INF 574: MESH PARAMETERIZATION

9 décembre 2022

CUI Zhipu, XIA Junkai

1
INTRODUCTION TO THE PROJECT

In order to texture a 3d model, a common practice is to map the 3d shape onto a 2D surface and add the texture to the surface, which is known as parametrization. There are many algorithms in this field with a large number of classical techniques.

In our project, we have implemented the Least Squares Conformal Maps(lscm) algorithm, mainly based on conformal maps to preserve the information for the original 3d shape. Our work is mainly based on the article *Least Squares Conformal Maps for Automatic Texture Atlas Generation* (Lévy B et al.).

2
PROCEDURE

The parametrization process is divided into three parts, segmentation, parameterization and packing.

— Segmentation : We try to divide the 3d object into several components, each of which is isometric to a disk.

— Parameterization : For each component, we have implemented the algorithm to project the 3d meshes on to a plane by solving linear equations.

— Packing : After we have calculated the 2d projection of each components in their own coordinate system, we then tried to unify these meshes into one general coordinate system and then put in on a plane.

3
OUR WORK

3.1 Parameterization

For a given 3d mesh in .obj file, we first wrote some IO interface to read and manipulate the files, thus obtaining a point cloud containing all the coordinates of the object and all the facets of the mesh.

We first transform each facet to a triangle by projecting the coordinates of its vertices onto a plane.

For each vertex, we use the complex number $\mathcal{X} = x + iy$ to represent its coordinates in the local orthogonal base, and $\mathcal{U} = u + iv$ to represent its coordinates after parameterization.

According to Cauchy-Riemann equations, we have

$$\frac{\partial \mathcal{U}}{\partial x} + i\frac{\partial \mathcal{U}}{\partial y} = 0$$

which cannot be satisfied by trangular meshes. So we try to minimizing the energy

$$C(T) = \left| \frac{\partial \mathcal{U}}{\partial x} + i\frac{\partial \mathcal{U}}{\partial y} \right|^2 A_T$$

By constructing the gradient matrix of the meshes and solving the linear equation, we can finally get the matrix of the coordinates of the vertices on the 2d plane.

We notice that

$$Number(vertices) - Number(triangles) \leq 2$$

So we fix the coordinates of the two points furthest apart to make the equation solvable.

3.2 Segmentation

For each of the edges of the mesh, we first calculate its sharpness, which is represented by the angle formed by two normal vectors of its neighboring faces.

We kept one percent of the edges of the highest sharpness, which are most likely to be the boundaries of different charts. Then we implemented the algorithm to look for the feature curves by applying DFS algorithm on each of the edges. Halfedge structures are used during the process.

Then we applied the Dijkstra algorithm to calculate the distance to the set of feature edges for each facet. Charts are created based on the facets whose distance to feature being local maximum. Charts are then merged if their distances are smaller than a certain threshold.

3.3 Packing

After the original mesh is divides into several charts, we project each of them to a corresponding plate. Then we calculate both horizontal and vertical diameters to represent the boundaries of these 2d meshes. Finally we use the rectangle packing to put all the meshes in a single plate. We didn't copy the algorithm given by the article witch always minimises the space surrounded by the bottom line of the chart waited to be inserted and the horizon of the existing charts. Instead, we have simply used the rectangle boundary, and used rectangle packing solver to get the final result.

4 RESULTS

We have tested our projection algorithm on a saddle, the result is satisfying.

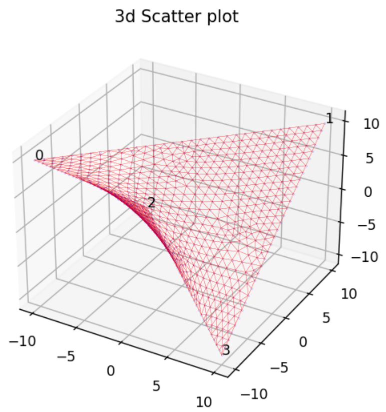

Figure 1 – A saddle in 3d space

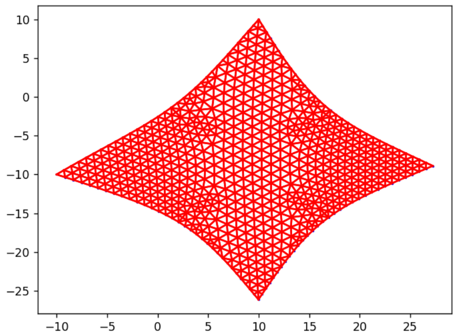

Figure 2 – Its projection on 2d surface

We have tested our feature extraction algorithm on the mesh of a rabbit. The feature curves are represented in red on the left image. We can see that its ears, head, legs, tail are well separated. The right one shows the different charts which are represented in different colors.

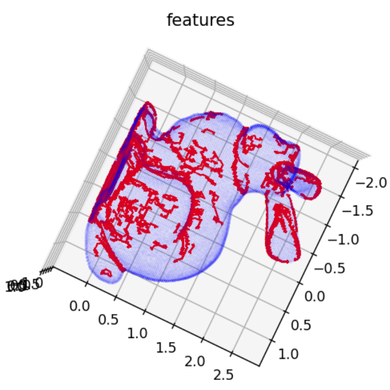

Figure 3 – Feature curves of a rabbit mesh

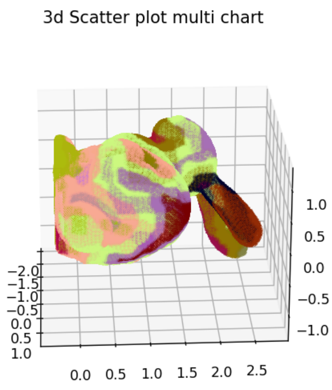

Figure 4 – Different charts of the mesh

Each chart is projected on a 2d surface. An example is given below

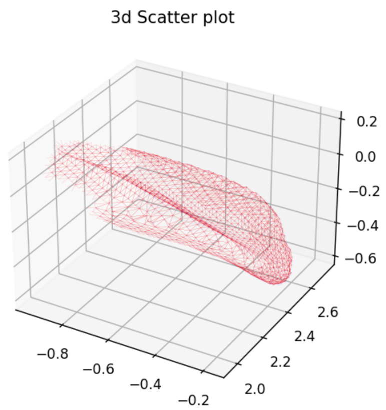

Figure 5 – The rabbit's ear in the 3d space

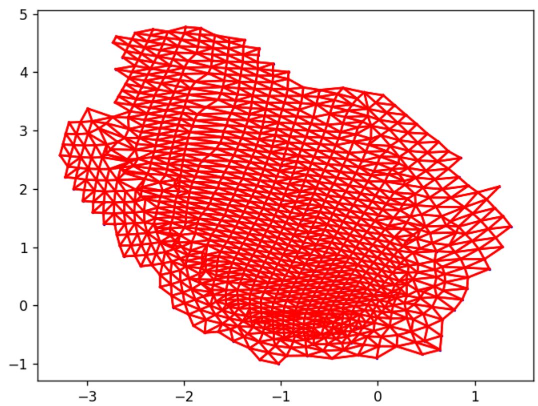

Figure 6 – Its projection on 2d surface

Finally the charts are projected on a same surface by using packing algorithms. The local coordinates are transformed and put in a general coordinate system. The results are shown as below(links between the vertices are not plotted).

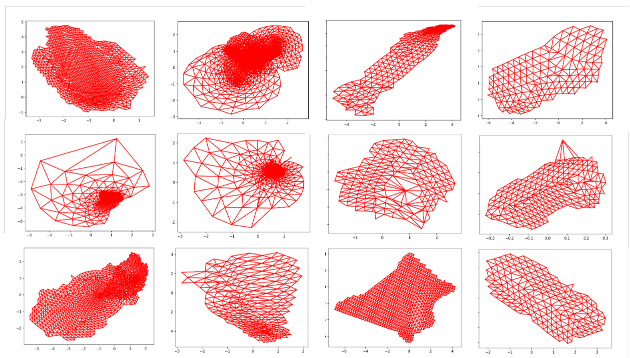

Figure 7 – Part of the collections of the 2d projections of rabbit's mesh

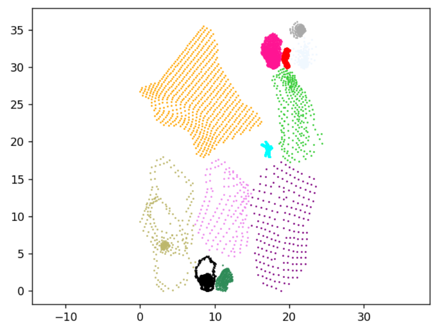

Figure 8 – Final parameterization of the rabbit mesh

We have also tested our algorithm on other meshes, the result are shown as below.

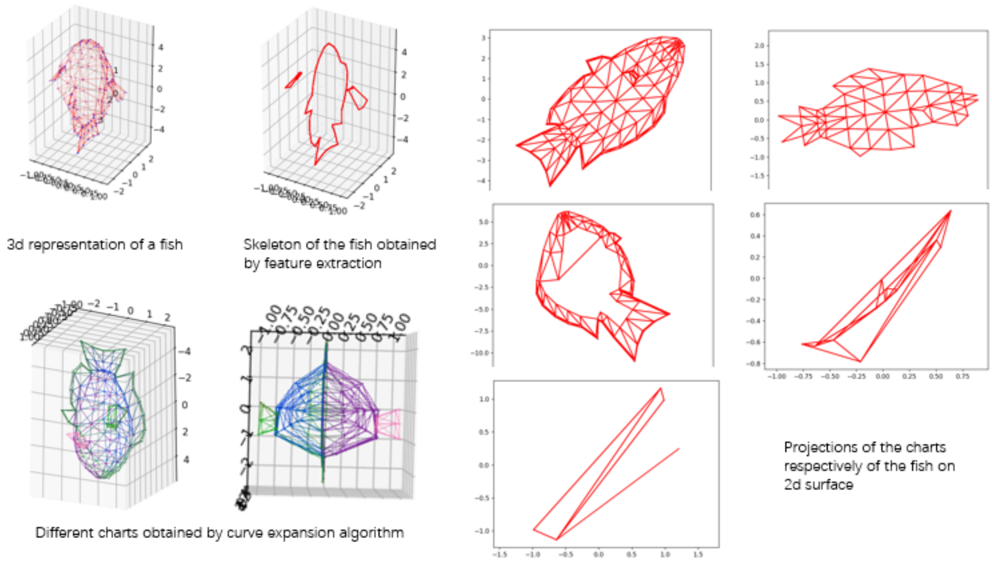

Figure 9 – The results of our algorithms applied on a mesh of the fish

5 PROBLEMS THAT NEED TO BE IMPROVED

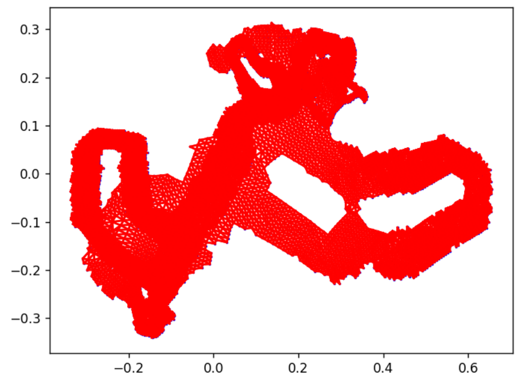

Figure 10 – Example of a chart not well manipulated

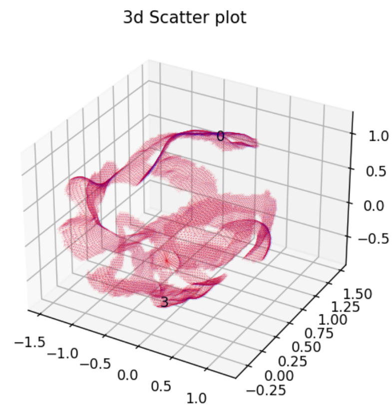

Figure 11 – There may be intersections after the projection process

There are still some problems to be solved. Because of the limitation of our computers, for the feature extraction part we have chosen one percent of the edges with the highest sharpness instead of the five percent as indicated in the origin article. As a result, some of the charts obtained may not be isometric to a disc, which results to the overlapping and intersections after the projection process.

There exist also some problems like the intersection for the fish fin as shown above. This may be solved by tuning some key hyper parameters like the max sequence length utilised in the feature extraction process.

6 CONCLUSION

In this project, we have implemented several methods to transform a 3d mesh into the combination of several 2d shapes.

However, there are still several problems remained to be solved, In the original article, the author kept five percent of the edges as features, which is a bit overwhelming for our computers. We have kept only one percent in practice, as a result, some of the facets are not classified correctly, which leads to the charts not isometric to a disk.

Moreover, we have just implemented the rectangle packing instead of the algorithm described in the article witch saves mush space for the final packing.

7 SOURCE

1. Lévy B, Petitjean S, Ray N, et al. Least squares conformal maps for automatic texture atlas generation[J]. ACM transactions on graphics (TOG), 2002, 21(3) : 362-371.

2. Floater M S, Hormann K. Surface parameterization : a tutorial and survey[J]. Advances in multiresolution for geometric modelling, 2005 : 157-186.

3. Hormann K, Lévy B, Sheffer A. Mesh parameterization : Theory and practice[J]. 2007.

8 WEBSITE

You can find our code and the meshes on github : https ://github.com/xiajunkai328/inf574