Computer Graphics

Shape matching

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2018/2019 · May 09, 2018

Rilevazione Opinioni Studenti

Il questionario è completamente anonimo

Istruzioni:

- Disattivare il “blocco pop-up” del browser
- Visitare il sito https://www.uniroma1.it
- Cliccare su **Studenti** per accedere a **Infostud 2.0**
- Cliccare su **Corsi di laurea**
- Nel menu di sinistra, selezionare **Opinioni studenti**
- Inserire il codice OPIS e cliccare su **Vai al questionario**

Il codice OPIS è:

85G3I2HQ

Correspondence problems

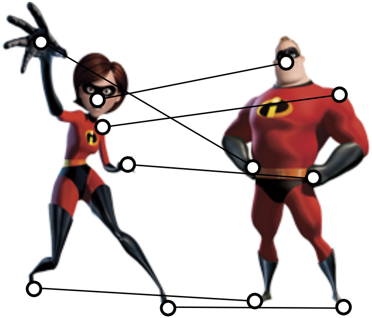
We can compare shapes, but...

$\stackrel{?}{=}$

Correspondence problems

We can compare shapes, but...

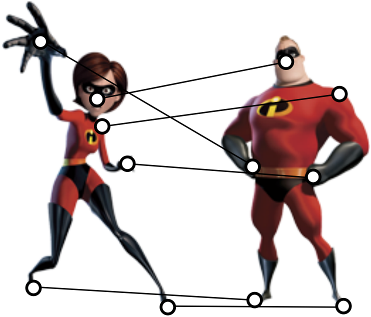
How to match them, point by point?



Correspondence problems

We can compare shapes, but...

How to match them, point by point?

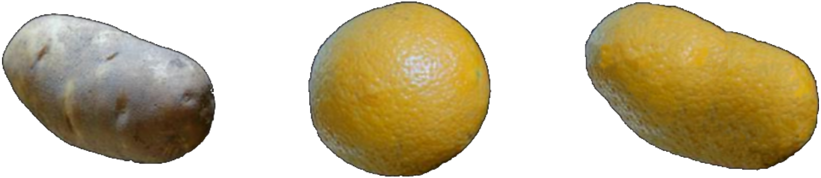


This is known as a matching, assignment, or correspondence problem

Still an open and challenging area of research!

Example: Texture transfer

With a correspondence we can transfer information (e.g. texture coordinates) across shapes

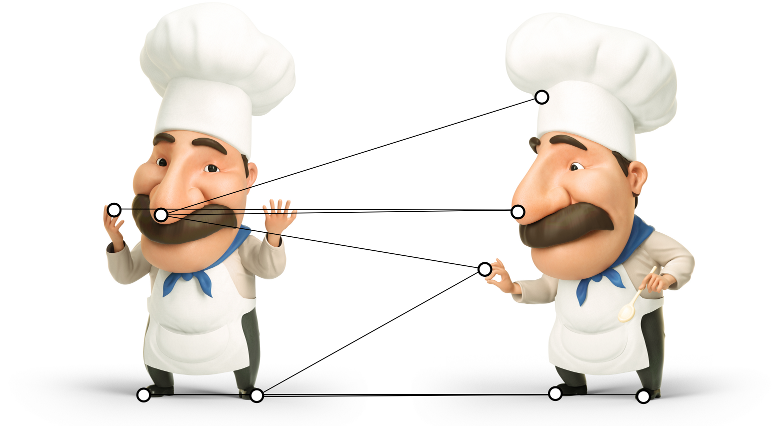


$\mathcal{X}$ $\mathcal{Y}$ texture of $\mathcal{Y}$ transferred to $\mathcal{X}$

Example: Texture transfer

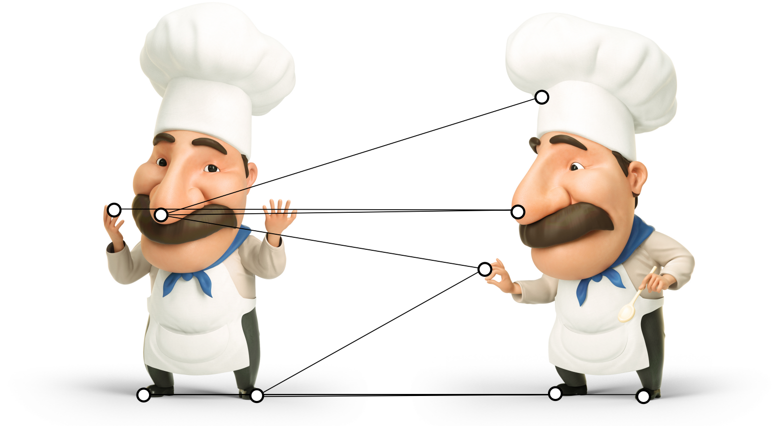
With a correspondence we can transfer information (e.g. texture coordinates) across shapes

Correspondence



Correspondence: at least one match for each sample (that is, not necessarily 1-to-1)

Correspondence



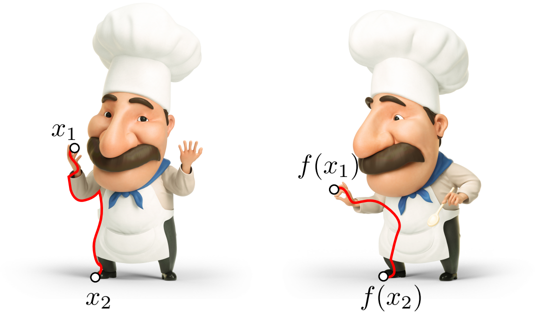
Correspondence: at least one match for each sample (that is, not necessarily 1-to-1)

By itself, it doesn't have anything to do with distortion or semantics

Metric distortion

For a map $f : \mathcal{X} \to \mathcal{Y}$ between metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_Y)$ we define its (absolute) distortion as:

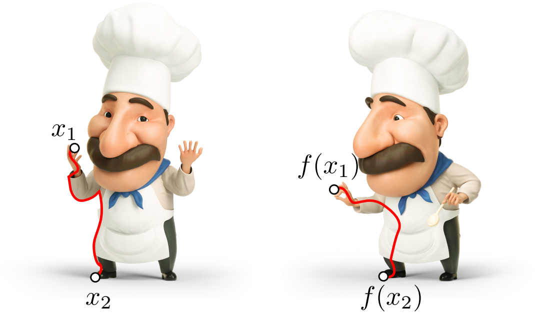
$$\mathrm{dis}(f) = \max_{x_1, x_2 \in \mathcal{X}} |d_{\mathcal{X}}(x_1, x_2) - d_{\mathcal{Y}}(f(x_1), f(x_2))|$$



Metric distortion

For a map $f : \mathcal{X} \to \mathcal{Y}$ between metric spaces $(\mathcal{X}, d_\mathcal{X})$ and $(\mathcal{Y}, d_Y)$ we define its (absolute) distortion as:

$$\mathrm{dis}(f) = \max_{x_1, x_2 \in \mathcal{X}} |d_\mathcal{X}(x_1, x_2) - d_\mathcal{Y}(f(x_1), f(x_2))|$$



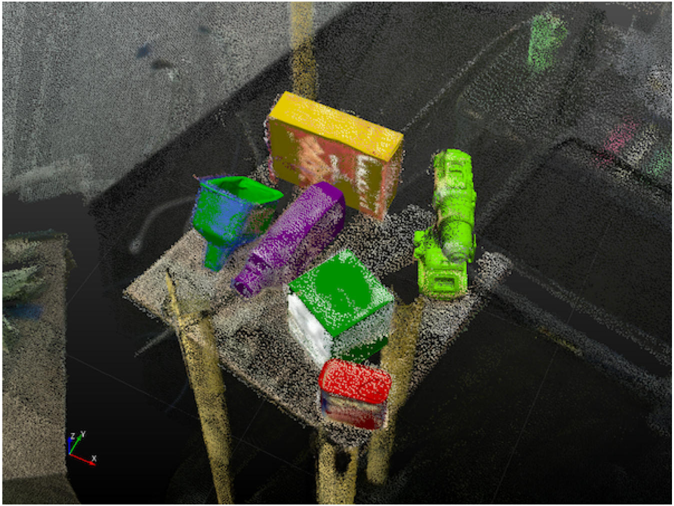
Similarly, for a correspondence $\mathcal{R}$ we define:

$$\mathrm{dis}(\mathcal{R}) = \max\{|d_\mathcal{X}(x_1, x_2) - d_\mathcal{Y}(y_1, y_2)| \ : \ (x_1, y_1), (x_2, y_2) \in \mathcal{R}\}$$

Example: Rigid correspondence

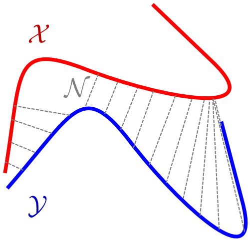
This is a simple case where $d_{\mathcal{X}} = d_{\mathcal{Y}} = \| \cdot \|_2$

Zero distortion means that the shapes are rigid



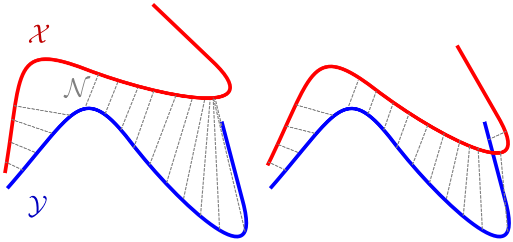
Example: ICP

At each step, we have both a correspondence and a rigid transformation



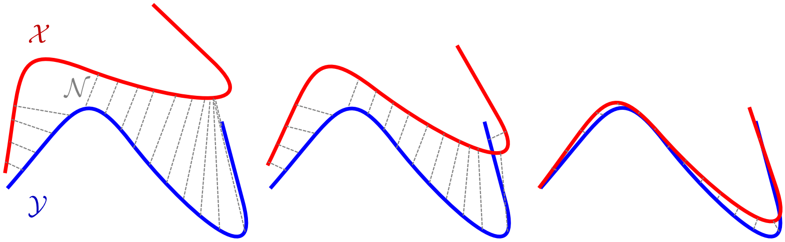
Example: ICP

At each step, we have both a correspondence and a rigid transformation



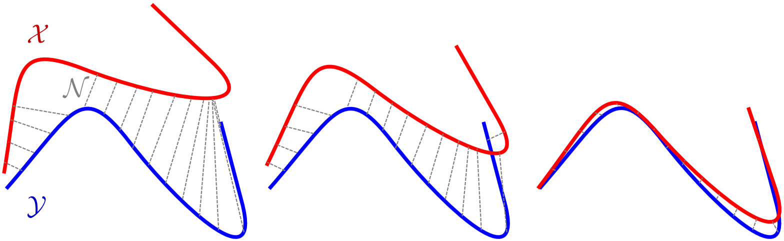
Example: ICP

At each step, we have both a correspondence and a rigid transformation



Example: ICP

At each step, we have both a correspondence and a rigid transformation



- In practice converges to local optima
- Requires a good initialization
- Only works well in the rigid setting

Example: Sort by distortion

If we know the correspondence π, then we can directly compute:

$$d(\mathcal{X}, \mathcal{Y}) = \max\{|d_{\mathcal{X}}(x_1, x_2) - d_{\mathcal{Y}}(y_1, y_2)| \ : \ (x_1, y_1), (x_2, y_2) \in \mathcal{R}\}$$

Example: Sort by distortion

If we know the correspondence π, then we can directly compute:

$$d(\mathcal{X}, \mathcal{Y}) = \max\{|d_{\mathcal{X}}(x_1, x_2) - d_{\mathcal{Y}}(y_1, y_2)| \ : \ (x_1, y_1), (x_2, y_2) \in \mathcal{R}\}$$

And we can use it to sort shapes by distortion:

$\mathcal{X}$ $\mathcal{Y}_1$ $\mathcal{Y}_2$ $\mathcal{Y}_3$ $\mathcal{Y}_4$

where

$$d_{\mathcal{P}}(\mathcal{X}, \mathcal{Y}_1) \leq d_{\mathcal{P}}(\mathcal{X}, \mathcal{Y}_2) \leq d_{\mathcal{P}}(\mathcal{X}, \mathcal{Y}_3) \leq d_{\mathcal{P}}(\mathcal{X}, \mathcal{Y}_4)$$

Correspondence matrix

A common representation is the correspondence matrix $\mathbf{T} \in \{0,1\}^{m \times n}$

$$\mathbf{T} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 1 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

where $T_{i,j} = 1$ means that $(x_i, y_j) \in \mathcal{R} \subset \mathcal{X} \times \mathcal{Y}$

Correspondence matrix

A common representation is the correspondence matrix $\mathbf{T} \in \{0,1\}^{m\times n}$

$$\mathbf{T} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 1 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

where $T_{i,j} = 1$ means that $(x_i, y_j) \in \mathcal{R} \subset \mathcal{X} \times \mathcal{Y}$

$\mathbf{T}$ encodes in the standard basis a linear map between function spaces:

$$T : (\mathcal{F}(\mathcal{X}) \to \mathbb{R}) \to (\mathcal{F}(\mathcal{Y}) \to \mathbb{R})$$

In other words, it maps functions to functions

Permutation matrix

A permutation matrix $\mathbf{P} \in \{0,1\}^{n\times n}$ is a special case where each row and each column have exactly one value equal to 1:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Permutation matrix

A permutation matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ is a special case where each row and each column have exactly one value equal to 1:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

This can be also written as $\mathbf{P1} = \mathbf{1}, \mathbf{P}^\top \mathbf{1} = \mathbf{1}$

Permutation matrix

A permutation matrix $\mathbf{P} \in \{0,1\}^{n\times n}$ is a special case where each row and each column have exactly one value equal to 1:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

This can be also written as $\mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^\top\mathbf{1} = \mathbf{1}$

Permutations encode one-to-one correspondences, meaning that they map indicator functions to indicator functions

Continuous relaxation

Replace $\{0, 1\}$ with continuous values in $[0, 1]$

We now have doubly-stochastic matrices where $\mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S}^\top\mathbf{1} = \mathbf{1}$:

$$\mathbf{S} = \begin{pmatrix} 0.2 & 0.3 & \cdots & 0.5 \\ 0.1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0.7 & 0 & \cdots & 0.1 \end{pmatrix}$$

Continuous relaxation

Replace $\{0,1\}$ with continuous values in $[0,1]$

We now have doubly-stochastic matrices where $\mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S}^\top\mathbf{1} = \mathbf{1}$:

$$\mathbf{S} = \begin{pmatrix} 0.2 & 0.3 & \cdots & 0.5 \\ 0.1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0.7 & 0 & \cdots & 0.1 \end{pmatrix}$$

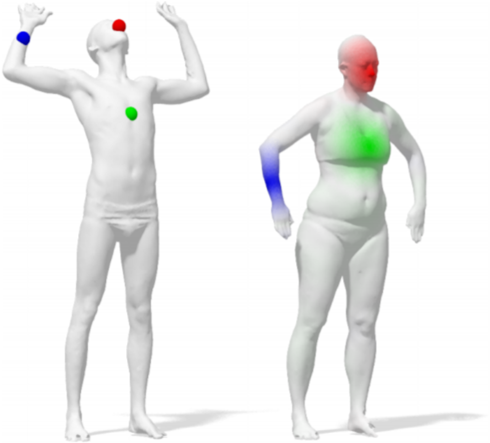
This is also called a soft map

Doubly-stochastic matrices are convex combinations of permutations:

$$\mathbf{S} = t\mathbf{P}_1 + (1-t)\mathbf{P}_2\,, \quad t \in [0,1]$$

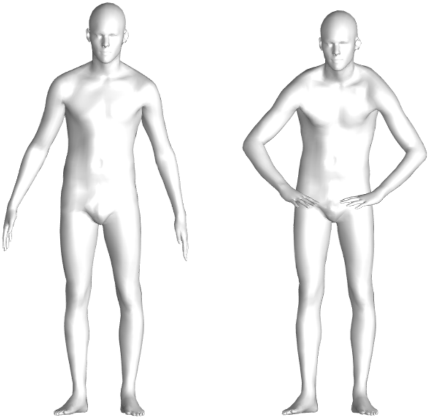
Example: Soft maps

Each row / column of a soft map can be interpreted as a probability distribution on the shape:



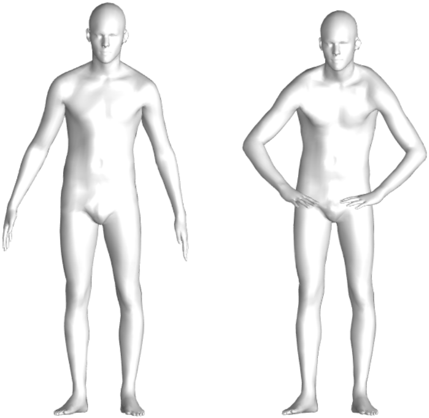
Visualizing correspondence

How to visualize correspondence between shapes?



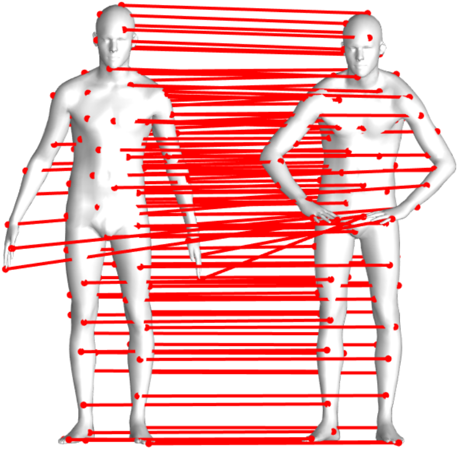
Visualizing correspondence

How to visualize correspondence between shapes?

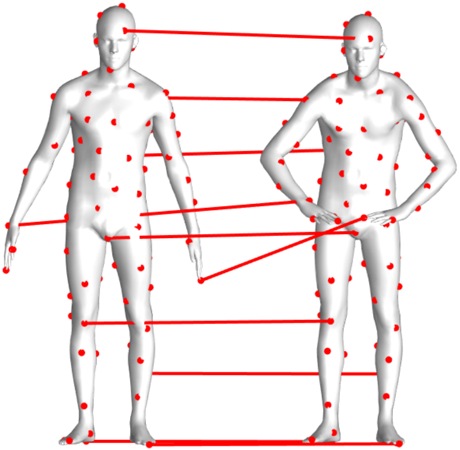


- It might be not one-to-one
- It might be sparse

Visualizing correspondence: Points and lines

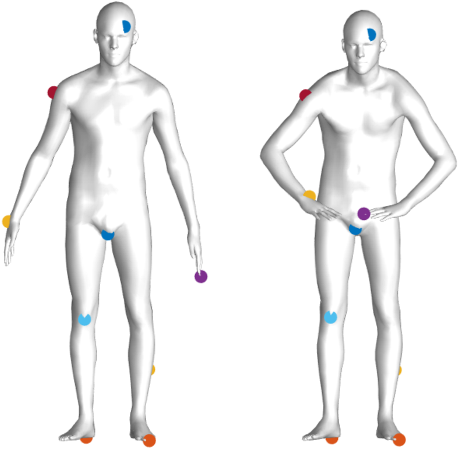


Visualizing correspondence: Points and lines



Useful for visualizing sparse matches

Visualizing correspondence: Points and colors



Useful for visualizing sparse matches (e.g., between landmarks)

Visualizing correspondence: Dense colors

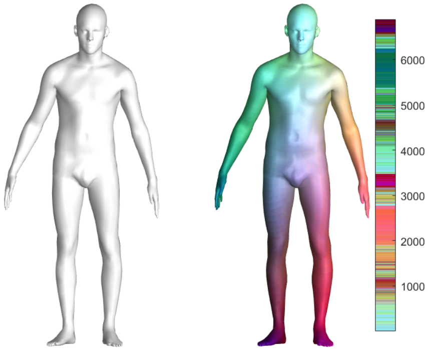
Dense matches ($\sim$all points) can be used to color the entire mesh

- Create a colormap with one color for each point ($n \times 3$ matrix)

Visualizing correspondence: Dense colors

Dense matches ($\sim$all points) can be used to color the entire mesh

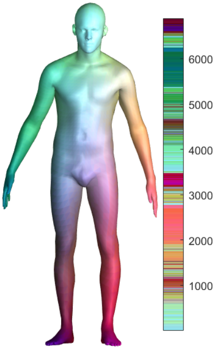
- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$



Visualizing correspondence: Dense colors

Dense matches ($\sim$all points) can be used to color the entire mesh

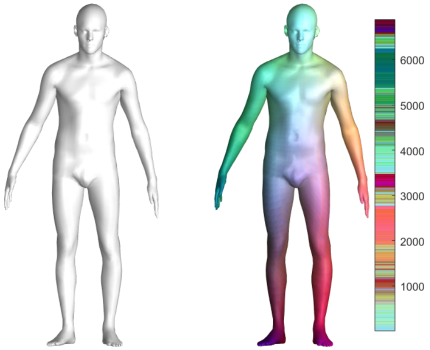
- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$
- For smooth colors: interpret x, y, z as R, G, B



Visualizing correspondence: Dense colors

Dense matches ($\sim$all points) can be used to color the entire mesh

- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$
- For smooth colors: interpret x, y, z as R, G, B

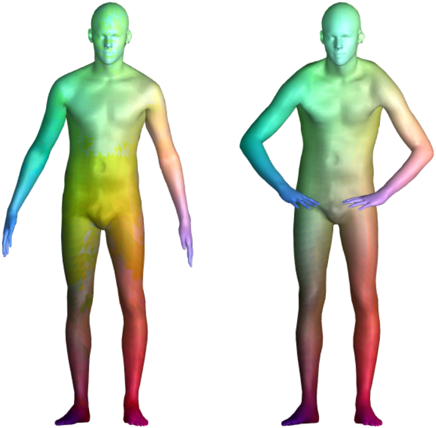


Warning: New versions of Matlab require shading `flat` (not `interp`)

Visualizing correspondence: Dense colors

Note: The map may be not 1-to-1

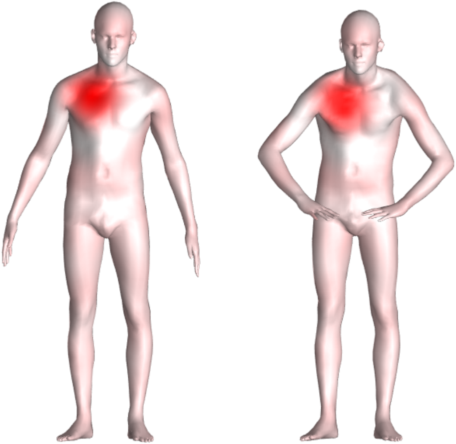
Trick: Obtain the source colormap by pulling back the target colormap



This hides the lack of surjectivity in the correspondence

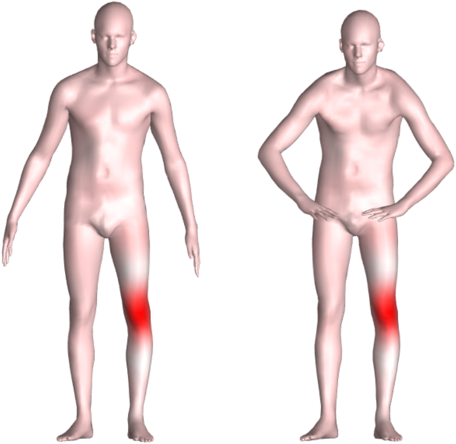
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



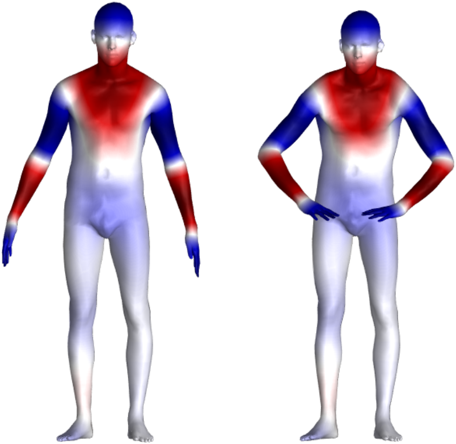
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



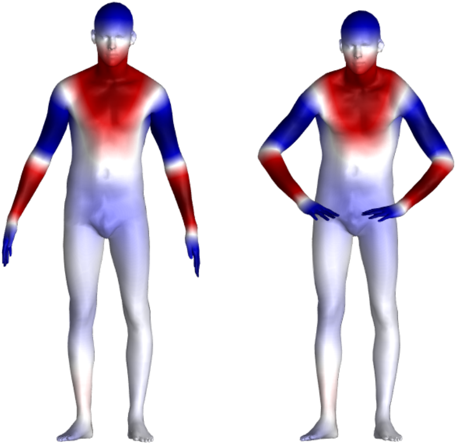
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



This also works for non-surjective and in general soft maps

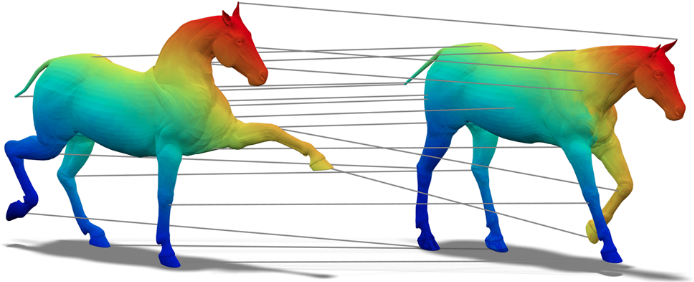
Maps in the Laplacian eigenbases

Consider two isometric shapes $\mathcal{X}, \mathcal{Y}$ with Laplacian eigenfunctions $\{\phi_i\}, \{\psi_j\}$ respectively spanning the functional spaces $\mathcal{F}(\mathcal{X}), \mathcal{F}(\mathcal{Y})$.

Maps in the Laplacian eigenbases

Consider two isometric shapes $\mathcal{X}, \mathcal{Y}$ with Laplacian eigenfunctions $\{\phi_i\}, \{\psi_j\}$ respectively spanning the functional spaces $\mathcal{F}(\mathcal{X}), \mathcal{F}(\mathcal{Y})$.

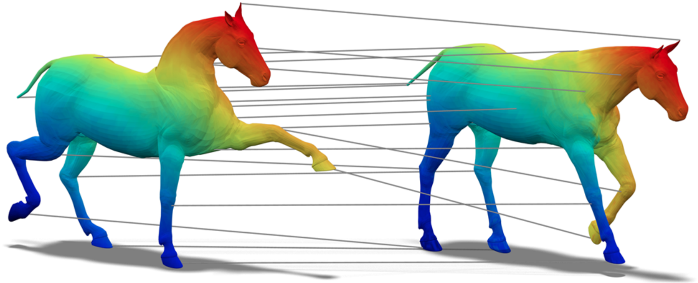
Now let $T : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{Y})$ be the ground-truth map.



Maps in the Laplacian eigenbases

Consider two isometric shapes $\mathcal{X}, \mathcal{Y}$ with Laplacian eigenfunctions $\{\phi_i\}, \{\psi_j\}$ respectively spanning the functional spaces $\mathcal{F}(\mathcal{X}), \mathcal{F}(\mathcal{Y})$.

Now let $T : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{Y})$ be the ground-truth map.



How does the matrix representation of T look like, in the Laplacian eigenbases?

Maps in arbitrary (orthogonal) bases

$f : \mathcal{X} \to \mathbb{R}$

$$g = Tf$$

$g : \mathcal{Y} \to \mathbb{R}$

Maps in arbitrary (orthogonal) bases

$f : \mathcal{X} \to \mathbb{R}$

$$\mathbf{g} = \mathbf{T}\mathbf{f}$$

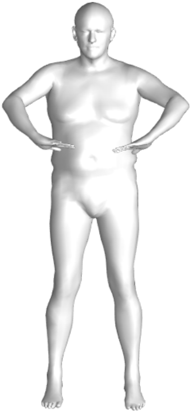
$g : \mathcal{Y} \to \mathbb{R}$

Maps in arbitrary (orthogonal) bases

$f : \mathcal{X} \to \mathbb{R}$

ortho. basis $\mathbf{A}$

$$\mathbf{g} = \mathbf{T}\mathbf{f}$$



$g : \mathcal{Y} \to \mathbb{R}$

ortho. basis $\mathbf{B}$

Maps in arbitrary (orthogonal) bases

$f : \mathcal{X} \to \mathbb{R}$

ortho. basis $\mathbf{A}$

$$(\mathbf{B}\mathbf{B}^\top \mathbf{g}) = \mathbf{T}(\mathbf{A}\mathbf{A}^\top \mathbf{f})$$

$g : \mathcal{Y} \to \mathbb{R}$

ortho. basis $\mathbf{B}$

Maps in arbitrary (orthogonal) bases

$$\mathbf{B}^\top \mathbf{g} = (\mathbf{B}^\top \mathbf{T} \mathbf{A}) \mathbf{A}^\top \mathbf{f}$$

$f : \mathcal{X} \to \mathbb{R}$

ortho. basis $\mathbf{A}$

$g : \mathcal{Y} \to \mathbb{R}$

ortho. basis $\mathbf{B}$

Maps in arbitrary (orthogonal) bases

$f : \mathcal{X} \to \mathbb{R}$

ortho. basis $\mathbf{A}$

$$\mathbf{B}^\top \mathbf{g} = \mathbf{C}\mathbf{A}^\top \mathbf{f}$$

matrix $\mathbf{C} = \mathbf{B}^\top \mathbf{T}\mathbf{A}$
is map T wrt $\mathbf{A}, \mathbf{B}$ bases

$g : \mathcal{Y} \to \mathbb{R}$

ortho. basis $\mathbf{B}$

Maps in arbitrary (orthogonal) bases

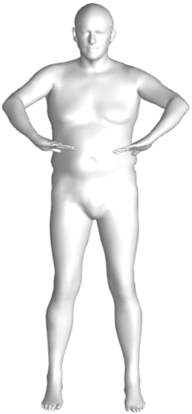
$f : \mathcal{X} \to \mathbb{R}$

ortho. basis $\mathbf{A}$

$$\mathbf{g} = \mathbf{B}\mathbf{C}\mathbf{A}^\top \mathbf{f}$$

matrix $\mathbf{C} = \mathbf{B}^\top \mathbf{T} \mathbf{A}$
is map T wrt $\mathbf{A}, \mathbf{B}$ bases

matrix $\mathbf{T} = \mathbf{B}\mathbf{C}\mathbf{A}^\top$
goes back to the std bases

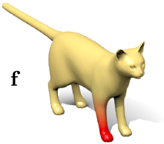


$g : \mathcal{Y} \to \mathbb{R}$

ortho. basis $\mathbf{B}$

Maps in the Laplacian eigenbases

$\mathbf{f}$



$\downarrow$

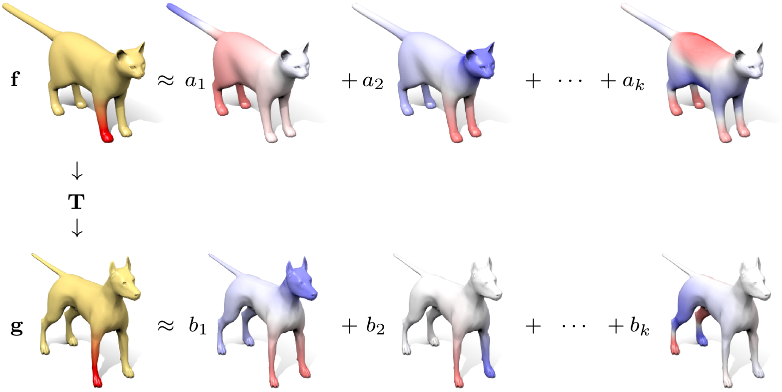
$\mathbf{T}$

$\downarrow$

$\mathbf{g}$

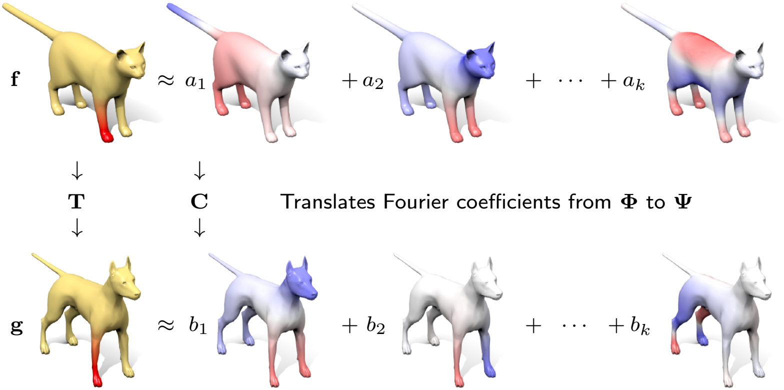
Ovsjanikov et al. 2012

Maps in the Laplacian eigenbases



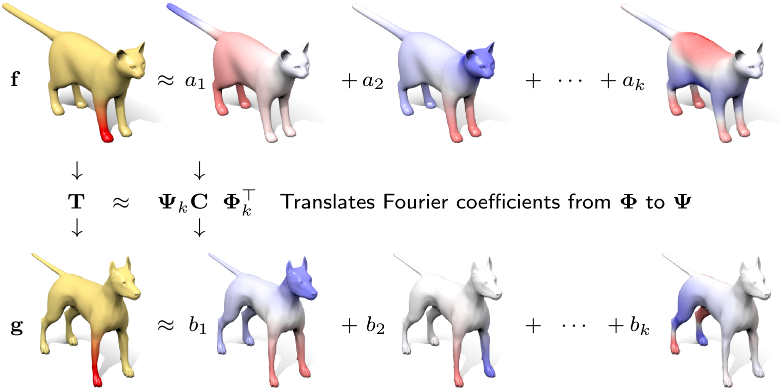
Ovsjanikov et al. 2012

Maps in the Laplacian eigenbases



Ovsjanikov et al. 2012

Maps in the Laplacian eigenbases



$$\boldsymbol{\Psi}_k^\top \mathbf{g} = \mathbf{C} \boldsymbol{\Phi}_k^\top \mathbf{f}$$

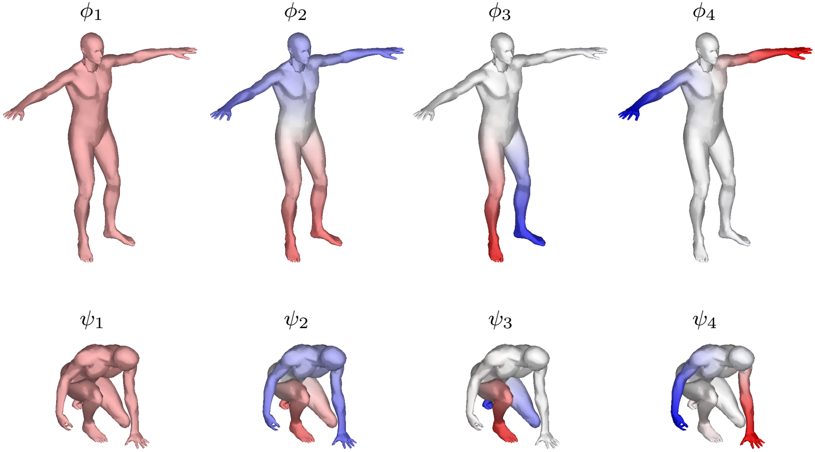
where $\boldsymbol{\Phi}_k = (\phi_1, \ldots, \phi_k)$, $\boldsymbol{\Psi}_k = (\psi_1, \ldots, \psi_k)$ are LB eigenbases

Note: The bases are truncated to the first k eigenfunctions

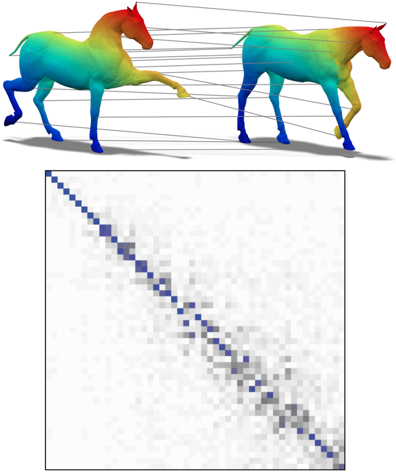
Ovsjanikov et al. 2012

Laplacian eigenfunctions

The Laplacian is invariant to isometries



Maps in the Laplacian eigenbases



$$\mathbf{C} = \mathbf{\Psi}_k^\top \mathbf{T} \mathbf{\Phi}_k \Rightarrow c_{ij} = \langle \psi_i, T\varphi_j \rangle$$

For isometric shapes, $\mathbf{C}$ is diagonal since $\boldsymbol{\psi}_i = \pm \mathbf{T} \boldsymbol{\phi}_i$

Rank of a map

In the standard basis, a one-to-one correspondence is written as a permutation matrix in $\mathbb{R}^{n\times n}$

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Rank of a map

In the standard basis, a one-to-one correspondence is written as a permutation matrix in $\mathbb{R}^{n\times n}$

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Each column is a basis vector, so $\operatorname{rank}(\mathbf{P}) = n$

Rank of a map

In the Laplacian eigenbasis, a correspondence is written as a generic matrix in $\mathbb{R}^{k\times k}$

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & & & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kk} \end{pmatrix}$$

Rank of a map

In the Laplacian eigenbasis, a correspondence is written as a generic matrix in $\mathbb{R}^{k\times k}$

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & & & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kk} \end{pmatrix}$$

Due to truncation, now $\mathrm{rank}(\mathbf{C}) \leq k \ll n$

Rank of a map

In the Laplacian eigenbasis, a correspondence is written as a generic matrix in $\mathbb{R}^{k \times k}$

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & & & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kk} \end{pmatrix}$$

Due to truncation, now $\mathrm{rank}(\mathbf{C}) \leq k \ll n$

Functions mapped via $\mathbf{C}$ span a subspace of those mapped via $\mathbf{P}$

Solving for a map

$$\mathbf{C}\mathbf{\Phi}_k^\top \mathbf{f} = \mathbf{\Psi}_k^\top \mathbf{g}$$

Where f, g are corresponding functions (possibly noisy)

For example: GPS, HKS, indicators at corresponding points, etc.

Solving for a map

$$\mathbf{C}\mathbf{\Phi}_k^\top \mathbf{F} = \mathbf{\Psi}_k^\top \mathbf{G}$$

Where f, g are corresponding functions (possibly noisy)

For example: GPS, HKS, indicators at corresponding points, etc.

Solving for a map

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

Where f, g are corresponding functions (possibly noisy)

For example: GPS, HKS, indicators at corresponding points, etc.

Solving for a map

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

Where f, g are corresponding functions (possibly noisy)

For example: GPS, HKS, indicators at corresponding points, etc.

The correspondence $\mathbf{C}$ can now be solved for via least squares

$$\min_{\mathbf{C}} \|\mathbf{C}\hat{\mathbf{F}} - \hat{\mathbf{G}}\|_2^2$$

Solving for a map

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

Where f, g are corresponding functions (possibly noisy)

For example: GPS, HKS, indicators at corresponding points, etc.

The correspondence $\mathbf{C}$ can now be solved for via least squares

$$\min_{\mathbf{C}} \|\mathbf{C}\hat{\mathbf{F}} - \hat{\mathbf{G}}\|_2^2$$