Functional Maps
A Flexible Representation of Maps Between Shapes

Markus Sterflinger
Technische Universität München
14/5/2014
In the following:

- We don’t want to look at point-to-point or segment-to-segment representations.
- We are interested in a function, which maps functions on the one shape to functions on the other. Or: How should we paint the second cat?
Eigenfunctions of the Laplace-Beltrami operator form a basis of the functions on a manifold:

\[ f = \sum_{i=0}^{\infty} \langle f, \phi_i \rangle \phi_i \]
Let $M$, $N$ be two shapes and $T : M \rightarrow N$ a bijective mapping, $f : M \rightarrow \mathbb{R}$ a scalar function. Then we can define $g : N \rightarrow \mathbb{R}$ by $g = f \circ T^{-1}$. 
Let $M, N$ be two shapes and $T : M \rightarrow N$ a bijective mapping, $f : M \rightarrow \mathbb{R}$ a scalar function. Then we can define $g : N \rightarrow \mathbb{R}$ by $g = f \circ T^{-1}$.

Denote $T_F : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(N, \mathbb{R})$ as this transformation. $T_F$ is called the functional representation of $T$. 
Let $M$, $N$ be two shapes and $T : M \to N$ a bijective mapping, $f : M \to \mathbb{R}$ a scalar function. Then we can define $g : N \to \mathbb{R}$ by $g = f \circ T^{-1}$.

Denote $T_F : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$ as this transformation. $T_F$ is called the functional representation of $T$.

If $\mathcal{F}(M; \mathbb{R})$ and $\mathcal{F}(N; \mathbb{R})$ has a Basis $\{\phi_i^M\}$ and $\{\phi_i^N\}$ respectively, then we can represent any functions $f$ on $M$ and $g$ on $N$ as a vector $a = (a_0, a_1, a_2, \ldots)$ and $b = (b_0, b_1, b_2, \ldots)$, s.t.

$$f = \sum_{i=1}^{\infty} a_i \phi_i^M, \quad g = \sum_{i=1}^{\infty} b_i \phi_i^N$$
It follows:

**Theorem**

The map $T_F$ can be represented as a (possibly infinite) matrix $C$. In particular $T_F$ is linear.
It follows:

**Theorem**

*The map $T_F$ can be represented as a (possibly infinite) matrix $C$. In particular $T_F$ is linear.*

- $C$ is defined as $C_{i,j} = \langle T_F(\phi_i^M), \phi_j^N \rangle$
It follows:

**Theorem**

*The map $T_F$ can be represented as a (possibly infinite) matrix $C$. In particular $T_F$ is linear.*

- $C$ is defined as $C_{i,j} = \langle T_F(\phi_i^M), \phi_j^N \rangle$
- We write $T_F(f) = Ca$
Task:

Given two shapes $M$ and $N$. Find the underlying map $T$. 
Task

**Task:**
Given two shapes $M$ and $N$. Find the underlying map $T$.

**Approach:**
First recover $C$ and then calculate $T$. 
Task:

Given two shapes $M$ and $N$. Find the underlying map $T$.

Approach:

First recover $C$ and then calculate $T$.

- How can we recover $C$ efficiently?
Task:

Given two shapes $M$ and $N$. Find the underlying map $T$.

Approach:

First recover $C$ and then calculate $T$

- How can we recover $C$ efficiently?
- What properties has $C$?
Task

Given two shapes $M$ and $N$. Find the underlying map $T$.

**Approach:**
First recover $C$ and then calculate $T$

- How can we recover $C$ efficiently?
- What properties has $C$?
- How can we represent functions on the shapes efficiently?
Choice of basis: Straightforward

**Straightforward approach:** Basis of *Indicator functions* on the vertices
Choice of basis: Straightforward

**Straightforward approach:** Basis of **Indicator functions** on the vertices

- Every basis vector represents one vertex
Choice of basis: Straightforward

**Straightforward approach:** Basis of **Indicator functions** on the vertices

- Every basis vector represents one vertex
- Every bijective function $T : M \rightarrow N$ can be represented as a functional map
- $C$ becomes a permutation matrix. Corresponds to the original map
Choice of basis: Straightforward

**Straightforward approach:** Basis of **Indicator functions** on the vertices

- Every basis vector represents one vertex
- Every bijective function $T : M \rightarrow N$ can be represented as a functional map
- $C$ becomes a permutation matrix. Corresponds to the original map
Choice of basis: Straightforward

**Straightforward approach:** Basis of Indicator functions on the vertices

- Every basis vector represents one vertex
- Every bijective function $T : M \rightarrow N$ can be represented as a functional map
- $C$ becomes a permutation matrix. Corresponds to the original map

**BUT:**

- Many vertices $\Rightarrow$ Not compact
**Choice of basis: Straightforward**

**Straightforward approach:** Basis of **Indicator functions** on the vertices

- Every basisvector represents one vertex
- Every bijective function $T : M \rightarrow N$ can be represented as a functional map
- $C$ becomes a permutation matrix. Corresponds to the original map

**BUT:**

- Many vertices $\rightarrow$ Not compact
- Such a matrix could be stored much more efficiently
Choice of basis: Better

**Better:** Basis of the first $n$ eigenfunctions of the Laplace-Beltrami operator
Choice of basis: Better

**Better:** Basis of the first $n$ **eigenfunctions of the Laplace-Beltrami operator**

- Only few functions are needed to represent a function sufficiently ($n = 100$)
Choice of basis: Better

**Better**: Basis of the first $n$ eigenfunctions of the Laplace-Beltrami operator

- Only few functions are needed to represent a function sufficiently ($n = 100$)
- Space of functions spanned by the first $n$ eigenfunctions is stable under near-isometries
Choice of basis: Better

**Better:** Basis of the first $n$ eigenfunctions of the Laplace-Beltrami operator

- Only few functions are needed to represent a function sufficiently ($n = 100$)
- Space of functions spanned by the first $n$ eigenfunctions is stable under near-isometries
- near-isometries induce a nearly sparse matrix $C$
Functional Maps

Markus Sterflinger
Technische Universität München
Only cells with $|\mathcal{C}| > 0.11$ are shown.
(a) source
(b) ground-truth map
(c) head to tail map
Revisited: Task

Task:
Given two shapes \( M \) and \( N \). Find the underlying map \( T \).

Approach:
First recover \( C \) and then calculate \( T \)

- How can we recover \( C \) efficiently?
- What properties has \( C \)?
- How can we represent functions efficiently?
Revisited: Task

Task:
Given two shapes $M$ and $N$. Find the underlying map $T$.

Approach:
First recover $C$ and then calculate $T$

- How can we recover $C$ efficiently?
- What properties has $C$?
- How can we represent functions efficiently? ✓
Properties of the representation: Linearity of constraints

Function preservation:

Given $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$ and $a$ and $b$ their representation in $M$ and $N$. 
Properties of the representation: Linearity of constraints

Function preservation:

Given $f : M \rightarrow \mathbb{R}$ and $g : N \rightarrow \mathbb{R}$ and $a$ and $b$ their representation in $M$ and $N$.

$\implies$ New constraint for the matrix $C$: $Ca = b$
Properties of the representation: Linearity of constraints

Function preservation:

Given \( f : M \to \mathbb{R} \) and \( g : N \to \mathbb{R} \) and \( a \) and \( b \) their representation in \( M \) and \( N \).

\[ \implies \text{New constraint for the matrix } C: Ca = b \]

If we can find enough function correspondences we can recover \( C \)
Properties of the representation: Linearity of constraints

Function preservation:

Given \( f : M \rightarrow \mathbb{R} \) and \( g : N \rightarrow \mathbb{R} \) and \( a \) and \( b \) their representation in \( M \) and \( N \).

\[ \Rightarrow \text{New constraint for the matrix } C: Ca = b \]

If we can find enough function correspondences we can recover \( C \)

Descriptor preservation:

e.g. \( f(x) = \kappa(x) \)

Works for multidimensional functions \( \rightarrow k \) constraints
Landmark point correspondence:

Given $T(x) = y$ with $x \in M$ and $y \in N$
Properties of the representation: Linearity of constraints

**Landmark point correspondence:**

Given $T(x) = y$ with $x \in M$ and $y \in N$

Construct

- indicator functions around $x$ and $y$ and set $Ca = b$
Properties of the representation: Linearity of constraints

Landmark point correspondence:

Given \( T(x) = y \) with \( x \in M \) and \( y \in N \)
Construct
- indicator functions around \( x \) and \( y \) and set \( Ca = b \)
- distance functions \( f \) and \( g \) to \( x \) and \( y \) in \( M \) respectively \( N \).
Set \( Ca = b \)
Properties of the representation: Linearity of constraints

**Landmark point correspondence:**

Given $T(x) = y$ with $x \in M$ and $y \in N$.

Construct

- indicator functions around $x$ and $y$ and set $Ca = b$
- distance functions $f$ and $g$ to $x$ and $y$ in $M$ respectively $N$.
  Set $Ca = b$

**Segment correspondences:**

similar to above
Properties of the representation: Linearity of constraints

**Landmark point correspondence:**

Given \( T(x) = y \) with \( x \in M \) and \( y \in N \)

Construct

- indicator functions around \( x \) and \( y \) and set \( Ca = b \)
- distance functions \( f \) and \( g \) to \( x \) and \( y \) in \( M \) respectively \( N \).

Set \( Ca = b \)

**Segment correspondences:**

similar to above

\( \Rightarrow \) indicator or distance functions
Properties of the representation: Operator

Commutativity

Utilization of additional information of the shapes:
Properties of the representation: Operator

Commutativity

Utilization of additional information of the shapes:

- A given symmetry $S : M \rightarrow M$ induces a symmetry operator $S_F : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(M, \mathbb{R})$ by

$$S_F(f)(x) = f(S^{-1}(x))$$
Properties of the representation: Operator

Commutativity

Utilization of additional information of the shapes:

- A given symmetry \( S : M \rightarrow M \) induces a symmetry operator \( S_F : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(M, \mathbb{R}) \) by
  \[
  S_F(f)(x) = f(S^{-1}(x))
  \]

- The symmetry operator, just like the Laplace-Beltrami operator or derived operators (e.g. the heat operator) should be preserved under isometries
Properties of the representation: Operator Commutativity

Utilization of additional information of the shapes:

- A given symmetry $S : M \to M$ induces a symmetry operator $S_{\mathcal{F}} : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(M, \mathbb{R})$ by $S_{\mathcal{F}}(f)(x) = f(S^{-1}(x))$

- The symmetry operator, just like the Laplace-Beltrami operator or derived operators (e.g. the heat operator) should be preserved under isometries

$\implies$ Given two functional operators $S_{\mathcal{F}}$ and $R_{\mathcal{F}}$ on $M$ and $N$ respectively, $C$ should commute with them: $R_{\mathcal{F}} C = C S_{\mathcal{F}}$
Properties of the representation: Operator

Commutativity

Utilization of additional information of the shapes:

- A given symmetry $S : M \rightarrow M$ induces a symmetry operator $S_{\mathcal{F}} : \mathcal{F}(M, \mathbb{R}) \rightarrow \mathcal{F}(M, \mathbb{R})$ by
  $$S_{\mathcal{F}}(f)(x) = f(S^{-1}(x))$$

- The symmetry operator, just like the Laplace-Beltrami operator or derived operators (e.g. the heat operator) should be preserved under isometries

$$\Rightarrow$$ Given two functional operators $S_{\mathcal{F}}$ and $R_{\mathcal{F}}$ on $M$ and $N$ respectively, $C$ should commute with them: $R_{\mathcal{F}} C = C S_{\mathcal{F}}$

$$\Rightarrow$$ New linear constraint
Properties of the representation: Regularization

Constraints

- Every bijective map $T$ has a functional representation $T_F = C$
Properties of the representation: Regularization

Constraints

- Every bijective map $T$ has a functional representation $T_F = C$
- The opposite is not always true
Properties of the representation: Regularization Constraints

- Every bijective map $T$ has a functional representation $T_F = C$
- The opposite is not always true
- If we know, that the underlying map $T$ is bijective, does the matrix $C$ got additional constraints?
Properties of the representation: Regularization

Constraints

- Every bijective map $T$ has a functional representation $T_F = C$
- The opposite is not always true
- If we know, that the underlying map $T$ is bijective, does the matrix $C$ got additional constraints?

**Theorem**

*If the basis functions are discrete and orthonormal or if the underlying map $T$ is volume preserving, i.e. $\mu^M(x) = \mu^N(T(x))$ where $\mu^M$ and $\mu^N$ are volume elements on $M$ and $N$ respectively, then the matrix $C$ must be orthonormal.*
Properties of the representation: Regularization Constraints

\[ \implies \text{In natural setting, we expect orthonormal or nearly-orthonormal functional map matrices for a point-to-point map} \]

\[ \implies \text{We receive the inverse of } C \text{ by transposition} \]
Revisited (2): Task

Task:

Given two shapes $M$ and $N$. Find the underlying map $T$.

Approach:

First recover $C$ and then calculate $T$

- How can we recover $C$ efficiently?
- What properties has $C$?
- How can we represent functions efficiently? ✓
Let's consider the following task:

**Task:**
Given two shapes $M$ and $N$. Find the underlying map $T$.

**Approach:**
First recover $C$ and then calculate $T$.

- How can we recover $C$ efficiently?
- What properties has $C$? ✓
- How can we represent functions efficiently? ✓
Given a pair of discrete shapes with their corresponding Laplace-Beltrami eigenfunctions.
Algorithm

Given a pair of discrete shapes with their corresponding Laplace-Beltrami eigenfunctions.

**Algorithm for calculating $C$:**

1. Construct a system of constraints, mentioned above.
Algorithm

Given a pair of discrete shapes with their corresponding Laplace-Beltrami eigenfunctions.

Algorithm for calculating $C$:
1. Construct a system of constraints, mentioned above
2. Optimize

$$\min_C ||Ca - b||$$

with $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ matrix, s.t. $Ca_i = b_i$ is a linear constraint $\forall i \in \{1, \ldots, n\}$
Now we want to get the underlying bijection $T$ out of the functional representation $C$. 
Getting point-to-point

Now we want to get the underlying bijection $T$ out of the functional representation $C$.

**Straightforward approach:**

- For every $x \in M$ construct the indicator function $f : M \rightarrow \mathbb{R}$
Now we want to get the underlying bijection $T$ out of the functional representation $C$.

**Straightforward approach:**

- For every $x \in M$ construct the indicator function $f : M \to \mathbb{R}$
- Calculate $g = T_F(f) = Ca$
Getting point-to-point

Now we want to get the underlying bijection $T$ out of the functional representation $C$.

**Straightforward approach:**

- For every $x \in M$ construct the indicator function $f : M \rightarrow \mathbb{R}$
- Calculate $g = T_F(f) = Ca$
- Declare $y \in N$ as the point, where $g(y)$ obtains the maximum

Requires $O(V_M V_N)$ operations for meshes with $V_M$ and $V_N$ vertices
Getting point-to-point

Now we want to get the underlying bijection $T$ out of the functional representation $C$.

**Straightforward approach:**
- For every $x \in M$ construct the indicator function $f : M \rightarrow \mathbb{R}$
- Calculate $g = T_F(f) = Ca$
- Declare $y \in N$ as the point, where $g(y)$ obtains the maximum
- Requires $O(V_M V_N)$ operations for meshes with $V_M$ and $V_N$ vertices
Getting point-to-point

Better:

Consider the delta function around \( x \):

\[
\delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i^M(x) \phi_i^M(\cdot)
\]
Better:

Consider the delta function around $x$:

$$
\delta_x = \lim_{t \to 0^+} k^M_t(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t \lambda_i} \phi^M_i(x) \phi^M_i(\cdot)
$$

$\delta_x$ has the coefficients $a_i = \phi^M_i(x)$
Getting point-to-point

Better:

1. Consider the delta function around $x$:
   \[
   \delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i^M(x) \phi_i^M(\cdot)
   \]
   \[
   \delta_x \text{ has the coefficients } a_i = \phi_i^M(x)
   \]
   \[
   \text{Write these coefficients as columns in a matrix } \Phi^M \text{ for every } x \in M \text{ (Doing the same for every } y \in N \text{ gives us } \Phi^N)\]
Getting point-to-point

Better:

- Consider the delta function around $x$:
  \[ \delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t \lambda_i} \phi_i^M(x) \phi_i^M(\cdot) \]
- $\delta_x$ has the coefficients $a_i = \phi_i^M(x)$
- Write these coefficients as columns in a matrix $\Phi^M$ for every $x \in M$ (Doing the same for every $y \in N$ gives us $\Phi^N$)
- $C\Phi^M$ is the image of all delta functions in $M$
Getting point-to-point

Better:

- Consider the delta function around $x$:
  \[ \delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i^M(x) \phi_i^M(\cdot) \]

- $\delta_x$ has the coefficients $a_i = \phi_i^M(x)$

- Write these coefficients as columns in a matrix $\Phi^M$ for every $x \in M$ (Doing the same for every $y \in N$ gives us $\Phi^N$)

- $C\Phi^M$ is the image of all delta functions in $M$

- Finding the nearest neighbor of every point in $C\Phi^M$ in $\Phi^N$ leads us to a point-to-point correspondence
Getting point-to-point

Better:

- Consider the delta function around $x$:
  \[
  \delta_x = \lim_{t \to 0^+} k_t^M(x, \cdot) = \lim_{t \to 0^+} \sum_{i=0}^{\infty} e^{-t \lambda_i} \phi_i^M(x) \phi_i^M(\cdot)
  \]

- $\delta_x$ has the coefficients $a_i = \phi_i^M(x)$

- Write these coefficients as columns in a matrix $\Phi^M$ for every $x \in M$ (Doing the same for every $y \in N$ gives us $\Phi^N$)

- $C\Phi^M$ is the image of all delta functions in $M$

- Finding the nearest neighbor of every point in $C\Phi^M$ in $\Phi^N$ leads us to a point-to-point correspondence

- With a good search algorithm this can be done efficiently
We can use the observations we made above to refine our given matrix $C$ and make it closer to a point-to-point map.
We can use the observations we made above to refine our given matrix $C$ and make it closer to a point-to-point map.

- Every column of the matrix $C\Phi^M$ must coincide with one column of $\Phi^N$ and $C$ should be orthonormal.
We can use the observations we made above to refine our given matrix $C$ and make it closer to a point-to-point map.

Every column of the matrix $C\Phi^M$ must coincide with one column of $\Phi^N$ and $C$ should be orthonormal.

**Algorithm:**

1. For each column $x$ of $C\Phi^M$ find the closest $\tilde{x}$ in $\Phi^N$
We can use the observations we made above to refine our given matrix $C$ and make it closer to a point-to-point map.

Every column of the matrix $C\Phi^M$ must coincide with one column of $\Phi^N$ and $C$ should be orthonormal

Algorithm:
1. For each column $x$ of $C\Phi^M$ find the closest $\tilde{x}$ in $\Phi^N$
2. Find optimal orthonormal $C$ minimizing $\sum \|Cx - \tilde{x}\|$
**Improvement**

- We can use the observations we made above to refine our given matrix $C$ and make it closer to a point-to-point map.
- Every column of the matrix $C\Phi^M$ must coincide with one column of $\Phi^N$ and $C$ should be orthonormal

**Algorithm:**

1. For each column $x$ of $C\Phi^M$ find the closest $\tilde{x}$ in $\Phi^N$
2. Find optimal orthonormal $C$ minimizing $\sum ||Cx - \tilde{x}||$
3. Iterate until convergence
The big advantage of this method is, that the constraints of the representation are linear. This makes the optimization feasible and efficient. This representation can achieve the same results than state-to-the-art methods. Other methods can benefit from the refinement in the post-processing step.
(a) SCAPE

(b) TOSCA
(a) SCAPE

(b) TOSCA
Thank you for your attention

Questions?
Choice of basis: Demands

Demands:

- **Compactness**: Only few basis elements are required to approximate most natural functions
- **Stability**: Space of functions spanned by basis functions must be stable under small shape deformations
Properties of the representation: Continuity

- \( T_F(f) = Ca \) is continuous under changes of \( a \) thus under changes of the **input function**
Properties of the representation: Continuity

- $T_F(f) = Ca$ is continuous under changes of $a$ thus under changes of the **input function**
- The **image** $Ca$ of any function will be smooth
Properties of the representation: Continuity

- $T_F(f) = Ca$ is continuous under changes of $a$ thus under changes of the input function
- The image $Ca$ of any function will be smooth
- The matrix is continuous under changes: Every matrix $C$ can be seen as a functional map
Properties of the representation: Continuity

- $T_F(f) = Ca$ is continuous under changes of $a$ thus under changes of the **input function**
- The **image** $Ca$ of any function will be smooth
- The **matrix** is continuous under changes: Every matrix $C$ can be seen as a functional map
- major advantage over point-to-point or segment-to-segment bijections, which are inherently discrete
proof

**Theorem**

*If the basis functions are discrete and orthonormal or if the underlying map $T$ is volume preserving, then the matrix $C$ must be orthonormal.*

- Suppose $T$ is volume-preserving, i.e. $\mu^M(x) = \mu^N(T(x))$ where $\mu^M$ and $\mu^N$ are volume elements on $M$ and $N$ respectively.
- $C_{i,j} = \langle T(\phi^M_i), \phi^N_j \rangle = \int_N \phi^M_i \circ T^{-1}(y) \phi^N(y) \mu^N(y)$
- Now consider $S = T^{-1} : N \to M$ and let $D$ be its functional representation.
- $D_{j,i} = \langle S(\phi^N_j), \phi^M_i \rangle = \int_M \phi^N_j \circ T(x) \phi^M(x) \mu^M(x)$
- $D^T = C$