Eurographics Workshop on 3D Object Retrieval (2017)
I. Pratikakis, F. Dupont, and M. Ovsjanikov (Editors)

A framework based on Compressed Manifold Modes for robust local spectral analysis

S. Haas[1], A. Baskurt[1], F. Dupont[2] and F. Denis[2]

[1]Univ Lyon, INSA de Lyon, LIRIS, F-69621, LYON, France
[2]Univ Lyon, Univ Lyon 1, LIRIS, F-69622, LYON, France

Abstract

Compressed Manifold Modes (CMM) were recently introduced as a solution to one of the drawbacks of spectral analysis on triangular meshes. The eigenfunctions of the Laplace-Beltrami operator on a mesh depend on the whole shape which makes them sensitive to local aspects. CMM are solutions of an extended problem that have a compact rather than global support and are thus suitable for a wider range of applications. In order to use CMM in real applications, an extensive test has been performed to better understand the limits of their computation (convergence and speed) according to the compactness parameter, the mesh resolution and the number of requested modes. The contribution of this paper is to propose a robust choice of parameters, the automated computation of an adequate number of modes (or eigenfunctions), stability with mutltiresolution and isometric meshes, and an example application with high potential for shape indexation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Spectral analysis of meshes relies on the study of the eigenvectors of specifically defined mesh operators. It is used to solve a wide variety of problems such as mesh compression, segmentation, smoothing, watermarking or correspondence. The Laplacian operator is the well-known operator that is defined on a function f as $\Delta f = \mathrm{div}(\overrightarrow{\mathrm{grad}}(f))$ and specifically in $\mathbb{R}^2$ as $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$. The Laplace-Beltrami operator (LBO) is its extension on a manifolds, which, in our case, is a smooth surface inside $\mathbb{R}^3$. The eigenfunctions of the Laplacian operator on $\mathbb{R}^2$ (i.e. the solutions of equation $\Delta f = \lambda f$ where λ is a real number) form the basis for the spectral analysis of functions that are defined on $\mathbb{R}^2$. The eigenfunctions of the LBO have the same role for functions that are defined on the surface.

The LBO discrete counterpart is thus used to perform spectral analysis on a 3D triangle mesh [Tau95]. The eigenfunctions of the discrete LBO are usually referred to as the Manifold Harmonic Basis (MHB) of the considered mesh [Lev06], [VL08] and [RBG*09]. So, the surface decomposition over the MHB can be seen as the Fourier spectrum of the mesh and as such allow its spectral analysis. It is independent of the extrinsic shape properties and invariant under natural shape deformation [Rus07]. They have already been studied for a wide range of applications such as building shape descriptors [RWP06], point descriptors and shape signature [SOG09], shape matching [OMMG10] or shape segmentation [SOCG10].

The Compressed Manifold Modes (CMM) were introduced by Neumann et al. in order to overcome some limitations of global spectral analysis [NVT*14]. This paper presents a framework based on the CMM tool and demonstrates how local spectral analysis can be used in many applications.

Our contributions are: a set of tool parameters and mesh properties that demonstrate robustness and are thus suitable for applications; a simple algorithm to automatically compute an adequate number of CMM functions for a given shape; and we demonstrate our algorithm stability for varying resolutions of the same shape and isometric deformations.

As example application, we propose a coarse matrix comparison that shows very promising results in shape matching.

2. Background, previous and related work

MHB eigenfunctions present a major drawback: they are global and depend on the whole shape. It is thus difficult to interpret them. Moreover, they are sensitive to local modifications of the mesh such as holes and noise that appear in scanned meshes. That global aspect also hampers partial shape recognition. In order to overcome these limitations, T. Neumann et al. introduced the Compressed Manifold Basis (CMB) and its components are called the Compressed Manifold Modes (CMM). The MHB functions are the solutions φ_k of the equation

$$\Delta \varphi_k = -\lambda_k \varphi_k, k \in \mathbb{N}, \lambda_k \in \mathbb{R}$$

DOI: 10.2312/3dor.20171054

with Δ representing the LBO. Usually, a set of K eigenfunctions corresponding to the smallest eigenvalues λ_k is considered. The CMM are the solutions of the generalized eigenvalues problem that induces sparsity by introducing the ℓ_1-norm and a parameter $\mu \in \mathbb{R}^+$:

$$\min_{\varphi_k} \sum_{k=1}^{K} \langle \varphi_k | \Delta \varphi_k \rangle + \frac{\mu}{N} \|\varphi_k\|_1 \text{ such that } \langle \varphi_k | \varphi_j \rangle = \delta_{kj}$$

where δ_{kj} is the Kronecker delta that enforces orthogonality of the CMM or eigenfunctions, and N is the number of vertices of the mesh. These functions have a compact support whose relative size is controlled with parameter μ. When $\mu = 0$, we have the original problem the solution of which is the MHB; larger μ values yield more compact solutions.

3. CMM computation speed and convergence

3.1. Available implementations and test

The CMM computation algorithm proposed by Neumann et al. is based on reformulating the original problem and solving it with the alternating direction method of multipliers (ADMM) as described in [BPC*11]. They claim that, even though the minimization problem is not convex due to the orthogonality constraints, local minima are reached and sufficient for practical applications. Python code has been made available to reproduce some of their results. An accelerated version of the algorithm was later designed by Houston [Hou15]. He claims this new algorithm is about 47% faster than the original one on average, and provides Matlab code to test it. Finally, a new paradigm regarding the ℓ_1-norm discretization has lately been promoted by Bronstein et al. [BCKS16]. The minimization problem is modified into a sequence of eigendecomposition ones, which avoids non-convex optimization and thus achieves a drastic speedup in runtime. We tested the original algorithm and its accelerated counterpart in a python environment. Our goal was to identify the tool limitations and check its applicability.

3.2. Impact of parameters on convergence

We evaluated both the original algorithm and its accelerated version using various shapes. For some typical cases, we also changed the mesh resolution. Each mesh was then tested using different combinations of the compactness parameter μ and the number K of requested eigenfunctions. The conditions of our test case are given below:

- 22 meshes ranging from 453 to 74764 vertices.
- 10 values for the compactness parameter ranging from 1 to 1000.
- The number of requested eigenfunctions ranged from 5 to 40.
- Every mesh has been tested with every possible combination of $K \backslash \mu$ parameters.
- Computation was performed on a PC with an Intel i7-4710HQ CPU @ 2.5GHz with 8Go RAM.

Table 1 shows the number of converging cases for the original algorithm. Results for the accelerated algorithm are very similar. The convergence criterion is fully described in [NVT*14]. Non convergence happens when the 15000 iterations limit is reached. The algorithm converged in less than 50% of the test cases and it clearly appears that non convergence happens when the number of requested eigenfunctions is large and the support is not compact enough. This is very probably due to the orthogonality constraint that is easier to fulfill while the eigenfunctions can have approximately separate supports. Speed was acceptable (less than 1 minute to get the CMM) in most converging cases when the number of requested eigenfunctions was 20 or less.

$K \backslash \mu$	1	2	5	10	20	50	100	200	500	1000
5	11	15	20	22	21	22	22	21	19	16
10	0	0	4	11	12	20	21	19	18	16
20	0	0	0	0	0	9	11	16	14	16
40	0	0	0	0	0	0	0	5	10	13

Total number of converging meshes among 22 for the $K \backslash \mu$ combination.

Table 1: *Convergence with the original CMM algorithm*

4. Finding an adequate number of eigenfunctions

An important objective of our work was to design an algorithm that automatically determines the number of eigenfunctions that best fit the application needs. Indeed, automation is essential for a tool in practical applications: it is not acceptable to rely on the user to fine tune various parameters in order to achieve its goal every time the tool is called. The choices that we present below are based on an extensive set of experiments on usual indexation databases: SHREC15 [LZC*15], SCAPE [ASK*05], FAUST [BRLB14], the Bunny [TL94], Armadillo and Bimba models. The algorithm stopping criterion was designed with a shape matching application in mind.

4.1. Method

The same compactness parameter is used for all meshes ($\mu = 20$). It is the control key to the number of eigenfunctions. The higher it is, the more eigenfunctions one gets for a given shape. Its fine tuning should be done once according to the targeted type of application. It must be remembered that higher compactness parameter values yield more eigenfunctions and might increase drastically the computation time.

Let us then define vertex coverage: a vertex from the mesh is said covered by the CMM when at least one eigenfunction exceeds a preset percentage ThV of its absolute maximum value on that vertex. Next, we consider the mesh is covered when at least a given percentage ThM of the vertices are covered.

The algorithm starts computing 1 eigenfunction and increases this number until the mesh is covered. Every step is initialized with the previously found eigenfunctions and one additional constant function; the resulting eigenfunctions are then tested against the stopping criterion. The process starts over until the criterion is reached. Both thresholds can be adjusted according to the type of desired application. For the application that we present at the end of this paper, the values that gave overall the best results were $ThV = 0.1$ and $ThM = 0.95$.

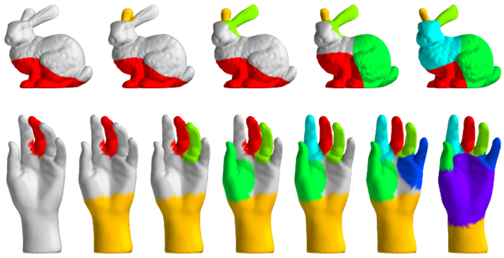

Figure 1: *Algorithm progression*

Figure 2: *Effect of compactness parameter changes*

4.2. Results

On average, for all our test cases, we get about 8 eigenfunctions with the selected compactness parameter. On the SHREC15 dataset, the minimum is 3 eigenfunctions and the maximum is 19. More than 90% of the meshes yield between 4 and 12 eigenfunctions. Two examples in the Figure 1 show the algorithm progression. A vertex is colored when it is covered. When different eigenfunctions overlap, the color of the vertex corresponds to the one with the highest value. These examples present an interesting extraction of the intrinsic global shape aspect. Changing either ThV or ThM threshold only modifies the number of steps of the algorithm that keeps adding new eigenfunctions until stopped. On the other hand, for a given mesh, compactness parameter changes have an impact on the eigenfunctions themselves as can be seen in the Figure 2. The compactness parameter can be adapted to the desired application type. However, it must be kept coherent with the mesh resolution (the greater compactness, the finer details are captured which requires higher resolution) and fit the application exact needs.

In all the tests we made so far with our choice of parameters, the eigenfunction supports have always proved to be connex when the meshes do not have too many holes.

5. Stability over two mesh transformations

The process stability has been tested when a mesh is subsampled with the idea of speeding up the process for a given shape by changing its resolution. In order to address moving shapes, stability was also tested on isometrically deformed shapes.

5.1. Effects of subsampling

We tested subsampling on the Armadillo and the Bimba models using Alliez and Desbrun algorithm [AD01]. For the Bimba, we started with the original mesh that comprises 75000 vertices and subsampled it successively to 30000, 9000, 4000 and 1200 vertices.

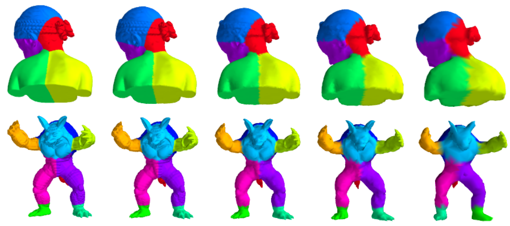

Figure 3: *Stability with multiresolution shapes*

For the Armadillo, the 61400 vertices original model was subsampled to 25500, 8400, 3400 and 900 vertices. As can be seen on the Figure 3, subsampling keeps the shape general aspect while removing details; these modifications did not change the output of the algorithm that shows robustness with respect to the mesh density and level of details. For large meshes, this very interesting result effectively promotes the idea of subsampling to overcome the computation time limitation. When the compactness parameter is $\mu = 20$, 1000 vertices is a good target provided that this level of subsampling is coherent with the shape itself. With a non-optimized python algorithm on a PC with an Intel i7-4710HQ CPU @ 2.5GHz with 8Go RAM, computation on the original Armadillo took 245.7 sec. whereas it was only 1.44 s for the 900 vertices subsampled mesh. From unacceptable computation delays in many applications, we drop to perfectly reasonable speed.

5.2. Isometric deformation

Figure 4: *Stability with isometric deformation*

We evaluated our method on mesh databases that comprise different poses of the same shape. This was done with the SHREC15 and SCAPE datasets. For example, the SCAPE dataset contains 70 different positions of the same human subject, and our algorithm found very similar solutions: 66% showed one configuration (with the same number of eigenfunctions identically located on the shape and found in the same order), and two other close configurations were found for 17% and 10% of the remaining meshes (they include one additional eigenfunction). The other results were minor variants of the 3 main configurations (e.g. the eigenfunctions are located in a different order from the original configuration). Some examples are given in the Figure 4. These very promising results show the stability of the method regarding isometric deformations.

6. Application

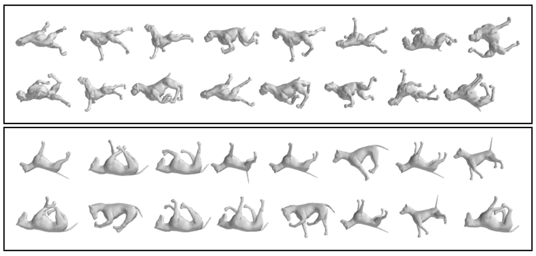

Figure 5: *Two sample groups from the SHREC15 database*

We present here a simple method based on the eigenfunction coverage found by our previous algorithm, and propose coarse shape matching results based on it.

Our method first normalizes the eigenfunctions that have been previously found (their values are thus comprised between -1 and 1, their maximum being always 1). Then, we associate every vertex of the mesh with the function whose value is highest at it. The regions that we obtain have always proved to be connected and we order them according to their eigenfunction discovery rank. We then build a symmetrical boolean matrix that describes the region contiguity.

A coarse matrix comparison was run on the SHREC15 dataset: we grouped together meshes with the same matrix. Two clusters of meshes that were obtained with this comparison are shown in Figure 5. Although these results do not compare yet with state-of-the-art algorithms, the intrinsic simplicity of the tool makes it an encouraging new research direction for shape matching and indexation.

7. Conclusion and future work

The Compressed Manifold Modes that have been recently introduced by Neumann et al. seem an interesting tool that addresses one of the major drawbacks of spectral analysis in shape matching, segmentation and other typical applications. We performed an advanced study of that tool and found a robust set of parameters that demonstrate algorithmic convergence at an acceptable speed. From these results, we proposed an algorithm that automatically computes an adequate number of CMM eigenfunctions that properly cover a given shape. We also showed that this method is stable and yields the same eigenfunctions regardless of the resolution and isometric deformation of the mesh, and gave a simple application with promising outcomes. There are several directions for our future work on the subject:

- try the very recent proposition from Bronstein et al. and evaluate the performance gain for our method;
- apply our method to movements and study its applicability to analyze moving shapes.

References

[AD01] ALLIEZ P., DESBRUN M.: Valence-driven connectivity encoding for 3d meshes. *Computer Graphics Forum 20*, 3 (2001), 480–489. 3

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: Shape Completion and Animation of People. *ACM Trans. Graph. 24*, 3 (July 2005), 408–416. 2

[BCKS16] BRONSTEIN A., CHOUKROUN Y., KIMMEL R., SELA M.: Consistent Discretization and Minimization of the L1 Norm on Manifolds. In *2016 Fourth International Conference on 3D Vision (3DV)* (Oct. 2016), pp. 435–440. 2

[BPC*11] BOYD S., PARIKH N., CHU E., PELEATO B., ECKSTEIN J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn. 3*, 1 (Jan. 2011), 1–122. 2

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and Evaluation for 3d Mesh Registration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), pp. 3794–3801. 2

[Hou15] HOUSTON K.: Compressed manifold modes: Fast calculation and natural ordering. *CoRR abs/1507.00644* (2015). 2

[Lev06] LEVY B.: Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006* (Washington, DC, USA, 2006), SMI '06, IEEE Computer Society, pp. 13–. 1

[LZC*15] LIAN Z., ZHANG J., CHOI S., ELNAGHY H., EL-SANA J., FURUYA T., GIACHETTI A., GULER R. A., LAI L., LI C., LI H., LIMBERGER F. A., MARTIN R., NAKANISHI R. U., NETO A. P., NONATO L. G., OHBUCHI R., PEVZNER K., PICKUP D., ROSIN P., SHARF A., SUN L., SUN X., TARI S., UNAL G., WILSON R. C.: Non-rigid 3d Shape Retrieval. In *Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2015), 3DOR, Eurographics Association, pp. 107–120. 2

[NVT*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed Manifold Modes for Mesh Processing. *Comput. Graph. Forum 33*, 5 (Aug. 2014), 35–44. 1, 2

[OMMG10] OVSJANIKOV M., MÃL'RIGOT Q., MÃL'MOLI F., GUIBAS L.: One Point Isometric Matching with the Heat Kernel. *Computer Graphics Forum 29*, 5 (2010), 1555–1564. 1

[RBG*09] REUTER M., BIASOTTI S., GIORGI D., PATANÈ G., SPAGNUOLO M.: Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics 33*, 3 (2009), 381–390. 1

[Rus07] RUSTAMOV R. M.: Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 225–233. 1

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-Beltrami Spectra As 'Shape-DNA' of Surfaces and Solids. *Comput. Aided Des. 38*, 4 (Apr. 2006), 342–366. 1

[SOCG10] SKRABA P., OVSJANIKOV M., CHAZAL F., GUIBAS L.: Persistence-based segmentation of deformable shapes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops* (June 2010), pp. 45–52. 1

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1383–1392. 1

[Tau95] TAUBIN G.: A Signal Processing Approach to Fair Surface Design. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 351–358. 1

[TL94] TURK G., LEVOY M.: Zippered Polygon Meshes from Range Images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 311–318. 2

[VL08] VALLET B., LÉVY B.: Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum 27*, 2 (2008), 251–260. 1