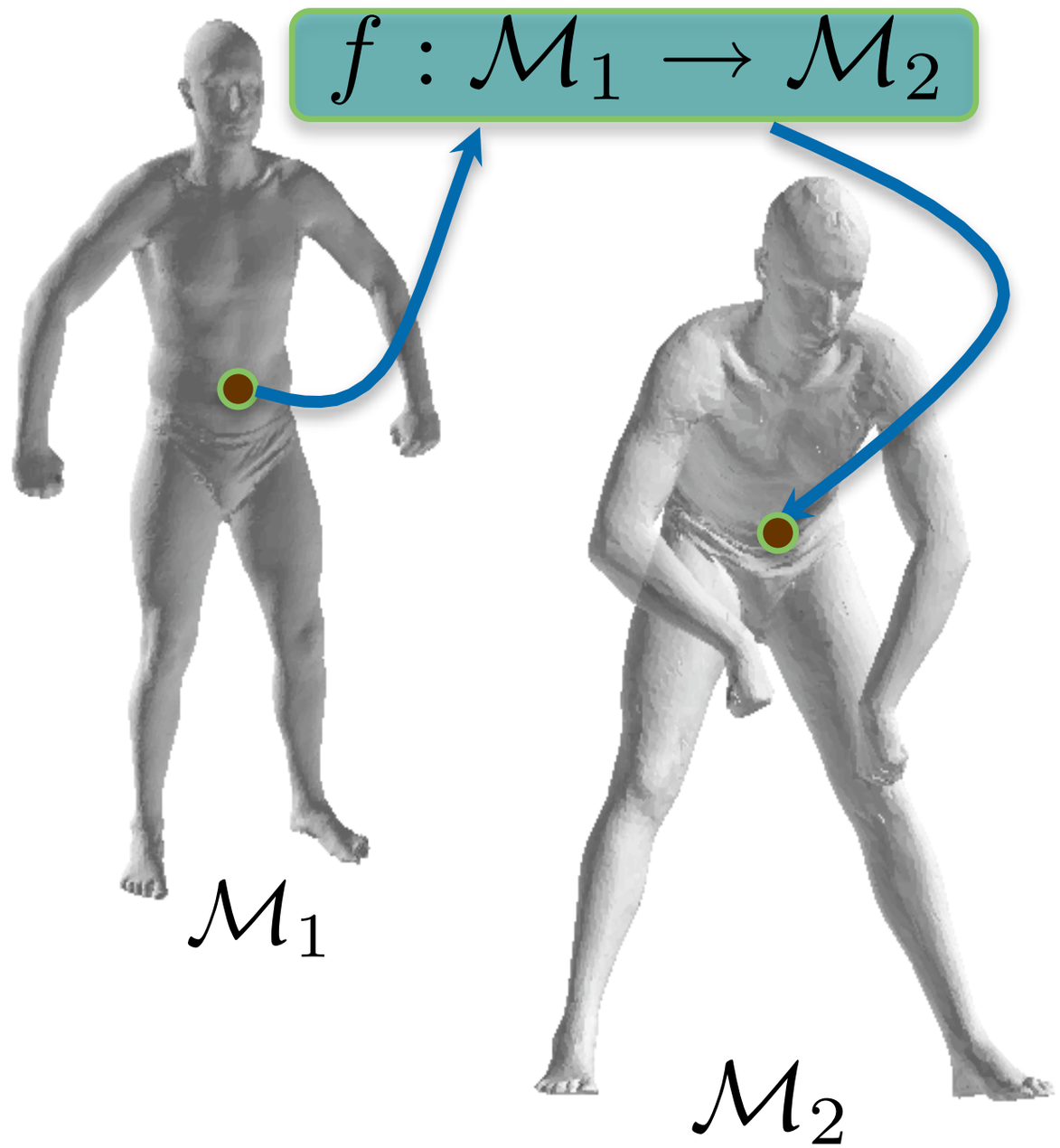
CS 468
Data-driven Shape Analysis

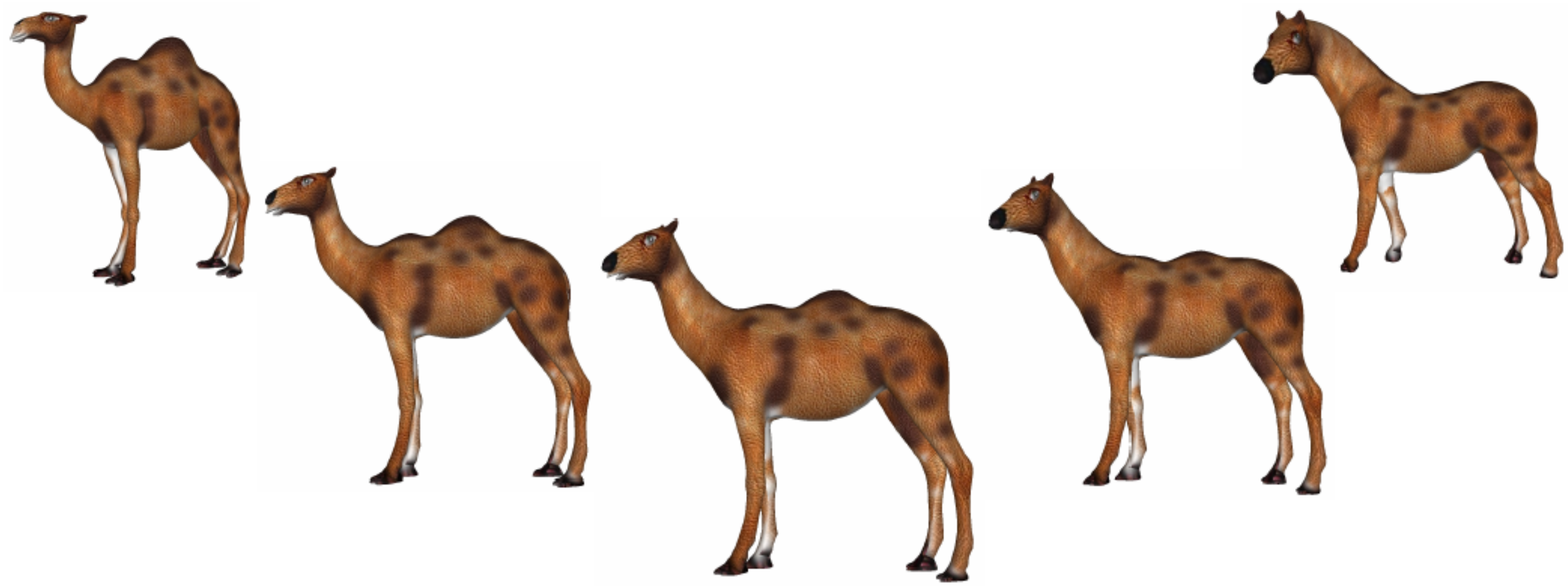
Intrinsic Maps

April 15, 2014

Inter-surface Map

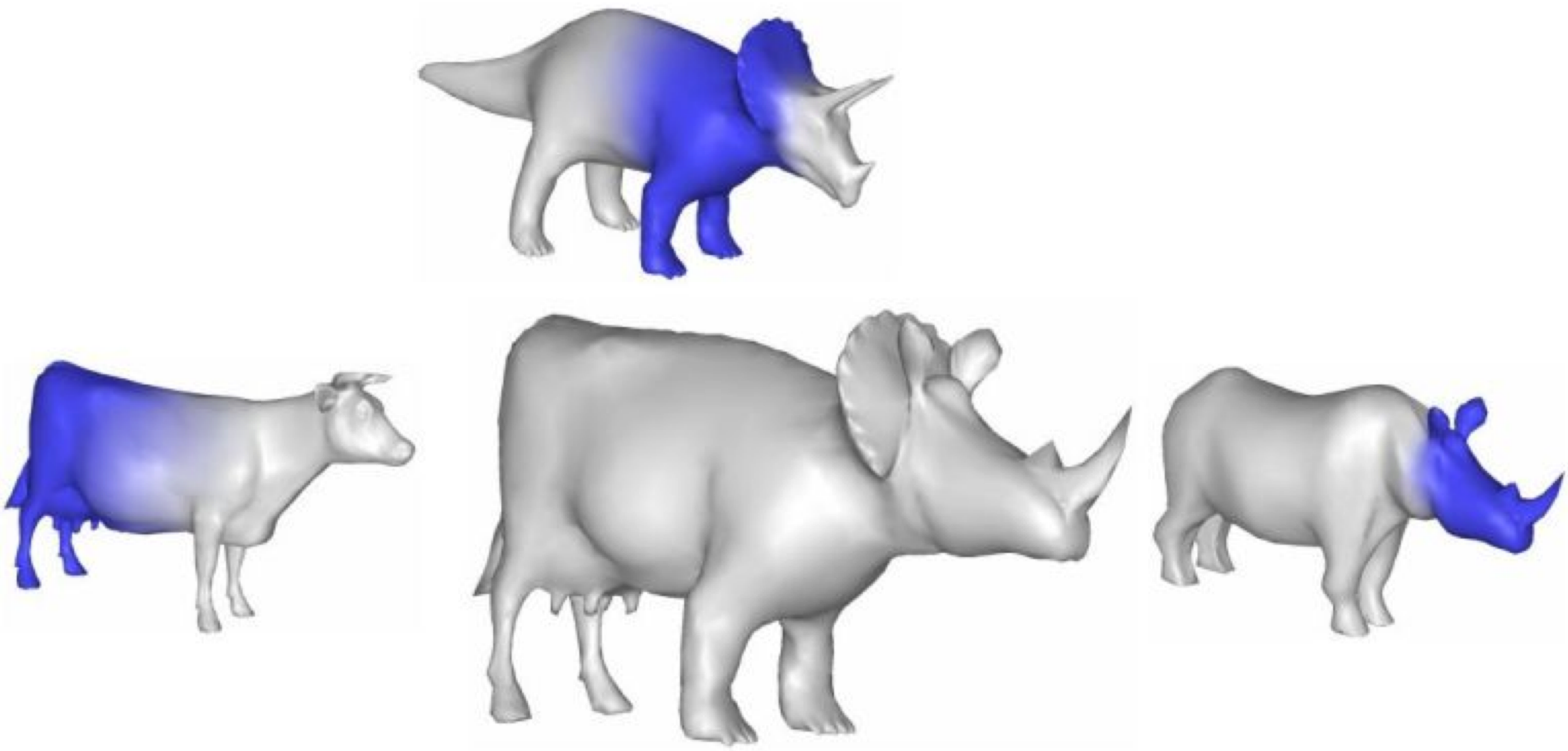


Applications



Kraevoy and Sheffer 2004

Applications



Kraevoy and Sheffer 2004

Desired Properties

Given two (or more) shapes find a map f, that is:

- Automatic
- Fast to compute
- Bijective (if we expect to have a global correspondence)
- Low-distortion

Desired Properties

Given two (or more) shapes find a map f, that is:

- ✘ **Automatic**
- Fast to compute
- Bijective (if we expect to have a global correspondence)
- Low-distortion

Consider a simple case…

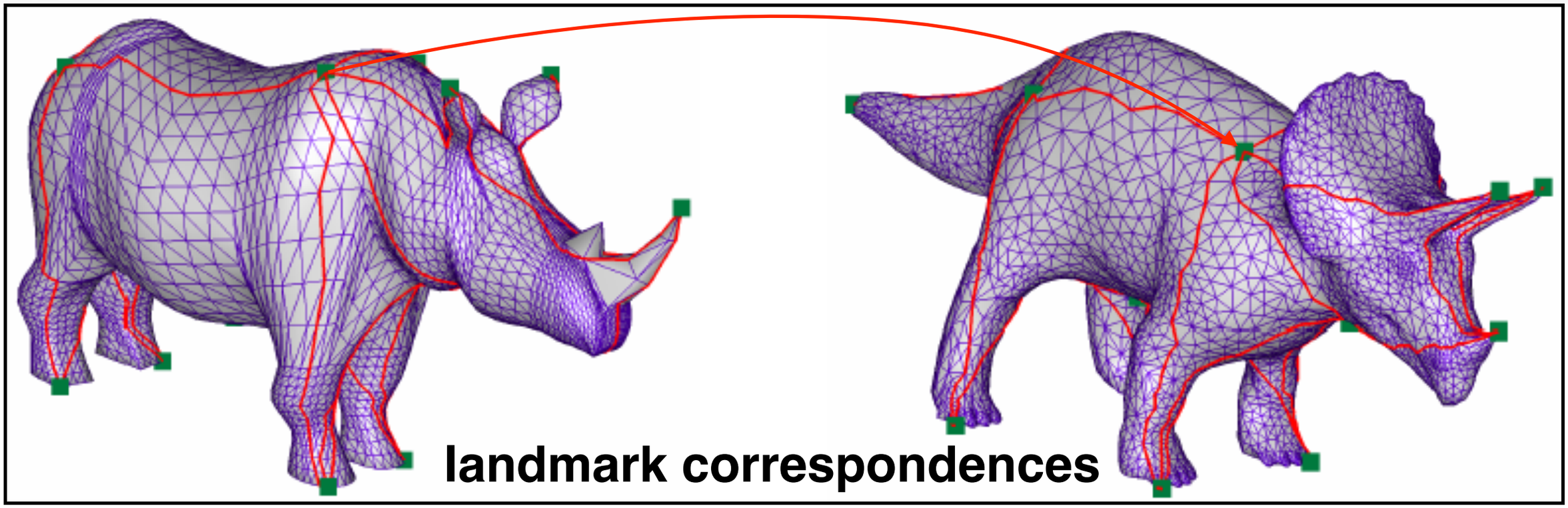
Desired Properties

Given two (or more) shapes find a map f, that is:

- ✘ **Automatic**
- ✘ **Fast to compute**
- ✓ Bijective (if we expect to have a global correspondence)
- ✘ **Low-distortion**

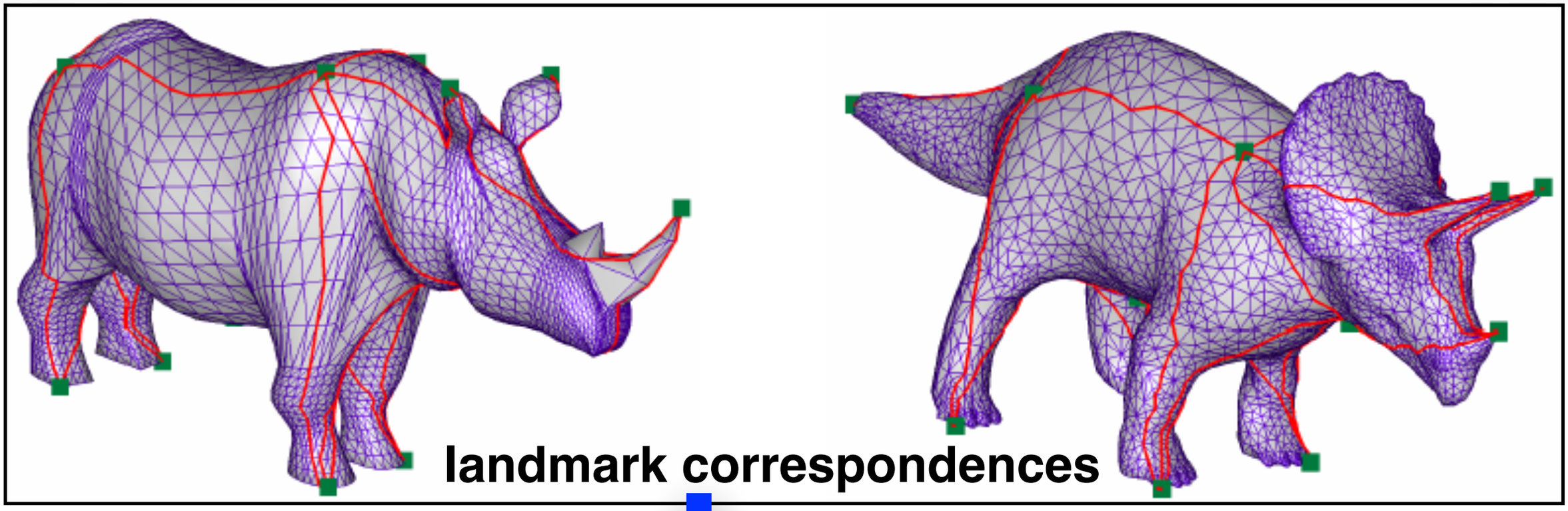
Consider a simple case…

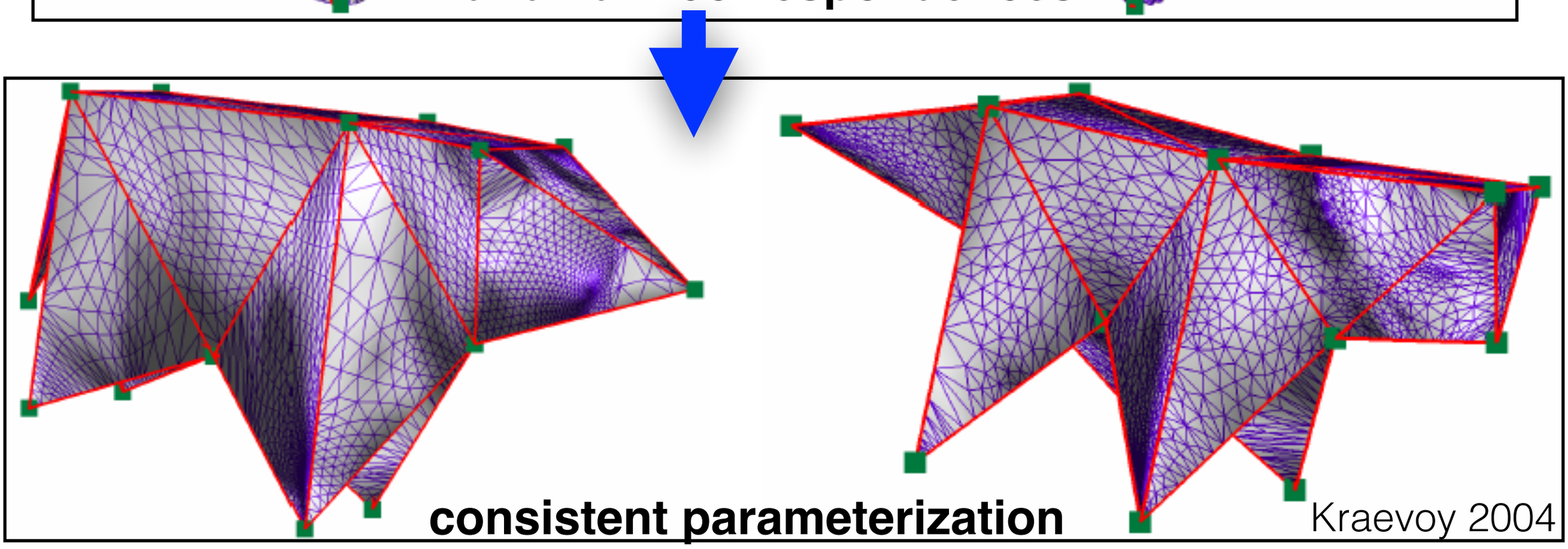
Consistent Re-meshing



Kraevoy 2004

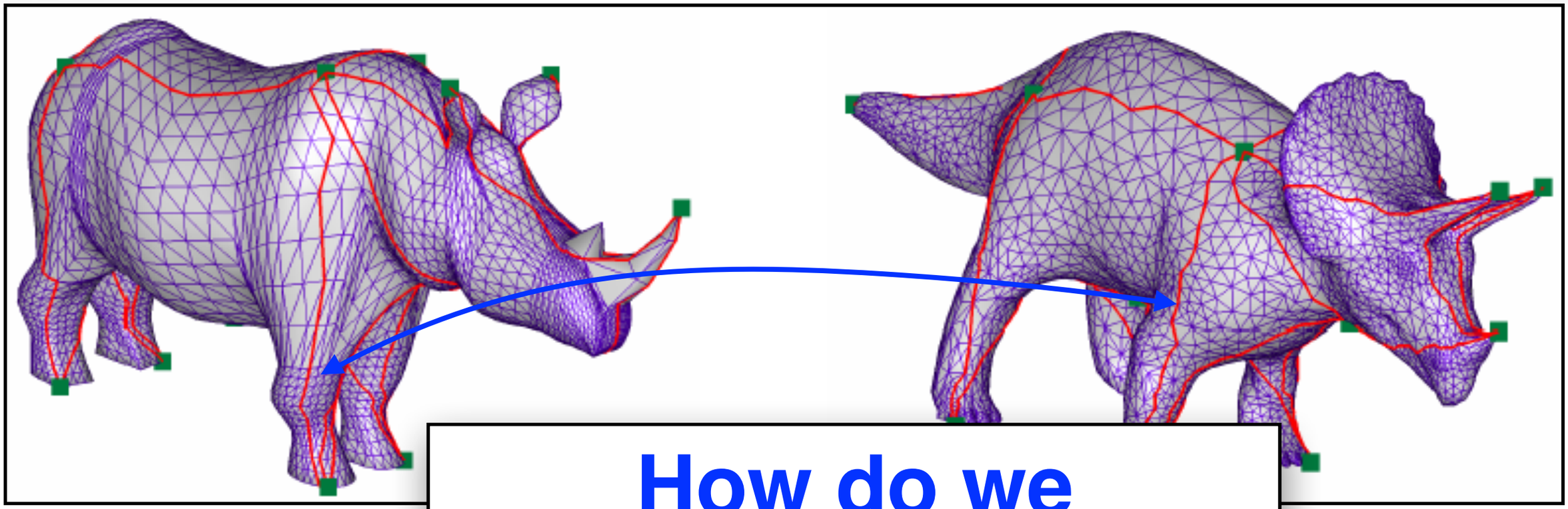
Consistent Re-meshing



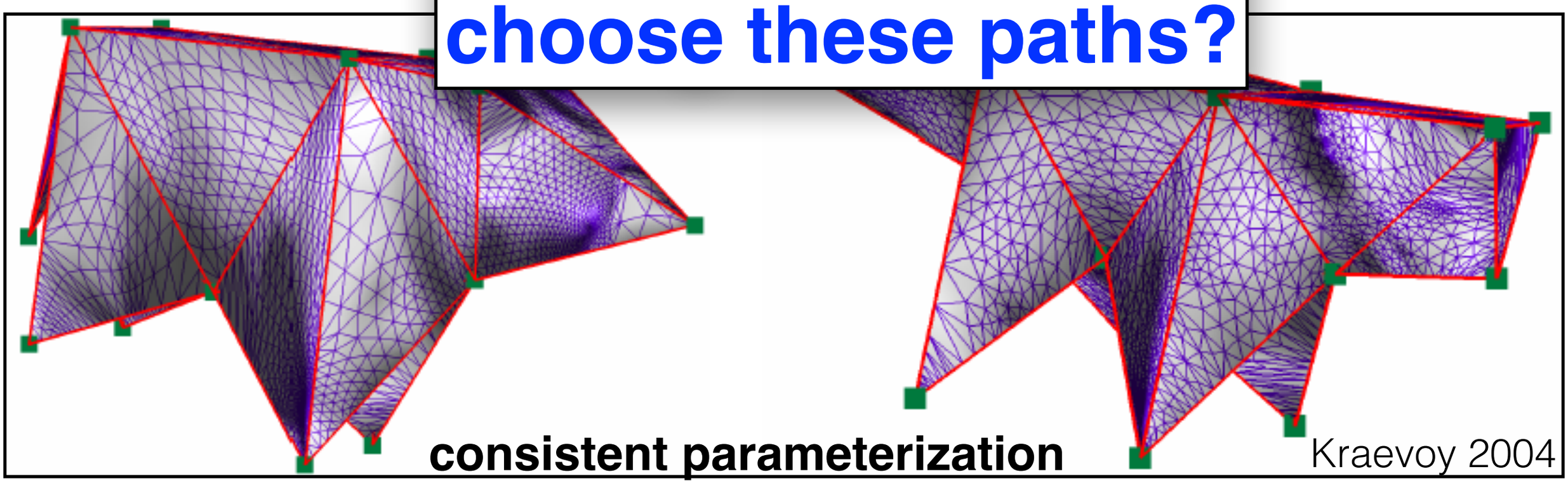


Kraevoy 2004

Consistent Re-meshing



How do we choose these paths?



consistent parameterization

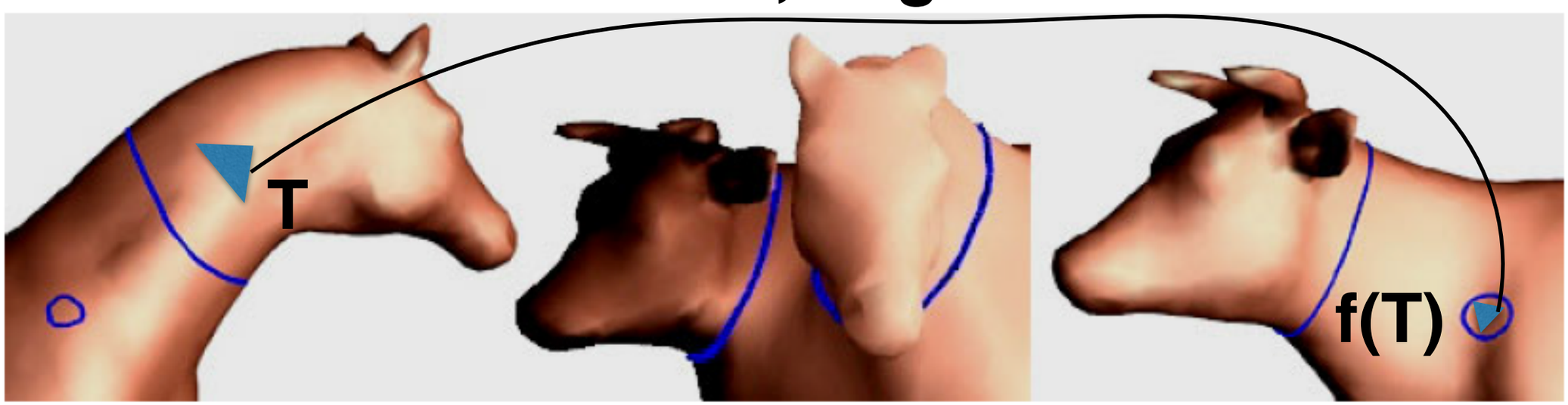
Kraevoy 2004

Distortion Metrics

Compare triangles T and f(T)

- Angles (conformal map)
- Areas
- Stretch

E.g. small conformal distortion, large area distortion:



Schreiner et al. 2004

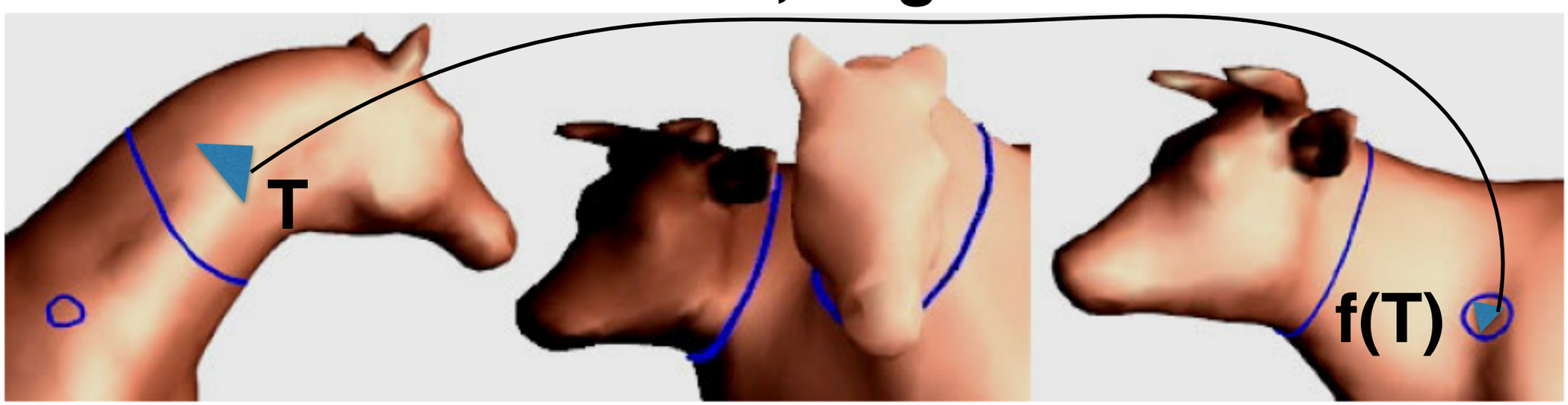
Distortion Metrics

Compare triangles T and f(T)

- Angles (conformal map)
- Areas
- Stretch

NOTE: isometry preserves all

E.g. small conformal distortion, large area distortion:



Schreiner et al. 2004

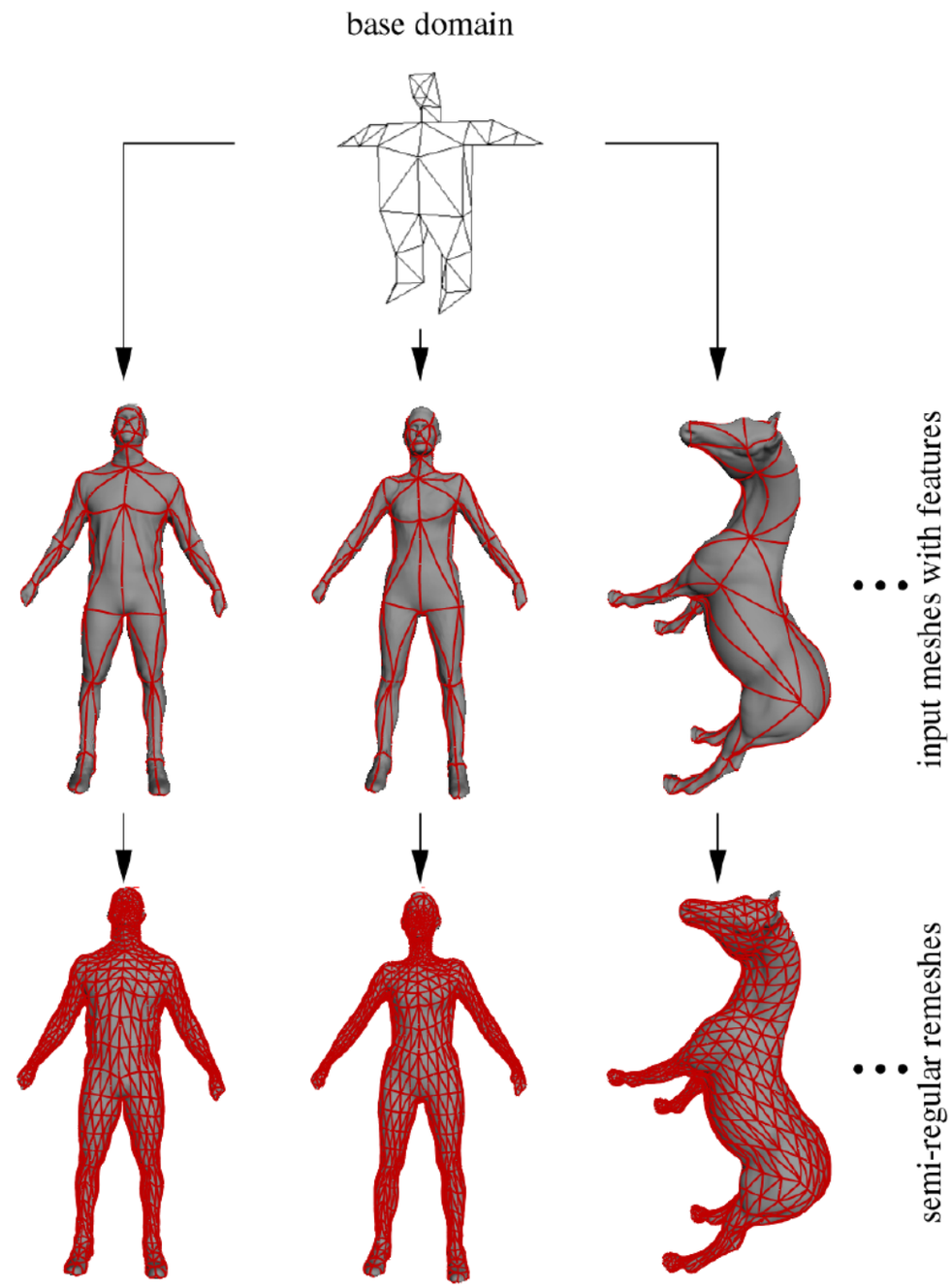
Pros and Cons

Pros:

- Apps!

Cons:

- Need many manual landmark points
- Hard to minimize the distortion



Praun et al. 2001

Automatic Landmarks

Consider an algorithm:

- Set landmark correspondences
- Measure energy
- Repeat and return minimal energy

Automatic Landmarks

Consider an algorithm:

- Set landmark correspondences
- Measure energy
- Repeat and return minimal energy

Problems?

Automatic Landmarks

Consider an algorithm:

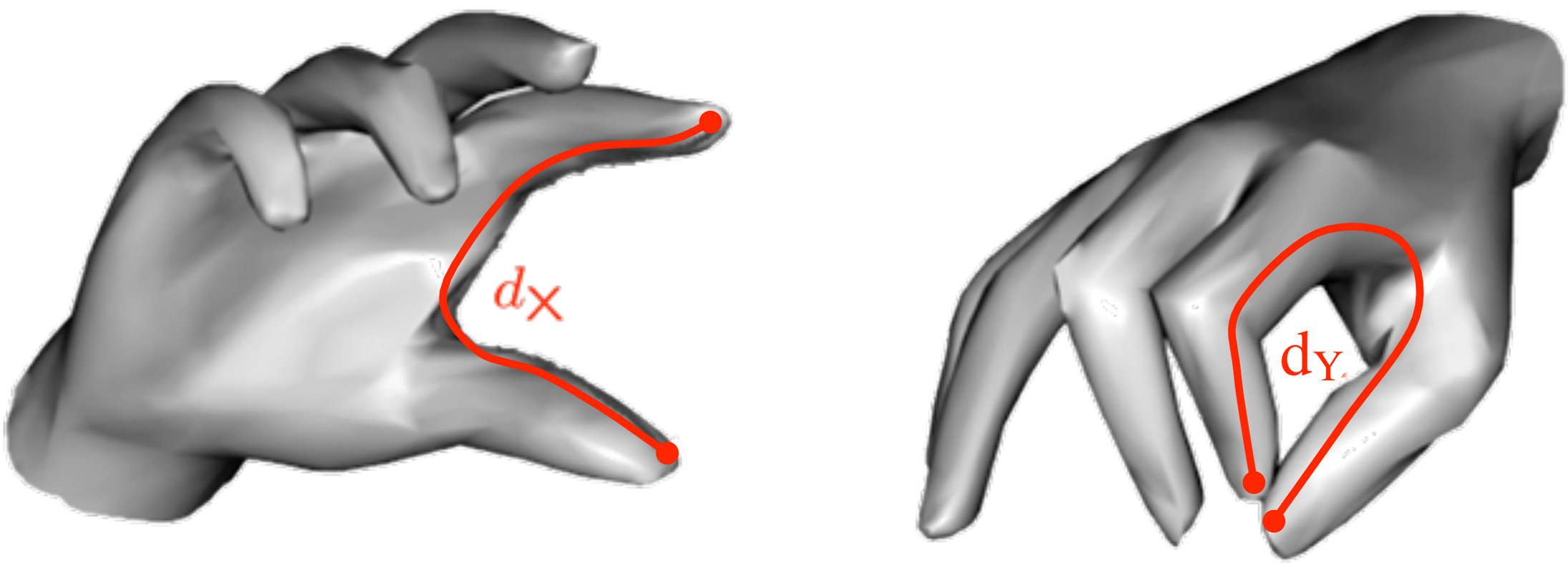
- Set landmark correspondences
- ➡ **Measure energy** ← **Choice of energy greatly affects the results and the optimization**
- Repeat and return minimal energy

Gromov-Hausdorff Distance

Gromov-Hausdorff

Compare shapes as metric spaces

Shape = metric space (X, d)
Invariance = isometry w.r.t. d

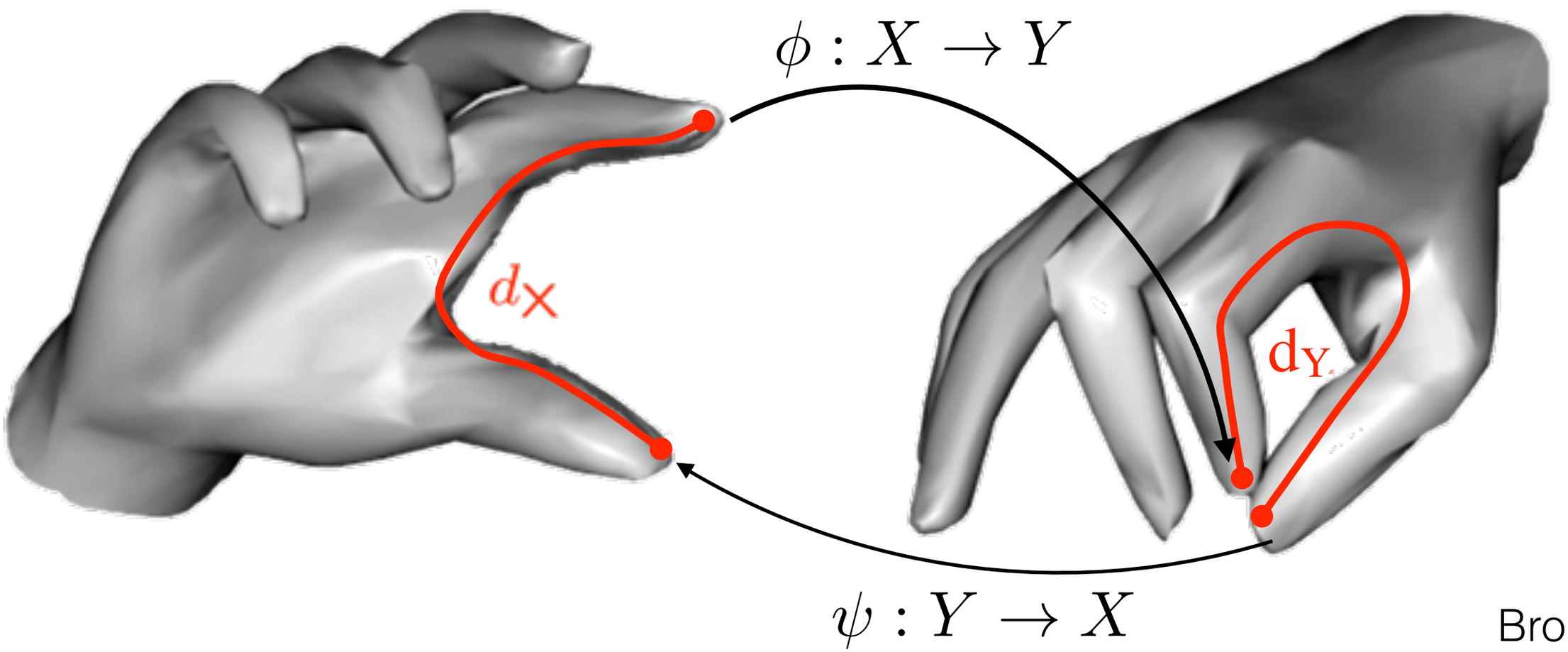


Bronstein

Gromov-Hausdorff

Compare shapes as metric spaces

Shape = metric space (X, d)
Invariance = isometry w.r.t. d



Bronstein

Gromov-Hausdorff

Compare shapes as metric spaces

$$d_{\mathsf{GH}}(\mathsf{X},\mathsf{Y}) = \frac{1}{2} \inf_{\substack{\varphi:\mathsf{X}\to\mathsf{Y} \\ \psi:\mathsf{Y}\to\mathsf{X}}} \max\{\operatorname{dis}\varphi,\ \operatorname{dis}\psi, \operatorname{dis}(\varphi,\psi)\}$$

where:

$$\operatorname{dis}\varphi = \sup_{x,x'\in\mathsf{X}} \left| d_{\mathsf{X}}(x,x') - d_{\mathsf{Y}}(\varphi(x),\varphi(x')) \right|$$

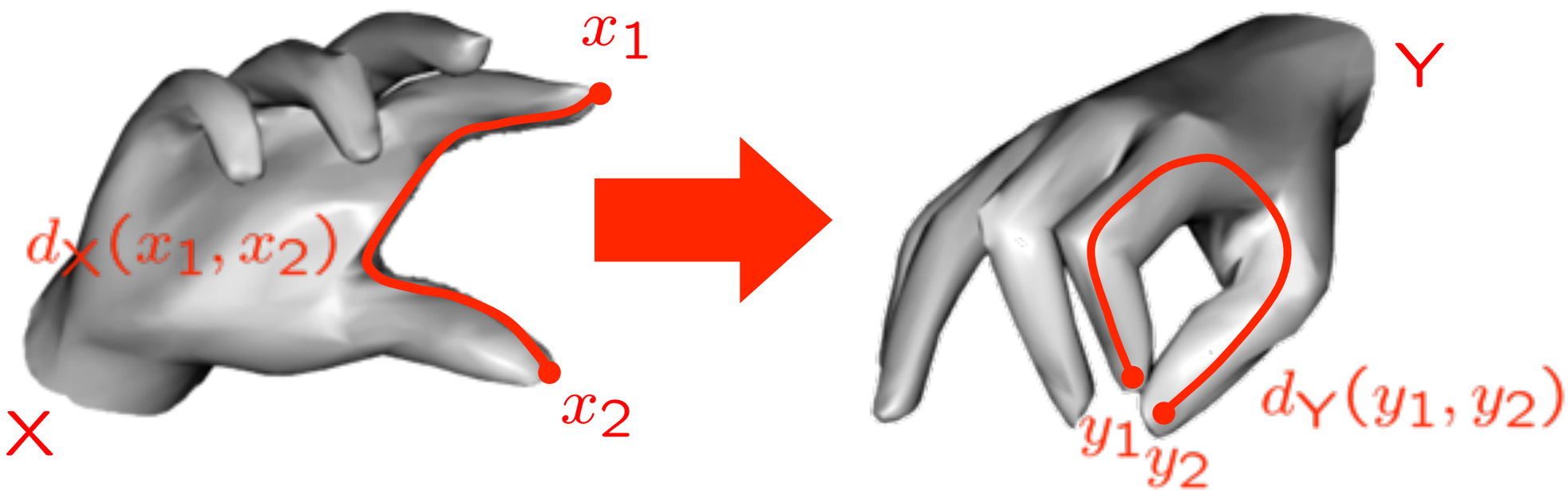
$$\operatorname{dis}\psi = \sup_{y,y'\in\mathsf{Y}} \left| d_{\mathsf{Y}}(y,y') - d_{\mathsf{X}}(\psi(y),\psi(y')) \right|$$

$$\operatorname{dis}(\varphi,\psi) = \sup_{x\in\mathsf{X},\, y\in\mathsf{Y}} \left| d_{\mathsf{X}}(x,\psi(y)) - d_{\mathsf{Y}}(y,\varphi(x)) \right|$$

Bronstein

Generalized MDS

Search for a permutation



Generalized multidimensional scaling (GMDS)

$$d_{\mathrm{int}}(X, Y) = \min_{\{y_1, \ldots y_n\} \subset Y} \left\| d_X(x_i, x_j) - d_Y(y_i, y_j) \right\|$$

A. Bronstein, M. Bronstein, R. Kimmel, *PNAS* 2006, *SIAM JSC* 2006

Bronstein

Pros and Cons

Pros:

- Good distance for non-isometric metric spaces

Cons:

- Non-convex
- HUGE search space (i.e. permutations)

Practice

Heuristics to explore the permutations

➡ **Solve at a very coarse scale => interpolate**

- Coarse-to-fine
- Partial Matching

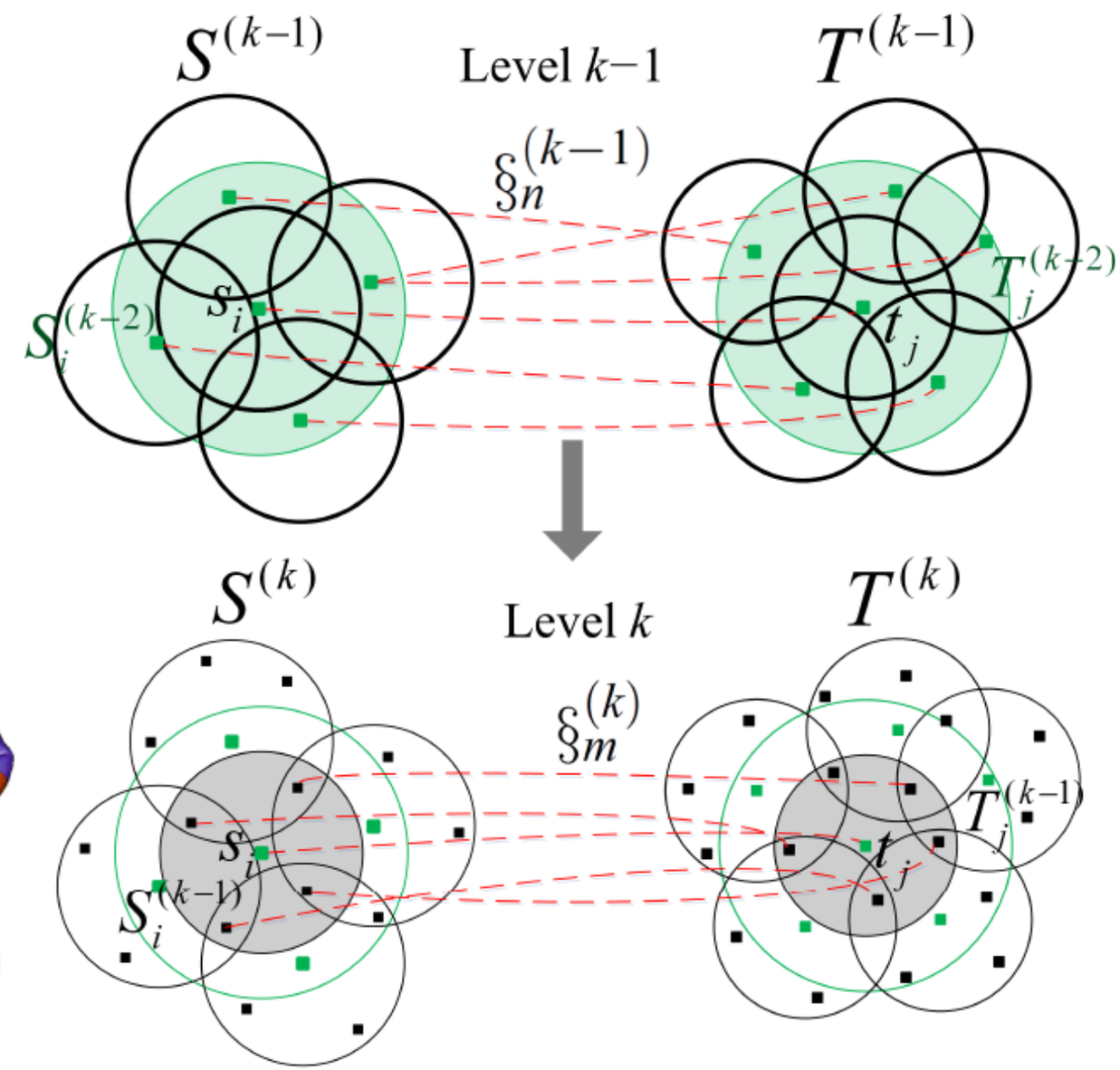
Bronstein'08

Practice

Heuristics to explore the permutations

- Solve at a very coarse scale => interpolate
- ➡ **Coarse-to-fine**
- Partial Matching

Bronstein'08



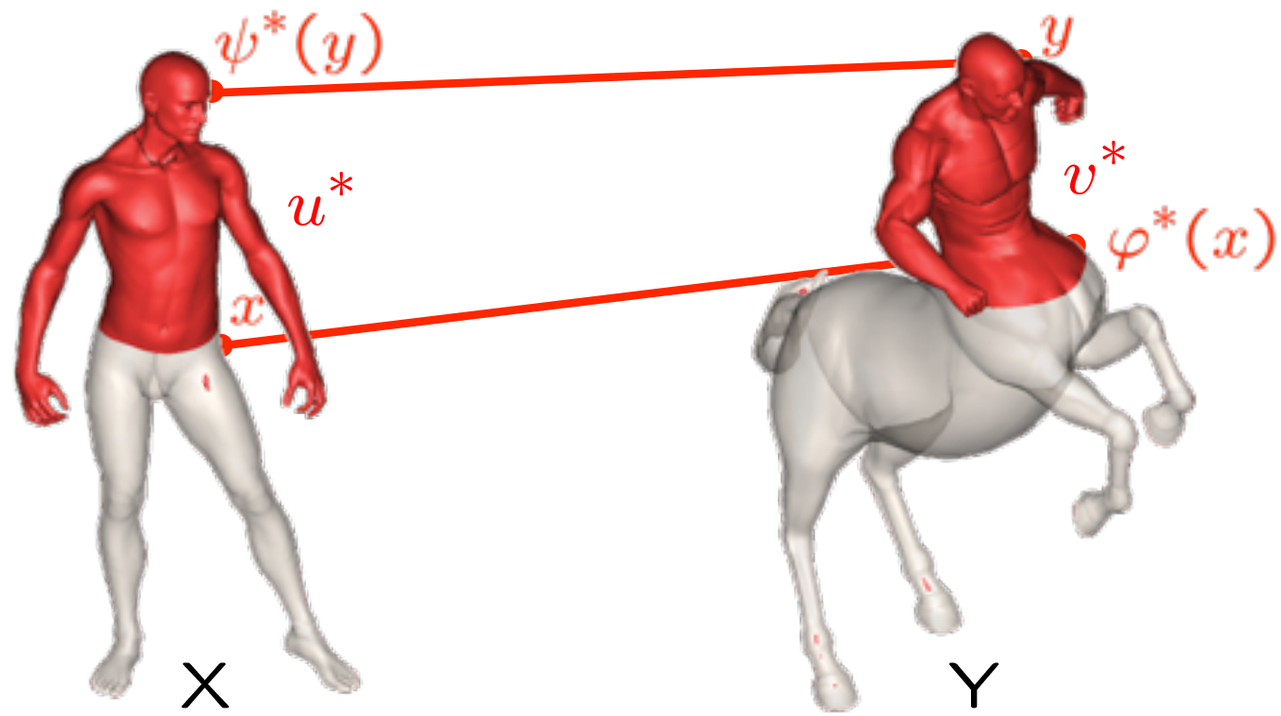
Sahillioglu'12

Practice

Heuristics to explore the permutations

- Solve at a very coarse scale => interpolate
- Coarse-to-fine

➡ **Partial Matching**



- Find correspondence φ^*, ψ^* minimizing distortion between current parts u^*, v^*
- Select parts u^*, v^* minimizing the distortion with current correspondence φ^*, ψ^* subject to $\lambda(u^*, v^*) \leq \lambda_0$

A. Bronstein, M. Bronstein, A. Bruckstein, R. Kimmel, *IJCV* 2008

Bronstein

Properties

Given two (or more) shapes find a map f, that is:

- ✓ Automatic
- ✘ **Fast to compute**
- ✘ **Bijective (if we expect to have a global correspondence)**
- ✓ Low-distortion

Unless failed to find an optima

Proper non-isometry is HARD!

How hard is it to match Isometric Shapes?

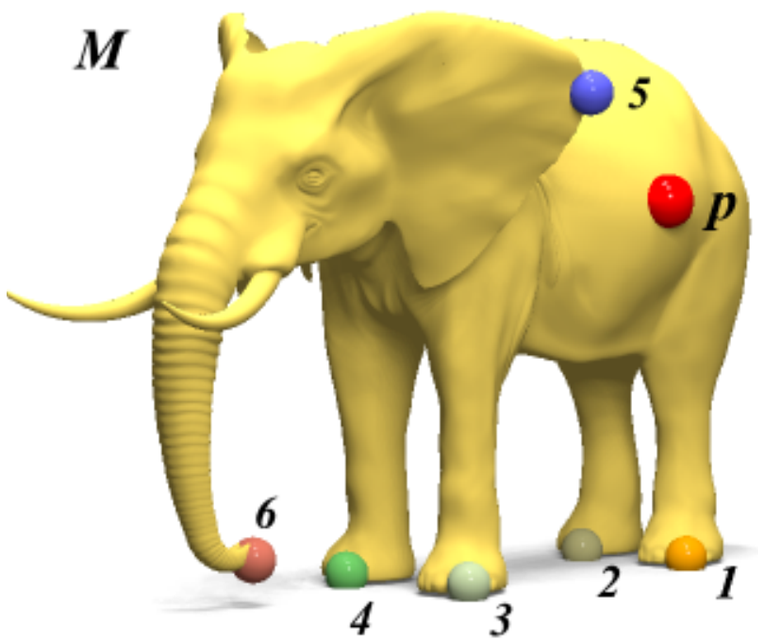
Proper non-isometry is HARD!

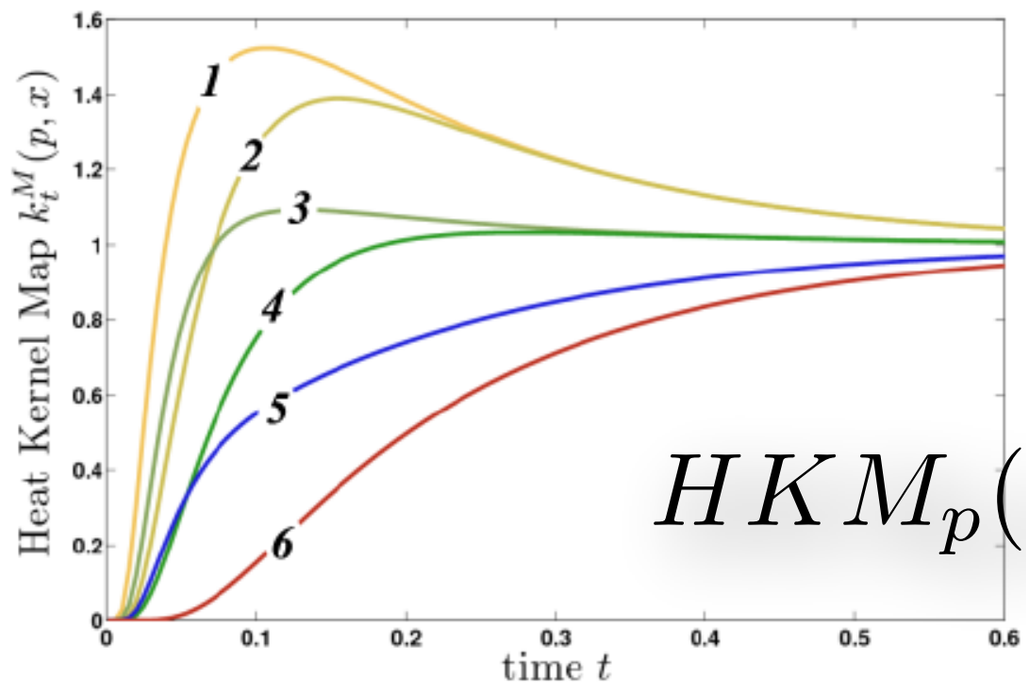
How hard is it to match Isometric Shapes?

E.g. how many point-to-point correspondences do we need to define a map between two isometric shapes?

Heat Kernel Map

Only need to match **one** point!



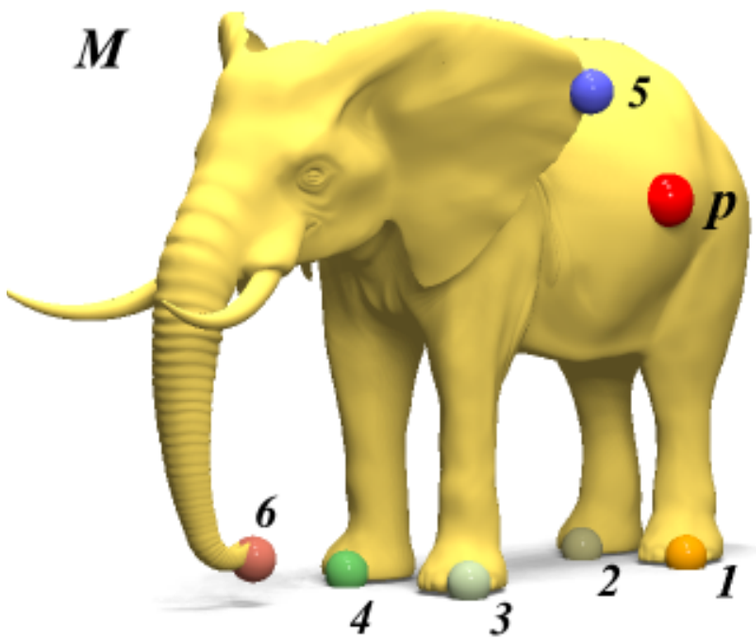


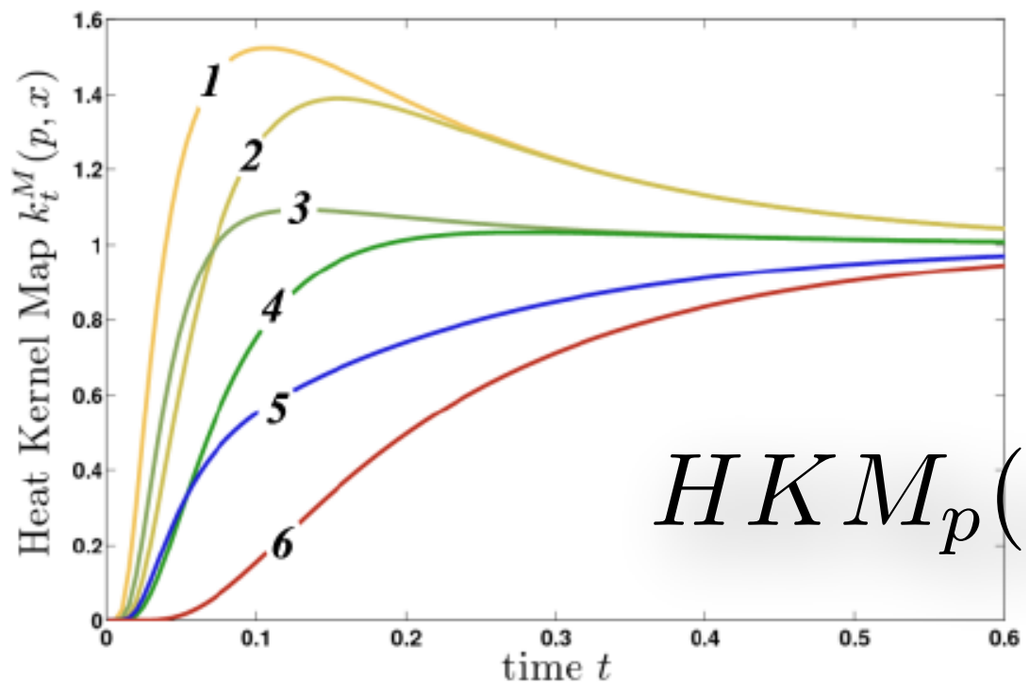
$$HKM_p(x,t) = k_t(p,x)$$

Ovsjanikov'10

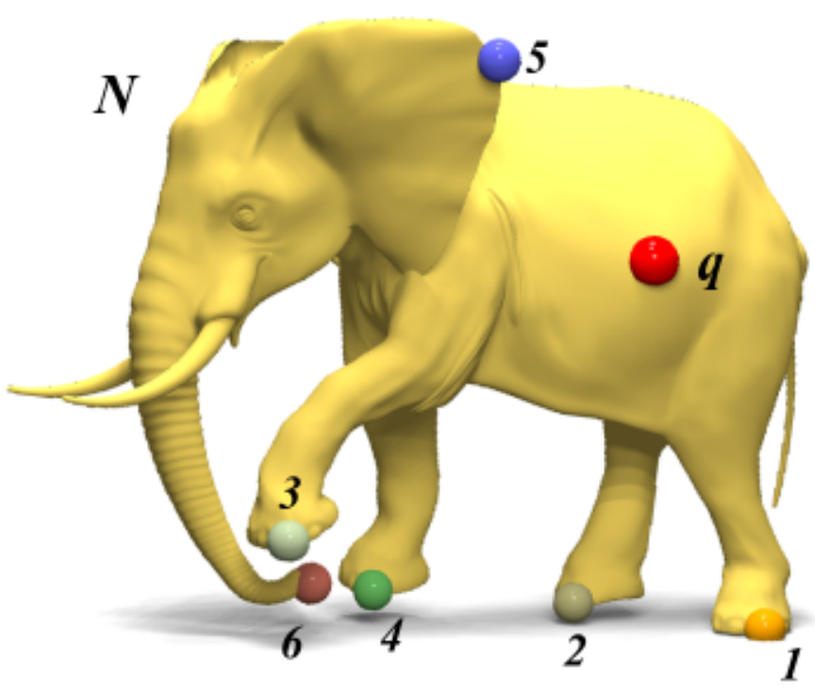
Heat Kernel Map

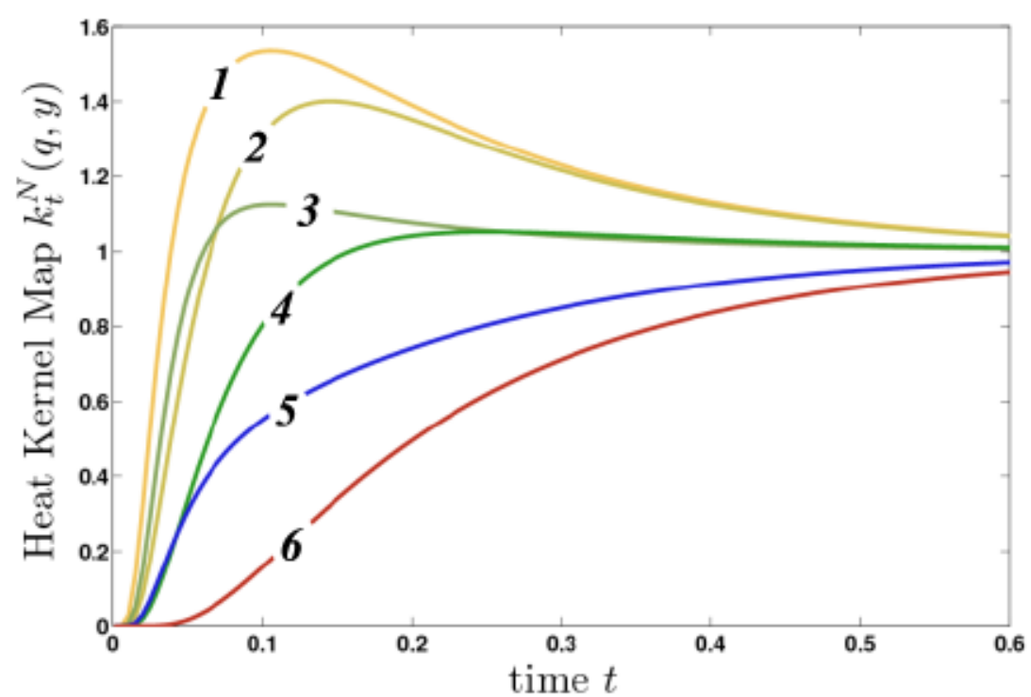
Only need to match **one** point!





$$HKM_p(x,t) = k_t(p,x)$$





Ovsjanikov'10

Heat Kernel Map

Pros:

➡ **The search space is TINY**

- Naturally works in partial case

Cons:

- Sensitive to deviations from isometry

Ovsjanikov'10

Heat Kernel Map

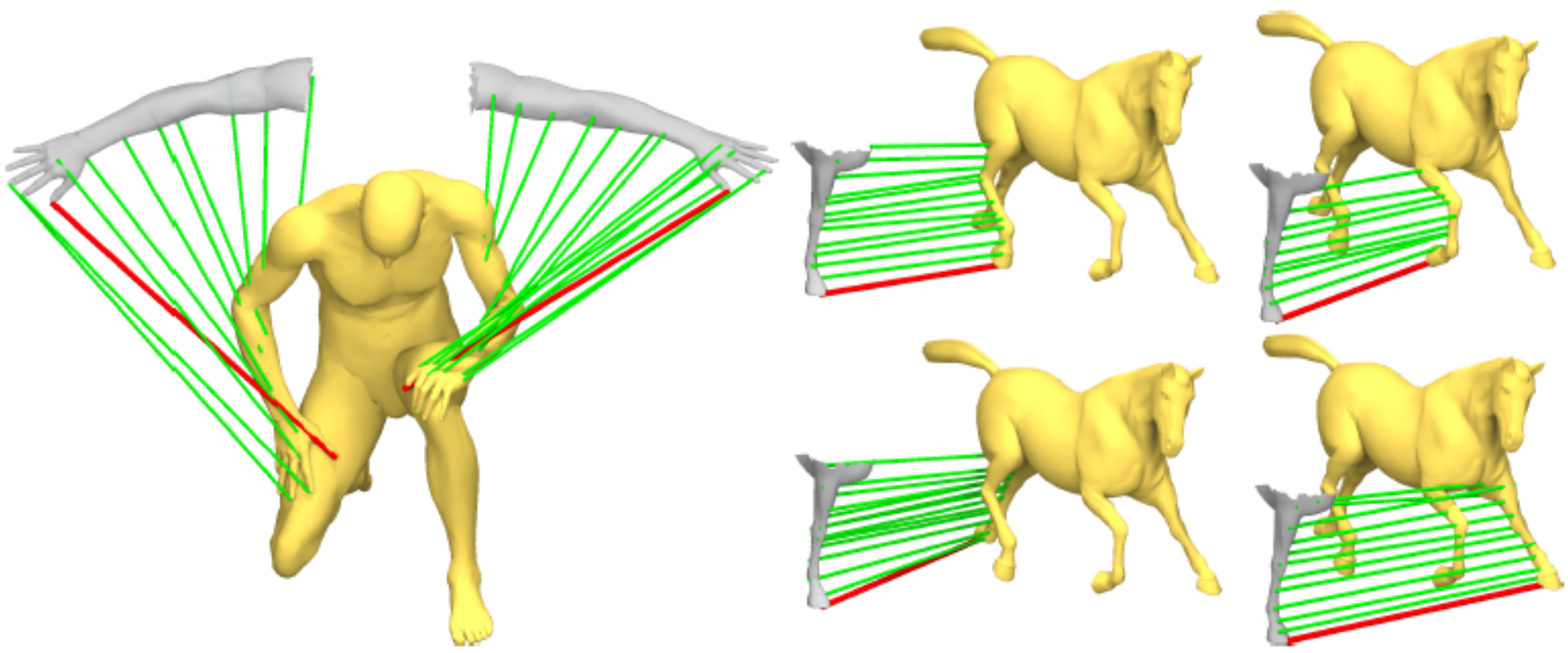
Pros:

- The search space is TINY

➡ **Naturally works in partial case**

Cons:

- Sensitive to deviations from isometry



Ovsjanikov'10

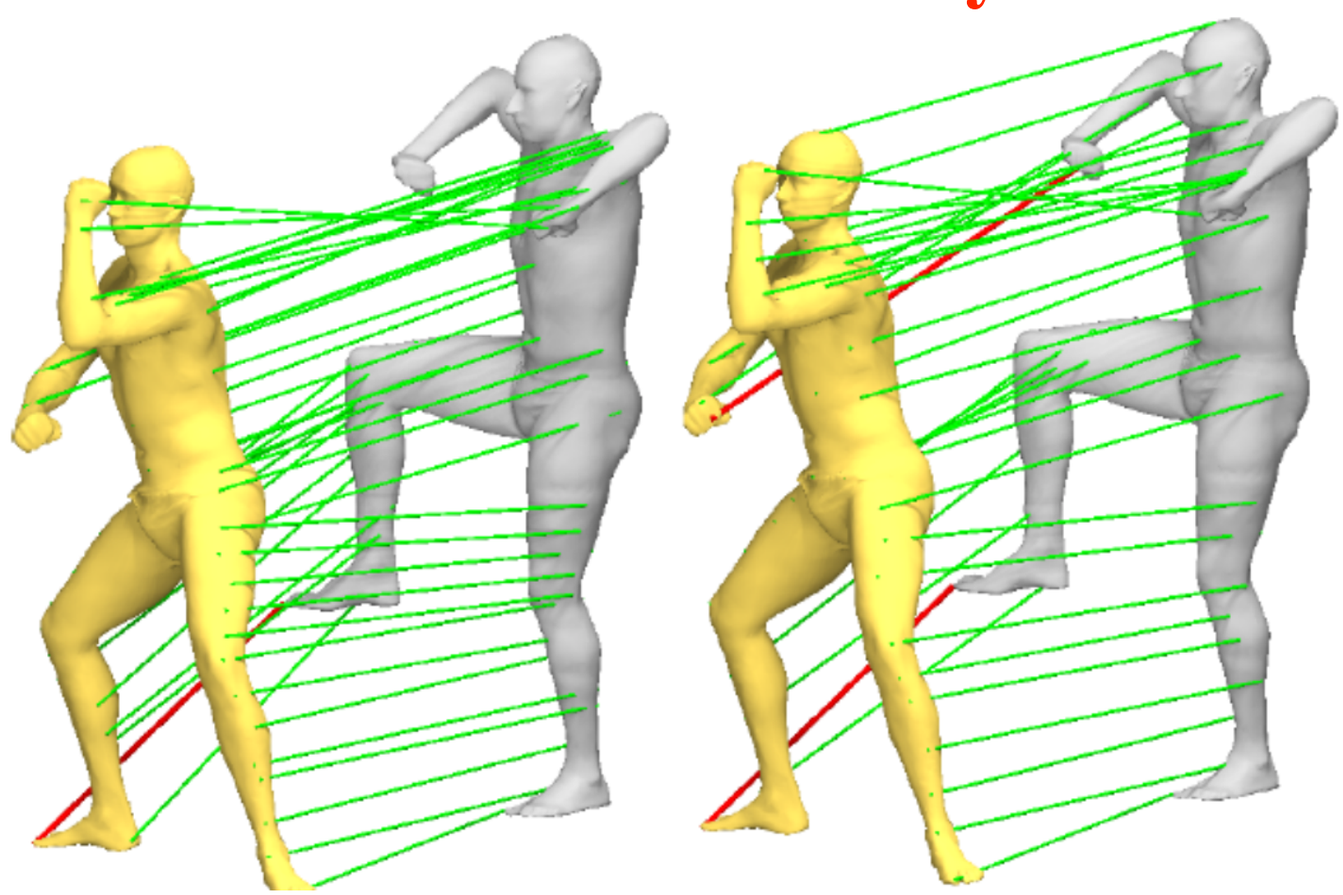
Heat Kernel Map

Pros:

- The search space is TINY
- Naturally works in partial case

Cons:

➡ **Sensitive to deviations from isometry**



Ovsjanikov'10

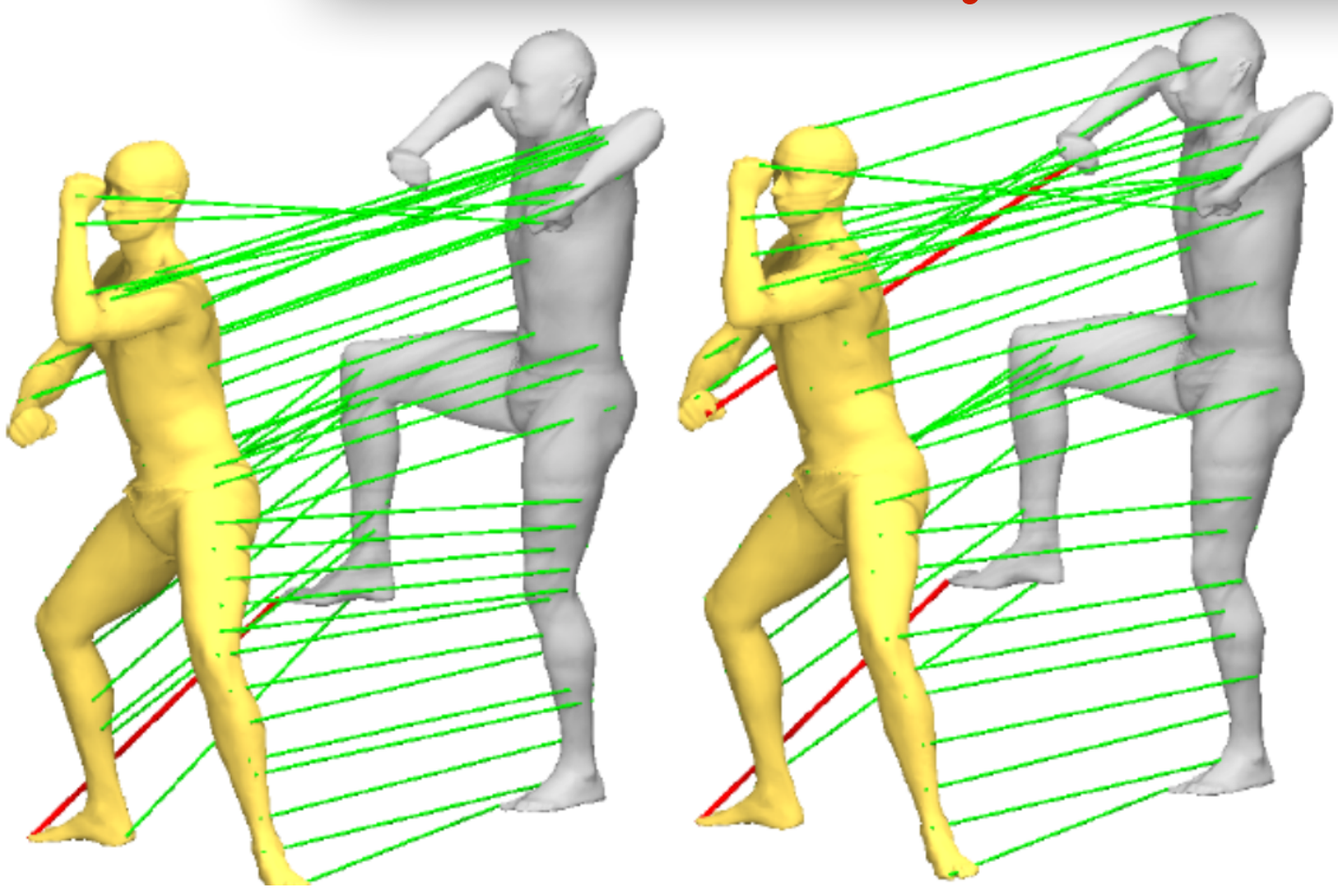
Heat Kernel Map

Pros:

- The search space is T
- Naturally works in pa

Cons:

➡ **Sensitive to deviations from isometry**



Ovsjanikov'10

Conformal Geometry

Isometry Revisited

Another definition of isometry:

- Angle-preserving (conformal)
- Area-preserving

Isometry Revisited

Another definition of isometry:

- Angle-preserving (conformal)
- Area-preserving

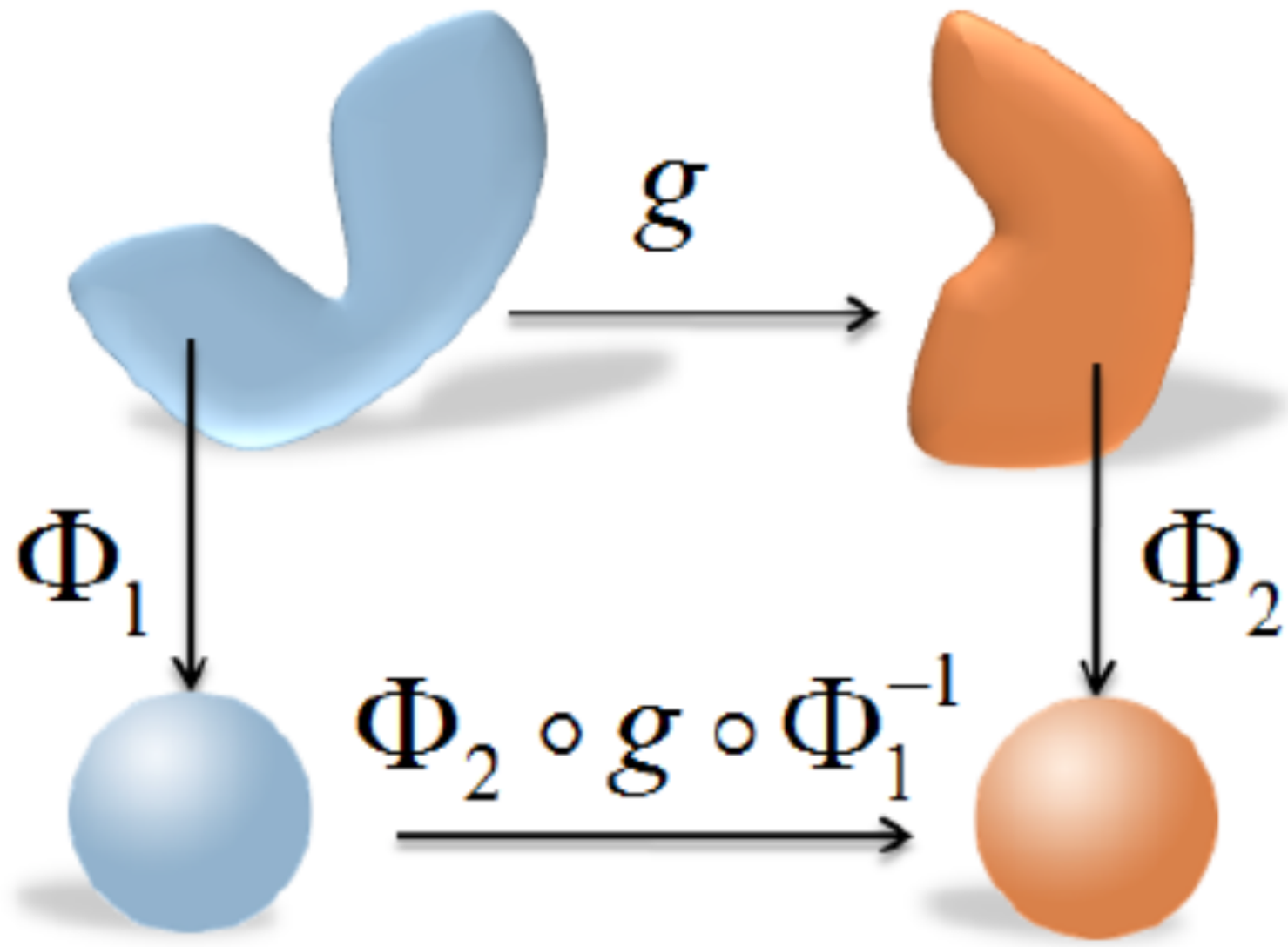
$$\text{isometries} \subseteq \text{conformal maps}$$

Hard! **Easier**

Conformal Maps

Two easy subproblems

- Conformal map to a sphere
- Conformal map between spheres



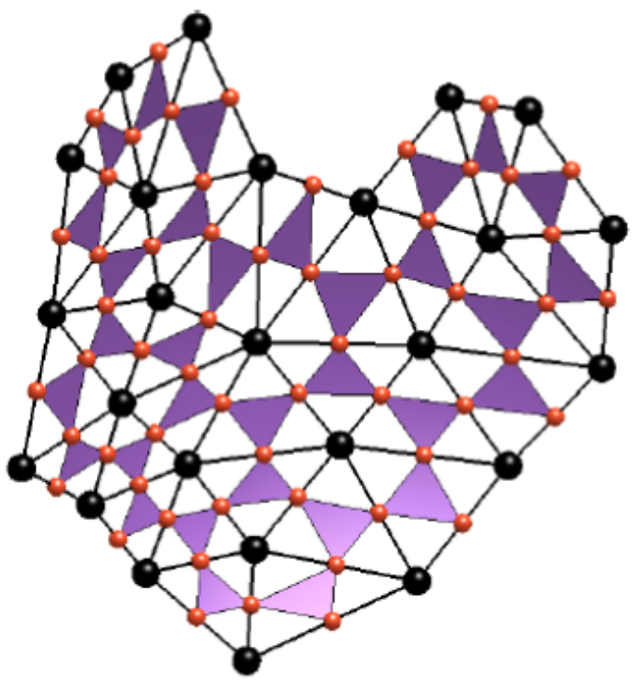
Lipman'10

Conformal Mapping

Two easy subproblems

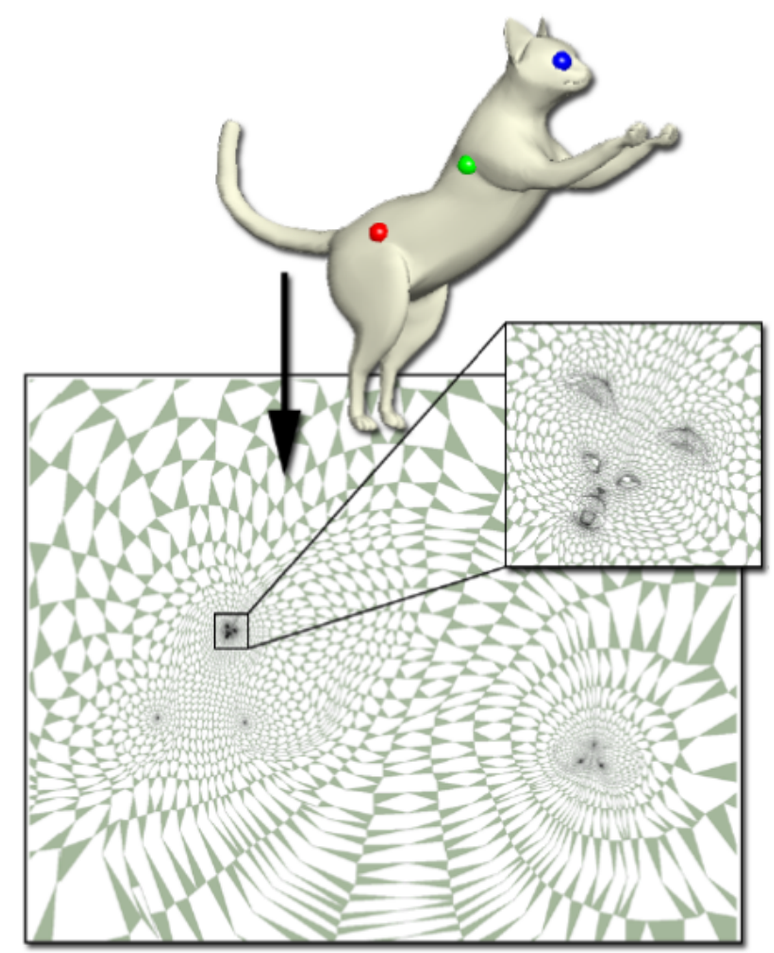
➡ **Conformal map to a sphere**

- Conformal map between spheres



mid-edge uniformization

"unwarped" sphere:



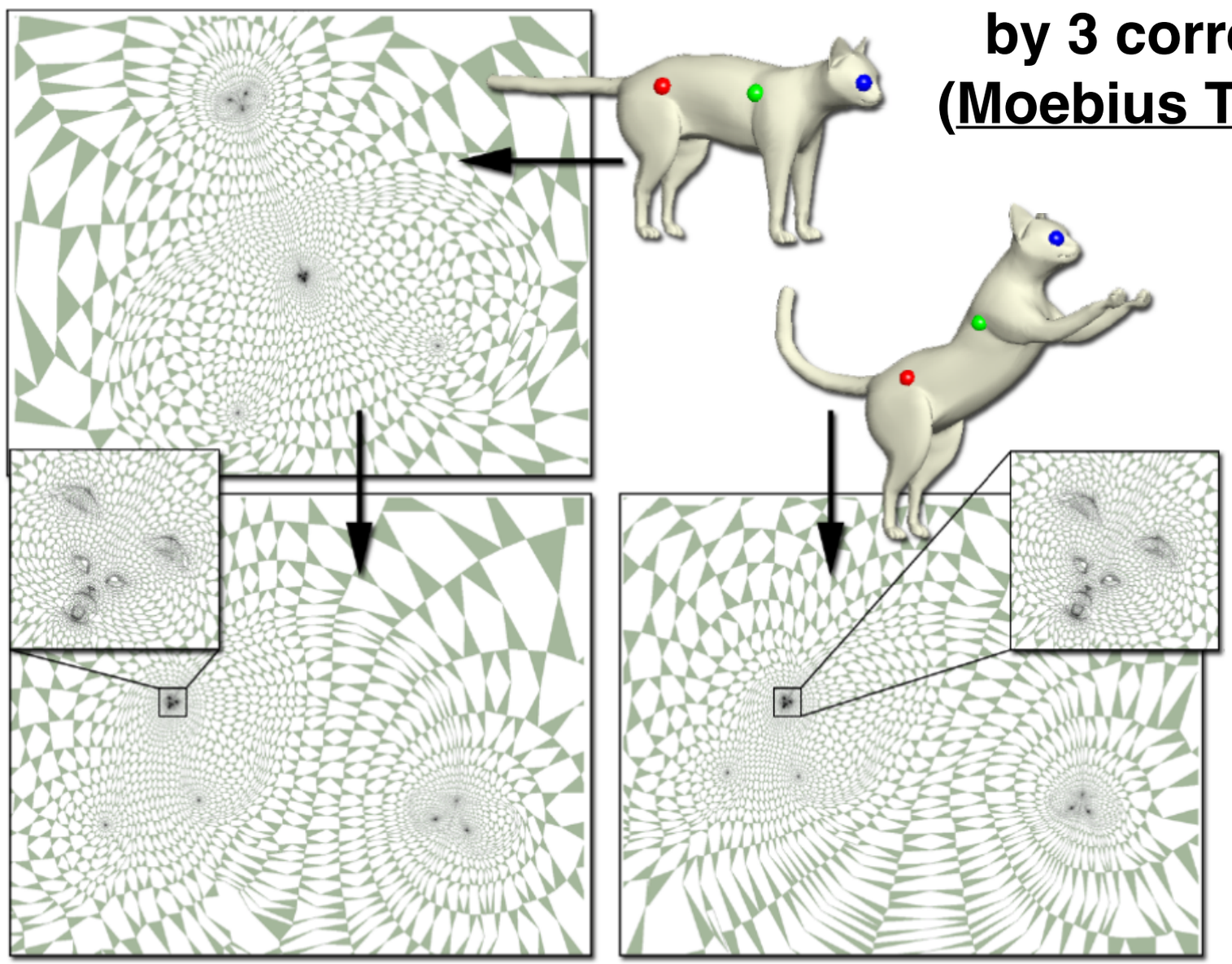
Lipman'10

Conformal Mapping

Two easy subproblems

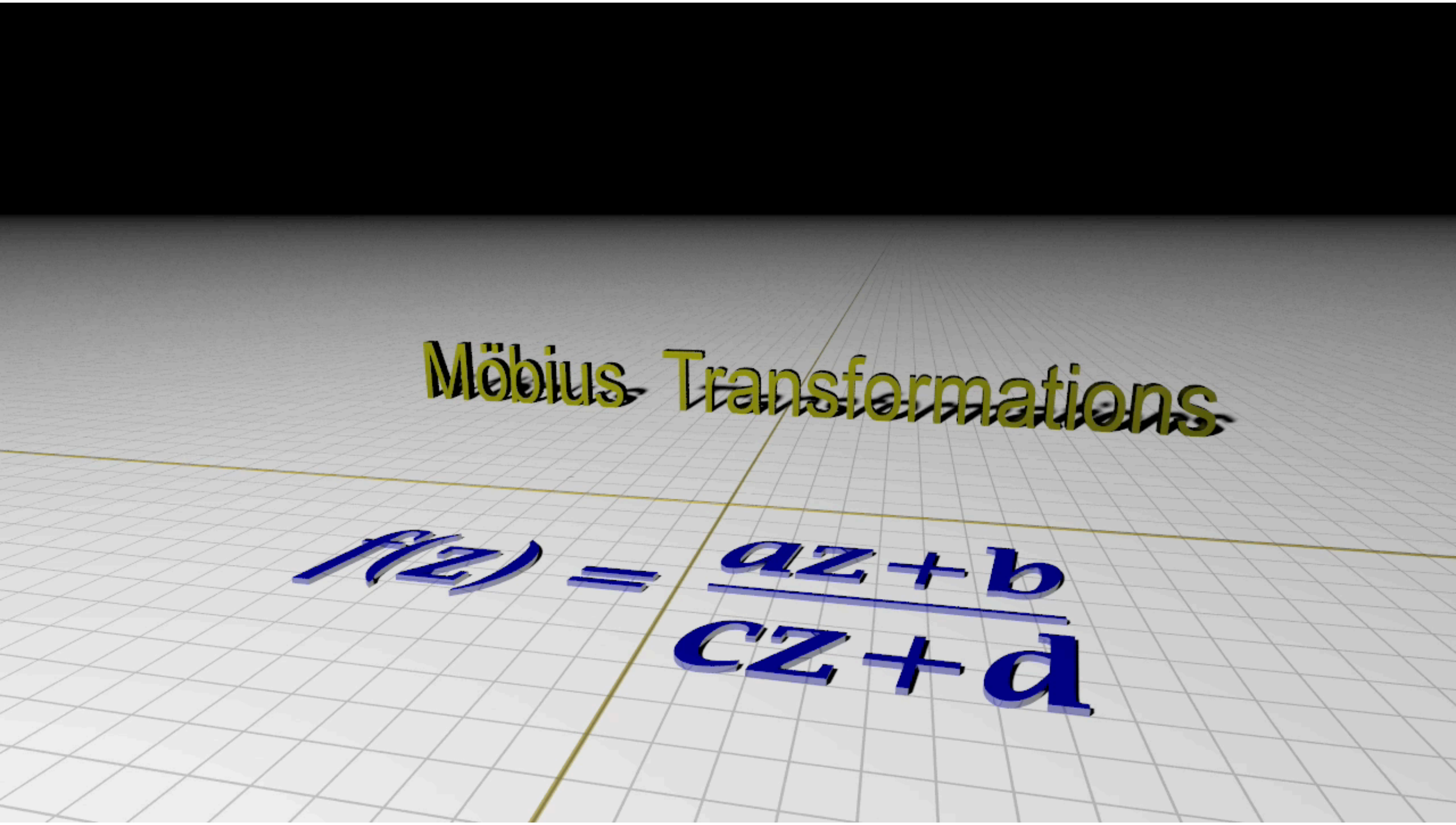
- Conformal map to a sphere
- ➡ **Conformal map between spheres**

Conformal Map is uniquely defined by 3 correspondences (Moebius Transformation)



Lipman'10

Moebeius Transformations



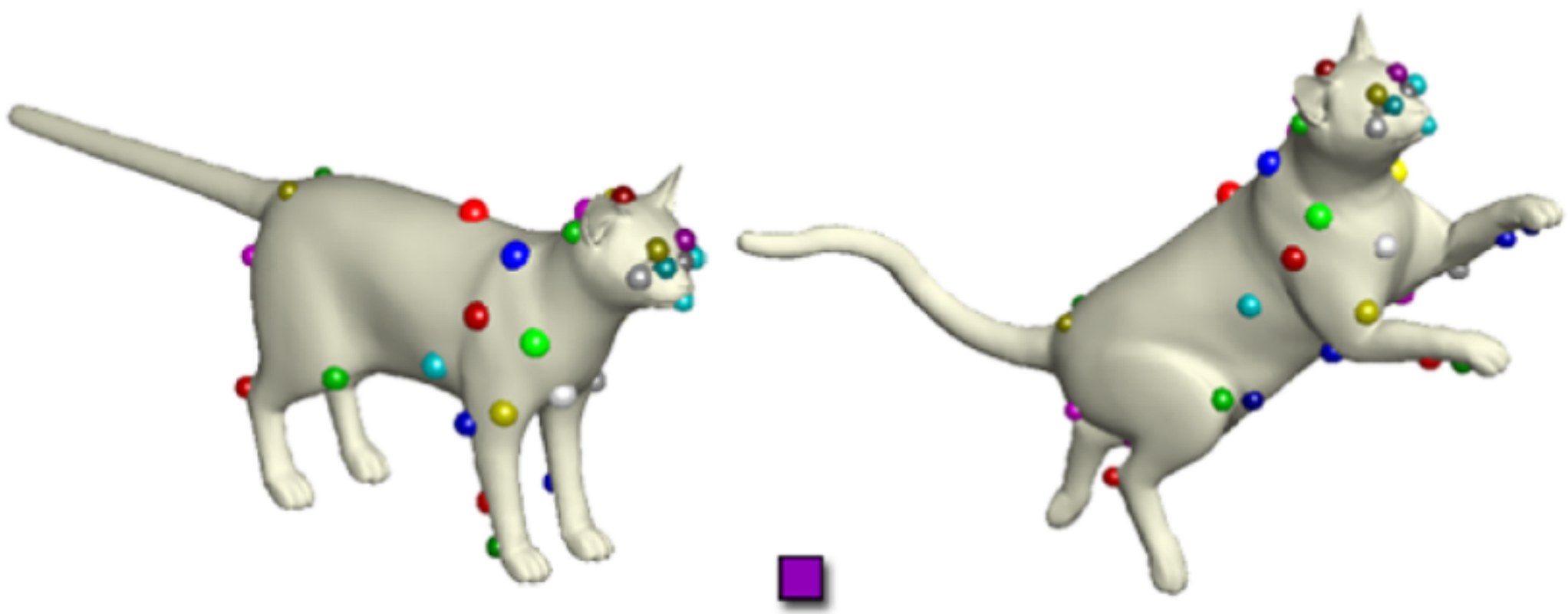
http://www.ima.umn.edu/~arnold/moebius/

Arnold and Rogness

Moebius Voting

Algorithm for Isometric Shapes:

- Repeat for many triplets:
- Propose 3 correspondences
- Compute a conformal map
- Pick the one that has the smallest area distortion



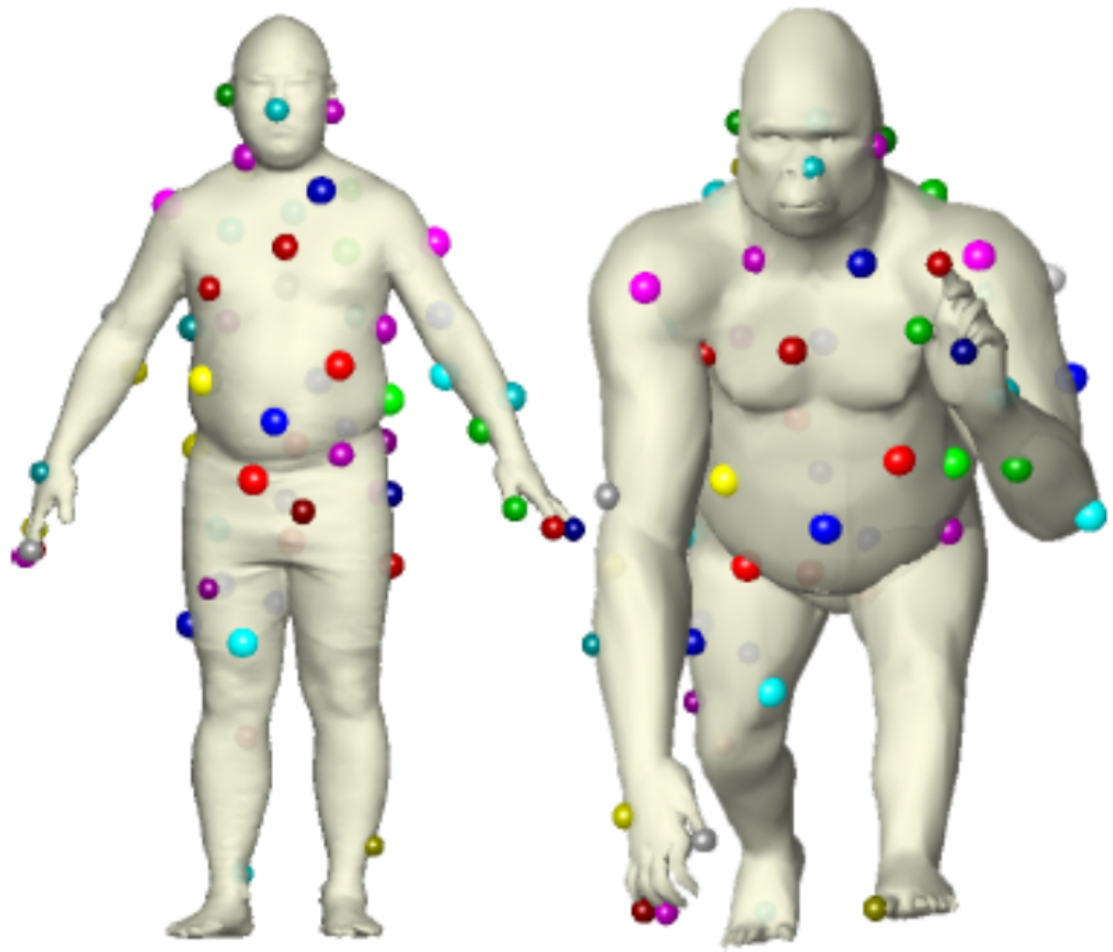
Lipman'10

Moebius Voting

Algorithm for **Non-**Isometric Shapes:

- Repeat for many triplets:
- Propose 3 correspondences
- Compute a conformal map

➡ **VOTE based on the area distortion**



Lipman'10

Conformal Mapping

Pros:

- Efficient
- Can handle some non-isometry

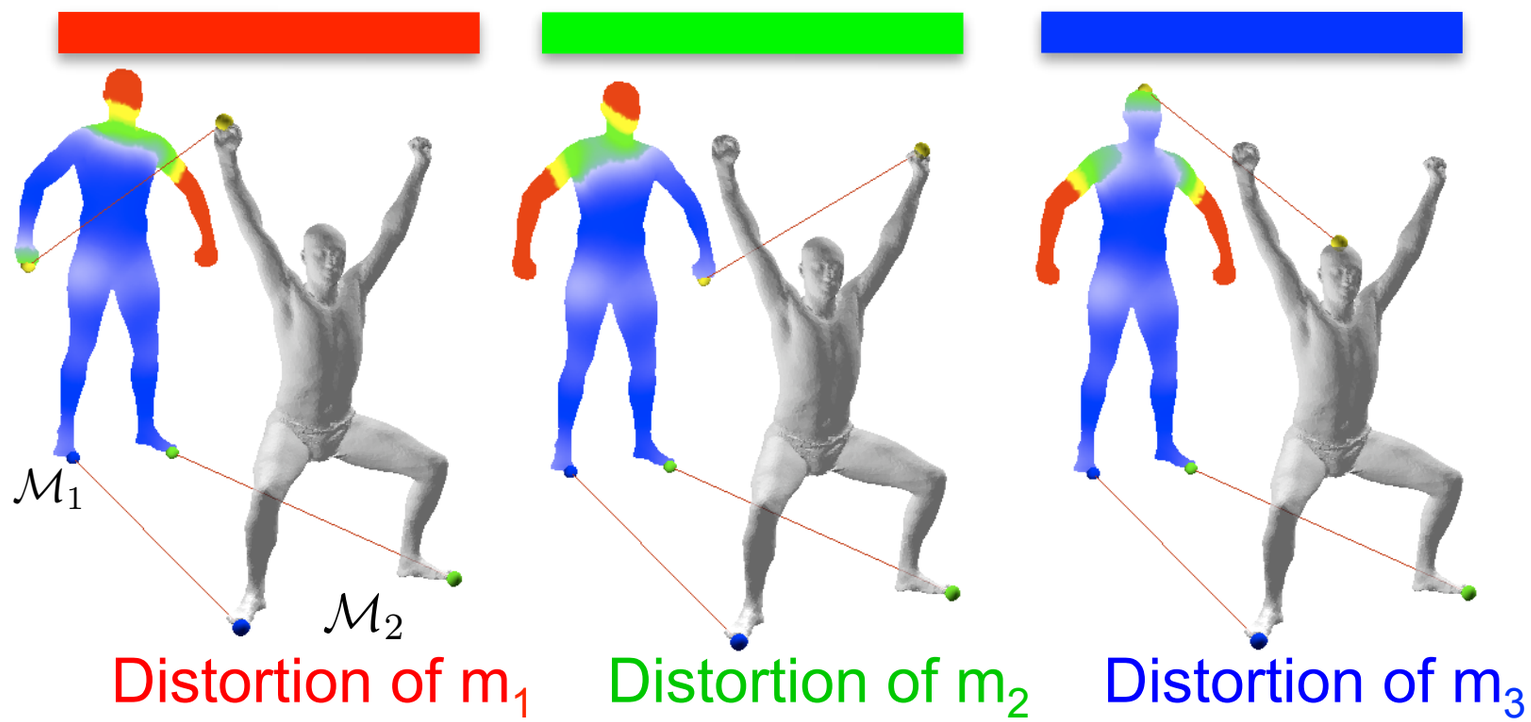
Cons:

- Does not provide a smooth or continuous map
- Does not optimize global distortion
- Works for genus 0 manifold surfaces

Lipman'10

Blended Intrinsic Maps

Blend conformal maps into a smooth map

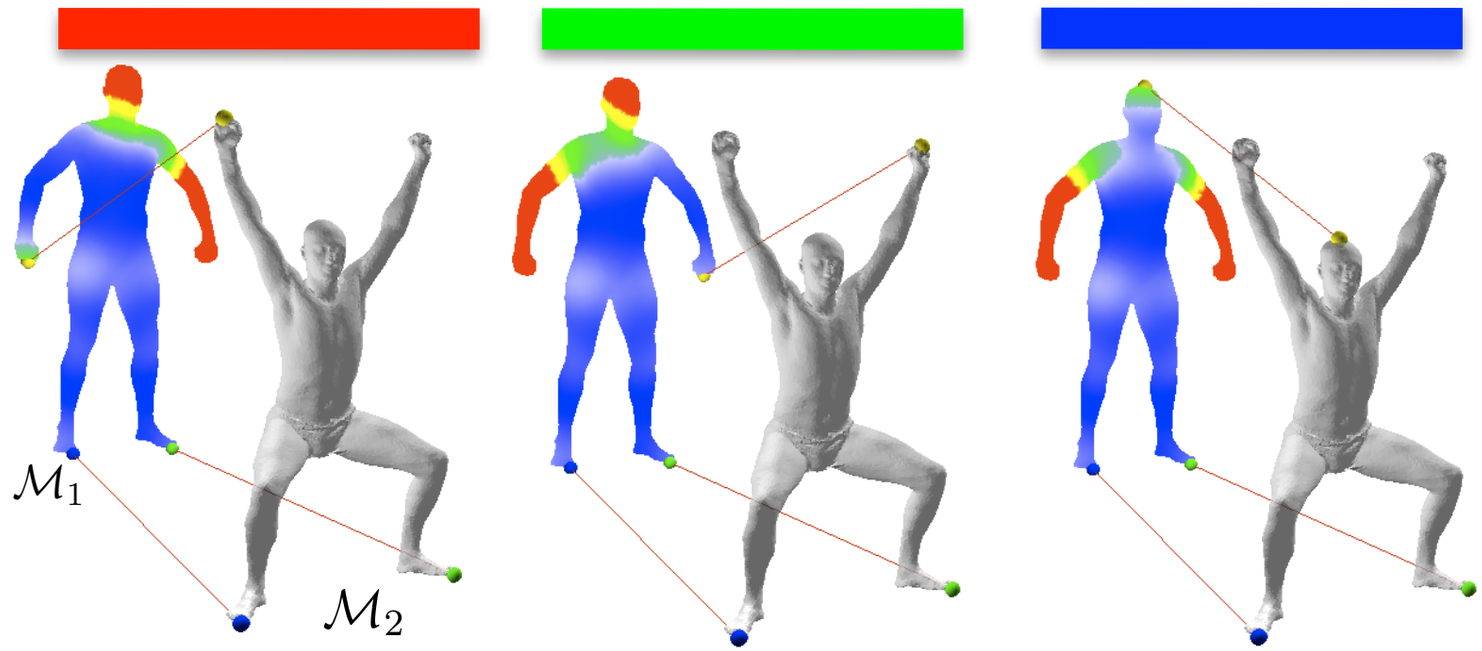


These conformal maps introduce area distortions in different regions

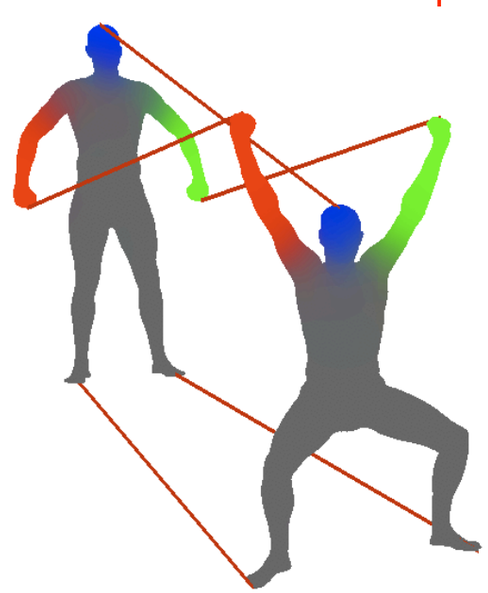
Kim'11

Blended Intrinsic Maps

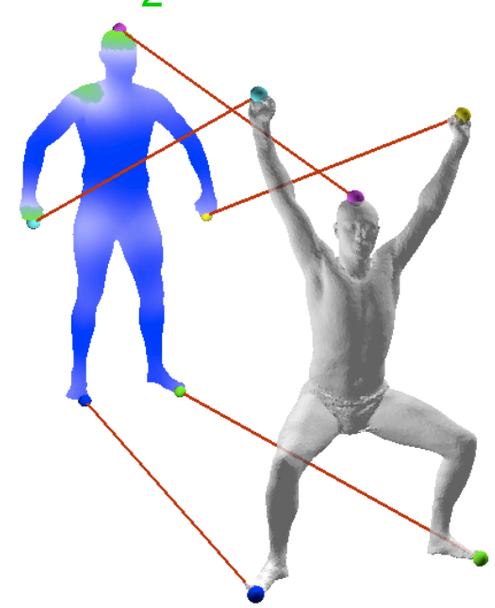
Blend conformal maps into a smooth map



Distortion of m_1 Distortion of m_2 Distortion of m_3



Blending Weights for m_1, m_2, and m_3



Distortion of the Blended Map

Kim'11

Blended Intrinsic Maps

Algorithm:

- Generate consistent maps
- Find blending weights (per-point weight for each map)
- Blend maps

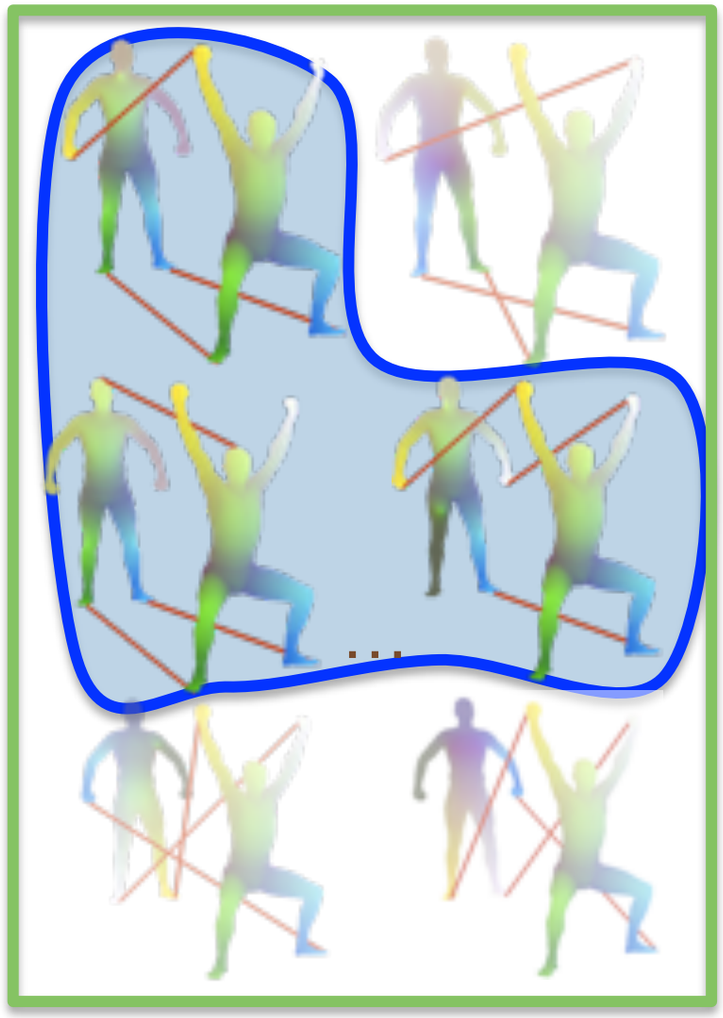
Kim'11

Blended Intrinsic Maps

Algorithm:

➡ **Generate consistent maps**

- Find blending weights (per-point weight for each map)
- Blend maps



Set of consistent candidate maps

Kim'11

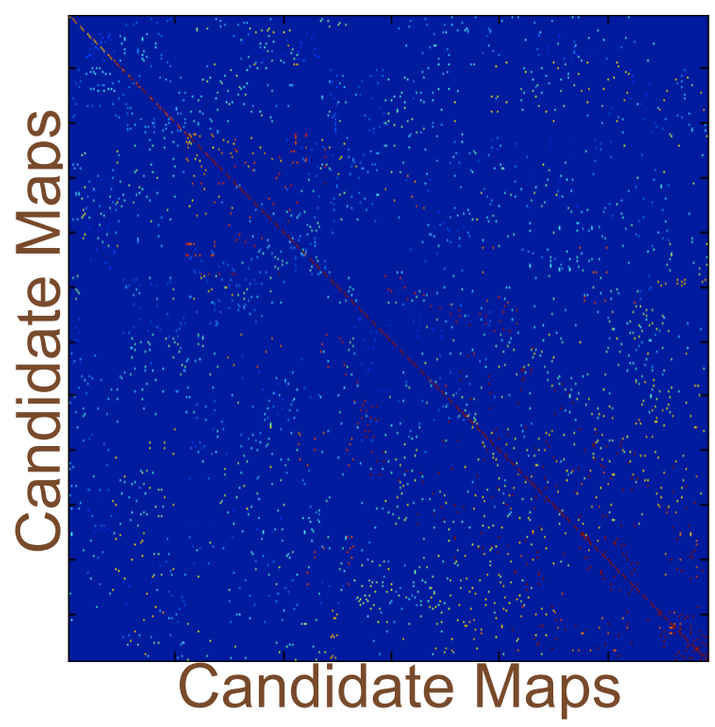
Blended Intrinsic Maps

Algorithm:

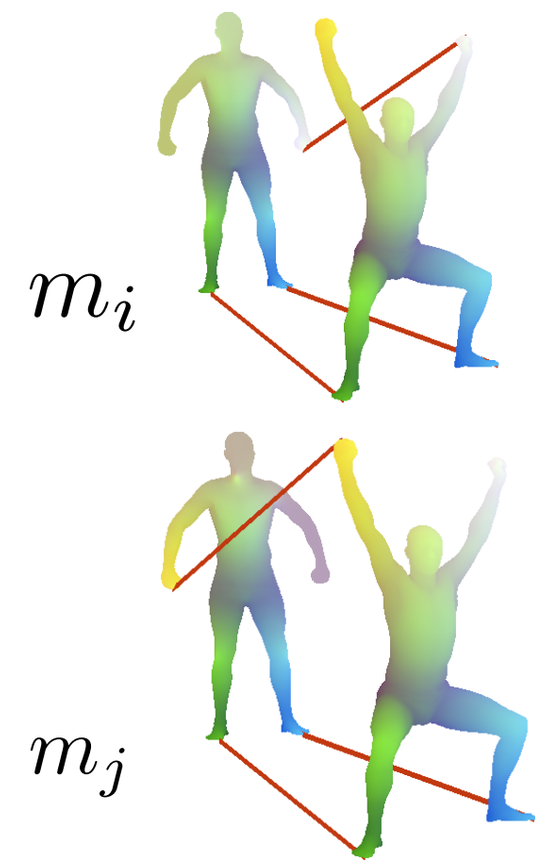
➡ **Generate consistent maps**

- Find blending weights (per-point weight for each map)
- Blend maps

$$\mathbf{B}_{i,j} = \int_{M_1} c_i(p) c_j(p) S_{i,j}(p) dA(p)$$



Map similarity matrix



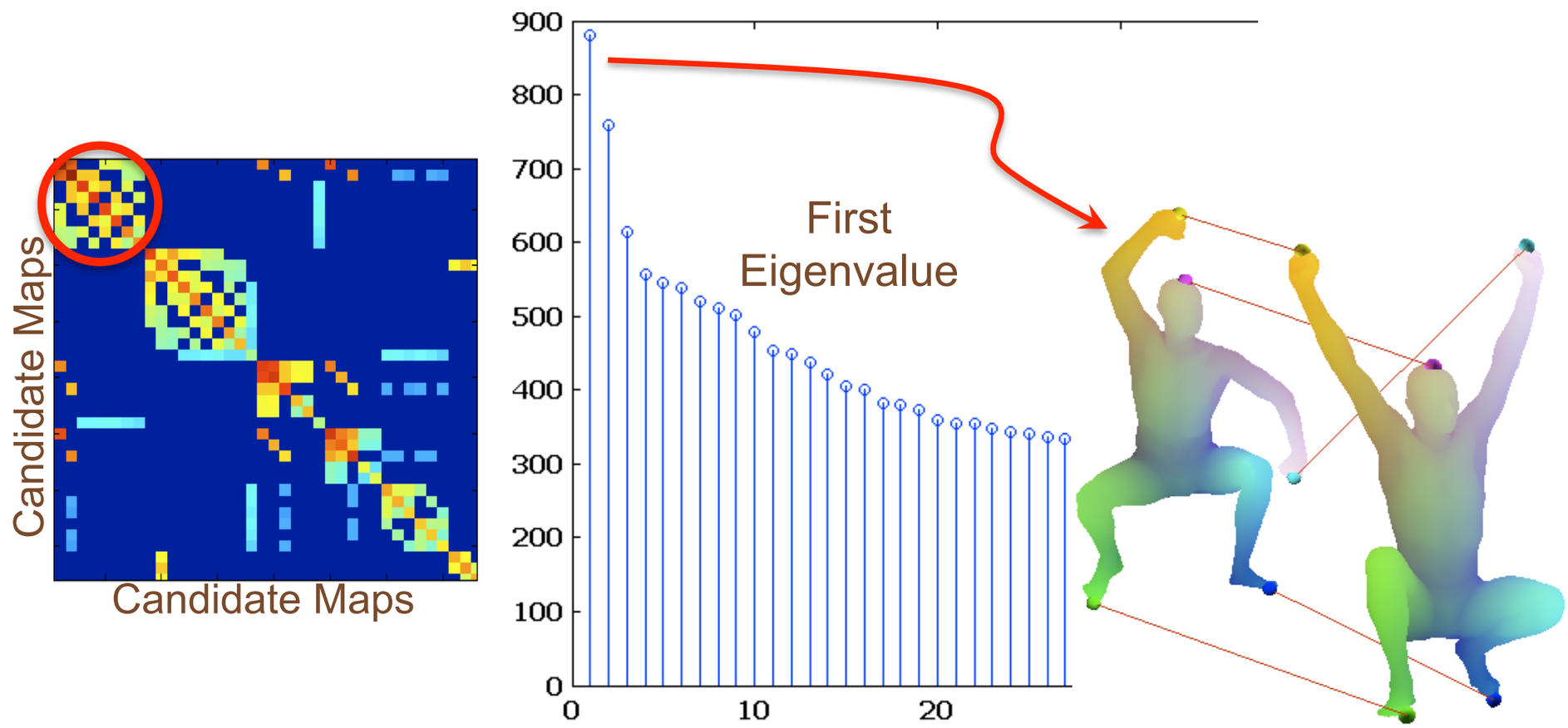
Kim'11

Blended Intrinsic Maps

Algorithm:

➡ **Generate consistent maps**

- Find blending weights (per-point weight for each map)
- Blend maps



Eigen-analysis to find "blocks" of mutually-similar maps

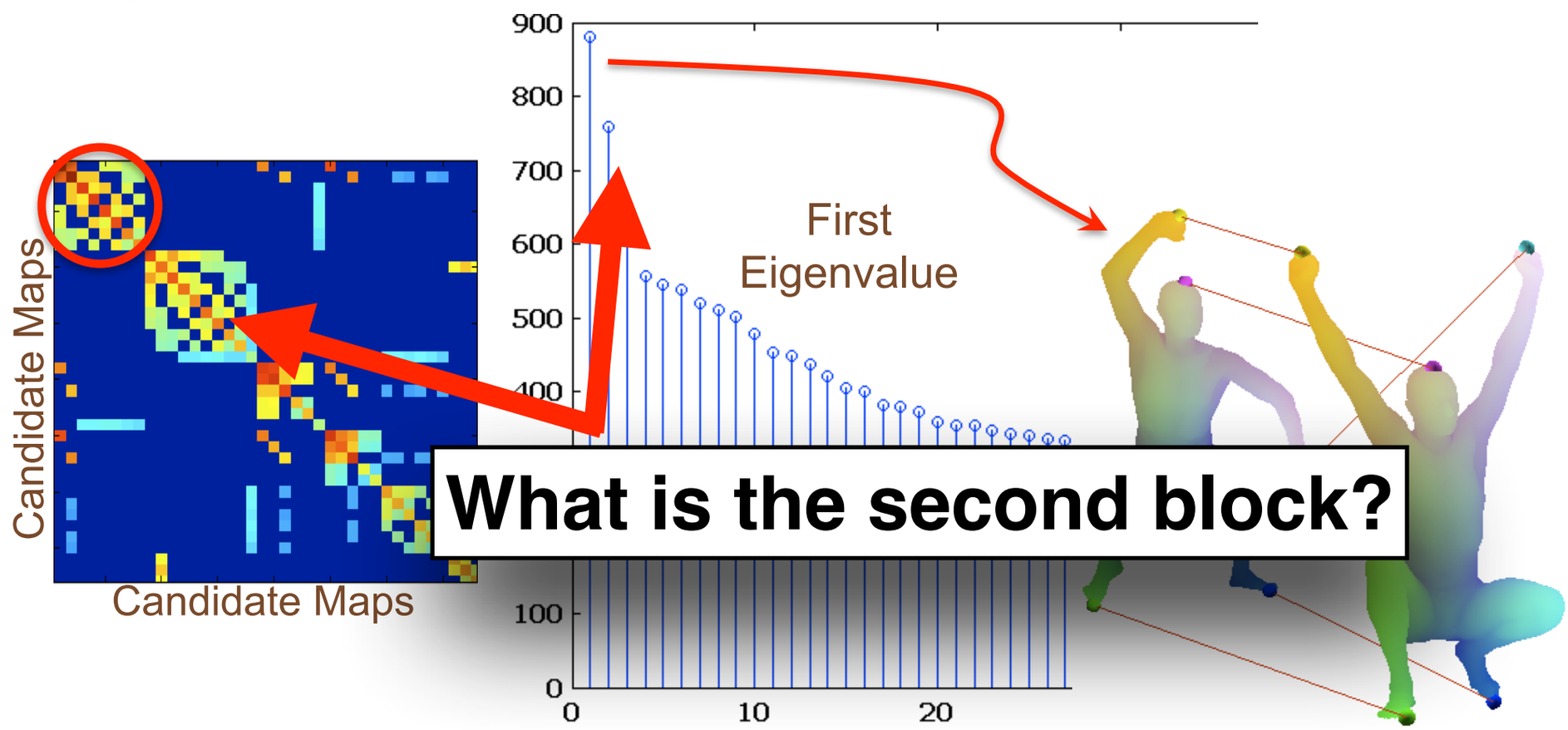
Kim'11

Blended Intrinsic Maps

Algorithm:

➡ **Generate consistent maps**

- Find blending weights (per-point weight for each map)
- Blend maps



Eigen-analysis to find "blocks" of mutually-similar maps

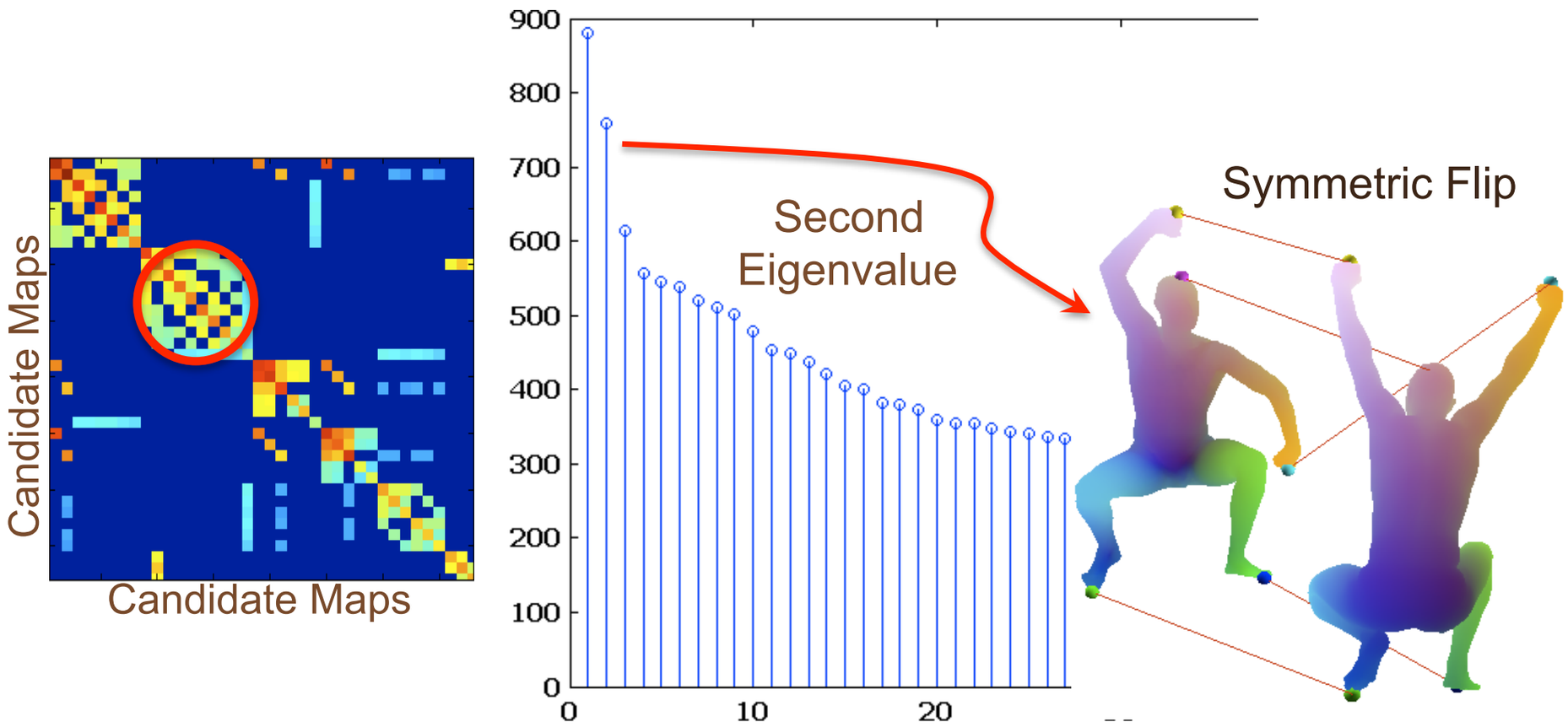
Kim'11

Blended Intrinsic Maps

Algorithm:

➡ **Generate consistent maps**

- Find blending weights (per-point weight for each map)
- Blend maps



Eigen-analysis to find "blocks" of mutually-similar maps

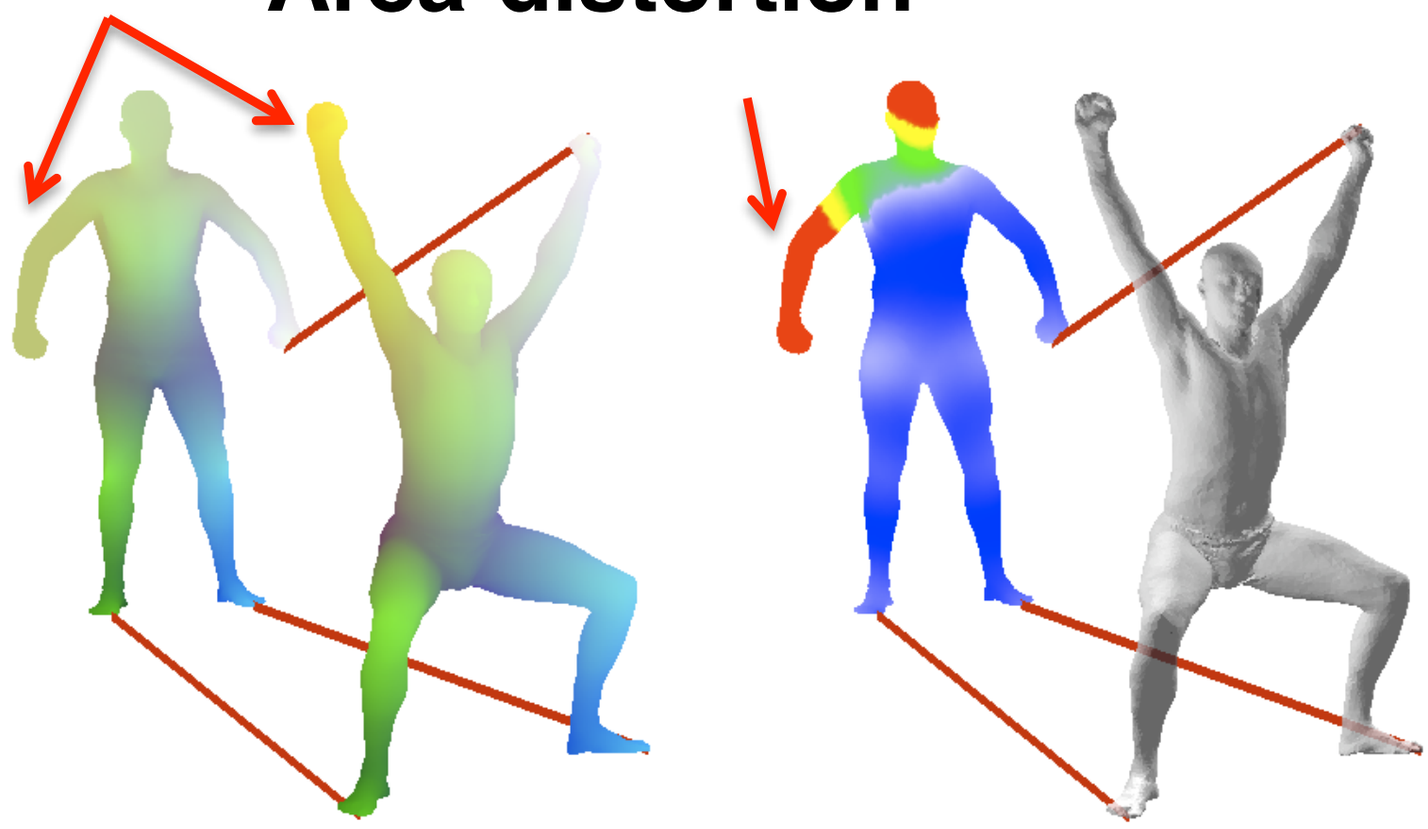
Kim'11

Blended Intrinsic Maps

Algorithm:

- Generate consistent maps
- ➡ **Find blending weights (per-point weight for each map)**
- Blend maps

Area-distortion



Candidate Map

Blending Weight

$c_i(p)$

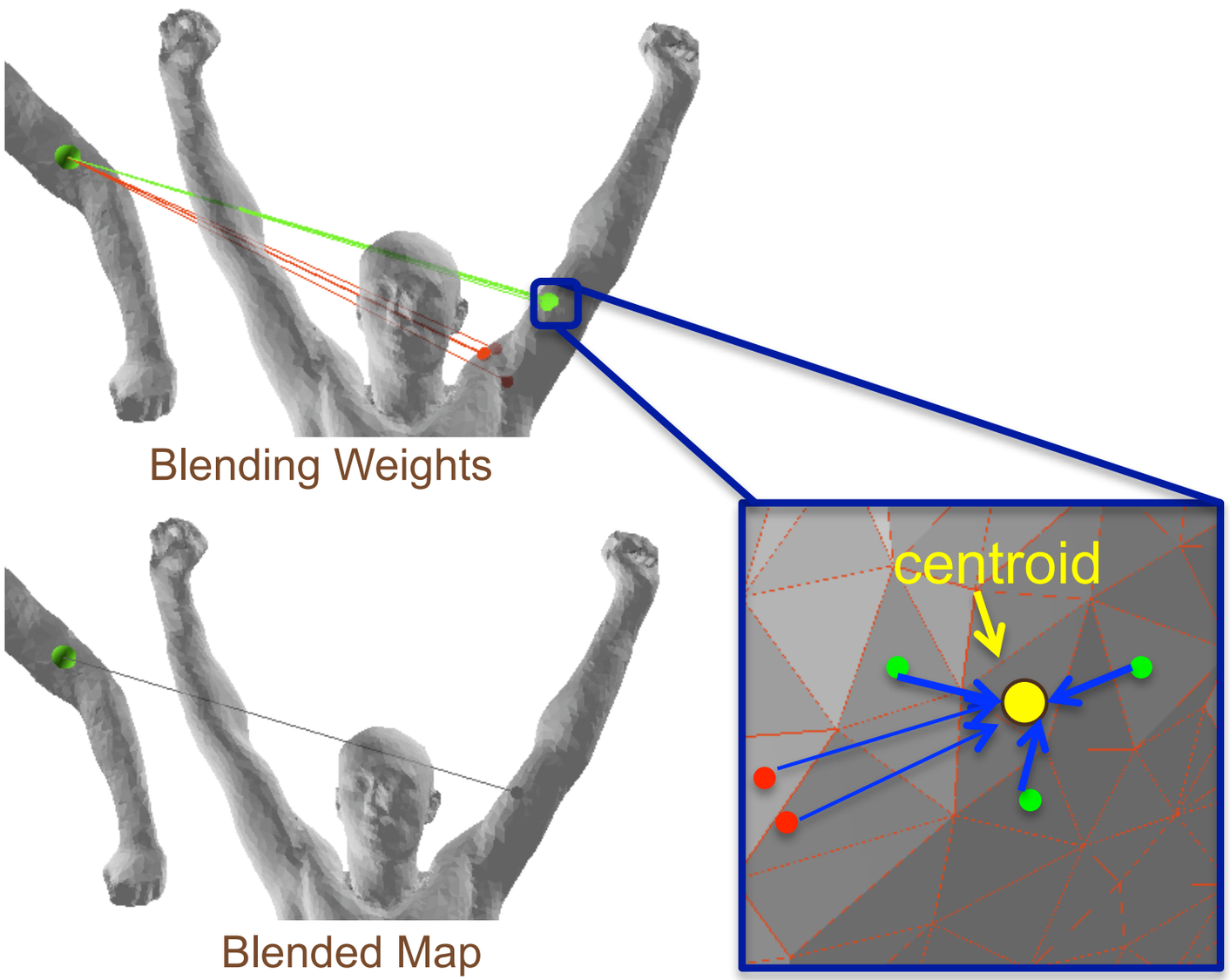
Kim'11

Blended Intrinsic Maps

Algorithm:

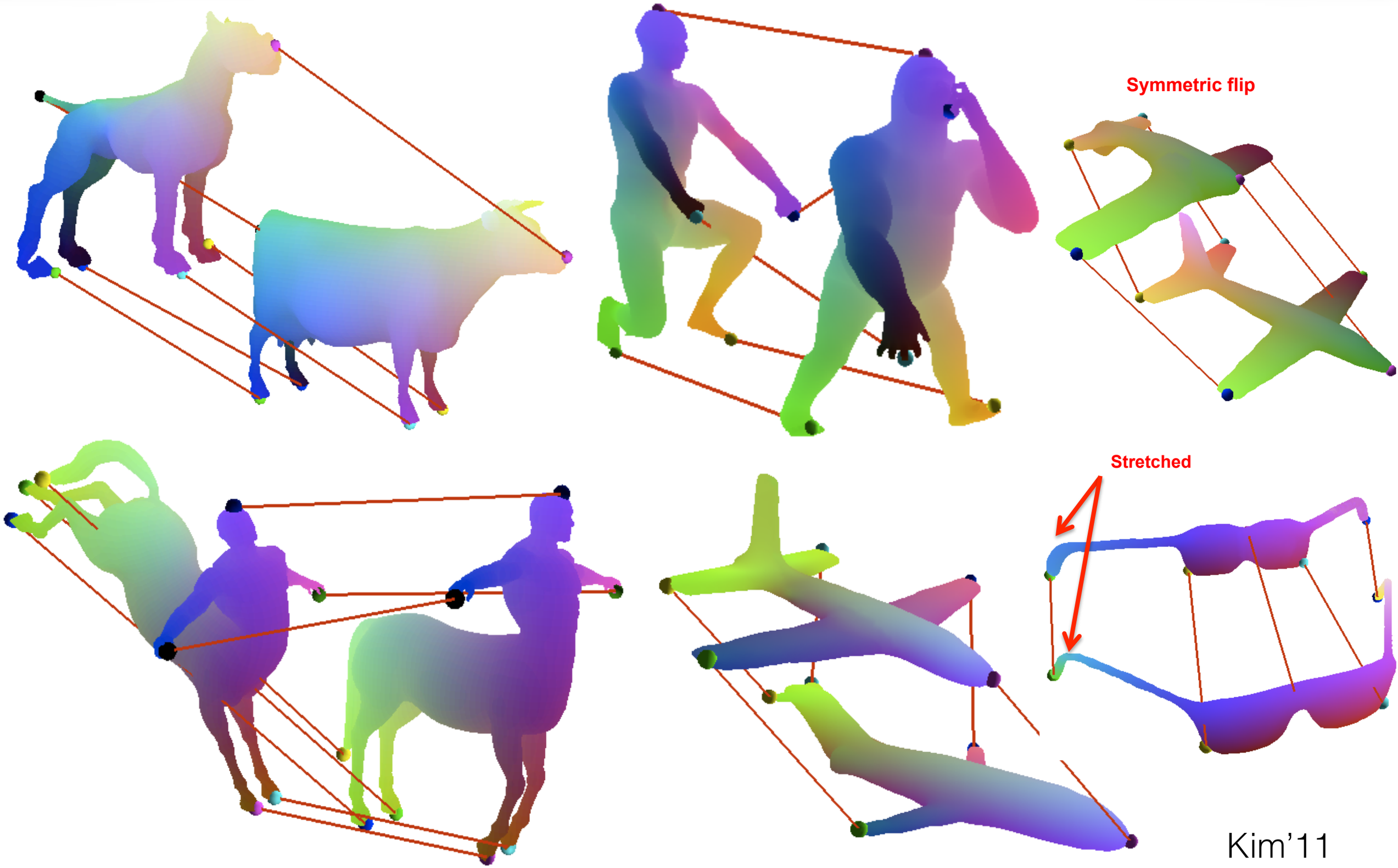
- Generate consistent maps
- Find blending weights (per-point weight for each map)

➡ **Blend maps**



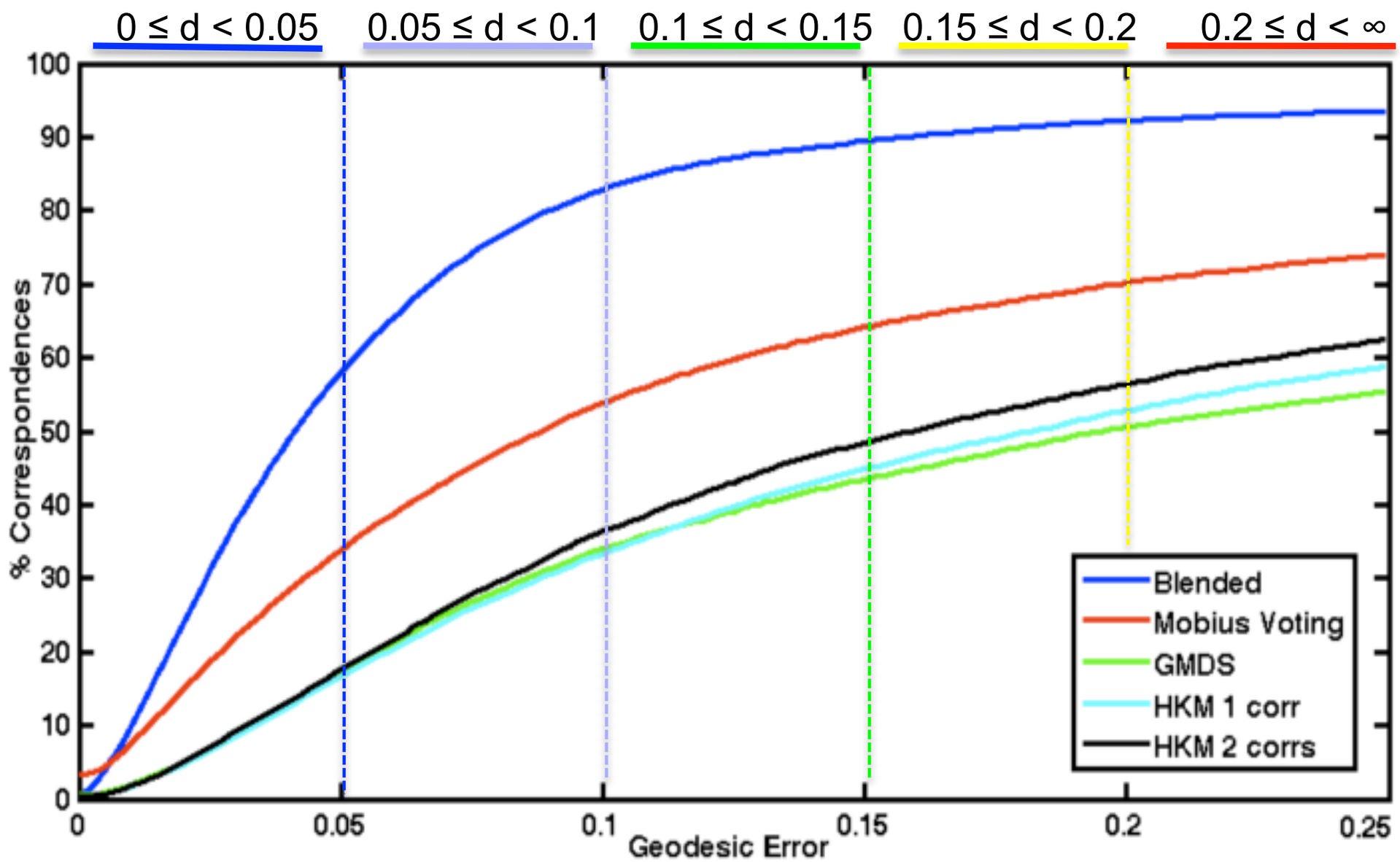
Kim'11

Some Examples



Kim'11

Evaluation



Kim'11

Blended Intrinsic Maps

Pros

- Highly non-isometric shapes
- Efficient

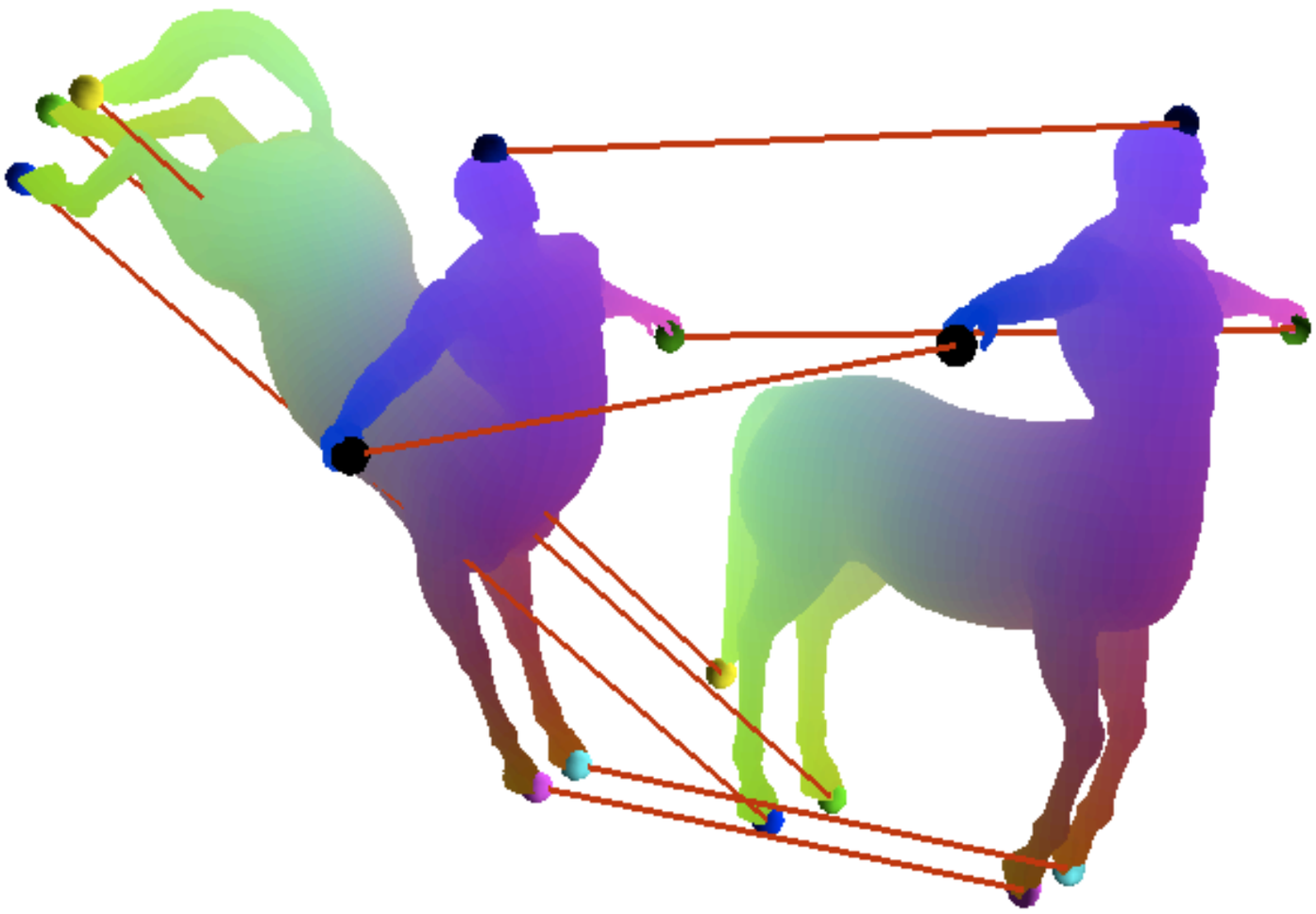
Cons

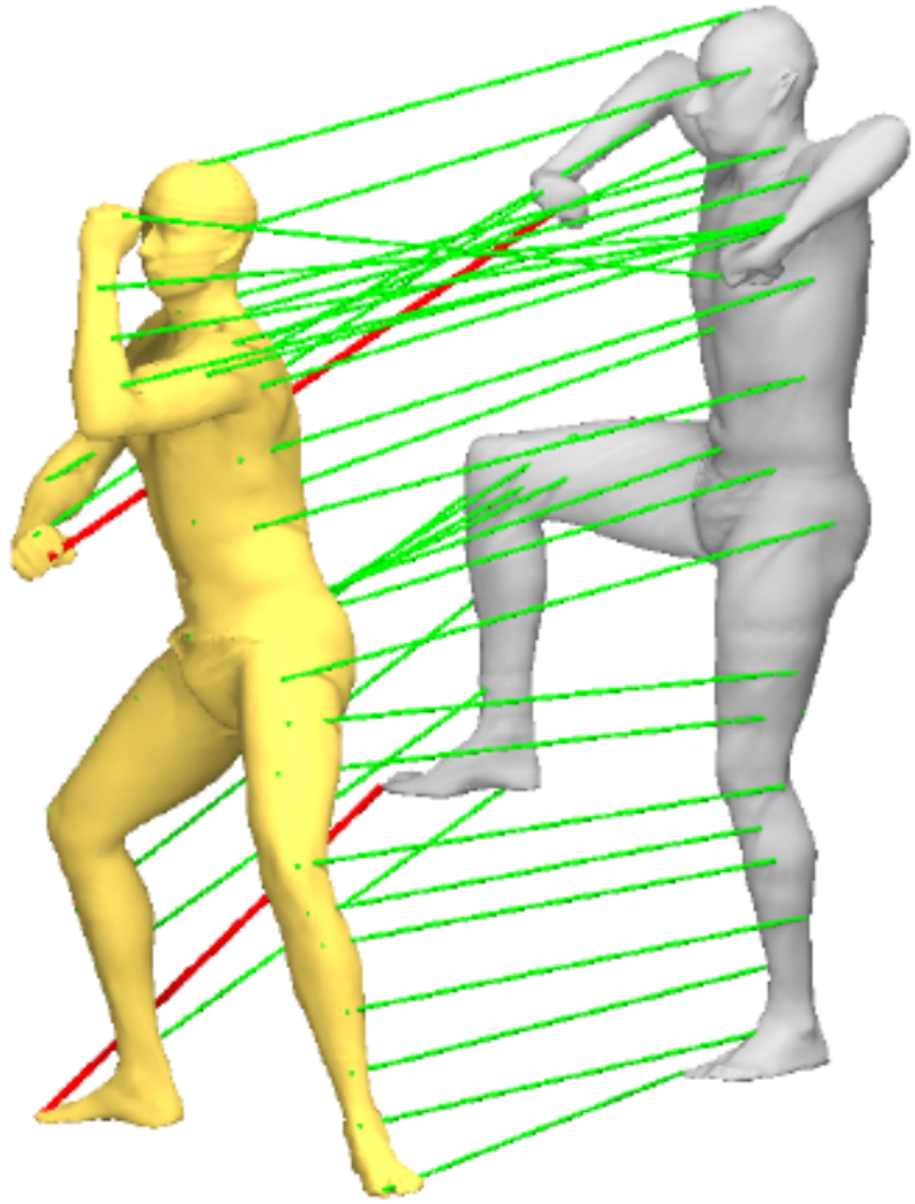
- Still has a lot of area distortion for some shapes
- Genus 0 manifold surfaces

Kim'11

Functional Maps

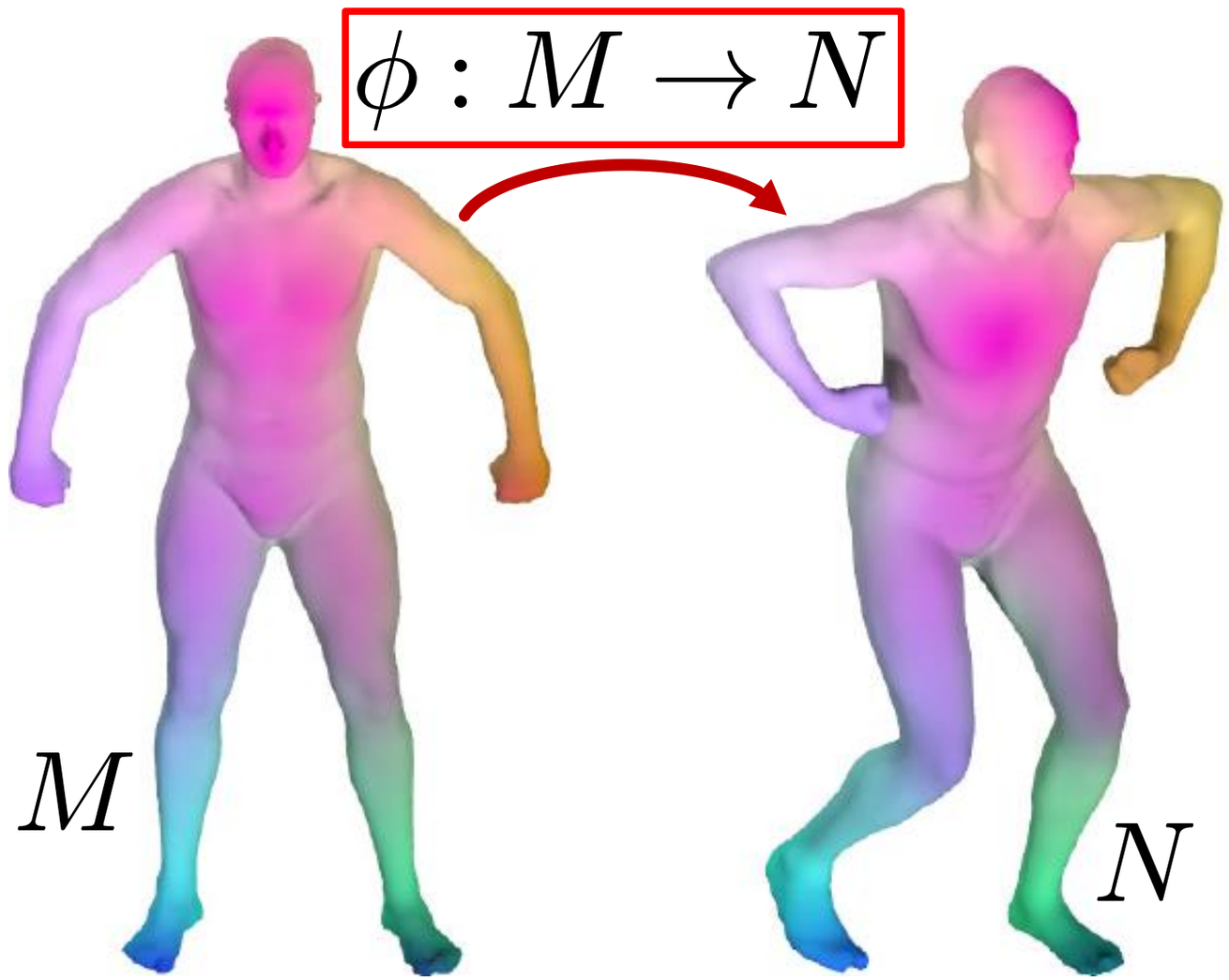
What is a map?





Functional Maps

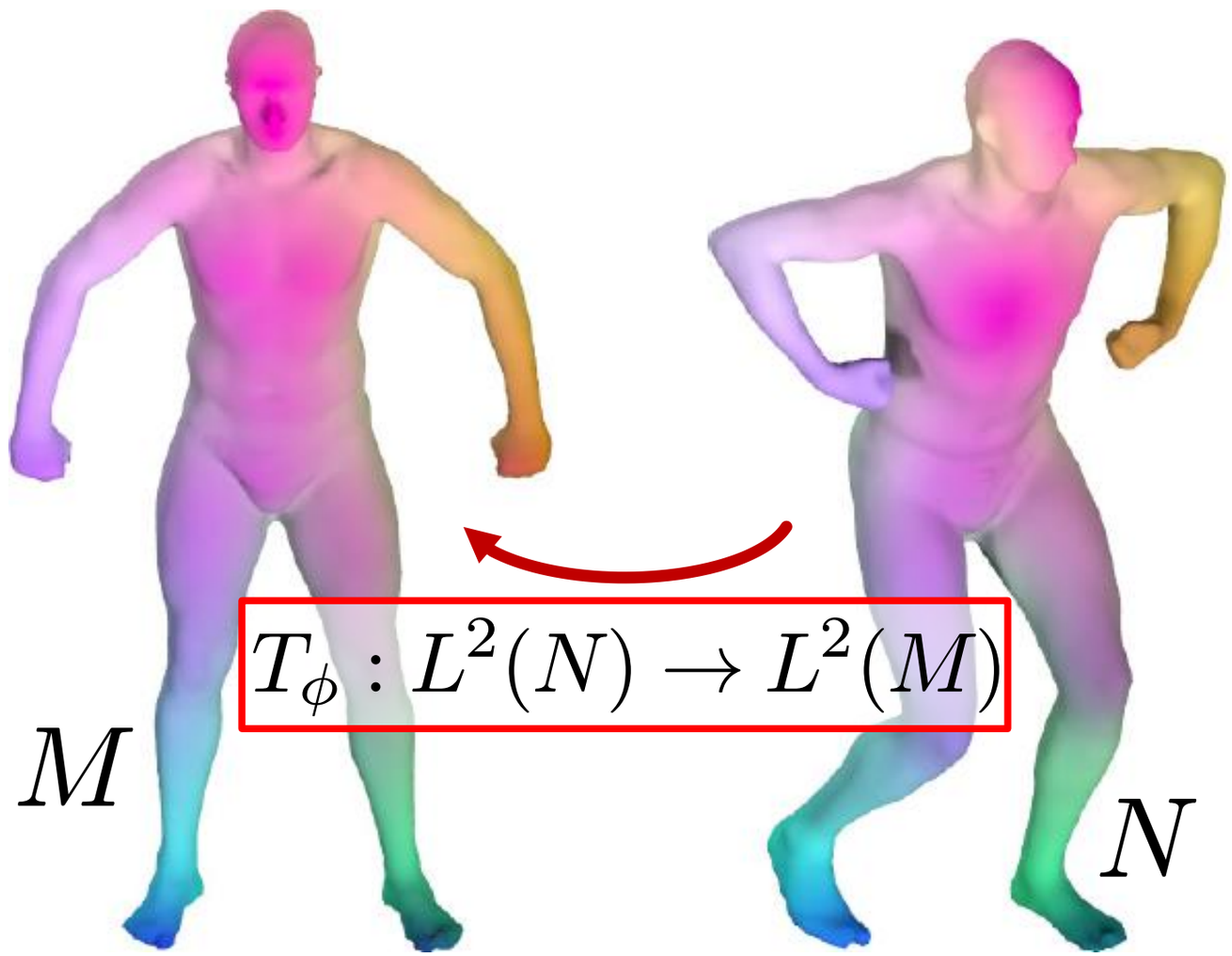
Map functions rather than points



Ovsjanikov'12
Slides by Solomon

Functional Maps

Map functions rather than points



Ovsjanikov'12
Slides by Solomon

Functional Maps

How to represent functions **on surfaces**?

Ovsjanikov'12
Slides by Solomon

Functional Maps

How to represent functions **on surfaces**?

$f(x) = a_1 \cdot$  $+ a_2 \cdot$  $+ a_3 \cdot$  $+ \cdots$

Laplace-Beltrami

Ovsjanikov'12
Slides by Solomon

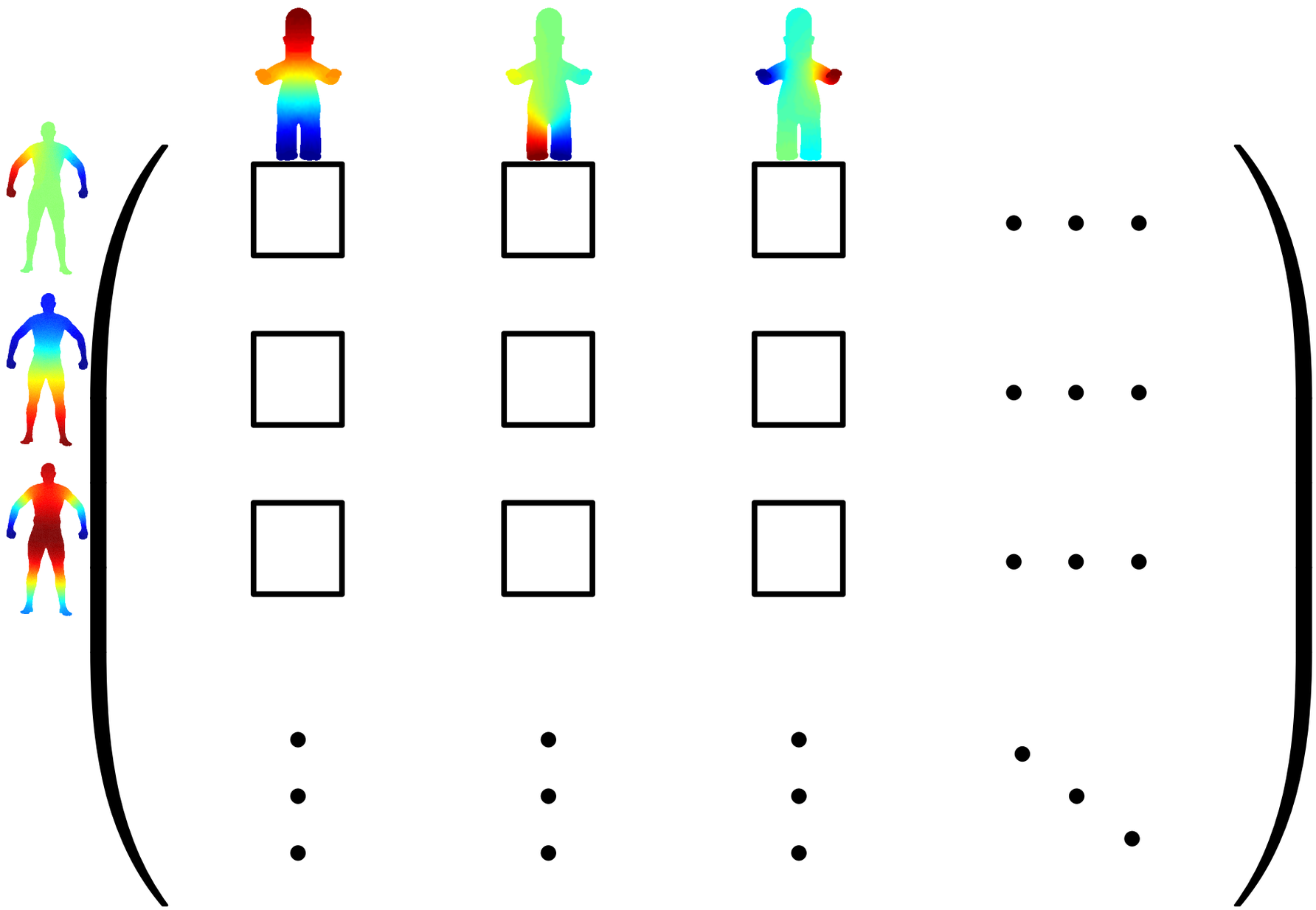
Functional Maps

How to represent functional maps?

Ovsjanikov'12
Slides by Solomon

Functional Maps

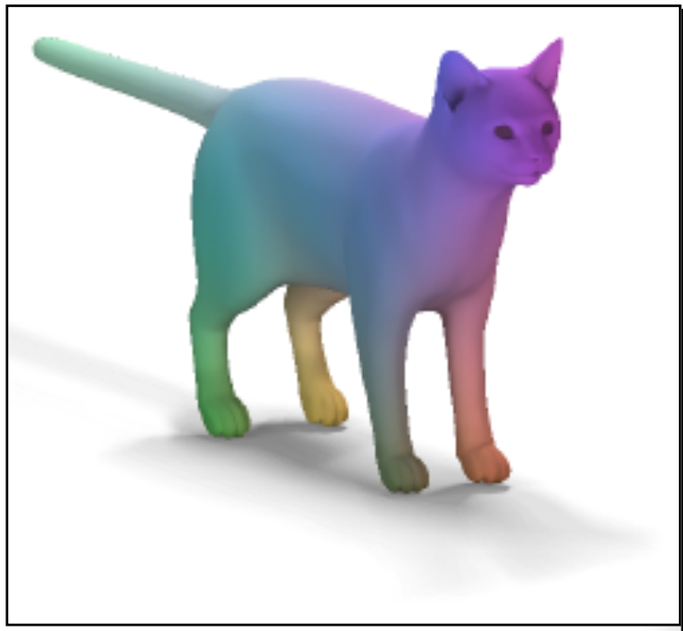
How to represent functional maps?

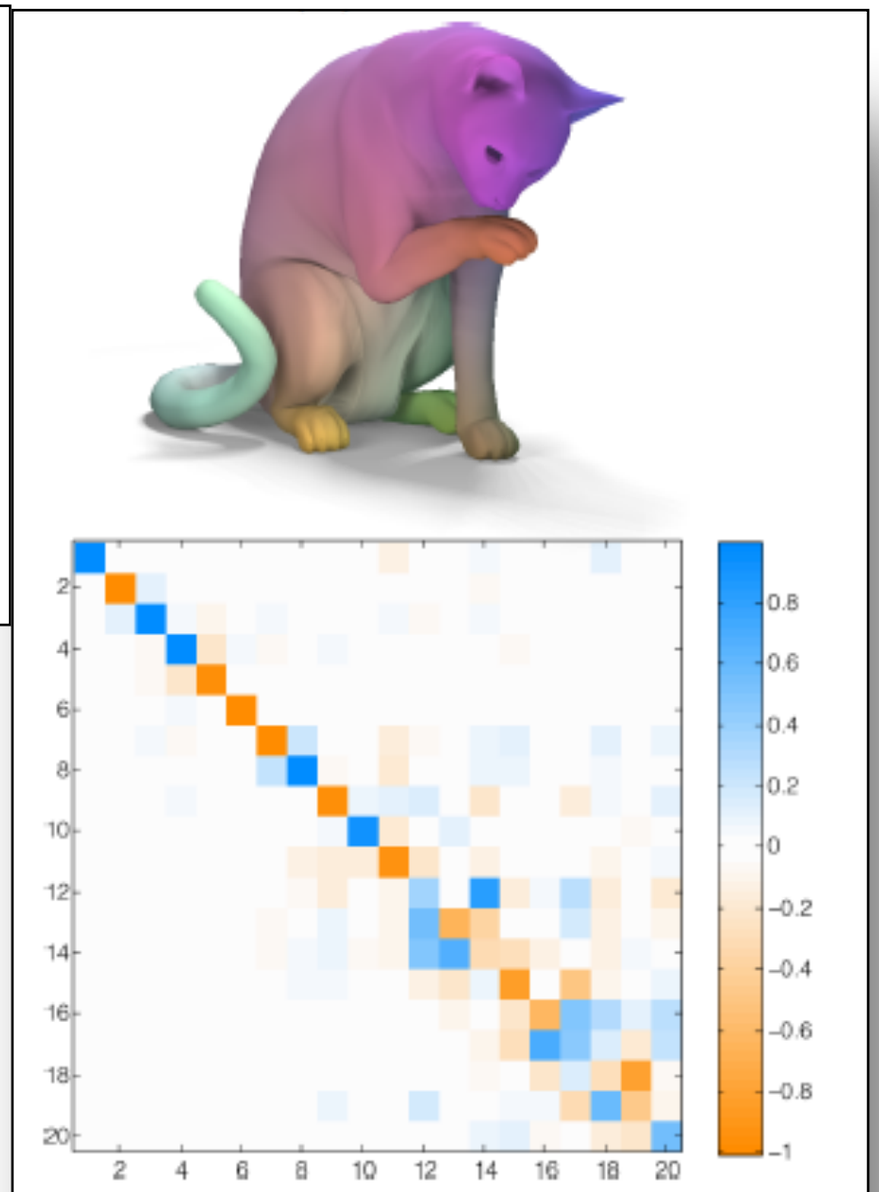


Functional Map Matrix, change of basis

Ovsjanikov'12
Slides by Solomon

Example Maps

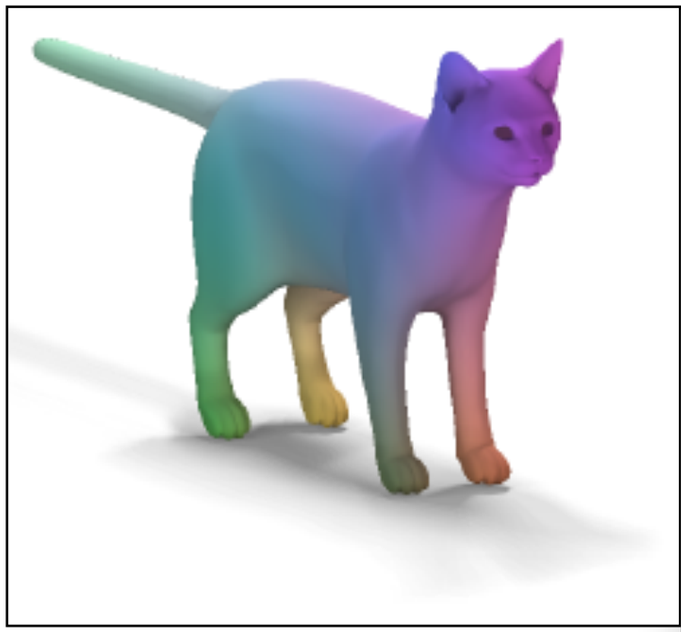


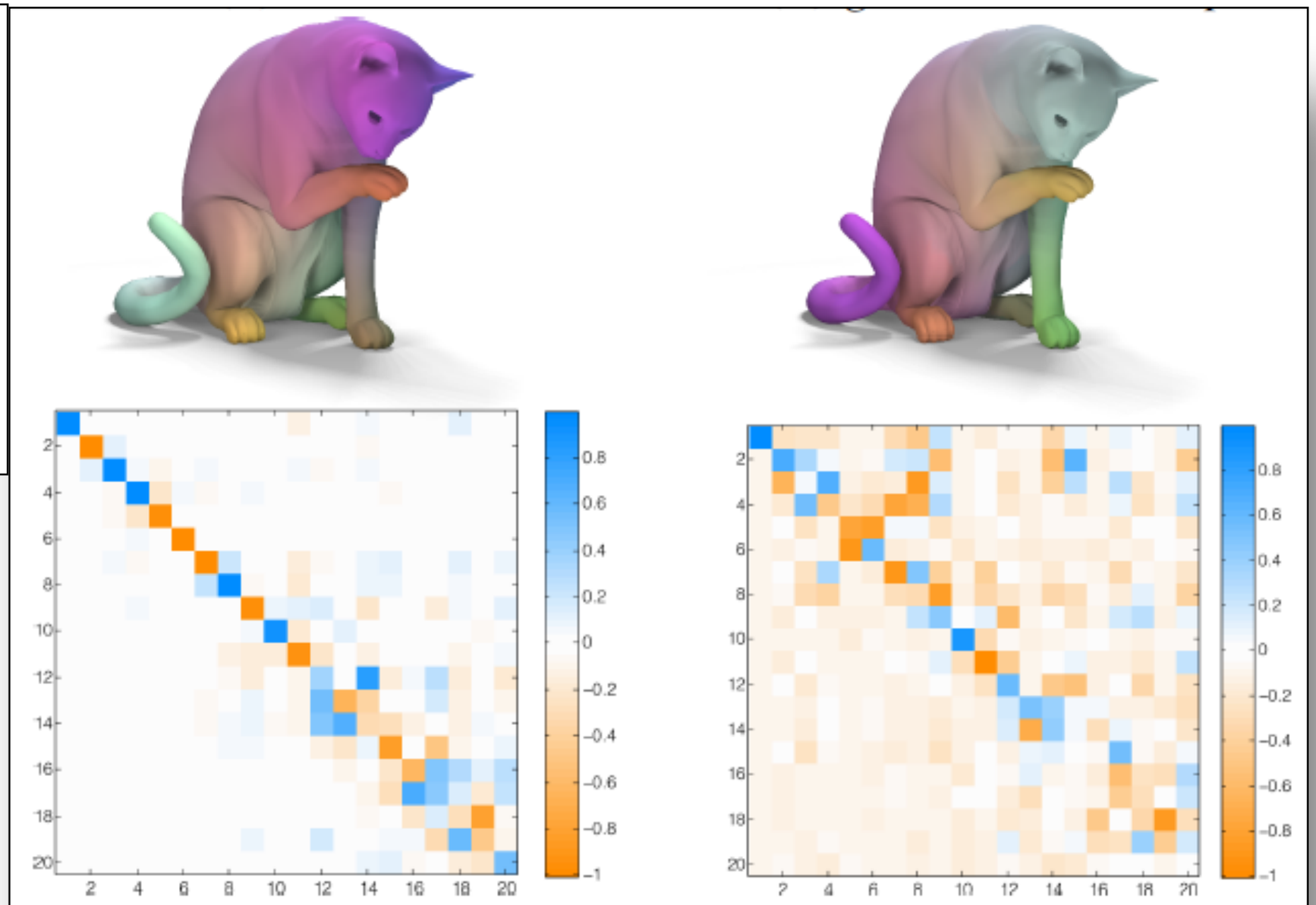


(c) *left to right map*

Ovsjanikov'12

Example Maps

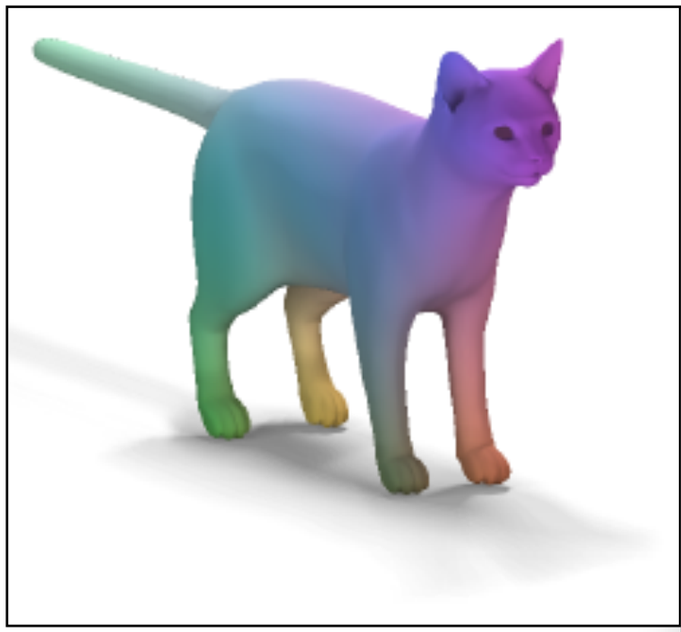


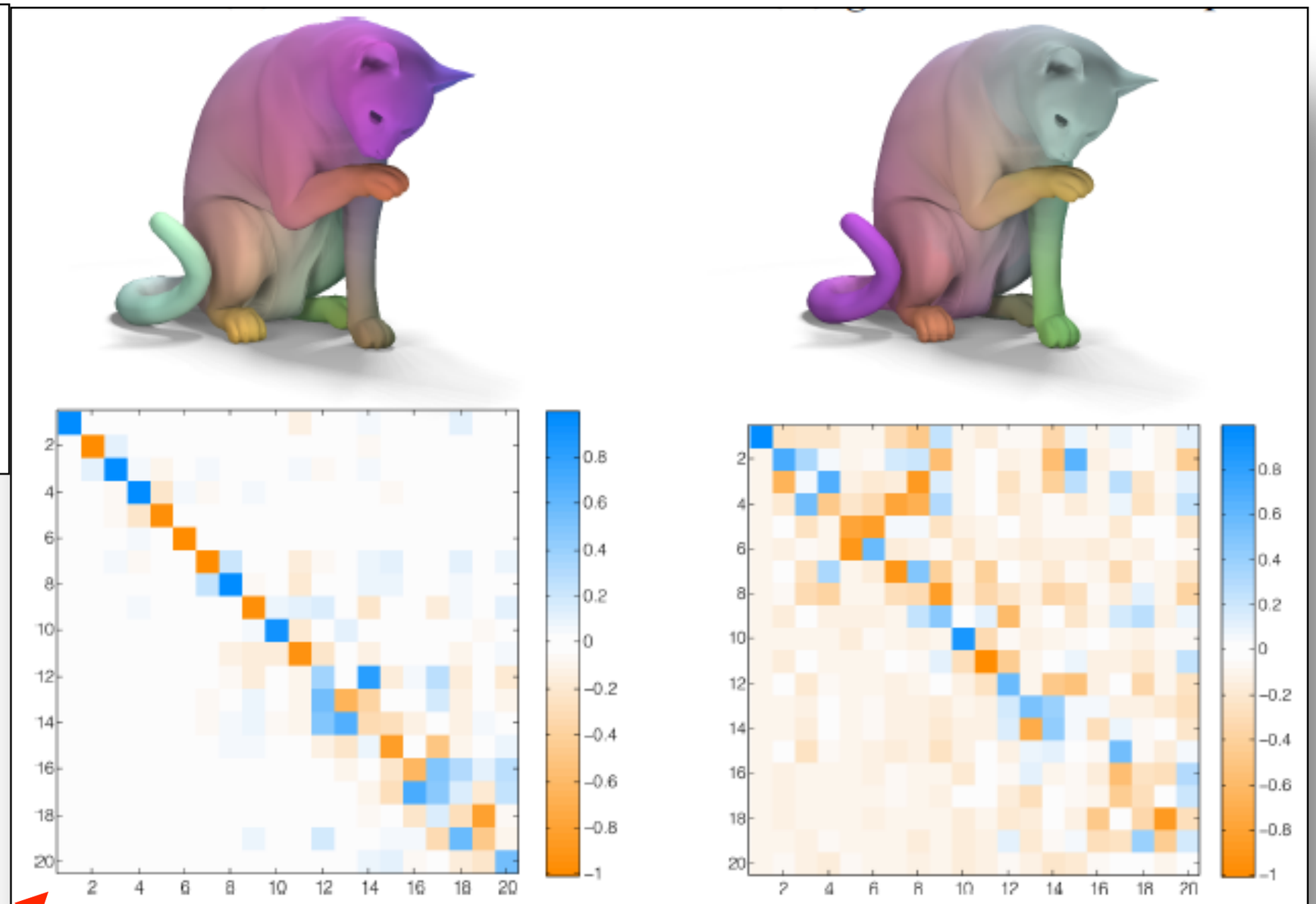


(c) *left to right map* **(d)** *head to tail map*

Ovsjanikov'12

Example Maps





(c) *left to right map* (d) *head to tail map*

The reason it looks like identity is not by chance!

Ovsjanikov'12

Functional Maps

Simple Algorithm

- Compute some geometric functions to be preserved: A, B
- Solve in least-squares sense for C, $B = C\,A$

Additional Considerations

- Favor commutativity
- Favor orthonormality (if shapes are isometric)
- Efficiently getting point-to-point correspondences

Ovsjanikov'12

Functional Maps

Pros

- Sparse representation
- Linear
- New way of thinking about maps

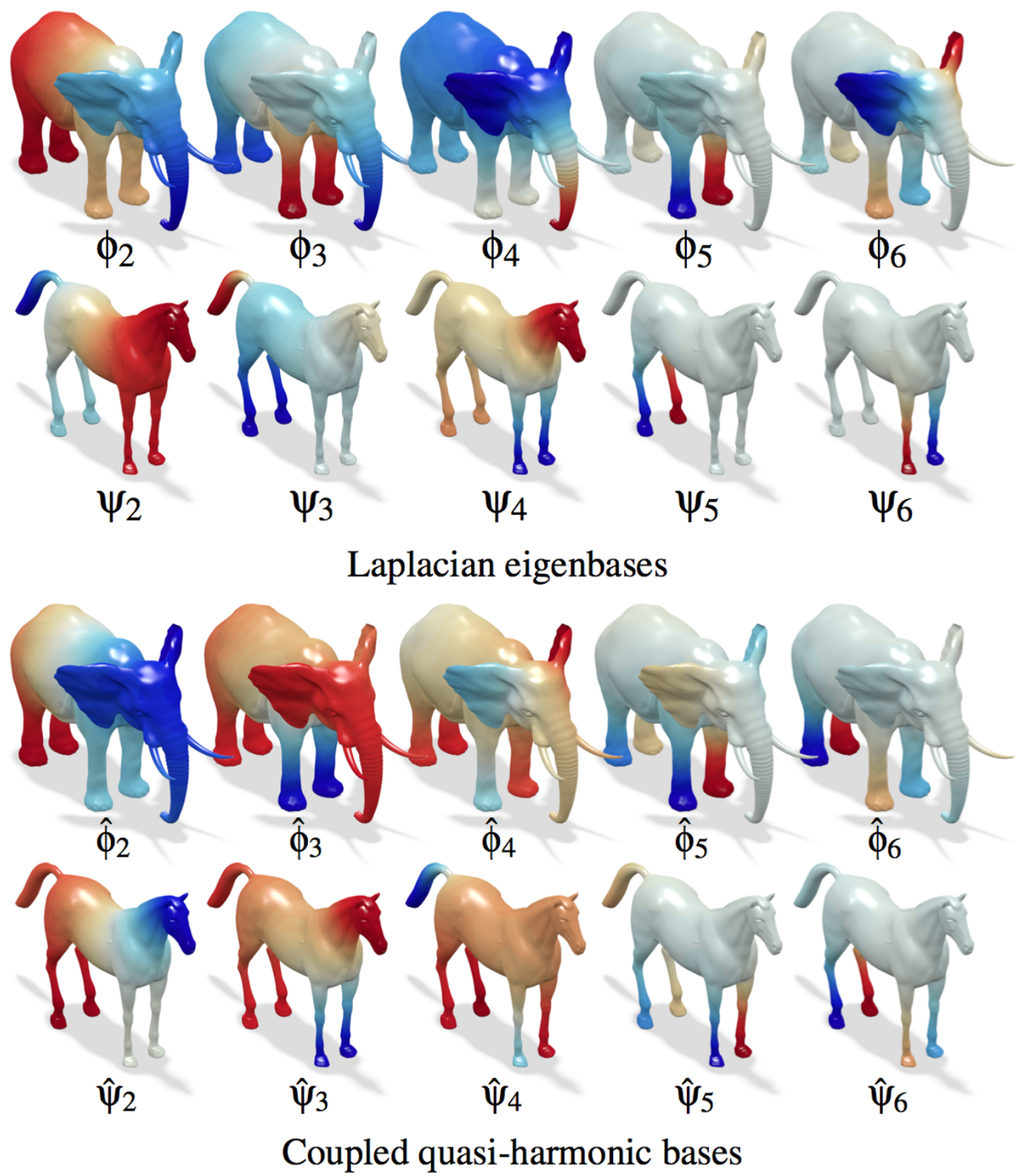
Cons

- More work is needed for non-isometric surfaces

Ovsjanikov'12

Choice of Basis

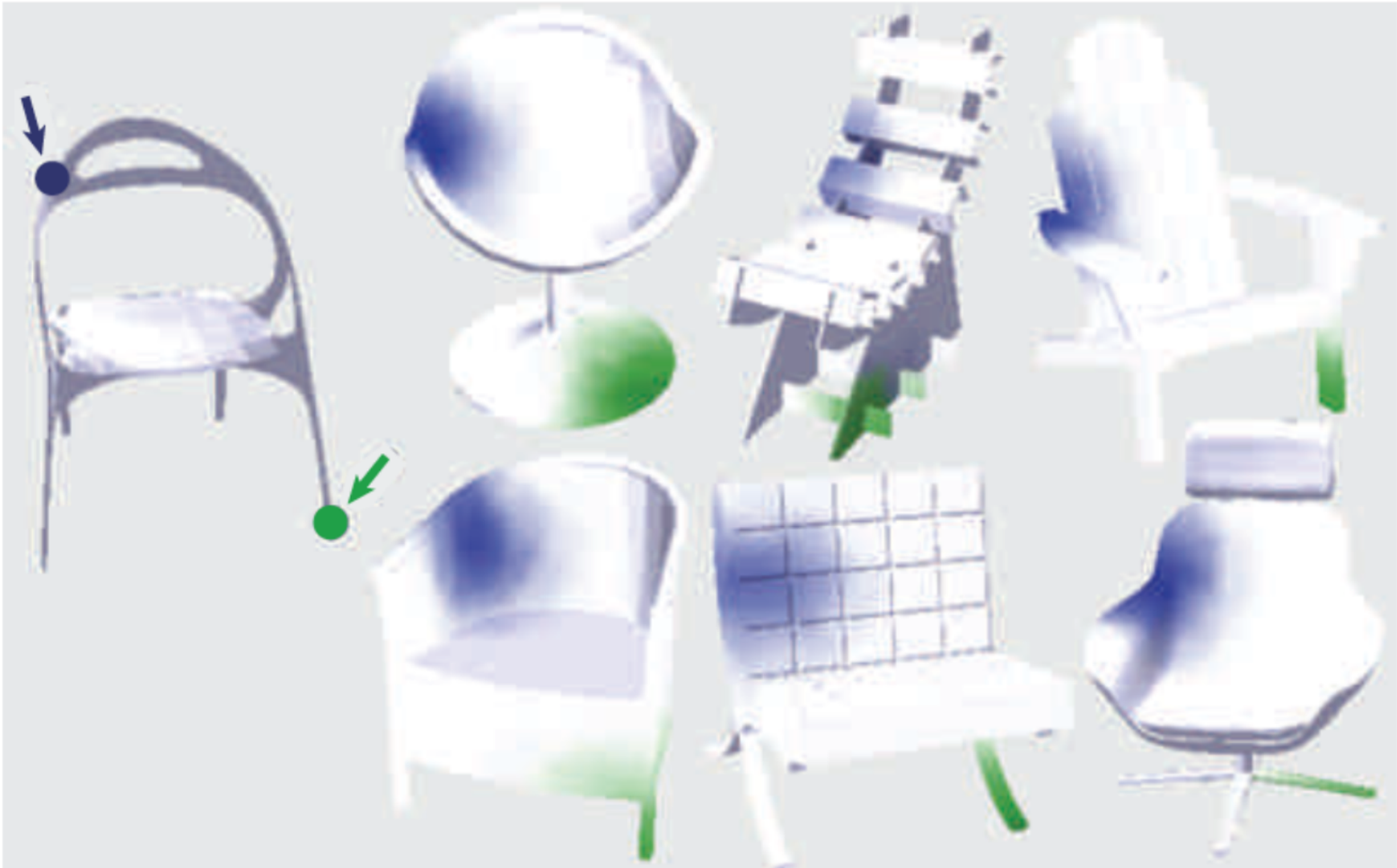
Coupled quasi-harmonic basis

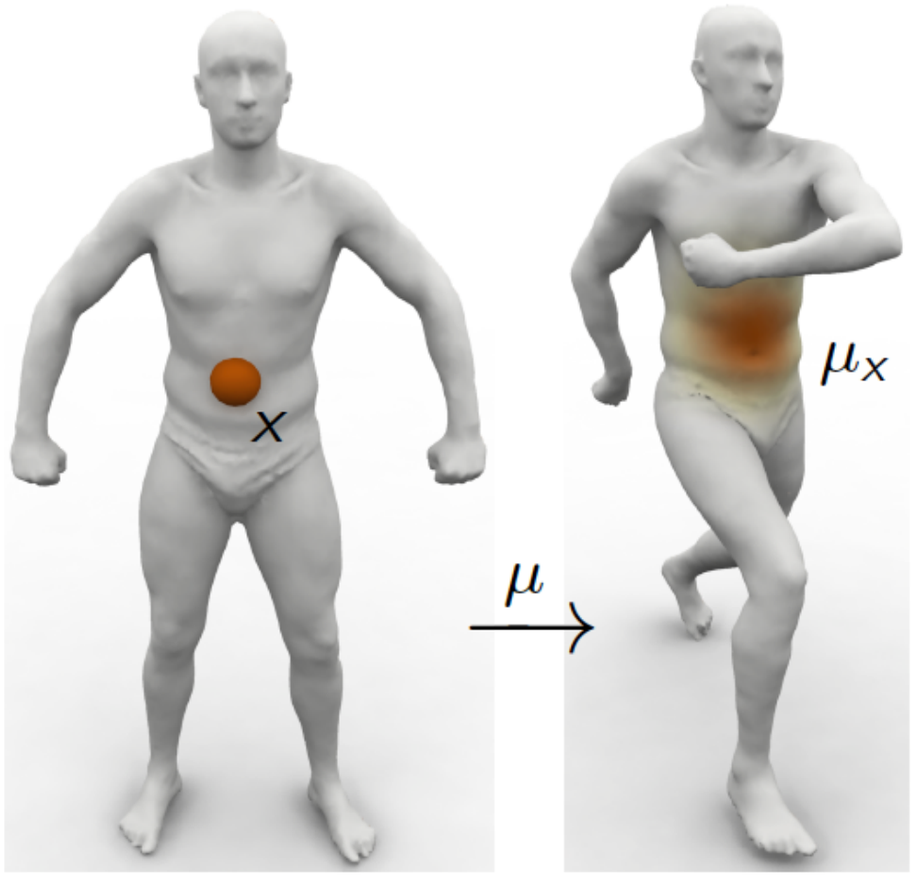


Bronstein'12

Soft/Fuzzy Maps

Points mapping to probability distributions to cope with mapping ambiguity

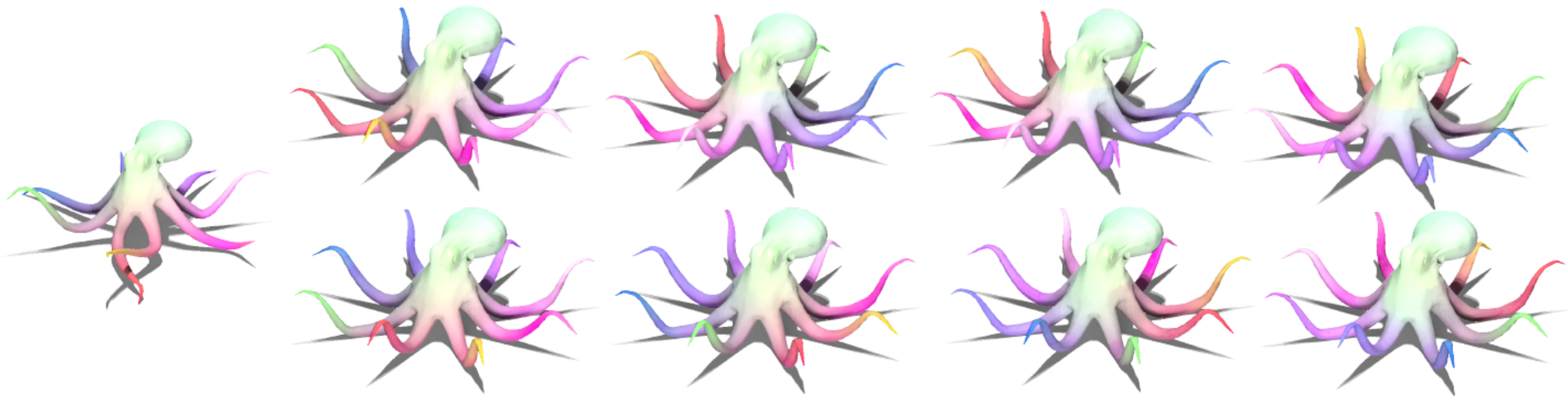




Kim'12, Solomon'12, Solomon'13

Mapping Symmetric Shapes

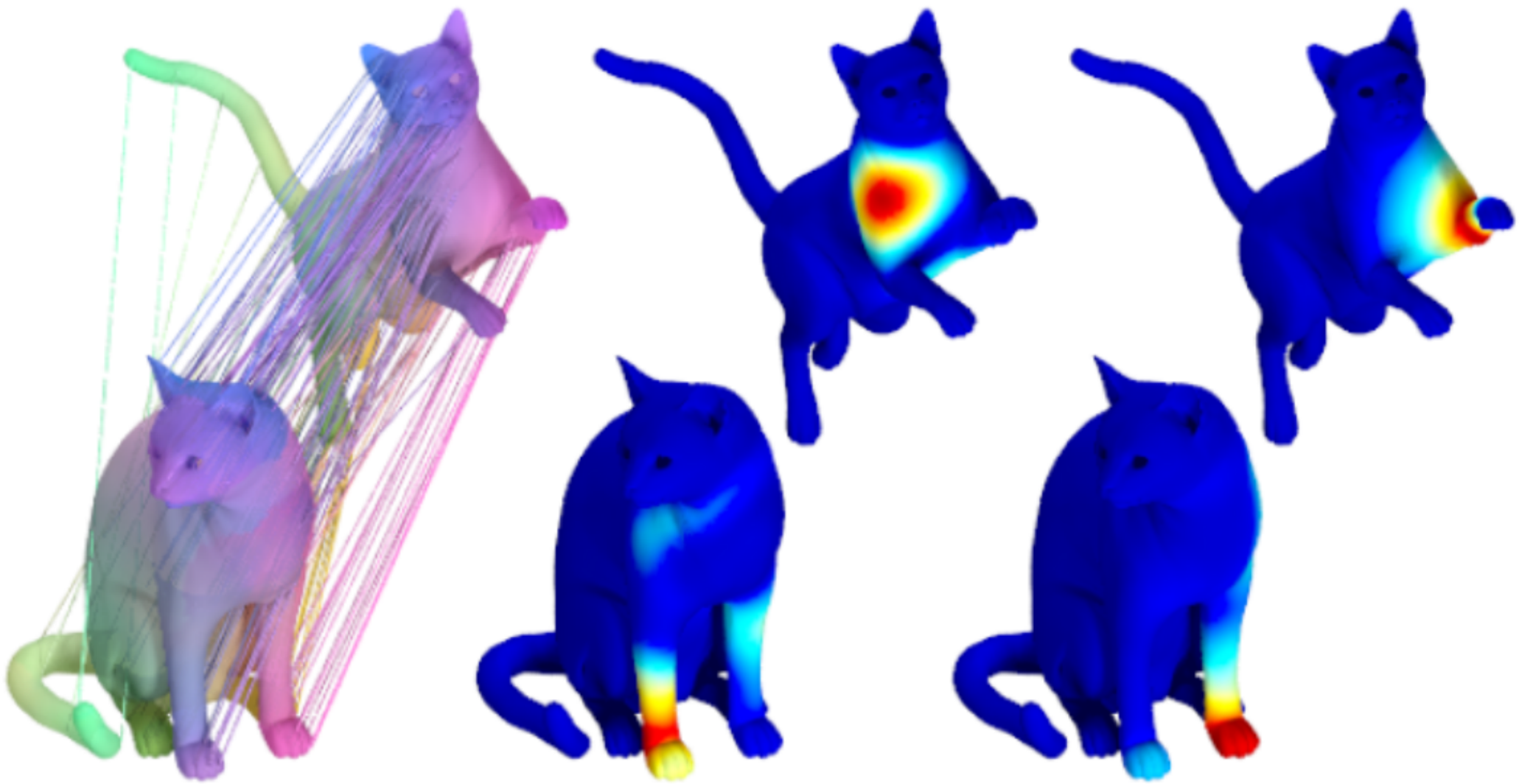
Separate basis into symmetric and anti-symmetric part



Ovsjanikov'13

Map Visualization

Find high-distortion areas in multi-scale fashion



Ovsjanikov'13

Homework

Notes

- Due Apr 20
- Include some info on how to use your program
- Include relevant sourcecode files and brief description
- Send a link to a zip file to vk2@stanford.edu
  (or e-mail directly if size < 5Mb)
- ALWAYS mark axes in ALL plots / figures!

- Questions?

References

- Cross-Parameterization and Compatible Remeshing of 3D Models.
  V. Kraevoy and A. Sheffer. SIGGRAPH 2004
- Consistent Mesh Parameterizations. E. Praun, W. Sweldens, P. Schröder. SIGGRAPH 2001
- Inter-Surface Mapping. J. Schreiner, A. P. Asirvatham, E. Praun, H. Hoppe. SIGGRAPH 2004
- Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching.
  A. M. Bronstein, M. M. Bronstein, R. Kimmel. PNAS 2006
- One Point Isometric Matching with the Heat Kernel.
  M. Ovsjanikov, Q. Mérigot, F. Mémoli and L. Guibas. SGP 2010
- Moebius Voting for Surface Correspondence. Y. Lipman, T. Funkhouser. SIGGRAPH 2009
- Blended Intrinsic Maps. V. Kim, Y. Lipman, T. Funkhouser. SIGGRAPH 2011
- Functional Maps: A Flexible Representation of Maps Between Shapes.
  M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher and L. Guibas. SIGGRAPH 2012
- Coupled quasi-harmonic bases.
  A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, R. Kimmel. Eurographics 2012.
- Dirichlet Energy for Analysis and Synthesis of Soft Maps. J. Solomon, L. Guibas, and A. Butscher. SGP 2013
- Shape Matching via Quotient Spaces.
  M. Ovsjanikov, Q. Mérigot, V. Pătrăucean, L. Guibas. SGP 2013
- Analysis and Visualization of Maps Between Shapes.
  M. Ovsjanikov, M. Ben-Chen, F. Chazal and L. Guibas. CGF 2013