Networks of Shapes and Images

Leonidas Guibas
Stanford University

October 2014
Networks of Images
As we acquire more and more data, our data sets become increasingly interconnected and inter-related, because

• we capture information about the same objects in the world multiple times, or data about multiple instances of an object

• natural and human design often exploits the re-use of certain elements, giving rise to repetitions and symmetries

• objects are naturally organized into classes or categories exhibiting various degrees of similarity

Data sets are often best understood not in isolation, but in the context provided by other related data sets.
Relations Between Visual Data
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
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
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 beliefs, opinions, etc
- Operators on functions
  - Maps of functions to functions
    - Laplace-Beltrami operator on a manifold $M$

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

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

Heat diffusion

Laplace Beltrami eigenfunctions
Functional Maps
(a.k.a. Operators)

[M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, L. G., Siggraph ’12]
Starting from a Regular Map $\varphi$

$\varphi$: lion $\rightarrow$ cat
Attribute Transfer via Pull-Back

$T_\phi: \text{cat} \rightarrow \text{lion}$
Functional Map Representation

**Definition**

For a fixed choice of basis functions \( \{ \phi^M \} \) and \( \{ \phi^N \} \), and a bijection \( T : M \to N \), define its **functional representation** as a matrix \( C \), s.t. for all \( f = \sum_i a_i \phi_i^M \), if \( T_F(f) = \sum_i b_i \phi_i^N \) then:

\[
b = Ca
\]

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

\[
C_{ij} = \langle T_F(\phi_i^M), \phi_j^N \rangle.
\]
The Operator View of Maps

from cat to lion

$F$ is a linear operator (matrix) $F : L^2(\text{cat}) \rightarrow L^2(\text{lion})$

Functions on cat are transferred to lion using $F$
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.
Estimating the Mapping Matrix

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

$$Ca \approx b$$

where $f = \sum_i a_i \phi_i^M$, $g = \sum_i b_i \phi_i^N$

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

$$Ca \approx b$$

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 \( S_1 : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(M, \mathbb{R}) \) and \( S_2 : \mathcal{F}(N, \mathbb{R}) \rightarrow \mathcal{F}(N, \mathbb{R}) \).

\[
\begin{align*}
\mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} \mathcal{F}(N, \mathbb{R}) \\
\mathcal{F}(M, \mathbb{R}) & \xrightarrow{C} \mathcal{F}(N, \mathbb{R}) \\
\end{align*}
\]

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.
Lemma 1:
The mapping is *isometric*, if and only if the functional map matrix commutes with the Laplacian:

$$C \Delta_1 = \Delta_2 C$$
Regularization

Lemma 2:

The mapping is *locally volume preserving*, if and only if the functional map matrix is *orthonormal*:

\[ C^T C = I \]
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:

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]
Understanding Intrinsic Distortions

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

Area distortion

Conformal distortion
Classical Approach to Relating Shapes

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

Challenges:
- Point-wise information only, hard to aggregate
- Noisy
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.
A metric is defined by a functional inner product

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

So we can compare \( M \) and \( N \) by comparing \( h^N(F(f), F(g)) \)

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

\[
\int_{\text{lion}} F(f)F(g) \, d\mu_l \neq \int_{\text{cat}} fg \, d\mu_c
\]

Both can be considered as inner products on the cat
There exists a linear operator

$$\mathbf{V} : L^2(\text{cat}) \rightarrow L^2(\text{cat})$$

such that

$$\langle f, g \rangle_{\text{after}} = \langle f, \mathbf{V}(g) \rangle_{\text{before}}$$
Area-Based Shape Difference:

\[ V \approx F^T F \]

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

\[ \int_{lion} F(f)F(g) = \int_{cat} fV(g) \]
A Small Example of V

\[ \int_N F(f)F(g) = \int_M fV(g) \]
Consider a different inner-product of functions ... get information about conformal distortion

\[ \int_{N} \nabla F(f) \nabla F(g) = \int_{M} \nabla f \nabla R(g) \]

The choice of inner product should be driven by the application at hand.
Shape Differences in Collections
Comparing Differences I

\[ D_1 \xrightarrow{F_1} M \xrightarrow{F_2} D_2 \]

\[ N_1 \xrightarrow{\cdots} \]

\[ N_2 \]
Intrinsic Shape Space

Area

Conformal

$D_2$ $M$ $F_1$ $N_1$

$D_1$

$F_2$

$N_2$

\ldots
Intrinsic Shape Space
Localized Comparisons

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

supported in ROI

$D_1 \rho$ to $D_2 \rho$
Exaggeration of Difference in RoI

Rest | ROI | Output: magnified distortion at 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

Input

Entire SCAPE

Output
Shape Analogies

A B A

C D

B

output

A B

C D

output
Comparing Differences III

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

\[ \text{Spec}(D_{M,N}) \sim \text{Spec}(D_{P,Q}) \]
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.
The Network View
Map Networks for Related Data

Maps vs. similarities

Networks of “samenesses”
A Functorial View of Data

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
Cycle-Consistency $\equiv$ 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

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

Conversely, such a low-rank condition can be used to regularize functional maps.
Maps vs. Distances/Similarities
Networks vs. Graphs
Exploitation of the Wisdom in a Collection
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:

  ![Car Images](image1.png) ![Car Images](image2.png) ![Car Images](image3.png) ![Car Images](image4.png)

  ✧ similar objects of the same class:

  ![Cow Images](image5.png) ![Cow Images](image6.png) ![Cow Images](image7.png) ![Cow Images](image8.png)

✧ 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) 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
  
  $$f = \sum_{j=1}^{M} f_j b_j^i = B_i f$$

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

Binary indicator function  Reconstructed function  Thresholded reconstructed function

Reconstruction error vs. Number of Bases
Functional map:

- A linear map between functions in two functional spaces
  \[ f' = X_{ij} f \quad X_{ij} \in \mathcal{R}^{M \times M} \]

- Can be recovered by a set of probe functions

Joint Estimation of Functional Maps,
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:**
  \[ f_{ij}^{\text{reg}} = \| X_{ij} \Lambda_i - \Lambda_j X_{ij} \|^2 \]

- Correspond bases of similar spectra
- Enforce sparsity of map

\( \Lambda_i, \Lambda_j \) diagonal matrices of Laplacian eigenvalues
Incorporating map cycle consistency:

A transported function along any loop should be identical to the original function:

\[
X_{i_k i_0} \cdots X_{i_1 i_2} X_{i_0 i_1} f = f \quad \iff \quad X_{i_j} Y_i = Y_j, \quad \forall (i, j) \in \mathcal{G}
\]

Consistency term:

\[
f_{\text{cons}} = \sum_{(i,j) \in \mathcal{G}} w_{i,j} f_{i,j} \quad \text{cons} = \sum_{(i,j) \in \mathcal{G}} w_{i,j} \|X_{i_j} Y_i - Y_j\|_2^2
\]

Image global similarity weight via GIST
Joint Estimation of Functional Maps, III

- Plato’s allegory of the cave

\[ X_{ij} \approx Y_j^{-1} Y_i \]

\( X \, 30\times30, \, Y \, 30\times20 \)
Joint Estimation of Functional Maps, IV

Overall optimization

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

Alternating optimization:

- Fix Y, solve X \(\iff\) Independent QP problems
  \[
  X_{ij}^* = \arg \min_X \left( f_{ij}^{\text{feature}} + \mu f_{ij}^{\text{reg}} + \lambda f_{ij}^{\text{cons}} \right)
  \]

- Fix X, solve Y \(\iff\) Eigenvalue problem
  \[
  \min \ \text{trace}(Y^T W Y) 
  \]
  \[
  \text{s.t. } Y^T Y = I_m 
  \]

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

Source image

Target image

Without cycle consistency

With cycle consistency
Generating Consistent Segments

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

\[ f_{\text{map}} = \sum_{(i,j) \in G} w_{ij} \|X_{ij} f_i - f_j\|^2 \]

- Agreement with normalized cut scores:

\[ f_{\text{seg}} = \sum_{i=1}^N f_i^T B_i^T L_i B_i f_i \]

We look for network fixed points!

Joint optimization:

\[
\min f_{\text{seg}} + \gamma f_{\text{map}} \quad \text{s.t.} \quad \sum_{i=1}^N \|f_i\|^2 = 1
\]
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

### iCoseg data set

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

#### Kuettel'12 (Supervised)

<table>
<thead>
<tr>
<th>Class</th>
<th>Image+transfer</th>
<th>Full model</th>
<th>Unsupervised Fmaps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alaska Bear</td>
<td>87.6</td>
<td>91.4</td>
<td>90.5</td>
</tr>
<tr>
<td>Red Sox Players</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stonehenge1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stonehenge2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Liverpool FC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ferrari</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taj Mahal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elephants</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pandas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kite panda</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gymnastics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skating</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot Balloons</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Liberty Statue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brown Bear</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MSRC

#### Unsupervised performance comparison

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

#### Supervised performance comparison

<table>
<thead>
<tr>
<th>Class</th>
<th>Vicente '11</th>
<th>Kuettel '12</th>
<th>Fmaps -s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cow</td>
<td>94.2</td>
<td>92.5</td>
<td>94.3</td>
</tr>
<tr>
<td>Plane</td>
<td>83.0</td>
<td>86.5</td>
<td>91.0</td>
</tr>
<tr>
<td>Car</td>
<td>79.6</td>
<td>88.8</td>
<td>83.1</td>
</tr>
<tr>
<td>Sheep</td>
<td>94.0</td>
<td>91.8</td>
<td>95.6</td>
</tr>
<tr>
<td>Bird</td>
<td>95.3</td>
<td>93.4</td>
<td>95.8</td>
</tr>
<tr>
<td>Cat</td>
<td>92.3</td>
<td>92.6</td>
<td>94.5</td>
</tr>
<tr>
<td>Dog</td>
<td>93.0</td>
<td>87.8</td>
<td>91.3</td>
</tr>
</tbody>
</table>

### PASCAL

#### 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:

- Map network
- Consistent (apple bucket)
- Consistent (girl in pink)
- Inconsistent (cycle not closed)

Must deal with non-total maps

Related to topological persistence / persistent homology
Consistent Functional Maps

- Latent functions: \( \mathbf{Y}_i = (y_{i1}, \ldots, y_{iL}) \)
- Discrete variables: \( z_i = \{z_{il} \in \{0, 1\}, 1 \leq l \leq L\} \)
- Relationship: \( \mathbf{Y}_i \text{Diag}(z_i) = \mathbf{Y}_i \)
- Consistency: \( \mathbf{X}_{ij} \mathbf{Y}_i = \mathbf{Y}_j \text{Diag}(z_i), \quad (i, j) \in \mathcal{E}. \)
Consistent Functional Maps

The consistency regularization

\[ f_{\text{cons}} = \mu \sum_{(i,j) \in \mathcal{E}} \|X_{ij}Y_i - Y_j \text{Diag}(z_i)\|^2 \]

\[ + \gamma \sum_{i=1}^{N} \|Y_i - Y_i \text{Diag}(z_i)\|^2, \]

Overall optimization

\[ \{X_{ij}^*\} = \arg\min_{X_{ij}} \left( \mu f_{\text{cons}} + \sum_{(i,j) \in \mathcal{E}} f_{\text{pair}} \right) \]
Framework
Initialization

- Solve for consistent segmentation with ALL images together

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

\[ = s_k^T L s_k, \]

- Pick the first M eigenvectors

- Each object class is initialized as:

\[ C_k = \{ i, \text{ s.t. } \|s_{ik}\| \geq \max_i \|s_i\|/2 \} \]
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

\[
\begin{align*}
\min_{s_{ik}, i \in C_k} & \quad \sum_{k=1}^{M} \sum_{(i,j) \in \mathcal{E} \cap (C_k \times C_k)} \|X_{ij}s_{ik} - s_{jk}\|^2 \\
+ & \quad \gamma \sum_{l \neq k} \sum_{i \in C_k \cap C_l} (s_{il}^Ts_{ik})^2 + \mu \sum_{k=1}^{M} \sum_{i \in C_k} s_{ik}^T L_i s_{ik} \\
\text{subject to} & \quad \sum_{i \in C_k} \|s_{ik}\|^2 = |C_k|, \quad 1 \leq k \leq K.
\end{align*}
\]
Expand each object class by propagating segmentations to other images

\[
\max_{s_{ik}} \frac{1}{|\mathcal{N}(i) \cap C_k|} \sum_{j \in \mathcal{N}(i) \cap C_k} (s_{ik}^T X_{ji} s_{jk})^2 \\
- \gamma \sum_{l \neq k, i \in C_l} (s_{ik}^T s_{il})^2 - \mu s_{ik}^T L_i s_{ik}
\]

subject to \( \|s_{ik}\|^2 = 1 \)
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

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

### Performance comparison on the MFCFlickr dataset

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

### Performance comparison on the PASCAL-multi dataset

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


Butterfly + blossom: green: butterfly in orange; yellow: butterfly in yellow; cyan: red flower
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/
The Network is the Abstraction
Abstractions Emerge from the Network

(Approximately) Cycle-Consistent Diagram
Abstraction – Colimit

Colimits glue parts together to make a whole

\[ \lim \bigcup_i F_i = \bigcup_i F_i / \sim \]
Abstraction – Approximate Colimit

Find projections that “play well” with maps on network edges,

\[
\lim_{\longrightarrow} \mathcal{F}_i = \bigsqcup_i \mathcal{F}_i \mathcal{F}_4 \sim \mathcal{F}_5 \mathcal{F}_6
\]

“Colimit” = Latent space = Abstraction

\[
Y_j X_{ij} \approx Y_i \quad \text{or} \quad X_{ij} \approx Y_j^+ Y_i
\]
The Network is the Abstraction

a co-limit

\[ Y_i \]

\[ Y_j \]

\[ X_{ij} \]
Consistent Shape Segmentation

[Q. Huang, F. Wang, L. Guibas, ’14]
First Build a Network

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

\[ C_i X_{ij} \approx D_j \]
Algebraic Dependencies Between Maps

Cycle consistency or closure

consistent cycles

inconsistent cycles
The Pipeline

Step 1
Original shapes with noisy maps
Cleaned up maps

Step 2
Consistent basis functions extracted
Joint Map Optimization

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

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

The $Y_i$ give us the desired latent spaces

Dual ADMM

Step 2: Perturb the above $X$ to force the factorization

$$\sum_{1 \leq i,j \leq N} \| X_{ij}^* - Y_j^+ Y_i \|_F^2 + \mu \sum_{i=1}^N \sum_{1 \leq k < l \leq L} (y_{ik}^T y_{il})^2$$

Non-linear least squares

Gauss-Newton descent
Consistent Shape Segmentation

Via 2\textsuperscript{nd} order MRF on each shape independently
Networks of Shapes and Images
Depth Inference from a Single Image
Input Image  
Kinect Scan  
Depth Recovery
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

- Functions over data
- Maps between data
- Networks of data sets
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: