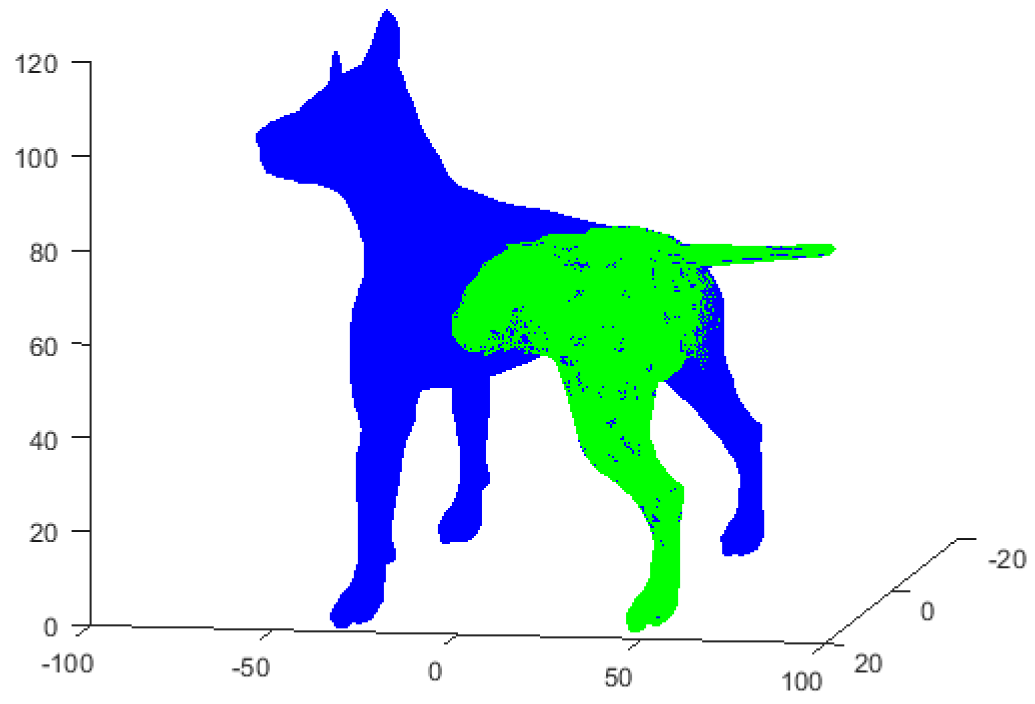
Project report

Based on the article

(Correspondence-Free Region Localization for Partial Shape Similarity via Hamiltonian Spectrum Alignment, 2019)



By Tsachi Blau

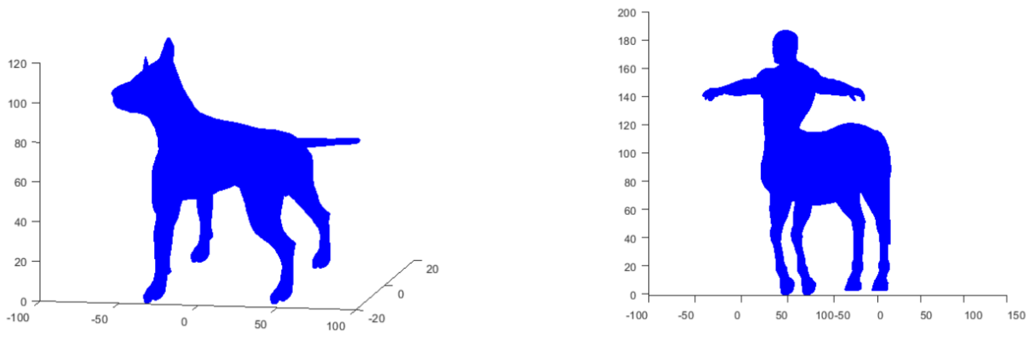
tsachiblau@campus.technion.ac.il

https://github.com/tsachiblau/geometricComputerVisionProject

Contents

Introduction

Assessing similarity between non rigid shapes is an active research topic. This task is being solved by using descriptors. The descriptors give us information about every point. The descriptors can be used for a lot of tasks. In this Project we will focus of the task of similarity between two shapes. We will take a full shape

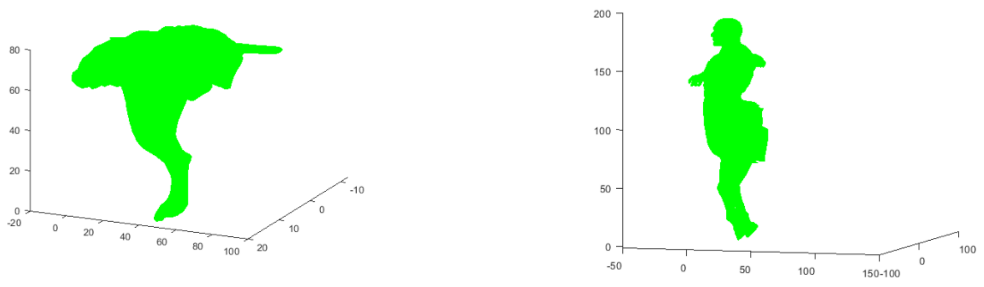


Full shape of a Dog and a Centaur.

The Centaur has 10,000 points and 20,000 triangles

The Dog has 10,000 points and 20,000 triangles

And partial shape

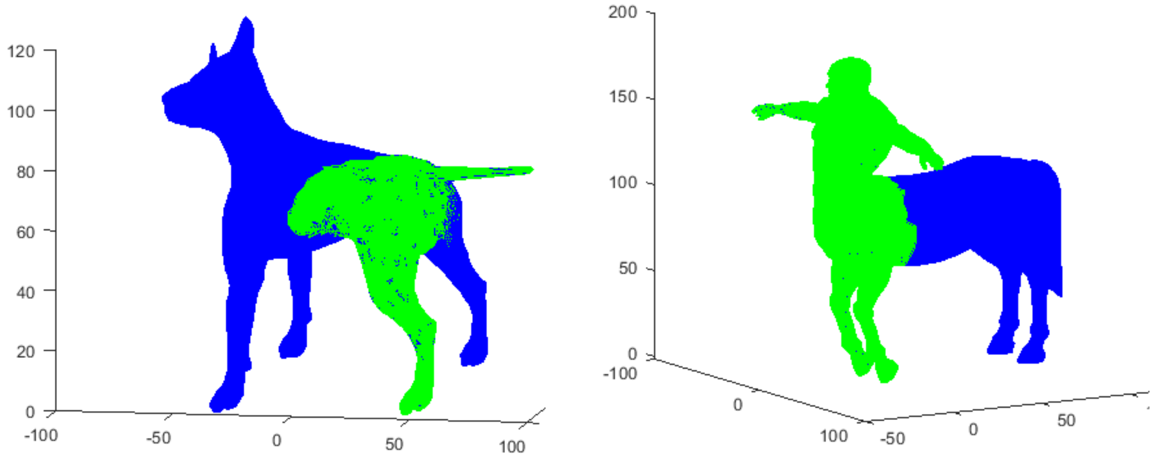


Partial shape of Dog and Centaur.

The partial of the Centaur has 7000 points and 14,000 triangles

The partial of the Dog has 2000 points and 5000 triangles

Will try to fit the partial shape to the whole shape



Whole shape of a Dog and A Centaur, the partial part in colored in green, the rest of the body is in blue.

In this mission we will use Laplace Beltrami and scale invariant Laplace Beltrami as descriptors.

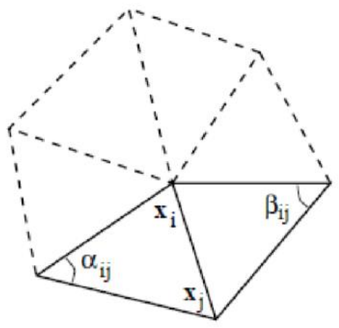
Background

Laplace Beltrami

The Laplace Beltrami is defined as

$$W_{ij} = \begin{cases} -\frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}) & i \neq j,\ j \in N_i \\ \sum_{k\in N_i}\frac{1}{2}(\cot\alpha_{ik} + \cot\beta_{ik}) & i = j \\ 0 & otherwise \end{cases}$$

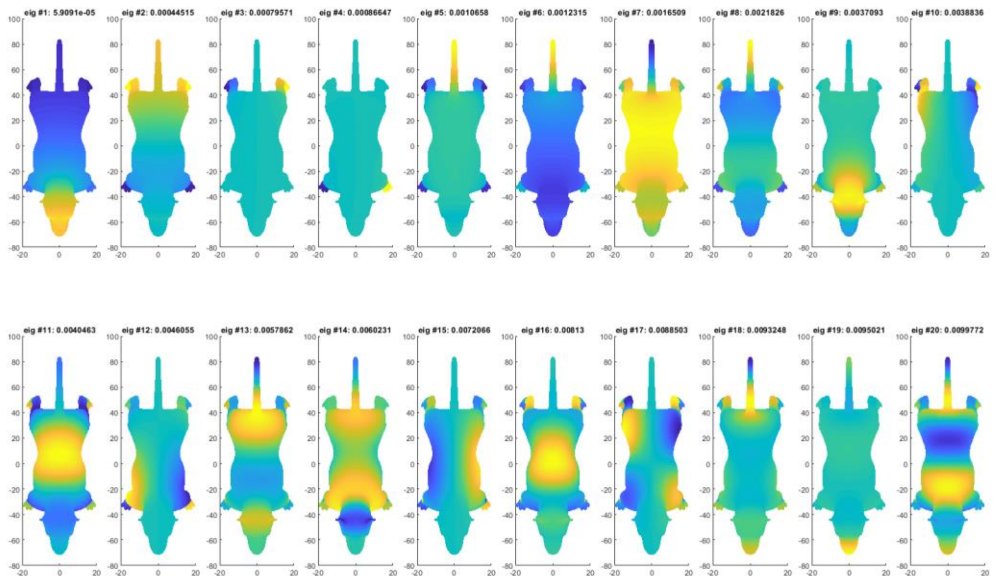
$$A_{ii} = \begin{cases} \frac{1}{3}\sum_j A_{ij} & i \neq j,\ j \in N_i \\ 0 & otherwise \end{cases}$$

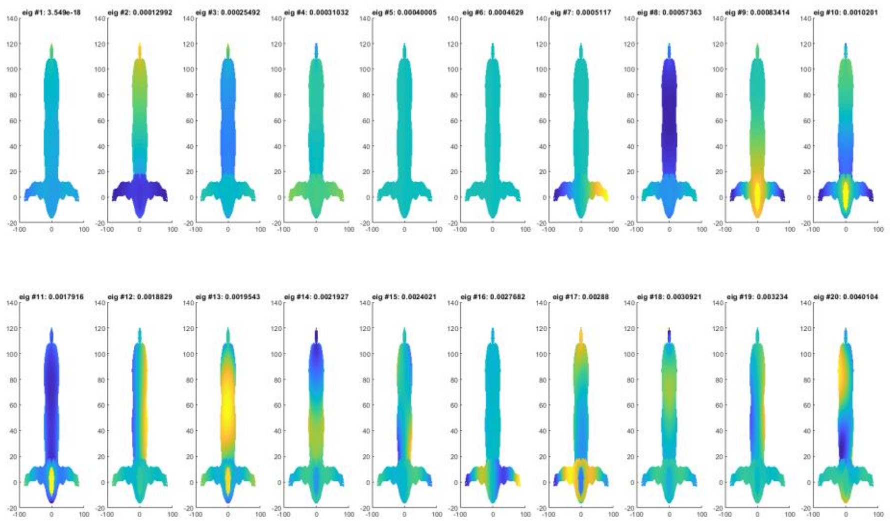


W_{ij} is defined with the angles α_{ij} and β_{ij} of every vertex

A_{ij} is $\frac{1}{3}$ of the sum of all the triangles that limit point x_i

We can decompose Laplace Beltrami matrix to eigenvectors and eigenvalues. The eigenvalues are the frequencies of the functions that compose the shape (these functions are also called eigenfunctions). We can color the shape with the value of every eigenfunction (this is a good sanity check).





These are the eigenfunction of the shape, displaying on the shape.

The eigenfunction go from the lower frequency(top left) and go the 20th frequency (bottom right).

We can see that the lower frequency changes slowly and as it go higher the function changes faster and faster.

Every image title is showing the eigenvalue of the image(which correspond to the function frequency).

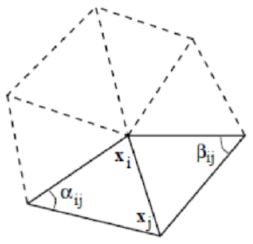
Laplace Beltrami – Dirichlet boundary condition

When calculating the Laplace Beltrami on the partial shape we are having a problem on the edges. The vertex on the edges don't have neighbors all around them and it will give us different results then the results of the same vertex on the complete shape. The body is not close and thus the formula is not valid for this case. A different approach need to replace the current one. The new formula takes care of the edges of the partial shape.

the partial shape Laplace Beltrami formula:

$$W_{ij} = \begin{cases} -\frac{1}{2}\left(\cot\alpha_{ij} + \cot\beta_{ij}\right) & e_{ij} \in E_{int} \\ 0 & i \neq j \; and \; (i \in V_{bdr} \; or \; v_j \in V_{bdr}) \\ \sum_{k \in N_i} \frac{1}{2}(\cot\alpha_{ik} + \cot\beta_{ik}) & i = j \; and \; v_i \in V_{int} \\ 1 & i = j \; and \; v_i \in V_{brd} \end{cases}$$

$$A_{ij} = \begin{cases} \frac{1}{3}\sum_j A_{ij} & i \neq j, \; j \in N_i \; and \; v_i \in V_{int} \\ 0 & otherwise \end{cases}$$



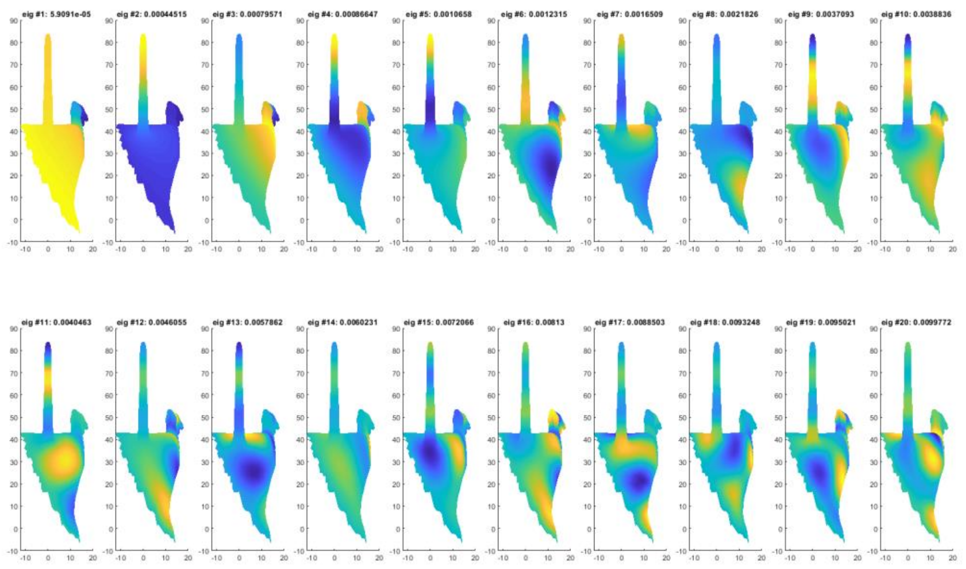
We can see that the values of the functions will be zero on every point on the boundary. If $v_i \in boundary$ then W_{ij} is either 0 when we are not on the diagonal or 1 when we are on the diagonal. And therefore, we get:

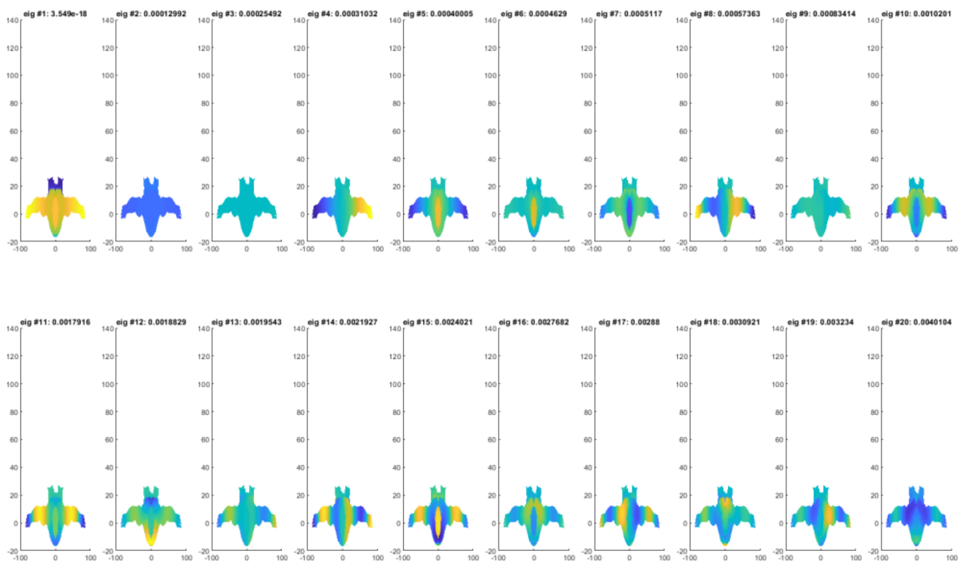
$$(W \cdot v)_i = v_i$$

On the other hand, from the other side of the equation $W \cdot v = A \cdot v \cdot \mu$, we get

$$(A \cdot v) = A_{ii} \cdot v_i = 0 \cdot v_i = 0$$

In order to satisfy the equation v_i must be zero. (this proof will stand also for the scale invariant case)





These are the eigenfunction of the partial shape, displaying on the partial shape.

The eigenfunction go from the lower frequency(top left) and go the 20th frequency (bottom right).

We can see that the lower frequency changes slowly and as it go higher the function changes faster and faster.

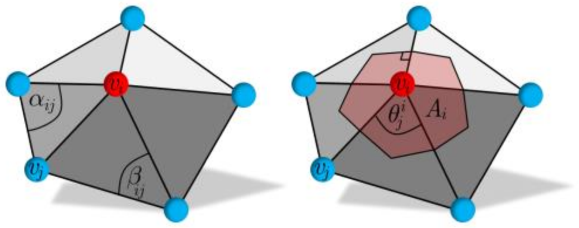
Every image title is showing the eigenvalue of the image.

Scale invariant Laplace Beltrami

We will look now into another descriptor. This descriptor takes into account the convexity of the vertex and those its invariant to scale.

$$W_{ij} = \begin{cases} -\frac{1}{2}\left(\cot\alpha_{ij} + \cot\beta_{ij}\right) & i \neq j,\, j \in N_i \\ \sum_{k\in N_i}\frac{1}{2}\left(\cot\alpha_{ik} + \cot\beta_{ik}\right) & i = j \\ 0 & otherwise \end{cases}$$

$$Q_{ij} = \left|2\pi - \sum_{v_i\in N_i}\theta_j^i\right|$$



In the figure we are explaining the way we are calculating the LBO

We are summing all the angles around v_i in abs.

the calculation of W remains the same.

The extension of the descriptor to Dirichlet condition is the same as in Laplace Beltrami.

Hamiltonian

In this Project we use a mask vector. Defines as

$$(1) \qquad v_\tau(x) = \begin{cases} 0 & x \in R \\ \tau & else \end{cases}$$

Equation 1

the definition of the mask vector v

The purpose of this mask vector is to determine where to choose vertices for the Laplacian calculation. The eigenvalues decomposition of the Hamiltonian with the same notations of the Laplacian Beltrami is in Eq (2)

$$(2) \qquad \left(W + A \cdot diag(v)\right) \cdot \Psi = A \cdot \Psi \cdot \mathrm{diag(v)}$$

Equation (2)

This is the eigen decomposition of the Hamiltonian

Now we will show that the eigen values of the Hamiltonian with the oracle mask v will give us the exact eigenvalues of the partial shape.

Lemma 1

Let $R \subseteq X$ be a region of X, and let $v_\tau : X \longrightarrow \{0, \tau\}$ be a finite step potential defined as in Eq. 1. Then, for the Hamiltonian $\Delta_X + v_\tau$ all the eigenfunctions ψ_i with eigenvalue $\lambda_i < \tau$ vanish pointwise for $X \notin R$.

Lemma 2

Let $R \subseteq X$ be a region of X, and let $v_\tau : X \longrightarrow \{0, \tau\}$ be a finite step potential defined as in Eq. 1. Now let $\{(\lambda_i, \psi_i)\}_{i=1}^{k}$be k eigenpairs of $\Delta_X + v_\tau$ increasingly ordered and such that $\lambda_k < \tau$. If (μ_i, ϕ_i) are the eigenpairs of Δ_R computed with Dirichlet boundary conditions then, $\mu_i = \lambda_i$ for all $i = 1, \dots, k$

The conclusion of both of the Lemma is simple, if we will calculate the eigenvalues of the Laplacian with the correct v_τ then we will get the same eigenvalues as in the partial shape. So the task is to find the correct v.
Another important property of the Hamiltonian is that the eigenfunctions are zero on the egdes. That's why we choose Dirichlet boundary condition!

Derivative of eigenvalues of matrix

In the optimization we need to take the derivative of the eigenvalues of the matrix denote by Δ. In our case the matrix Δ is real and symmetric. Those we can use the Theorem 1 from article (On differentiating eigenvalues and eigenvectors, 1985).

Theorem 1.

Let X_0 ne a real symmetric nxn matrix. Let u_0 be a normalized eigenvector associated with a simple eigenvalue λ_0 of X_0. Then a real-valued function λ and a vector function u arte defined for all X in some neighborhood $N(X_0) \subset R^{nxn}$ of X_0, such that

$$\lambda(X_0) = \lambda_0, u(X_0) = u_0$$

And

$$Xu = \lambda u, \quad u'u = 1, \quad X \in N(X_0)$$

Then

$$d\lambda = u_0'(dX)u_0$$

We will use this theorem for our optimization problem through gradient descent.

We will focus on the derivative of the first eigen value with the first eigenvector.

$$\left\|\lambda\big(\Delta + diag(V)\big) - \mu\right\|_W^2 = \frac{1}{\mu^2}\left\|\lambda\big(\Delta + diag(v)\big) - \mu\right\|^2$$

Let's take the derivative with respect to v_i

$$\frac{d}{dv_i}\frac{1}{\mu^2}\left\|\lambda\big(\Delta + diag(V)\big) - \mu\right\|^2 = \frac{1}{\mu^2}\cdot 2\cdot(\lambda-\mu)\cdot\frac{d}{dv_i}\Big(\lambda\big(\Delta + diag(V)\big)\Big) =$$

$$\frac{1}{\mu^2}\cdot 2\cdot(\lambda-\mu)\cdot u'\frac{d}{dv_i}diag(v)u = \frac{1}{\mu^2}\cdot 2\cdot(\lambda-\mu)\cdot u[i]^2$$

This expression $\frac{d}{dv_i}diag(v)$ is a zeros matrix except of 1 value in the i^{th} place which is equal to $v[i]^2$

For every value in v this expression will be the same, so we get

$$\frac{d}{dv}\frac{1}{\mu^2}\left\|\lambda\big(\Delta + diag(V)\big) - \mu\right\|^2 = \frac{1}{\mu^2}\cdot 2\cdot(\lambda-\mu)\cdot u^\circ u$$

The sign $^\circ$ between $u^\circ u$ means that the vector u is multiply element by element.

The shape of this expression is $nx1$

This development that we just did is valid to every real symmetric matrix and thus its valid for the LBO case.

Method

Minimization

We will try for find v that will bring this optimization problem to minnima

$$(3) \qquad \min_{v\geq 0}\left\|\lambda\big(\Delta_X + diag(v)\big) - \mu\right\|_w^2$$

Equation(3)

The minimization problem

The w norm is defined as

$$(4) \qquad \|\lambda-\mu\|_w^2 = \sum_{i=1}^{k}\frac{1}{\mu_i^2}(\lambda_i-\mu_i)^2$$

Equation (4)

The definition of w norm

In this project we will try to optimize over two Laplacian at the same time. The Optimization expression will be

$$(5) \qquad \min_{v\geq 0}\left\|\lambda\left(\Delta_X + diag(v)\right) - \mu\right\|_w^2 + \left\|\lambda\left(\Delta_{LBO\_X} + diag(v)\right) - \mu\right\|_w^2$$

Equation (5)

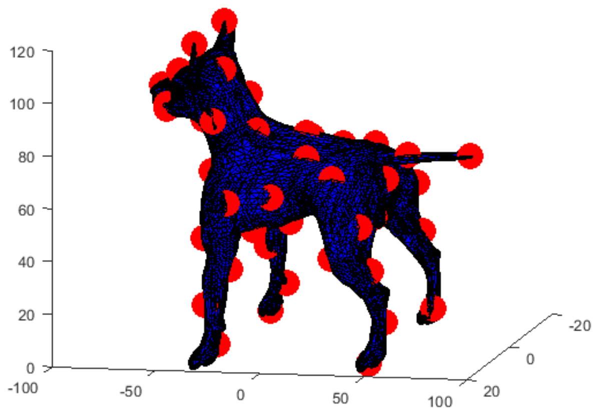
The second optimization objective that we will use as a comparison.

Initialization

In this minimization problem is not convex, thus gradient descent will not necessarily bring us to the global minimum. The way to handle this issue is to initialize v for N different initializations and to run the algorithm N times. Every initial v will get us to some minimum and we will eventually choose the best results that we got.

Initializing V

We will measure the geodesic distance between every pair of points and we will pick a seed points in such a way that every two seed points will be further then some distance D. We will center a gaussian around every seed. The gaussian will change with the geodesic distance and will determine the value of the vector v.



In this image we can see the seeds for the Dog shape.

We will center gaussian around every seed points and we will start the algorithm from every seed.

Initializing τ

τ determines the height of the potential hole. Let's say that we will take k values in account when optimizing the problem. So, we need to set τ value so it will be larger than the k^{th} eigenvalue. The value determines how many eigenvalue will survive in the Hamiltonian. Only the eigenvalues that are smaller than τ satisfy $\mu_i = \lambda_i$. Since there is no harm in picking larger tau, we will choose $\tau = 10 \cdot \lambda_k$

Step size

I create a function that changes the step size. The range of this function is huge, but apparently it works well. The range is:

$$\alpha \in (1e-4, 1e10)$$

When the loss doesn't change we will double the step size and when the error jump up and down we will update the step size to be 4 times smaller.

Number of iterations

We are doing 550 steps for every problem.

Results

In this section we will show 2 results for every shape.

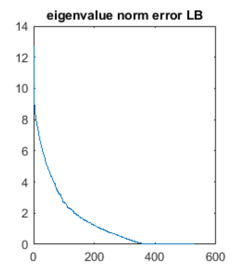
1. With the original implementation of the paper
2. With the extension shown in Eq(5), with LBO.

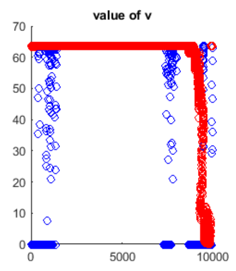
Every result is combined with 6 – 8 graphs

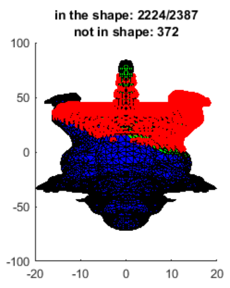
1. Top left shows the error that we obtain by Eq(3)
2. Bottom left the k_{th} eigenvalues of the Hamiltonian(X) and the eigenvelues of the partial shape (Y)
3. On the second column we have the same as 1 and 2 just with the LBO
4. Top 3_{rd} column we have the value of the mask vector v,
   the red dots are the initial values and the blue dots are the current v values.
5. Bottom 3_{rd} column we have step size α
6. Top 4_{th} column we have the whole shape in blue,
   the partial shape in green and the current v values in red dots that is taken.
7. Bottom 4_{th} column is the number of points that are taken in account in the partial shape(points from v).
   on the title we see how many TP points we have and how many FP we have
   (We consider value of v to be in the partial shape if its bigger then some threshold).

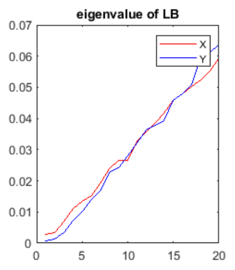
Dog

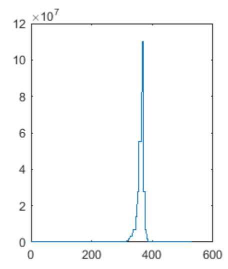
Here we use only LB

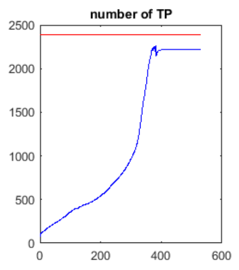




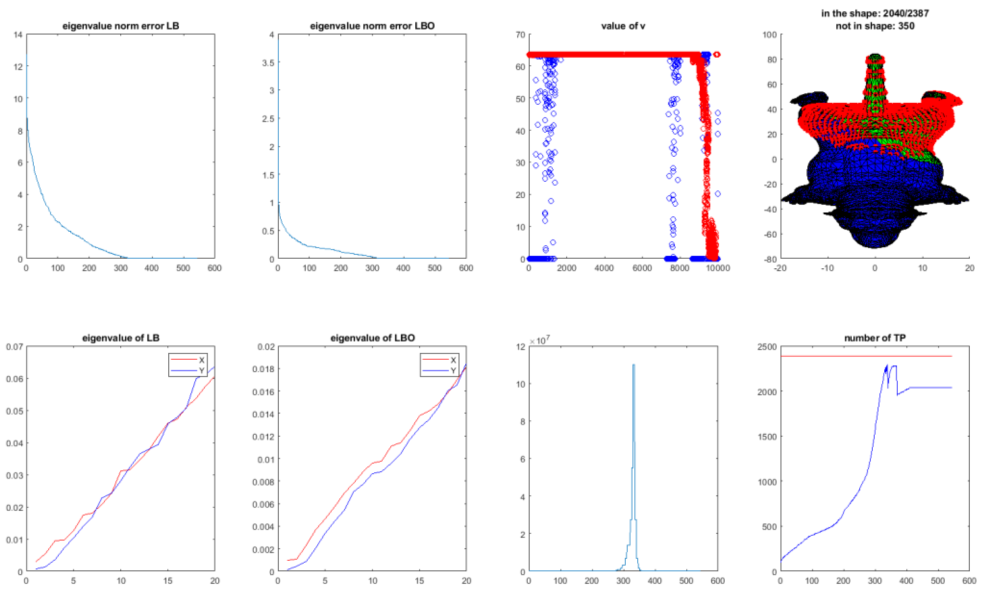






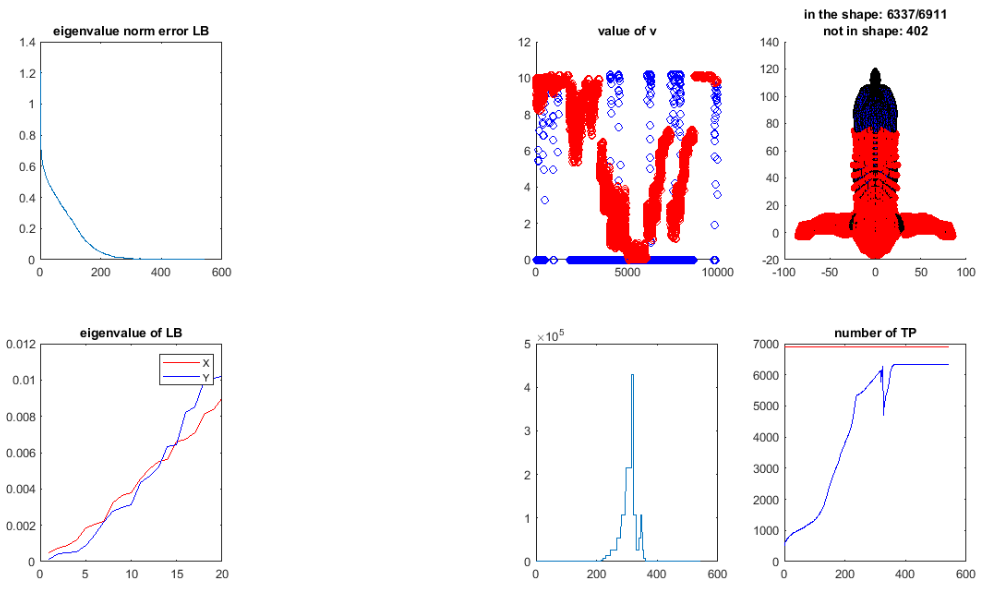


Here we use LB and LBO

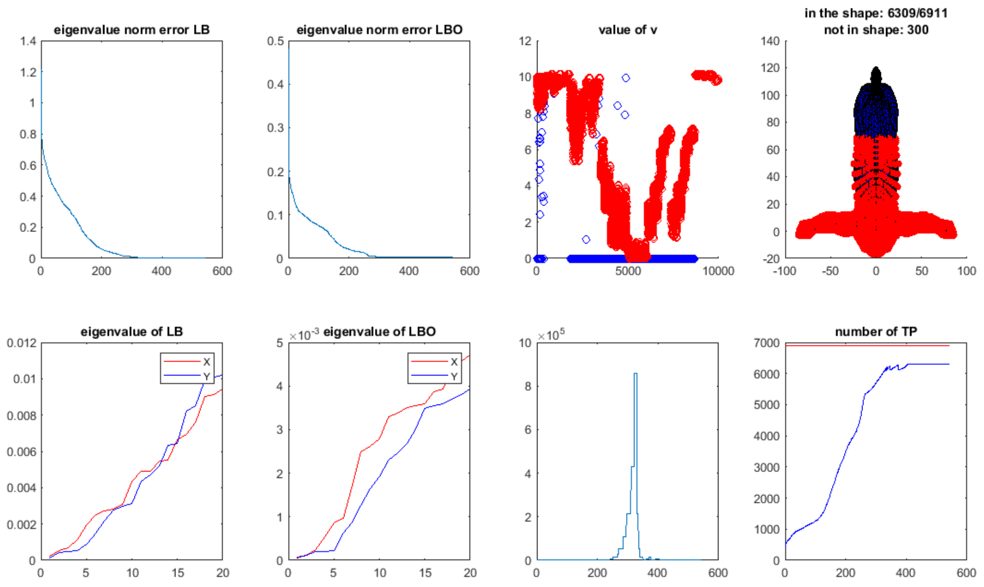


Centaur

Here we use only LB



Here we use LB and LBO



Discussion

So far the results are not conclusive. I can't say that one method is better than the other, they are pretty much the same. With the same number of steps and the same step size function we get similar TP and FP numbers.