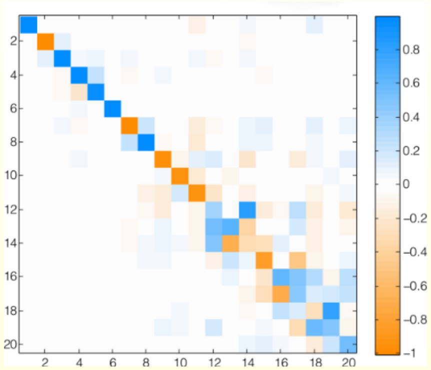
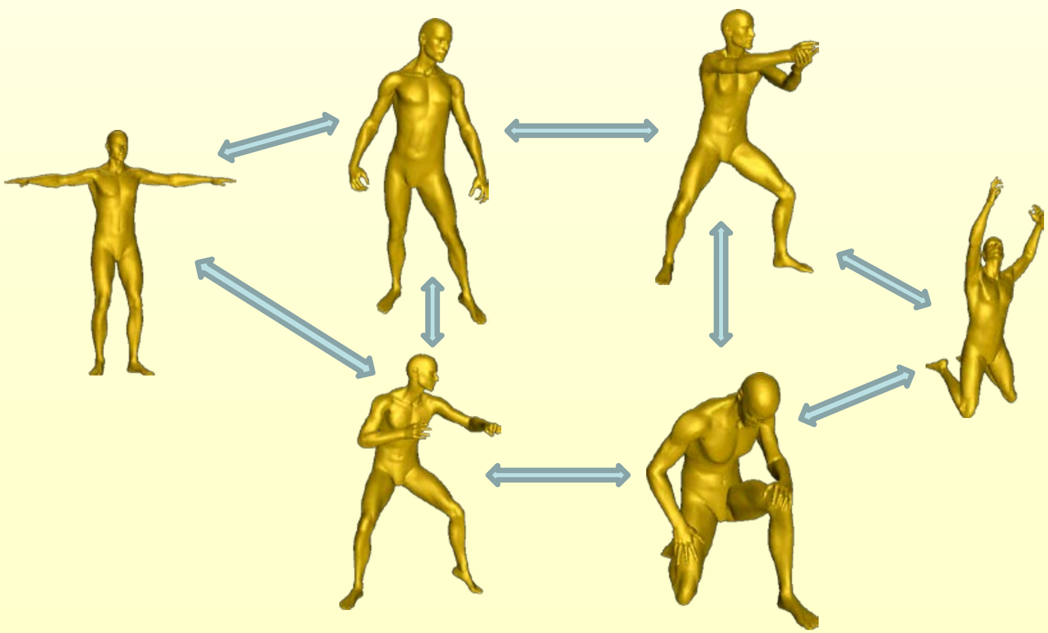
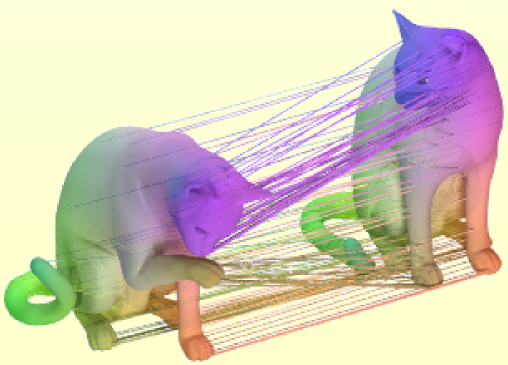
Shape Differences, Functional Map Networks, Applications

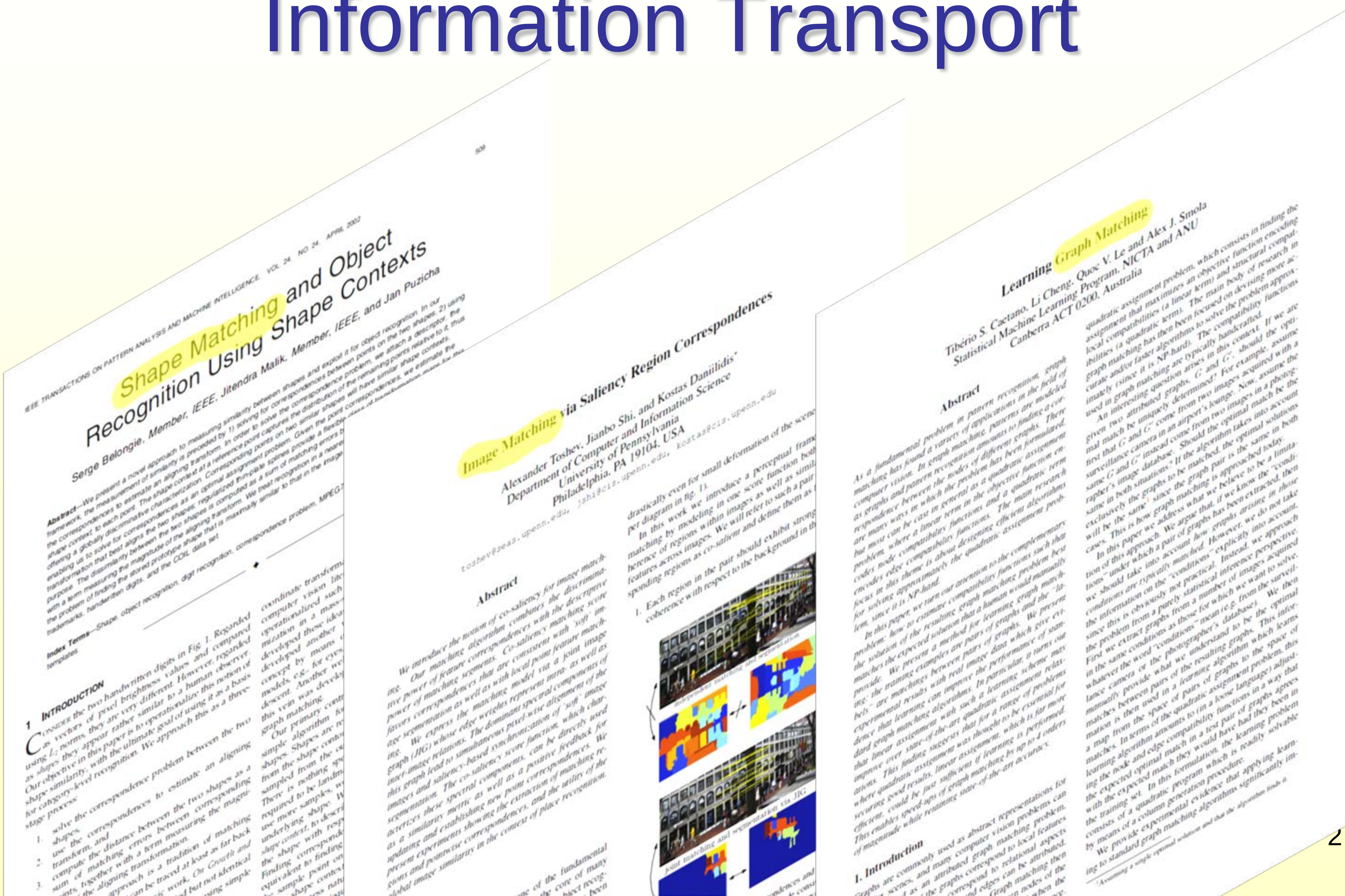
Leonidas Guibas
Stanford University





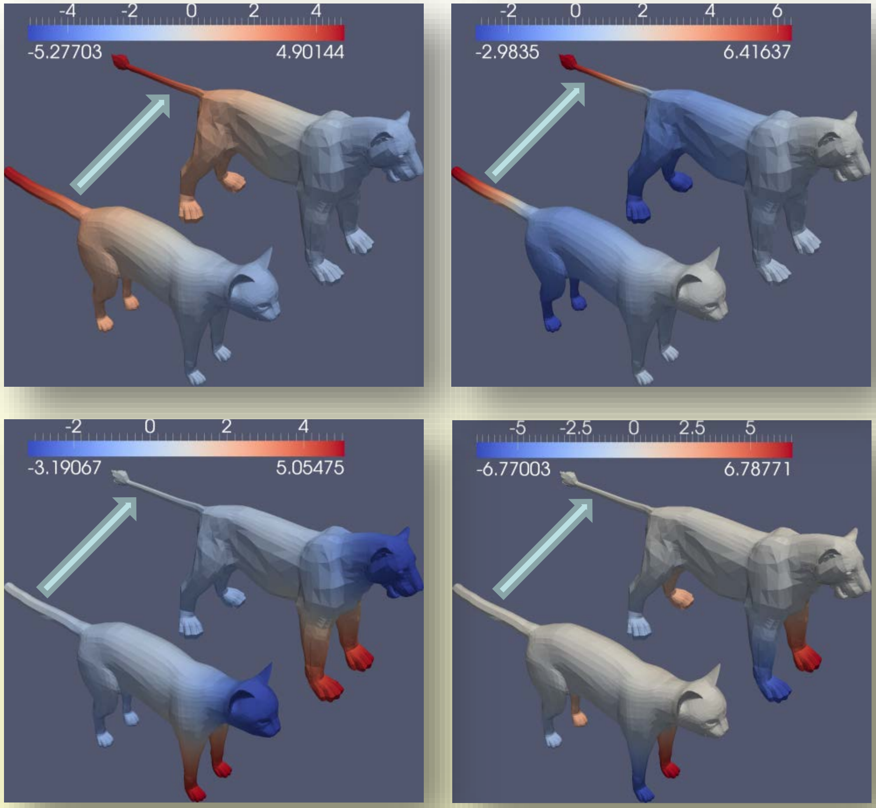


Matching Enables Information Transport

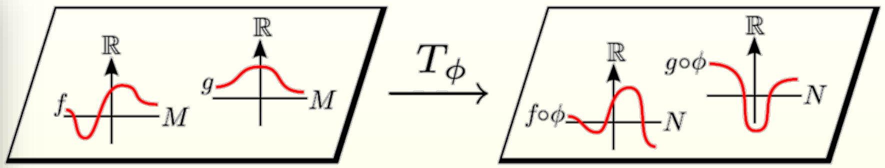


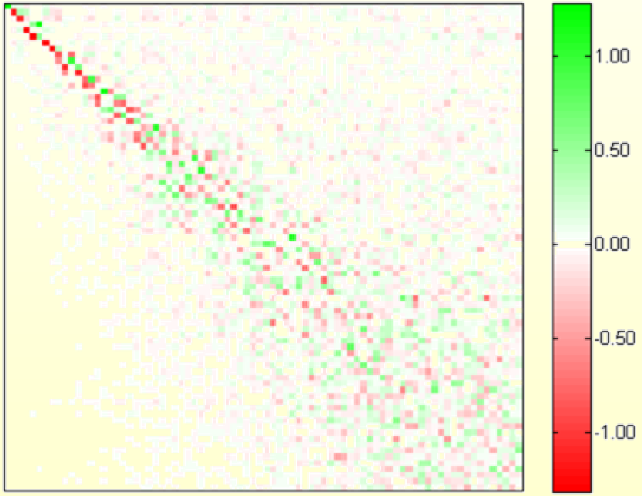
Functional Maps as Information Transporters

from cat to lion



Functions on cat are transferred to lion using T_φ



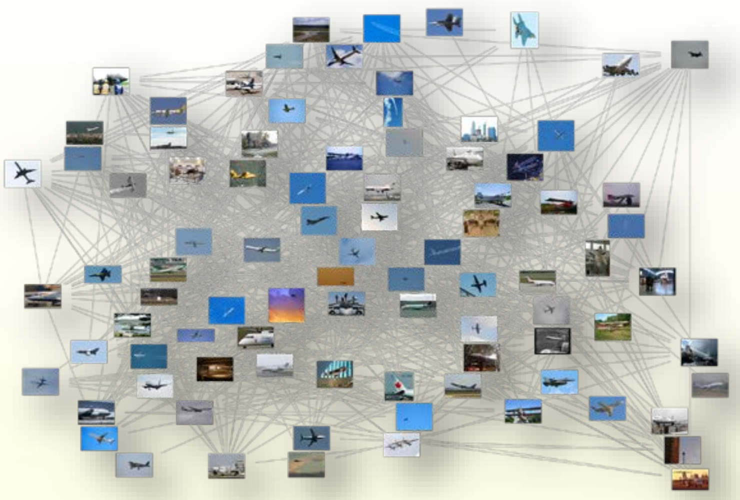


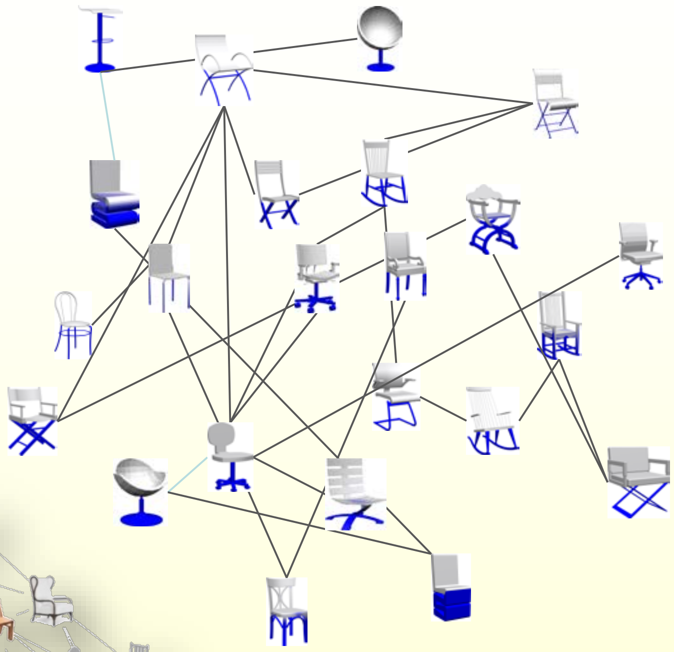
T_φ is a linear operator (matrix)

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

The Network View: Information Transport Between Visual Data

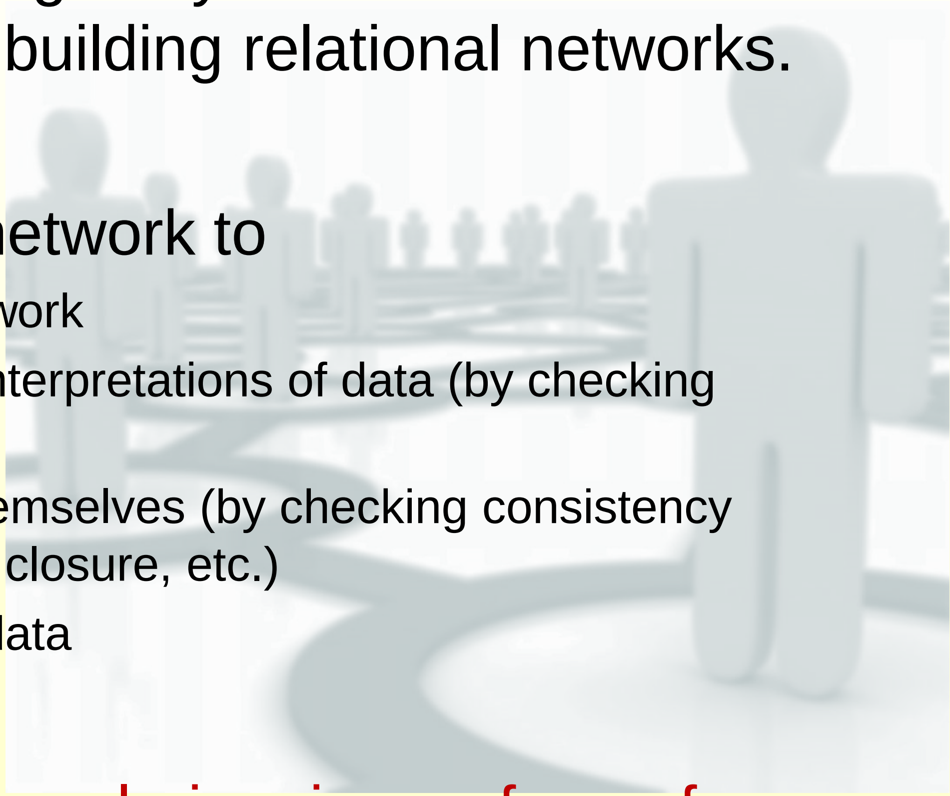
Networks of Shapes and Images





Societies, or
Social Networks of Data Sets

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

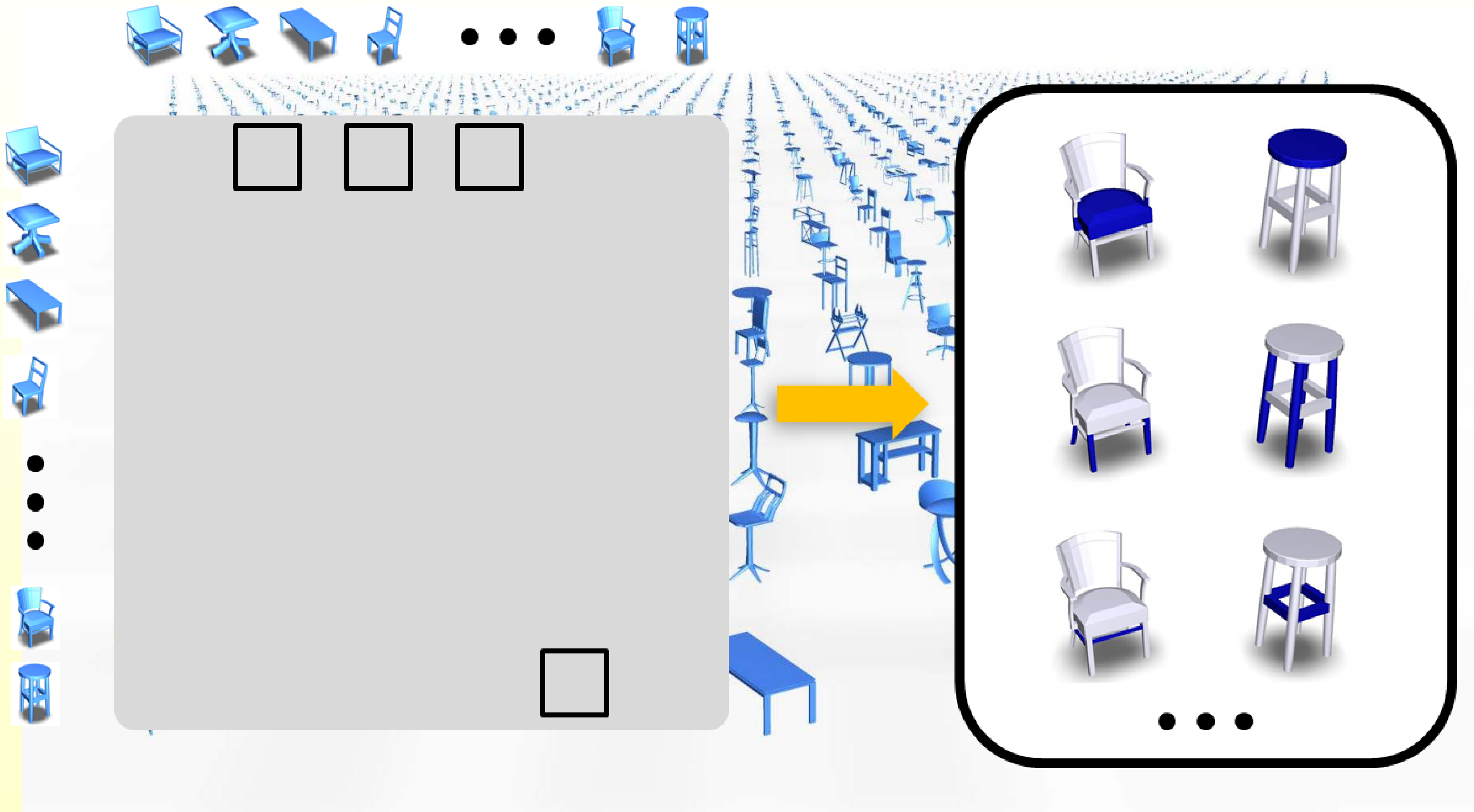


We can exploit the relational network to

- 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 a regularizer in any form of joint data analysis.

Semantic Structure Emerges from the Network



[Q. Huang, F. Wang, L. Guibas, '14]

The Operator View

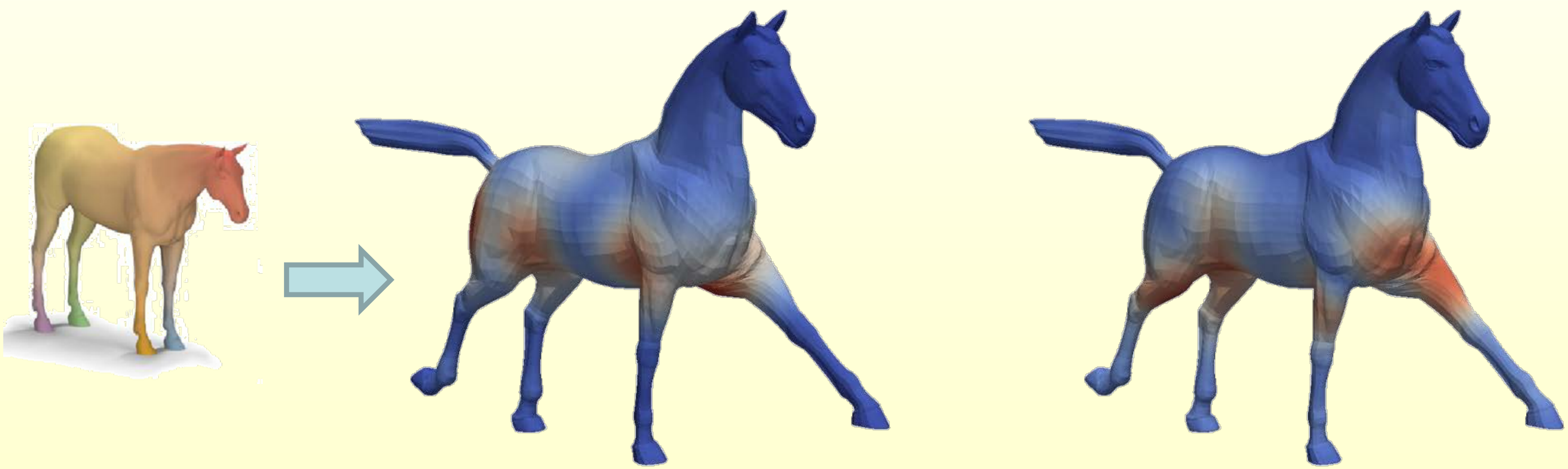
Shape Differences

[R. Rustamov, M. Ovsjanikov, O. Azercot, M. Ben-Chen, F. Chazal, L.G. Siggraph ’13]

vs.

Understanding Intrinsic Distortions

- Where and how are shapes different, locally and globally, irrespective of their embedding



Area distortion

Conformal distortion

A Functional View of Distortions

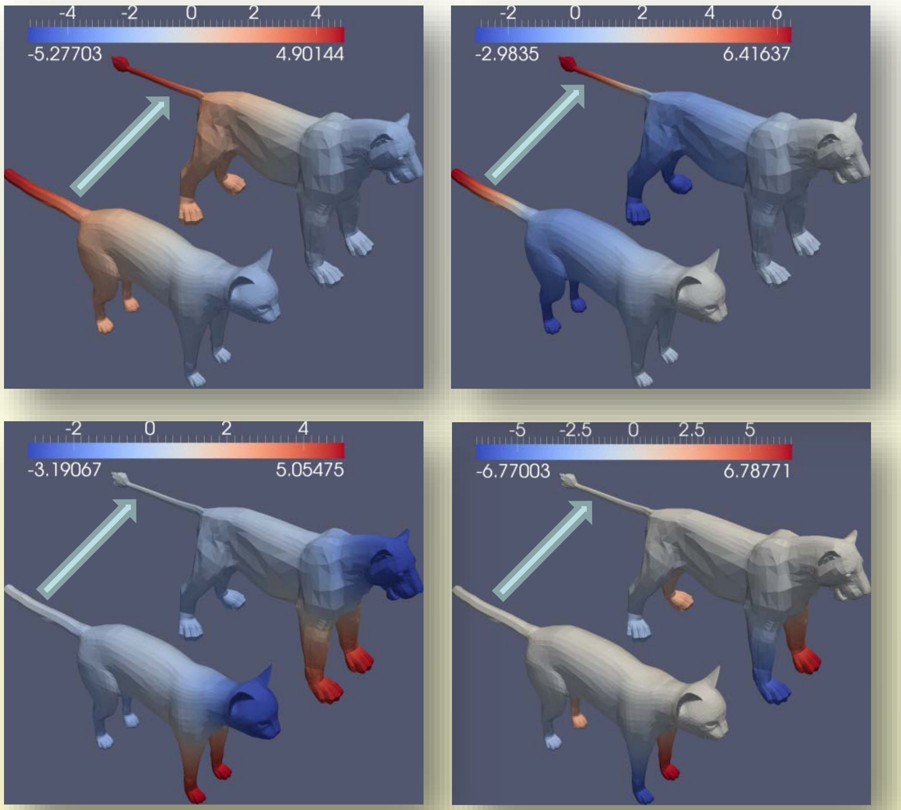
To measure distortions induced by a map, track how inner products of vectors change after transporting.

To measure distortions induced by a map, track how inner products of functions change after transporting.

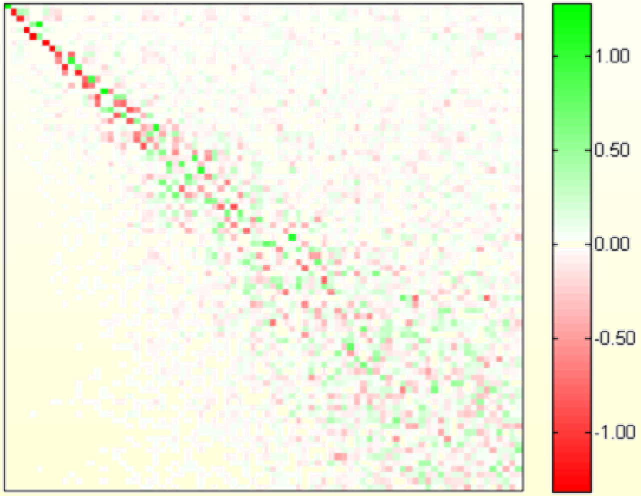
Riemann

Input: Functional Map F

from cat to lion



Functions on cat are transferred to lion using F

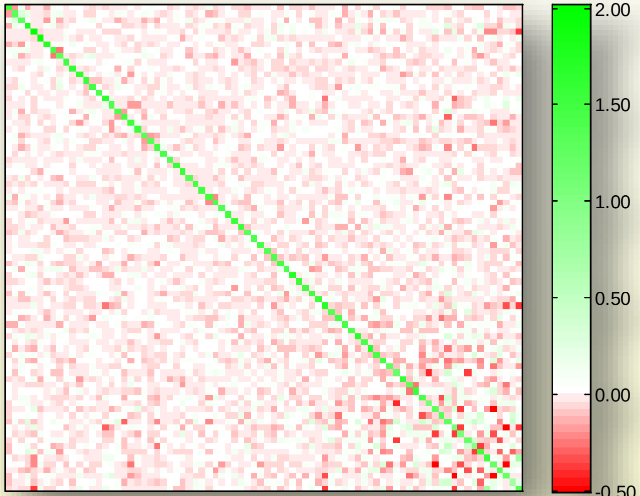


F is a linear operator (matrix)

$$F : L^2(cat) \rightarrow L^2(lion)$$

Output: A Shape Difference

V – area-based shape difference

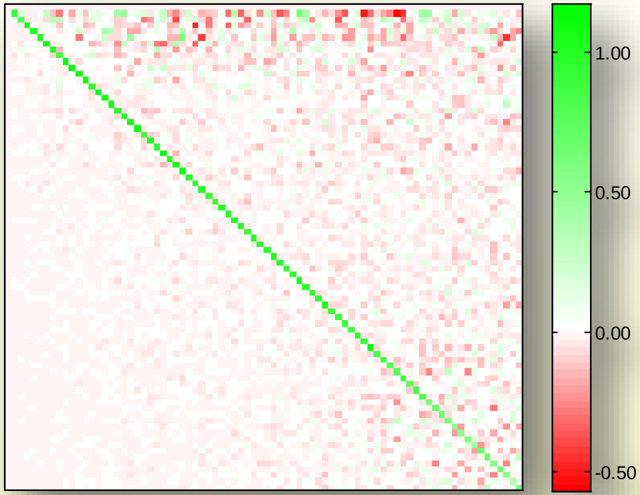


linear operator (matrix)

$$V : L^2(cat) \rightarrow L^2(cat)$$

$$\int_N F(f)F(g) = \int_M fV(g)$$

R – conformal shape difference



linear operator (matrix)

$$R : L^2(cat) \rightarrow L^2(cat)$$

$$\int_N \nabla F(f) \nabla F(g) = \int_M \nabla f \nabla R(g)$$

Also operators

The Art of Measurement

- A metric is defined by a functional inner product

$$h^M(f,g) = \int_M f(x)g(x)d\mu(x)$$

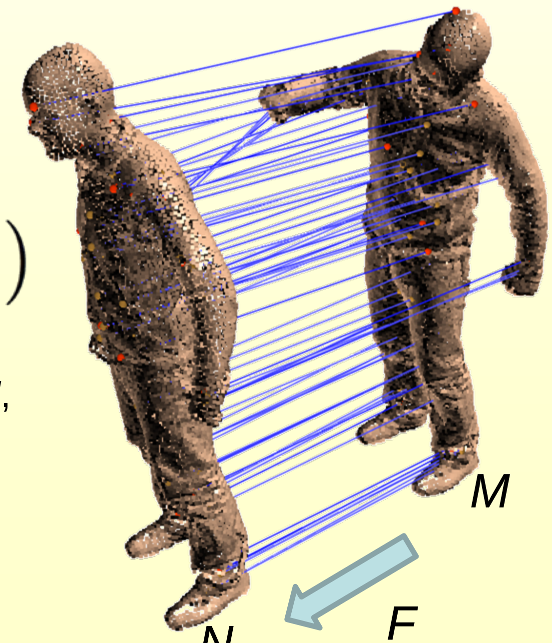
Riemann

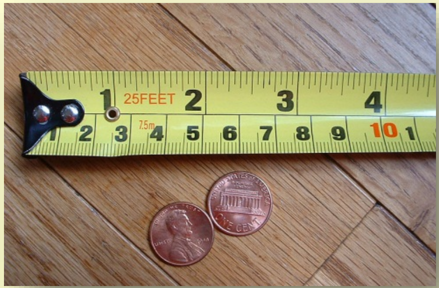
- So we can compare M and N by comparing

$$h^N(F(f),F(g))$$

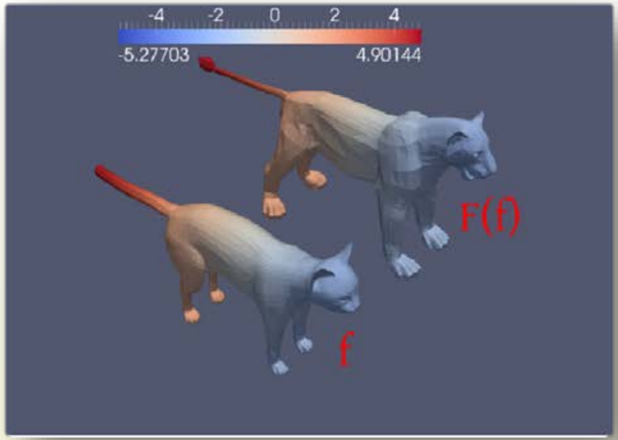
$$h^M(f,g)$$

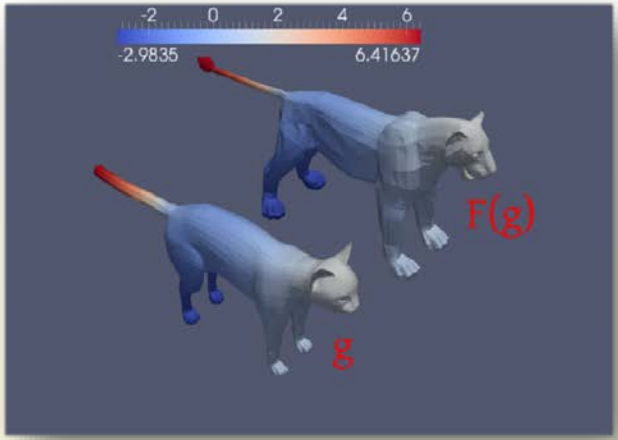
The functional map F transports these functions to N, where we repeat this measurement with the inner product $h^N(F(f),F(g))$





Measurement Discrepancies



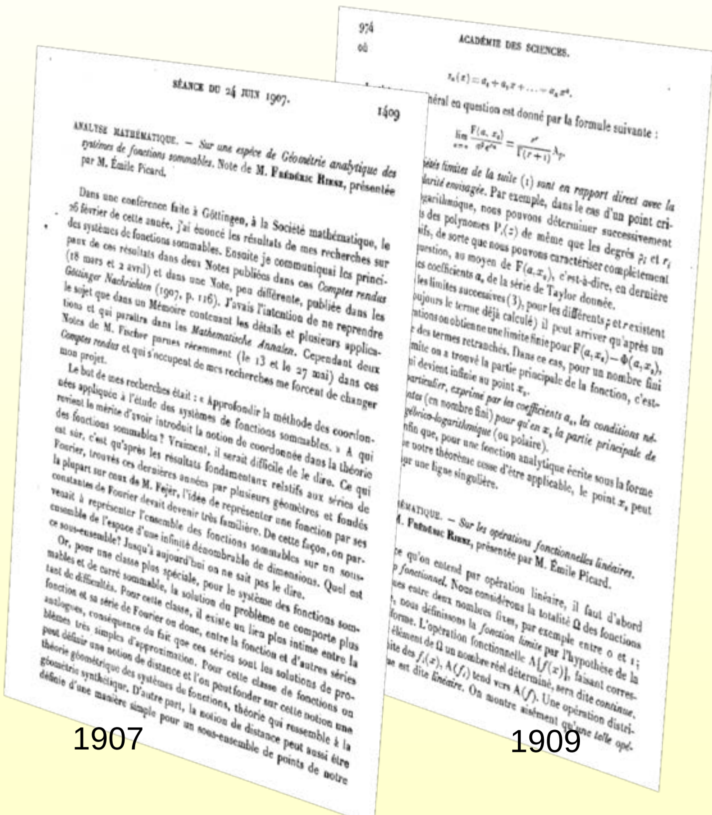


$$\int_{lion} \underbrace{F(f)F(g)\, d\mu_l}_{\text{after}} \neq \int_{cat} \underbrace{fg\, d\mu_c}_{\text{before}}$$

Both can be considered as inner products on the cat

The Universal Compensator

Comptes Rendus Hebdomadaires des
Séances de l'Académie des Sciences de Paris



1907

1909

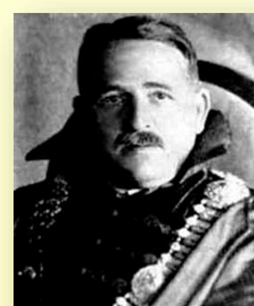
Riesz Representation Theorem

There exists a linear operator

$$V : L^2(cat) \to L^2(cat)$$

such that

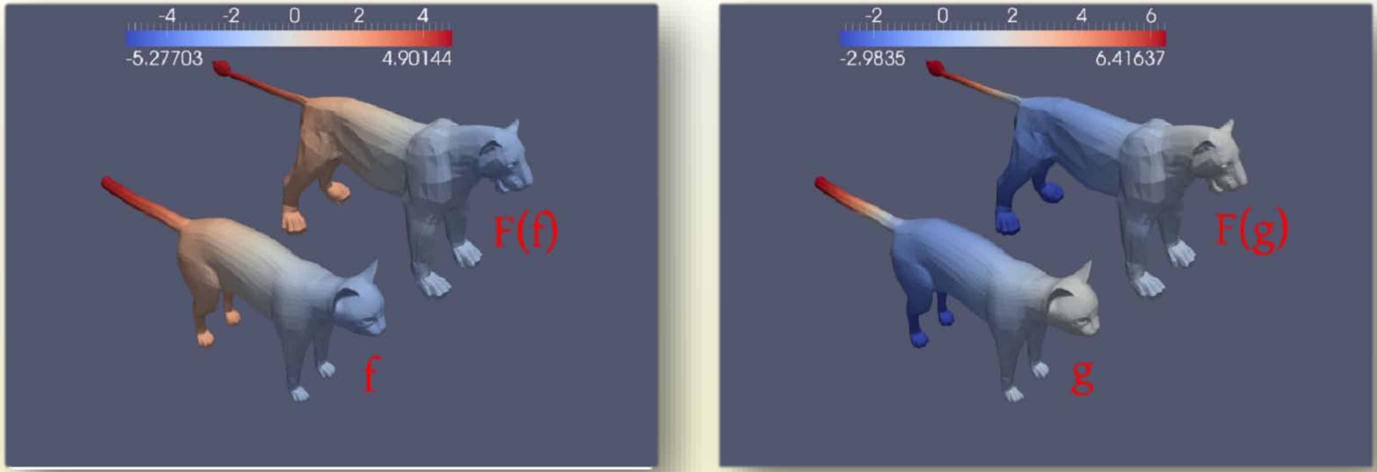
$$\langle f, g \rangle_{\text{after}} = \langle f, V(g) \rangle_{\text{before}}$$



Frigyes Riesz

Area-Based Shape Difference:

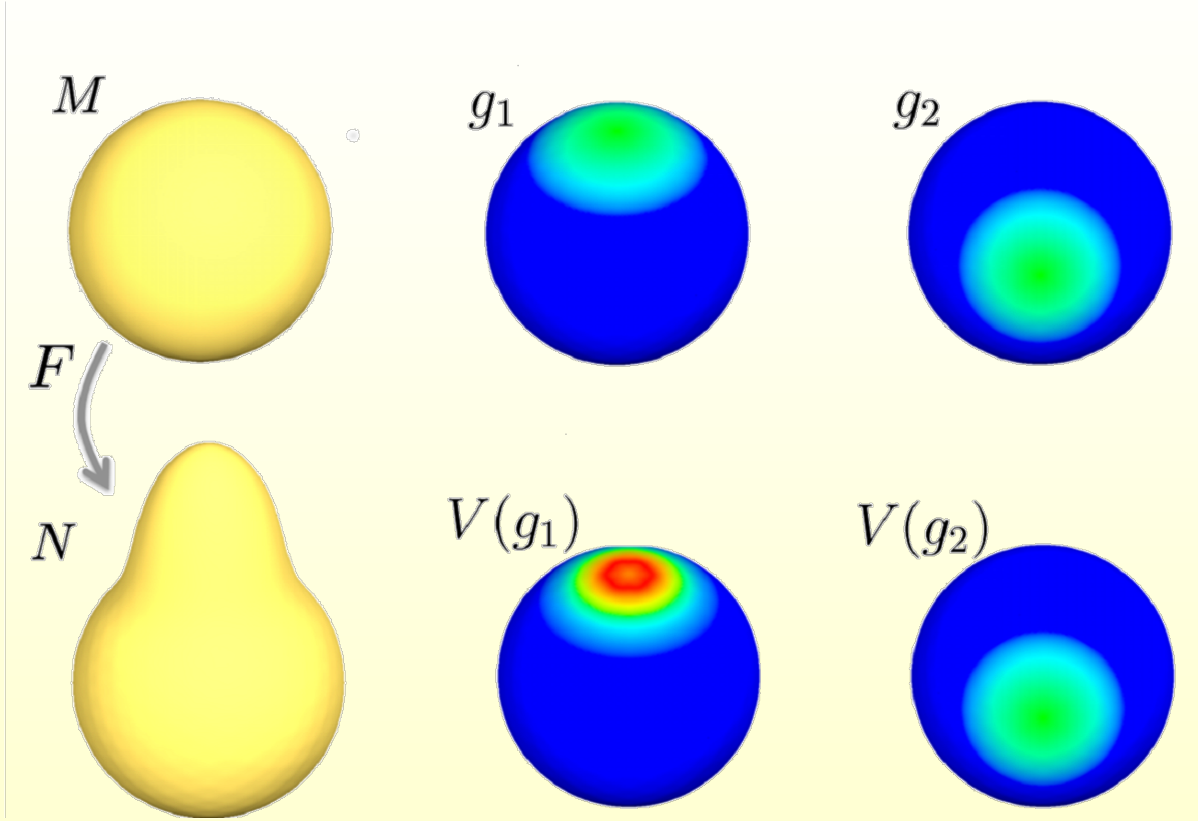
$$V \approx F^T F$$



$$\int_{lion} F(f)F(g) \neq \int_{cat} fg$$

$$\int_{lion} F(f)F(g) = \int_{cat} f\,V(g)$$

A Small Example of V



Note that *V* maps functions on *M* to functions on *M*

$$\int_N F(f)F(g) = \int_M fV(g)$$

Conformal Shape Difference: *R*

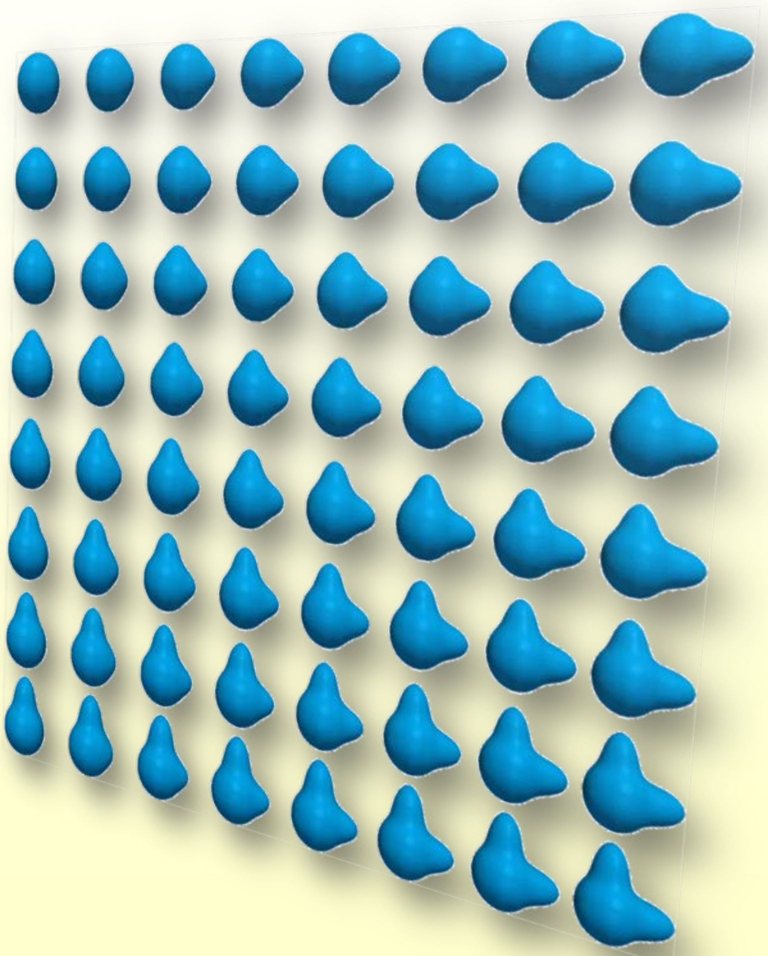
Consider a different inner-product of functions ...

get information about conformal distortion

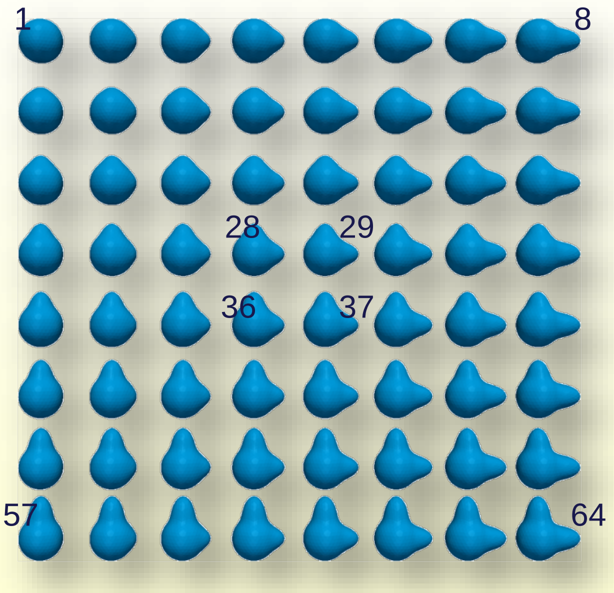
$$\int_N \nabla F(f) \nabla F(g) = \int_M \nabla f \nabla R(g)$$

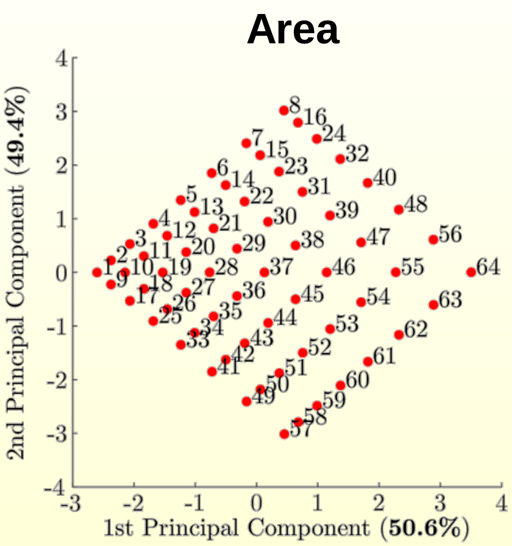
area preservation + angle preservation = isometry

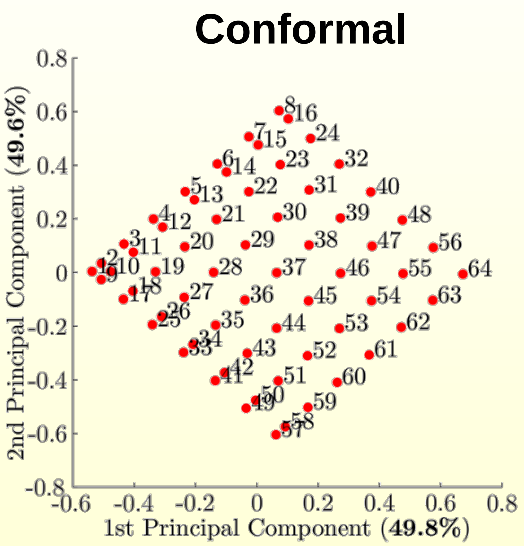
Shape Differences in Collections

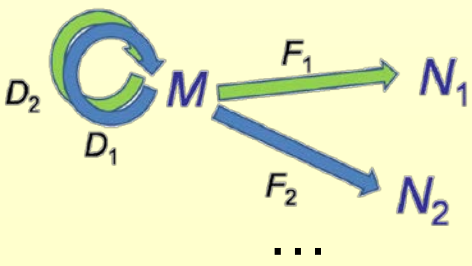


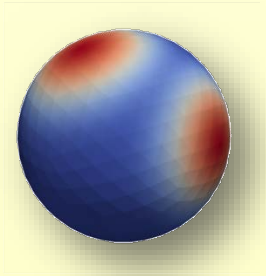
Intrinsic Shape Space

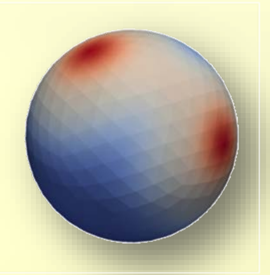




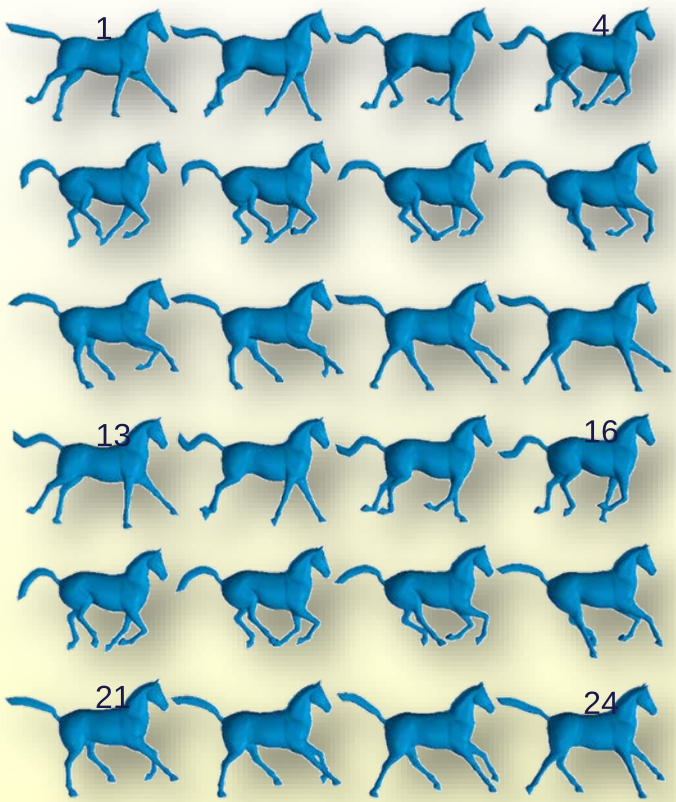


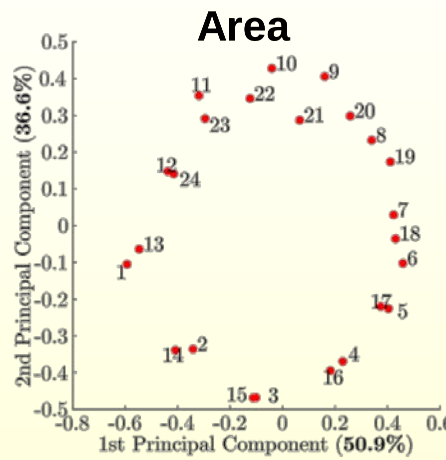


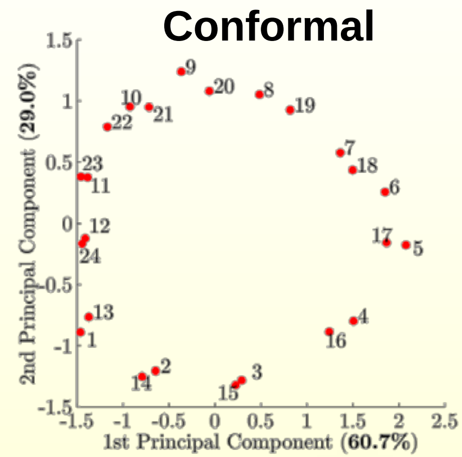


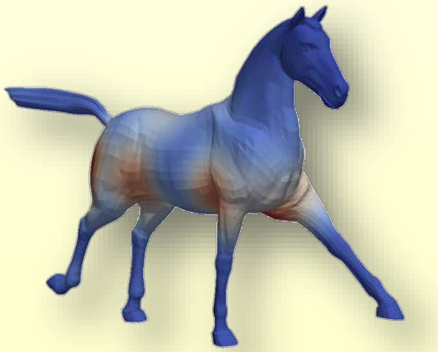


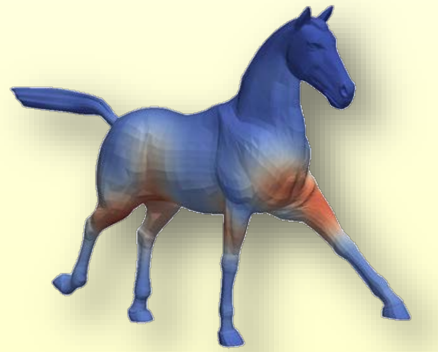
Intrinsic Shape Space



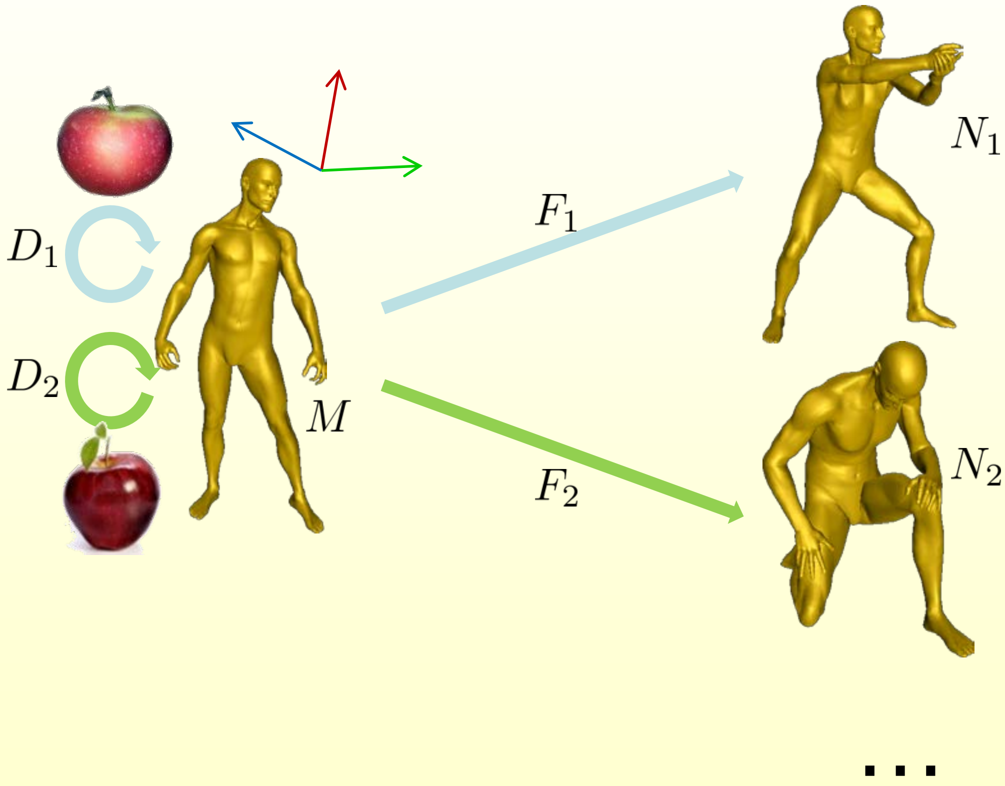




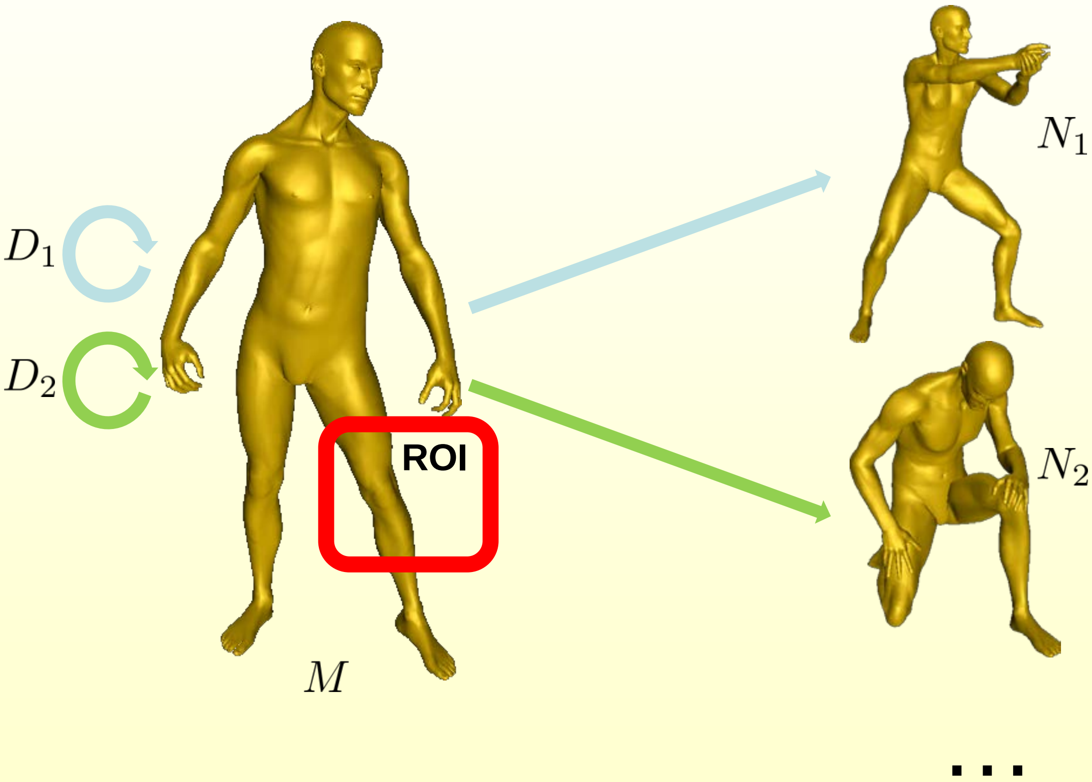




Comparing Differences I



Localized Comparisons

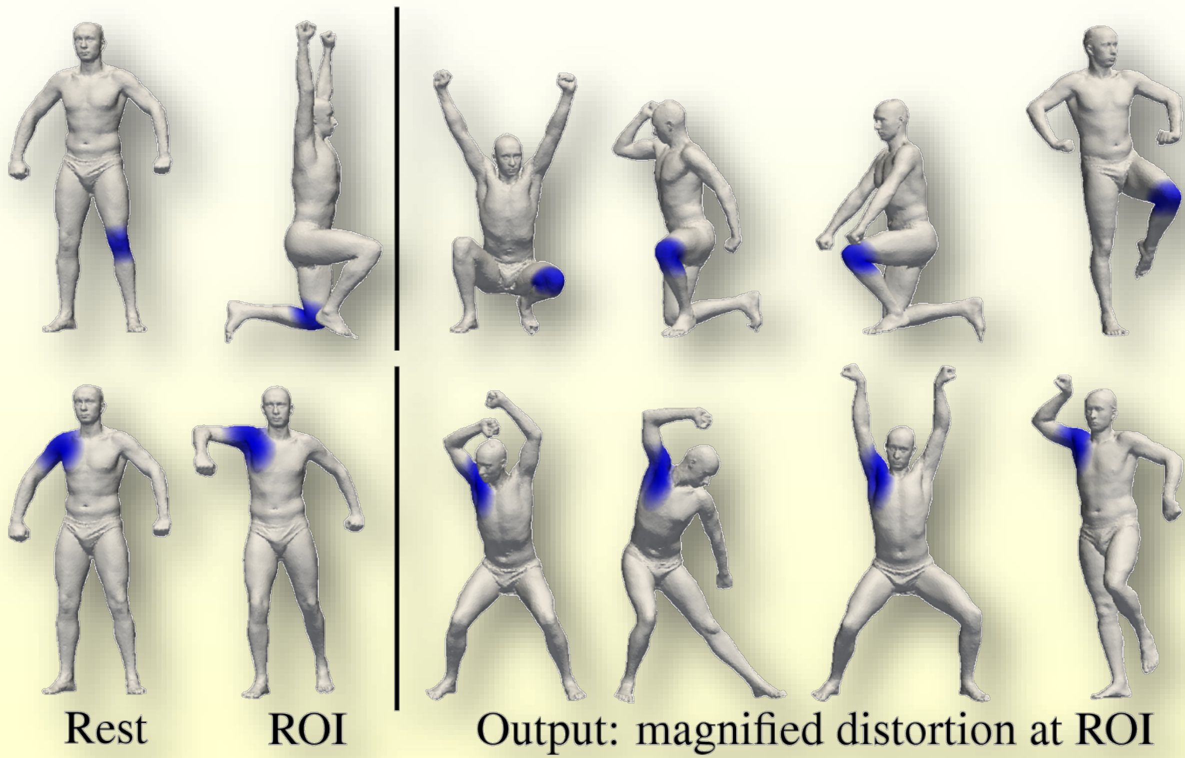


$$\rho : M \to \mathbb{R}$$

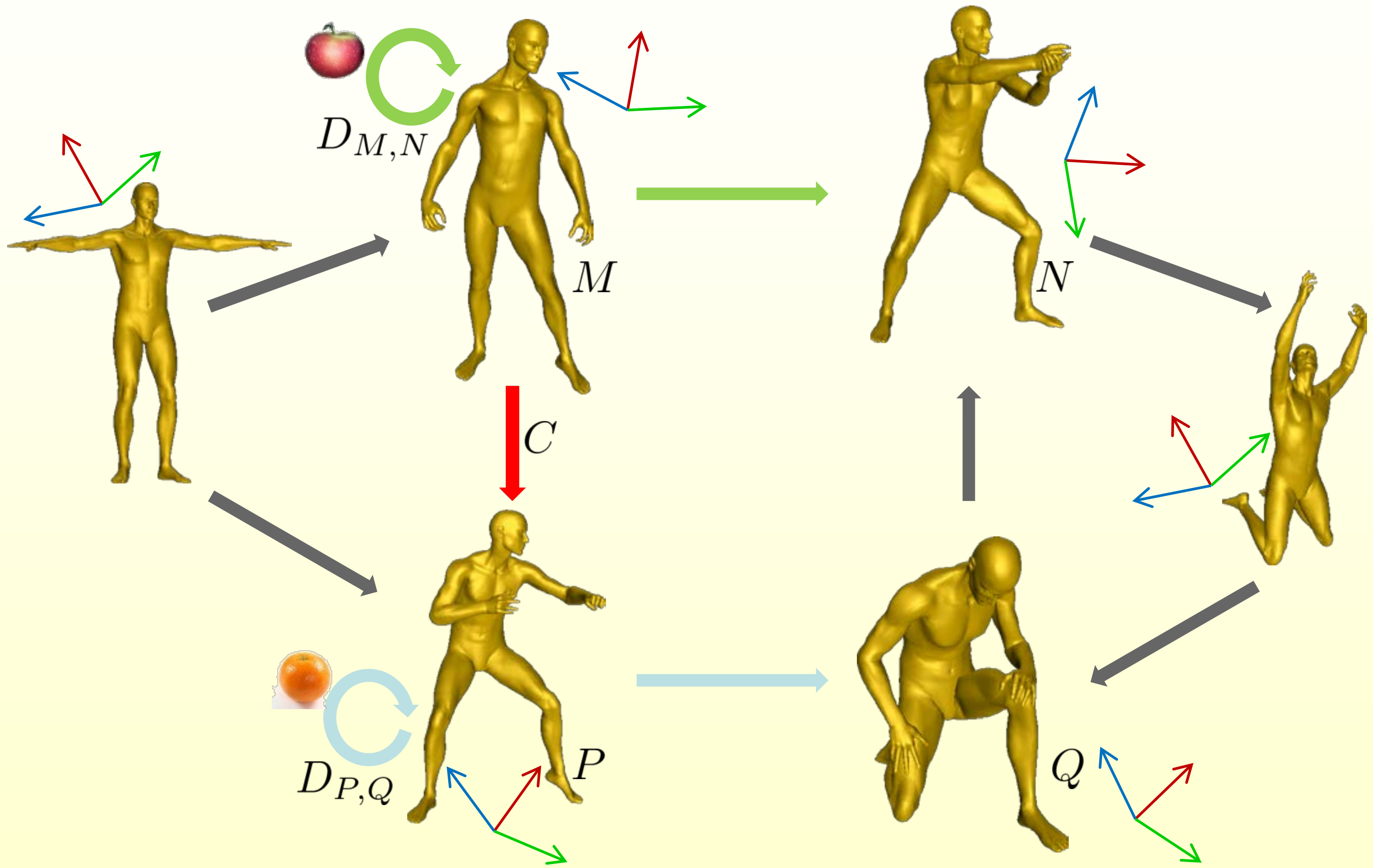
supported in RoI

$$D_1\rho \text{ to } D_2\rho$$

Exaggeration of Difference in RoI

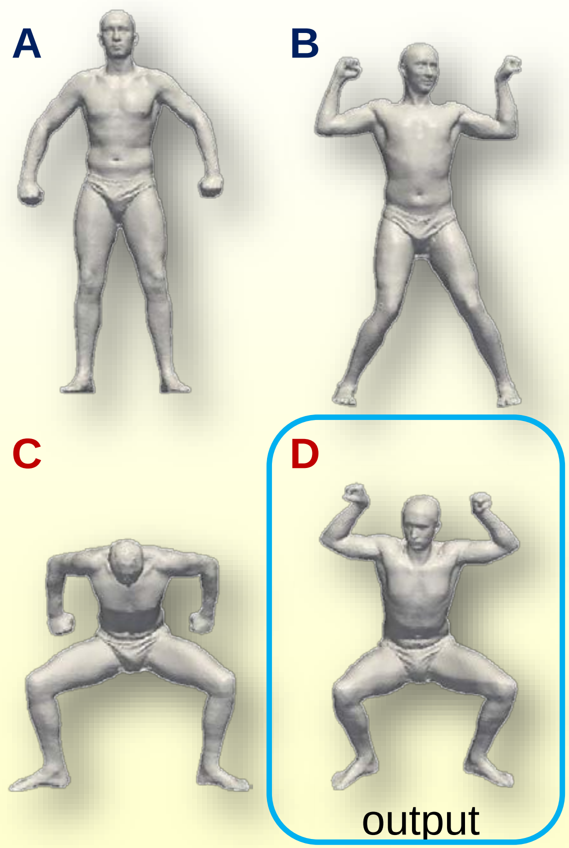


Comparing Differences II



$$D_{M,N} \sim C^{-1} D_{P,Q} C$$

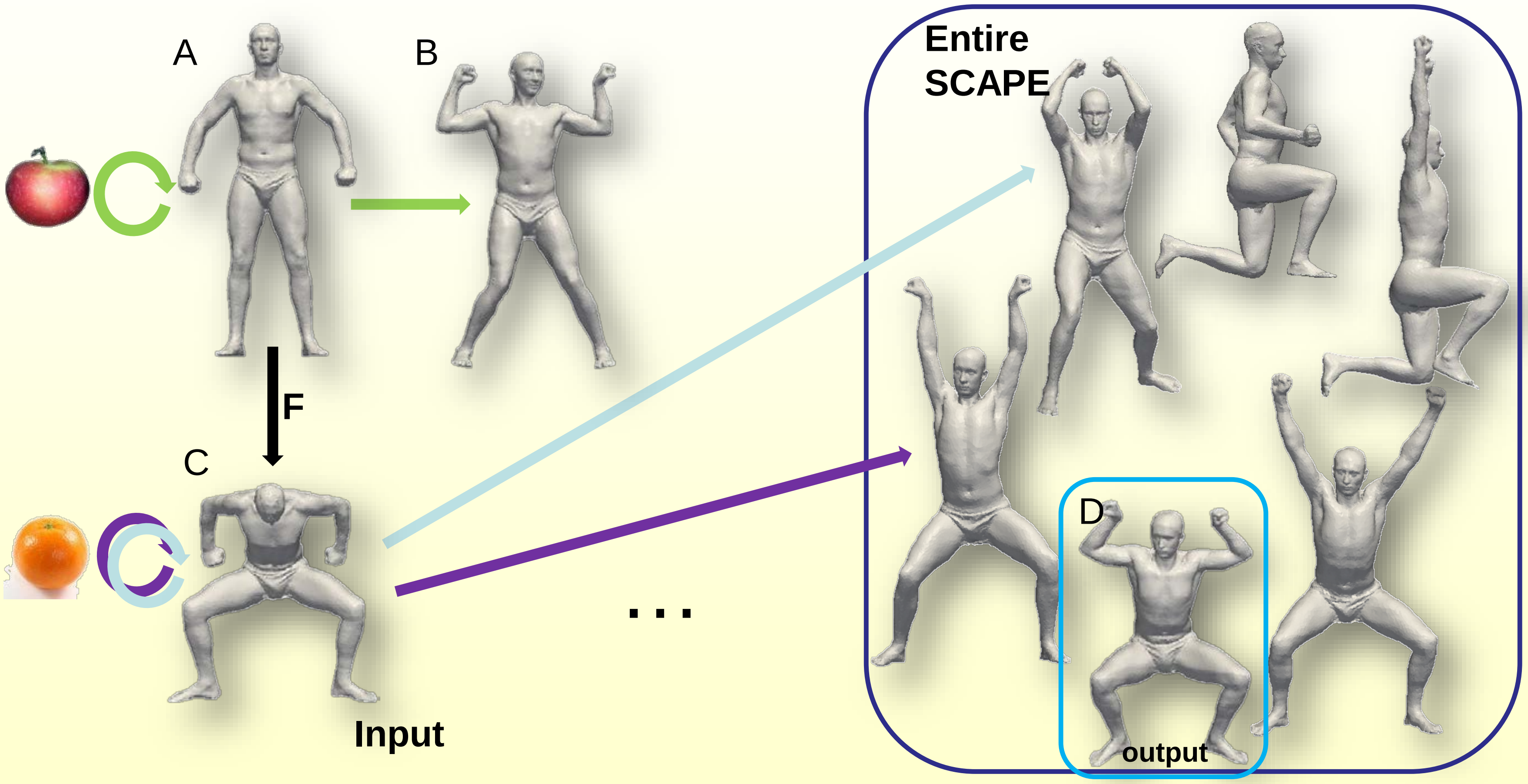
Analogies: D relates to C as B relates to A



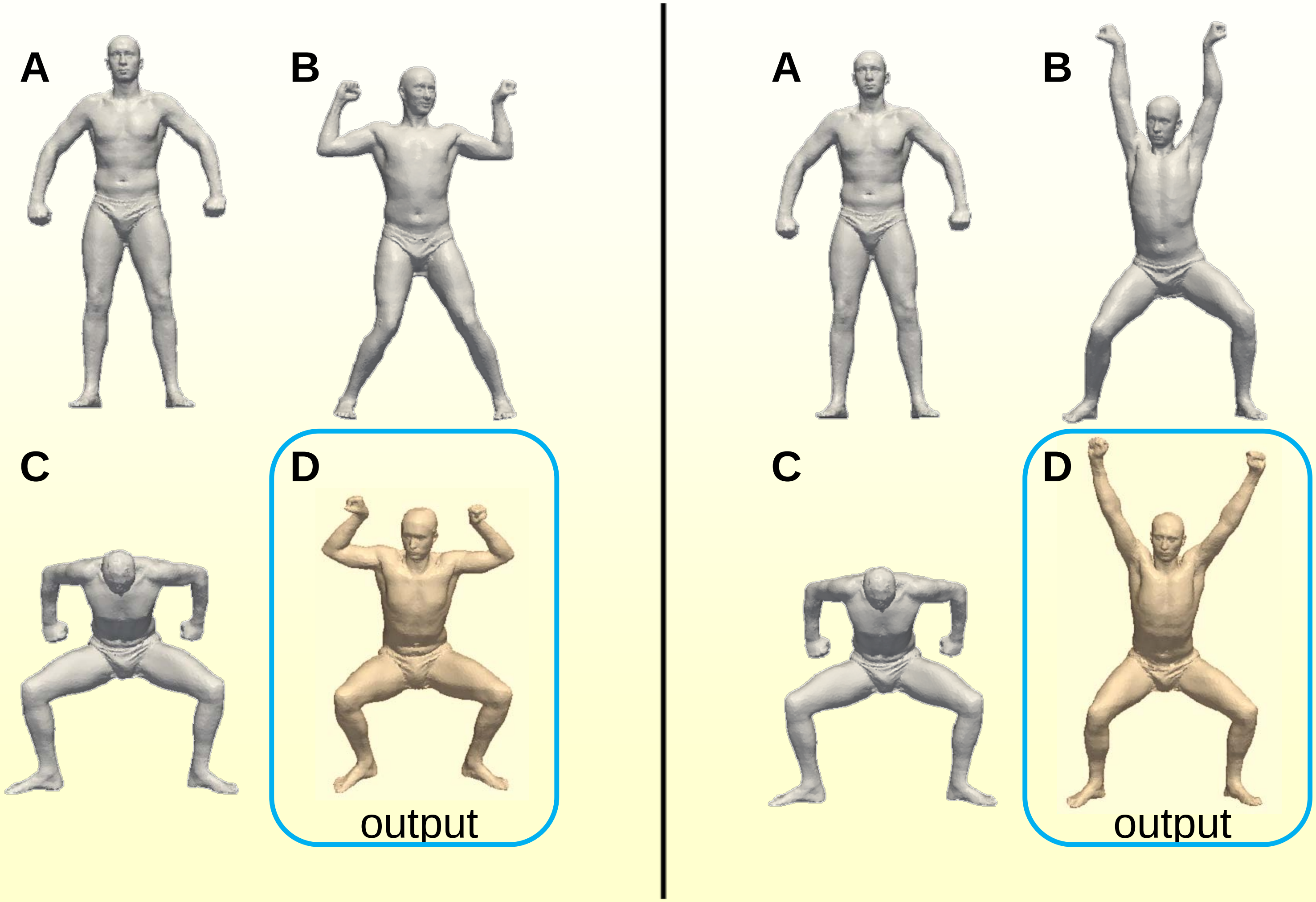
D = C + (B – A)

hands raised up

Analogies: D relates to C as B relates to A



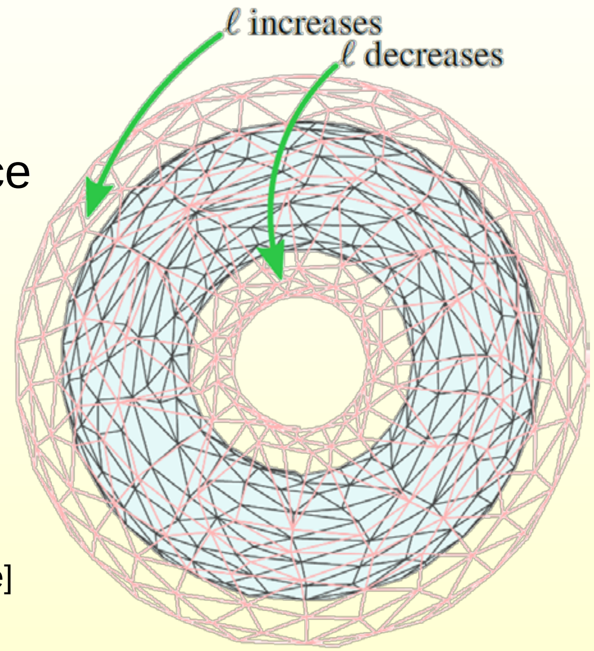
Shape Analogies

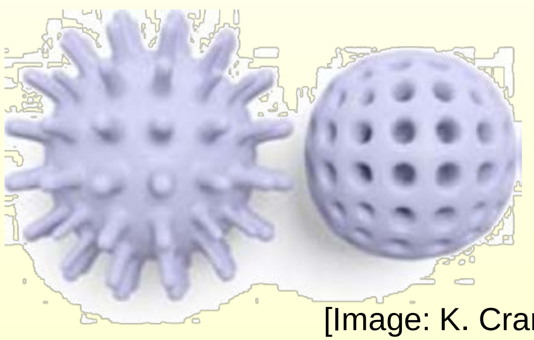


Extrinsic Shape Differences
Shape Synthesis

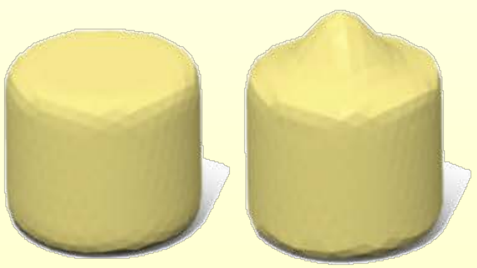
[E. Corman, J. Solomon, M. Ben-Chen, L. J. Guibas, and M. Ovsjanikov, 2017]

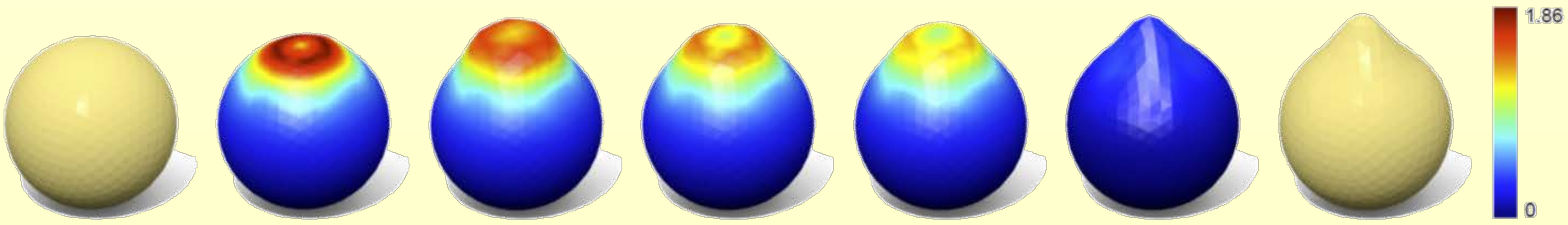
Intrinsic differences of an offset surface
capture extrinsic distortions of the original surface



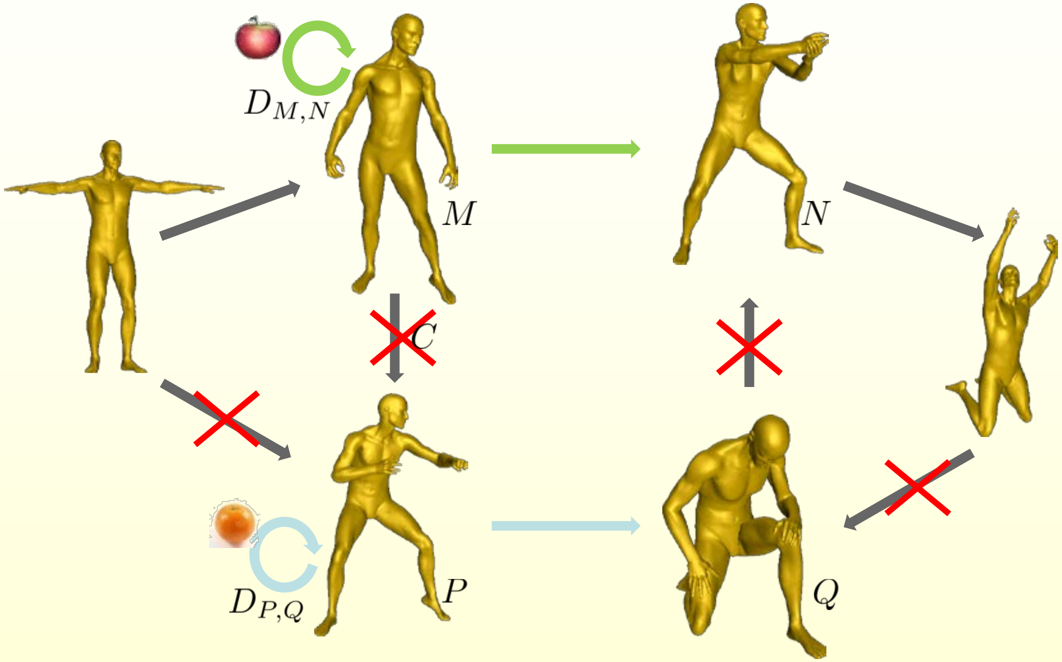


[Image: K. Crane]



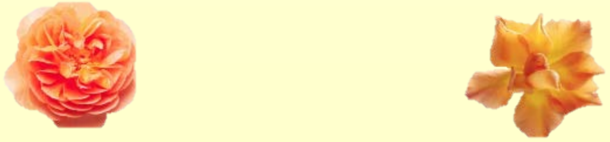


Comparing Differences III

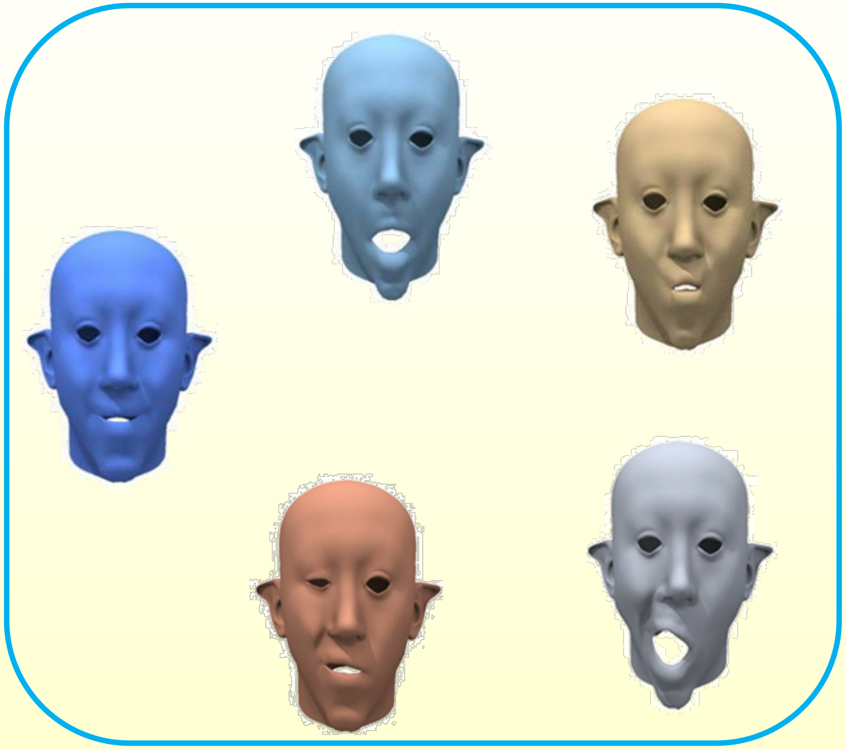


$$D_{M,N} \sim C^{-1} D_{P,Q} C$$

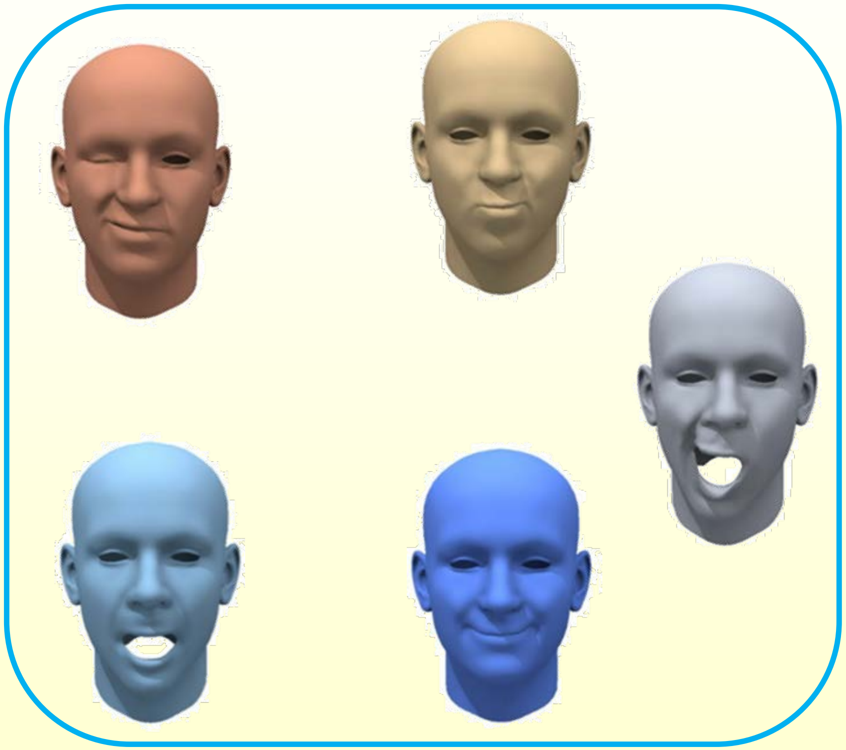
$$Spec(D_{M,N}) \sim Spec(D_{P,Q})$$



Aligning Disconnected Collections

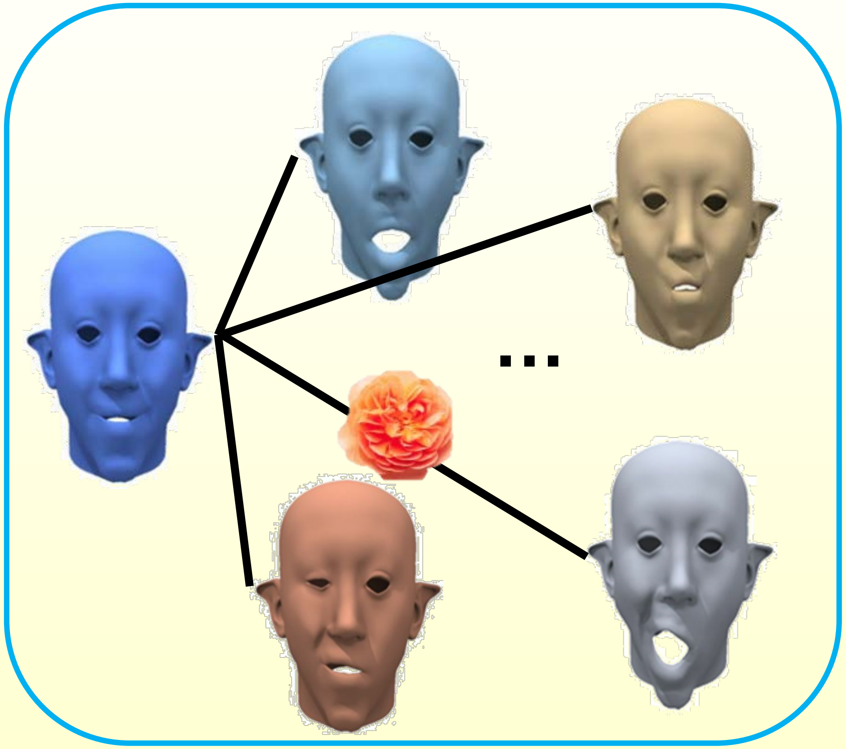


First Collection

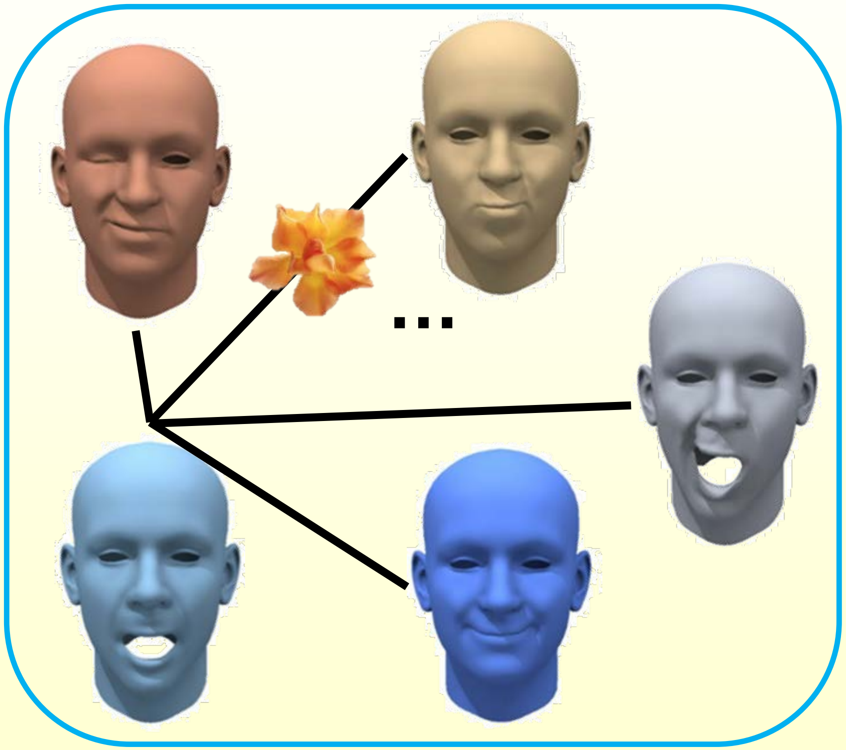


Second Collection

Aligning Disconnected Collections

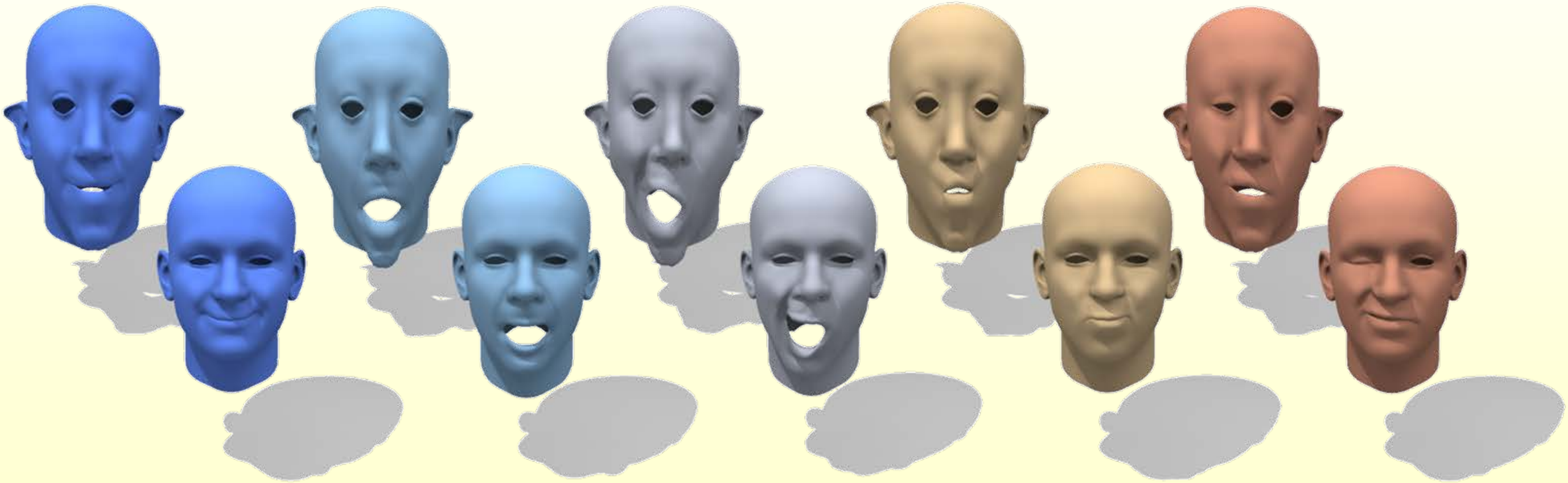


Complete graph



Complete graph

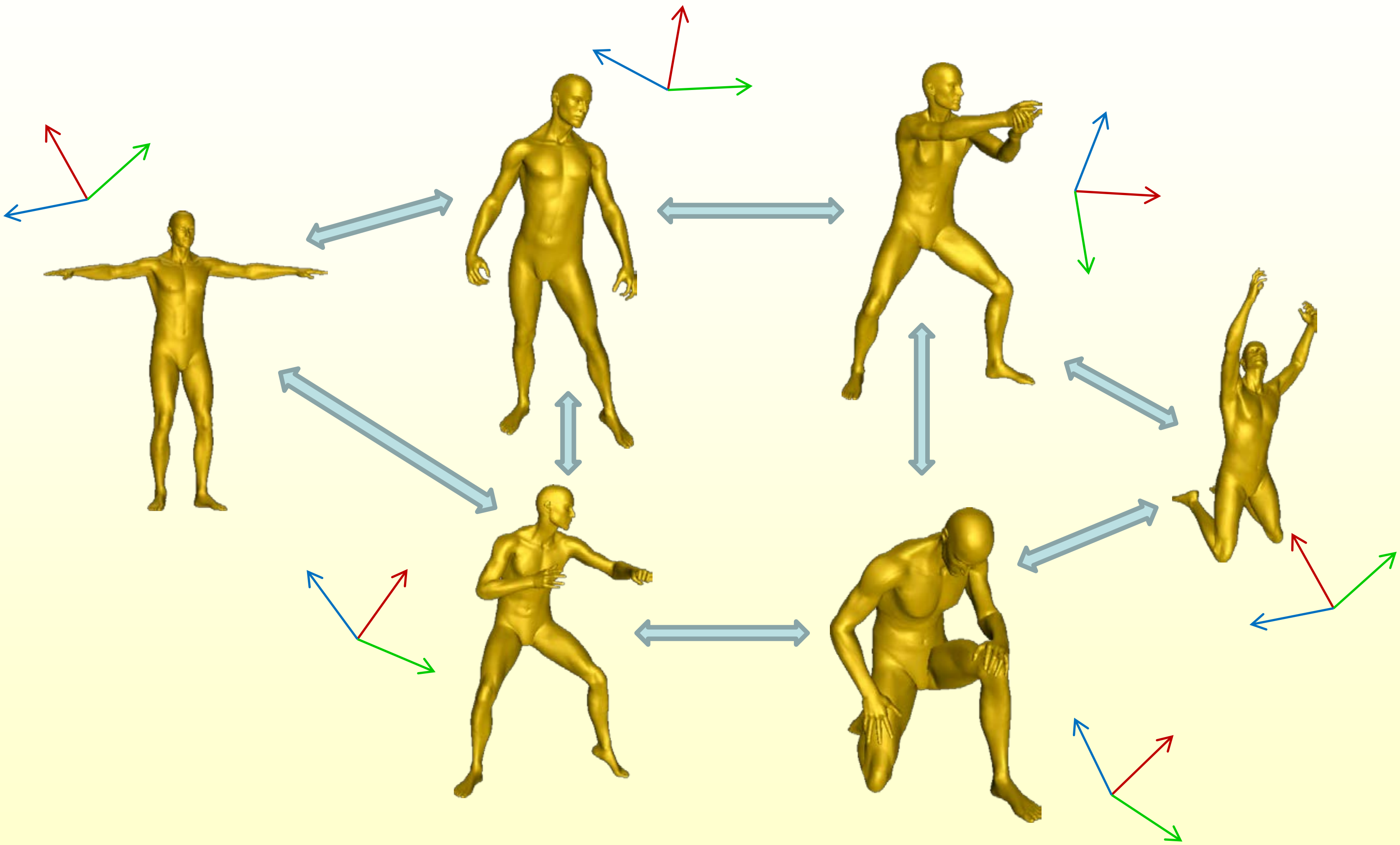
Aligning, Without "Crossing the River"



Comparing the differences is sometimes easier than comparing the originals

Large Networks: Consistency of Network Transport

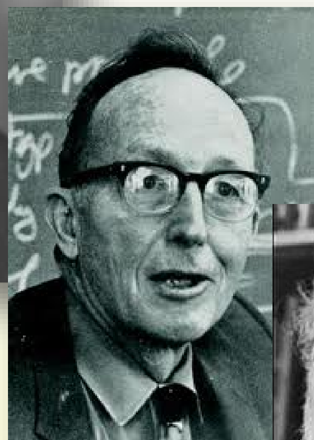
Map Networks for Related Data



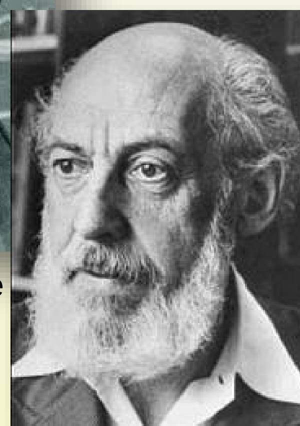
Networks of “samenesses”

A Functorial View of Data

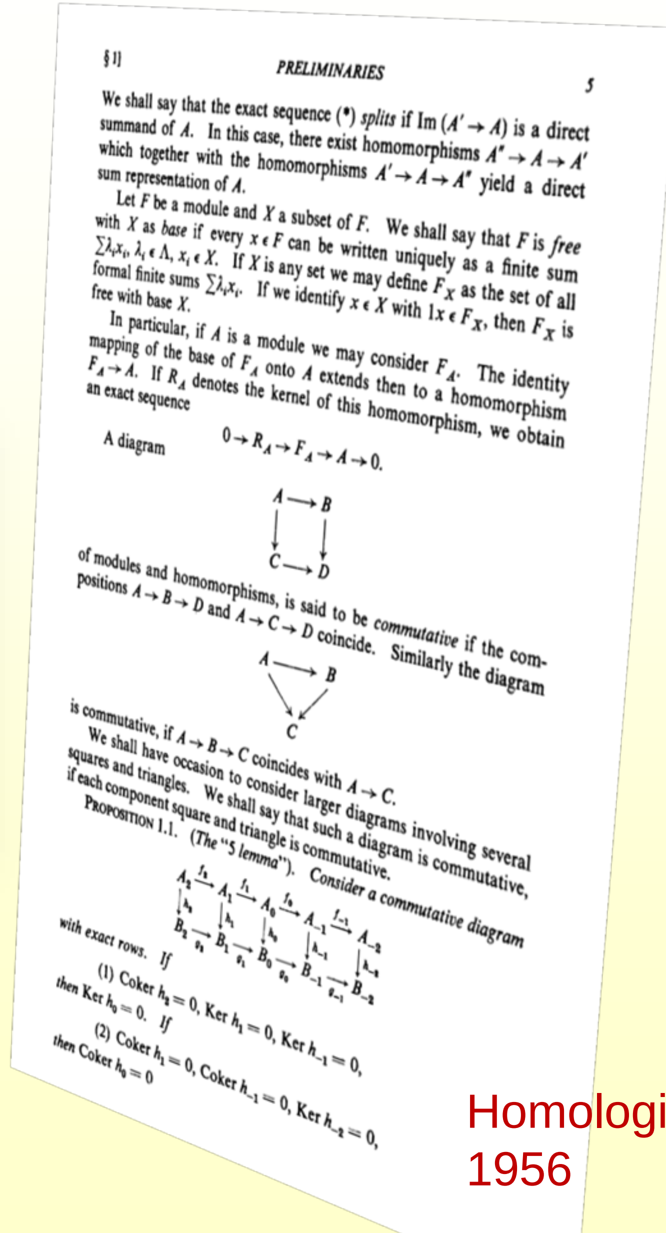
Herni Cartan



Saunders MacLane



Samuel Eilenberg



§ 1] PRELIMINARIES 5

We shall say that the exact sequence (*) *splits* if Im $(A' \to A)$ is a direct summand of A. In this case, there exist homomorphisms $A'' \to A \to A'$ which together with the homomorphisms $A' \to A \to A''$ yield a direct sum representation of A.

Let F be a module and X a subset of F. We shall say that F is *free* with X as *base* if every $x \in F$ can be written uniquely as a finite sum $\sum \lambda_i x_i$, $\lambda_i \in \Lambda$, $x_i \in X$. If X is any set we may define F_X as the set of all formal finite sums $\sum \lambda_i x_i$. If we identify $x \in X$ with $1x \in F_X$, then F_X is free with base X.

In particular, if A is a module we may consider F_A. The identity mapping of the base of F_A onto A extends then to a homomorphism $F_A \to A$. If R_A denotes the kernel of this homomorphism, we obtain an exact sequence

$$0 \to R_A \to F_A \to A \to 0.$$

A diagram

$$\begin{array}{ccc} A & \longrightarrow & B \\ \downarrow & & \downarrow \\ C & \longrightarrow & D \end{array}$$

of modules and homomorphisms, is said to be *commutative* if the compositions $A \to B \to D$ and $A \to C \to D$ coincide. Similarly the diagram

$$\begin{array}{ccc} A & \longrightarrow & B \\ & \searrow \; \swarrow & \\ & C & \end{array}$$

is commutative, if $A \to B \to C$ coincides with $A \to C$.

We shall have occasion to consider larger diagrams involving several squares and triangles. We shall say that such a diagram is commutative, if each component square and triangle is commutative.

PROPOSITION 1.1. (*The "5 lemma"*). *Consider a commutative diagram*

$$\begin{array}{ccccccccc} A_2 & \xrightarrow{f_2} & A_1 & \xrightarrow{f_1} & A_0 & \xrightarrow{f_0} & A_{-1} & \xrightarrow{f_{-1}} & A_{-2} \\ \downarrow h_2 & & \downarrow h_1 & & \downarrow h_0 & & \downarrow h_{-1} & & \downarrow h_{-2} \\ B_2 & \xrightarrow{g_2} & B_1 & \xrightarrow{g_1} & B_0 & \xrightarrow{g_0} & B_{-1} & \xrightarrow{g_{-1}} & B_{-2} \end{array}$$

with exact rows. If

(1) Coker $h_2 = 0$, Ker $h_1 = 0$, Ker $h_{-1} = 0$,

then Ker $h_0 = 0$. *If*

(2) Coker $h_1 = 0$, Coker $h_{-1} = 0$, Ker $h_{-2} = 0$,

then Coker $h_0 = 0$

The Information is in the Maps

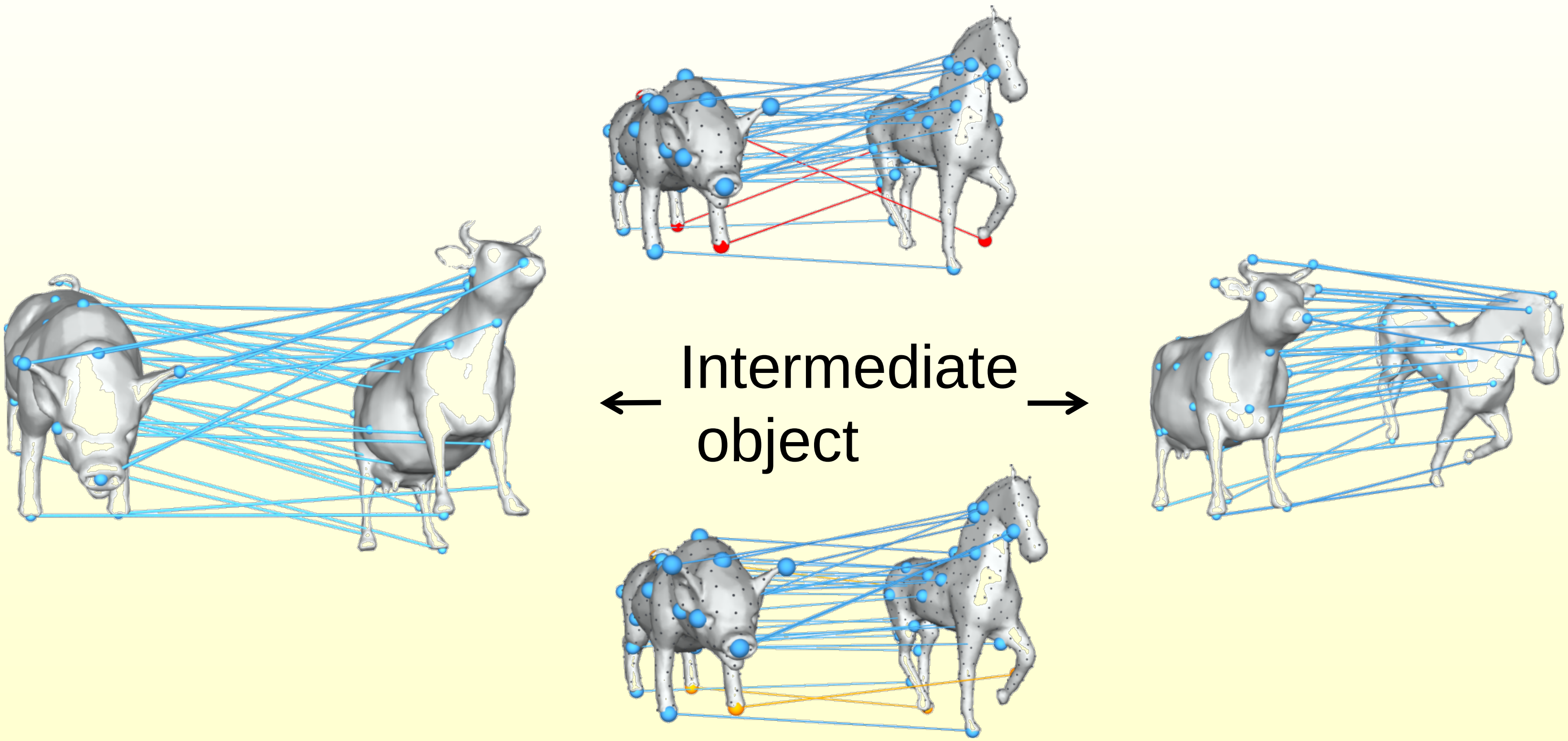
Homological Algebra 1956

Yes, But With a Statistical Flavor

- Yes, straight out of the playbook of homological algebra / algebraic topology
- But, the maps
  - are not given by canonical constructions
  - they have to be estimated and can be noisy
  - the network acts as a regularizer …
  - commutativity still very important
  - imperfections of commutativity in function transport convey valuable information: consistency vs. variability – “curvature” in shape space

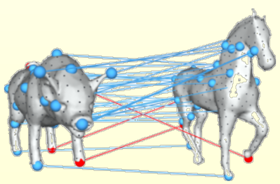
Fixing Maps

[Q. Huang, G. Zhang, L. Gao, S. Hu, A. Bustcher, and L. Guibas, 2012]

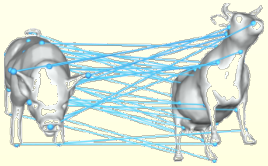


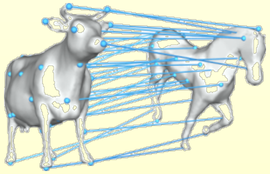
Cycle-Consistency ≡ Low-Rank

- In a map network, commutativity, path-invariance, or cycle-consistency are equivalent to a low rank or semidefiniteness condition on a big mapping matrix



$$X = \begin{pmatrix} I_m & X_{1,2} & \cdots & X_{1,n} \\ X_{1,2} & I_m & \cdots & \cdots \\ \vdots & \vdots & I_m & X_{(n-1),n} \\ X_{n,1} & \vdots & X_{n,(n-1)} & I_m \end{pmatrix}.$$





- Conversely, such a low-rank condition can be used to
  - regularize and clean up functional maps
  - extract shared structure

Map processing!

Map Synchronization by Factorization

$$X = \begin{bmatrix} I_m & X_{12} & \cdots & X_{1n} \\ X_{21} & I_m & \cdots & \vdots \\ \vdots & \vdots & \ddots & X_{n-1,n} \\ X_{n1} & \cdots & X_{n,n-1} & I_m \end{bmatrix}$$

$$X_{ij} = X_{j1} X_{i1}^T$$

$$= \begin{bmatrix} I_m \\ \vdots \\ X_{n1} \end{bmatrix} \begin{bmatrix} I_m & \cdots & X_{n1}^T \end{bmatrix}$$

Map Synchronization

- SDP formulation

  [Y, Chen, L. Guibas, Q. Huang, 2014]

- Recovery guarantees

  [Q. Huang, L. Guibas, 2013]

Shared Structure Discovery

Entity Extraction in Images

[F. Wang, Q. Huang, L. G., ICCV ’13]

- Task: jointly segment a set of related images
  - same object, different viewpoints/scales:

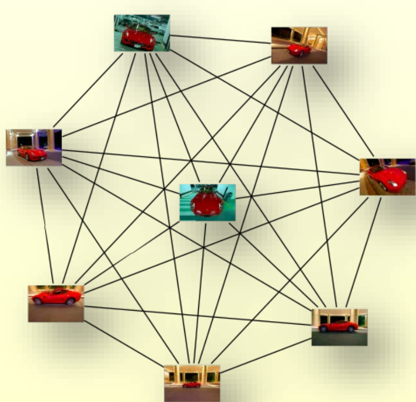
  - similar objects of the same class:

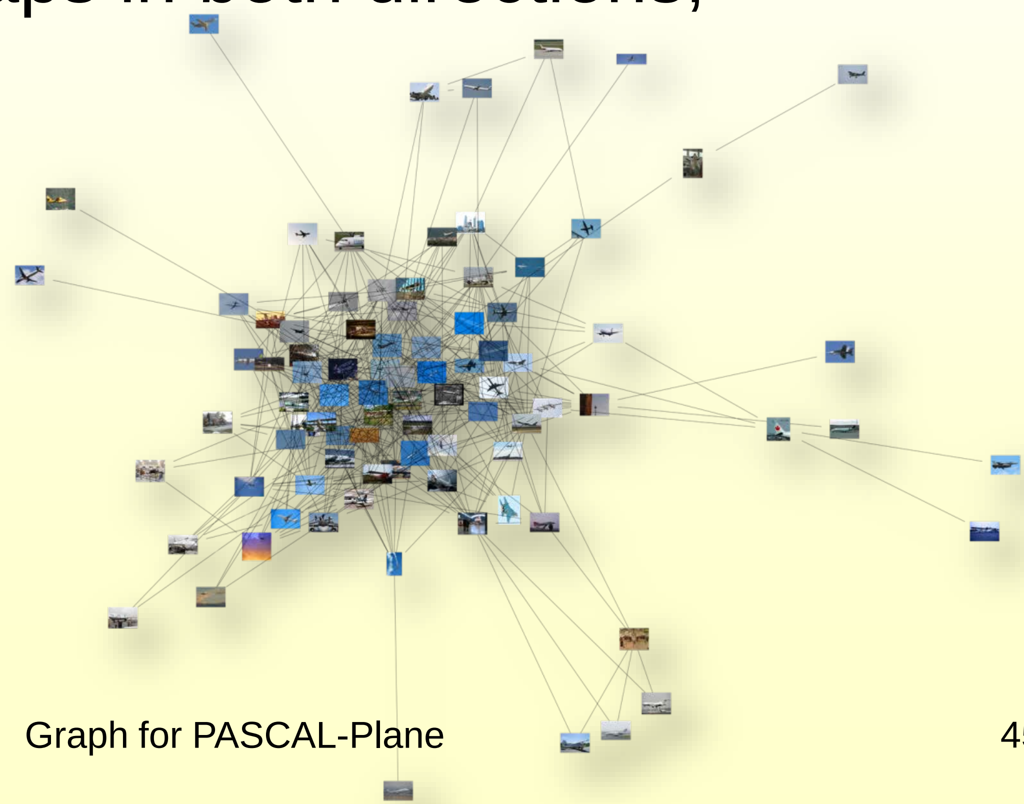
- Benefits and challenges:
  - Images can provide weak supervision for each other
  - But exactly how should they help each other? How to deal with clutter and irrelevant content?

Co-Segmentation via an Image Network

- Image similarity graph based on GIST
  - Each edge has global image similarity w_{ij} and functional maps in both directions;
  - Sparse if large.



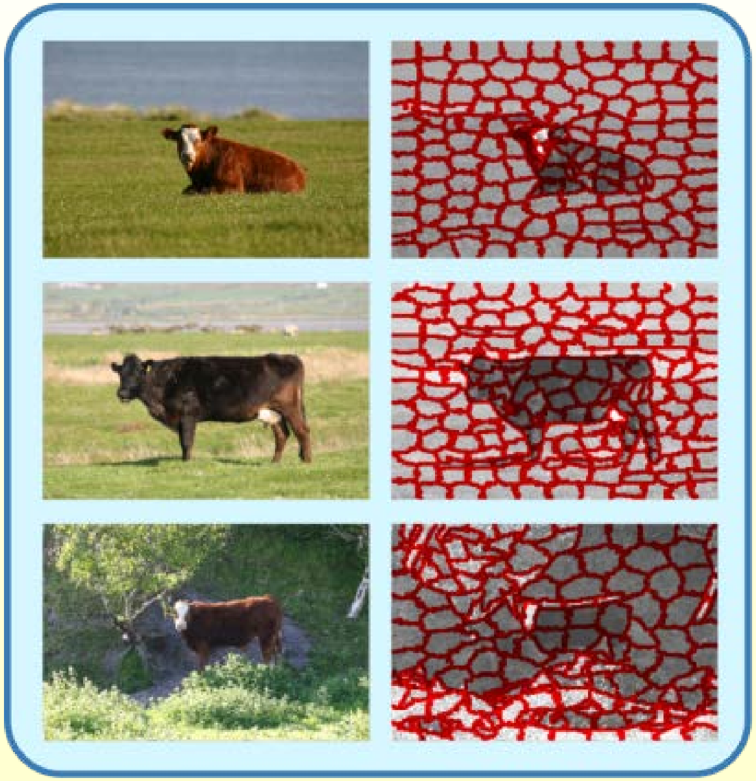
Graph for iCoseg-Ferrari



Graph for PASCAL-Plane

Superpixel Representation

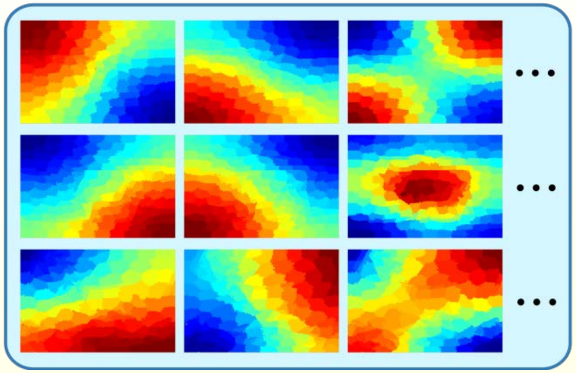
- Over-segment images into super-pixels
- Build a graph on super-pixels
  - Nodes: super-pixels
  - Edges weighted by length of shared boundary



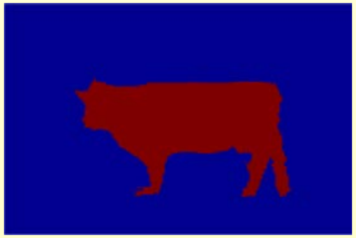
Encoding Functions over Graphs

- Basis of functional space
  - : First M Laplacian eigenfunctions of the graph

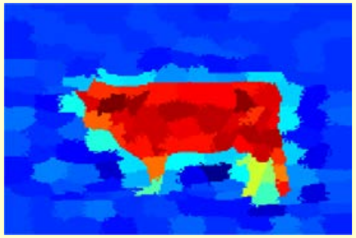
$$f = \sum_{j=1}^{M} f_j b_i^j = B_i \mathbf{f}$$



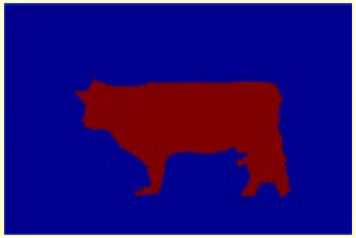
- Reconstruct any function with small error (M=30)



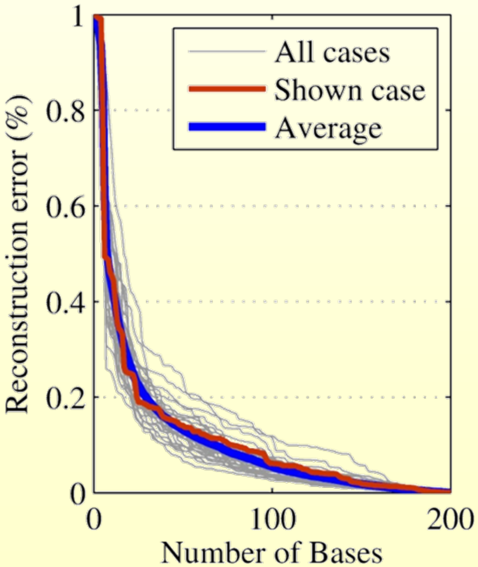
Binary indicator function



Reconstructed function



Thresholded reconstructed function



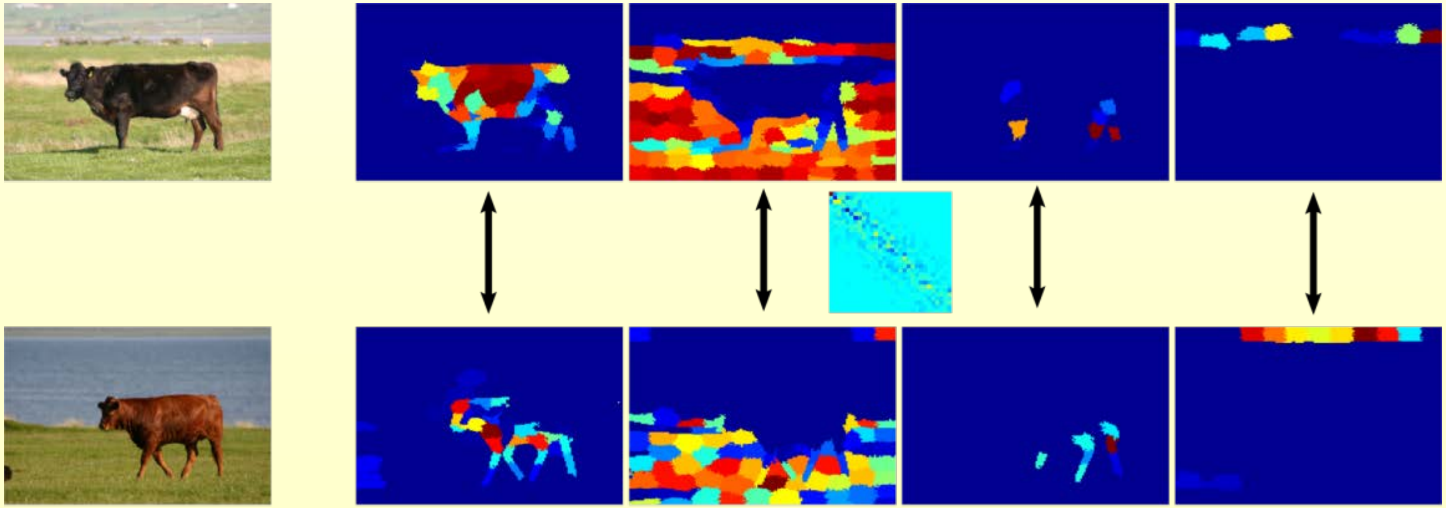
Reconstruction error

Joint Estimation of Functional Maps, I

- Functional map:
  - A linear map between functions in two functional spaces

$$\mathbf{f}' = X_{ij}\mathbf{f} \qquad X_{ij} \in \mathcal{R}^{M \times M}$$

  - Can be recovered by a set of probe functions



Joint Estimation of Functional Maps, I

- Recover functional maps by aligning image features:

$$f_{ij}^{\text{feature}} = \|X_{ij}D_i - D_j\|_1$$

- Features (probe functions) for each super-pixel:
  - average RGB color, 3-dimensional;
  - 64 dimensional RGB color histogram;
  - 300-dimensional bag-of-visual-words.

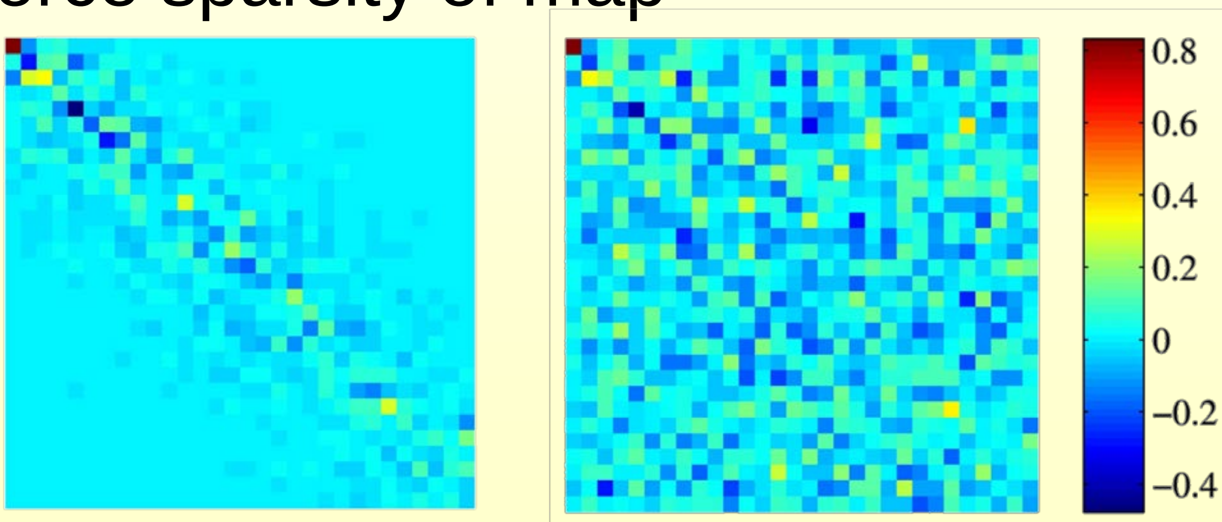
Joint Estimation of Functional Maps, II

- Regularization term:

Λ_i, Λ_j diagonal matrices of Laplacian eigenvalues

$$f_{ij}^{\text{reg}} = \|X_{ij}\Lambda_i - \Lambda_j X_{ij}\|^2$$

  - Correspond bases of similar spectra
  - Enforce sparsity of map



Map with regularization　　Map without regularization

Joint Estimation of Functional Maps, III

- Incorporating map cycle consistency:
  - A transported function along any loop should be identical to the original function:

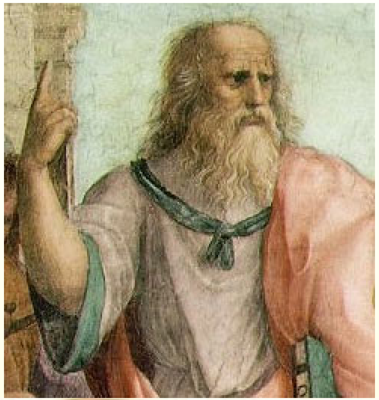
$$X_{i_k i_0} \cdots X_{i_1 i_2} X_{i_0 i_1} \mathbf{f} = \mathbf{f} \quad \Longleftrightarrow \quad X_{ij} Y_i = Y_j, \quad \forall (i,j) \in \mathcal{G}$$

  - Consistency term:

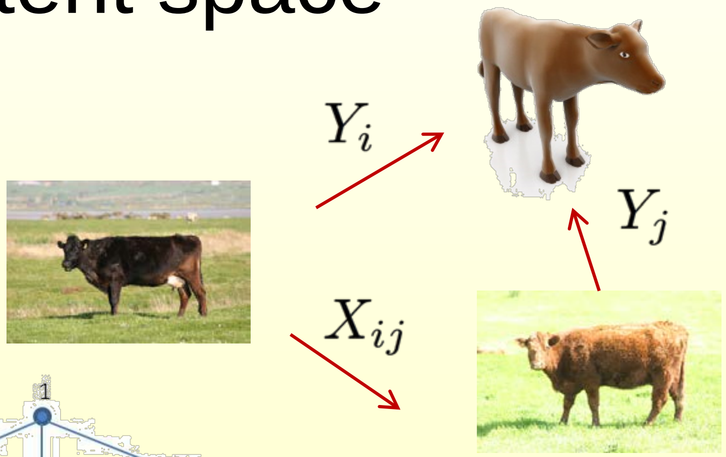
$$f^{\text{cons}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} f_{ij}^{\text{cons}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} \|X_{ij} Y_i - Y_j\|_{\mathcal{F}}^2$$

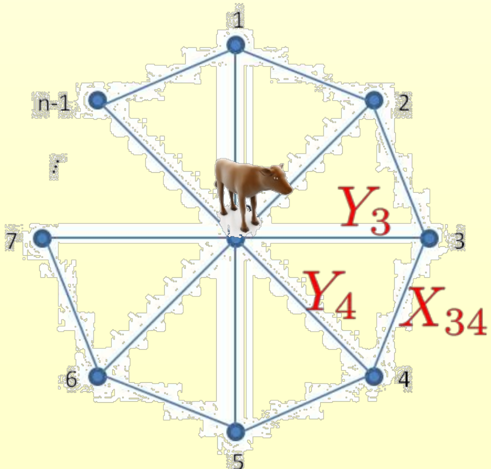
Image global similarity weight via GIST

Joint Estimation of Functional Maps, III

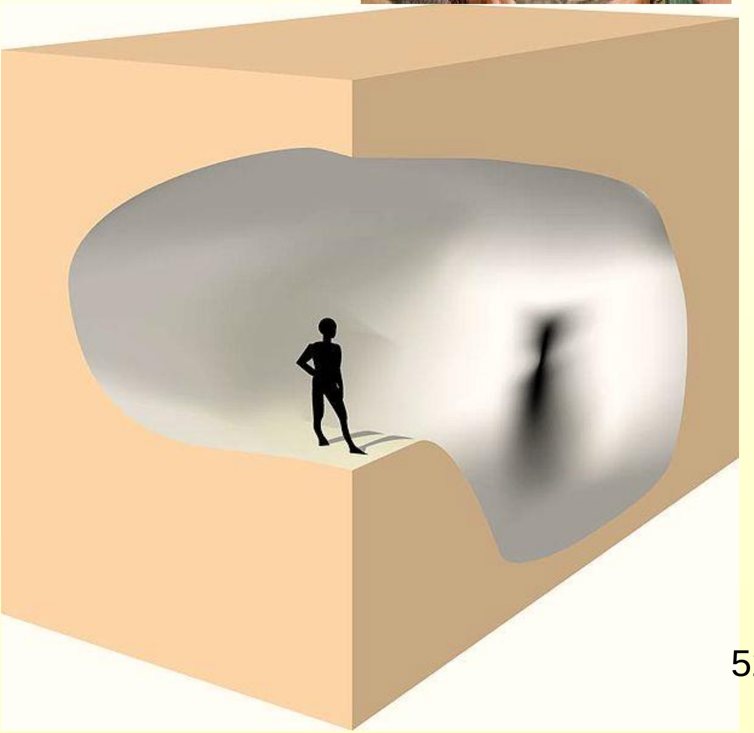


- Plato's allegory of the cave: a latent space





$$X_{ij} \approx Y_j^{-1} Y_i$$



X 30x30, *Y* 30x20

Joint Estimation of Functional Maps, IV

- Overall optimization

$$\min \sum_{(i,j)\in\mathcal{G}} w_{ij}\left(f_{ij}^{\text{feature}} + \mu f_{ij}^{\text{reg}} + \lambda f_{ij}^{\text{cons}}\right)$$
$$s.t. \quad Y^T Y = I_m$$

- Alternating optimization:
  - Fix Y, solve X ⟹ Independent QP problems

$$X_{ij}^{\star} = \arg\min_X \left(f_{ij}^{\text{feature}} + \mu f_{ij}^{\text{reg}} + \lambda f_{ij}^{\text{cons}}\right)$$

  - Fix X, solve Y ⟹ Eigenvalue problem

$$\min \quad \mathrm{trace}(Y^T W Y) \qquad s.t. \quad Y^T Y = I_m$$

$$W_{ij} = \begin{cases} \sum_{(i,j')\in\mathcal{G}} w_{ij'}(I_m + X_{ij'}^T X_{ij'}) & i = j \\ -w_{ij}(X_{ji} + X_{ij}^T) & (i,j)\in\mathcal{G} \\ 0 & \text{otherwise.} \end{cases}$$

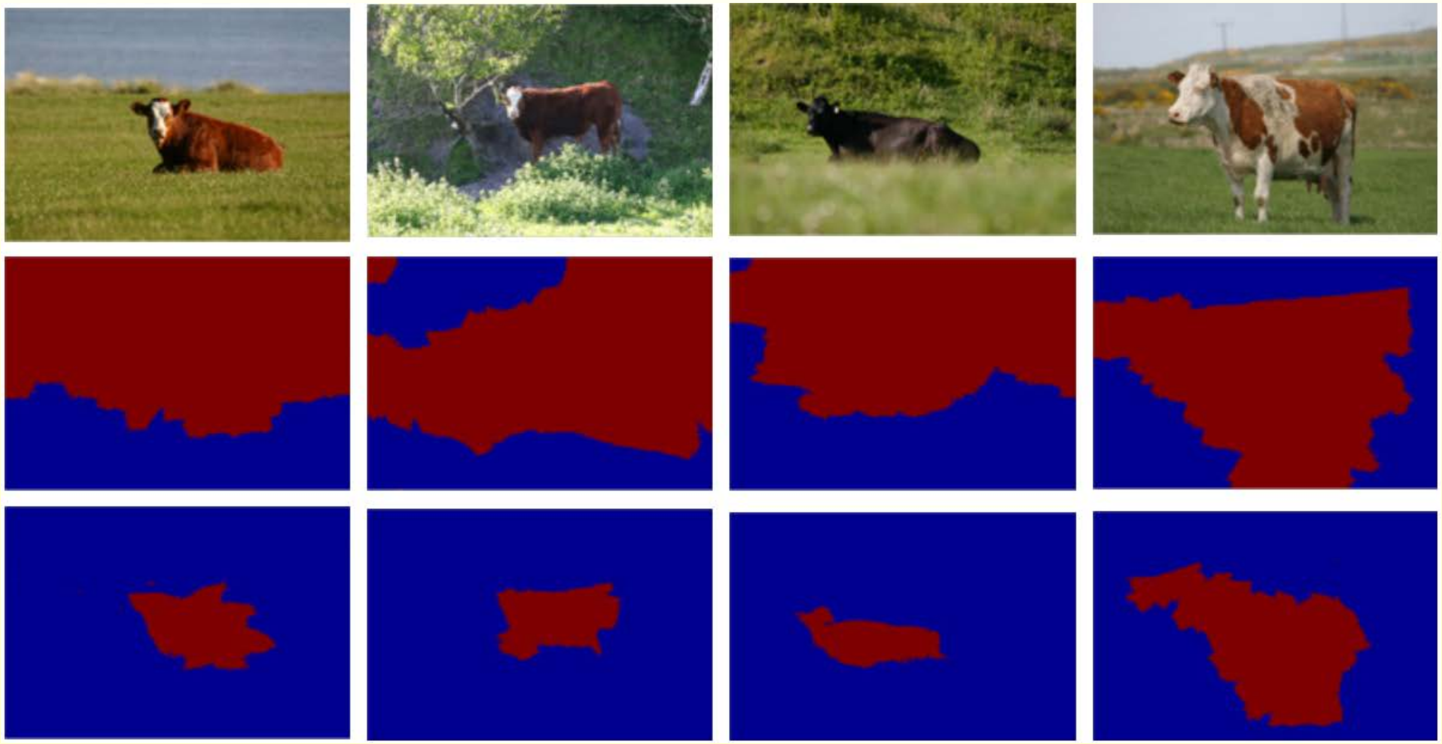
Consistency Matters

Source image

Target image

Without cycle consistency

With cycle consistency



Generating Consistent Segmentations

- Two objectives for segmentation functions
  - consistent under functional map transportation

$$f^{\text{map}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} \|X_{ij}\mathbf{f}_i - \mathbf{f}_j\|_{\mathcal{F}}^2$$

  - agreement with normalized cut scores:

We look for network fixed points!

$$f^{\text{seg}} = \sum_{i=1}^{N} \mathbf{f}_i^T B_i^T L_i B_i \mathbf{f}_i$$

Easy to incorporate labeled images with ground truth segmentation

- Joint optimization:

$$\min \; f^{\text{seg}} + \gamma f^{\text{map}} \quad s.t. \sum_{i=1}^{N} \|\mathbf{f}_i\|^2 = 1$$

Eigen-decomposition problem

Experiments

- iCoseg dataset
  - Very similar or the same object in each class;
  - 5~10 images per class.
- MSRC dataset
  - Similar objects in each class;
  - ~30 images per class.
- PASCAL data set
  - Retrieved from PASCAL VOC 2012 challenge;
  - All images with the same object label;
  - Larger scale;
  - Larger variability.

- iCoseg data set
- New unsupervised method
  - Mostly outperforms other unsupervised methods
  - Sometimes even outperforms supervised methods
  - Supervised input is easily added and further improves the results

Supervised method

Class	Joulin ’10	Rubio ’12	Vicente ’11	Fmaps -uns
Alaska Bear	74.8	86.4	90.0	**90.4**
Red Sox Players	73.0	90.5	90.9	**94.2**
Stonehenge1	56.6	87.3	63.3	**92.5**
Stonehenge2	86.0	88.4	88.8	87.2
Liverpool FC	76.4	82.6	87.5	**89.4**
Ferrari	85.0	84.3	89.9	**95.6**
Taj Mahal	73.7	88.7	91.1	**92.6**
Elephants	70.1	75.0	43.1	**86.7**
Pandas	84.0	60.0	92.7	88.6
Kite	87.0	89.8	90.3	**93.9**
Kite panda	73.2	78.3	90.2	**93.1**
Gymnastics	90.9	87.1	91.7	90.4
Skating	82.1	76.8	77.5	78.7
Hot Balloons	85.2	89.0	90.1	**90.4**
Liberty Statue	90.6	91.6	93.8	**96.8**
Brown Bear	74.0	80.4	95.3	88.1
Average	78.9	83.5	85.4	**90.5**

Kuettel’12 (Supervised)		Unsupervised Fmaps
Image+transfer	Full model	
87.6	91.4	90.5

MSRC

Unsupervised performance comparison

Class	N	Joulin '10	Rubio '12	Fmaps -uns
Cow	30	81.6	80.1	**89.7**
Plane	30	73.8	77.0	**87.3**
Face	30	84.3	76.3	**89.3**
Cat	24	74.4	77.1	**88.3**
Car(front)	6	87.6	65.9	87.3
Car(back)	6	85.1	52.4	**92.7**
Bike	30	63.3	62.4	**74.8**

Supervised performance comparison

Class	Vicente '11	Kuettel '12	Fmaps -s
Cow	94.2	92.5	**94.3**
Plane	83.0	86.5	**91.0**
Car	79.6	88.8	83.1
Sheep	94.0	91.8	**95.6**
Bird	95.3	93.4	**95.8**
Cat	92.3	92.6	**94.5**
Dog	93.0	87.8	91.3

PASCAL

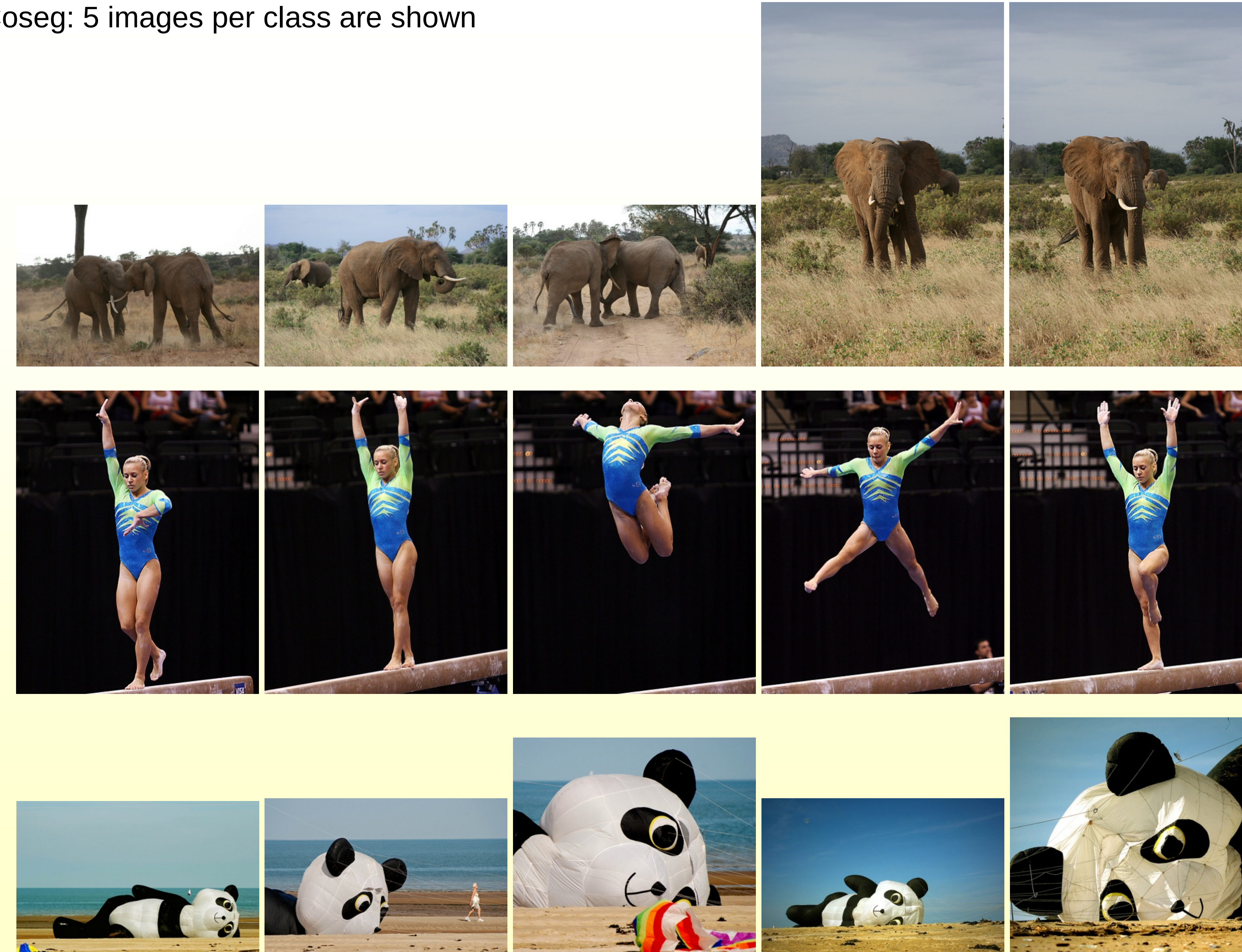
Class	N	L	Kuettel '12	Fmaps -s	Fmaps -uns
Plane	178	88	90.7	**92.1**	89.4
Bus	152	78	81.6	**87.1**	80.7
Car	255	128	76.1	**90.9**	82.3
Cat	250	131	77.7	**85.5**	82.5
Cow	135	64	82.5	**87.7**	85.5
Dog	249	121	81.9	**88.5**	84.2
Horse	147	68	83.1	**88.9**	87.0
Sheep	120	63	83.9	**89.6**	86.5

- New method mostly outperforms the state-of-the-art techniques in both supervised and unsupervised settings

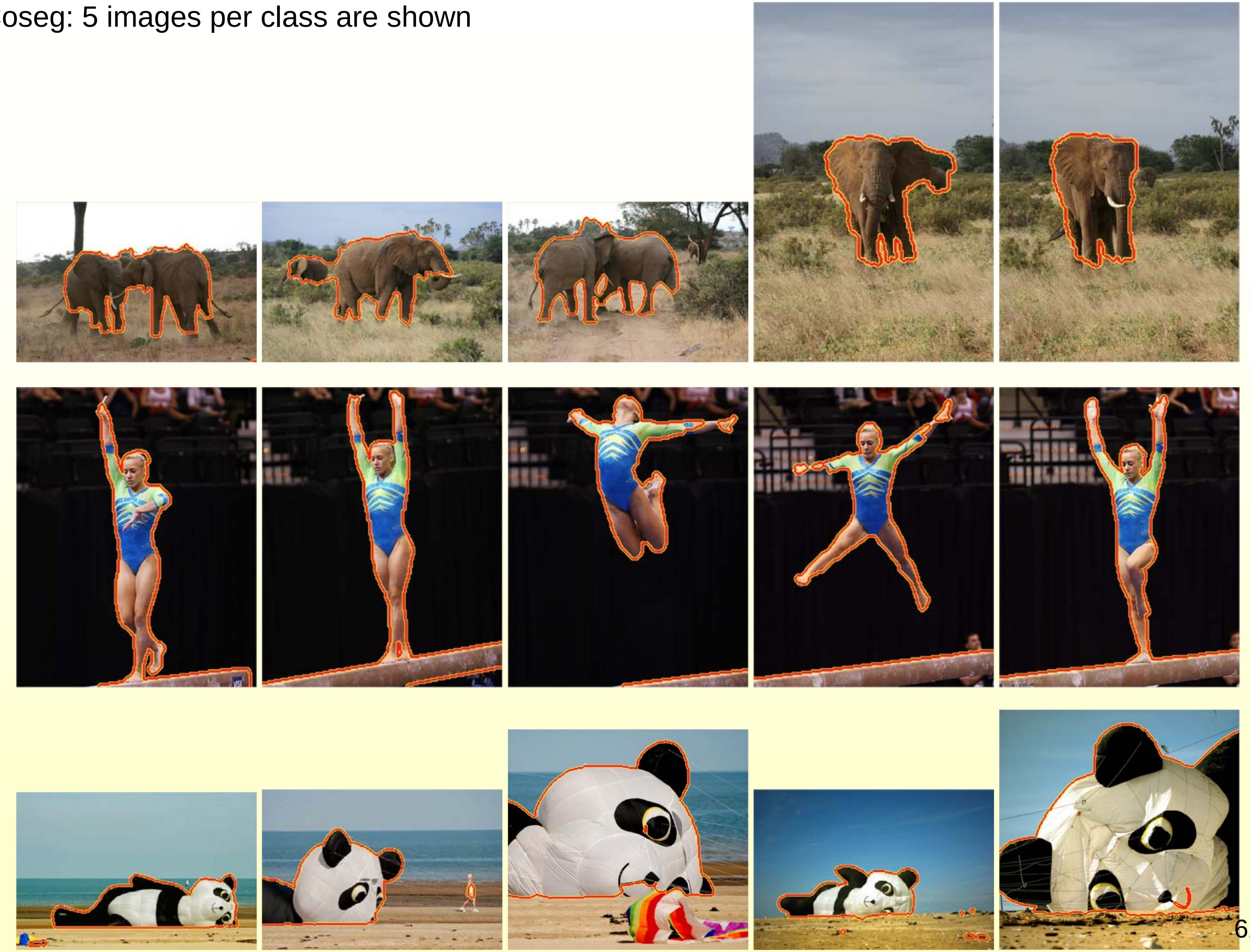
iCoseg: 5 images per class are shown

iCoseg: 5 images per class are shown

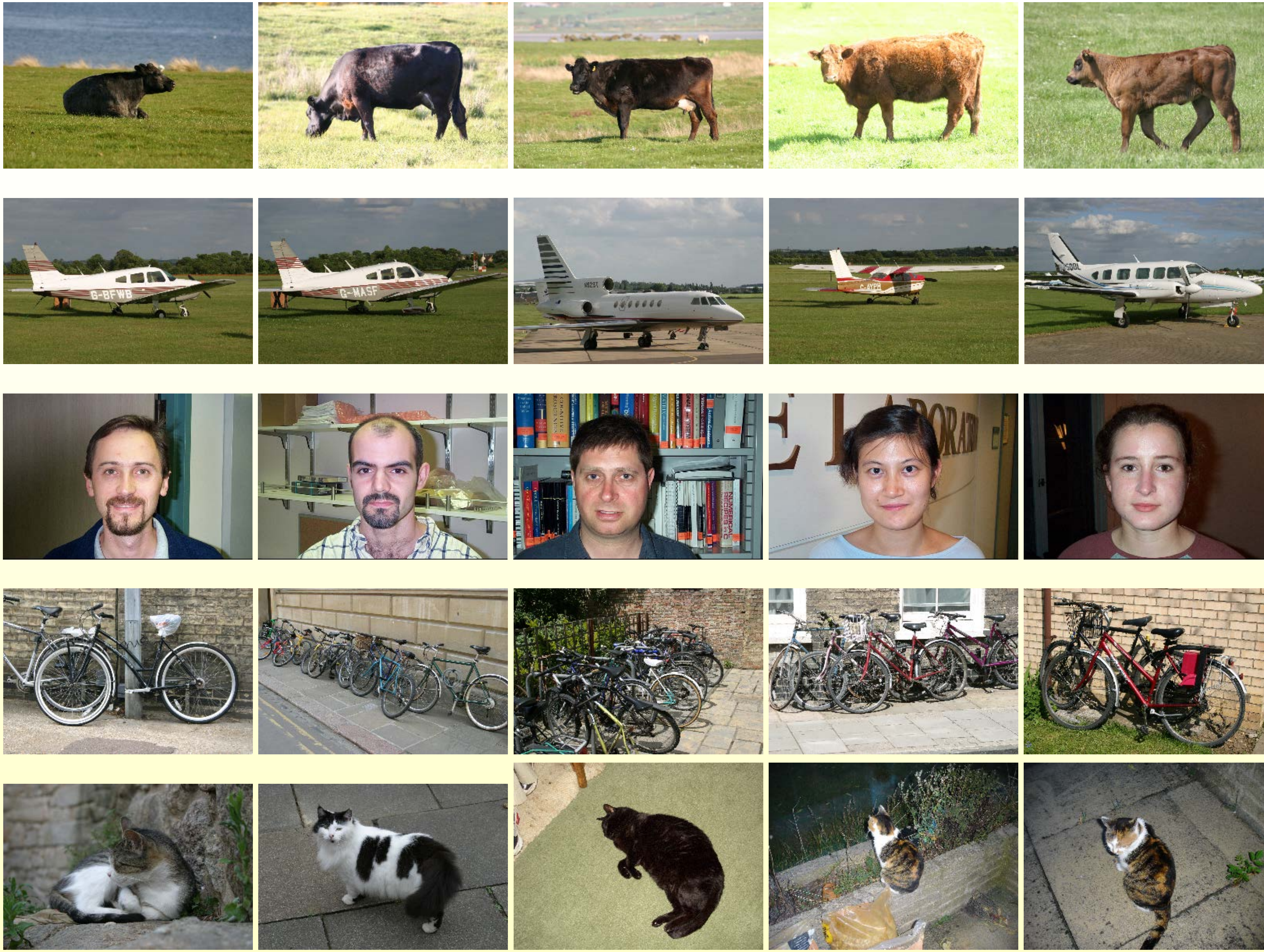
iCoseg: 5 images per class are shown



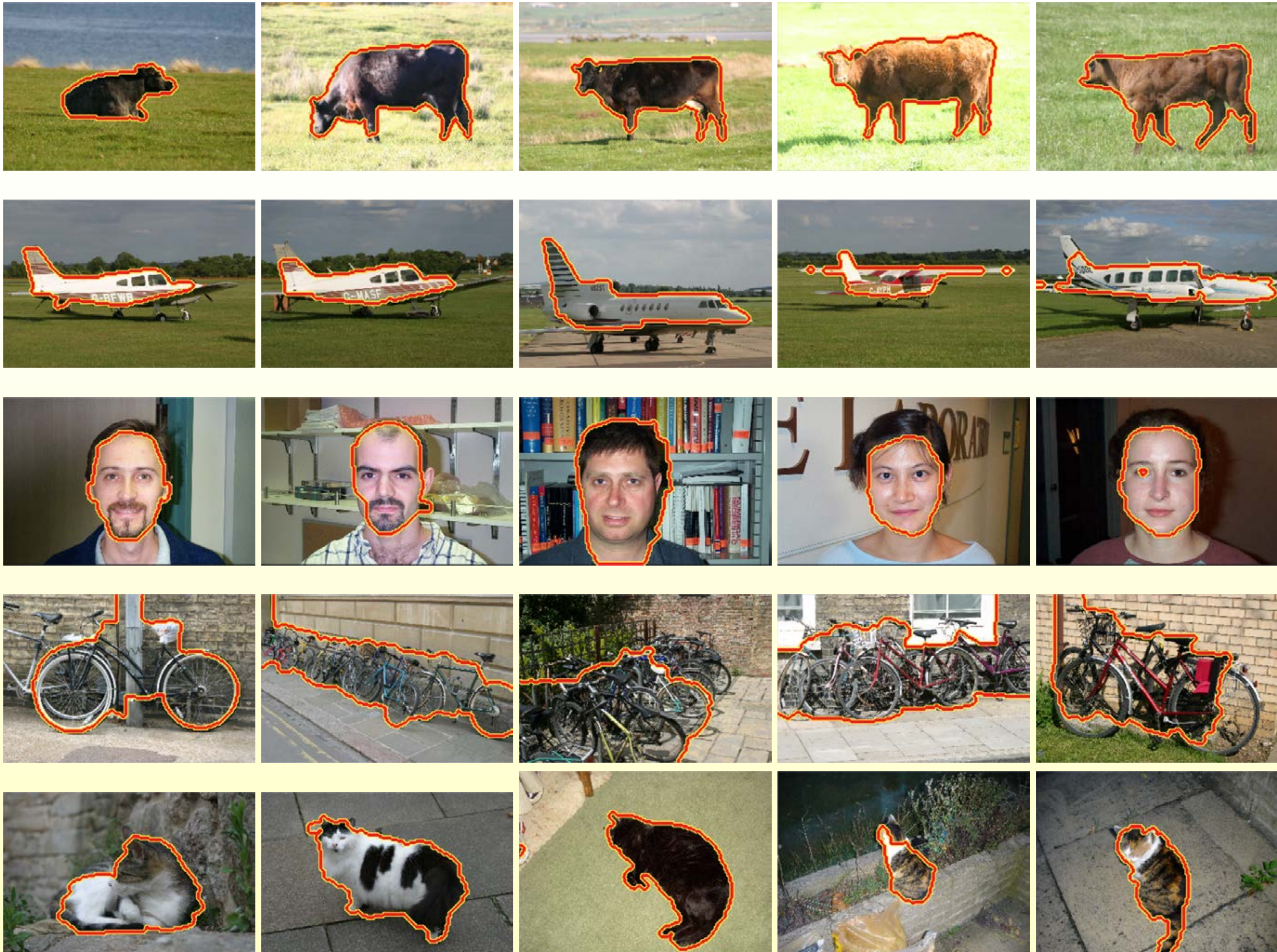
iCoseg: 5 images per class are shown



MSRC: 5 images per class are shown

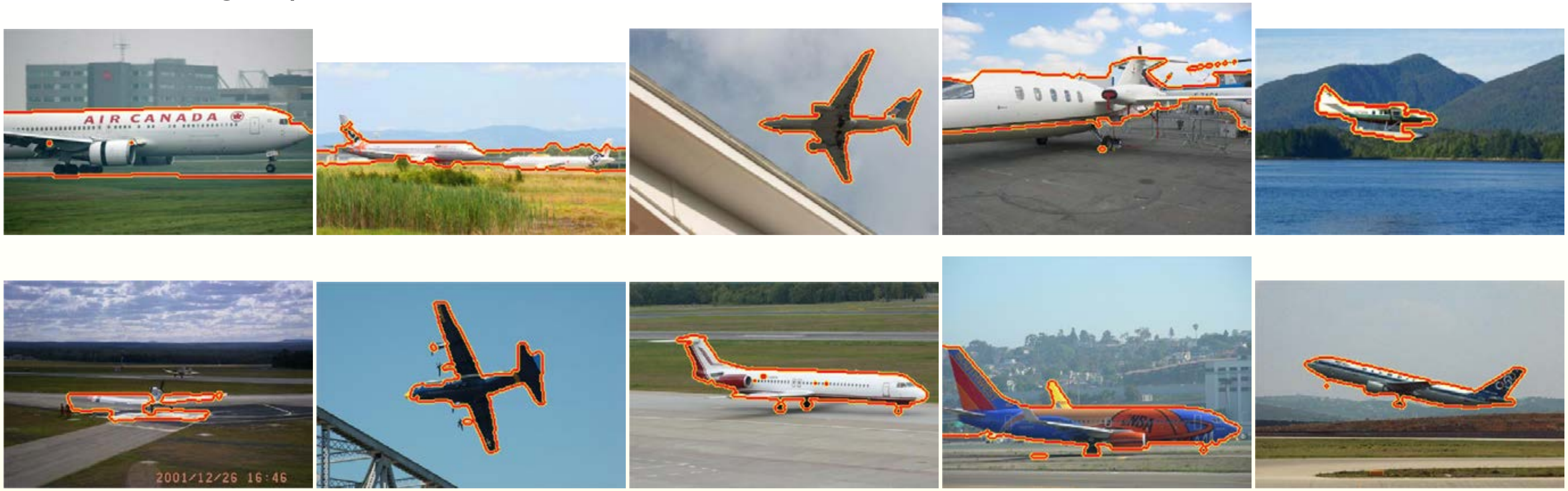


MSRC: 5 images per class are shown



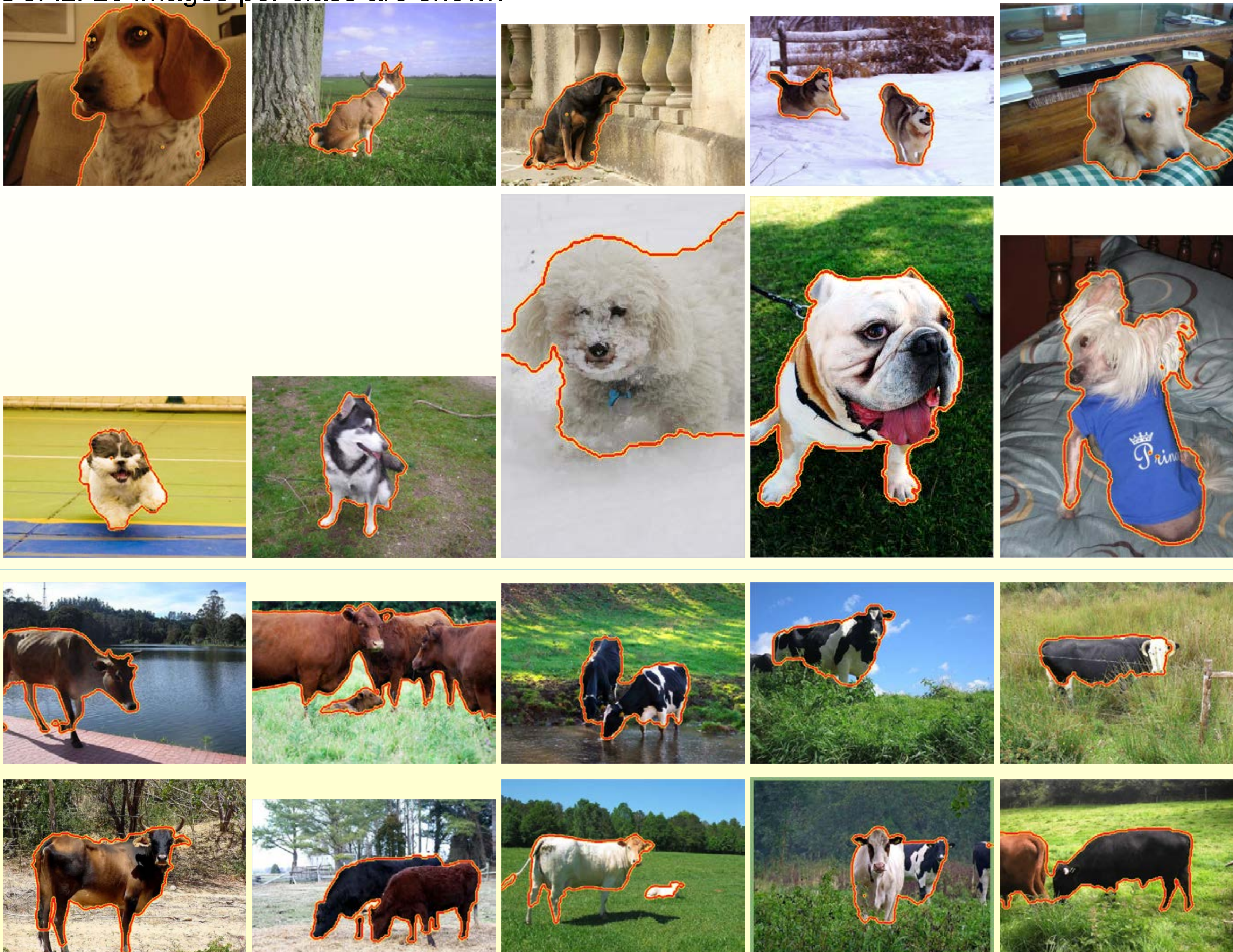
PASCAL: 10 images per class are shown

PASCAL: 10 images per class are shown



PASCAL: 10 images per class are shown

PASCAL: 10 images per class are shown



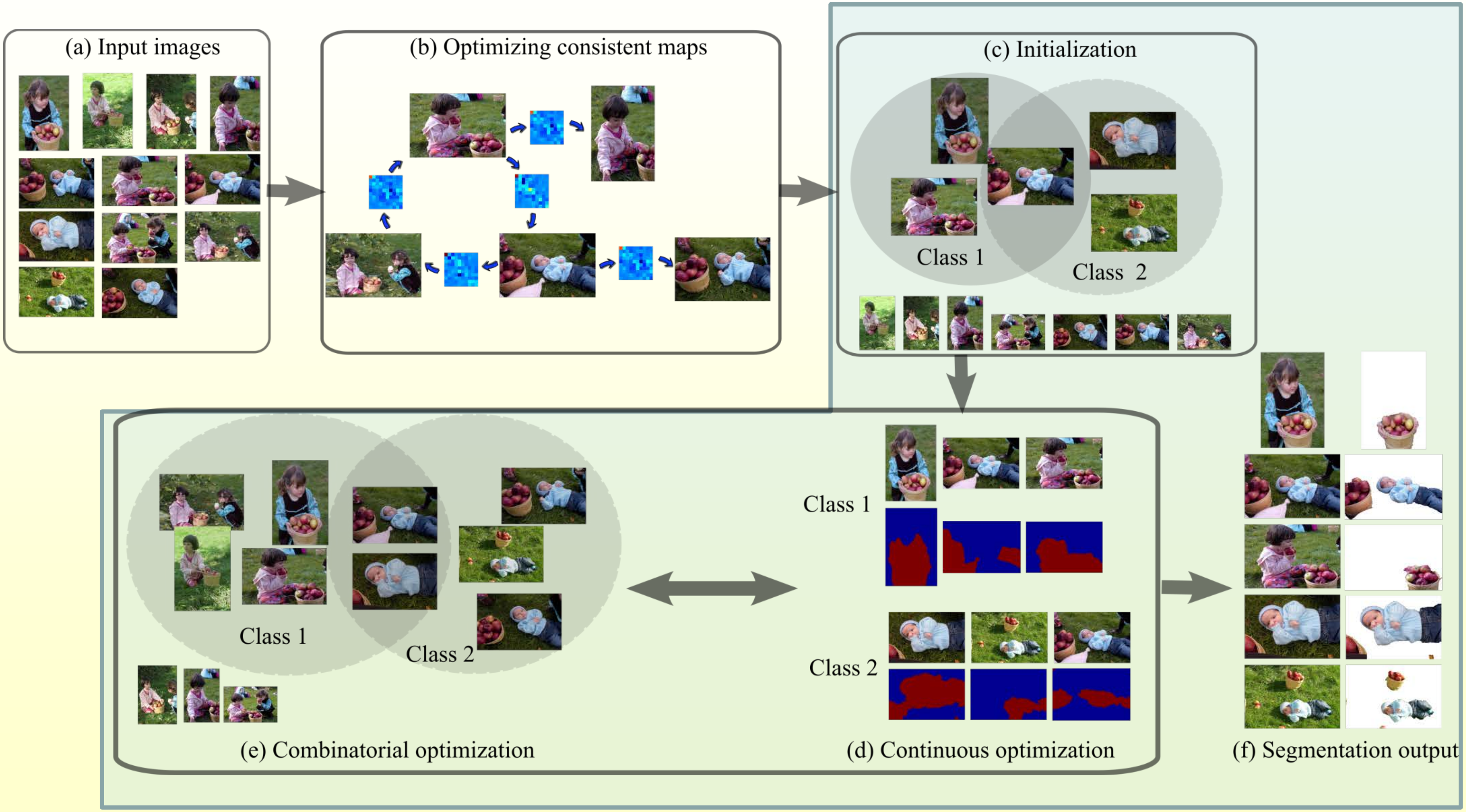
Multi-Class Co-Segmentation

[F. Wang, Q. Huang, M. Ovsjanikov, L. G., CVPR'14]

- Input:
  - A collection of N images sharing M objects
  - Each image contains a subset of the objects

- Output
  - Discovery of what objects appear in each image
  - Their pixel-level segmentation

Framework



(a) Input images
(b) Optimizing consistent maps
(c) Initialization
Class 1
Class 2
Class 1
Class 2
Class 1
Class 2
Class 1
Class 2
(e) Combinatorial optimization
(d) Continuous optimization
(f) Segmentation output

Optimizing Segmentation Functions

- Alternating between:
  - Continuous optimization:
    - Optimal segmentation functions in each class
  - Combinatorial optimization:
    - Class assignment by propagating segmentation functions

Experimental Results

- Accuracy
  - Intersection-over-union
  - Find the best one-to-one matching between each cluster and each ground-truth object.
- Benchmark datasets
  - MSRC: 30 images, 1 class (degenerated case);
  - FlickrMFC data set: 20 images, 3~6 classes
  - PASCAL VOC: 100~200 images, 2 classes

Experimental Results

class	N	M	Kim'12	Kim'11	Joulin '10	Mukherjee '11	Ours
Apple	20	6	40.9	32.6	24.8	25.6	**46.6**
Baseball	18	5	31.0	31.3	19.2	16.1	**50.3**
butterfly	18	8	29.8	32.4	29.5	10.7	**54.7**
Cheetah	20	5	32.1	40.1	50.9	41.9	**62.1**
Cow	20	5	35.6	43.8	25.0	27.2	**38.5**
Dog	20	4	34.5	35.0	32.0	30.6	**53.8**
Dolphin	18	3	34.0	47.4	37.2	30.1	**61.2**
Fishing	18	5	20.3	27.2	19.8	18.3	**46.8**
Gorilla	18	4	41.0	38.8	41.1	28.1	**47.8**
Liberty	18	4	31.5	41.2	44.6	32.1	**58.2**
Parrot	18	5	29.9	36.5	35.0	26.6	**54.1**
Stonehenge	20	5	35.3	49.3	47.0	32.6	**54.6**
Swan	20	3	17.1	18.4	14.3	16.3	**46.5**
Thinker	17	4	25.6	34.4	27.6	15.7	**68.6**
Average	-	-	31.3	36.3	32.0	25.1	**53.1**

Performance comparison on the MFCFlickr dataset

class	N	Joulin'10	Kim'11	Mukherjee'11	Ours
Bike	30	43.3	29.9	42.8	**51.2**
Bird	30	47.7	29.9	-	**55.7**
Car	30	59.7	37.1	52.5	**72.9**
Cat	24	31.9	24.4	5.6	**65.9**
Chair	30	39.6	28.7	39.4	**46.5**
Cow	30	52.7	33.5	26.1	**68.4**
Dog	30	41.8	33.0	-	**55.8**
Face	30	70.0	33.2	40.8	60.9
Flower	30	51.9	40.2	-	**67.2**
House	30	51.0	32.2	66.4	56.6
Plane	30	21.6	25.1	33.4	**52.2**
Sheep	30	66.3	60.8	45.7	**72.2**
Sign	30	58.9	43.2	-	**59.1**
Tree	30	67.0	61.2	55.9	**62.0**

Performance comparison on the MSRC dataset

class	N	NCut	MNcut	Ours
Bike + person	248	27.3	30.5	**40.1**
Boat + person	260	29.3	32.6	**44.6**
Bottle + dining table	90	37.8	39.5	**47.6**
Bus + car	195	36.3	39.4	**49.2**
bus + person	243	38.9	41.3	**55.5**
Chair + dining table	134	32.3	30.8	**40.3**
Chair + potted plant	115	19.7	19.7	**22.3**
Cow + person	263	30.5	33.5	**45.0**
Dog + sofa	217	44.6	42.2	**49.6**
Horse + person	276	27.3	30.8	**42.1**
Potted plant + sofa	119	37.4	37.5	**40.7**

Performance comparison on the PASCAL-multi dataset

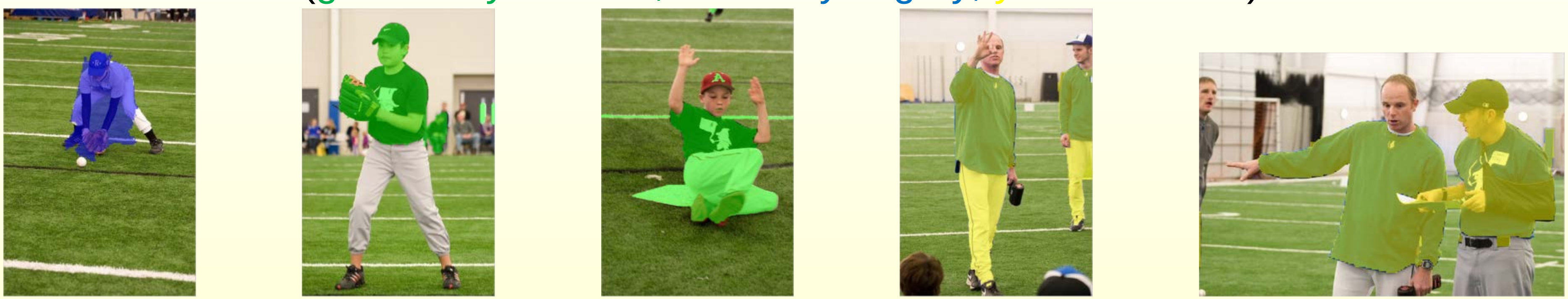
Apple + picking

Baseball + kids

Butterfly + blossom

Apple + picking (red: apple bucket; magenta: girl in red; yellow: girl in blue; green: baby; cyan: pum

Baseball + kids (green: boy in black; blue: boy in grey; yellow: coach.)



Butterfly + blossom (green: butterfly in orange; yellow: butterfly in yellow; cyan: red flowe

Cheetah + Safari

Cow + pasture

Dog + park

Dolphin + aquarium

Cheetah + Safari (red: cheetah; yellow: lion; magenta: monkey.)

Cow + pasture (red: black cow; green: brown cow; blue: man in blue.)

Dog + park (red: black dog; green: brown dog; blue: white dog.)

Dolphin + aquarium (red: killer whale; green: dolphin.)

Fishing + Alaska

Gorilla + zoo

Liberty + statue

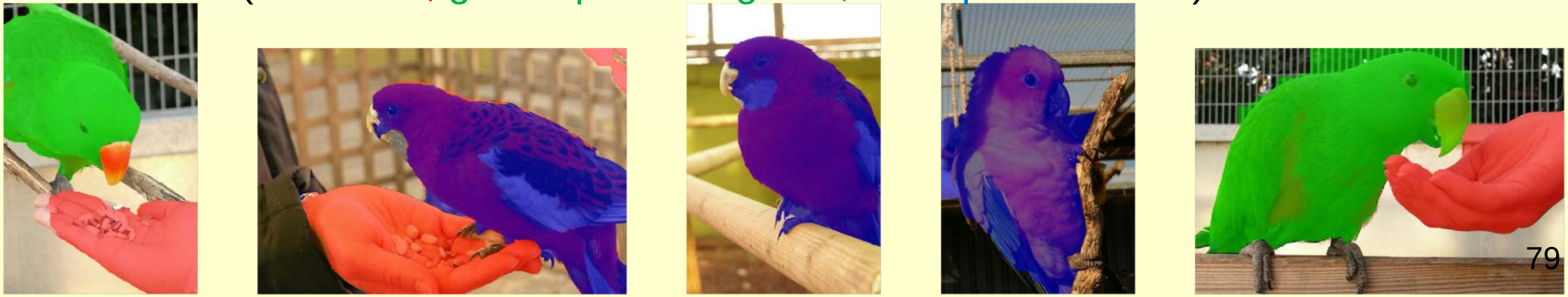
Parrot + zoo

Fishing + Alaska (blue: man in white; green: man in gray; magenta: woman in gray; yellow: salmon.

Gorilla + zoo (blue: gorilla; yellow: brown orangutan)

Liberty + statue (blue: empire state building; green: red boat; yellow: liberty statue.)

Parrot + zoo (red: hand; green: parrot in green; blue: parrot in red.)



Stonehenge

Swan + zoo

Thinker + Rodin

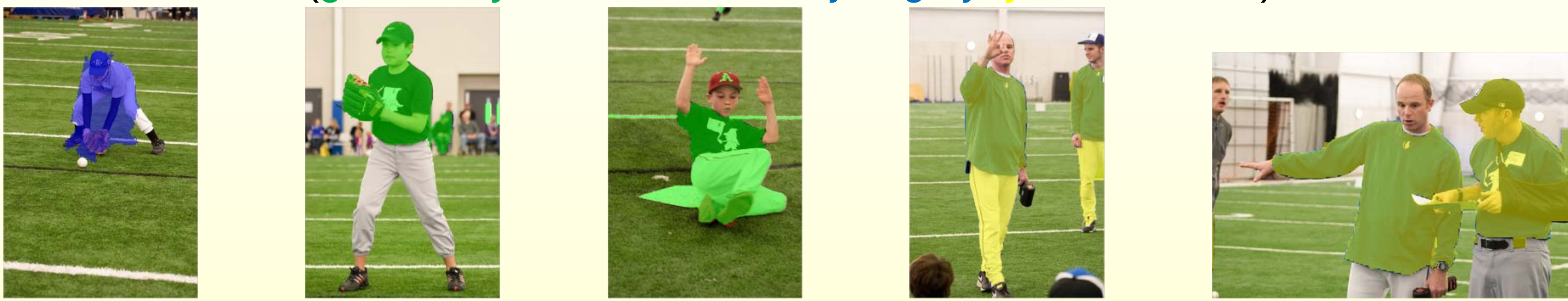
Stonehenge (blue: cow in white; yellow: person; magenta: stonehenge.)

Swan + zoo (blue: gray swan; green: black swan.)

Thinker + Rodin (red: sculpture Thinker; green: sculpture Venus; blue: Van Gogh.)

Apple + picking (red: apple bucket; magenta: girl in red; yellow: girl in blue; green: baby; cyan: pump

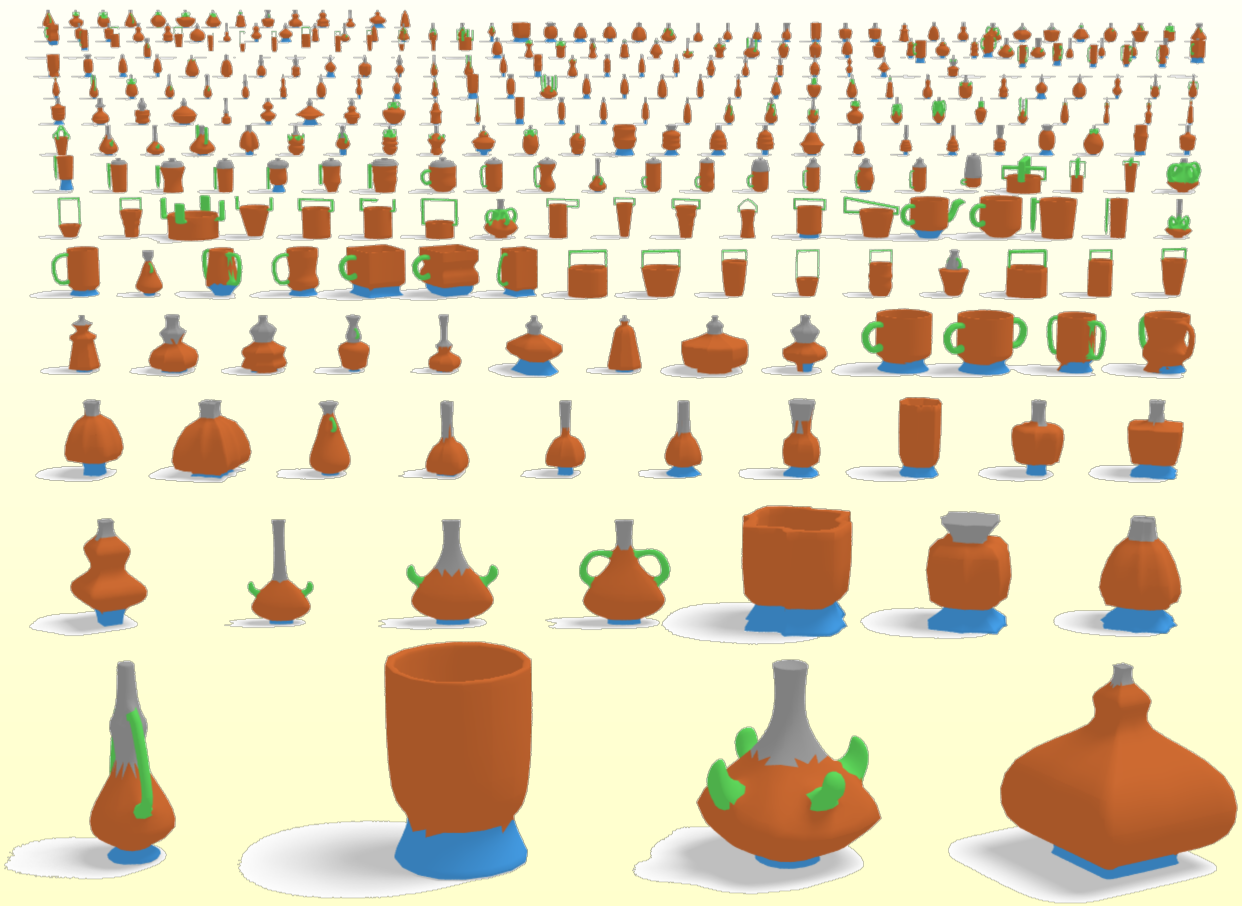
Baseball + kids (green: boy in black; blue: boy in grey; yellow: coach.)



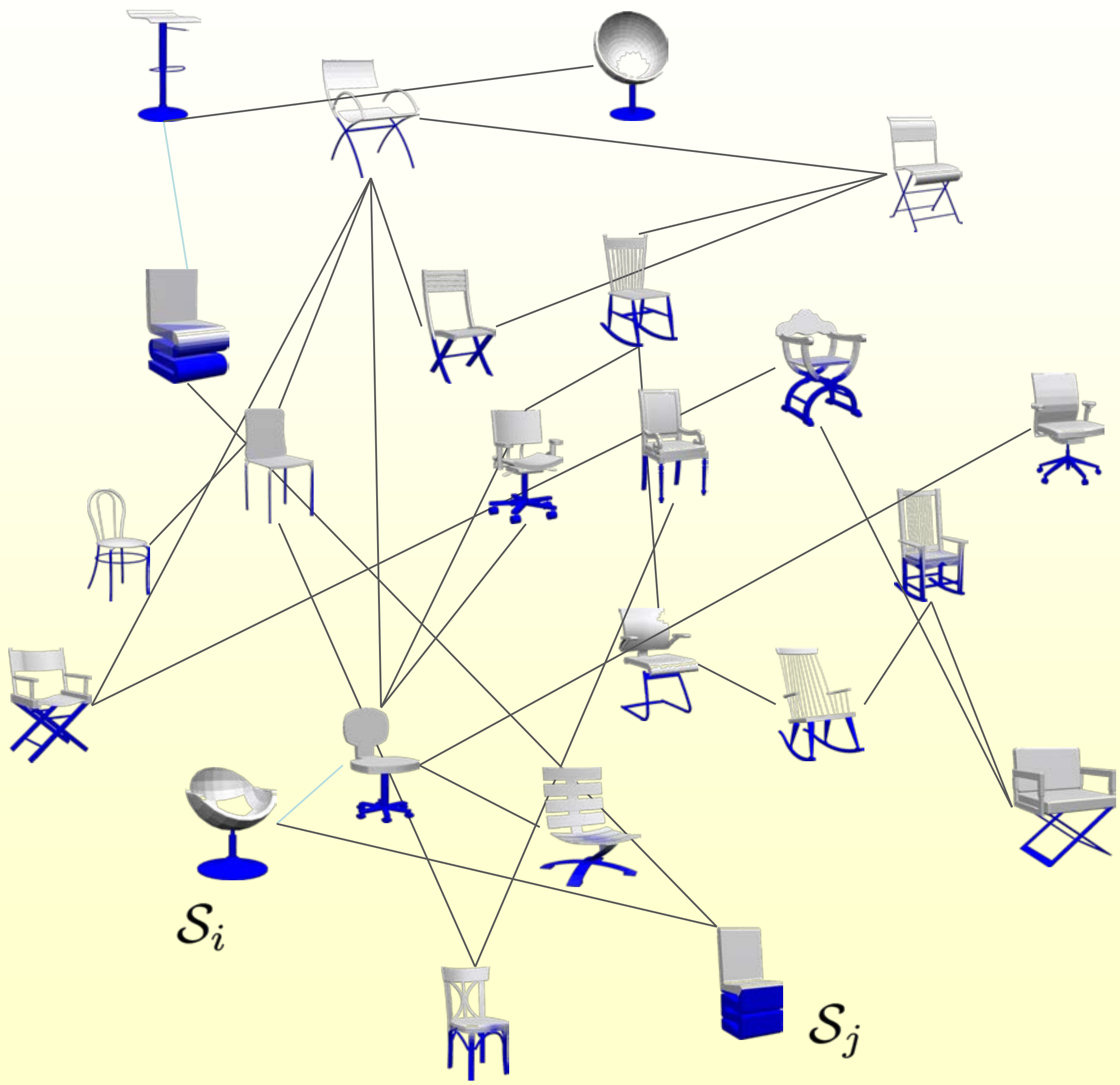
Butterfly + blossom (green: butterfly in orange; yellow: butterfly in yellow; cyan: red flowe

Consistent Shape Segmentation

[Q. Huang, F. Wang, L. Guibas, ’14]



First Build a Network

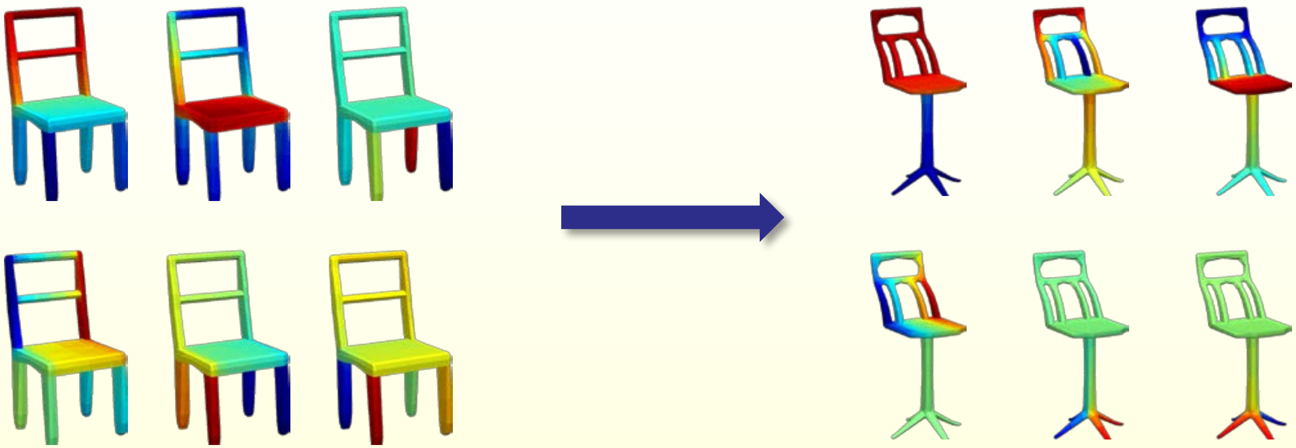


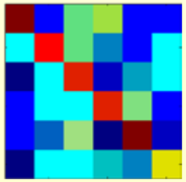
distance histogram

Use the D2 shape descriptor and connect each shape to its nearest neighbors

$$\mathcal{G} = (\mathcal{F}, \mathcal{E})$$

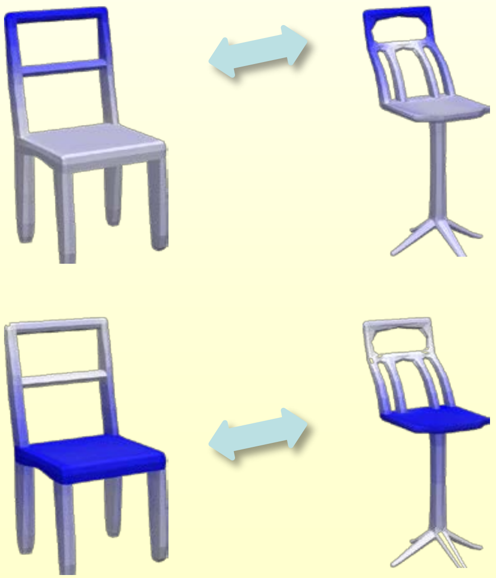
Start From Noisy Shape Descriptor Correspondences





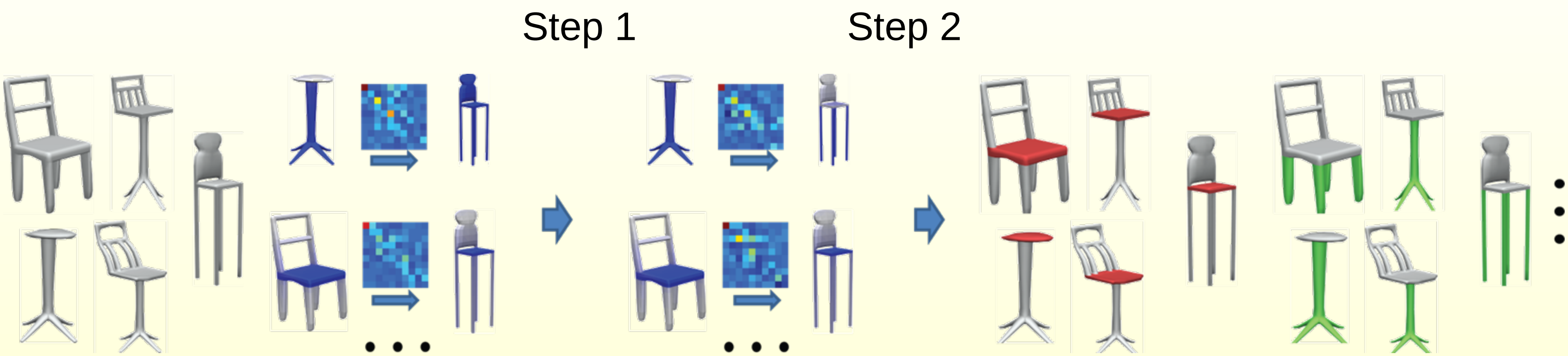
Lift to
functional form

$$C_i X_{ij} \approx D_j$$



$C_i \bullet \quad \bullet \quad \bullet \; D_i$

The Pipeline



Original shapes with noisy maps

Cleaned up maps

Consistent basis functions extracted

Joint Map Optimization

- Step 1: Convex low-rank recovery using robust PCA – we minimize over all *X*

convex!

trace norm

$\|X\|_\star = \sum_i \sigma_i(X)$

$$X^\star = \lambda\|X\|_\star + \min_X \sum_{(i,j)\in\mathcal{G}} \|X_{ij}C_{ij} - D_{ij}\|_{2,1}$$

$\|A\|_{2,1} = \sum_i \|\vec{a}_i\|$

Dual ADMM

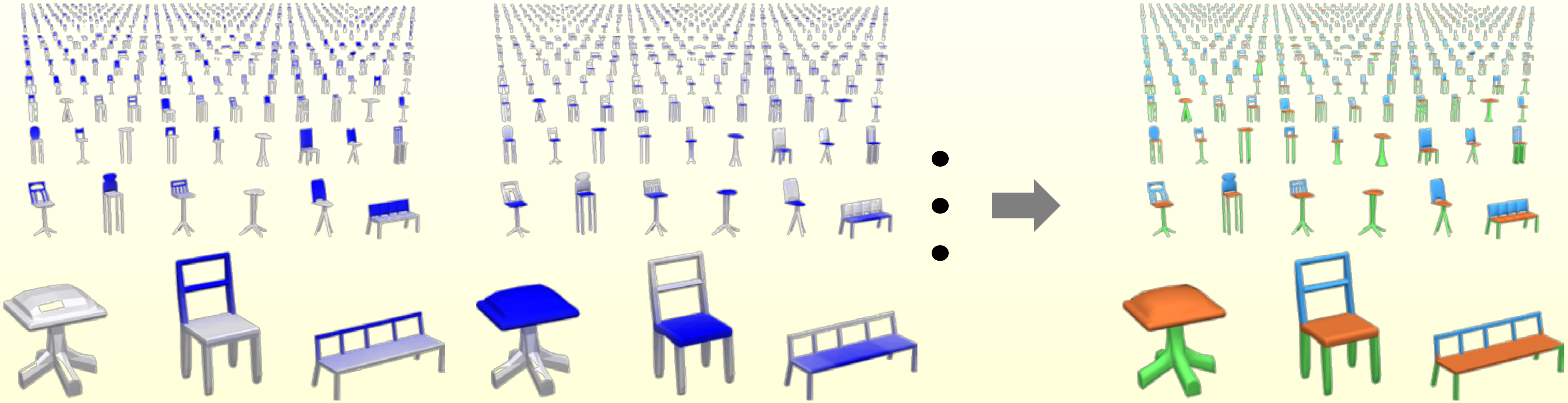
- Step 2: Perturb the above *X* to force the factorization

$$\sum_{1\le i,j\le N} \|X^\star_{ij} - Y_j^+ Y_i\|_F^2 + \mu \sum_{i=1}^{N} \sum_{1\le k<l\le L} (\mathbf{y}_{ik}^T \mathbf{y}_{il})^2$$

Non-linear least squares
Gauss-Newton descent

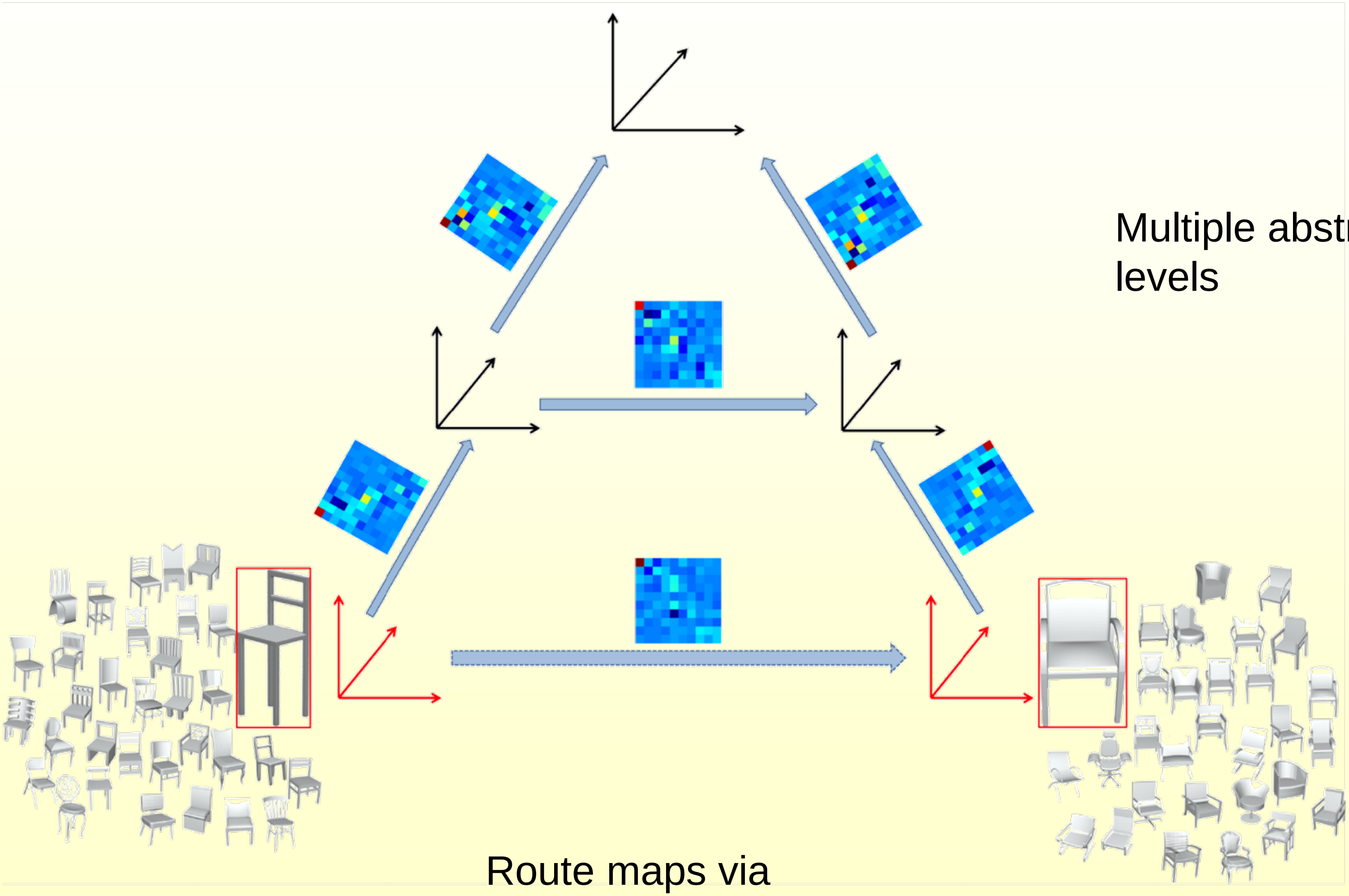
The Y_i give us the desired latent spaces

Consistent Shape Segmentation



Via 2nd order MRF on each shape independently

Hierarchical Scaling



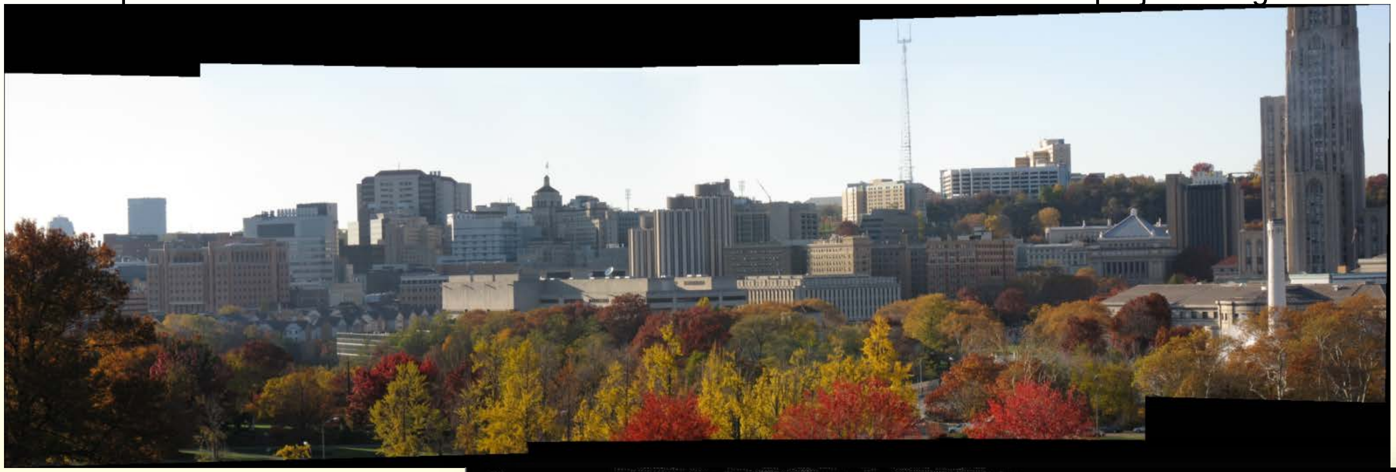
Multiple abstraction levels

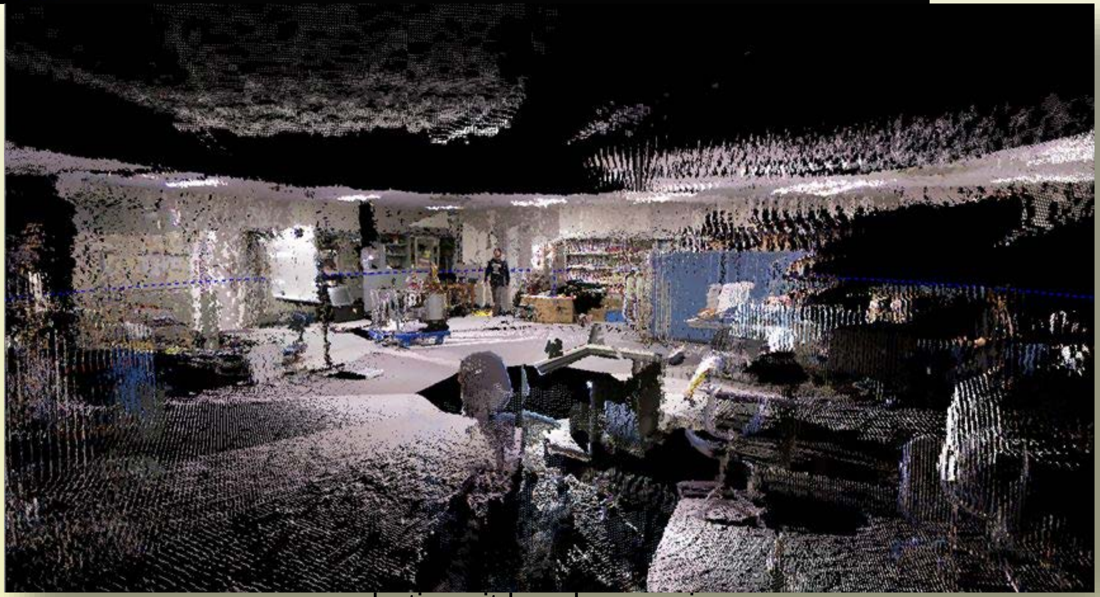
Route maps via the abstractions

The Network is the Abstraction

Mosaicking or SLAM at the Level of Functions

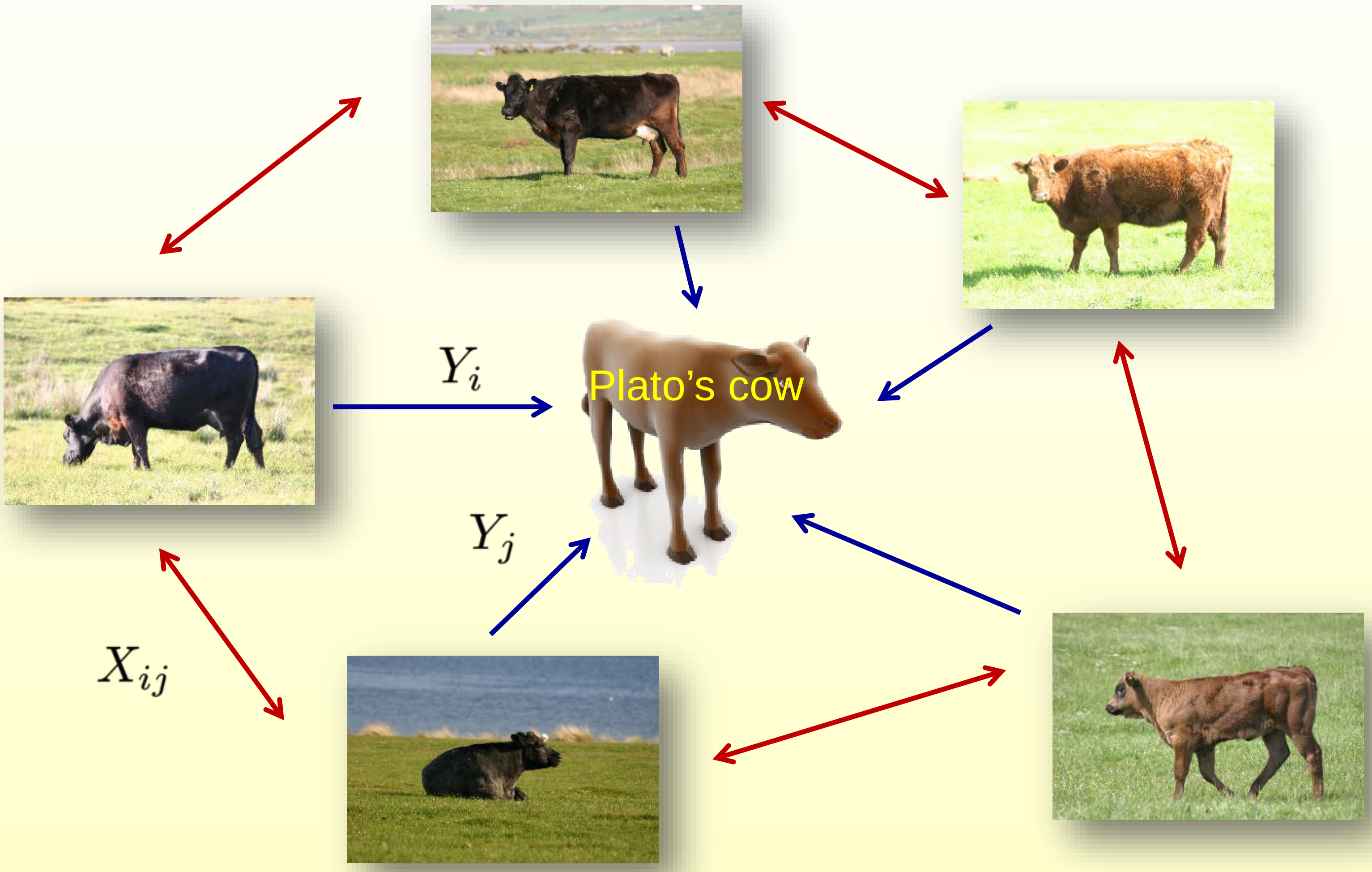
http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15463-f08/www/proj4/www/gme/



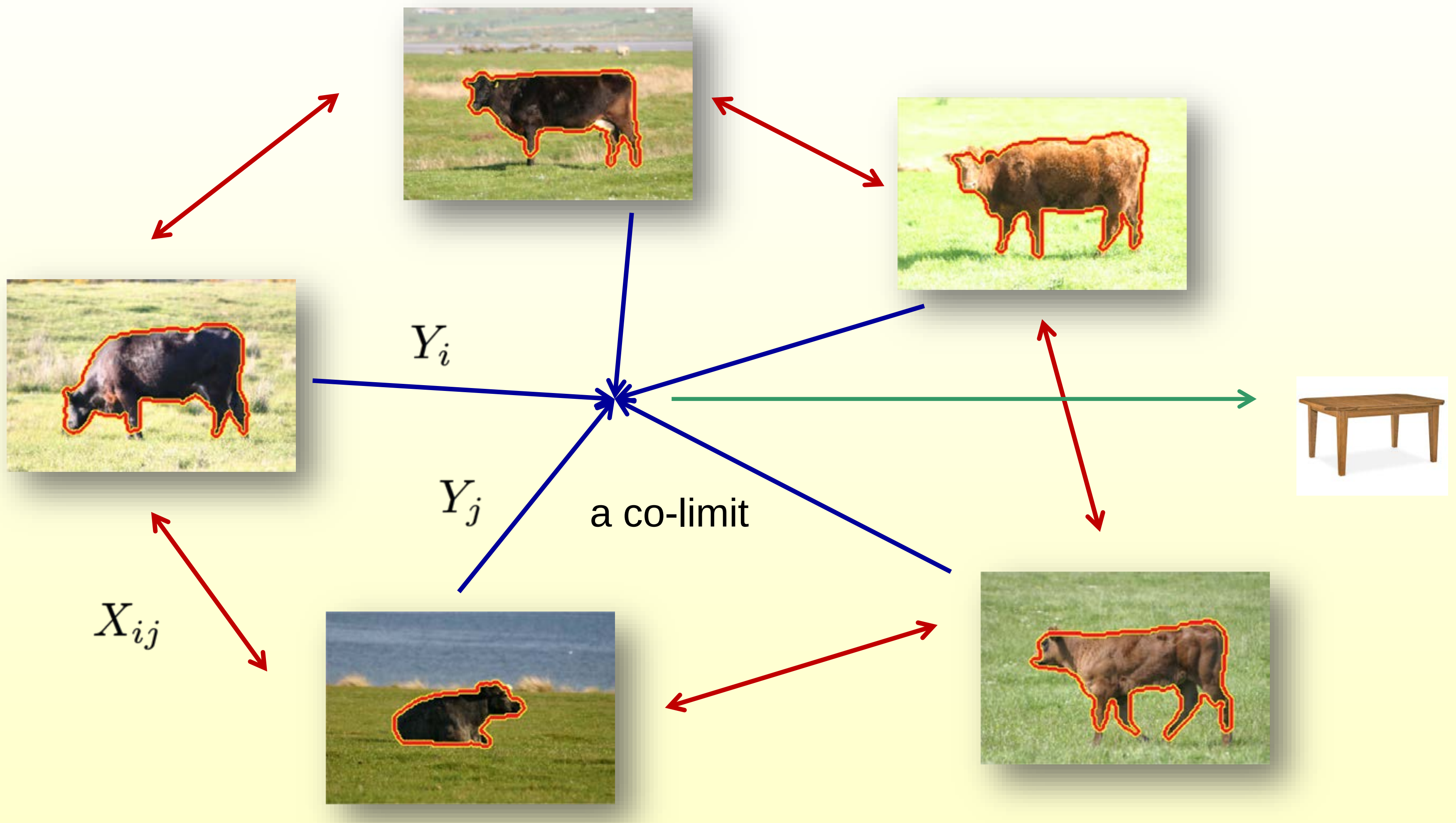


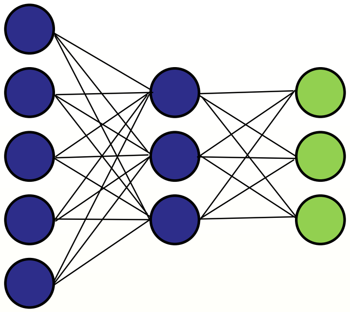
robotics.ait.kyushu-u.ac.jp

The Network is the Abstraction

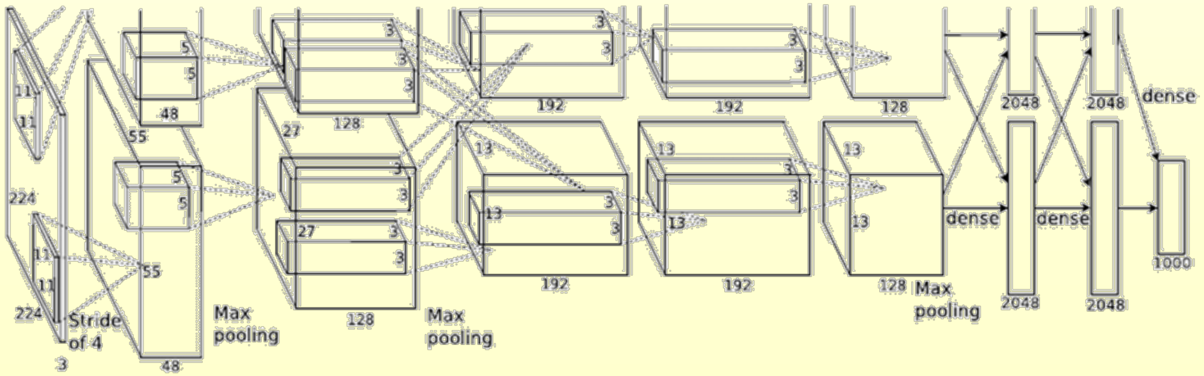


The Network is the Abstraction





Functional Maps and Deep Nets



Learning for 3D Data

Category

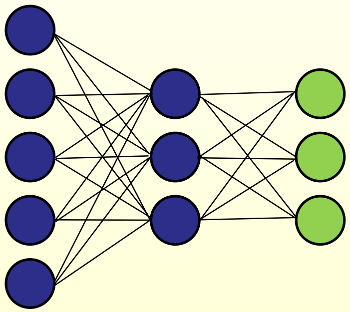
Material

Parts

Size

Functionality

Mass



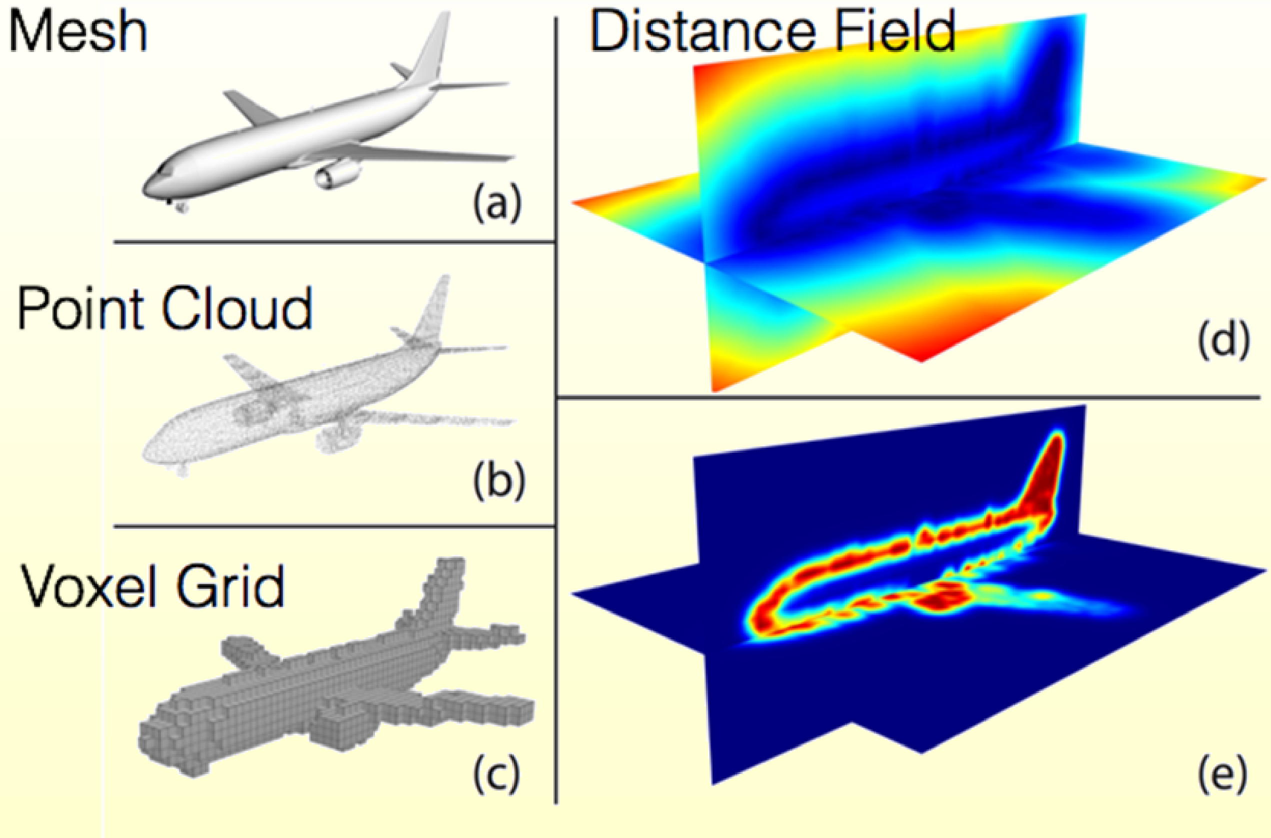
…

…

Build 3D knowledge base

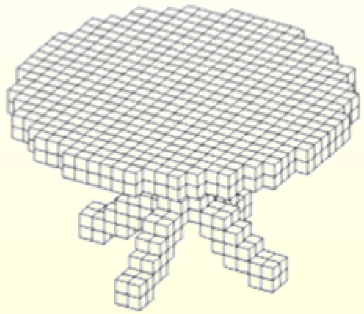
Design deep learning methods

An Issue of Representation



Extant Approaches

Volumetric



3DShapeNets by Z. Wu et al. CVPR 15

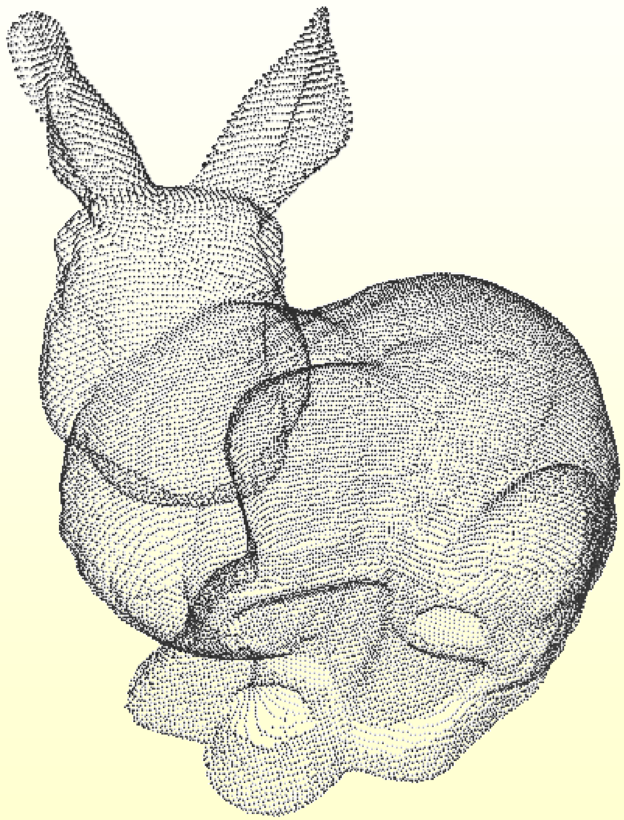
VoxNet by D. Maturana et al. IEEE/RSJ 15

Multi-View

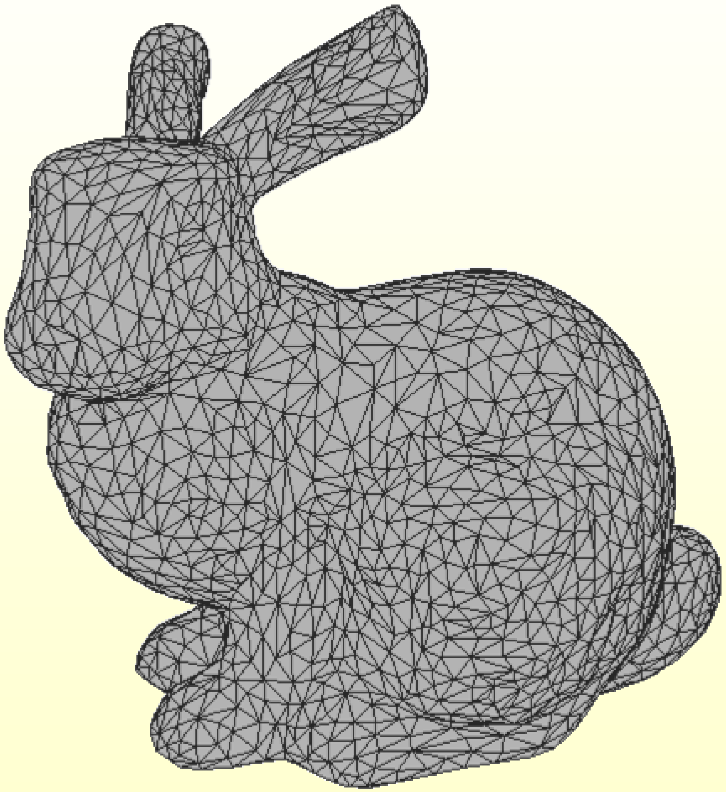
MVCNN by H. Su et al. ICCV 15

DeepPano by B. Shi et al. IEEE/SPL 15

Most Popular Representations



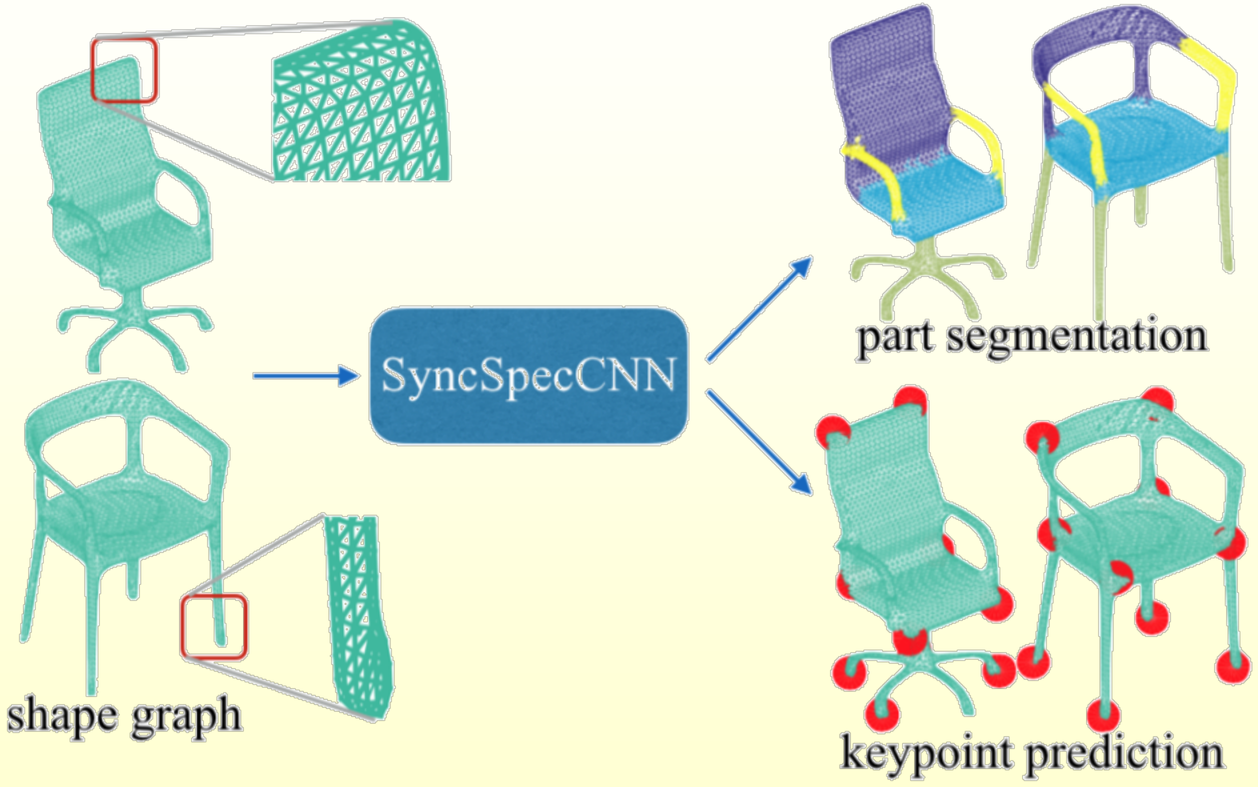
Point cloud



Mesh

Functional Maps in Graph-Based Deep Nets

Synchronized Spectral CNN

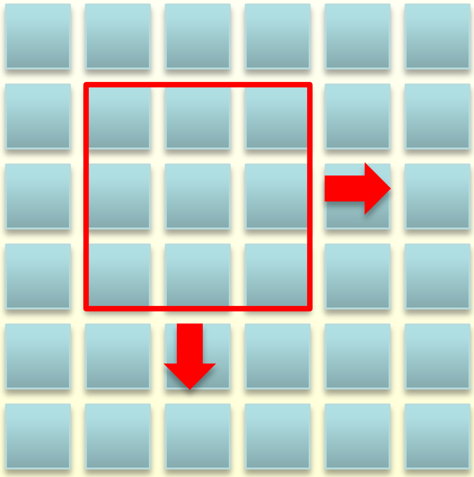


Input: shape graph
equipped with vertex functions

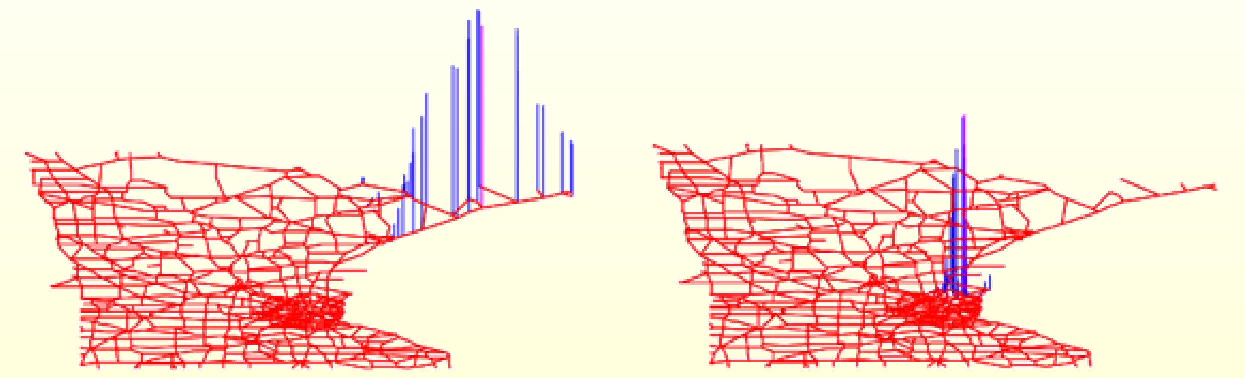
Output: semantic functions

[L, Yi, H. Su, LG, 2017]

The Difficulty of CNN Parameter Sharing on General Graphs



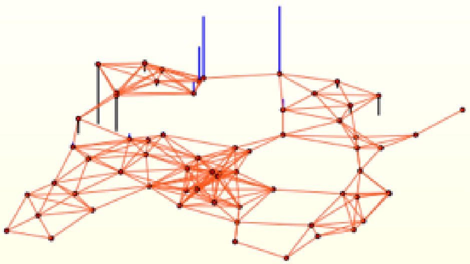
Grid



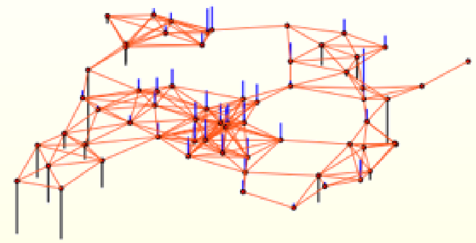
Shuman et al. 2013

General Graph

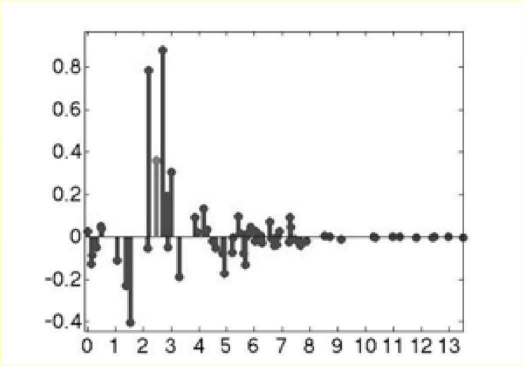
Generalized Convolution via Graph Fourier Transform



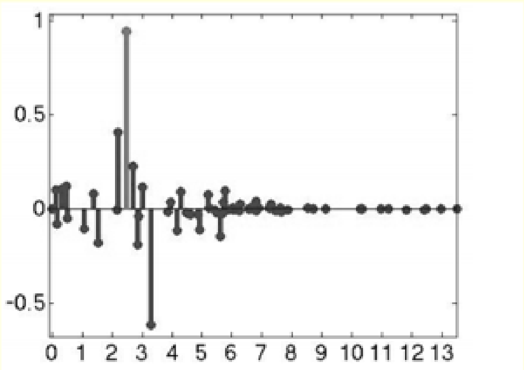
*



Convolution in the spatial domain



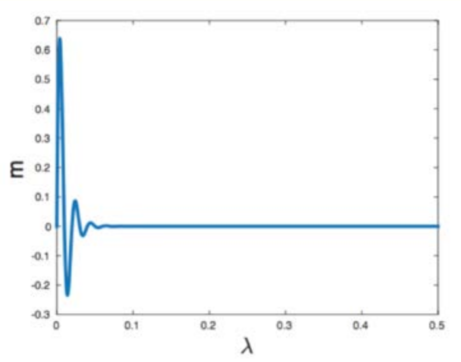
•



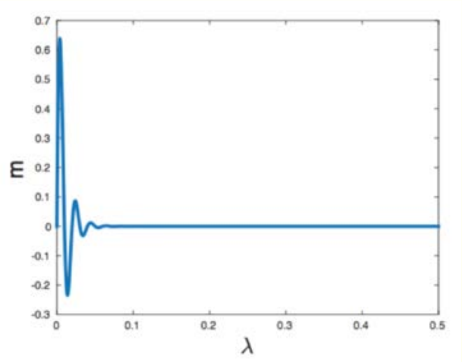
Multiplication in the spectral domain

Cross Domain Discrepancy

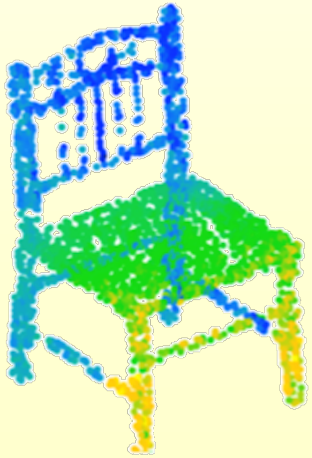
Spectral Domain 1



Spectral Domain 2

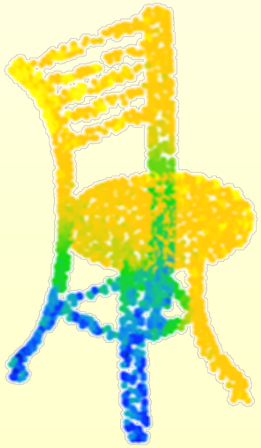


Spectral domains are independently defined for each shape graph



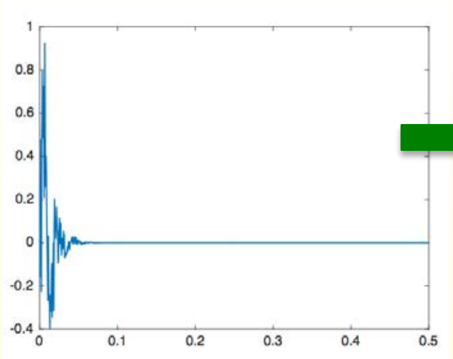
The same spectral function can induce very different spatial functions on different graphs

Cross domain parameter sharing is not valid

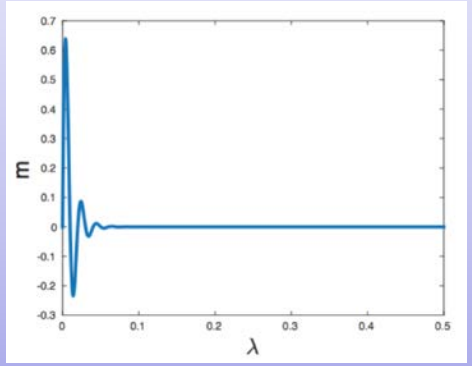


Different Domains Need to Be Synchronized

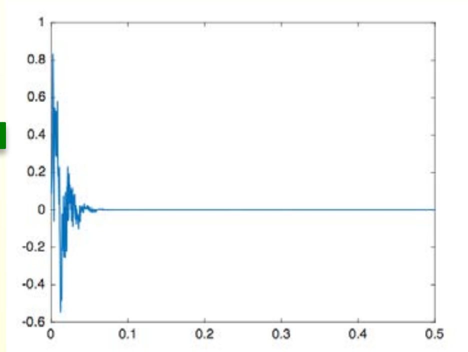
Spectral Domain 1

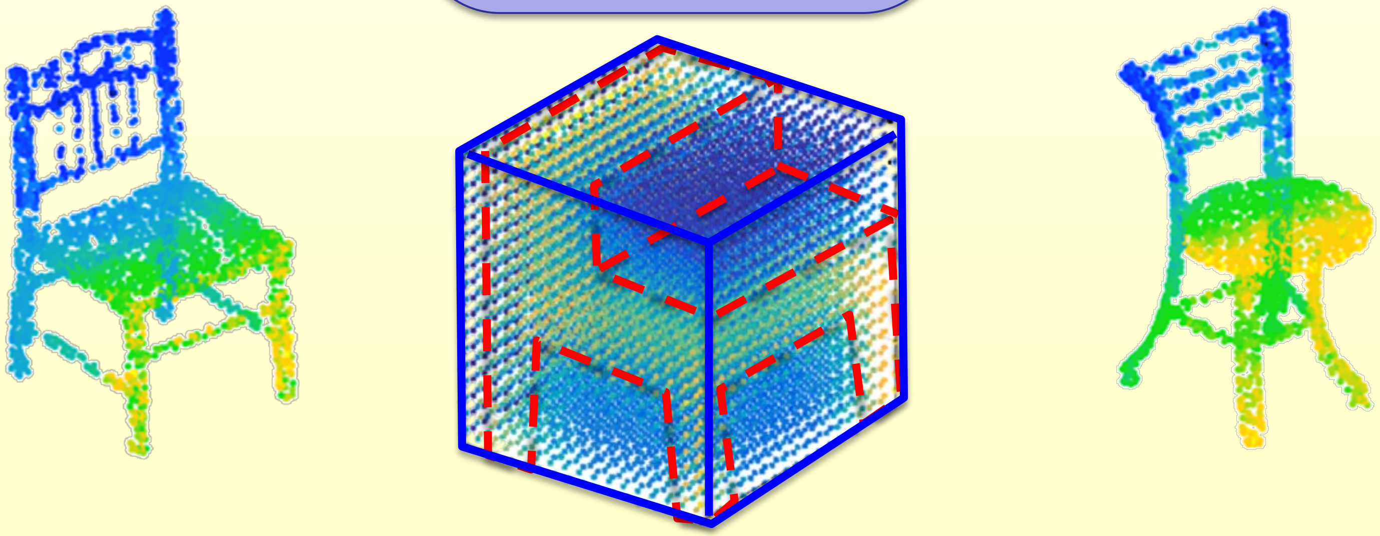


Canonical Domain

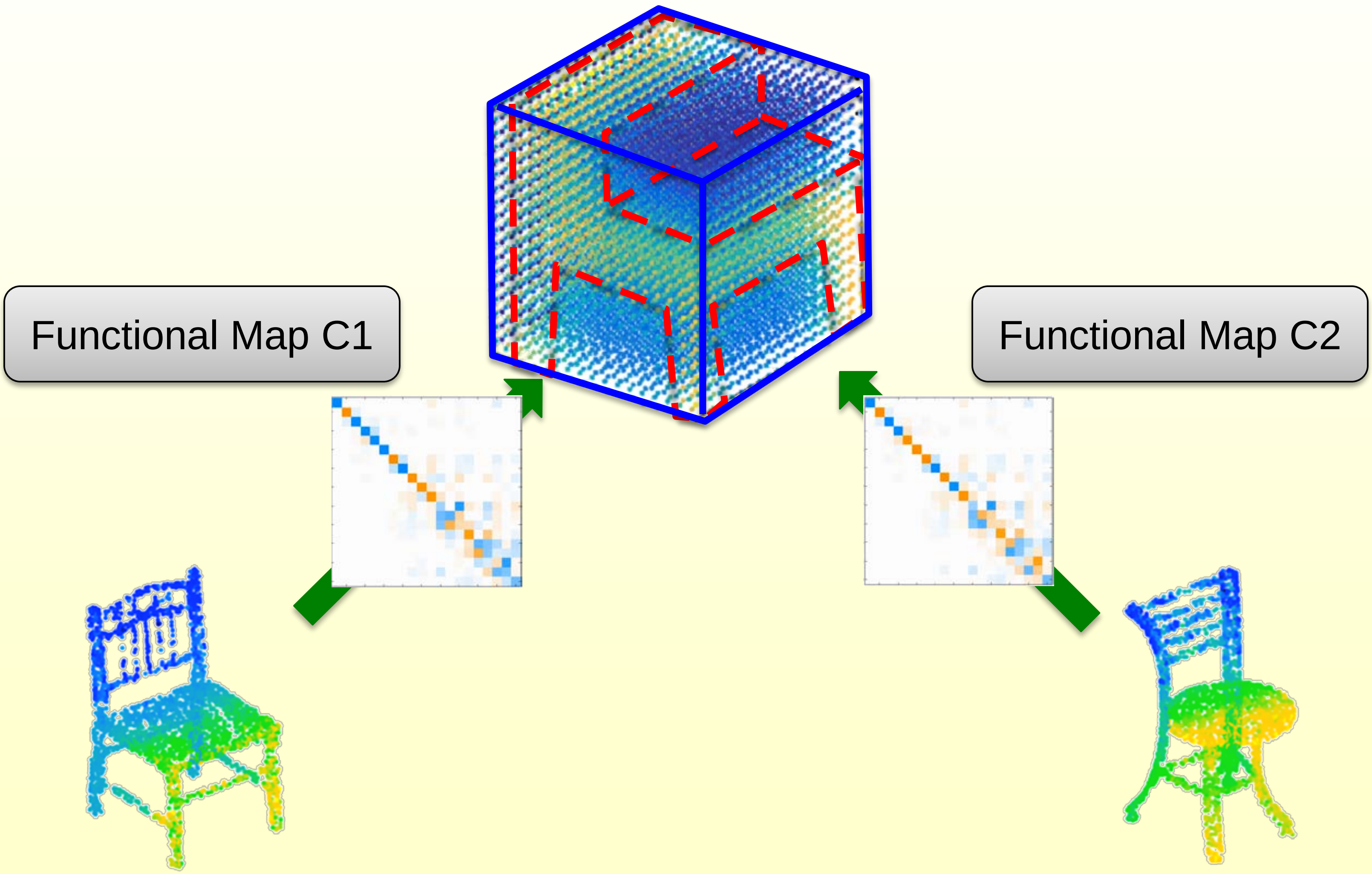


Spectral Domain 2

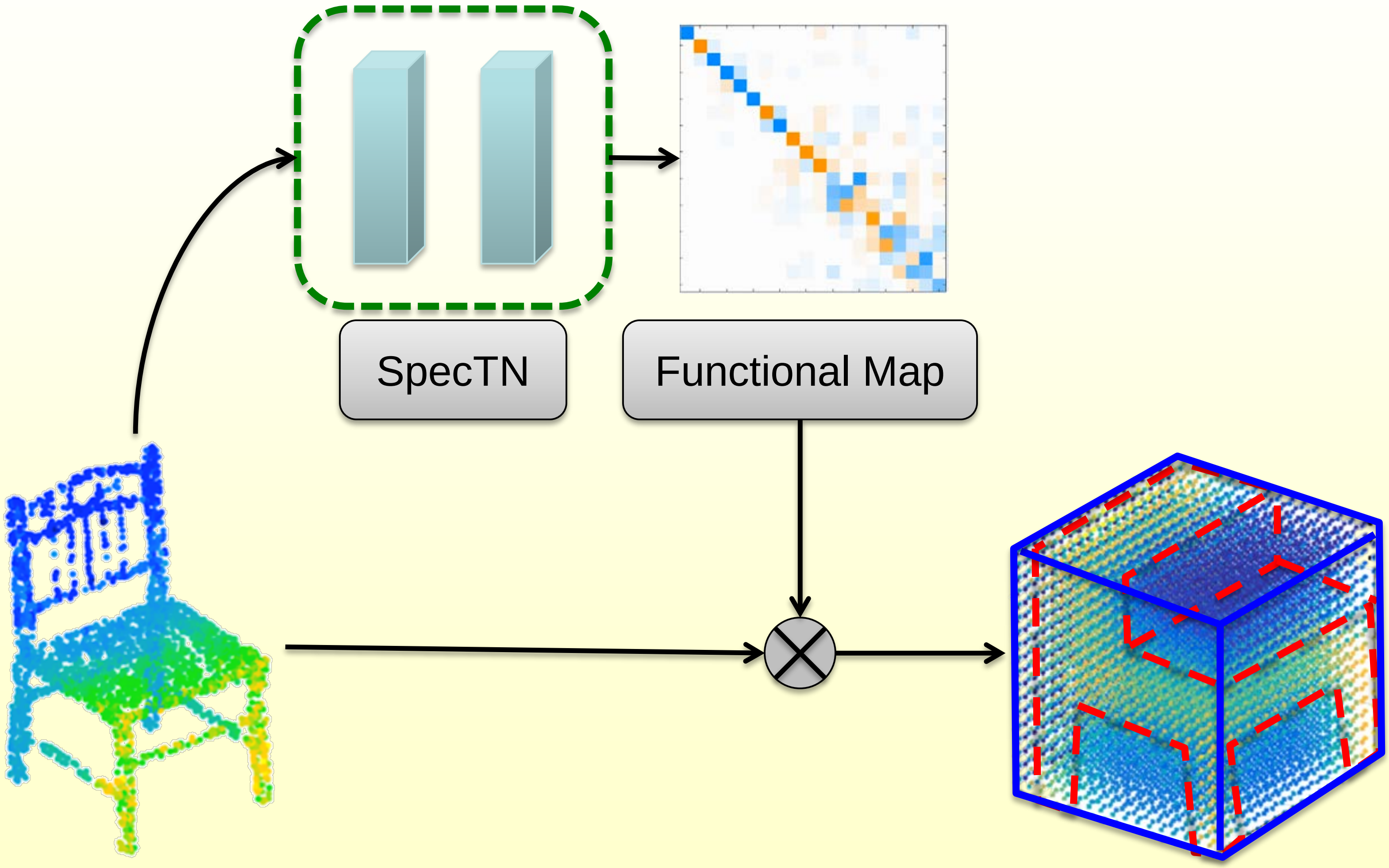




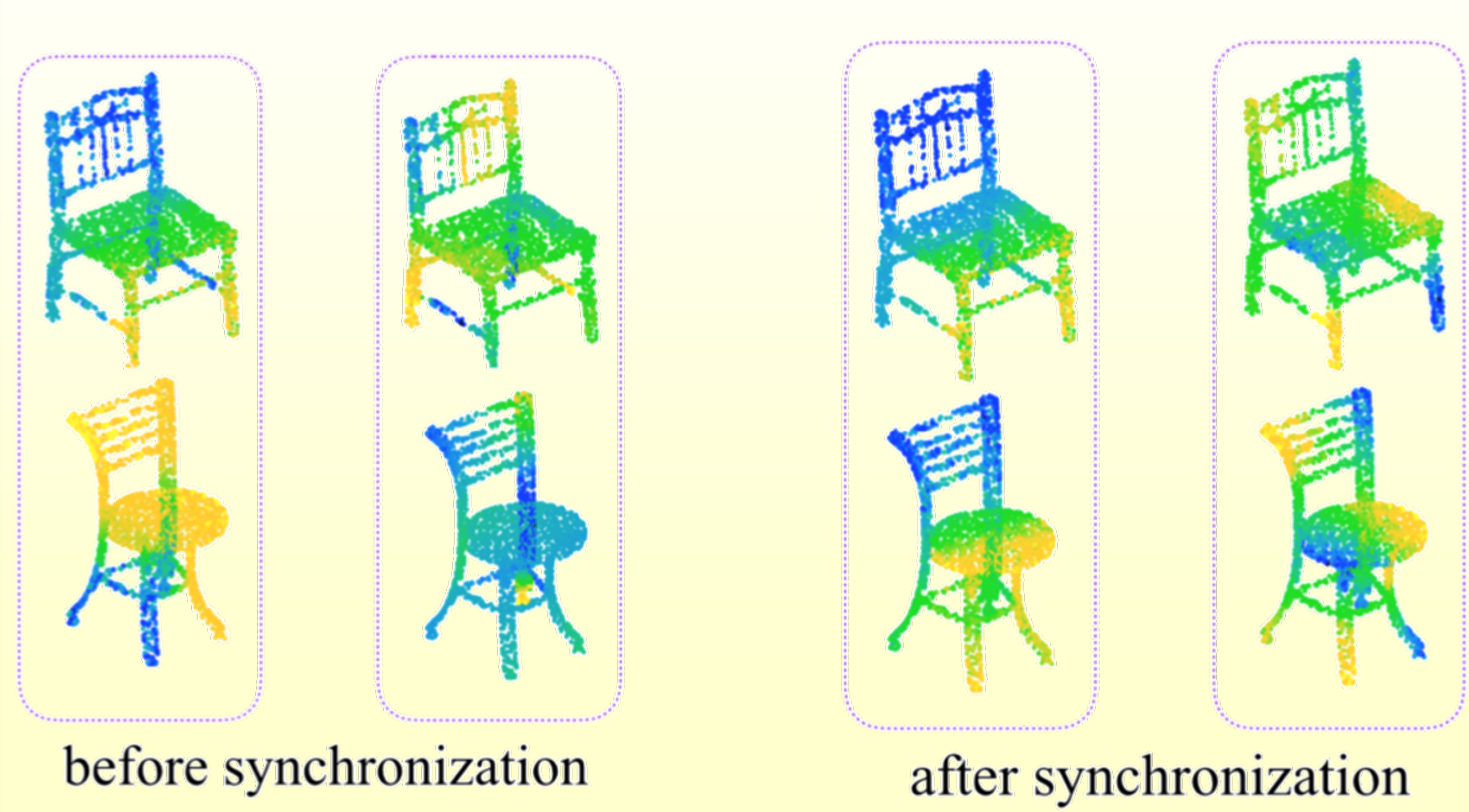
Functional Maps for Domain Synchronization



Spectral Transformer Network



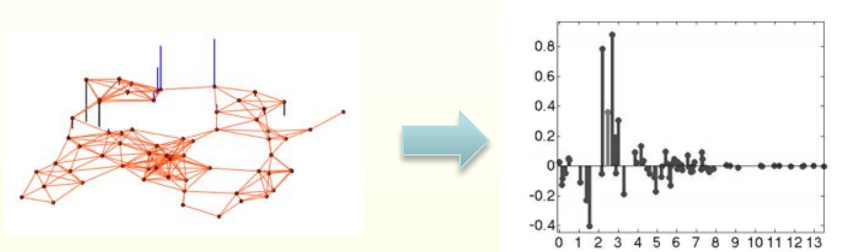
Synchronization Visualization



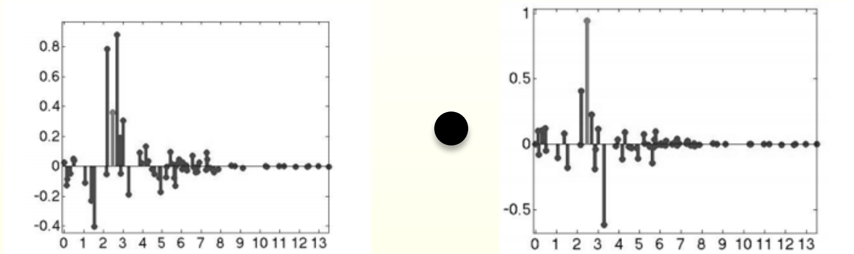
Key Ideas Summary

- Using spectral multiplication to replace spatial convolution, to allow parameter sharing at different locations on a shape.
- Using spectral transformer network to generate functional maps and synchronize different spectral domains, so as to allow parameter sharing across different shapes.

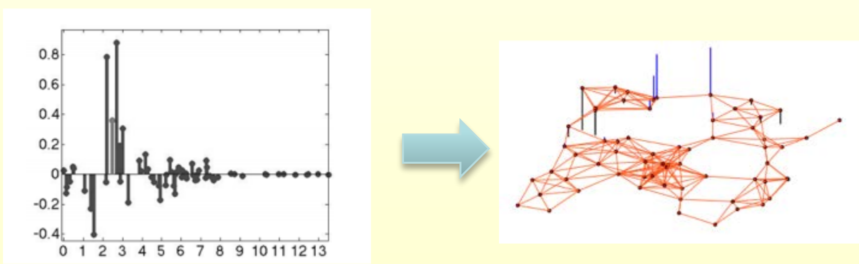
Basic Network Operations



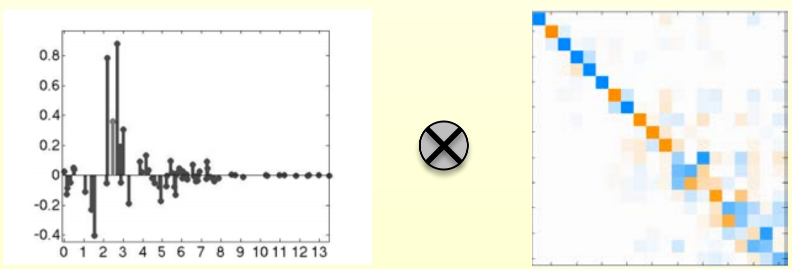
Forward Transform



Spectral Multiplication

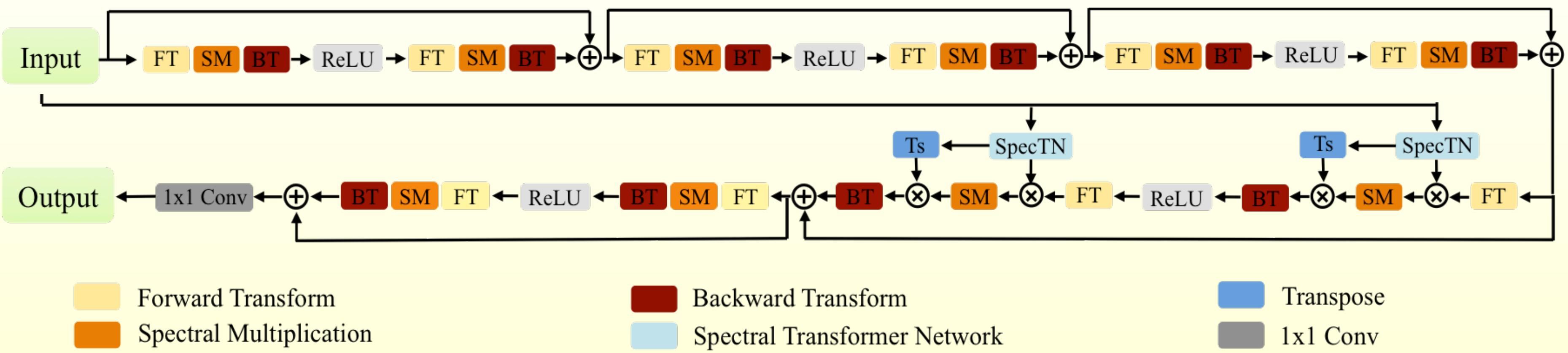


Backward Transform



Synchronization

Network Architecture

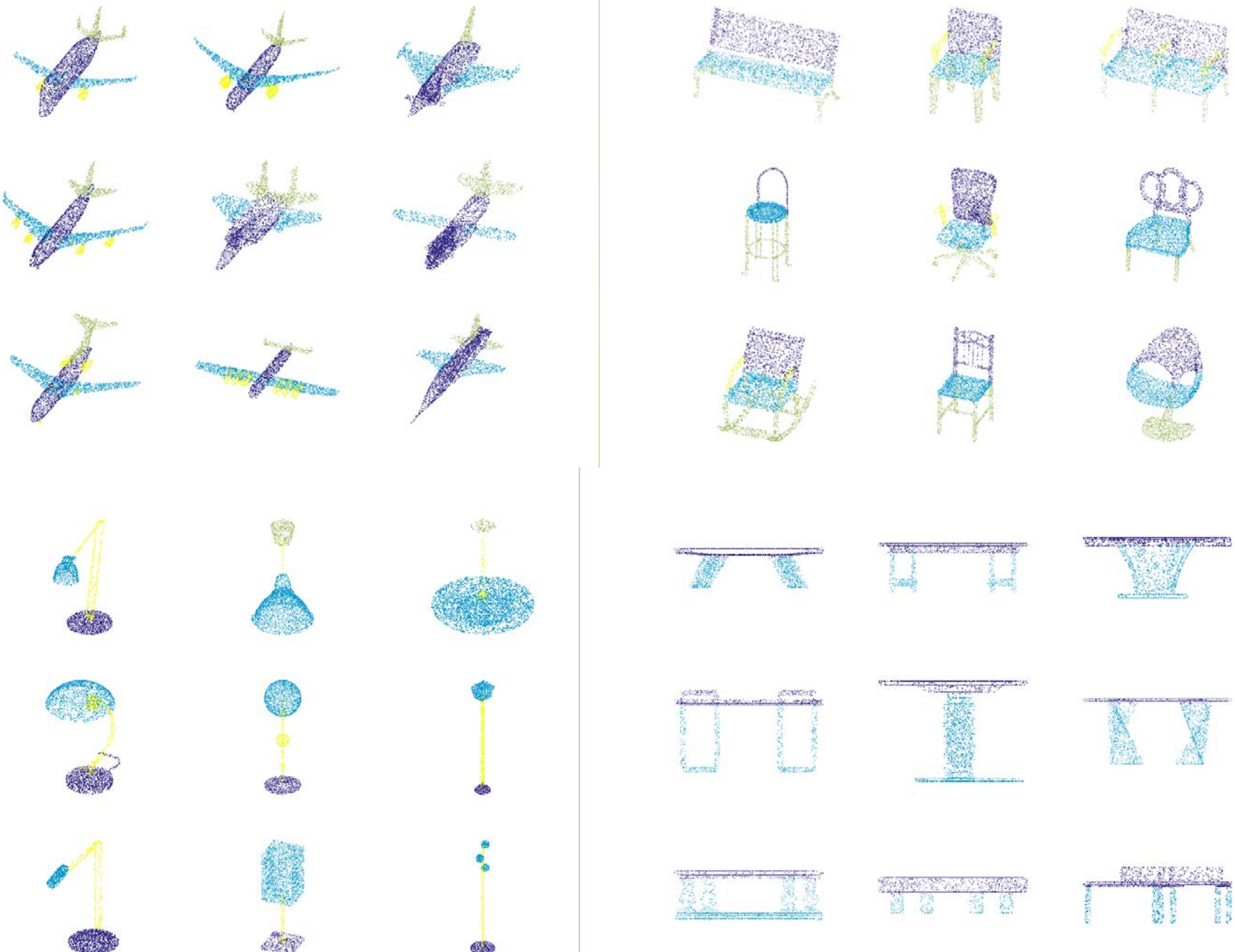


Application

Part Segmentation

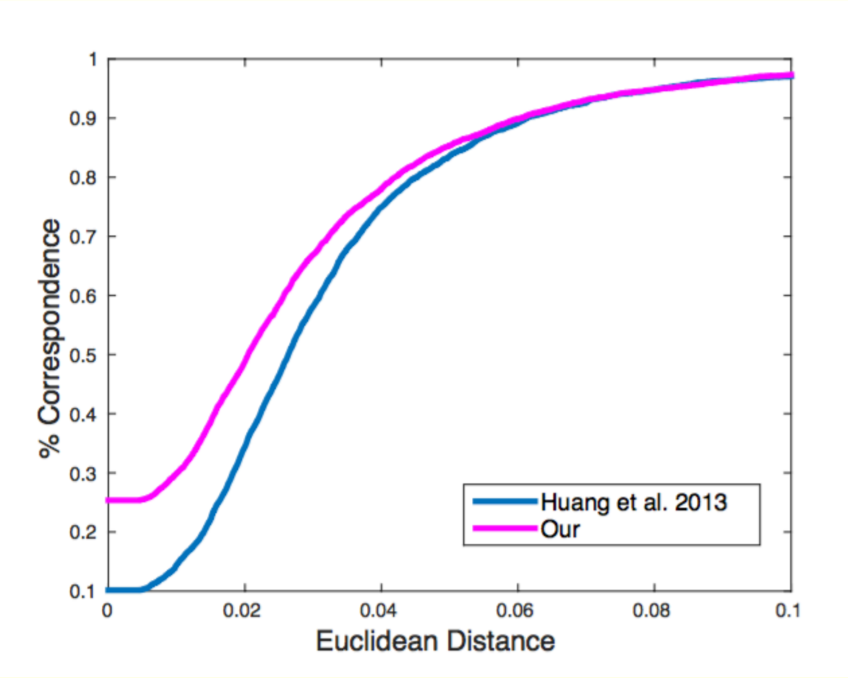
category	mean	plane	bag	cap	car	chair	ear-phone	guitar	knife	lamp	laptop	motor-bike	mug	pistol	rocket	skate-board	table
Wu14 [26]	-	63.20	-	-	-	73.47	-	-	-	74.42	-	-	-	-	-	-	74.76
Yi16 [28]	81.43	80.96	78.37	77.68	**75.67**	87.64	61.89	91.79	85.36	80.59	95.58	**70.59**	91.85	**85.94**	53.13	69.81	75.33
ACNN [2]	79.63	76.35	72.89	70.80	72.72	86.12	71.14	87.84	81.98	77.43	95.49	45.68	89.49	77.41	49.23	82.05	76.71
Voxel CNN	79.37	75.14	72.80	73.28	70.00	87.17	63.50	88.35	79.58	74.43	93.92	58.67	91.79	76.41	51.16	65.25	77.08
Ours1	83.48	80.61	81.62	76.92	73.86	88.65	74.48	89.03	85.34	83.47	95.53	62.74	92.01	80.88	**62.10**	82.23	81.36
Ours2	**84.74**	**81.55**	**81.74**	**81.94**	75.16	**90.24**	**74.88**	**92.97**	**86.10**	**84.65**	**95.61**	66.66	**92.73**	81.61	60.61	**82.86**	**82.13**

IoU for part segmentation on 16 categories. Ours1 represents a variation of our framework without SpecTN and Ours2 corresponds to our full pipeline with SpecTN.



Application: Keypoint Prediction

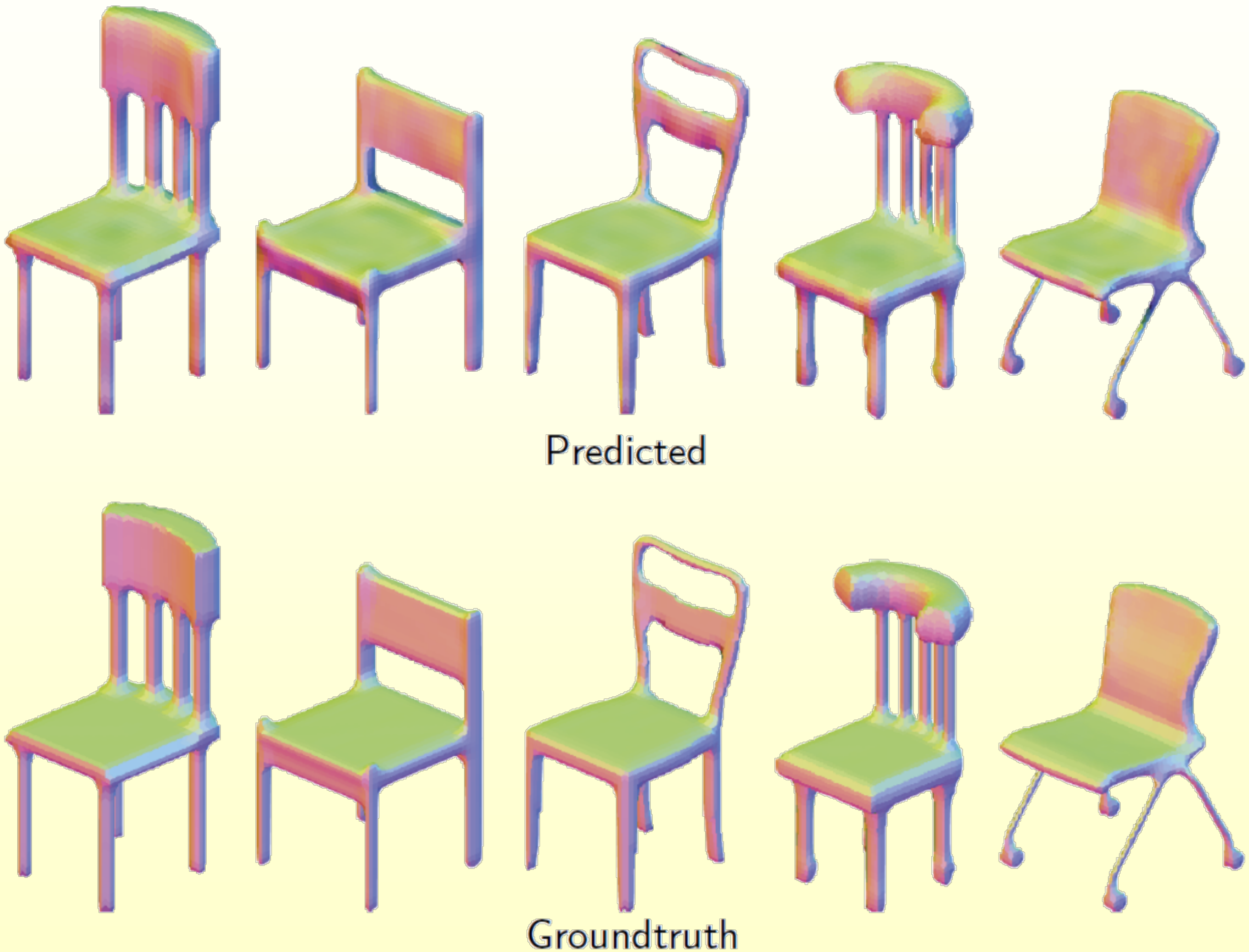
Keypoint Prediction



Comparison with previous states via PCK curve

Prediction Visualization

Application: Normal Prediction

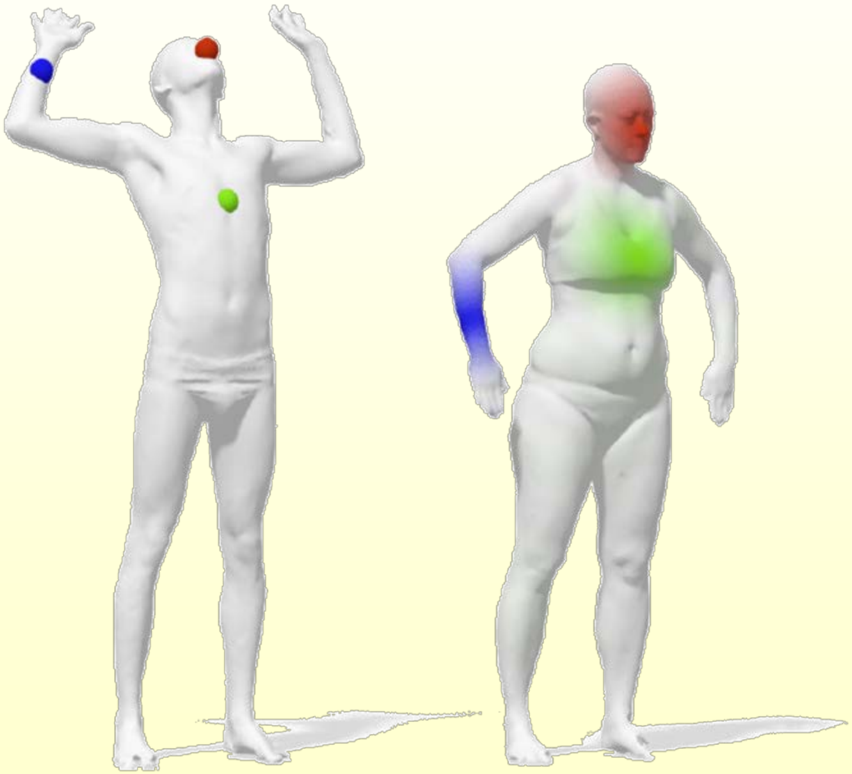


Deep Nets in the Computation of Functional Maps

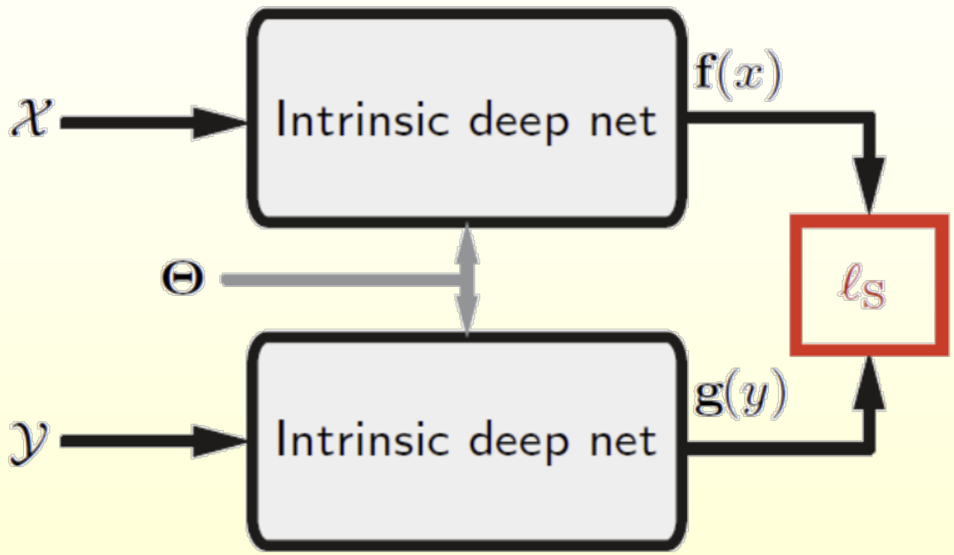
[O. Litany, T. Remez, E. Rodolà , A. M. Bronstein, M. M. Bronstein, 2017]

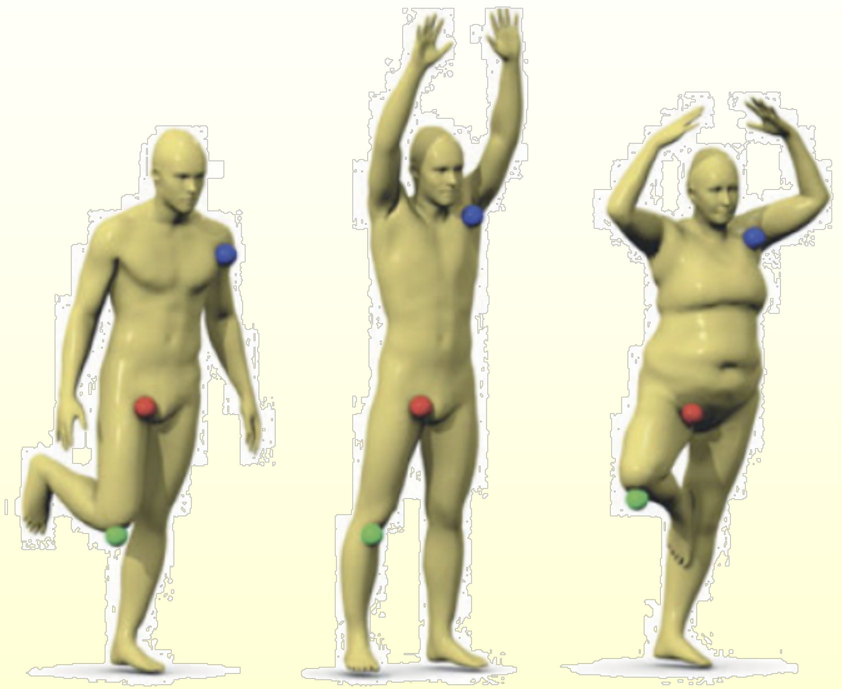
Improving Functional Maps with Deep Nets

- Goal: improve the descriptors used during the functional map computation to make the resulting map closer to point-to-point
- In ML, common to view a correspondence problem as a labeling problem
- Fmaps can be thought of as soft correspondences



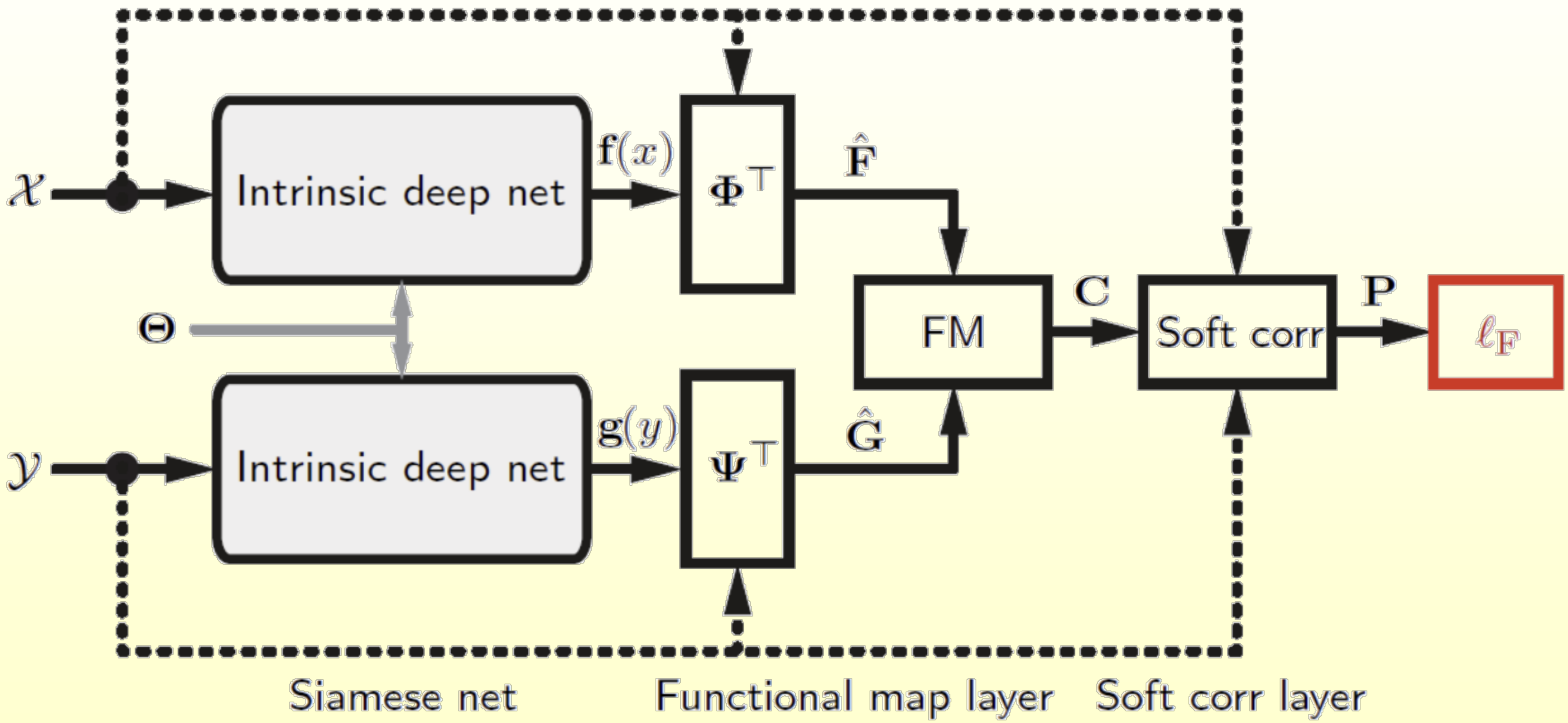
Siamese Metric Learning



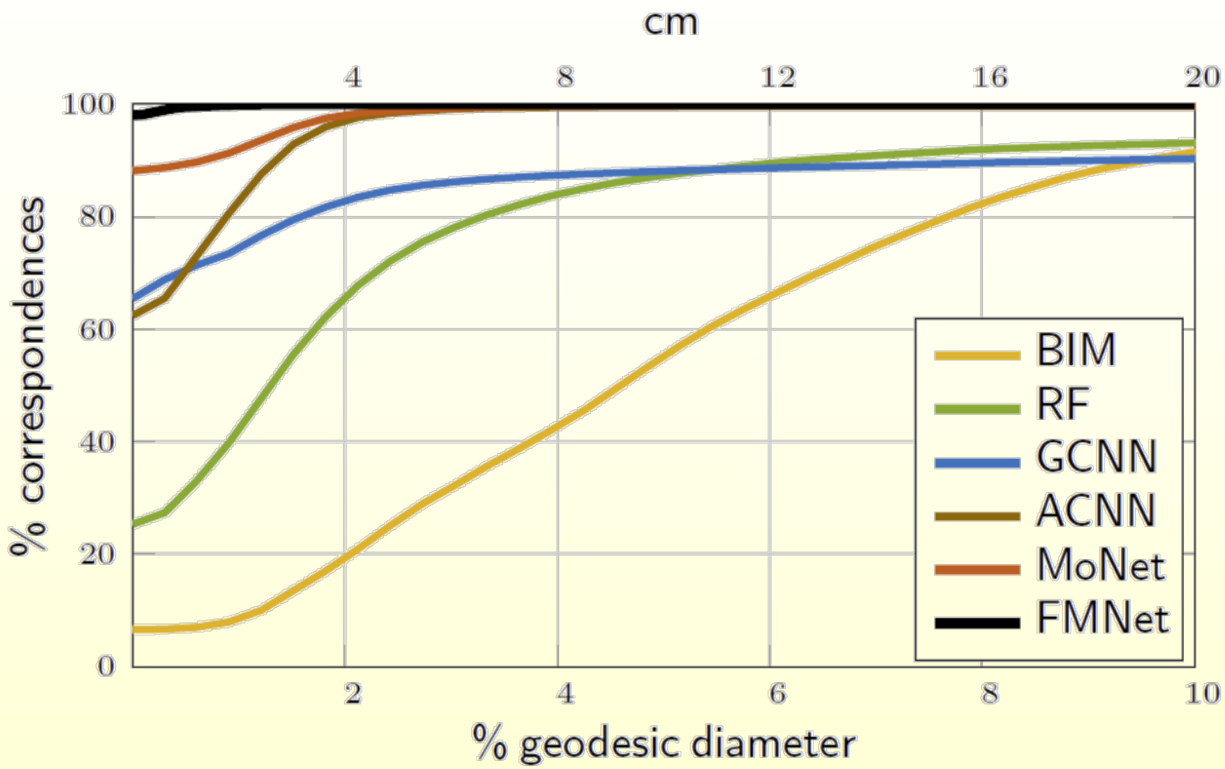


Poitwise feature cost
$$\ell_S(\boldsymbol{\Theta}) = \gamma \sum_{x,x^+} \|\mathbf{f}_{\boldsymbol{\Theta}}(x) - \mathbf{f}_{\boldsymbol{\Theta}}(x^+)\|_2^2 + (1-\gamma) \sum_{x,x^-} [\mu - \|\mathbf{f}_{\boldsymbol{\Theta}}(x) - \mathbf{f}_{\boldsymbol{\Theta}}(x^-)\|_2^2]_+$$

FMNet: Structured Correspondences with FMaps



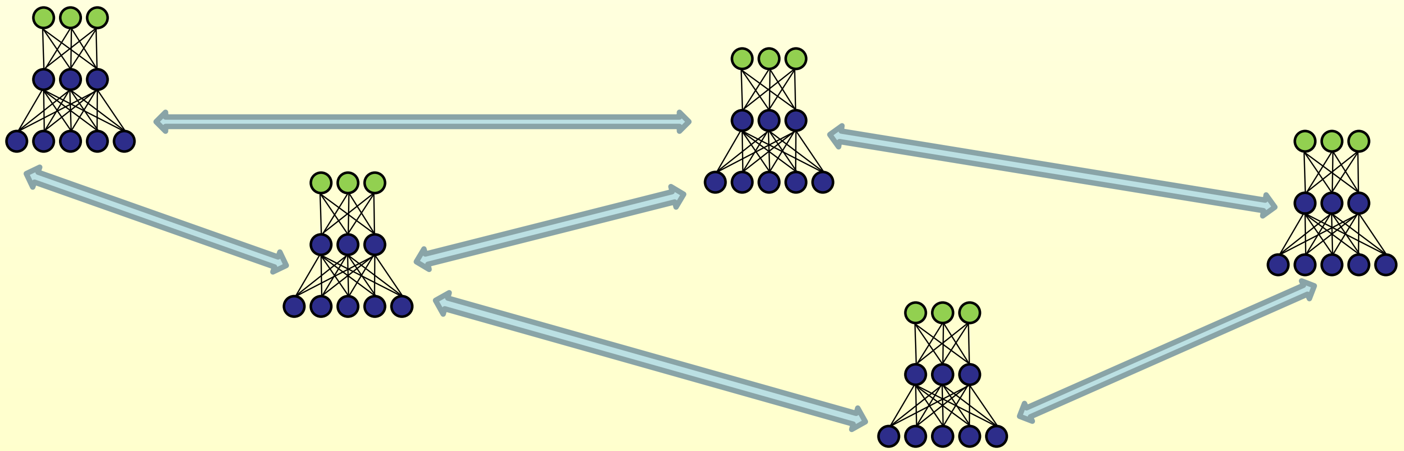
Correspondence Quality



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

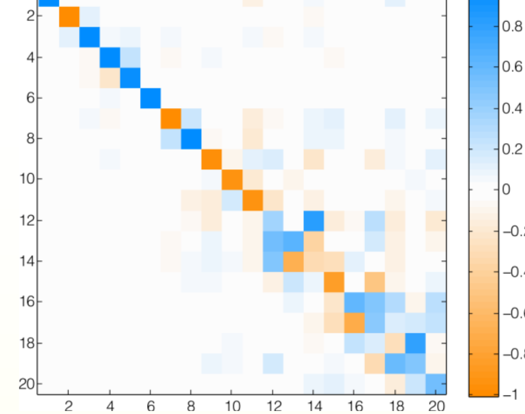
Methods: Kim et al. 2011 (BIM); Rodolà et al. 2014 (RF); Boscaini et al. 2015 (ADD); Masci et al. 2015 (GCNN); Boscaini et al. 2016 (ACNN); Monti et al. 2016 (MoNet); Litany et al. 2017 (FMNet); data: Bogo et al. 2014 (FAUST); benchmark: Kim et al. 2011

Horizontal and Vertical Networks



Open Problems on Functional Maps

Open Problems on Functional Maps, I



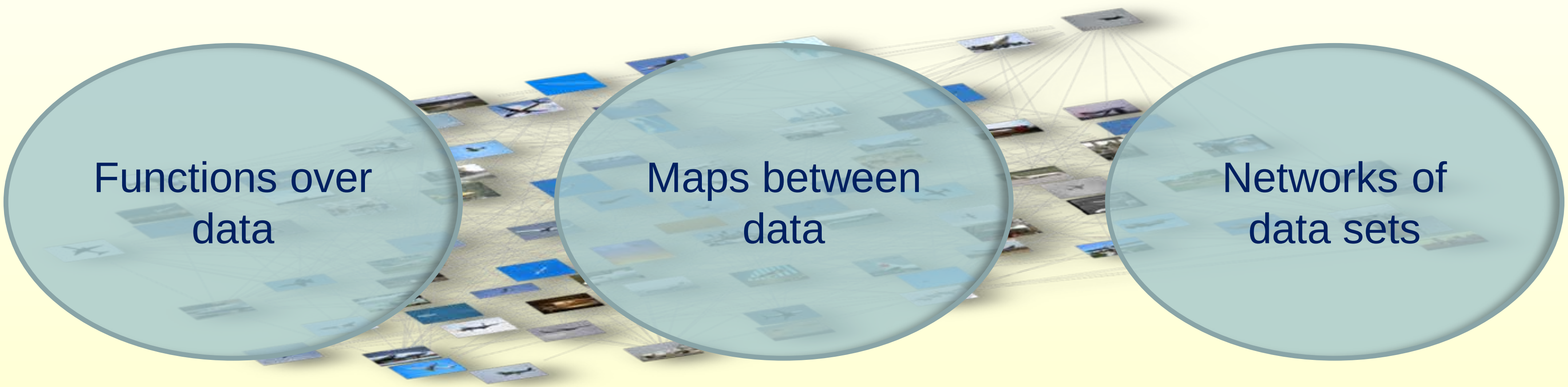
- Basis selection
  - Wavelets?
  - Replace regularizers by exact commutation?
- Probe function selection and learning
  - Transport quality
  - Different maps for different functional subspaces?
- Subclasses of functional maps
  - Point-to-point: how close? new projection algorithms?
  - Positive functional maps (soft maps)
- Statistics on functional maps
  - Encoding map distributions
  - Maplets?

Open Problems on Functional Maps, II

- Network formation
- Map processing and reconstruction
- Latent spaces
  - Stability
  - Hierarchical and non-hierarchical structures
  - Map-based clustering
- Use of supervision / learning
- Functional maps and deep nets
  - Moving both vertically and horizontally
  - Nets that learn algebraic structure: “homomorphic descriptors”
  - Parametrizing the space of learning objectives

Functoriality

- Classical "vertical" view of data analysis:
  - Signals to symbols
    - from features, to parts, to semantics …



- A new "horizontal" view based on peer-to-peer signal relationships
  - so that semantics emerges from the network

Thank You