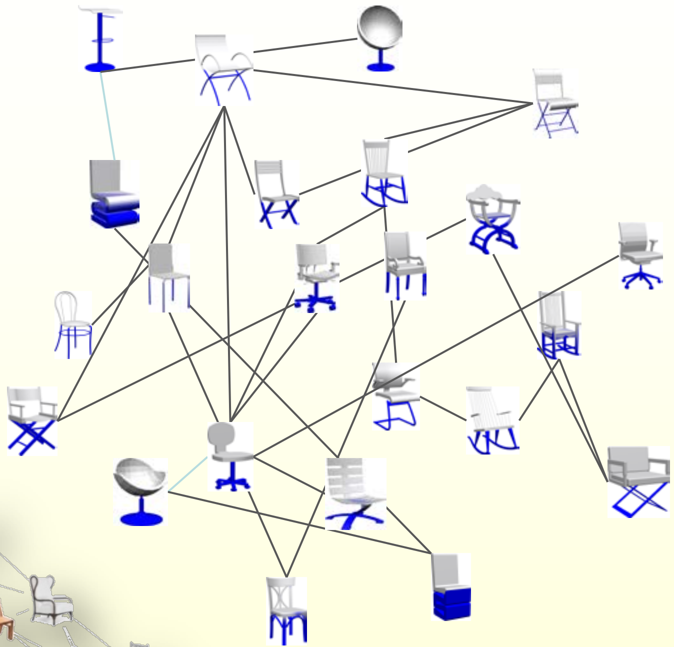
Information Transport in 2D and 3D Between Visual Media

Leonidas Guibas
Stanford University

October 2014
BAVM

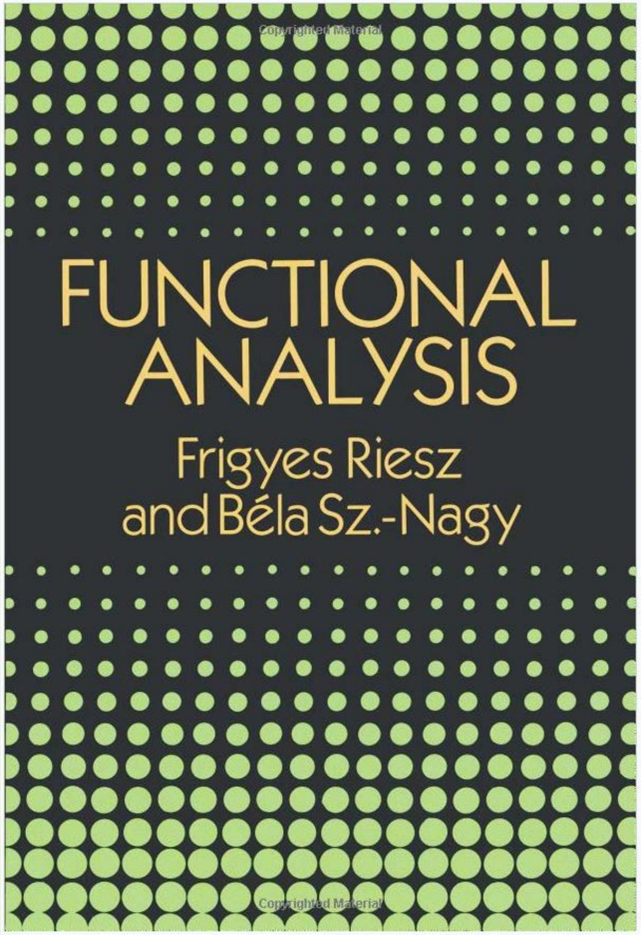
Image Networks

But also of Shapes, Mixed, Etc.

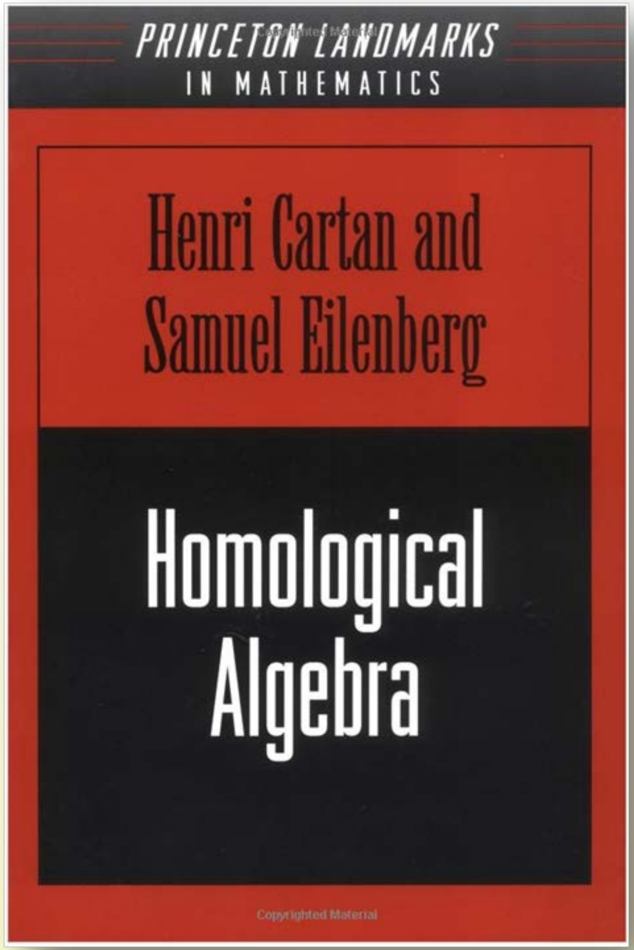


Relations Between Visual Data

Linear Operators

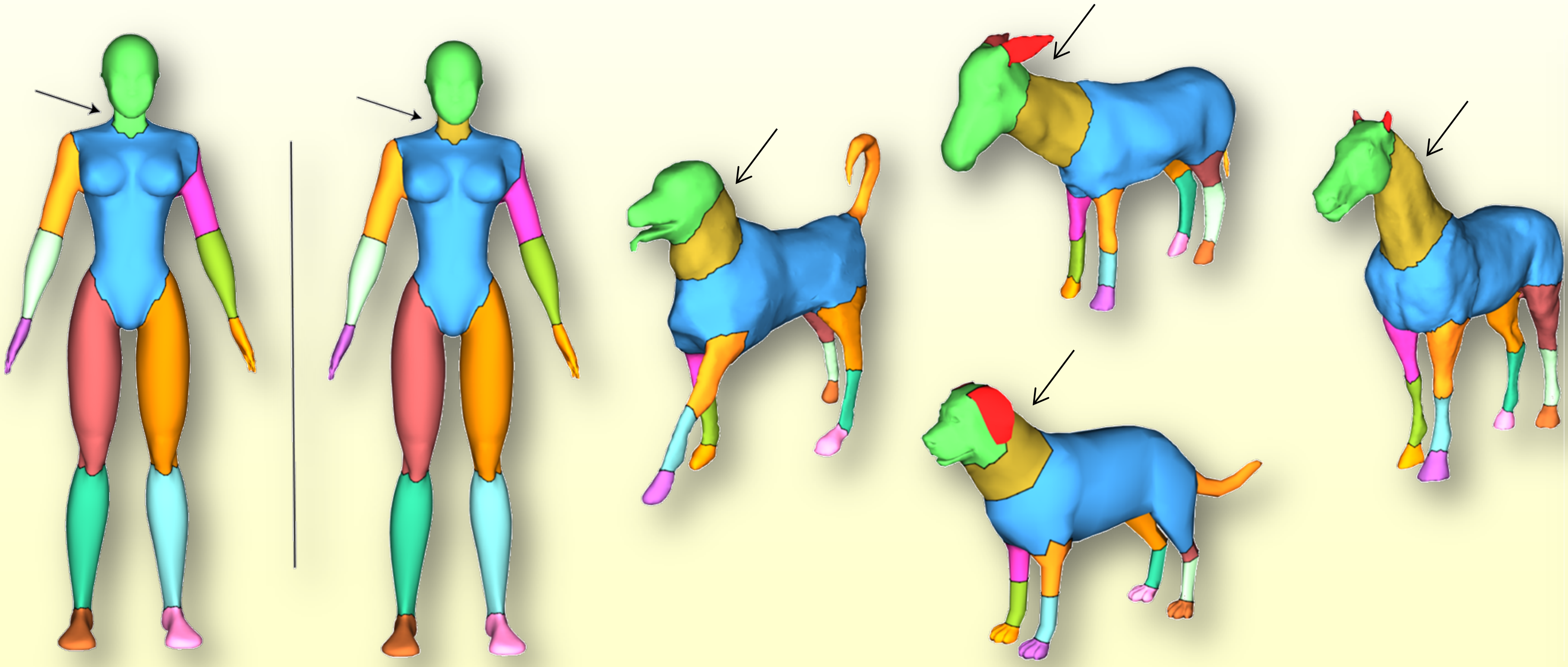


Functors, Categories, Colimits



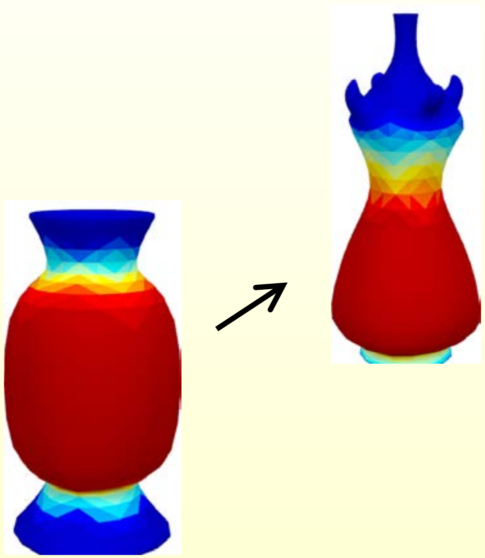
Each Data Set Is Not Alone

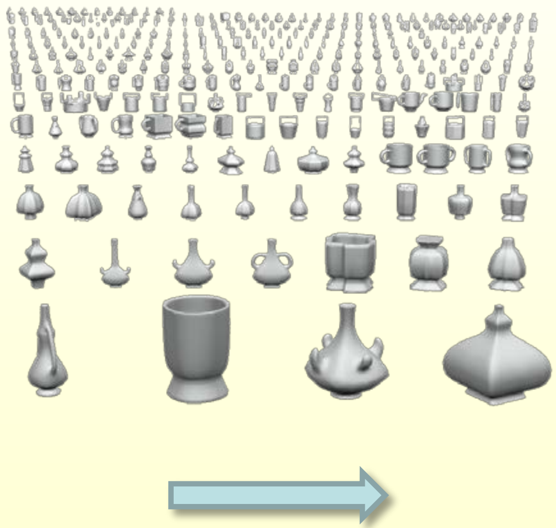
- The interpretation of a particular piece of geometric data is deeply influenced by our interpretation of other related data

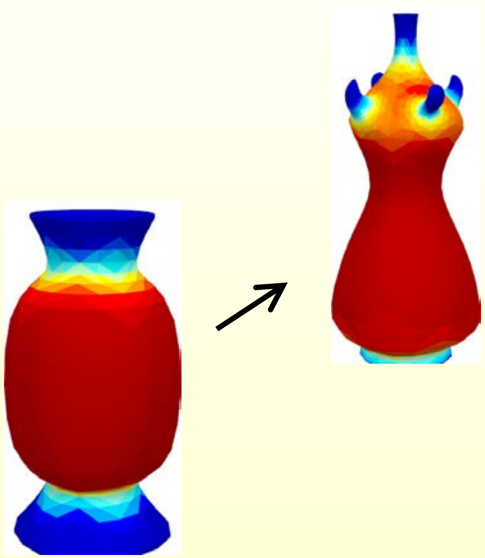


3D Segmentation

And Each Data Set Relation is Not Alone







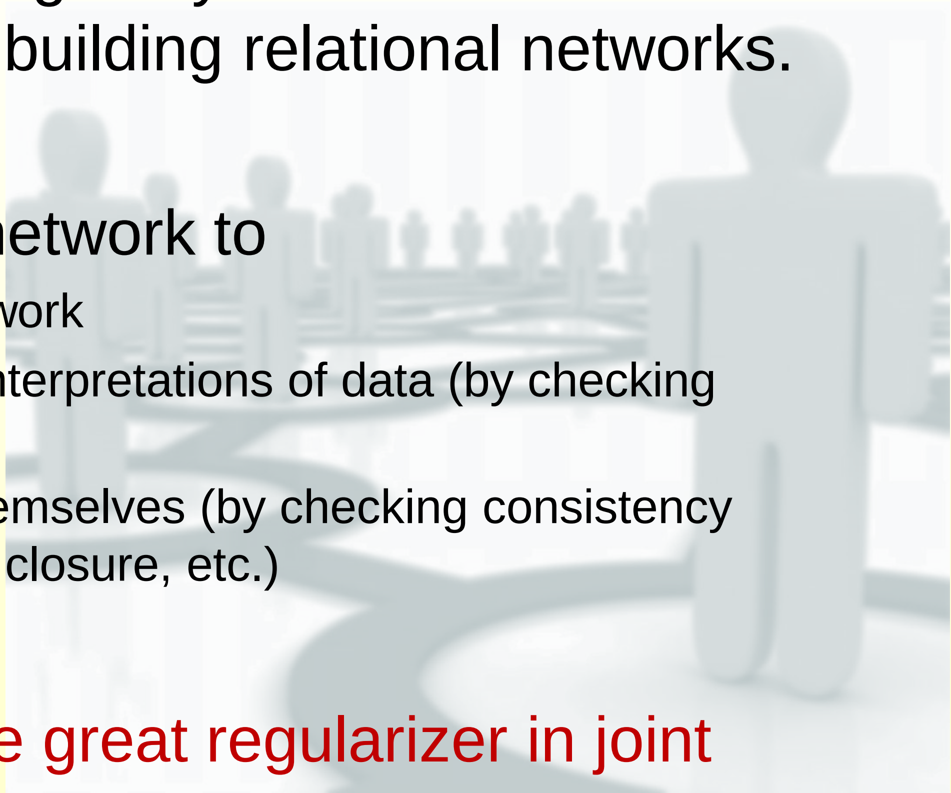
State of the art algorithm applied to the two vases

Map re-estimated using advice from the collection

3D Mapping

Societies, or
Social Networks of Data Sets

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

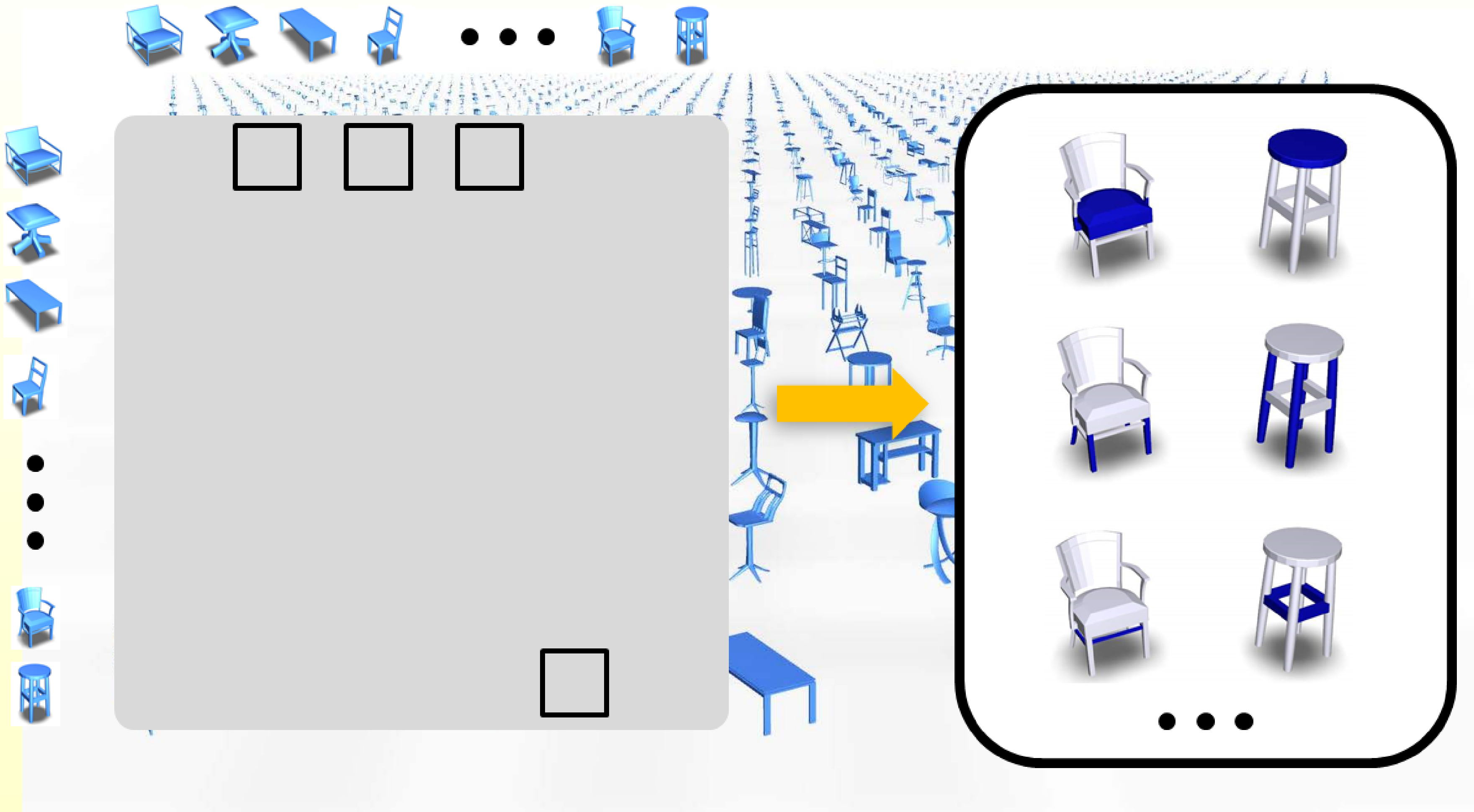


We can exploit the relational network to

- 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.)

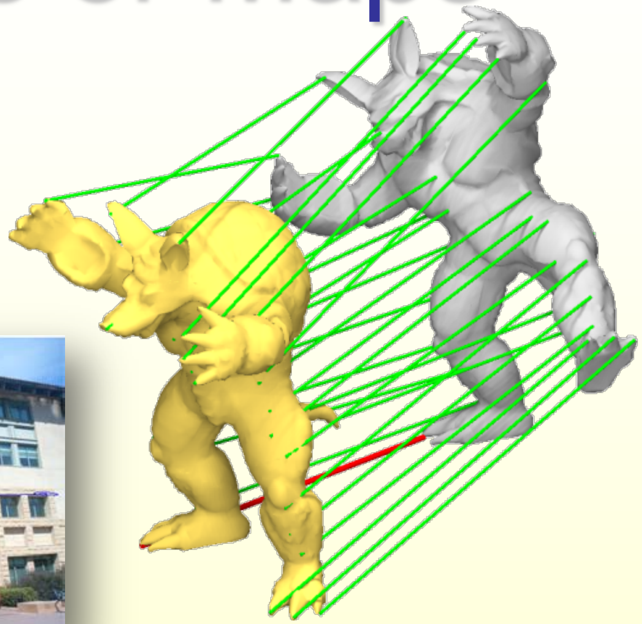
Thus the network becomes the great regularizer in joint data analysis.

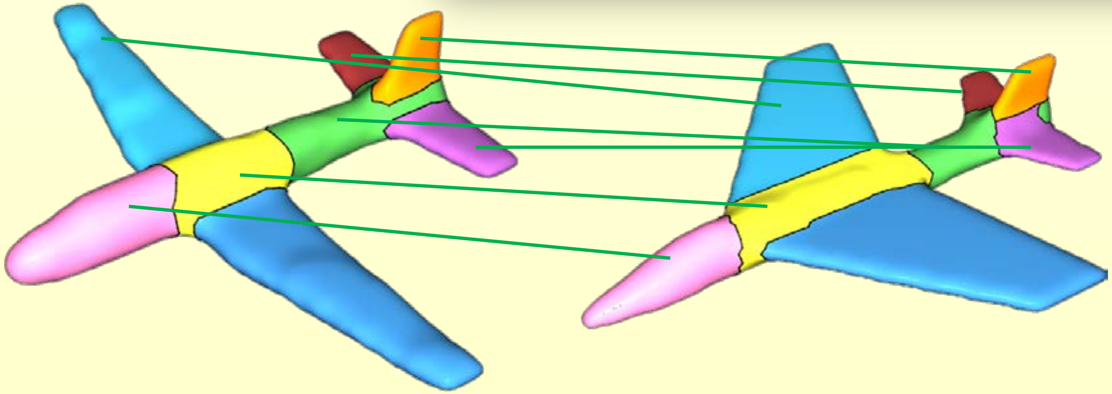
Semantic Structure Emerges from the Network



Key: Relationships as Collections of Correspondences or Maps

- Multiscale mappings
  - Point/pixel level
  - part level



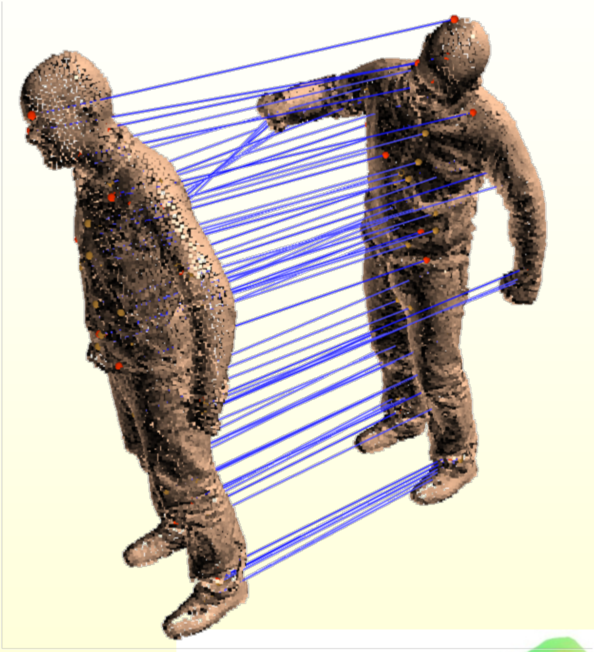


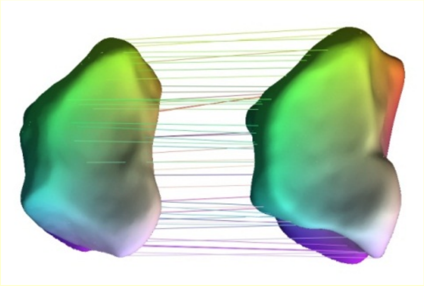
Maps capture what is the same or similar across two data sets

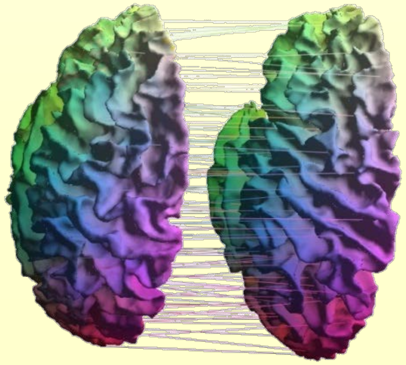
Relationships as First-Class Citizens

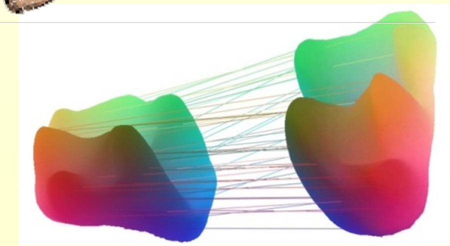
- How can we make data set relationships concrete, tangible, storable, searchable objects?

- How can we understand the "relationships among the relationships" or maps?



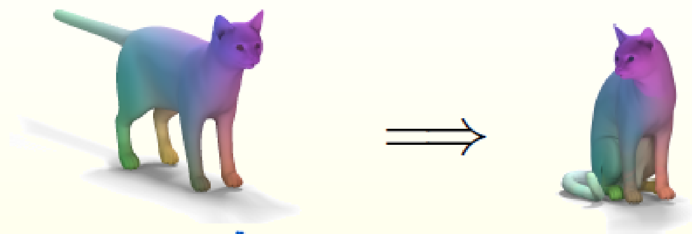




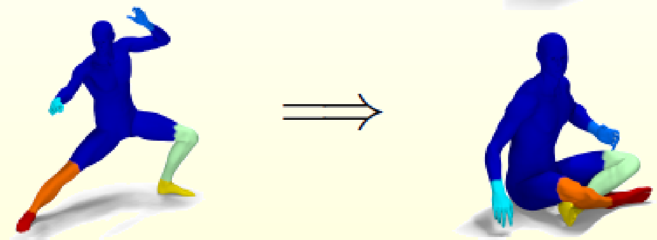


Good Correspondences or Maps are Information Transporters

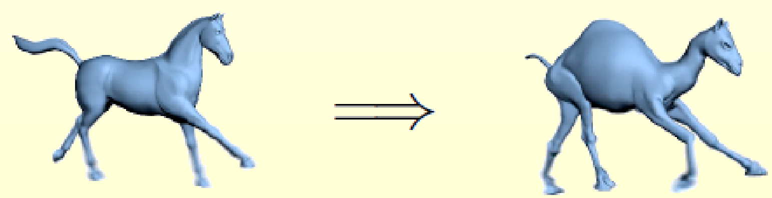
texture and parametrization

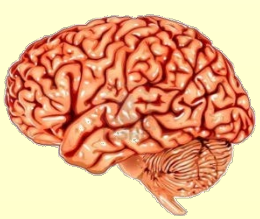


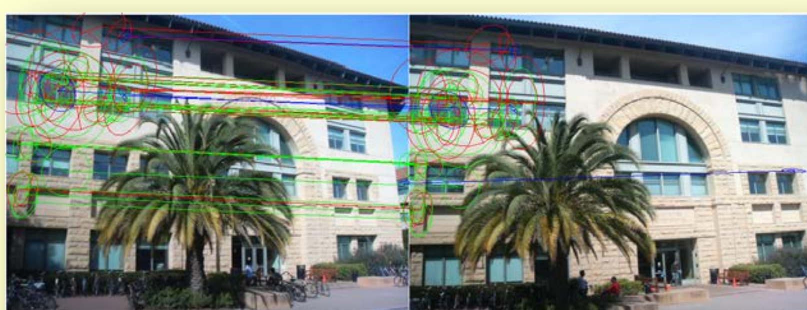
segmentation and labels



deformation



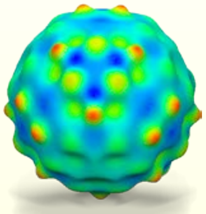




A Dual View: Functions and Operators

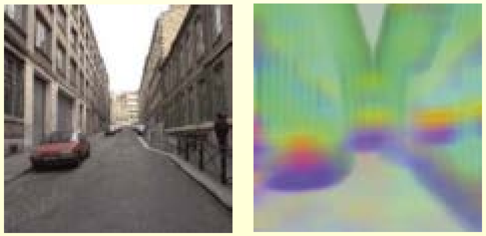
- Functions on data
  - Properties, attributes, descriptors, part indicators, etc.
  - But also opinions, beliefs, etc

- Operators on functions
  - Maps of functions to functions

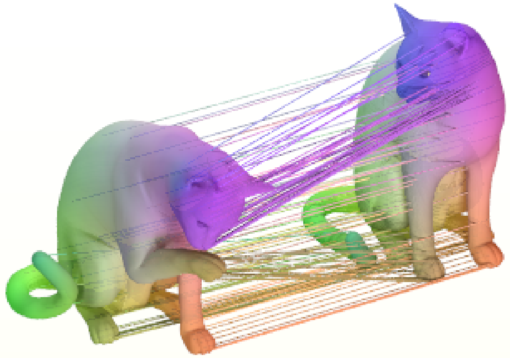


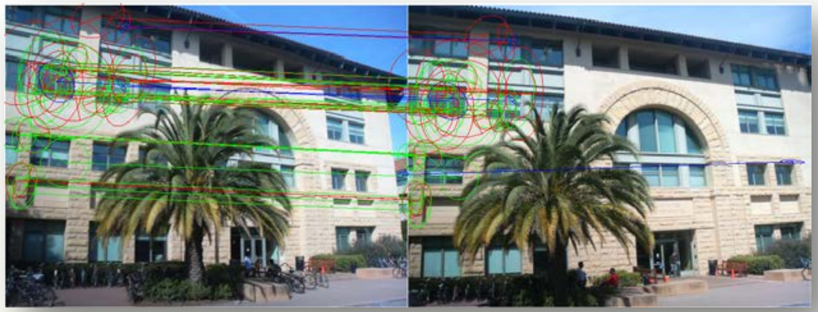
Curvature

Parts



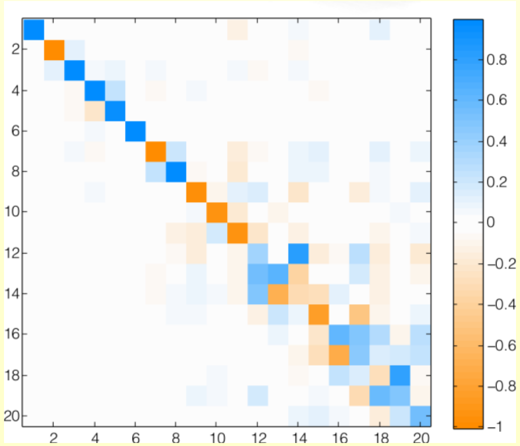
SIFT flow, C. Liu 2011



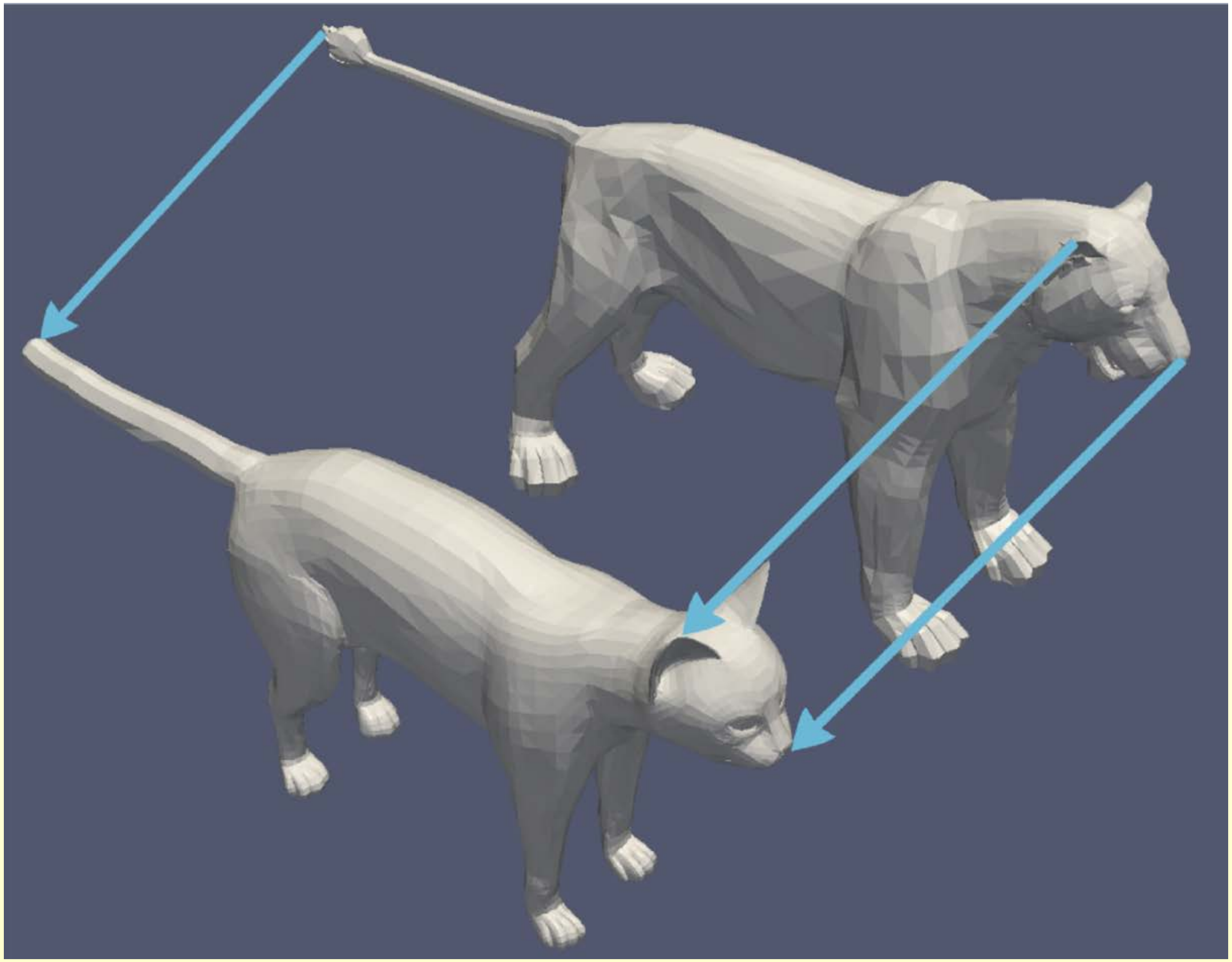


Functional Maps
(a.k.a. Operators)

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

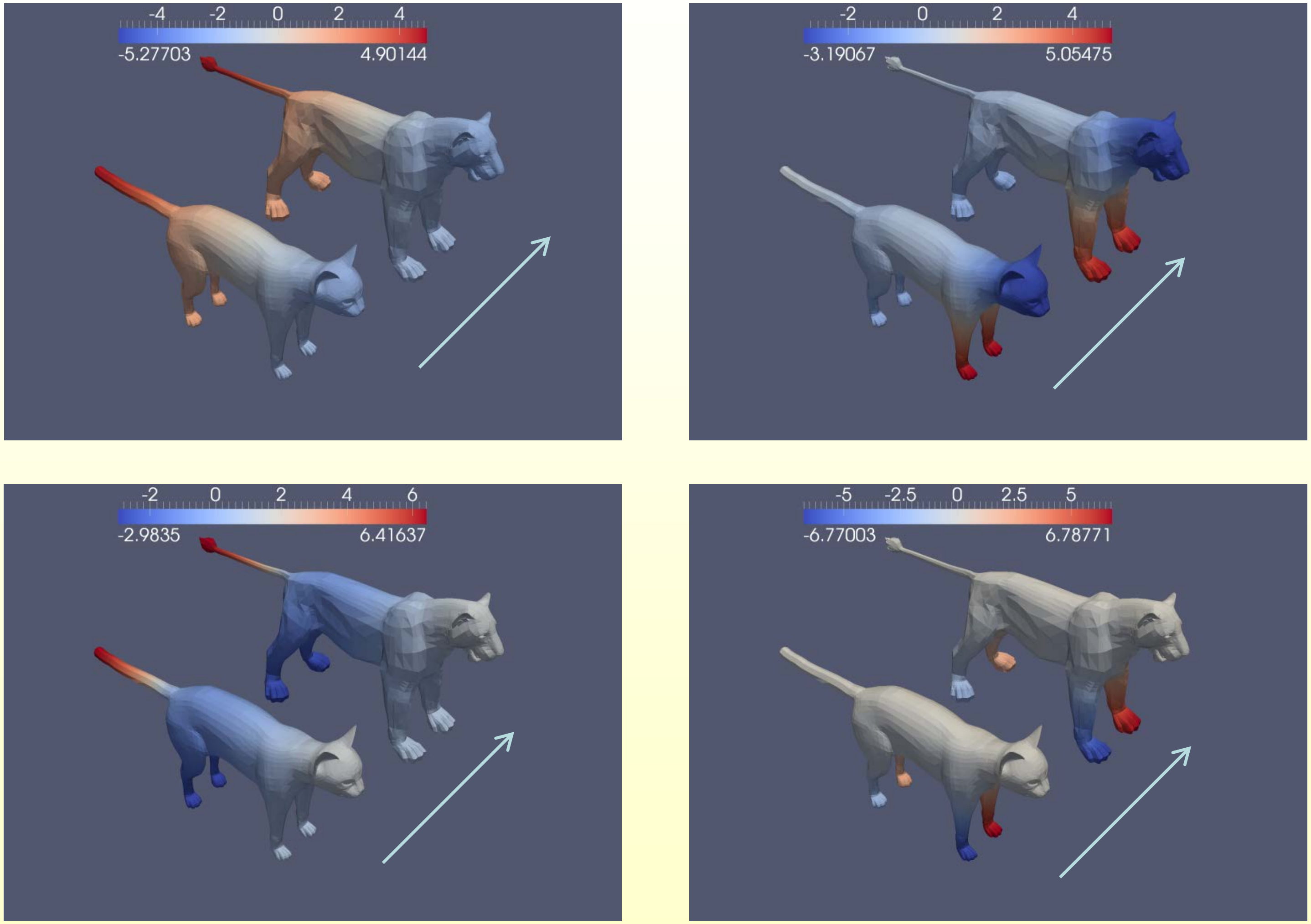


Starting from a Regular Map φ



φ: lion → cat

Attribute Transfer via Pull-Back

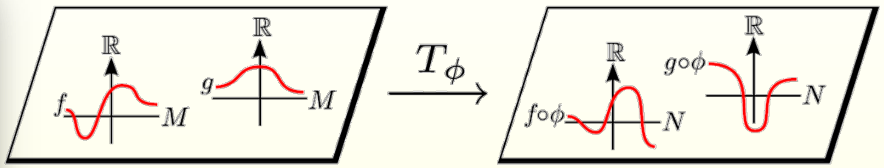


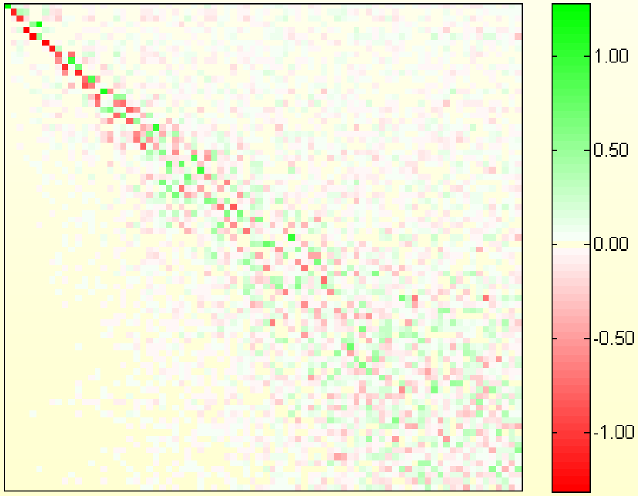
T_{φ}: cat $\rightarrow$ lion

A Contravariant Functor

from cat to lion

Functions on cat are transferred to lion using T_φ



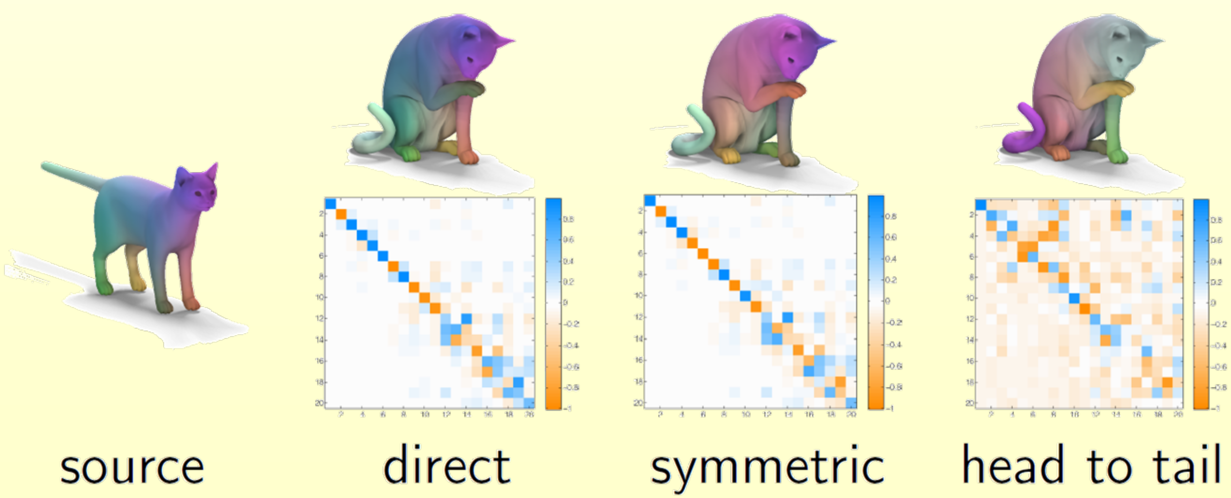


T_φ is a linear operator (matrix)

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

The Functional Framework

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



Using truncated Laplace-Beltrami basis

Estimating the Mapping Matrix

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

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

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

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

Function Preservation Constraints

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

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

Function preservation constraint is quite general and includes:

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

Commutativity Constraints

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

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

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

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

Regularization

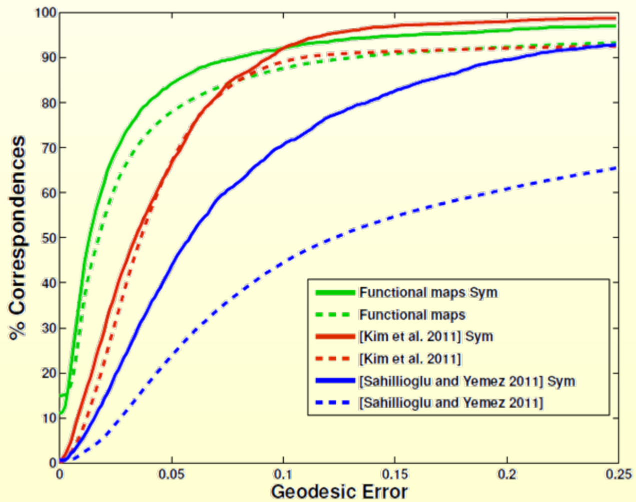
Lemma 1:

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

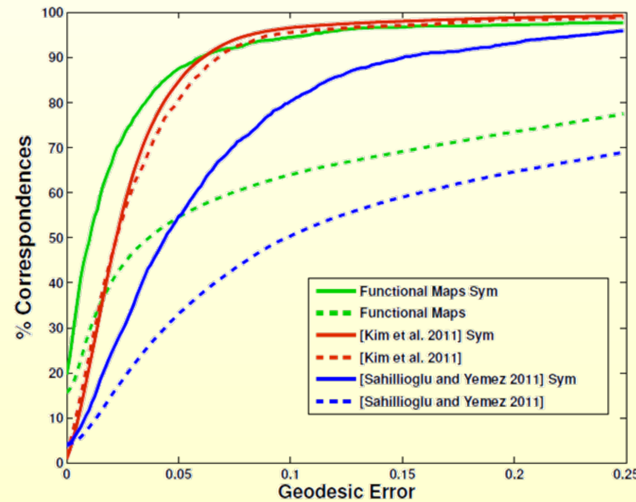
$$C\Delta_1 = \Delta_2 C$$

Map Estimation Quality

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



SCAPE



TOSCA

Roughly 10 probe functions + 1 part correspondence

App: Shape Differences

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

vs.

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

The Art of Measurement

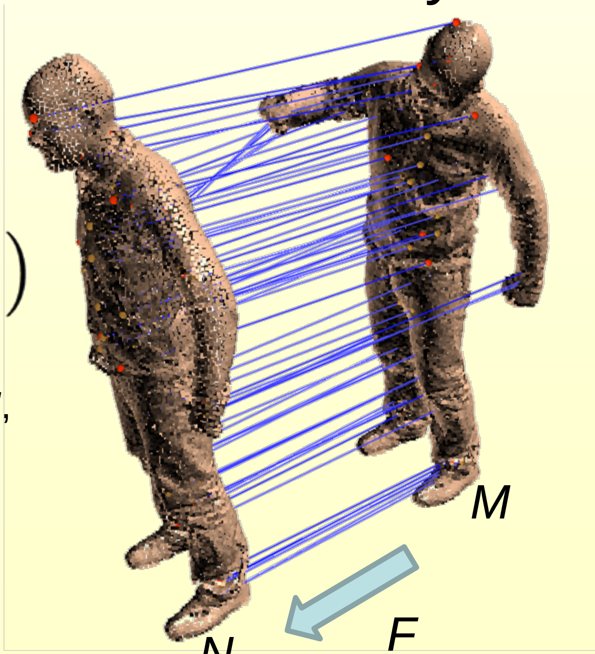
- A metric is defined by a functional inner product

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

Riemann

- So we can compare *M* and *N* by comparing

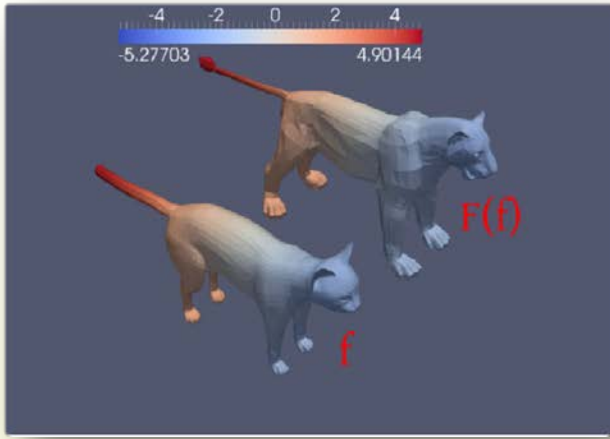
$$h^N(F(f),F(g))$$

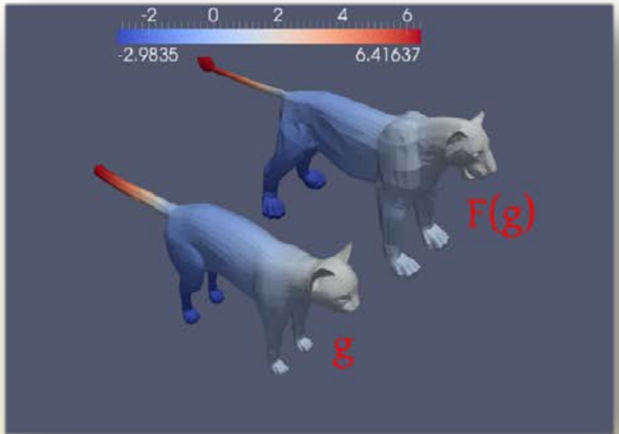


$$h^M(f,g)$$

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



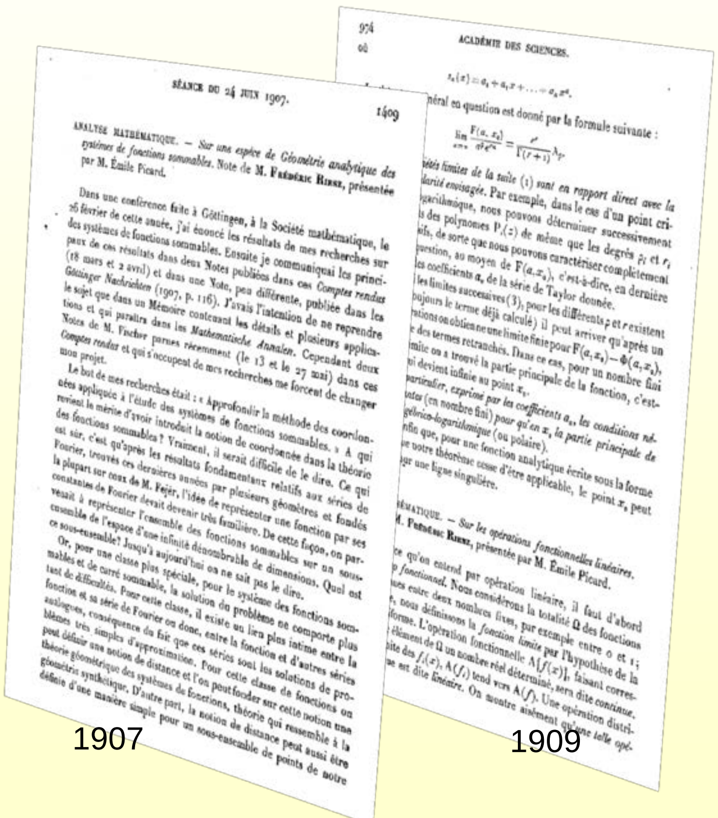


$$\int_{lion} \underbrace{F(f)F(g)\, d\mu_l}_{\text{after}} \neq \int_{cat} \underbrace{f g\, d\mu_c}_{\text{before}}$$

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

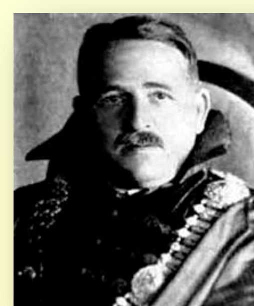
Riesz Representation Theorem

There exists a linear operator

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

such that

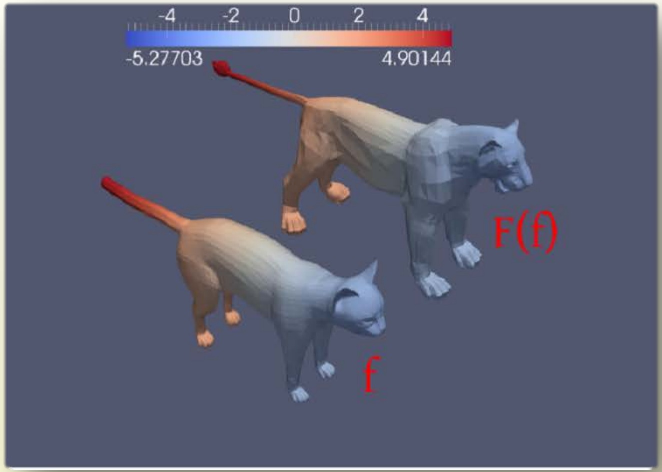
$$\langle f, g \rangle_{\text{after}} = \langle f, V(g) \rangle_{\text{before}}$$

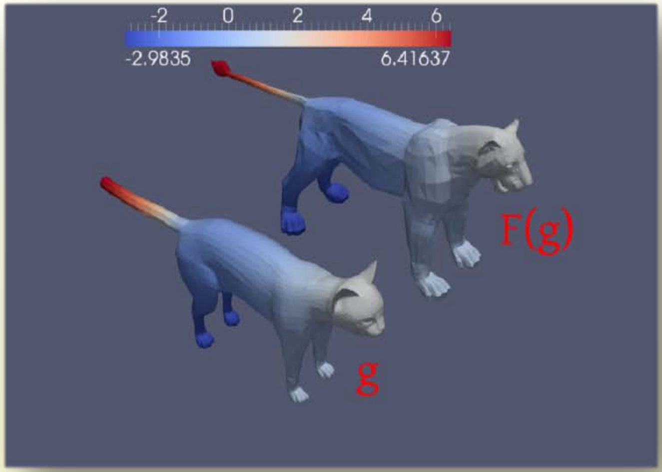


Frigyes Riesz

Area-Based Shape Difference:

$$V \approx F^T F$$



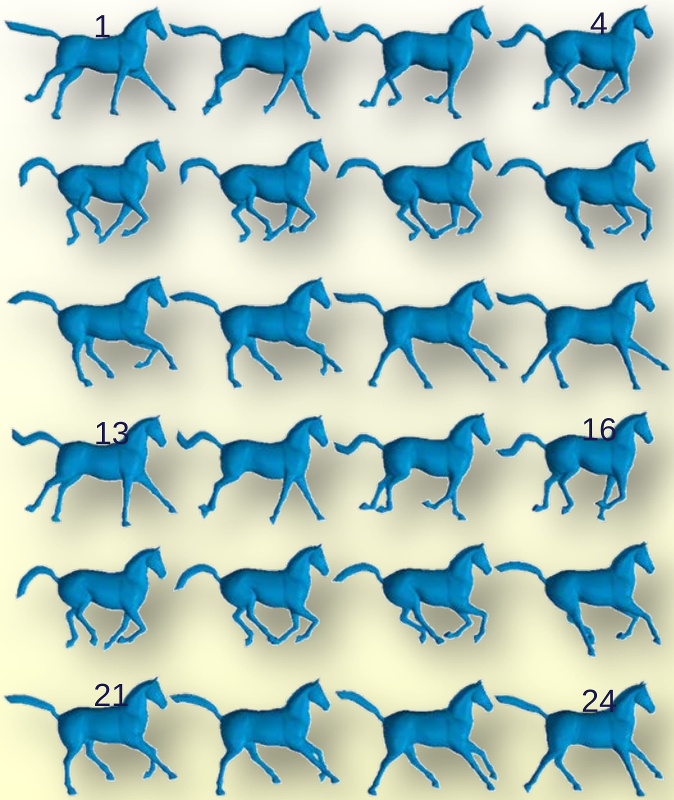


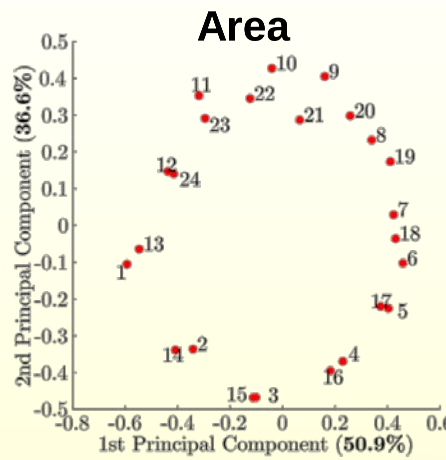
$$\int_{lion} F(f)F(g) \neq \int_{cat} fg$$

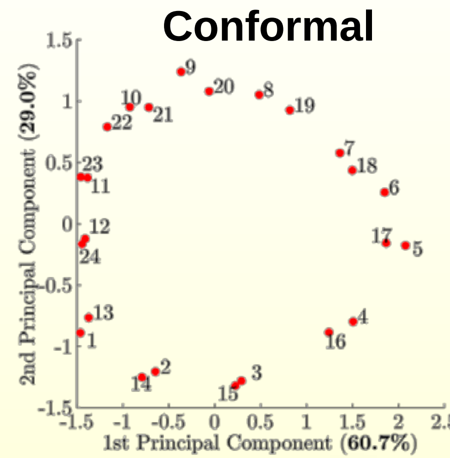
$$\Downarrow$$

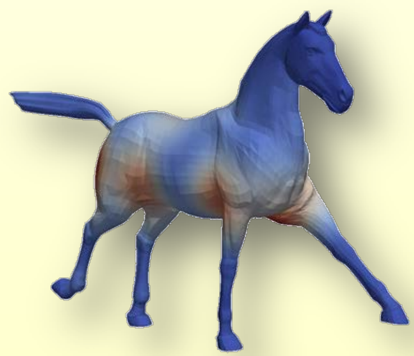
$$\int_{lion} F(f)F(g) = \int_{cat} f V(g)$$

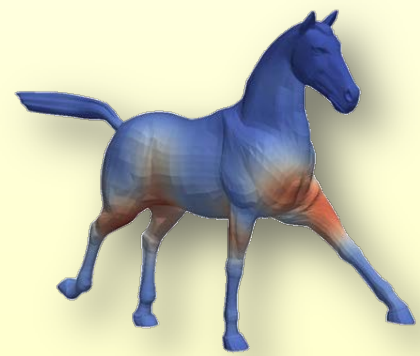
Intrinsic Shape Space



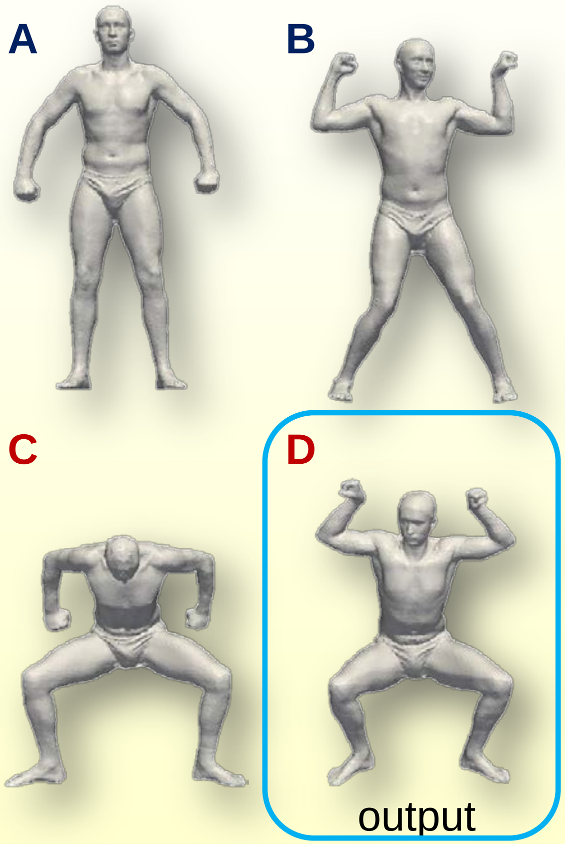






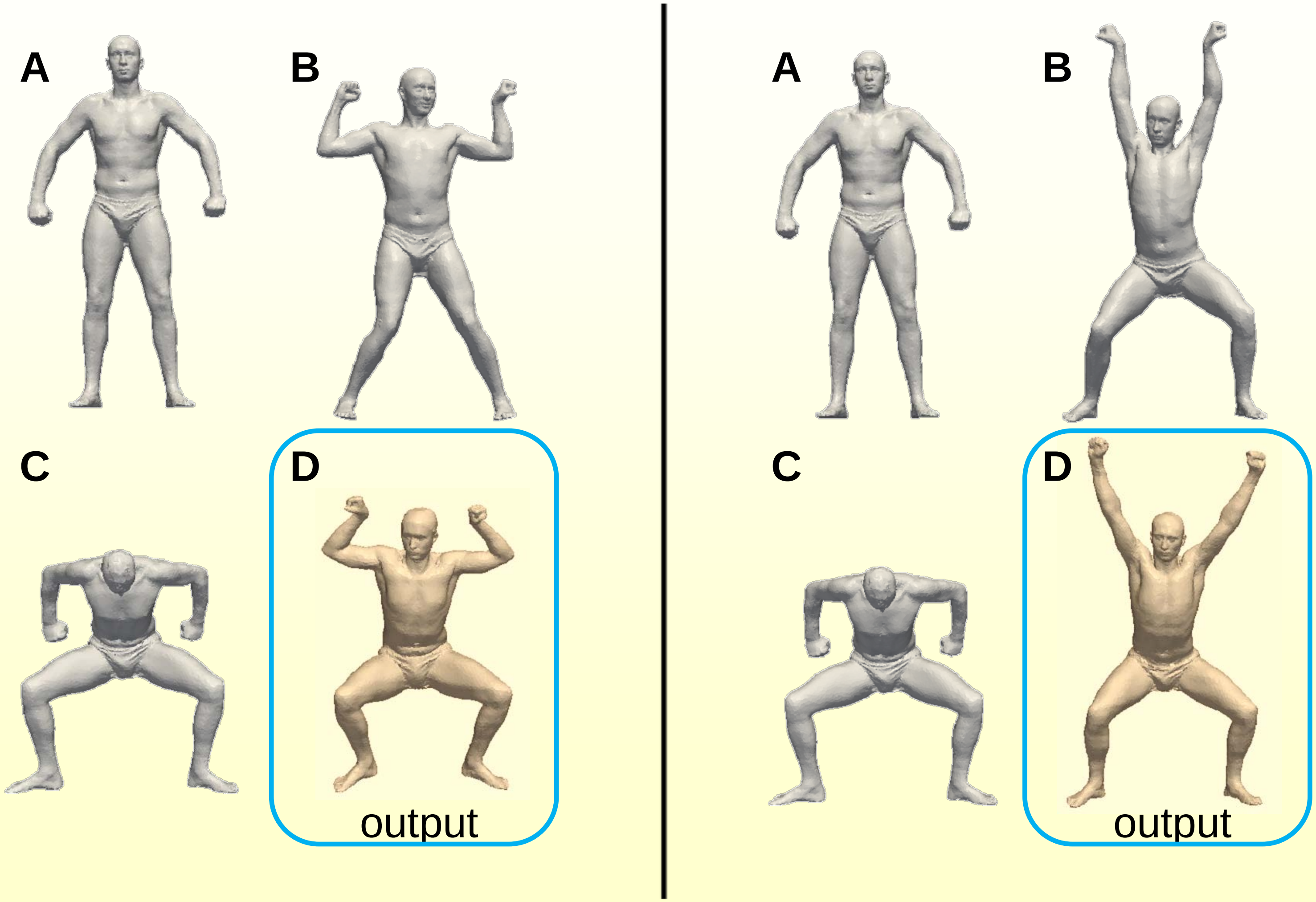


Analogies: D relates to C as B relates to A



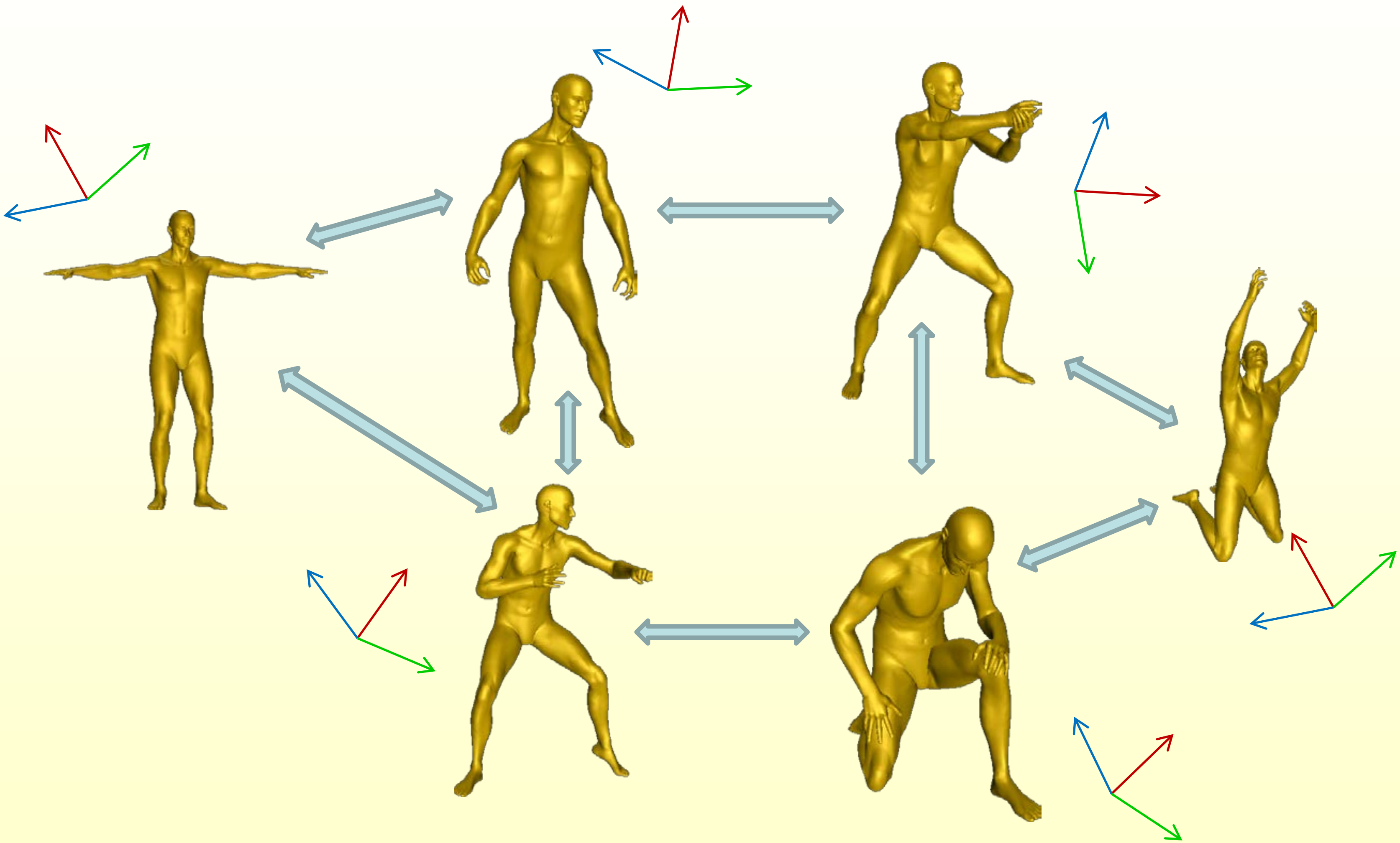
$$D = C + \underbrace{(B - A)}_{\text{hands raised up}}$$

Shape Analogies



The Network View

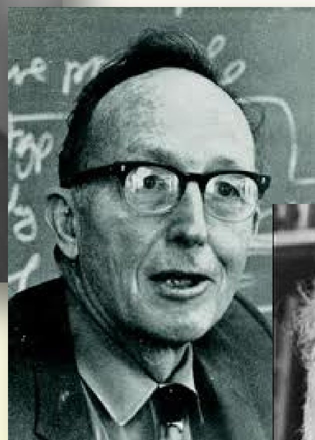
Map Networks for Related Data



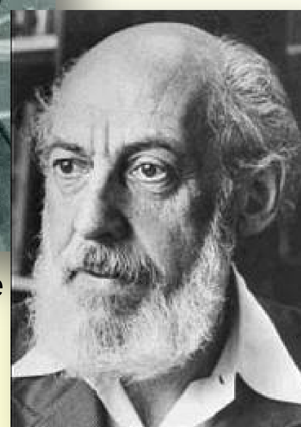
Networks of “samenesses”

A Functorial View of Data

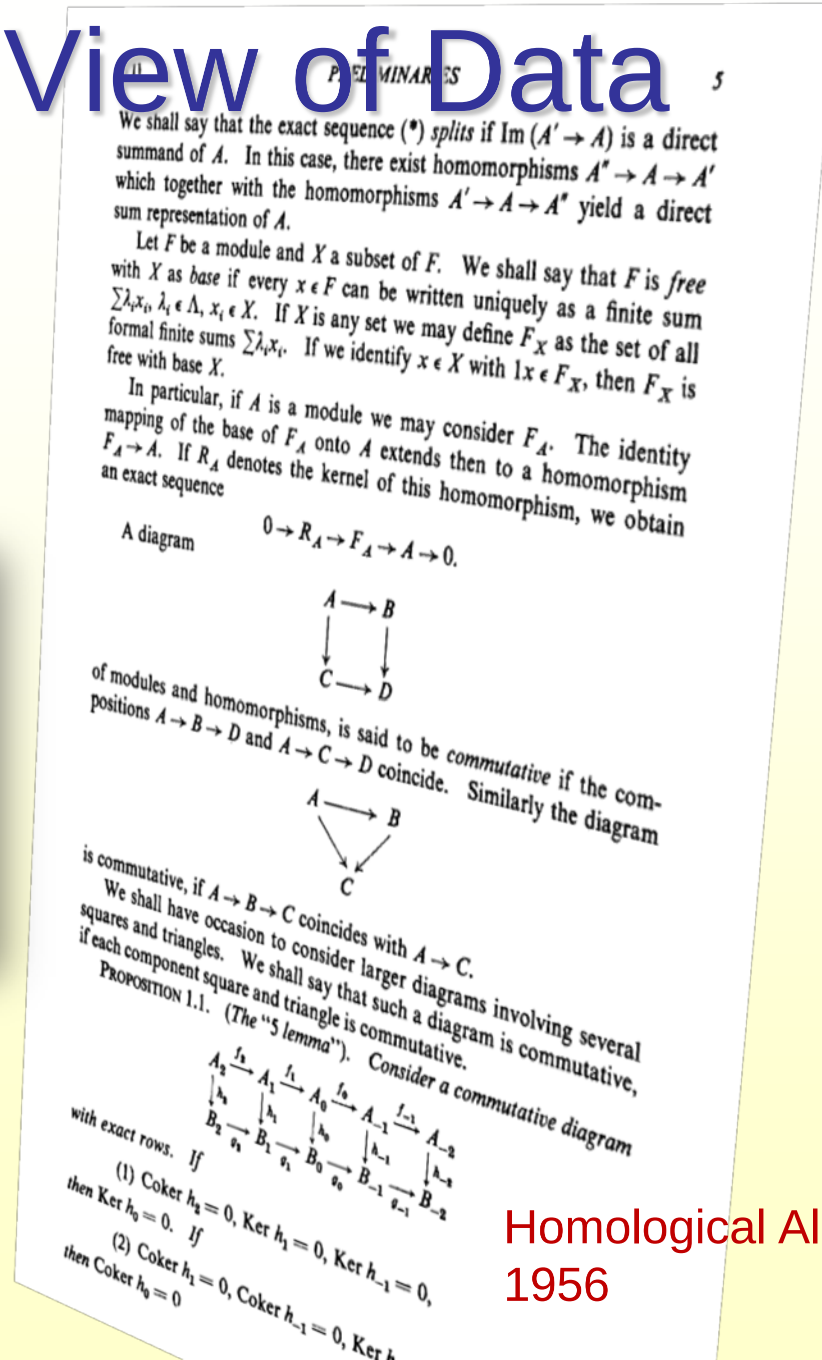
Herni Cartan



Saunders MacLane



Samuel Eilenberg



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

$$\begin{tikzcd} A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd}$$

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

$$\begin{tikzcd} A \arrow[rr] \arrow[dr] & & B \arrow[dl] \\ & C & \end{tikzcd}$$

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*

$$\begin{tikzcd} A_2 \arrow[r, "f_2"] \arrow[d, "h_2"] & A_1 \arrow[r, "f_1"] \arrow[d, "h_1"] & A_0 \arrow[r, "f_0"] \arrow[d, "h_0"] & A_{-1} \arrow[r, "f_{-1}"] \arrow[d, "h_{-1}"] & A_{-2} \arrow[d, "h_{-2}"] \\ B_2 \arrow[r, "\varphi_2"'] & B_1 \arrow[r, "\varphi_1"'] & B_0 \arrow[r, "\varphi_0"'] & B_{-1} \arrow[r, "\varphi_{-1}"'] & B_{-2} \end{tikzcd}$$

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 *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/image space

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

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

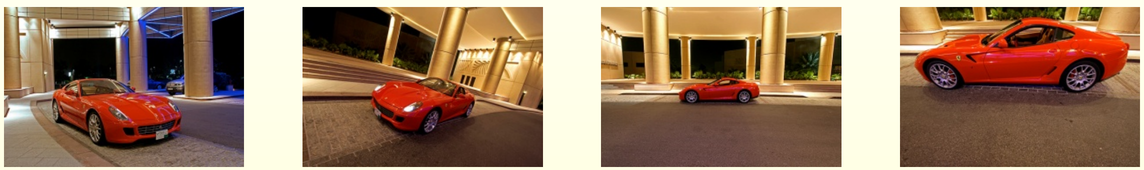
- Conversely, such a low-rank condition can be used to regularize functional maps

Shared Structure Discovery

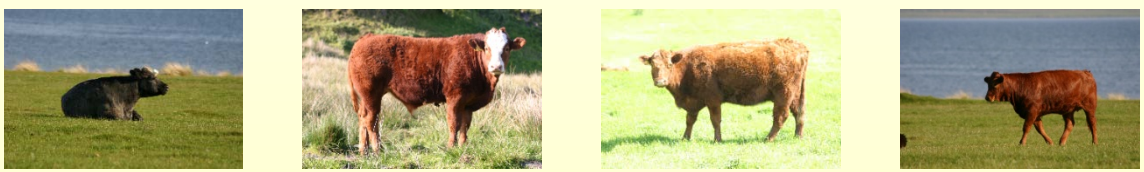
Entity Extraction in Images

[F. Wang, Q. Huang, L. G., ICCV ’13]

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



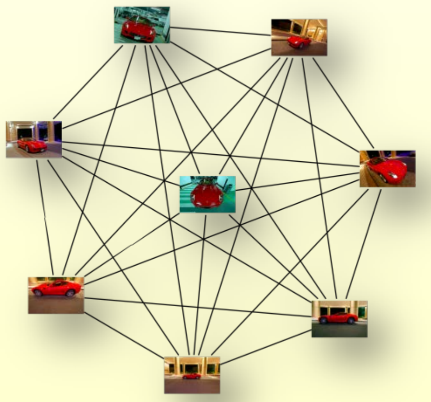
  - similar objects of the same class:



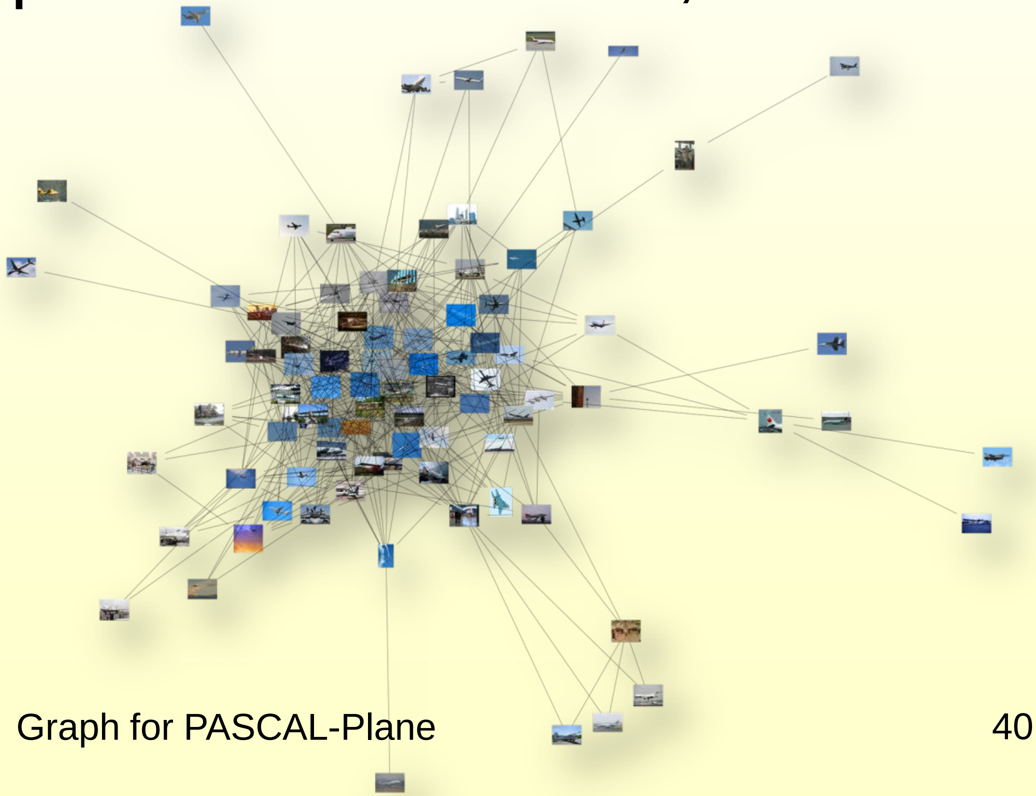
- Benefits and challenges:
  - Images can provide weak supervision for each other
  - But exactly how should they help each other? How to deal with clutter and irrelevant content?

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.

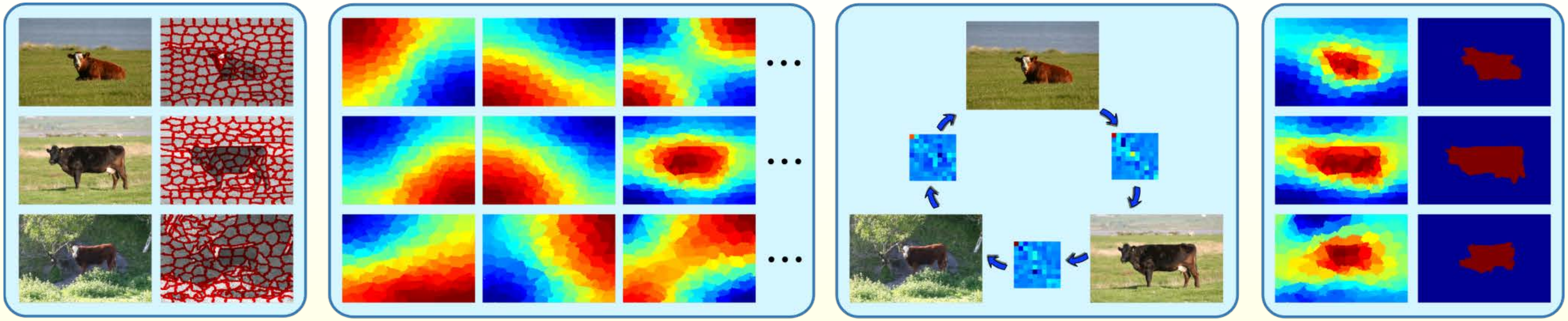


Graph for iCoseg-Ferrari



Graph for PASCAL-Plane

The Pipeline



(a) (b) (c) (d)

a) Superpixel graph representation of images

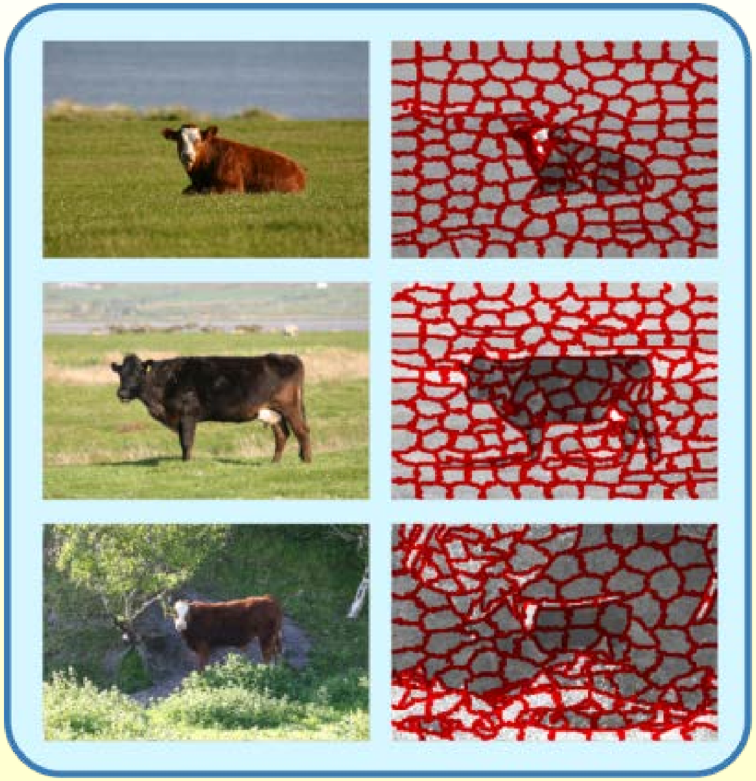
b) Functions over these graphs expressed in terms of the eigenvectors of the graph Laplacian

c) Estimation of functional maps along network edges such that
- Image features are preserved
- Maps are cycle consistent in the network

d) The “cow functions” emerge as the most consistently transported set

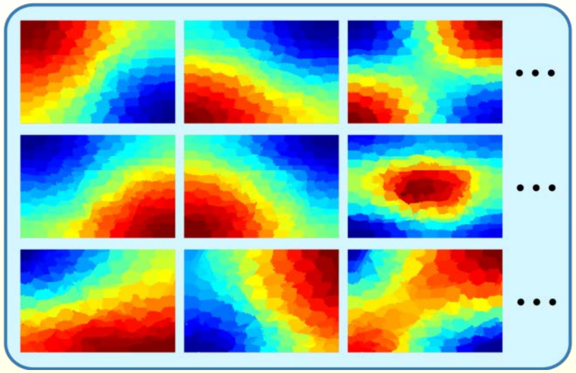
Superpixel Representation

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



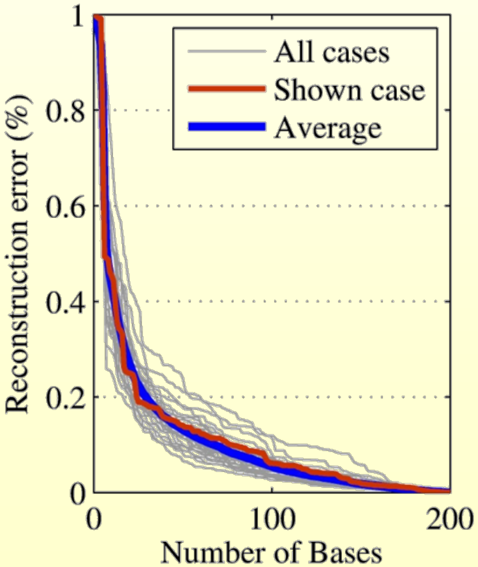
Encoding Functions over Graphs

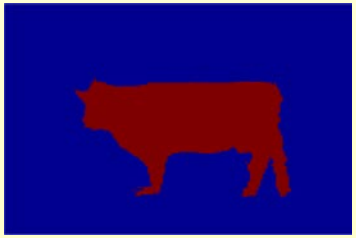
- Basis of functional space
  - : First M Laplacian eigenfunctions of the graph



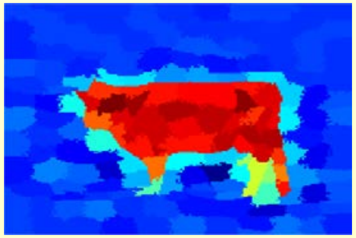
$$f = \sum_{j=1}^{M} f_j b_i^j = B_i \mathbf{f}$$

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

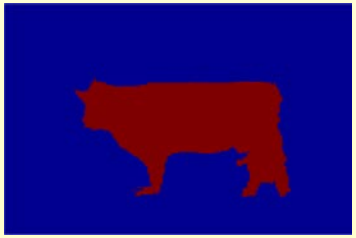




Binary indicator function



Reconstructed function



Thresholded reconstructed function

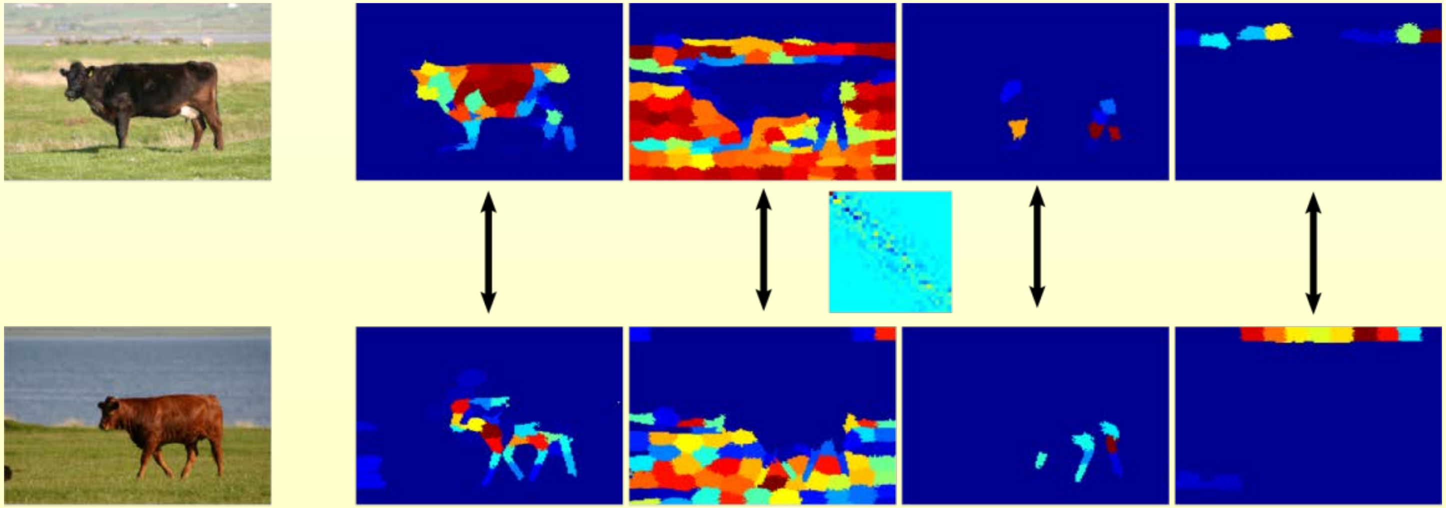
Reconstruction error

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.

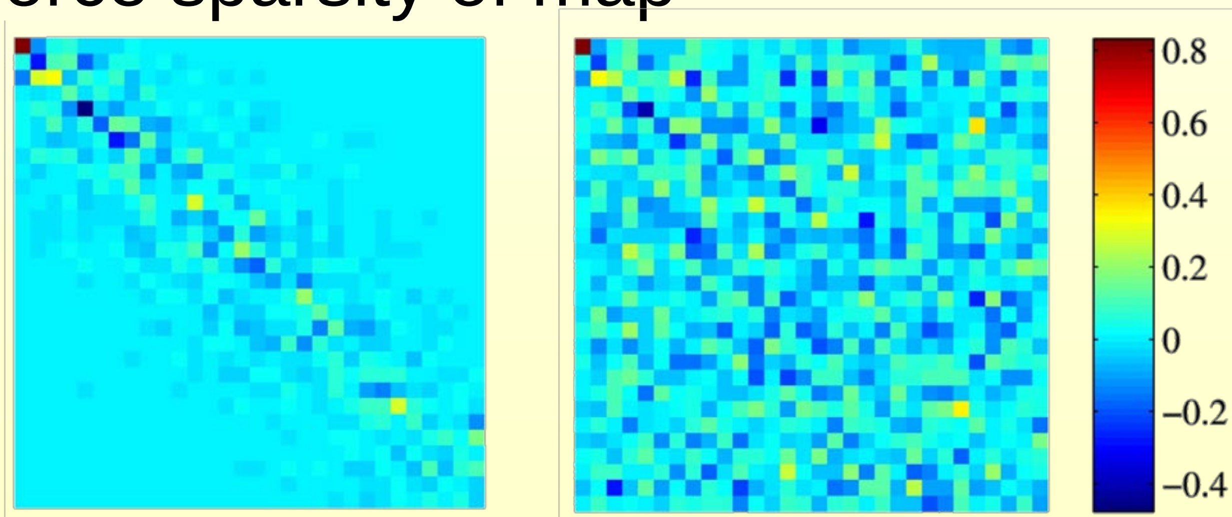
Joint Estimation of Functional Maps, II

- Regularization term:

Λ_i, Λ_j diagonal matrices of Laplacian eigenvalues

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

  - 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:

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

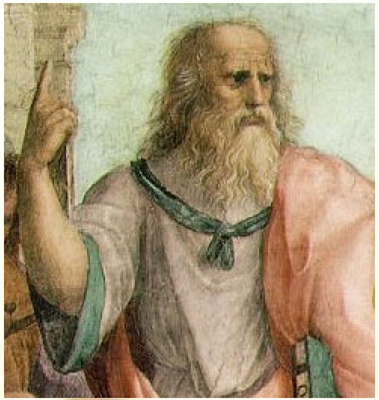
  - Consistency term:

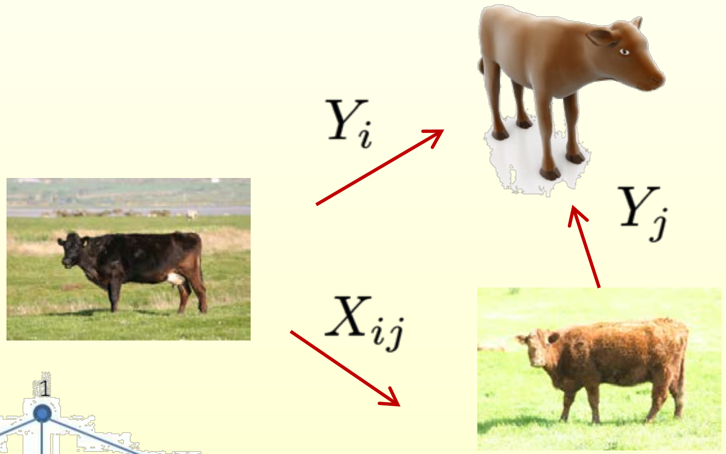
$$f^{\text{cons}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} f_{ij}^{\text{cons}} = \sum_{(i,j)\in\mathcal{G}} w_{ij} \|X_{ij} Y_i - Y_j\|_{\mathcal{F}}^2$$

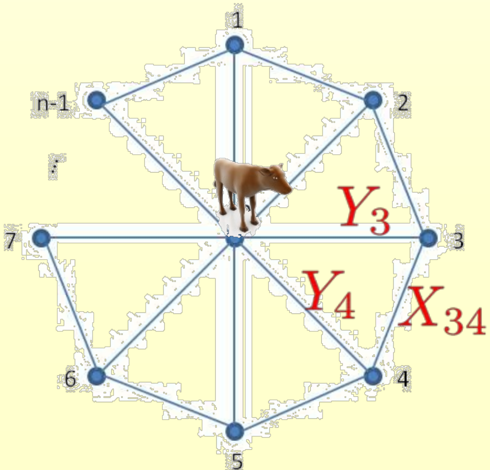
Image global similarity weight via GIST

Joint Estimation of Functional Maps, III

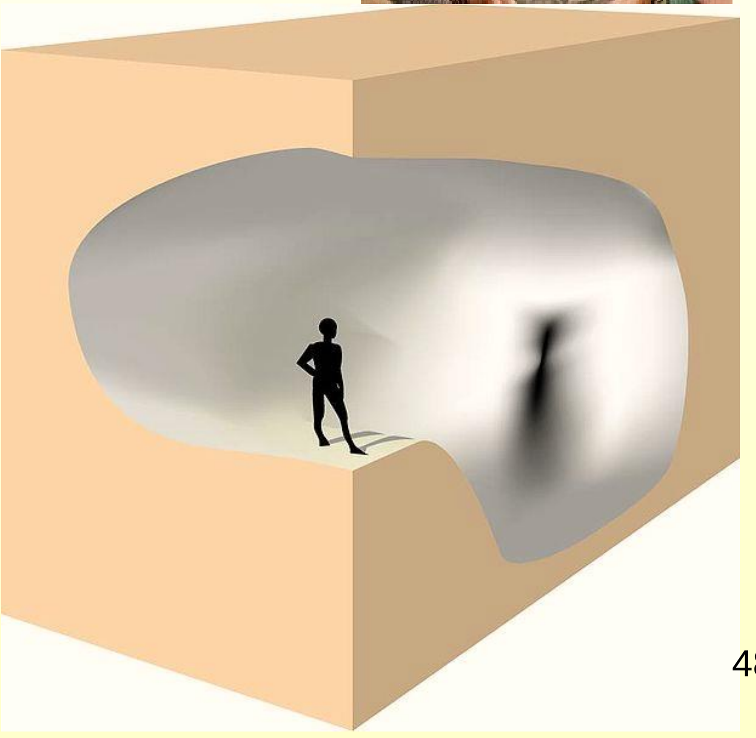
- Plato's allegory of the cave







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



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 $\Longrightarrow$ 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 $\Longrightarrow$ Eigenvalue problem

$$\begin{aligned} \min \quad & \operatorname{trace}(Y^T W Y) \\ s.t. \quad & Y^T Y = I_m \end{aligned} \qquad W_{ij} = \begin{cases} \sum_{(i,j')\in\mathcal{G}} w_{ij'}(I_m + X_{ij'}^T X_{ij'}) & i = j \\ -w_{ij}(X_{ji} + X_{ij}^T) & (i,j)\in\mathcal{G} \\ 0 & \text{otherwise.} \end{cases}$$

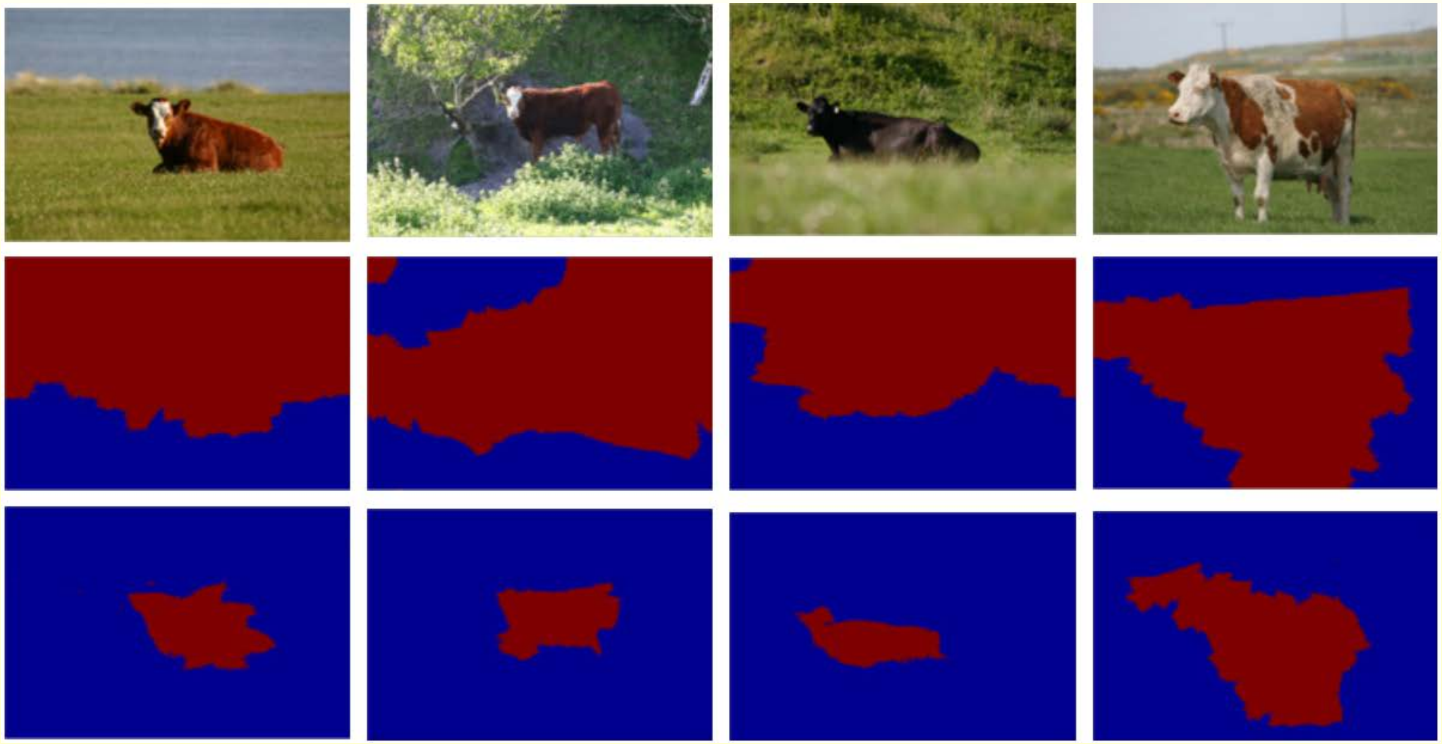
Consistency Matters

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

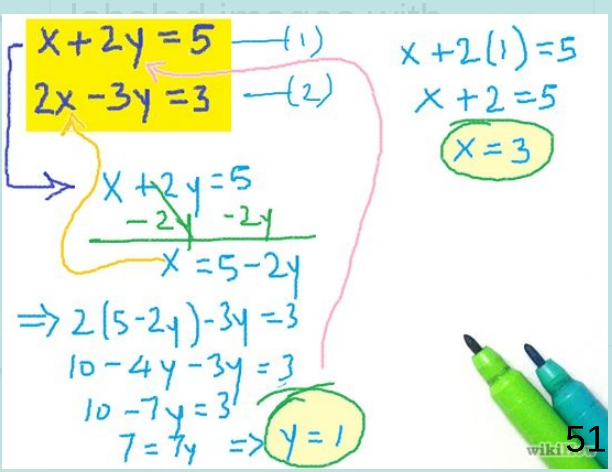
$$f^{\text{seg}} = \sum_{i=1}^{N} \mathbf{f}_i^T B_i^T L_i B_i \mathbf{f}_i$$

Easy to incorporate labeled images with

- Joint optimization:

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

We look for network fixed points!



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

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

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**

- 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

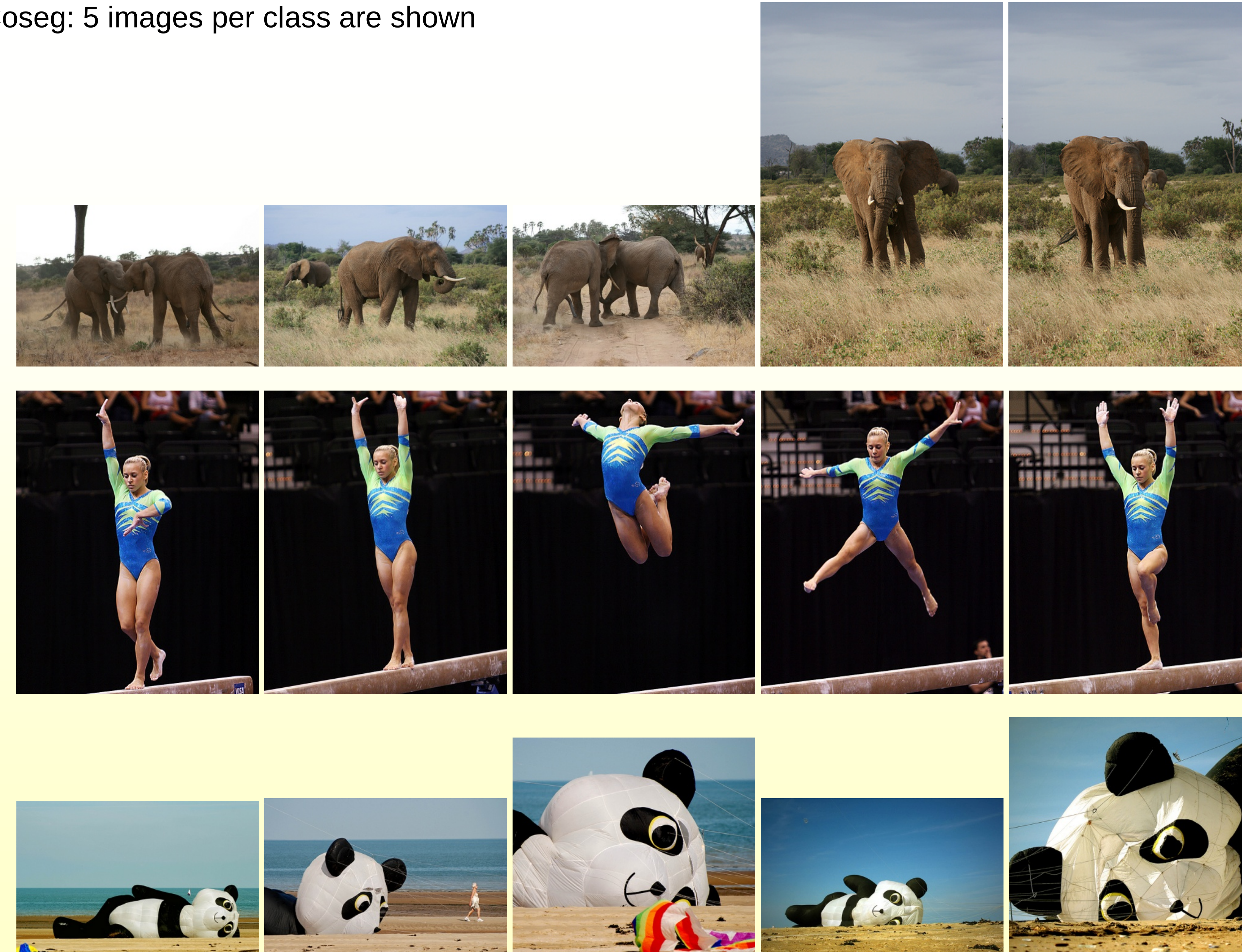
Class	N	L	Kuettel '12	Fmaps -s	Fmaps -uns
Plane	178	88	90.7	**92.1**	89.4
Bus	152	78	81.6	**87.1**	80.7
Car	255	128	76.1	**90.9**	82.3
Cat	250	131	77.7	**85.5**	82.5
Cow	135	64	82.5	**87.7**	85.5
Dog	249	121	81.9	**88.5**	84.2
Horse	147	68	83.1	**88.9**	87.0
Sheep	120	63	83.9	**89.6**	86.5

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

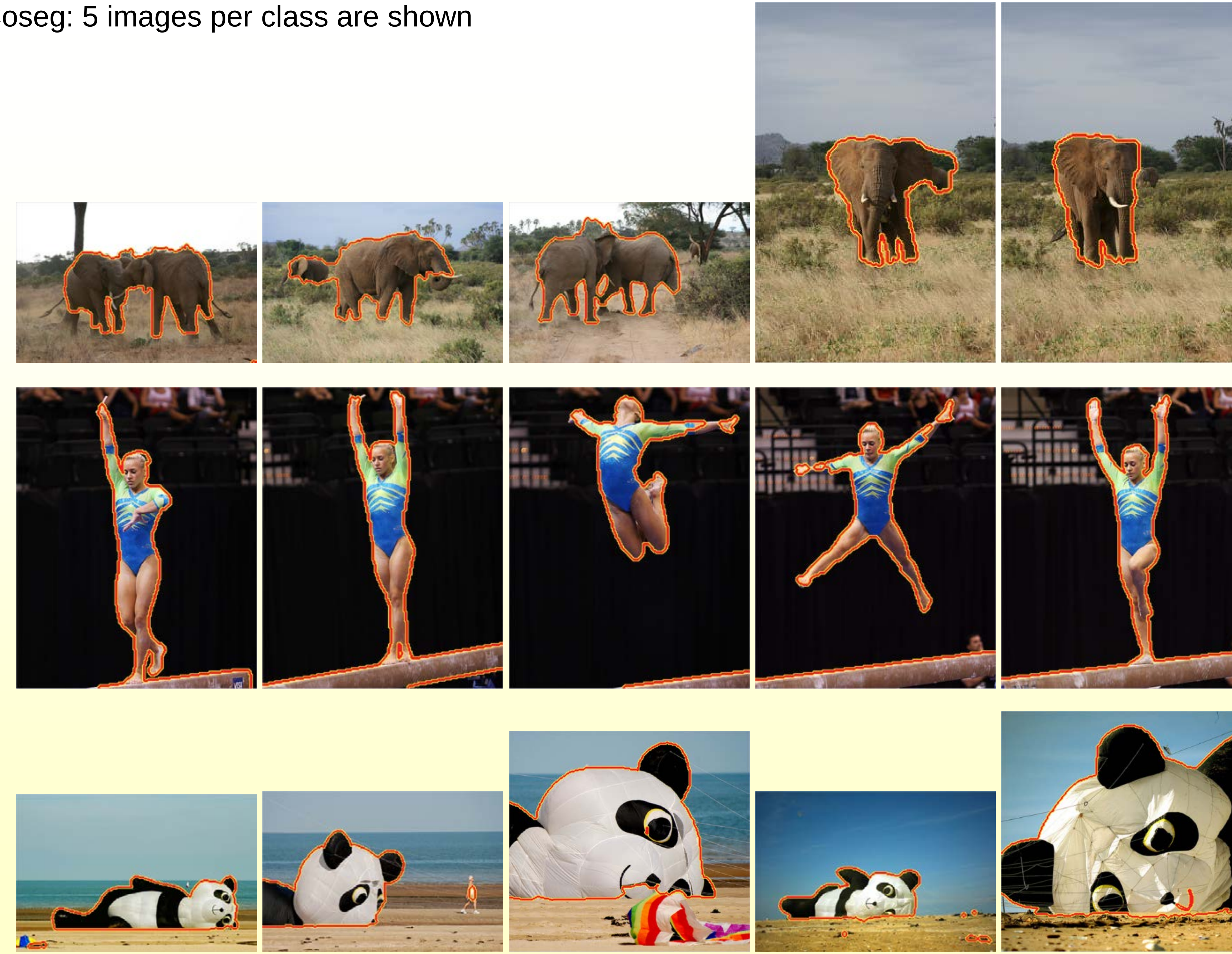
iCoseg: 5 images per class are shown

iCoseg: 5 images per class are shown

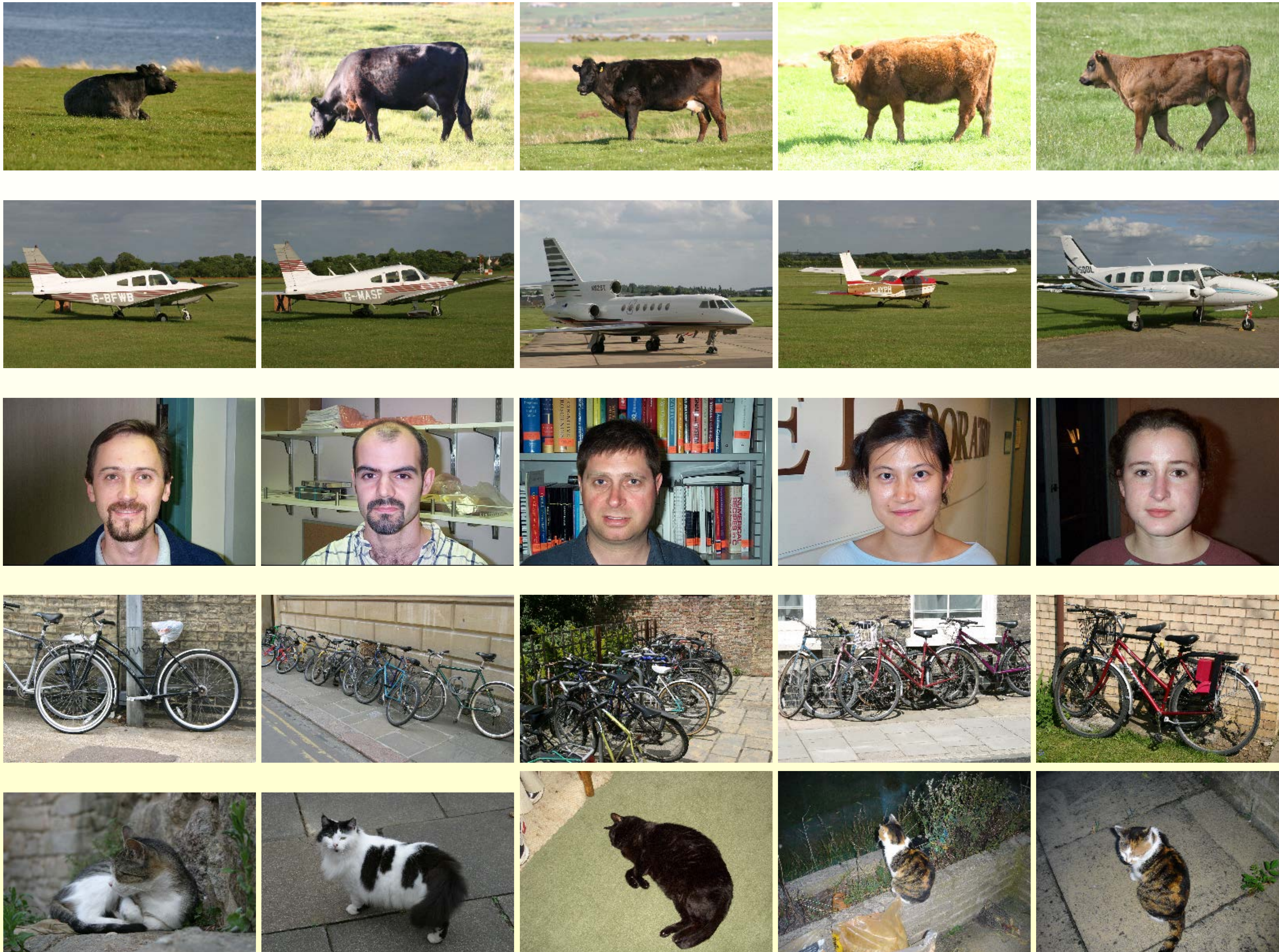
iCoseg: 5 images per class are shown



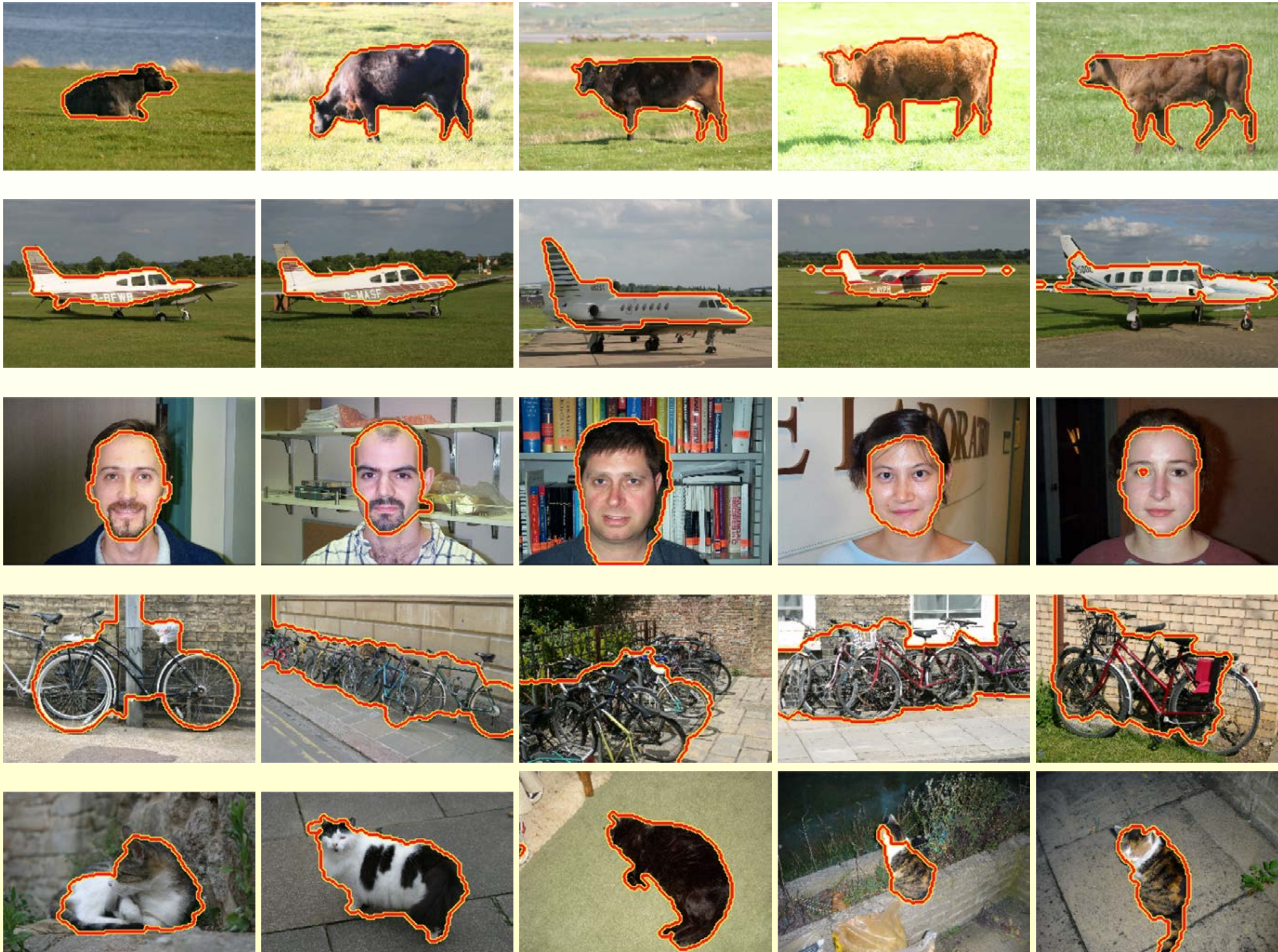
iCoseg: 5 images per class are shown



MSRC: 5 images per class are shown

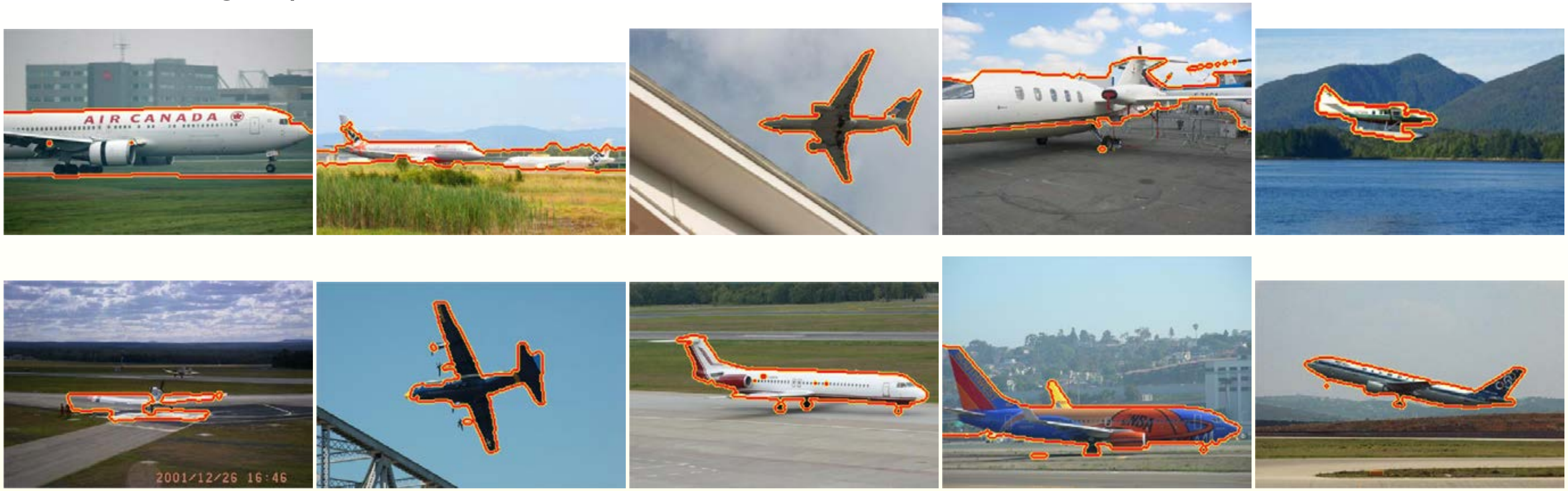


MSRC: 5 images per class are shown



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]

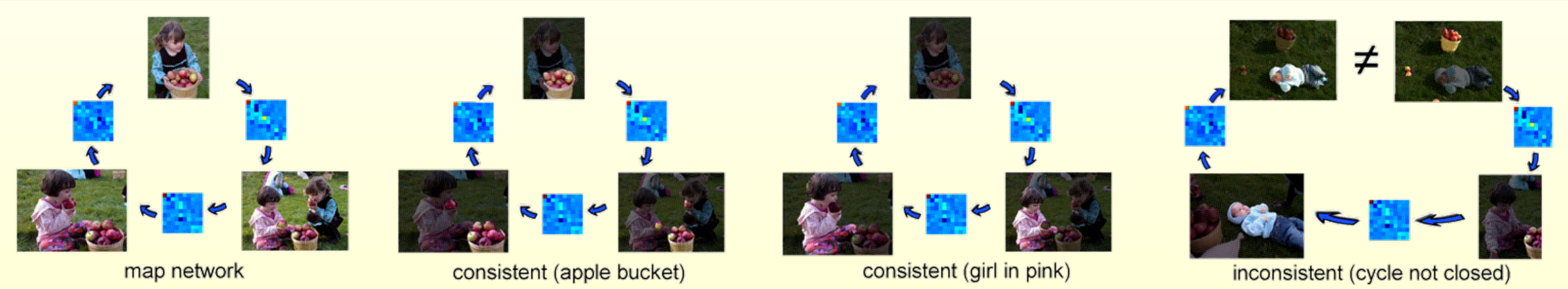
- Input:
  - A collection of N images sharing M objects
  - Each image contains a subset of the objects

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

Consistent Functional Maps

- Partial cycle consistency:



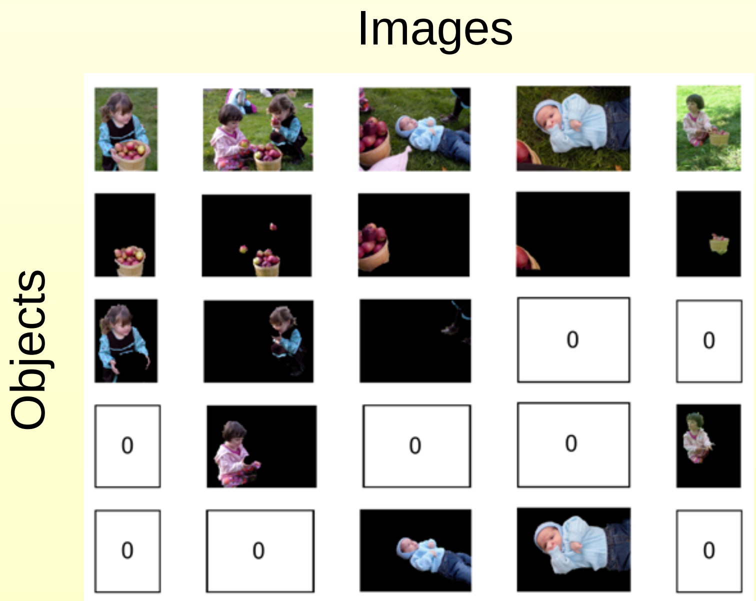
Must deal with non-total maps

Related to topological persistence / persistent homology

Consistent Functional Maps

- Latent functions: $\boldsymbol{Y}_i = (\boldsymbol{y}_{i1}, \cdots, \boldsymbol{y}_{iL})$
- Discrete variables: $\boldsymbol{z}_i = \{z_{il} \in \{0, 1\}, 1 \leq l \leq L\}$
- Relationship: $\mathbf{Y}_i \mathrm{Diag}(\mathbf{z}_i) = \mathbf{Y}_i$
- Consistency:

$$\boldsymbol{X}_{ij}\boldsymbol{Y}_i = \boldsymbol{Y}_j \mathrm{Diag}(\boldsymbol{z}_i), \quad (i, j) \in \mathcal{E}.$$



Consistent Functional Maps

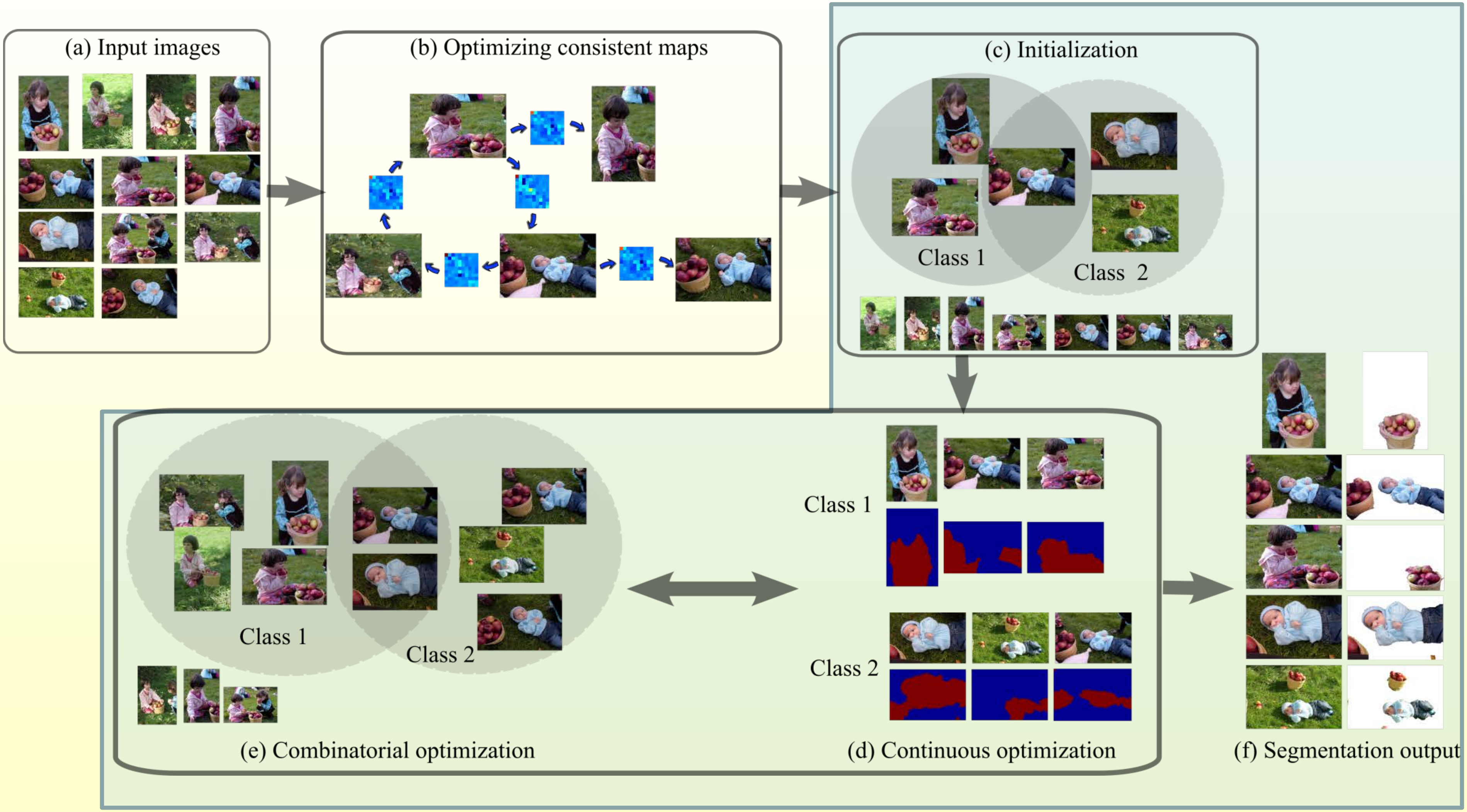
- The consistency regularization

$$f_{\text{cons}} = \mu \sum_{(i,j)\in\mathcal{E}} \|\mathbf{X}_{ij}\mathbf{Y}_i - \mathbf{Y}_j \text{Diag}(\boldsymbol{z}_i)\|^2 + \gamma \sum_{i=1}^{N} \|\mathbf{Y}_i - \mathbf{Y}_i \text{Diag}(\boldsymbol{z}_i)\|^2,$$

- Overall optimization

$$\{X_{ij}^{\star}\} = \text{argmin}_{X_{ij}} \left(\mu f_{\text{cons}} + \sum_{(i,j)\in\mathcal{E}} f_{\text{pair}}\right)$$

Framework


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

Initialization

- Solve for consistent segmentation with ALL images together

$$f_{seg} = \frac{1}{|\mathcal{G}|} \sum_{(i,j)\in\mathcal{G}} \|X_{ij}s_{ik} - s_{jk}\|_F^2 + \frac{\gamma}{N}\sum_{i=1}^{N} s_{ik}^T L_i s_{ik}$$
$$= s_k \overline{L} s_k,$$

- Pick the first M eigenvectors
- Each object class is initialized as:

$$\mathcal{C}_k = \{i,\ \text{s.t.}\ \|s_{ik}\| \geq \max_i \|s_i\|/2\}$$

Optimizing Segmentation Functions

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

Continuous Optimization

- Optimize segmentations in each object class
  - Consistent with functional maps
  - Align with segmentation cues
  - Mutually exclusive

$$\begin{aligned} \min_{s_{ik}, i \in \mathcal{C}_k} \quad & \sum_{k=1}^{M} \sum_{(i,j) \in \mathcal{E} \cap (\mathcal{C}_k \times \mathcal{C}_k)} \|X_{ij} s_{ik} - s_{jk}\|^2 \\ & + \gamma \sum_{l \neq k} \sum_{i \in \mathcal{C}_k \cap \mathcal{C}_l} (s_{il}^T s_{ik})^2 + \mu \sum_{k=1}^{M} \sum_{i \in \mathcal{C}_k} s_{ik}^T L_i s_{ik} \\ \text{subject to} \quad & \sum_{i \in \mathcal{C}_k} \|s_{ik}\|^2 = |\mathcal{C}_k|, \quad 1 \leq k \leq K. \end{aligned}$$

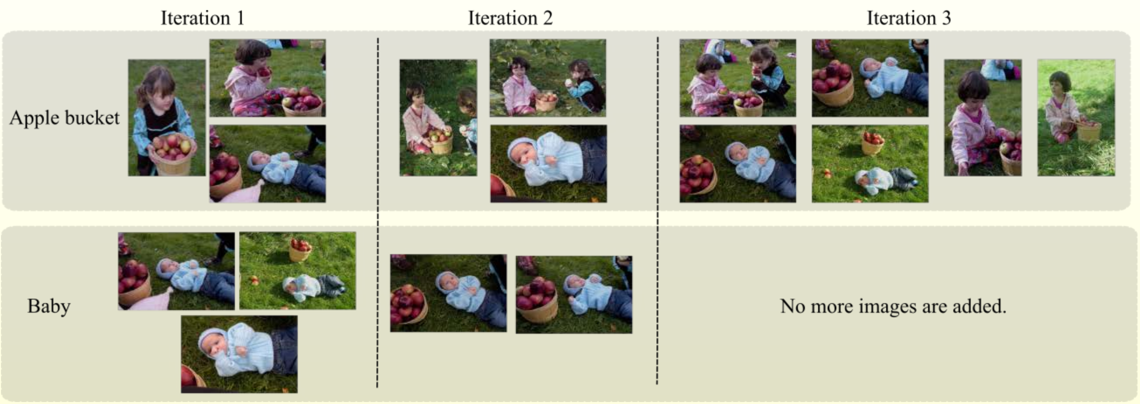
Combinatorial Optimization

- Expand each object class by propagating segmentations to other images

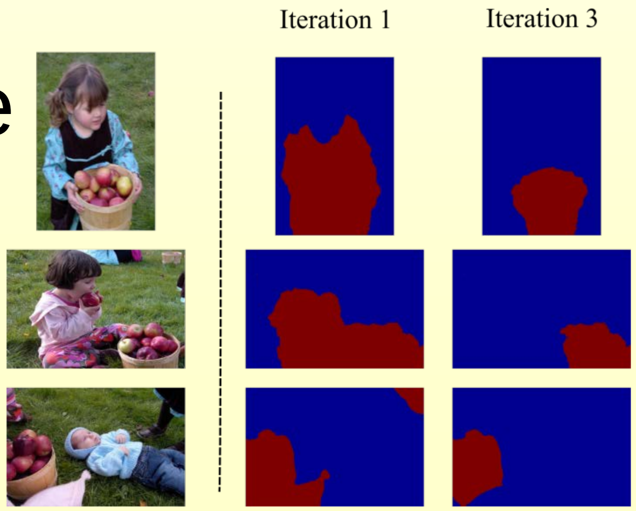
$$\begin{aligned}\max_{s_{ik}} \quad & \frac{1}{|\mathcal{N}(i)\cap\mathcal{C}_k|}\sum_{j\in\mathcal{N}(i)\cap\mathcal{C}_k}(s_{ik}^T X_{ji} s_{jk})^2 \\ & -\gamma\sum_{l\neq k, i\in\mathcal{C}_l}(s_{ik}^T s_{il})^2 - \mu s_{ik}^T L_i s_{ik} \\ \text{subject to} \quad & \|s_{ik}\|^2 = 1\end{aligned}$$

Optimizing Segmentation Functions

- More images will be included in each object class



- Segmentation functions are improved during iterations



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

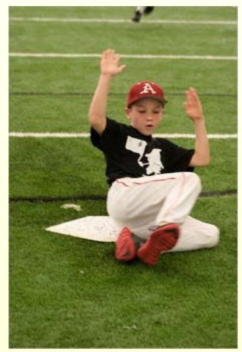
Apple + picking



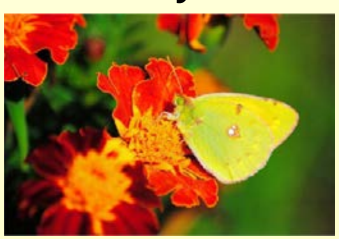


Baseball + kids





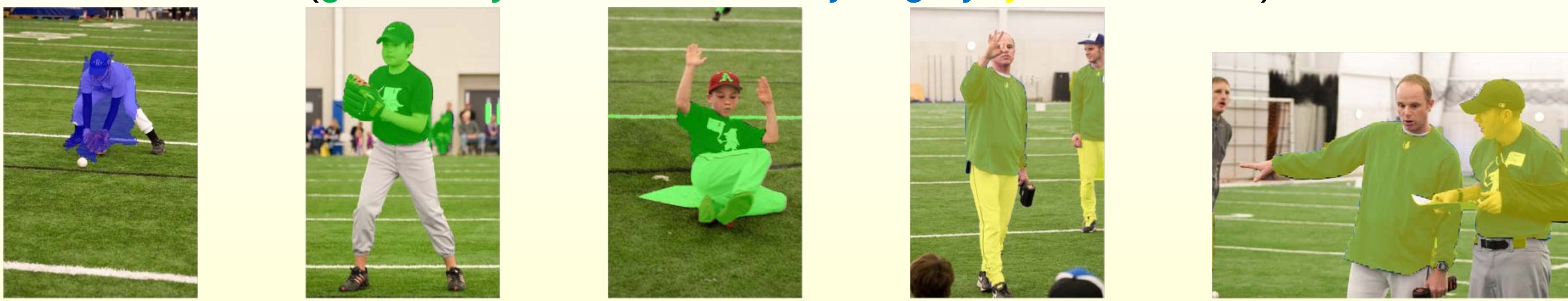
Butterfly + blossom





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

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



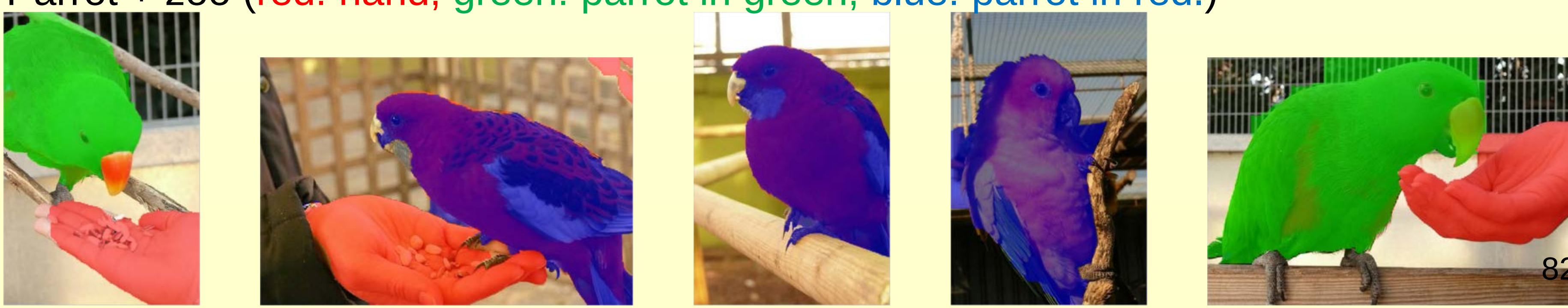


Fishing + Alaska (blue: man in white; green: man in gray; magenta: woman in gray; yellow: salmon.

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





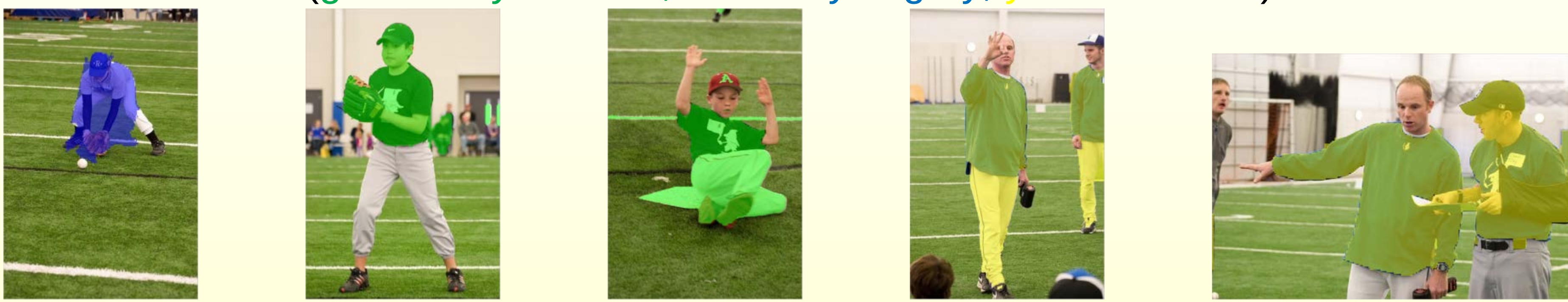
Stonehenge (blue: cow in white; yellow: person; magenta: stonehenge.)

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

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

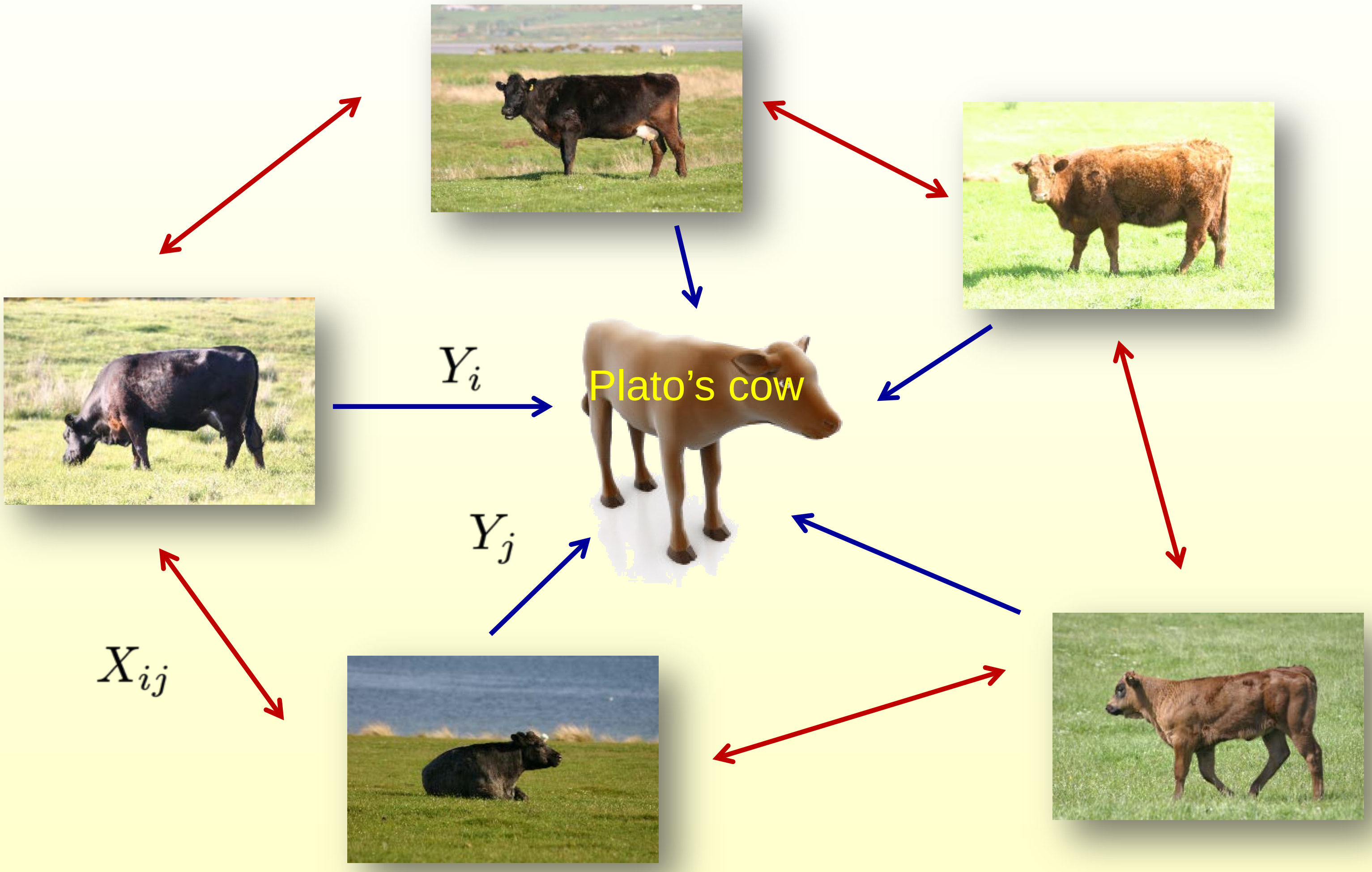
Apple + picking (red: apple bucket; magenta: girl in red; yellow: girl in blue; green: baby; cyan: pum

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

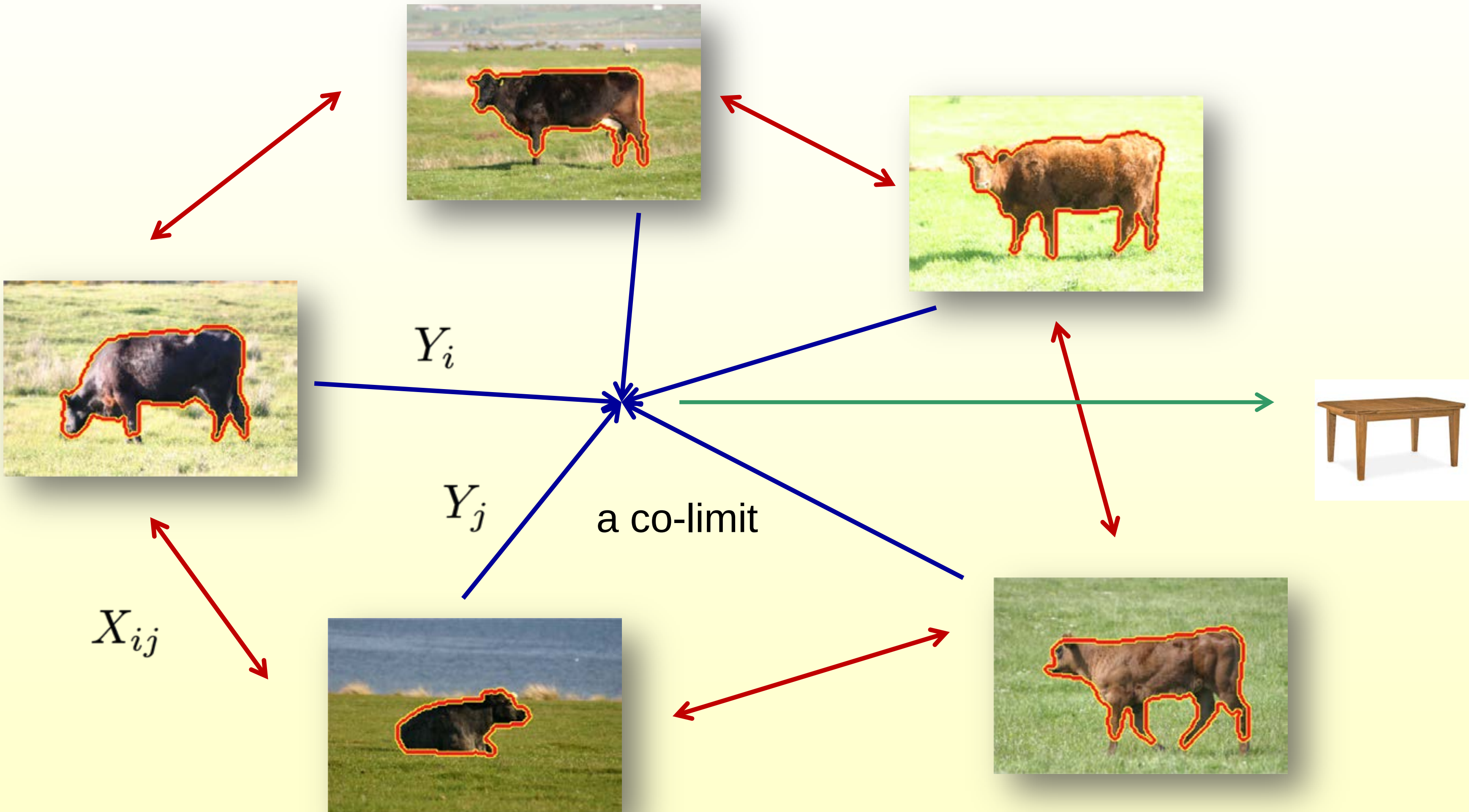


Butterfly + blossom (green: butterfly in orange; yellow: butterfly in yellow; cyan: red flowe

The Network is the Abstraction

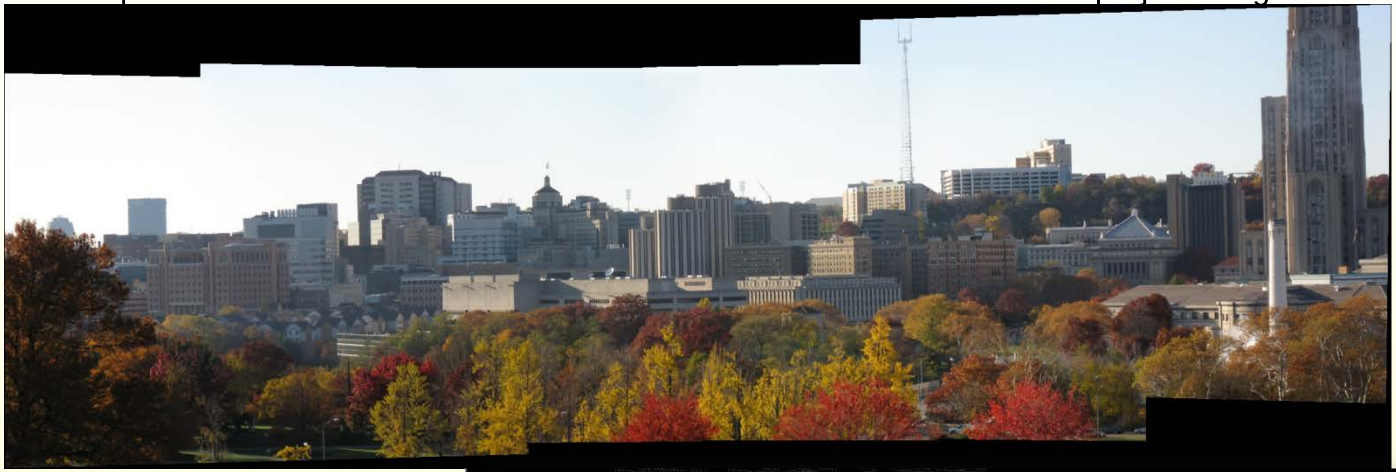


The Network is the Abstraction



Mosaicing or SLAM at the Level of Functions

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



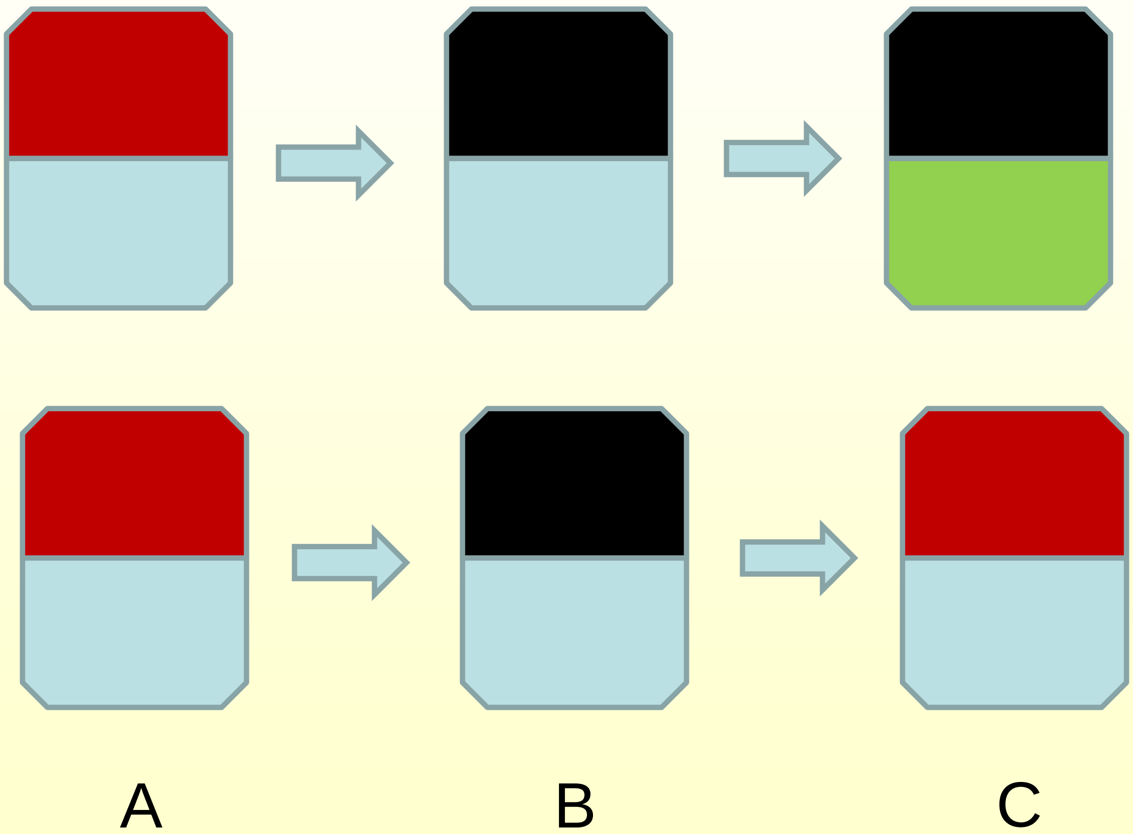
robotics.ait.kyushu-u.ac.jp

Networks of Shapes and Images

Depth Inference from a Single Image

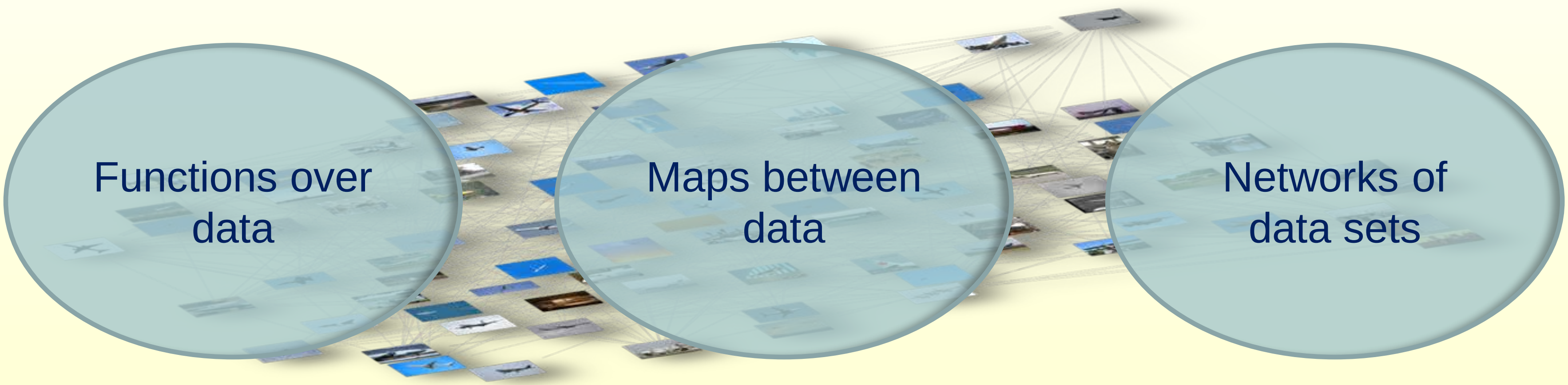
single image shape network inferred depth

Maps vs. Distances/Similarities
Networks vs. Graphs



Conclusion: Functoriality

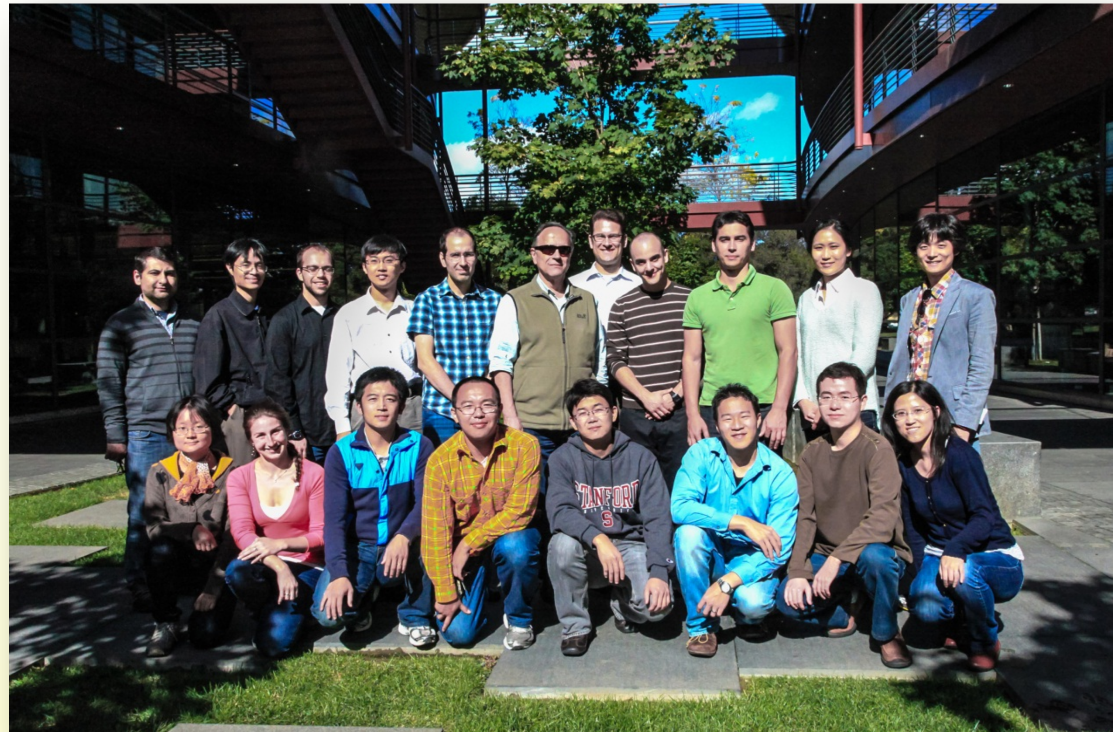
- Classical "vertical" view of data analysis:
  - Signals to symbols
    - from features, to parts, to semantics …



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

Acknowledgements

- Collaborators:
  - Current students: Justin Solomon, Fan Wang
  - Current and past postdocs: Adrian Butscher, Qixing Huang, Raif Rustamov
  - Senior: Mirela Ben-Chen, Frederic Chazal, Maks Ovsjanikov



- Sponsors:

htc quietly brilliant

ONR Office of Naval Research

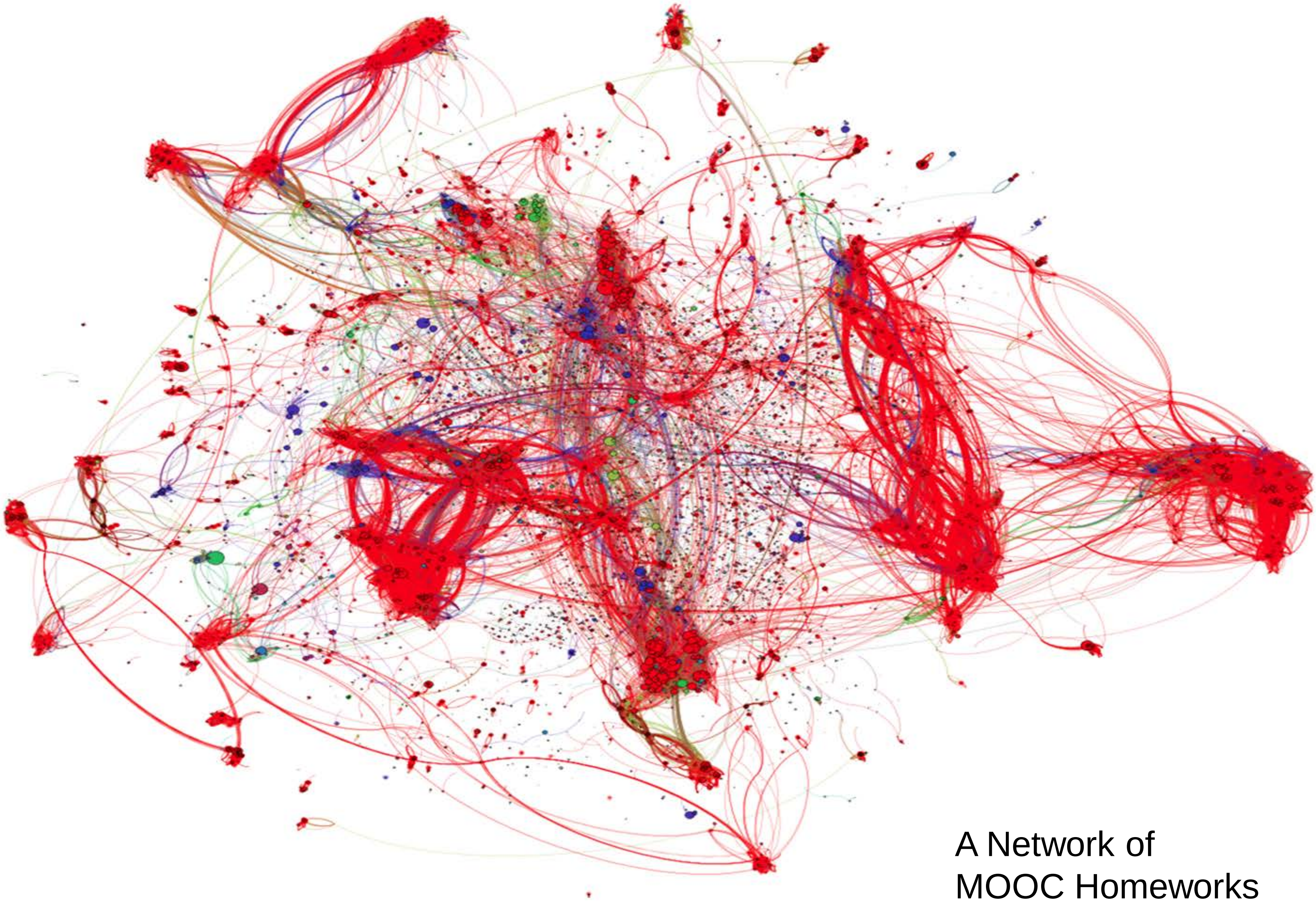
Microsoft

NOKIA Connecting People

Google

National Science Foundation WHERE DISCOVERIES BEGIN

DARPA



A Network of
MOOC Homeworks