SHAPE COMPARISON AND RETRIEVAL

INF 574

December 11, 2020

Alexis Groshenry, Julien Bergerot

ÉCOLE POLYTECHNIQUE
IP PARIS

CONTENTS

1
STUDIED METHODS

The purpose of our project is to group and classify different 3D objects using methods presented in the following paper [6]. To do so, the method relies on the Laplace-Beltrami operator. Indeed, researchs have shown that the eigenfunctions of this operator capture key features of the image, describing its intrinsic geometric properties.

1.1 Laplace-Beltrami operator

Firstly, the Laplace-Beltrami operator, called Δ, is defined on a closed compact manifold surface S. It satisfies the following equation $\Delta\psi = \lambda\psi$.

As said above, our interest lies in its eigenfunctions, meaning functions solution of this equation. Its eigenvalues are the λ associated. Our purpose is to compute these functions and values. However, solving this is highly difficult as we are in a continuous space. The solution kept is to render the operator discrete.

To do so, the operator becomes a matrix L. This matrix is defined as such :

$$L_{i,j} = \begin{cases} \Sigma_k m_{i,k}/s_i & \text{if i = j} \\ -m_{i,j}/s_i & \text{if i and j are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here, s_i denotes a specific area around the point indexed i and $m_{i,j}$ is a relation between the angle formed by the points i and j. Their exact computations are explained in the second **part**. Our equation becomes $L\vec{v} = \lambda\vec{v}$. Finally, we rewrite this problem by separating L as $L = S^{-1}M$ with $S_{i,j} = s_i$ and $M_{i,j} = m_{i,j}$. We have now a generalized eigenvalue problem $M\vec{v} = \lambda S\vec{v}$ where M is symmetric and S is definite positive.

1.2 Shape DNA

The first method evoked in Rustamov's paper [6] only requires a few of the smallest eigenvalues of this problem. These capture enough intrinsic information to characterize a mesh and process shape retrieval using a k-Nearest Neighbour algorithm for an input mesh among a known dataset.

1.3 Global Point Signature

Rustamov presents a second method, which he intends to be more robust to deformation and noise. He introduces a measure called GPS, for Global Point Signature, invariant through isometric deformation and thus, more efficient for shape retrieval. GPS is described, for each point p :

$$GPS(\mathbf{p}) = \left(\frac{1}{\sqrt{\lambda_1}}\phi_1(\mathbf{p}), \frac{1}{\sqrt{\lambda_2}}\phi_2(\mathbf{p}), ..\right) \tag{2}$$

where $\phi_i(p)$ is the p^{th}-coordinate of the i^{th} eigenvector, after normalization with the inner-product $||\phi_i||^2 = \phi^t S\phi$.

To measure the closeness of two points, we compute the Green function, which is defined such as $G(x, y) = GPS(x).GPS(y)$. The last step is to create an histogram depicting the original mesh. To do so, we select a certain amount of evenly distributed points and observe the distribution of their Green function values. To emphasize this result, we can separate the points in different classes, related to their distance to the origin in the GPS space. We then form m spheres around the origin and instead of a single histogram, we get $\frac{m(m+1)}{2}$ histograms. Finally, we can compare two meshes by computing the sum of the L2-distances between their histograms sets.

2
OUR IMPLEMENTATION

2.1 Computation of eigenvalues and eigenfunctions of the discrete Laplace-Beltrami operator

To deal with the mesh object, we reuse the Halfedge Structure implementation used in the labs. We first compute the matrices M and S of the discrete Laplace-Beltrami operator introduced above. This was pretty straightforward thanks to the Halfedge Structure : for an input mesh, we simply went through its vertices and computed the coefficients of these matrices by iterating on the incident faces for each vertex. Since the definition of the area used to compute S in the paper [6] was only defined by the following sketch, we assumed the inner vertex in a triangular mesh to be the circumcenter of this triangle. To compute its coordinates, we use a direct formula extracted from this *code*.

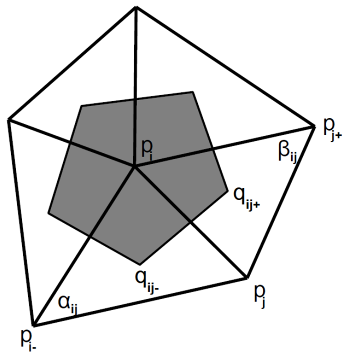

Figure 1: Definition of the area used for the discrete Laplace-Beltrami operator [6]

Having the M and S matrices, the next step is to compute, for a given number n, the n smallest eigenvalues of the generalized eigenvalue problem $MX = \lambda SX$. We first naively tried to compute all eigenvalues using the general eigensolver of the Eigen library[2] and to keep afterwards the smallest eigenvalues, but the computation failed, which was expected since we are working on matrices with more than 15 000 rows and columns on average. Thus, we looked for a C++ library implementing the Arnoldi method evoked in Rustamov's paper [6]. We finally found the Spectra library [5] that proposed different types of eigensolvers based on Arnoldi methods. We first tried to use a specific solver for generalize eigenvalue problems, but since we are looking for the smallest eigenvalues, this solver did not manage to find the right eigenvalues.

Thus, we transformed the problem to solve $S^{-1}MX = \lambda X$ since S is definite positive. We used another solver for classic eigenvalue problem but based on a shift invert method that allows to find the eigenvalues that are the closest to a given number σ. The principle of this method is to solve $(M - \sigma I)^{-1}x = \nu x$ instead of $Ax = \lambda x$ and $\nu = \frac{1}{\lambda - \sigma}$. The largest eigenvalues of this new problem are matching with the eigenvalues that are the closest to σ and the eigenvectors are the same. In our case, $\sigma = 0$. This eigensolver allowed us to compute 10 eigenvalues in a few seconds only for example.

2.2 Shape-DNA

The Shape-DNA method is implemented on python and uses the eigenvalues computed in the previous script. We used the sklearn python library [4] to plot a 2-D PCA of the eigenvalues. For the shape retrieval we also used sklearn [4] to classify an input shape using a k-nearest neighbours algorithm.

2.3 GPS-signature

For the GPS, we compute the GPS-matrix using the previous eigenvalues and eigenvectors and by renormalizing it with the inner-product introduced in the first part.

We then implemented the G2-distributions introduced in Rustamov's paper. Since there were very few details on how the histograms should be plotted, we inspired ourselves from the implementation of D2-distributions presented in Osada's paper [3]. We pick n random faces on the mesh and then pick randomly one of the three vertices. This allows us to extract a nxd submatrix from the global GPS-matrix.

Then, we compute the pairwise distances in this submatrix and we split the corresponding space with m evenly spaced spheres, all centered on the origin. At least, we compute the $\frac{m(m+1)}{2}$ histograms giving for each pair of parts of the d-dimensional space, the G2-distribution for the GPS-signatures of the corresponding vertices.

3 OUR RESULTS

3.1 Shape-DNA method

We tested the shape-DNA method on a first dataset [7]. We had to ignore an important part of this dataset because the mesh encoding of certain objects was wrong and so the Halfedge Builder failed converting it into a halfedge structure. As a consequence, we worked only on 90 meshes from this dataset.

We first checked if the shape-DNA method managed to capture an intrinsic description of a mesh by plotting the 2-D PCA obtained with the 5, 10 and then 20 smallest eigenvalues of the discrete Laplace-Beltrami operator. Each, the pca took between 1h30 and 2h to compute.

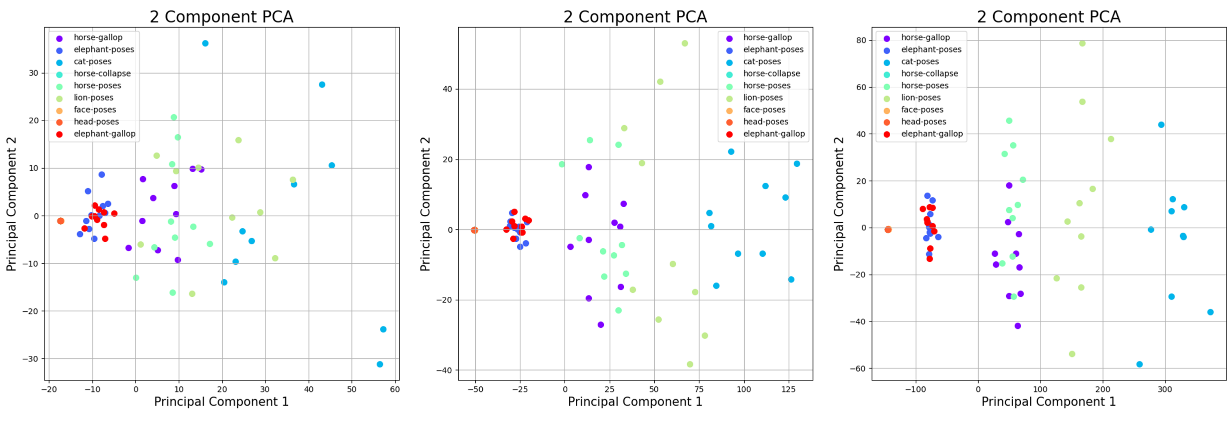

Figure 2: 2-D PCA for 5, 10 and 20 eigenvalues

We observe that groups appear and become more distinct as we increase the number of extracted eigenvalues. Moreover, we note that the classes representing the same object (for example "horse-poses" and "horse-gallop") are mixed. This leads us to think that this method is fitted for shape retrieval.

To test the performance of this method for shape retrieval, we considered a second dataset [1] containing 400 meshes of 20 different classes. We computed the 10 smallest eigenvalues of the 15 first meshes of each class and to use them as a training data for shape retrieval. We plotted the corresponding 2-D PCA and noticed that the distinction between classes was not as clear as in the first dataset.

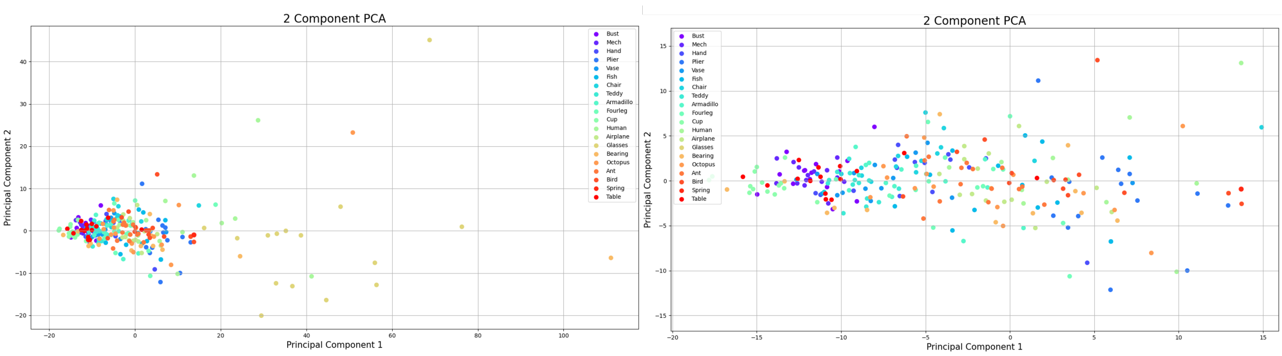

Figure 3: 2-D PCA for 10 eigenvalues (zoom on the right)

Then, we classified the five last meshes with a k-Nearest Neighbours algorithm and we obtained the following heatmap counting the predictions for each. If we get some good results for some classes, we can say that on the whole, the classification is not very efficient. However, this was expected since the PCA was not very good.

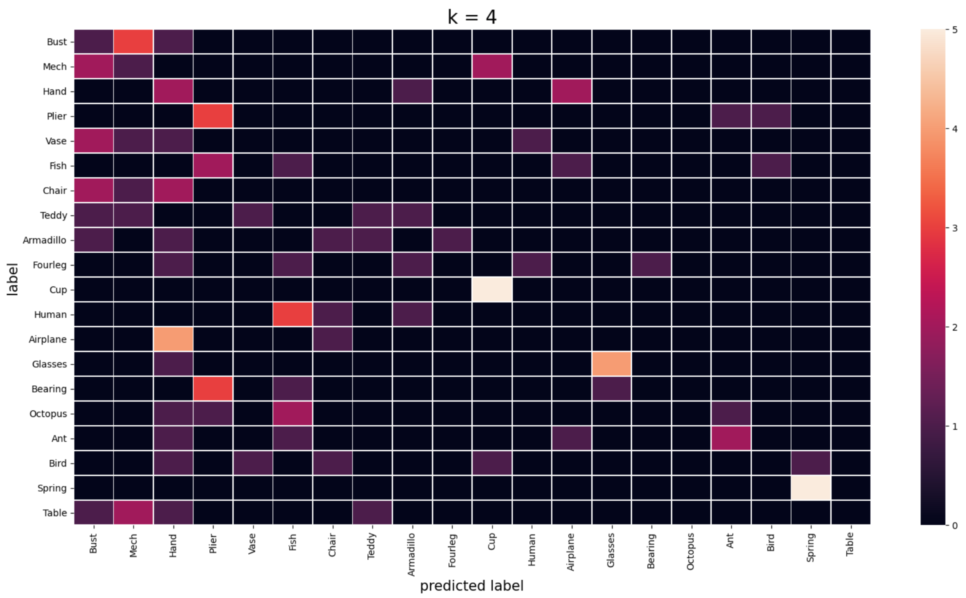

Figure 4: Heatmap for shape retrieval

We could have increased the number of extracted eigenvalues but because of the computation duration, we prefered to go back to the first dataset. We did the same classification process and classified for each class 3 meshes, using the 7 others as training data.

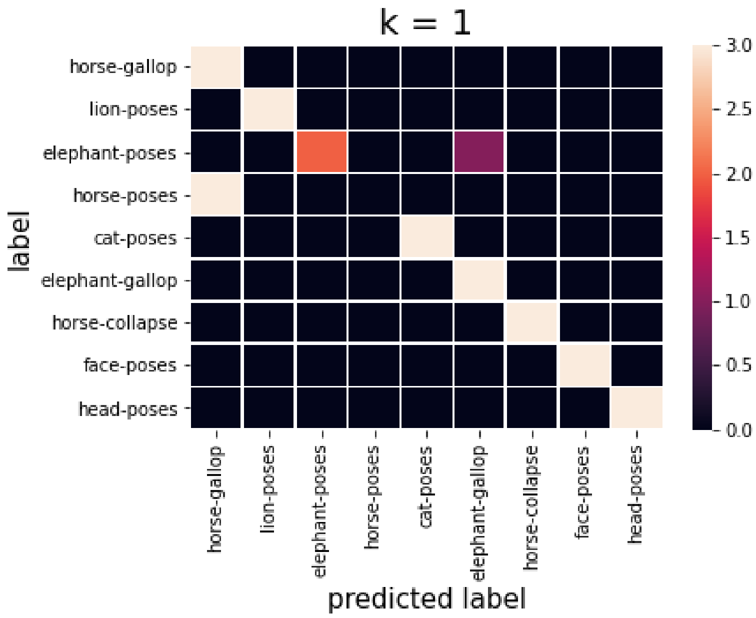

Figure 5: Heatmap for shape retrieval

The heatmap shows that the classification is very good. The classifier is only mistaking when there are several classes for the same object : it mixes horse-poses and horse-gallop, and elephant-poses and elephant-gallop.

3.2 GPS method

Since this method was much longer to complete, we only worked on the first dataset [7]. For each object, we created 36 histograms, meaning the variable m was set to 8. The following curves present an example of histograms comparison : we are considering two different faces, smiling (in yellow) and angry (in blue). These curves overlap but are not as smooth as those in the paper [6]. We tried several approaches (considering all the points, only a few picked randomly but evenly distributed) but were not able to get a better result than the one presented.

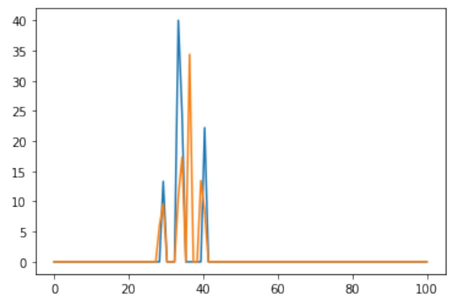

Figure 6: Histogram for smiling and angry faces

We wanted to classify the objects and decided to plot their resemblance in a heat map. The darker it is, the lower the L2-distance is, and therefore, the algorithm would bind them together. However, we do not find clear results as we did in the shape DNA part. We still observe that all the face are pretty close together. We do observe rectangle separating different classes, but it does not seem that elephants and cats differ a lot. The whole computation took approximately five hours.

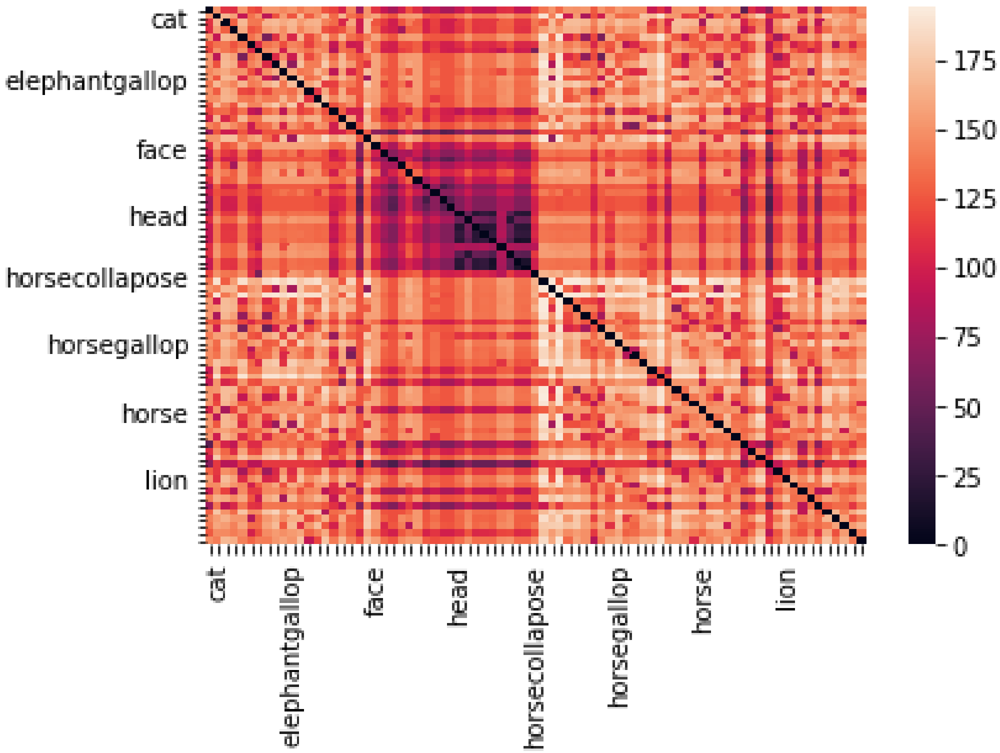

Figure 7: Heatmap for GPS method

4 CONCLUSION AND FINAL REMARKS

If the implementation of the shape-DNA method has given pretty good results for shape retrieval and classification notably for the first dataset, the GPS method has not given satisfying results. Indeed, we have met difficulties to find a proper implementation for plot the G2-distributions and we did not manage to realize curves that were as smooth and regular as in the original paper. Moreover, the author of the Spectra library we used to compute eigenvectors and eigenvalues, warns that sometimes the algorithm mistakes on some eigenvalues and eigenvectors and this may explain wrong values of the GPS.

REFERENCES

[1] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. "A Benchmark for 3D Mesh Segmentation". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3 (Aug. 2009).

[2] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[3] Robert Osada et al. "Shape Distributions". In: *ACM Transactions on Graphics* 21.4 (Oct. 2002), pp. 807–832.

[4] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[5] Yixuan Qiu. *Spectra: C++ Library For Large Scale Eigenvalue Problems*. https://spectralib.org. 2015.

[6] Raif M. Rustamov. "Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation". In: *Geometry Processing*. Ed. by Alexander Belyaev and Michael Garland. The Eurographics Association, 2007. isbn: 978-3-905673-46-3. doi: 10.2312/SGP/SGP07/225-233.

[7] Robert W. Sumner and Jovan Popovic. "Deformation Transfer for Triangle Meshes". In: *ACM Transactions on Graphics* 23.3 (Aug. 2004).