An Introductory Perspective on Functional Maps
Seminar: 3D Shape Matching and Applications in Computer Vision

Yizheng Xie
Technical University of Munich

yizheng.xie@tum.de

Figure 1. Visualization of a functional map matrix as heatmap and its corresponding texture transfer example between shapes. Figure adapted from [1].

Abstract

This report delves into functional maps [2], a representation of maps between shapes that differs from a typical point-to-point map. This representation, in the form of a small matrix, is efficient to compute and inference, and can encode near-perfect mapping information between shapes. Introduced in 2012, functional maps have been studied extensively for a decade and have numerous iterations and variations. This report aims to provide an intuitive understanding of this representation, its significance, and its applications for non-rigid shape matching.

1. Introduction

Shape matching is central to many geometry processing tasks. And in the domain of shape matching, functional maps [2] emerged in 2012 from the need to address non-rigid shape matching challenges. This stood in stark contrast to its rigid counterpart. While the problem of rigid shape matching had been considered as well-established: rigid correspondences can be compactly represented using rotations and translations, typically in a 4×4 matrix, the non-rigid shape matching wasn't benefiting from such constraints. Given that any point might be mapped to any other point, the solution space becomes exponential to the number of points. This made it challenging to limit the possible solutions leading them to design complex and intricate methods. Typically such methods weren't very straightforward. Common approaches involved for example restricting the matches to a sparse set of points [3] [4] or imposing additional constraints, such as enforcing the preservation of relative geodesic distances for the match [4].

The brilliance of functional maps lies in its proposal to offer a representation as compact as that of rigid alignment. This essentially means attempting to shrink the solution space of the non-rigid problems, making it closely resemble that of the rigid scenario.

This was done possible by shifting into the spectral domain. It is widely recognized that many complicated problems tend to simplify when solved in the spectral domain. This proved true for our non-rigid shape matching problem as well. Thus, the understanding and intuition can be further expanded from this perspective.

2. Background

2.1. Fourier Analysis

The starting point stems from the domain of Fourier Analysis. At its essence, for any specified function, Fourier Analysis enables us to break it down into its low-frequency components. By adding up these low-frequency parts, an approximation of the original function can be reconstructed. A desirable property of this process is that only the coefficients of these low-frequency parts need to be stored. This essentially means that a function can be represented, or compressed into its coefficients.

2.2. Image Compression

Let's delve into a practical extension: image compression. Visualizing sine waves as the underlying basis functions for images, in Figure 2, we are presented with an orig-

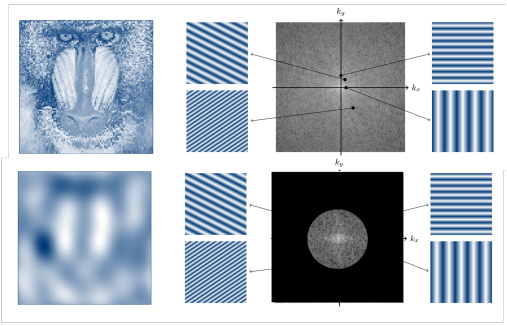


Figure 2. Image Compression Using Fourier Basis Functions. Image on the left, spectral coefficients on the right. Compression involves retaining only the low-frequency components. Figure adapted from [5].

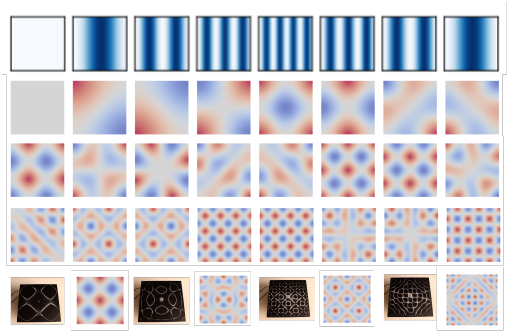


Figure 3. LBO Basis Functions (middle) on a flat square surface compared to Fourier basis (top row) and the similar patterns observed in a Chladni plate sand experiment. Figure adapted from [7] and [8].

inal image on one side and its complete spectral coefficients on the other, the act of compression can be achieved by restricting the representation to only its low-frequency components.

To further expand our understanding, we should familiarize ourselves with discretization and matrix notation. Consider every image as an extended vector (represented by f). The act of multiplying these basis functions (represented by Φ) with their corresponding coefficients (represented by a) and subsequently summing them up can be expressed as a matrix-vector multiplication. This operation is the process of recovering the original or approximated image from its coefficients.

$$f = \Phi \cdot a \tag{1}$$

For compression operation, the aforementioned equation is inverted. Fortunately, Fourier bases are orthogonal. This means that their inverse is simply their transpose. It's also important to note that if we compress the original, sharp image, by the low-frequency bases, the high-frequency information is eliminated, leading to loss in detail. But if we compress an image that is already compressed, no information is lost.

$$a = \Phi^{\dagger} \cdot f \tag{2}$$

or

$$a = \Phi^{T} \cdot f \tag{3}$$

2.3. Eigenfunctions of the Laplace-Beltrami Operator

The overall objective is to compress the correspondence information between shapes. However, this requires a shift from the Fourier basis to an alternative one, namely the eigenfunctions of the Laplace-Beltrami Operator (LBO). In the field of shape analysis, the LBO has been one of the most important tool that has been used all over the place, sometimes regarded as the "Swiss army knife" for shape analysis.

While a deep dive into the LBO isn't necessary for our discussion, it is vital to understand the fundamental properties of its eigenfunctions. Analogous to the Fourier basis, these eigenfunctions are characterized by their frequencies and are orthogonal. To provide a more visual understanding, see Figure 3 for an illustration. These patterns illustrate the vibration modes of surfaces and can be quite intriguing or even enlightening. A classic demonstration of this phenomenon is the Chladni plate sand experiment [6], where varying frequencies result in distinct sand patterns on a vibrating plate.

Given that any shape can vibrate, this allows us with a generally defined spectral basis, well-suited for arbitrary shape surfaces encountered in shape matching problems.

Drawing parallels to our earlier discussion on image compression, these eigenfunctions can be employed to approximate functions (including images) on various surfaces. The notational representation for reconstructing functions and compression or projection remains consistent with our previous ones.

2.4. Correspondences between Shapes

Recall again that our main objective is to approximate correspondences between two shapes. To further clarify, by "correspondence", we refer to a point-to-point mapping in our context. It can be represented as a permutation matrix (represented by P), which aligns (permutes) the vertex points between the two shapes. As an interpretation of its role, this point map serves in transferring functions from one shape to the other.

$$f_1 = P \cdot f_2 \tag{4}$$

For illustrative purposes, consider a color transfer. We

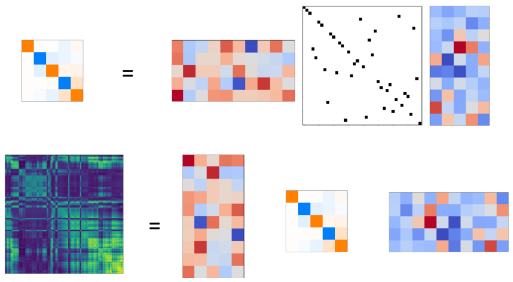


Figure 4. Functional Map and its recovered fuzzy Point Map

can designate a color to each vertex point on a shape and represent it as a long vector. The point map can be used to transfer or permute this color to corresponding vertexes on another shape.

The goal is to use our established basis functions to approximate this permutation matrix, compress it into compact spectral coefficients. To correctly compress the point map, it's crucial to recognize that the map bridge across two shapes, meaning that we need to use basis from both shapes. On an intuitive level, this can be done by squeezing it from two sides, and this reduced matrix is called the "functional map" (represented by C).

$$C = \Phi_1^{\dagger} \cdot P \cdot \Phi_2 \tag{5}$$

And drawing parallels to the previous discussion again, the recovery process from the functional map back to a point map can be achieved by applying the operations inversely.

$$P' = \Phi_1 \cdot C \cdot \Phi_2^{\dagger} \tag{6}$$

What's recovered, however, will not not a crisp, clear point map, instead a soft, diffused matrix, very roughly (not strictly) can be thought of as probability distributions. For a more intuitive grasp, think of this fuzzy matrix as the likelihood or chances of where this point is being mapped to. To retain a crisp hard point map, one can employ the argmax operation, where we identify the maximum value in each row and assign 1.

At this point, a natural question arises: Just how accurate is a functional map? To better understand functional map's accuracy, we look at a texture transfer example between two shapes in Figure 5. Using both a ground truth point map and a functional map representation with 30 basis functions, textures are transferred from the source shape to the target shape. But without revealing which method was used for which result, it's not straightforward to determine the method of each. The surprising outcome is that functional map representation is incredibly accurate despite the low frequency basis used. In comparison to the image compression example earlier, where the differences between the original and compressed versions were immediately evident, the distinctions here aren't as clear.

Figure 5. Texture Transfer: Source (left) and Target (right) visualizations obtained through ground truth Point Map and Functional Map. The specific method for each figure is intentionally not specified here. Figure from [1].

To truly grasp the high accuracy of the functional map, we need to dive into its core principles.

3. Method Overview

The concept of the functional map can be illustrated from four fundamental perspectives. By discussing each of these perspectives, our objective is to demonstrate the core principles that demonstrate the essence of the functional map. We have already touched upon the first perspective, and now we delve deeper into the rest of the three.

3.1. Four fundamental Perspectives

3.1.1 A Rank-k Approximation of a Point Map

Given our established understanding, the first perspective is that a functional map is described as a rank-k approximation of a point map. This is given by this equation:

$$C = \Phi_1^{\dagger} \cdot P \cdot \Phi_2 \tag{7}$$

3.1.2 Spectral Coefficients Translator

The second perspective can be seen by examining its input and output vectors. For this, it's helpful to compare it with the more straightforward point map.

In a point map, the role is clear: it transfers functions between shapes. For the functional map, the process becomes slightly more abstract, but effectively the same thing under the hood:

- Input: The coefficient of some function to feed into a basis matrix
- Output: The coefficient of some other function, coming out of an inverse basis matrix

The inner workings of the functional map can be thought of as a series of transformations:

1. First, the functional map receives a coefficient, recovering it from spectral domain back to spatial domain: a function.
2. This restored function is then transferred exactly by a point map.
3. The transferred function is compressed back into a coefficient by the third transform.

While point maps and functional maps might appear different, they're essentially the same thing. Both aim to transfer functions between shapes just in different domain. Point maps operate directly in the spatial domain, while functional maps operate in the spectral domain, dealing with function coefficients.

In simpler terms, think of a functional map as a translator. It takes function coefficients from one shape and transfers into function coefficients on another. Consider $b = \Phi_1^\dagger \cdot f_1$ and $a = \Phi_2^\dagger \cdot f_2$, the relationship is given by:

$$f_1 = P \cdot f_2 \tag{8}$$

$$b = C \cdot a \tag{9}$$

3.1.3 Stack of Coefficients to the target basis functions

To understand functional maps further, we need to examine the elements inside the matrix, especially the columns.

The columns within the matrix are themselves coefficients of something.

By looking at the definition again, when a point map is applied to the second basis functions, it transfers these basis functions to the first shape, which are then condensed into coefficients by the basis of shape 1. Crucially, every column of C represents the coefficients corresponding to each of the target basis functions.

$$C = \Phi_1^\dagger \cdot P \cdot \Phi_2 \tag{10}$$

$$C = \Phi_1^\dagger \cdot \Phi_{2a} \tag{11}$$

3.1.4 Alignment Transformation between two Basis

Our final perspective brings us to the alignment. A functional map aligns two bases. This interpretation becomes more intuitive when we consider that the elements in C are essentially coefficients of the target basis.

In matrix notation, this can be visualized as an expansion of these coefficients into an aligned basis.

$$\Phi_1 \cdot C = \Phi_{2a} \tag{12}$$

This concludes the four fundamental perspectives on functional maps. For a summary, see Figure 6.

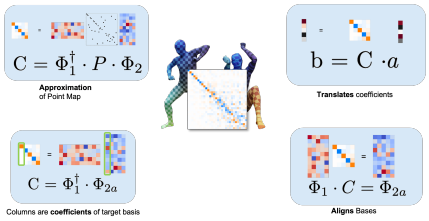


Figure 6. Four fundamental Perspectives of functional maps. Figure adapted from [1]

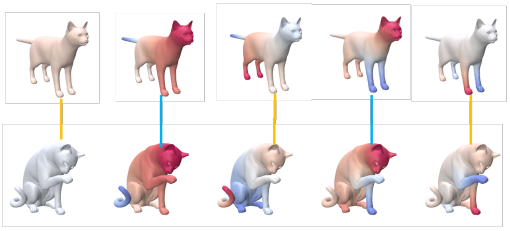


Figure 7. Top 5 LBO Basis functions of two different cats. From TOSCA Dataset [9]

3.2. Properties of Functional Maps

3.2.1 Diagonal entries

A question that naturally arises when examining the functional map is the noticeable diagonal pattern. What underlies this characteristic appearance?

Upon closer visual inspection in Figure 7, the answer comes from the similarities found in the basis functions. These functions exhibit great resemblances, either mirroring each other closely or displaying near complete opposites. This characteristic alignment or opposition results in the distinct diagonal form we observe. Indeed, the essence of the functional map's efficacy can be attributed to these inherent properties of the basis functions.

3.2.2 Eigenfunctions of the Laplace-Beltrami Operator (LBO)

The Laplace-Beltrami Operator (LBO) (represented by Δ)can be formally introduced here. Represented mathematically as:

$$\Delta(f) = -\operatorname{div} \nabla(f) \text{ [10]} \tag{13}$$

this operator is defined as the divergence of the gradient of a function.

The eigenfunctions associated with the LBO are characterized by:

$$\Delta\phi_i = \lambda_i \phi_i \quad \text{[10]} \tag{14}$$

Using discretizations for mesh surfaces, as proposed by [11], the LBO is given by a large square and sparse matrix with size of the number of vertices(represented by L). The eigenfunctions are subsequently obtained by solving the

eigenvalues and eigenvectors (represented by λ_i and ϕ_i)of this matrix, relative to a defined vertex mass(represented by M). This can be mathematically expressed as:

$$L\phi_i = \lambda_i M\phi_i \tag{15}$$

These eigenfunctions have the following important properties:

- Ordered based on their eigenvalues.
- Coarse to fine in scale, presenting a notion of frequency.
- Inherently smooth in nature.
- Sensitive to perturbations. For instance, when two eigenvalues are extremely close, even a slight deformation could lead them to suddenly swap.
- Despite the above instability, the space spanned by the top basis functions remains stable under near-isometries. This stability under near-isometries is crucial for the effectiveness of the functional map framework.

3.2.3 General Functional Maps Definition

For a fixed choice of basis functions $\left\{\phi^{\mathcal{M}}\right\}, \left\{\phi^{\mathcal{N}}\right\}$, and a linear transformation T_F between functions, a functional map is a matrix C, s.t. for any $f = \sum_i a_i \phi_i^{\mathcal{M}}$ if $T(f) = \sum_i b_i \phi_i^{\mathcal{N}}$, then:

$$\mathbf{b} = C\mathbf{a} \text{ [10]} \tag{16}$$

C_{ij} : coefficient of $T_F\left(\phi_j^{\mathcal{M}}\right)$ in the basis of $\phi_i^{\mathcal{N}}$. In an orthonormal basis: $C_{ij} = \int_{\mathcal{N}} T_F\left(\phi_j^{\mathcal{M}}\right)\phi_i^{\mathcal{N}} d\mu$

It's important to understand that the scope of this definition is not confined merely to shape matching. Rather, it provides a purely mathematical characterization of a functional map applicable to any concept expressible in this language. For example, there are extended applications of the functional map framework in areas like image segmentation [12]. But they will not be covered by this report.

Intuitively, the high-level definition presented here reiterates the perspectives already introduced earlier: a functional map matrix serves as a translator, it takes function coefficients from one space(shape) and transfers into function coefficients on another.

3.2.4 Specific Functional Maps Definition for Shape Matching

In the context of shape matching, which is our primary focus, the functional map can be formally defined as follows:

Given two shapes with $n_{\mathcal{M}}, n_{\mathcal{N}}$ points and a map: $P : \mathcal{N} \to \mathcal{M}$ $\mathbf{P} : n_{\mathcal{N}} \times n_{\mathcal{M}}$ matrix encoding the map P, one 1 per column with zeros everywhere else. If functions are represented in the reduced basis: $\Phi_{\mathcal{M}} : n_{\mathcal{M}} \times k_{\mathcal{M}}$ matrix of the first $k_{\mathcal{M}}$ eigenfunctions of $\Delta_{\mathcal{M}}$ as columns. $\Phi_{\mathcal{N}} : n_{\mathcal{N}} \times k_{\mathcal{N}}$ matrix of the first $k_{\mathcal{N}}$ eigenfunctions of $\Delta_{\mathcal{N}}$ as columns. The functional map matrix:

$$C = \Phi_{\mathcal{N}}^{+}\mathbf{P}^T\Phi_{\mathcal{M}} \quad {}^{+}\text{: left pseudo-inverse. [10]} \tag{17}$$

It is noted that this definition here again reiterates the perspectives we've previously explored. In essence, a functional map can be seen as a rank-k approximation of a point map.

3.2.5 Structure of the Functional Map Matrix

The structure of the functional map matrix reveals unique patterns as observed in Figure 1:

Sparsity Pattern:

- The functional map is notably sparse, with over 94% of its values falling below a threshold of 0.1.
- Most entries is located around the diagonal axis.
- Upon closer observation, the values exhibit a "funnel-shaped" pattern. This means that as it moves to higher frequencies, the entries become increasingly perturbed and chaotic.

High-frequency Perturbations:

- The irregularities observed in the high-frequency sections can be attributed to the swapping and instabilities of high-frequency eigenfunctions.
- Despite these perturbations, the space spanned by the eigenfunctions remains stable. This stability ensures that the functional representation can effectively and naturally encode such changes.

3.2.6 Accuracy of Functional Maps

To rigorously evaluate how accurately a functional map can represent a point map, it becomes essential to employ a metric. For this purpose, we turn our attention to the concept of geodesic distance.

The geodesic distance measures the length of the shortest path between two points on a manifold, with the constraint that the path does not leave the manifold.

It's important to emphasize the importance of this metric. While two points may seem close in a traditional Euclidean space, their intrinsic distance, measured by the geodesic metric, may be considerably large. In context of shape matching, we care for this intrinsic error.

A test was conducted where the average geodesic error of the functional map representation underlining a ground

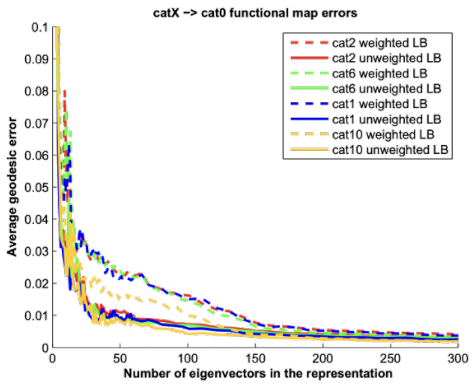


Figure 8. Average geodesic error vs number of basis used for functional map representation. Figure from [2]

truth point map was plotted against the number of basis functions employed. Remarkably, as evidenced by the results, as few as 30 basis functions can offer a satisfactory accuracy, aligning with our earlier introduced example seen in Figure 1 and Figure 5. The plot showcases a clear pattern: In the range of 0-20 basis, the reconstruction error sees a rapid decline. After this range, the reduction rate slows down, eventually plateauing. This trend, however, remains predominantly monotonic: additional basis generally yields improved accuracy.

In practical scenarios, a range of 20 to 100 basis proves adequate for most applications. That said, with contemporary advancements, researchers have came up with methods to harness an even larger set of basis functions, pushing the boundaries of accuracy in shape matching tasks.

3.2.7 Properties of Functional Maps

Functional maps have properties that, when understood and utilized, can help improve the quality of mapping in various applications.

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

$$C\Delta_{\mathcal{M}} = \Delta_{\mathcal{N}}C \text{ [10]} \tag{18}$$

Implies that isometries result in diagonal functional maps.

Given that the Laplace Operator simplifies to a diagonal matrix with its diagonal entries represented by the eigenvalues, this property suggests that isometric maps inherently possess a diagonal structure.

Lemma 2: The mapping is locally volume preserving, if and only if the functional map matrix is orthonormal. [10]

This can be interpreted to mean that "good" mappings exhibit orthonormal characteristics. With a foundational understanding that under optimal conditions, the basis functions are orthonormal, a perfect point map is a permutation, which is also orthonormal, one can infer that functional maps should also inherently be orthonormal.

These properties have important practical applications. Researchers have harnessed these structural properties, using them as regularization mechanisms to refine and optimize maps for better quality. Such properties have found applications for example, in unsupervised loss functions that is used in modern state-of-the-art pipelines [13].

3.2.8 Relationship with Rigid Alignment

In the early discussions of this report, we highlighted a historical limitation in the domain of non-rigid shape matching. Before the introduction of functional maps, the field lacked a compact representation comparable to the 4x4 rotation and translation matrix commonly employed in rigid shape matching scenarios.

With the insights and knowledge gathered from our deep dive into the fundamentals of functional maps, particularly referencing the fourth perspective, we can now appreciate the notion of alignment between two bases through functional maps.

This understanding naturally positions us to compare rigid alignment formulations to functional maps. At a foundational level, functional maps can be thought of as the spectral counterpart of rigid alignments. This implies that while traditional rigid alignment techniques utilize a 4x4 matrix to transform xyz coordinates of points, functional maps utilize a $k \times k$ matrix to transform the spectral embeddings of points, i.e., the basis functions. Such formulation reveals striking similarities, emphasizing their parallel nature.

3.2.9 Visualizing Functional Maps as Alignments

A natural question arises when considering the visualization of alignments: for rigid alignments between two shapes or point clouds, the visualization is inherently intuitive due to the coordinates representing actual points within our spatial dimensions. Yet, how does one visualize the alignment posed by a functional map? Given our physiological limitations as three-dimensional beings, the prospect of visualizing beyond three dimensions is challenging.

Nevertheless, one can still extract a subset of three dimensions from the k dimensional embeddings associated with the functional map. By treating these three dimensions analogously to xyz coordinates, we can represent them within our familiar Euclidean space. This methodology provides intriguing insights, as seen in Figure 9 and Figure 10. Selecting and visualizing the top three basis or embeddings offers a level of preservation of the shape's inherent attributes. For instance, the silhouette of a human figure or the tail of the cat. Exploring different choices of

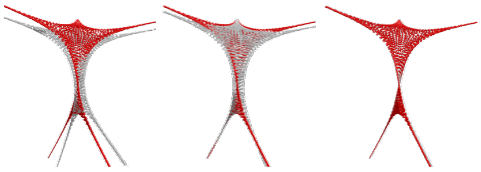


Figure 9. Visualize Functional Maps as Alignments. Top 3 spectral embeddings visualized as point cloud coordinates between two shapes being matching via ICP. Figure from [10]

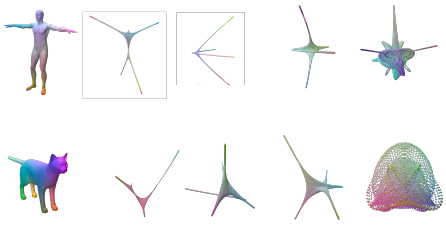


Figure 10. Spectral embeddings of a human and a cat. From left to right: top 1-3 basis, top 2-4 basis, top 3-5 basis, top 21-24 basis. From TOSCA Dataset [9]

dimensions from the basis generates diverse patterns, with complexity increasing as the frequency goes up as seen in Figure 10. On a high level, one might conceptualize a functional map as an alignment for these high-dimensional point clouds, facilitated by a $k \times k$ rigid transformation matrix.

Reflecting on our introduction, the profound significance of functional maps becomes evident: it has, for the first time, presented a compact representation for non-rigid shape matching that mirrors the precision and conciseness of a rigid transformation matrix.

This novel approach has reduced the solution space for the non-rigid shape matching problem from an N exponential solution space to a $k \times k$ matrix.

Furthermore, due to its formulation resemblance to the rigid transformation, the same methods from rigid matching problems can in turn be applied directly, as we can see in the next section.

4. Applications

Since its introduction in 2012, functional maps have gathered extensive attention over the past decade. They have been applied, not only to shape matching problems but also to areas such as shape interpolation [14] or even image segmentation [12].

However, in the scope of this report, our primary focus will remain on shape matching and map refinement methods. We aim to present a concise evolution of these techniques, tracing their trajectory from their inception in 2012

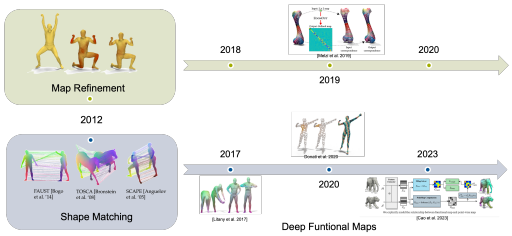
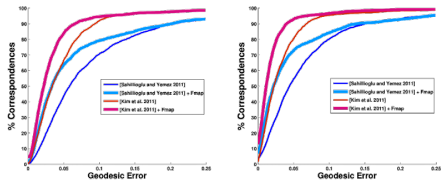


Figure 11. A selection of Applications of Functional Maps. [2] [15] [16] [17] [13]



(a) *SCAPE* (b) *TOSCA*

Figure 12. Improvement of the result from other methods via Functional Map ICP. Figure from [2]

to the prevalent approaches of today.

4.1. Map Refinement

4.1.1 Iterative Closest Point (ICP)

Building on our earlier discussion, it becomes clear that the formulation of the functional map closely aligns with that of rigid alignment. This resemblance provides a unique advantage, enabling the direct application of existing methodologies to the problems of non-rigid shape matching.

One such technique that can be seamlessly applied is the Iterative Closest Point (ICP) method for refining functional maps. The algorithm remains identical to its implementation in the rigid scenario: it iterates between determining point correspondences and updating the compact transformation matrix.

A visual representation of this process is provided in Figure 9.

In its early application in 2012, this straightforward technique was overlaid on top of some other state-of-the-art methods [3] [4] of the time. Remarkably, even such a simple incorporation led to substantial improvements in the performance of pre-existing methods.

4.1.2 Evolution of ICP: From BCICP to ZoomOut

As we trace the evolution of the ICP method up to the present day, more intricate methods have been proposed to leverage various structural properties, enhancing the refinement mechanism. One such noteworthy contribution was

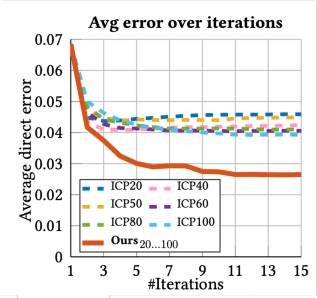
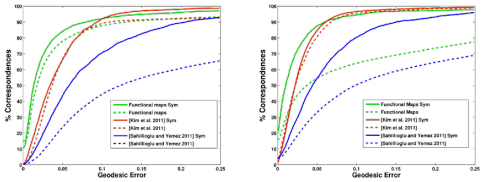


Figure 13. ZoomOut result compared to ICP. Figure from [16]



(a) *SCAPE* (b) *TOSCA*

Figure 14. Comparision of a basic shape matching pipeline using functional maps with other state-of-the-art methods from 2012 [3] [4]. Figure from [2]

the 2018 paper named as BCICP [18], which incorporated several complex techniques to refine the mappings.

What's more remarkable however, in 2019, a simpler and more efficient method named *ZoomOut* [16] rapidly gathered interest in the research community. This approach yielded comparable refinement results and was more computationally efficient than BCICP [18]. Even today, *ZoomOut* remains a preferred choice for many when refining mappings derived from contemporary pipelines.

The essence of *ZoomOut* parallels the ICP methodology with one single exception: the iterative process has in addition a spectral upsampling of the basis numbers used, i.e. increasing the spectral resolution during iterations.

While *ZoomOut*'s simplicity is a significant advantage, its performance notably surpasses that of the elementary ICP and rivals the BCICP method. The underlying intuition is intriguing. Instead of emphasizing the whole functional map matrix to adopt orthonormal or diagonal structural properties, the spectral upsampling ensures that each sub-matrix embedded within the map matrix satisfy the desired properties. Refer to Figure 13 for a comparative results of *ZoomOut* against ICP. For a more in-depth analysis, readers are referred to the original *ZoomOut* publication [16].

Subsequent to *ZoomOut*, more advanced iterations have been proposed, such as the *Consistent ZoomOut* [19]. However, the scope of this report does not extend to these more advanced developments.

4.2. Shape Matching with Functional Maps

Within the domain of shape matching that utilizes functional maps, methods can be very roughly divided into two primary categories: axiomatic methods (eg. smooth shells [20]) and deep functional maps. This report primarily delves into the deep functional map paradigm.

The foundational setup for deep functional maps is straightforward. It relies on the function preservation property, specifically that a functional map primarily serves as a translator of coefficients.

A Basic pipeline goes as follows [2] [10]: Given a pair of shapes $\mathcal{M}, \mathcal{N}$:

1. Compute the first $k(\sim 80-100)$ eigenfunctions of the LaplaceBeltrami operator. Store them in matrices: $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$ 2. Compute descriptor functions (e.g., Wave Kernel Signature) on $\mathcal{M}, \mathcal{N}$. Express them in $\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$, as columns of : $\mathbf{A}, \mathbf{B}$ 3. Solve $C_{\text{opt}} = \arg\min_C \|C\mathbf{A} - \mathbf{B}\|^2 + \|C\Delta_{\mathcal{M}} - \Delta_{\mathcal{N}}C\|^2$ $\Delta_{\mathcal{M}}, \Delta_{\mathcal{N}}$: diagonal matrices of eigenvalues of LB operator 4. Convert the functional map C_{opt} to a point to point map P.

On a high level, the features can either be sourced from traditional geometric descriptors, for example, HKS [21], or WKS [22], or could be from refined features generated by deep learning networks.

Afterwards these features are projected onto the spectral basis, thereby transforming into coefficients.

Leveraging the function preservation property, the process of deriving the functional map essentially becomes a linear solve, in addition to some regularization techniques mentioned earlier.

4.2.1 Results from 2012

In 2012, this very basic setup has been tested, using only traditional geometric descriptors and compared with the state-of-the-art methods at the time [3] [4], the results, as seen from Figure 14, demonstrated that even this basic procedure yielded results that could be arguably considered better.

It's important to highlight the difference between solving a functional map in the spectral domain versus directly solving a point map from point features. The former has a more compact solution space. Additionally, the lower frequency basis components intrinsically exhibit a low-pass filter effect, naturally having a smoothing effect for the matches. This highlights the important role of functional maps within such pipelines.

4.2.2 Deep Functional Maps: 2017-Present

From 2017 to the present day, one of the most popular developments in shape matching domain has been the rise of

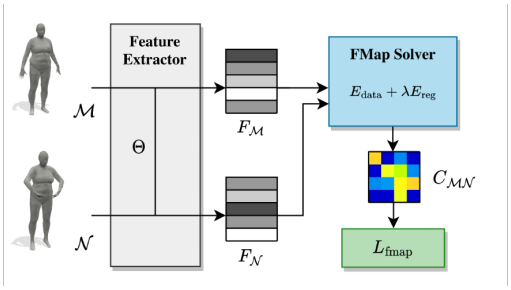


Figure 15. Common pipeline of deep functional map methods. Figure from [13]

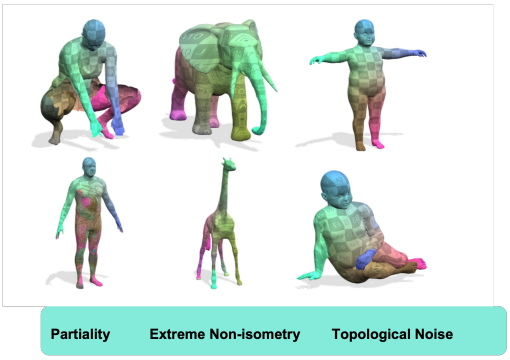


Figure 16. Examples of failure modes from [13]

deep functional map methods. These techniques are achieving state-of-the-art results in present days.

The basic paradigm of these methods has largely remained straightforward. The primary focus of these methods has been mainly geared towards improving the feature extractors and optimizing the objectives with careful considerations for the loss function.

To present a concise evolution without delving into exhaustive details, we select a subset of works:

- In 2017, the first deep functional map method was introduced [15].
- In 2020, an important regularization component for the FM solution was realized and recognized by the community [17].
- This recognition subsequently led to more explorations on regularization masks for FM solution in [23], and was adapted in [24] and most future subsequent works of deep functional map methods.
- In 2022, the dominant feature extractor emerged as the gold standard nowadays for shape surface learning [1].

4.2.3 2023: Current State-of-the-Art in Deep Functional Map Methods

The current state-of-the-art deep functional map method as proposed by [13] can be seen as a culmination of all preceding foundational works. However, it still distinguishes itself by introducing simple yet effective improvements.

It is worth noting that, techniques that are both straightforward and effective usually gain more public attentions. In the course of this report, we covered two such works: the functional map representation [2] and the *ZoomOut* approach [16]. And this work arguably exhibits a similar simplicity and efficacy to those.

To provide a concise highlight of its contributions:

1. Integration of point maps with functional maps. This combination has enhanced matching performance along with improvements in surprising aspects.
2. One of such aspects is the stabilization effect in higher spectral resolutions. This has enabled consistent performance enhancement while increasing the number of bases utilized. Thus, deep functional map methods can now harness up to 200 basis functions and potentially even more.

For a comprehensive overview and detailed evaluation metrics, the reader is referred to [13].

In light of the recent advancements, one could argue that the challenge of non-rigid matching, particularly in scenarios of near-isometries, has now been well-established.

5. Limitations and Discussions

While deep functional map methods have shown significant results, it is interesting to see the scenarios where these methods still fail as shown in Figure 16, such as partiality, extreme non-isometry, and the presence of topological noise.

A notable illustration of non-isometric failure is the challenge of matching largely different animals, such as an elephant and a giraffe. Without any landmarks or semantic cues, relying solely on geometric data would be challenging. There has emerging interest in integrating semantic knowledge, harnessing the capabilities of large language models, to aid in shape matching such as [25].

Further complications arise with functional map methods when addressing non-rigid, noisy point clouds. These often exhibit partiality, topological noise, and extreme outliers. Such factors compromise the stability of the LBO eigenfunctions. While some work has proposed a more robust formulation of the Laplacian [26], some has proposed to learn basis functions directly with neural networks [27] [28], the space for improvements is still large.

Moreover, more investigations into unsupervised losses could be interesting.

6. Conclusions

This report covered an introductory perspective on the functional map representation, aimed at equipping readers with an intuitive understanding. Four fundamental perspectives were covered: a rank-k approximation of a point map, spectral coefficient translator, coefficients of the target basis, and alignment transformation between bases. By comparing the functional map formulation with the rigid scenario, we can understand its significance in limiting the solution space closely to a rigid transformation matrix. This results in a compact, linear, and flexible representation for non-rigid shape matching.

In addition, we covered a variety of applications for functional maps, such as *ZoomOut* [16], and the deep functional map families [13]. With the advancements in these methodologies, near-isometric shape matching is arguably considered as well-established. However, challenges remain in scenarios with partiality, extreme non-isometry, topological noise, and noisy point clouds, opening up spaces for more explorations.

References

[1] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 1, 3, 4, 9

[2] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012. 1, 6, 7, 8, 9

[3] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 1, 7, 8

[4] Y Sahillioğlu and Yücel Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer graphics forum*, volume 30, pages 1461–1470. Wiley Online Library, 2011. 1, 7, 8

[5] Fourier transformation in image processing. https://medium.com/crossml/fourier-transformation-in-image-processing-84142263d734. 2

[6] Bruno Lévy. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006. 2

[7] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 2

[8] Youtube video on chladni plate patterns. https://youtu.be/wvJAgrUBF4w. 2

[9] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 4, 7

[10] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 1–60. 2016. 4, 5, 6, 7, 8

[11] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 4

[12] Fan Wang, Qixing Huang, and Leonidas J Guibas. Image co-segmentation via consistent functional maps. In *Proceedings of the IEEE international conference on computer vision*, pages 849–856, 2013. 5, 7

[13] Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *arXiv preprint arXiv:2304.14419*, 2023. 6, 7, 9, 10

[14] Ruqi Huang, Marie-Julie Rakotosaona, Panos Achlioptas, Leonidas J Guibas, and Maks Ovsjanikov. Operatornet: Recovering 3d shapes from difference operators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8588–8597, 2019. 7

[15] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 5659–5667, 2017. 7, 9

[16] Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. 7, 8, 9, 10

[17] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020. 7, 9

[18] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37(6):1–16, 2018. 8

[19] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020. 8

[20] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 8

[21] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 8

[22] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011. 8

[23] Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*, volume 38, pages 39–53. Wiley Online Library, 2019. 9

[24] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021. 9

[25] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. *arXiv preprint arXiv:2306.03253*, 2023. 9

[26] Nicholas Sharp and Keenan Crane. A laplacian for nonmanifold triangle meshes. In *Computer Graphics Forum*, volume 39, pages 69–80. Wiley Online Library, 2020. 9

[27] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020. 9

[28] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2038–2053, 2022. 9