondence quality comparison



Correspondence evaluated using asymmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

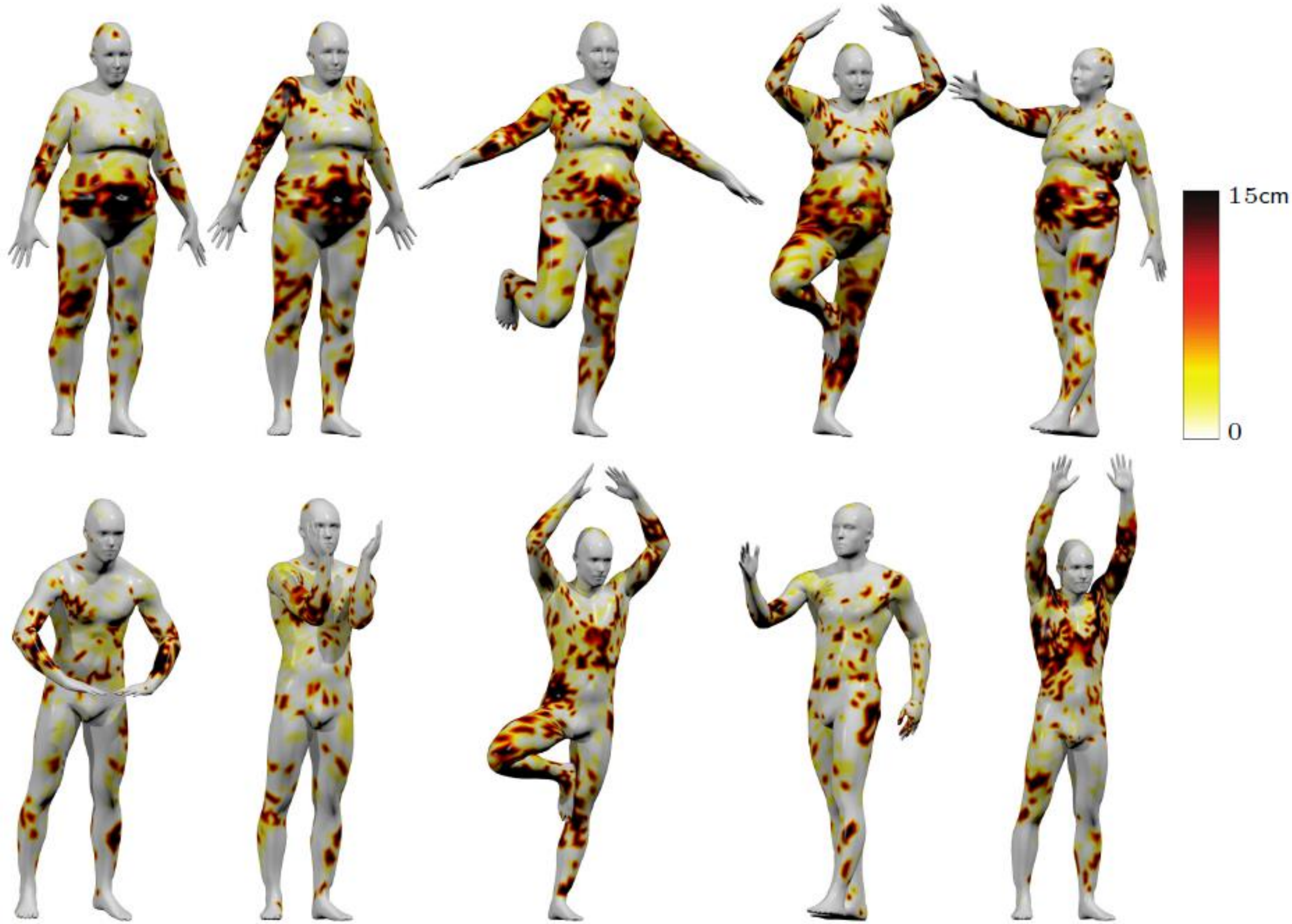
Methods: Kim et al. 2011 (BIM); Rodolà et al. 2014 (RF); Boscaini et al. 2015 (ADD); Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Shape correspondence error: Blended Intrinsic Map

Pointwise correspondence error (geodesic distance from groundtruth)

Kim, Lipman, Funkhouser 2011

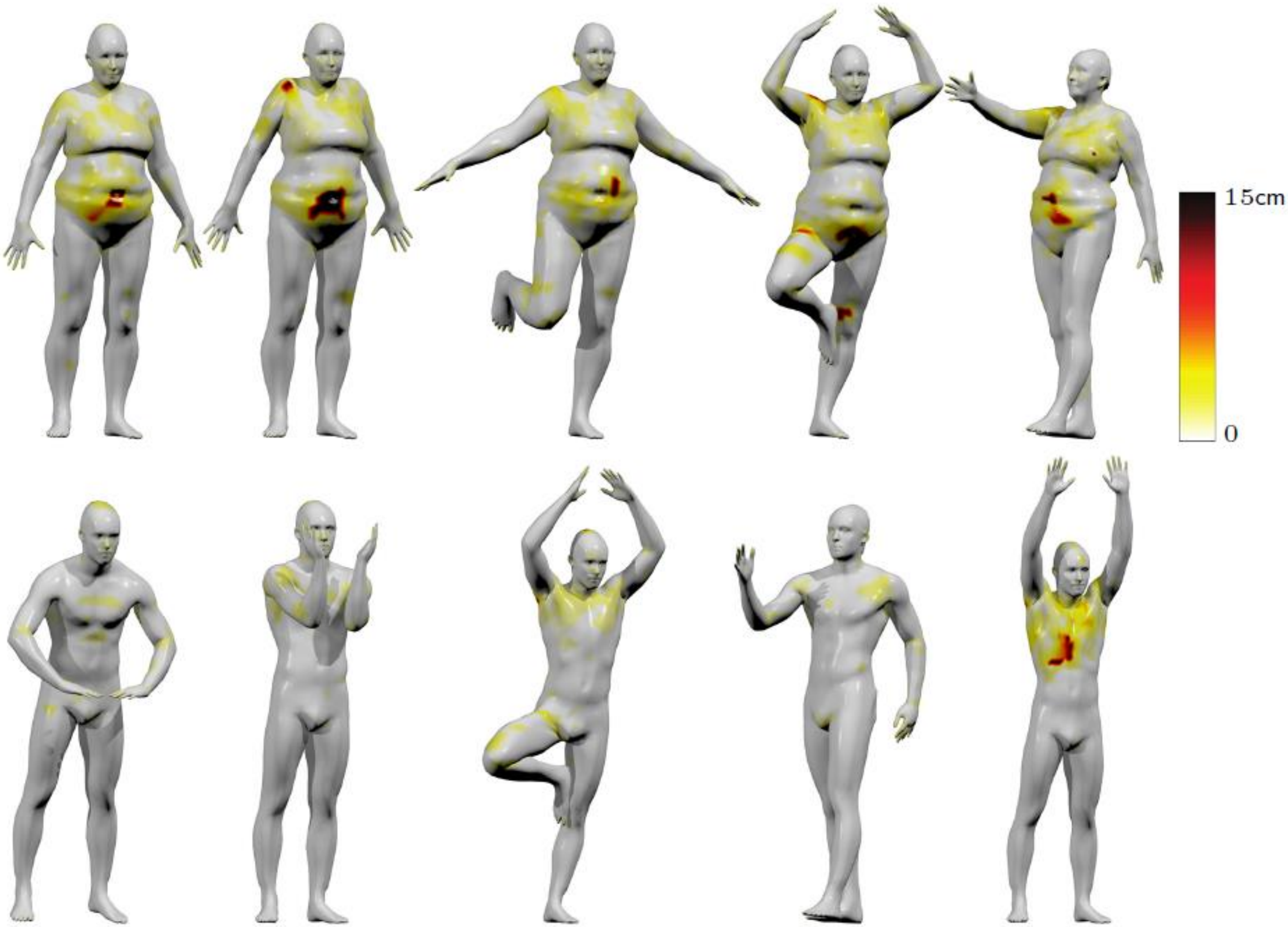
Shape correspondence error: Geodesic CNN



Pointwise correspondence error (geodesic distance from groundtruth)

Masci et al. 2015

Shape correspondence error: Anisotropic CNN



Pointwise correspondence error (geodesic distance from groundtruth)

Boscaini et al. 2016

Shape correspondence error: MoNet

Pointwise correspondence error (geodesic distance from groundtruth)

Monti et al. 2016

Shape correspondence visualization: MoNet

Reference

Texture transferred from reference to query shapes

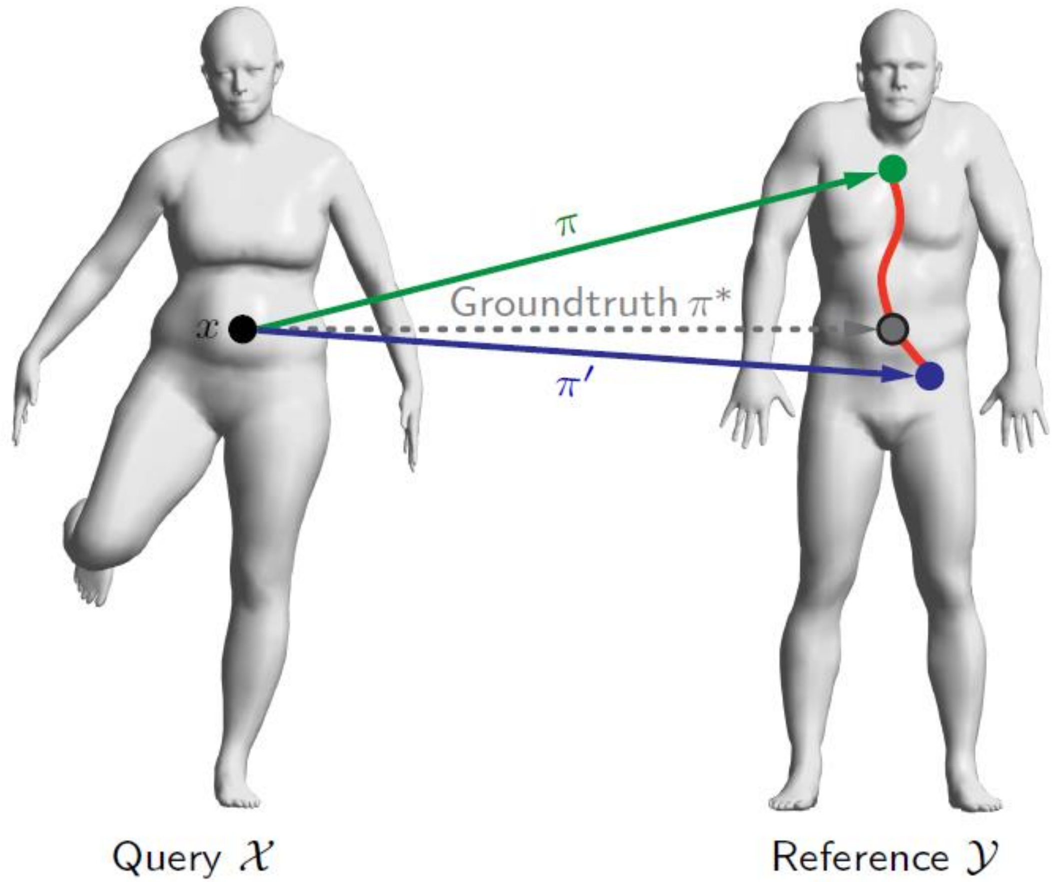
Monti et al. 2016

Correspondence on range images: MoNet

Pointwise correspondence error (geodesic distance from groundtruth)

Monti et al. 2016

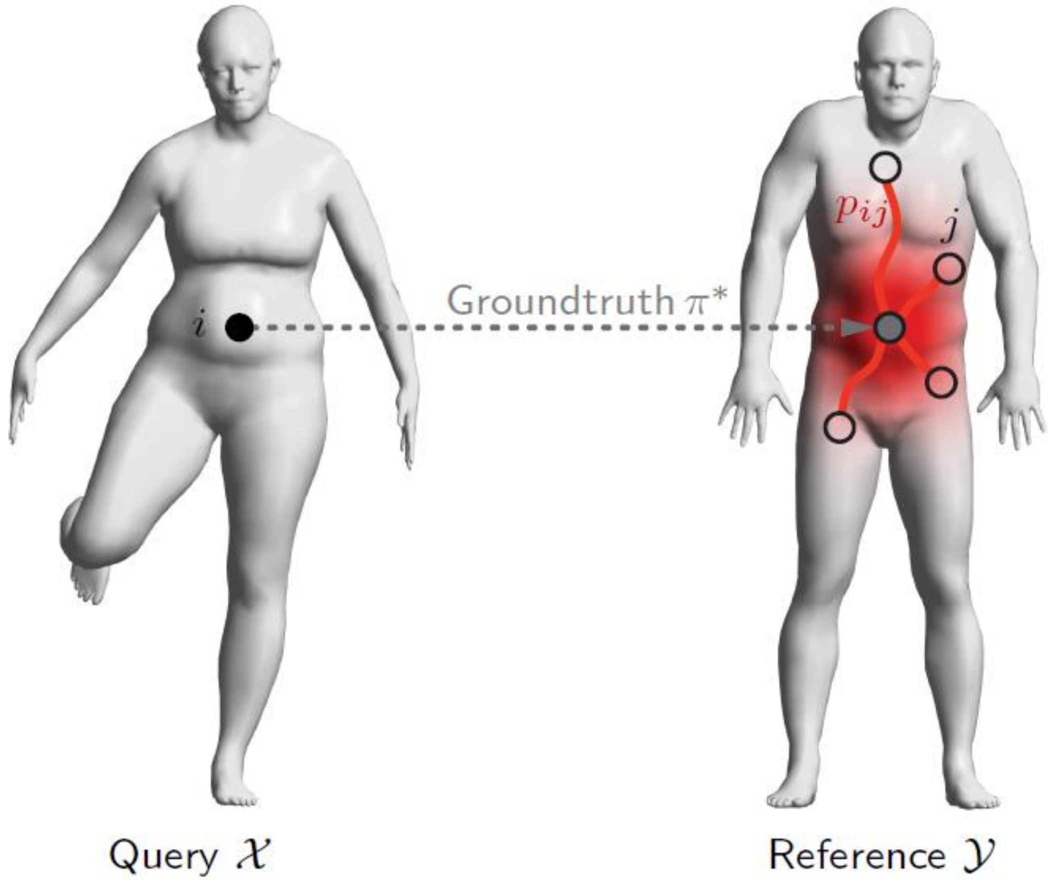
Correspondence as classification problem, revisited



Classification cost considers equally correspondences that deviate from the groundtruth (no matter how far)

Kim et al. 2011

Soft correspondence error

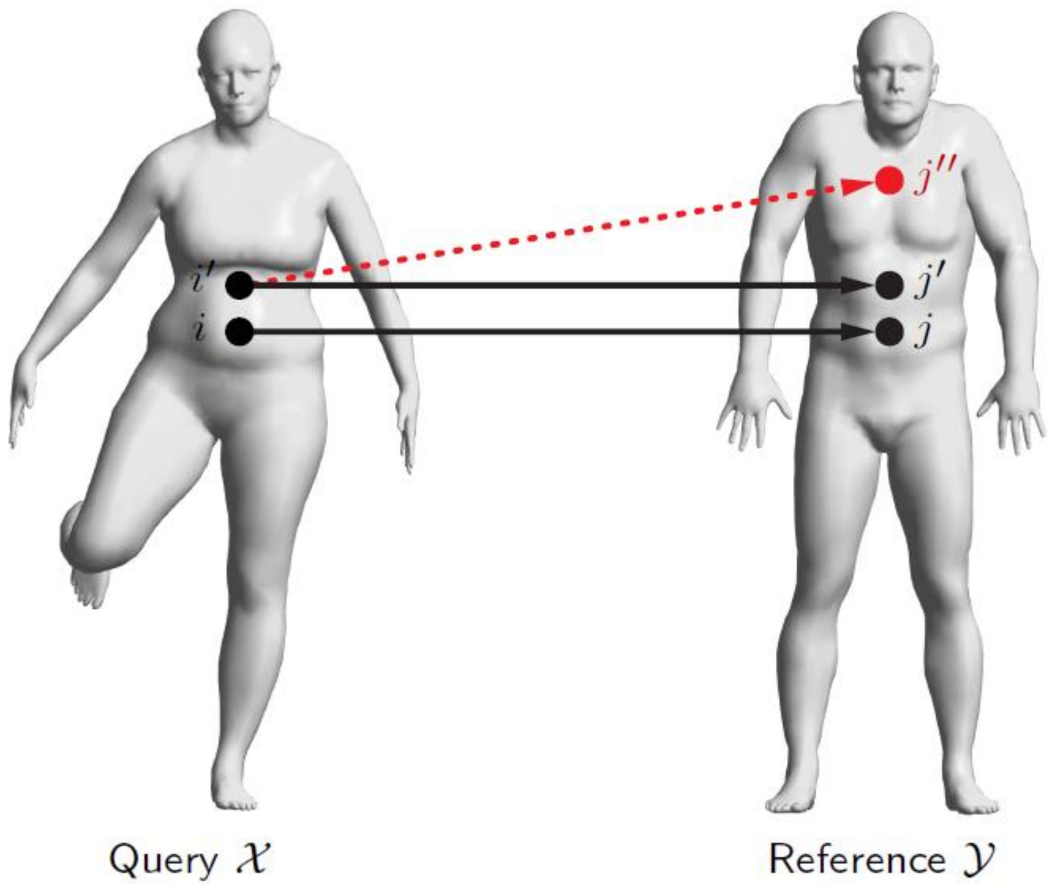


Soft correspondence error = probability-weighted geodesic distance from the groundtruth

$$\bar{\epsilon}_i = \sum_{j=1}^{m} p_{ij} d_{\mathcal{Y}}(\pi_i^*, j)$$

Kovnatsky et al. 2015; Litany et al. 2017

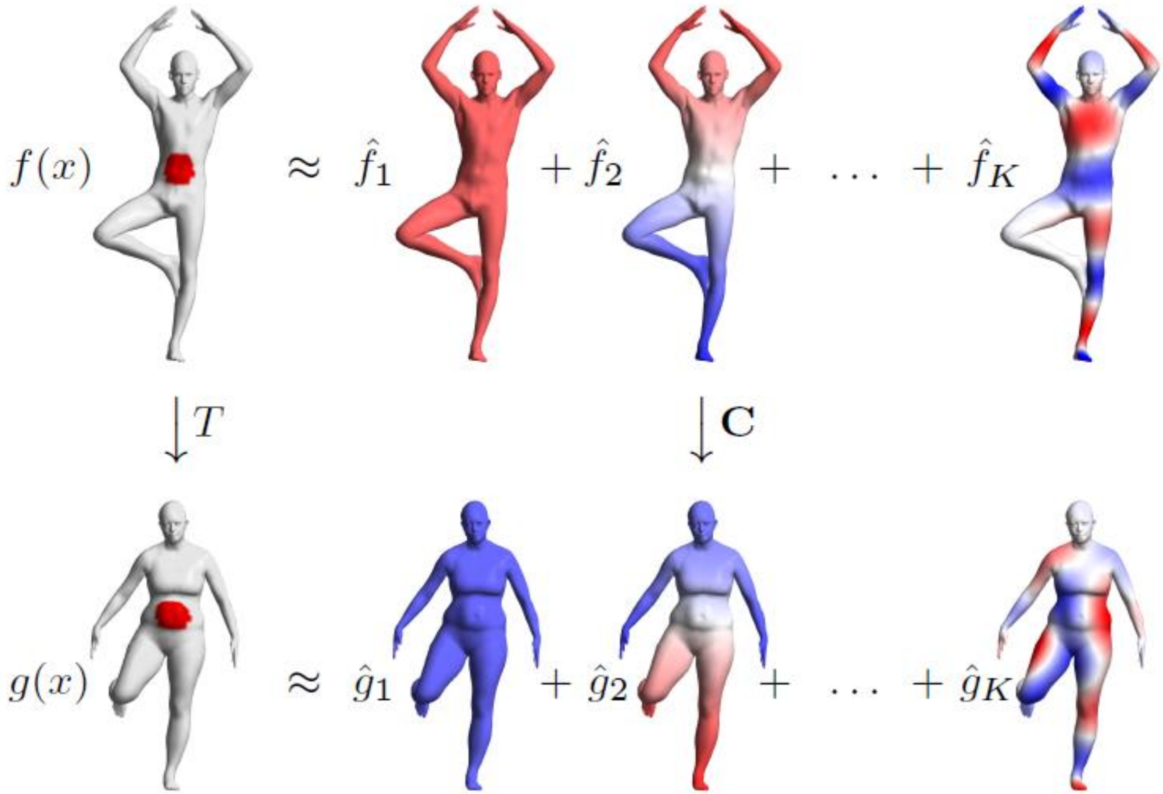
Pointwise vs Structured learning



Nearby points i, i' on query shape are **not guaranteed** to map to nearby points j, j' on reference shape at **test time**

Litany et al. 2017

Functional maps: spectral domain

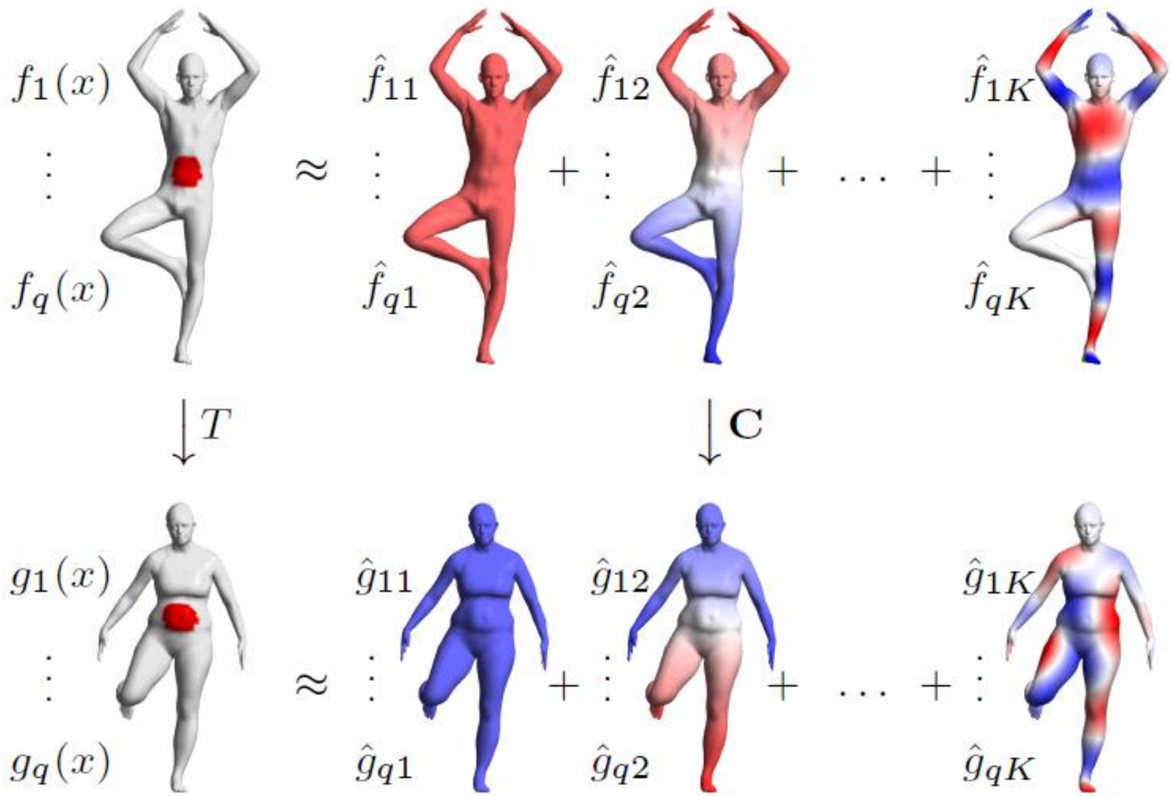


Functional correspondence T = linear map $\mathbf{C}$ between Fourier coefficients

$$\hat{\mathbf{g}}^\top = \hat{\mathbf{f}}^\top \mathbf{C}$$

Ovsjanikov et al. 2012

Functional maps: spectral domain

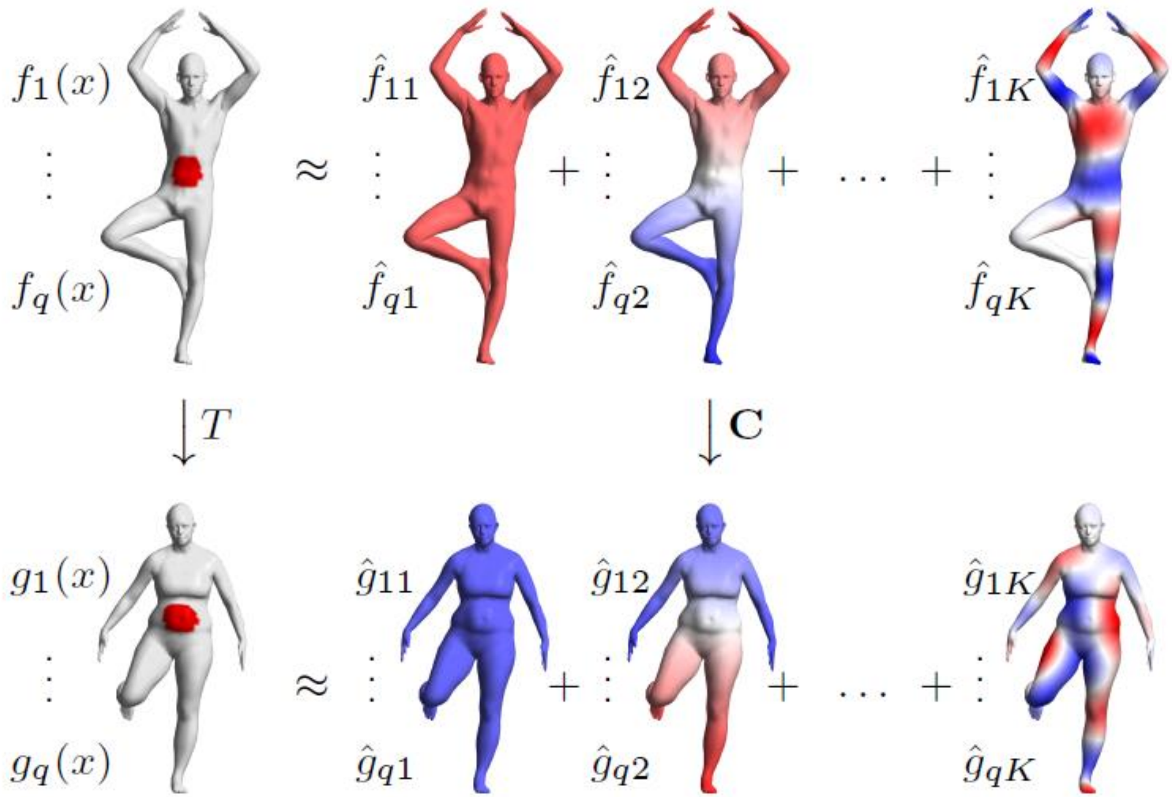


Recover correspondence from $q \geq k$ dimensional pointwise features

$$\begin{pmatrix} \hat{g}_{11} & \hat{g}_{12} & \cdots & \hat{g}_{1K} \\ \vdots & \vdots & & \vdots \\ \hat{g}_{q1} & \hat{g}_{q2} & \cdots & \hat{g}_{qK} \end{pmatrix} = \begin{pmatrix} \hat{f}_{11} & \hat{f}_{12} & \cdots & \hat{f}_{1K} \\ \vdots & \vdots & & \vdots \\ \hat{f}_{q1} & \hat{f}_{q2} & \cdots & \hat{f}_{qK} \end{pmatrix} \mathbf{C}$$

Ovsjanikov et al. 2012

Functional maps: spectral domain

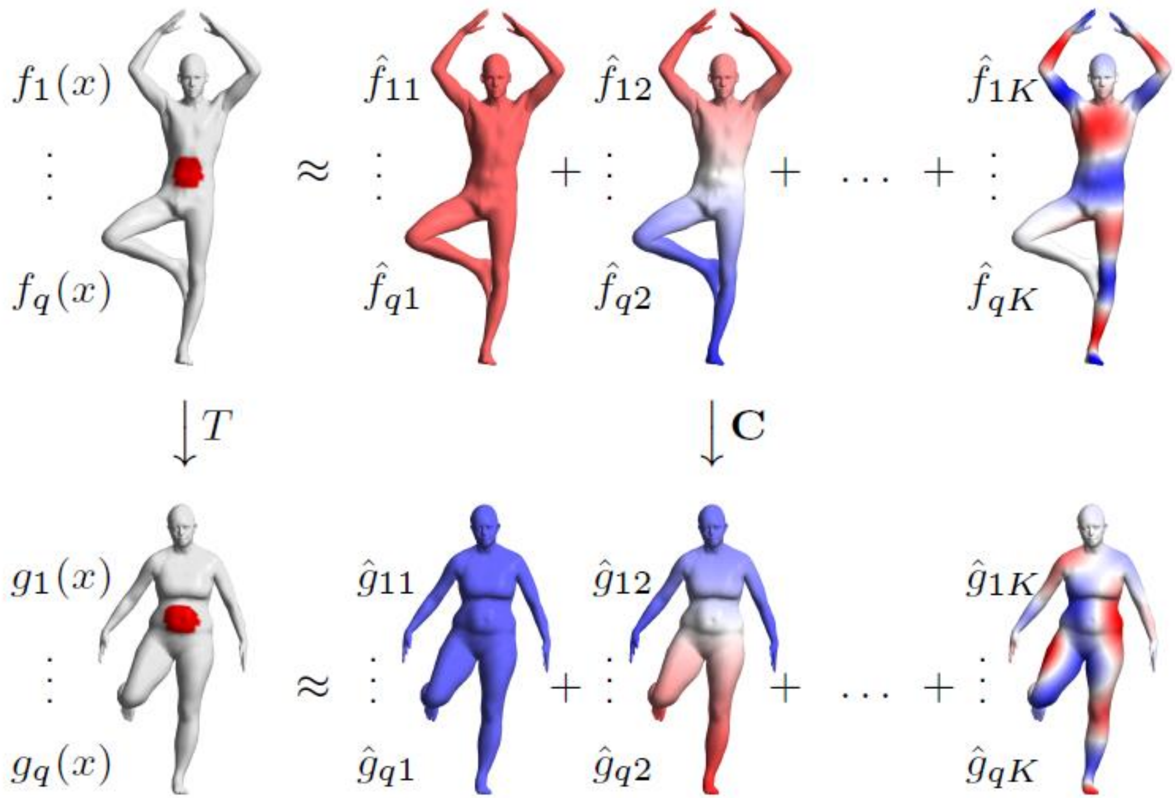


Recover correspondence from $q \geq k$ dimensional pointwise features

$$\hat{\mathbf{G}} = \hat{\mathbf{F}}\mathbf{C}$$

Ovsjanikov et al. 2012

Functional maps: spectral domain

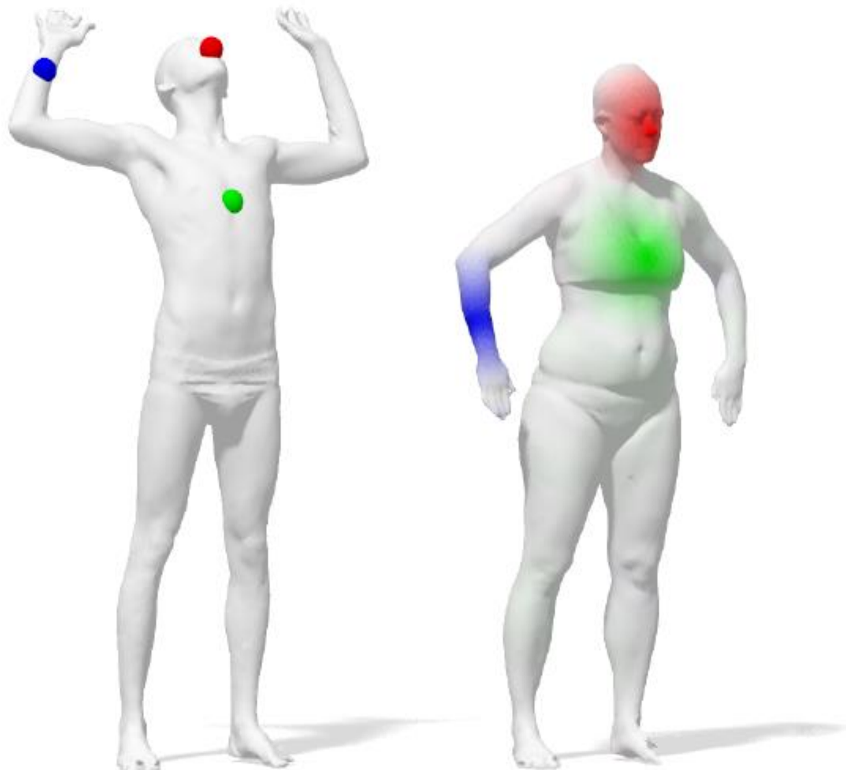


Recover correspondence from $q \geq k$ dimensional pointwise features

$$\mathbf{C}^* = \underset{\mathbf{C}}{\operatorname{argmin}} \|\hat{\mathbf{F}}\mathbf{C} - \hat{\mathbf{G}}\|_{\mathrm{F}}^2$$

Ovsjanikov et al. 2012

Functional maps: spatial domain

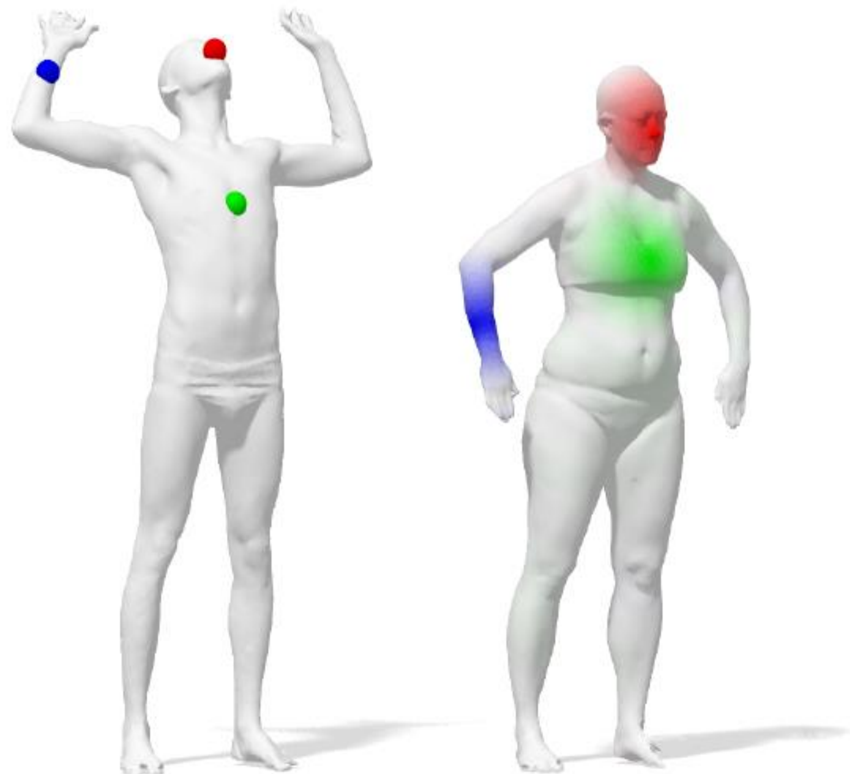


Rank-K approximation of spatial correspondence

$$\mathbf{T} \approx \mathbf{\Psi} \mathbf{C} \mathbf{\Phi}^\top$$

Ovsjanikov et al. 2012

Functional maps: spatial domain

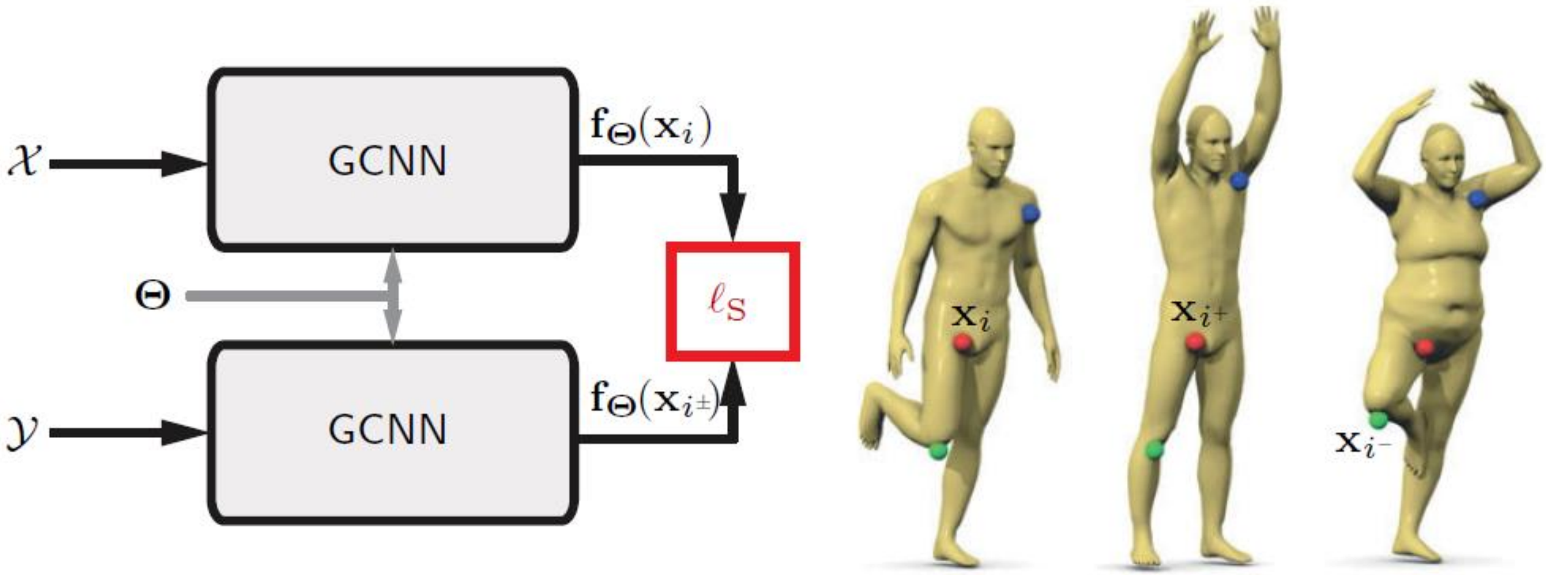


Probability p_{ij} of point j mapping to i

$$\mathbf{P} \approx |\mathbf{\Psi}\mathbf{C}\mathbf{\Phi}^\top|_{\|\cdot\|}$$

Ovsjanikov et al. 2012

Siamese metric learning



Training set — positive (i, i^+) and negative (i, i^-) pairs of points

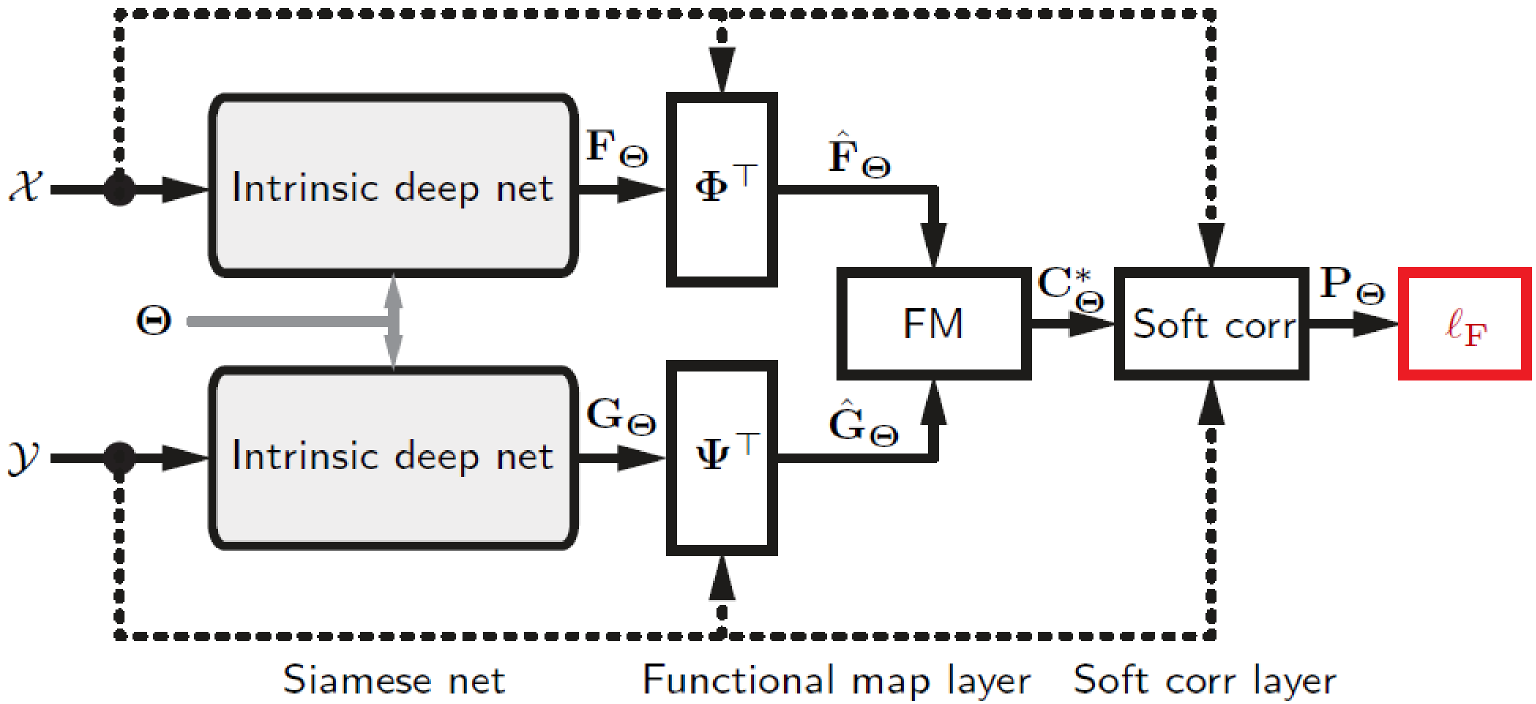
Siamese net — two net instances with shared parameters $\mathbf{\Theta}$

Poitwise feature cost

$$\ell_{\mathrm{S}}(\mathbf{\Theta}) = \gamma \sum_{i,i^+} \|\mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_{i^+})\|_2^2 + (1-\gamma) \sum_{i,i^-} \left[\mu - \|\mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_i) - \mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}_{i^-})\|_2^2\right]_+$$

Masci et al. 2015

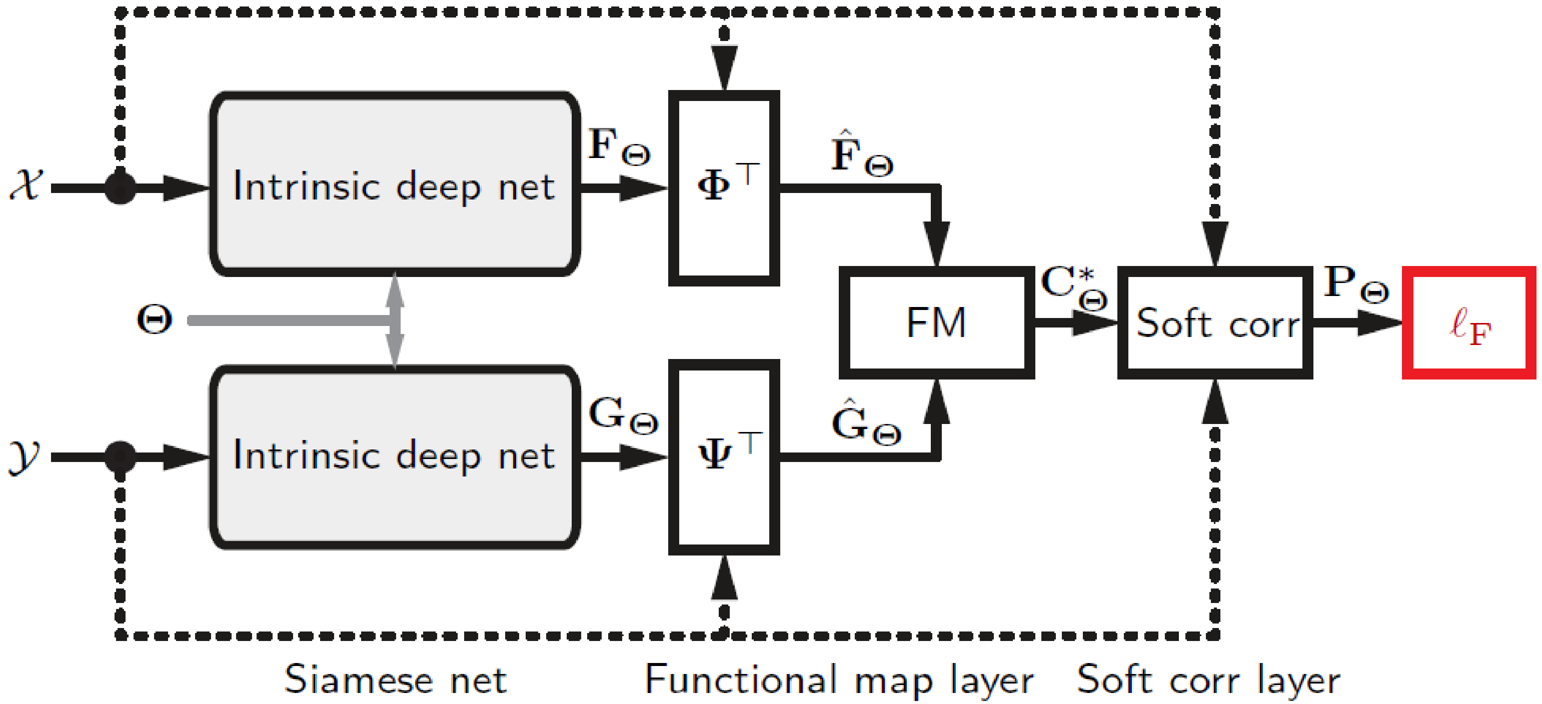
Structured correspondence with FMNet



Siamese net	two net instances with shared parameters $\boldsymbol{\Theta}$
Functional map layer	$\mathbf{C}^*_{\boldsymbol{\Theta}} = \underset{\mathbf{C}}{\operatorname{argmin}} \|\hat{\mathbf{F}}_{\boldsymbol{\Theta}}\mathbf{C} - \hat{\mathbf{G}}_{\boldsymbol{\Theta}}\|^2_{\mathrm{F}}$
Soft correspondence layer	$\mathbf{P}_{\boldsymbol{\Theta}} = \lvert \boldsymbol{\Psi}\mathbf{C}_{\boldsymbol{\Theta}}\boldsymbol{\Phi}^\top \rvert_{\|\cdot\|}$
Soft error cost	$\ell_{\mathrm{F}}(\boldsymbol{\Theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{\boldsymbol{\Theta},ij}\, d_{\mathcal{Y}}(\pi_i^*, j)$

Litany et al. 2017

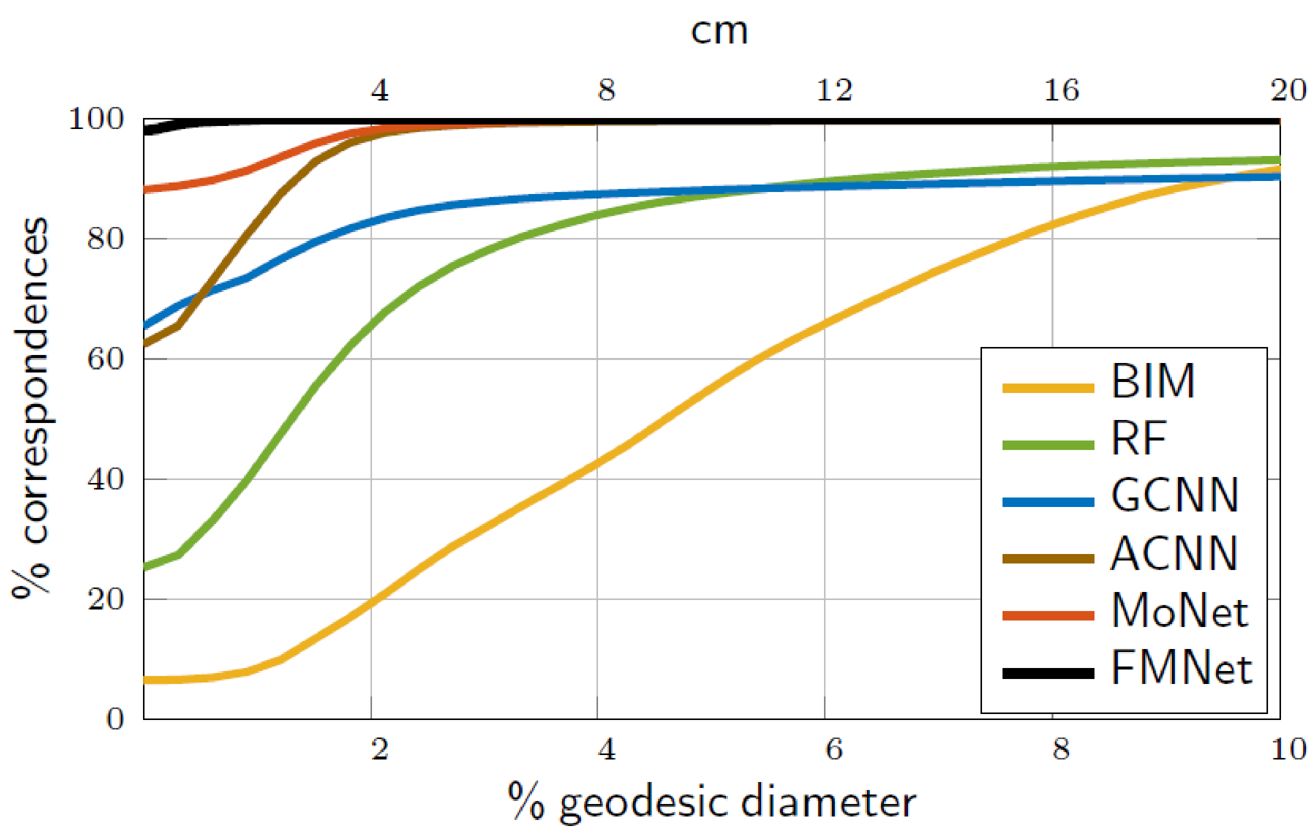
Structured correspondence with FMNet



Siamese net	two net instances with shared parameters $\boldsymbol{\Theta}$
Functional map layer	$\mathbf{C}^*_{\boldsymbol{\Theta}} = \hat{\mathbf{F}}^\dagger_{\boldsymbol{\Theta}} \hat{\mathbf{G}}_{\boldsymbol{\Theta}}$
Soft correspondence layer	$\mathbf{P}_{\boldsymbol{\Theta}} = \lvert \boldsymbol{\Psi} \mathbf{C}_{\boldsymbol{\Theta}} \boldsymbol{\Phi}^\top \rvert_{\lVert \cdot \rVert}$
Soft error cost	$\ell_{\mathrm{F}}(\boldsymbol{\Theta}) = \lVert \mathbf{P}_{\boldsymbol{\Theta}} \circ \mathbf{D}_{\mathcal{Y}} \rVert$

Litany et al. 2017

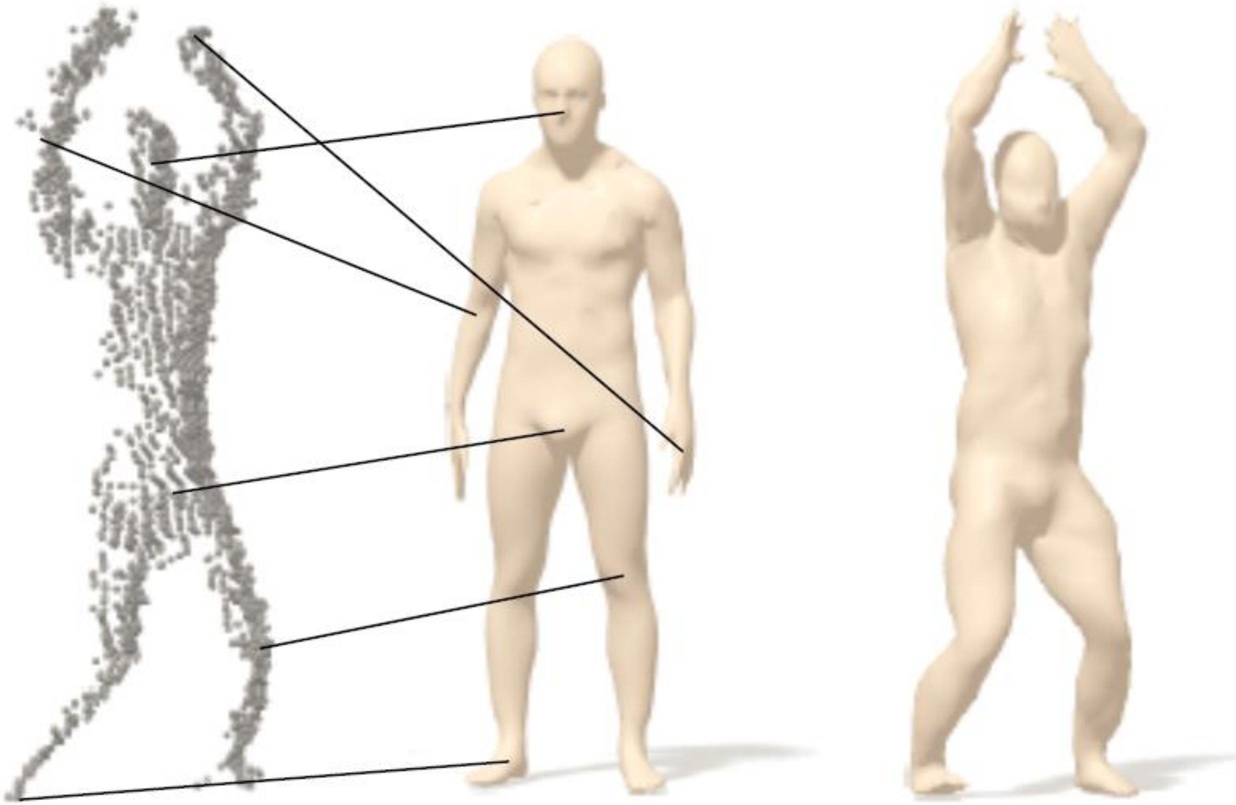
Correspondence quality comparison



Correspondence evaluated using asymmetric Princeton benchmark
(training and testing: disjoint subsets of FAUST)

Methods: Kim et al. 2011 (BIM); Rodolà et al. 2014 (RF); Boscaini et al. 2015 (ADD); Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet); Litany et al. 2017 (FMNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

3D shape analysis and synthesis



Input
3D scan

Correspondence to
Reference shape

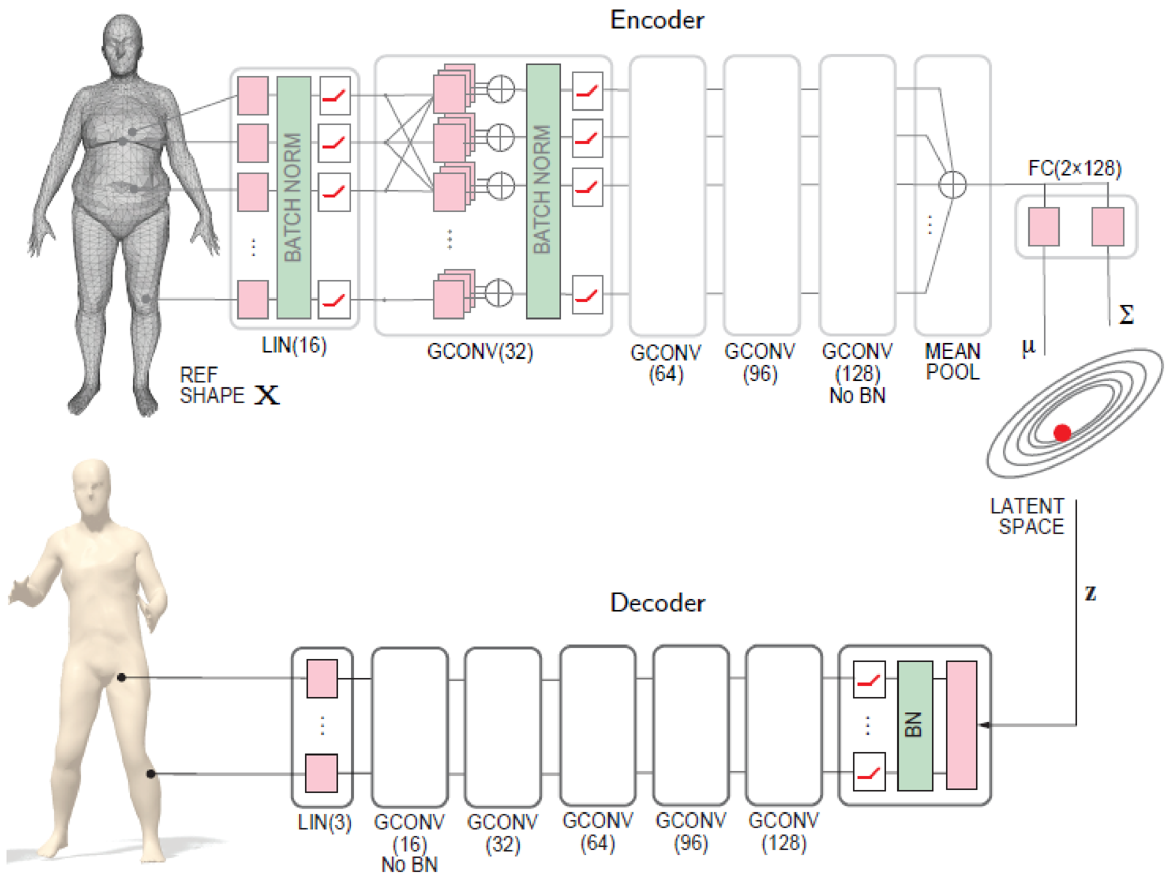
Deformation of
reference shape

Intrinsic Variational Autoencoder (VAE)

Minimize

$$\|\mathrm{Dec}(\mathrm{Enc}(\mathbf{X})) - \mathbf{X}\|_{\mathrm{F}}$$
$$+\lambda D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{X})\|p(\mathbf{z}))$$

w.r.t. net parameters



$\mathrm{Dec}(\mathrm{Enc}(\mathbf{X}))$

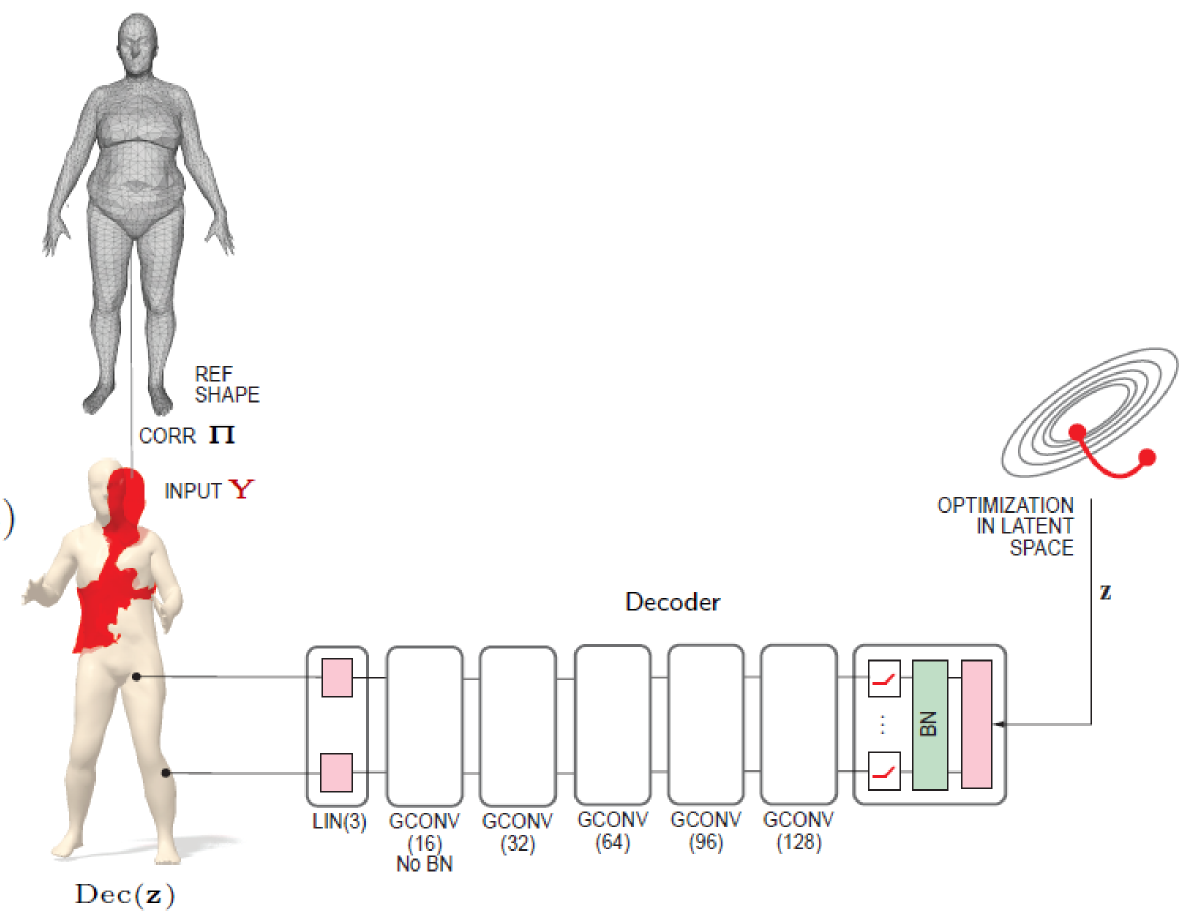
Litany et al. 2017; training on Dynamic FAUST (Bogo et al. 2017)

Shape completion

Minimize

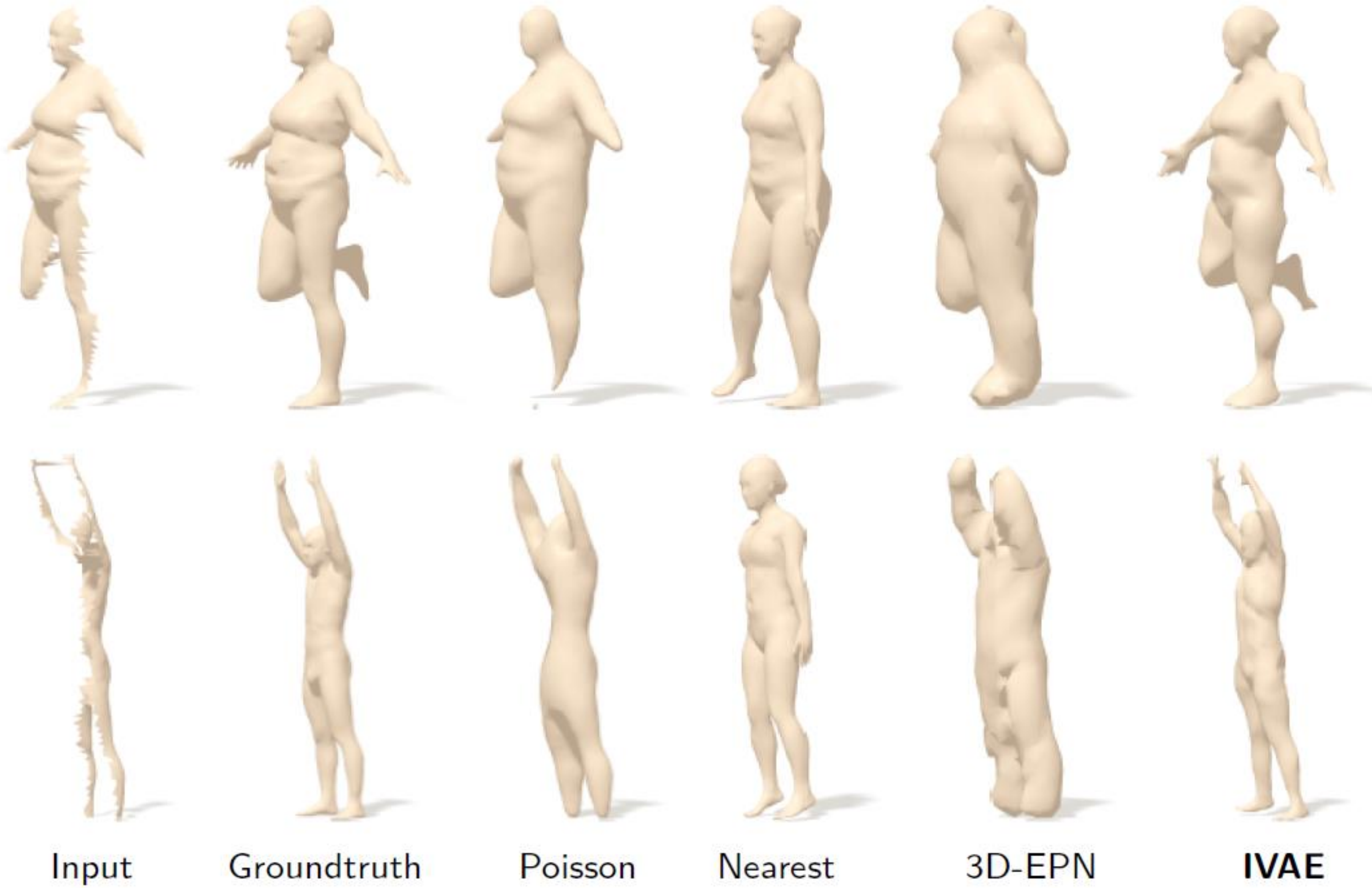
$$\|\mathrm{Dec}(\mathbf{z})\mathbf{\Pi} - \mathbf{T}\mathbf{Y}\|_{\mathrm{F}}$$

in alternating manner w.r.t. $\mathbf{z}$ and $\mathbf{T} \in \mathrm{SO}(3)$



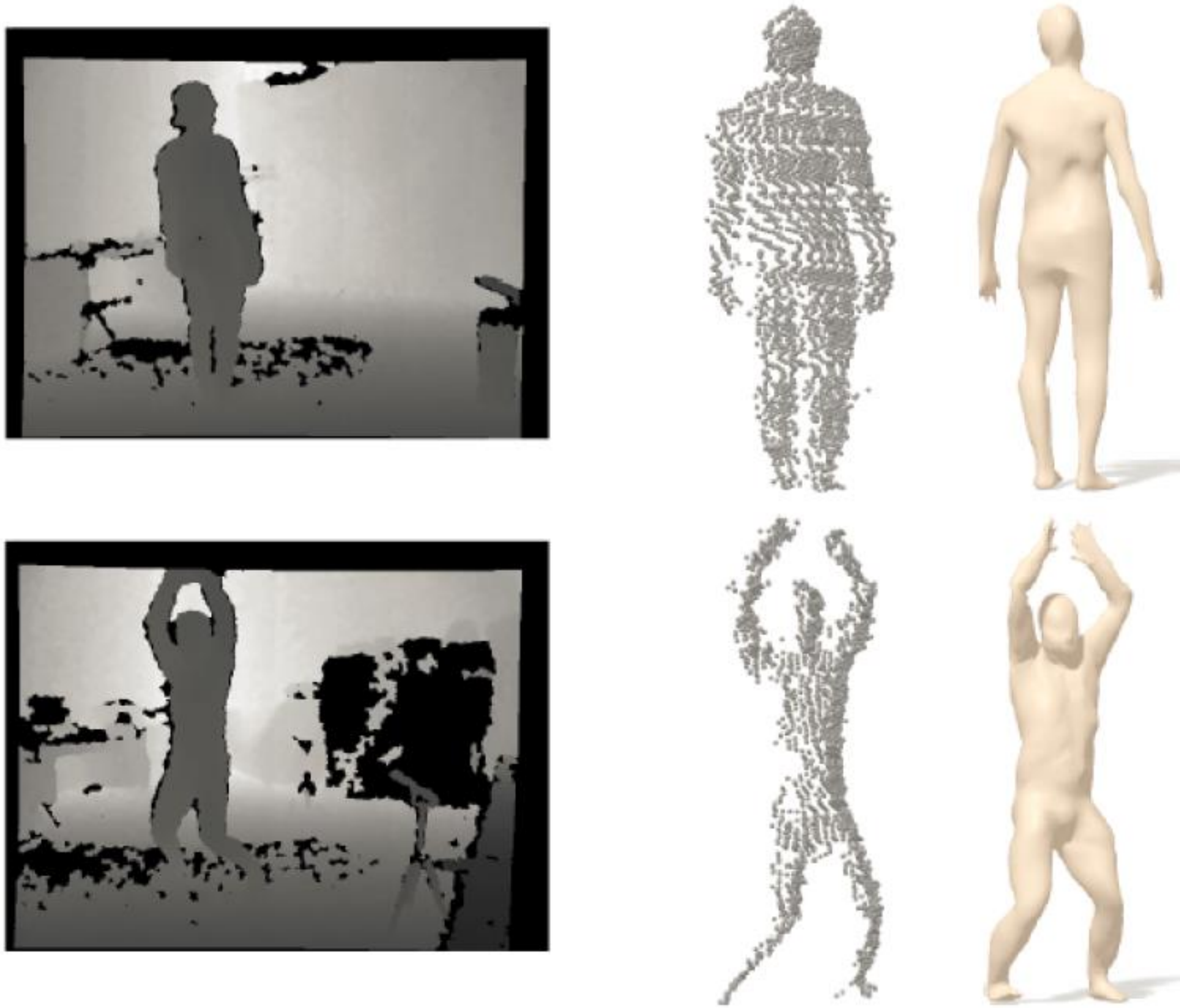
Litany et al. 2017

Shape completion comparison



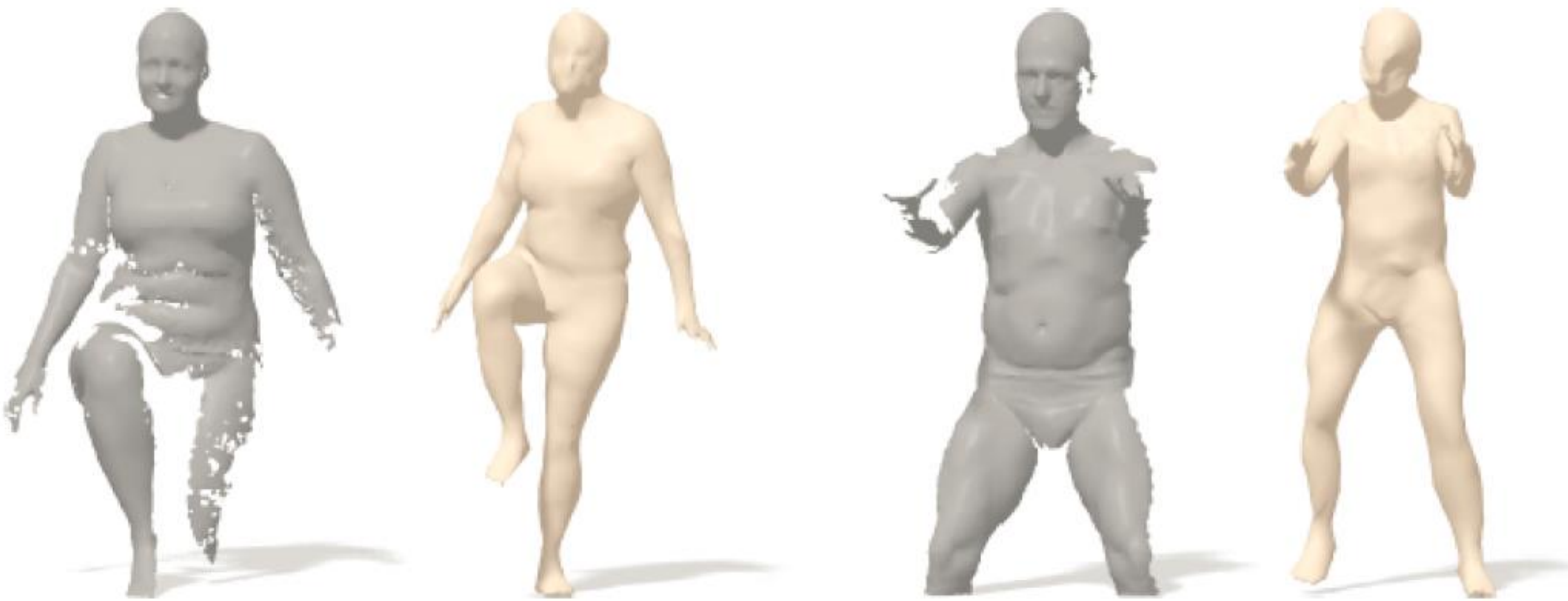
Methods: Litany et al. 2017; Dai et al. 2016 (3D-EPN); Kazhdan et al. 2013 (Poisson)

Shape completion examples



Litany et al. 2017; data: Ofli et al. 2014 (MHAD)

Shape completion examples



Litany et al. 2017; data: Bogo et al. 2014 (FAUST)