project report

Fa Long

April 7, 2017

1 introduction

The design of tangent vector fields on discrete surfaces is a basic building block for many geometry processing applications, especially in geometric processing and computer graphics, suppose we have an object, we want to measure and save some geometric or physical properties of the object. So instead of the mesh or point cloud that we generate from the object to describe its shape, we need vector fields to represent those properties. Such as advection field or strain tensor of the object. In order to represent different values clearly, we use different types of vector fields. N-polyvector is one of these vector field. Smoothness of vector fields is usually measured and optimized as the L^2 norm between the coefficients of the polynomials that represent the field. This is done only for efficiency reasons, but it is unclear how this relates to the curl of the field, which is the property that is more interesting for applications. The idea of the project is to explore how different norms affect curl of the field and its singularity structure. We minimize L^p norms rely on IRLS (Iteratively reweighted least squares) which is only slightly more expensive than minimizing a L^2 norm. This project can be used for global parametrization and quadrilateral meshing.

2 previous work

2.1 n-polyvector field

The ubiquitous Rotationally-Symmetric fields, commonly denoted as N -RoSy fields, are special vector sets comprising N unit-length vectors related by a rotation of an integer multiple of 2/N. however, N-rosy field are often too restrictive for application. Designer often wants to manually control the anisotropy of the field and allow deviation from uniformly sized quads and right angles in order to increase the design space and better adapt the discretization to the underlying shape, its semantics and articulation. So N-polyvector is been introduced, N-polyvector is a novel representation for general unordered vector sets in which no vector is necessarily related by any symmetry or magnitude to another. the presentation of n-polyvector is by the set of coefficients of a complex polynomial. We'll have further discuss about this later.

2.2 N-PolyVector fields with complex polynomials

Given a triangle mesh $M = V, \varepsilon, F$, we consider the plane of every triangle $f \in F$ as a discrete tangent space, and the entire collection of these planes as the tangent bundle of the surface. Consequently, in our framework, tangent vectors are defined on the faces of the triangles, and the tangent vector fields are thus piecewise constant. A vector field is defined as a given assignment of a single vector per face (formally, a section of the tangent bundle). A general N -PolyVector, $N > 0$, is an unordered set of N tangent vectors in a single face, and an N -PolyVector field is defined accordingly.

2.3 representation of n-polyvector on single face

Before we generate the n-polyvector field for all the faces in F,we first introduce how it can be represent on a single triangle face.
We know that a N-rosy field consist following vectors on each face:

$$\left\{ u_f exp(i * \frac{2\pi * k}{N}) | 0 \geqslant k \geqslant N-1 \right\}$$

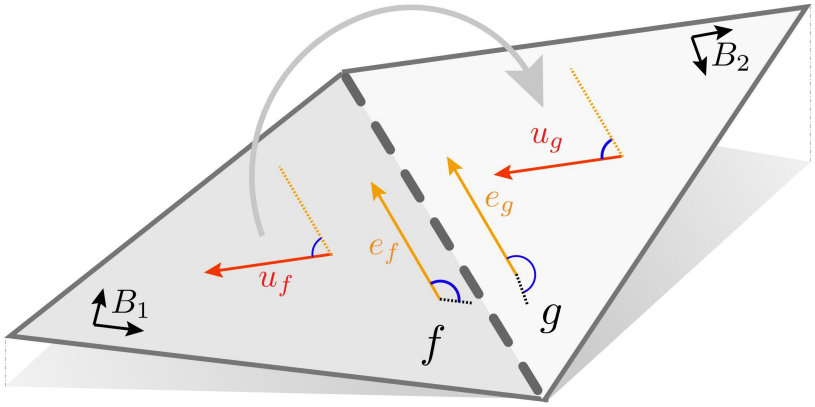
which can be simply represent by the N-th root of a complex number(u_f^N). so An N -RoSy field can be represented as the variety (root set) of the following complex polynomial: $p_f(z) = z^N - (u_f)^N$. It's immediately clear how we could generalize this representation to unambiguously encode a general unordered set of face-based vectors$\{u_0, u_1, ..., u_{N-1}\}$

$$P_f(z) = \{(z - u_0)(z - u_1)...(z - u_{N-1})\}$$

Alluding to the polynomial representation, we denote these unordered vector sets and their polynomial representation as N-PolyVectors, or N-PV in short.

2.4 LC connection

After we know how to represent polyvector on single triangle(face), now we consider how we could translate it from one triangle to another. here we using the LC-connection to achieve this.
We identify every tangent space (face) f with the complex plane C by choosing an arbitrary orthonormal basis. A vector in the tangent plane is then defined by a single complex number. However, two adjacent faces have different bases, and in order to compare adjacent vectors we require a discrete connection , which defines a parallel transport between neighbouring bases.



The discrete Levi-Civita (LC) connection is the change of bases between to different faces $f, g \in F$ across a common edge $e \in \delta$,which is defined separately bye_f, e_g, Formally, the vector u f in the tangent space of face f is LC-parallel to the corresponding vector u g in the tangent space of face g if and only if $u_g = u_f(e_f)^{-1}e_g$.
This formulation is conceptually equivalent to unfolding the triangle flap (f, g) onto a plane and simply translating the vector. This intuition leads to an equivalent, symmetric and more intuitive formulation: Following the flattening of the flap, we may unify the two representation bases by assigning a new mutual basis, so that the mutual edge vector e becomes the canonical real axis in both. Then the two vectors, one in each face, are LC-parallel if and only if they have an identical representation with respect to the unified basis. In this we, we can translate the polyvector field on one face to every face on our sample, and get a polyvector fields for the entire surface.

2.5 LC-parallel N -PolyVectors

LC-parallel N -PolyVectors are defined by having individual matching LC-parallel roots across edges. Instead of transporting the roots directly, we transport the coefficients of the polynomials (which are also vectors, or complex numbers).
Connections, as parallel transports, induce curvature inside closed domains. This curvature is measured by the change in vectors as they are parallel-transported around the boundary of that domain. The LC connection induces the Gaussian curvature, which, in the discrete setting, amounts to the sum of angle defects enclosed within the transport path. Con- sistent vector field design is equivalent to defining a trivial connection between faces. A N -PV field is considered perfectly smooth across an edge if its polynomial coefficients are LC-transported, which then means that every vector on one face is LC-transported to a single vector on another face. Therefore, we wish to solve for N -PV fields that define a trivial connection that is as-LC-as-possible.this means that we need to minimize the energy:

$$E_{smoothness} = \sum_{m=0}^{N} \sum_{a,b \equiv \varepsilon} |A_{ab} - A_b|^2$$

$A_{ab} = e^{in_A k} A_a$, which is the transport of coefficient A from triangle a to b.
minimize this energy is done by solving a sparse linear system,we'll discuss it in later chapter, after that, we can do the factorization and get the n-polyvector filed from the corresponding polynomial .note that this is in the least square sense, it's only because of the efficiency reason. instead of doing it this we, we try to minimize the energy using L^p norm, and evaluate the vector field and distribution of the energy we get using this new method.

3 our method

Now we trying to minimize the energy using L^p norm. this will change the topology of the n-polyvector field we generate, we apply IRLS (Iteratively reweighted least squares) to our method so that the calculation could be easier and more similar to our formal method.
so the L^p energy we want to minimize for each coefficient is:

$$\sum_{ab} |A_{ab} - A_b|^p$$

notice that $A_{ab} = e^{in_A k} A_a$, which is the transport of coefficient A from triangle a to b. according to IRLS, minimize this value equals to minimize:

$$\sum_{ab} W_i(l) |A_{ab} - A_b|^2$$

where $W_i(l) = |A_{ab} - A^{l-1}{}_b|^{p-2}$,$W_i(0) = 1$,and l represent the regression time

we want.
then the L^p differences for each coefficient is:

$$E_p(A) = w_{ab}|A_{ab} - A_b|^2 = w_{ab}(A_a^* A_a - e^{in_A k} A_a^* A_b + A_b A_b^*)$$

denote that w_{ab} is a element for edge ab in $W_i(l)$,and it's a fixed number,this means that the final energy we want to minimize equals to

$$E_D^{(A)} = \tilde{A}_a^* D_A A$$

where $D^{'}$is a little different from current one that:
For each unordered triangle flap (a,b):
Add w_{ab} to element (a,a) of D_A
Add w_{ab} to element (b,b) of D_A
Add $e^{in_A k} w_{ab}$ to element (a,b) of D_A
Add $e^{in_A k} w_{ab}$ in A k to element (b,a) of D_A.
Then the polynomial coefficients are harmonically interpolated by solving a sparse linear system, which is prefactored using a sparse LU factorization:

$$D_A \tilde{A} = 0, s.t \tilde{x_j} = \tilde{x_j}^0, j \in C,$$

where$\tilde{x}$ stacks all the degree- m coefficients for all faces in a single vector. The variables that correspond to the user constraints (C is the set of constrained faces) are fixed to the values $\tilde{x_j}^0$, extracted from the directional constraints.
After first calculation, we get the output coefficients ,using this we could calculate $W_i(1)$ by calculating the L^p norm of two coefficients within two adjacent triangles.

$$w_{ab} = \frac{1}{[(x_a e^{in_A k} - x_b)^{2-p} + (y_a e^{in_A k} - y_b)^{(2-p)}]^{\frac{1}{2-p}} + 0.0001}$$

w_{ab} is the update weight we apply to our function each time. and we loop back to generate $D_A^{'}$ again.same work for all other coefficients.
After obtaining the coefficient fields x_m forevery monomial degree m , we can factor the face-based polynomials to produce the roots, i.e., the actual vector sets. The root-finding is done by finding the eigenvalues of the companion matrix. An important property arises from this computation: the coefficient of each monomial degree is computed independently from others, and since the interpolation is harmonic, it obeys the maximum principle. Therefore, if the interpolation constraints for a certain coefficient degree are all zero, the entire mesh will get a zero coefficient for that degree. For instance, if the provided constraints are purely N -RoSy, the entire interpolation result is N -RoSy by definition since only the constant coefficient (x_0) will be interpolated. Moreover, degenerate configurations which are the result of coefficients $x_m, m \geqslant 1$, with a large magnitude can only be obtained if forced by the user-provided constraints

4 result

Our suggestion is that with L^2 norm the smoothness energy of the surface will equally distributed on each and every edge. while L^p norm tending to put all the energy on some of the edges and keep rest of the edge 0-energy. This also effect the topology property of the vectorfield we plot on the surface.with different p, energy could also be different.
we apply our method and formal method on different surface, and use color to represent the distribution of the energy on the surface. We also visualize the vectorfield we generate, with camera fixed, we could see the differences more clearly.

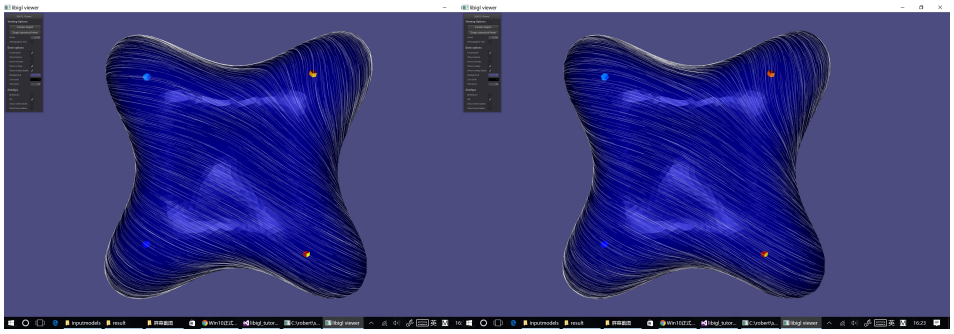


Figure 1: p=0.1,energy=143689

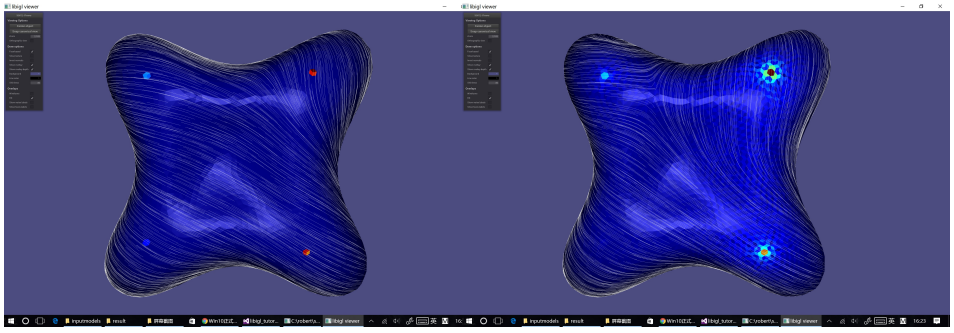


Figure 2: p=0.9,energy=327.041

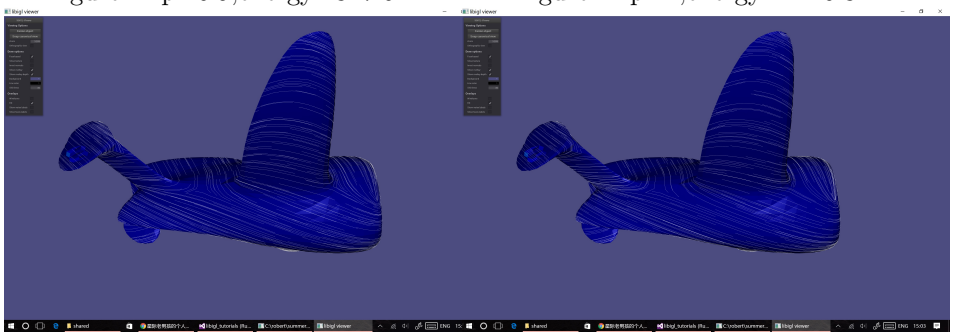


Figure 5: p=0.1,energy=93500.7

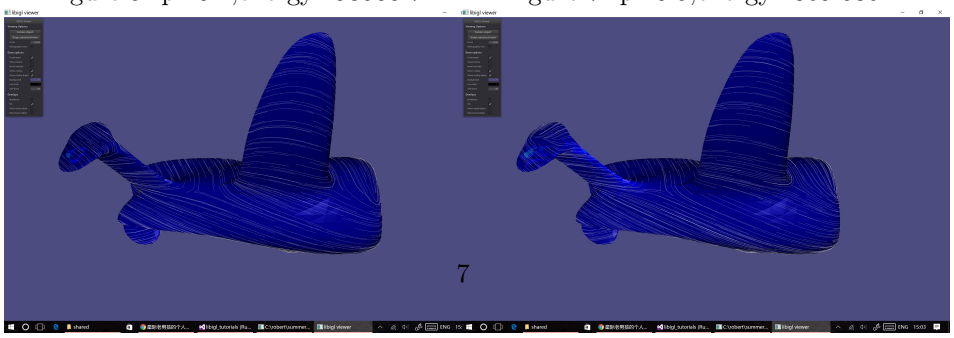


Figure 6: p=0.9,energy=201.171

Figure 3: p=0.5,energy=583.45

Figure 4: p=2,energy=2240.32

Figure 7: p=0.5,energy=369.036

Figure 8: p=2,energy=558.753

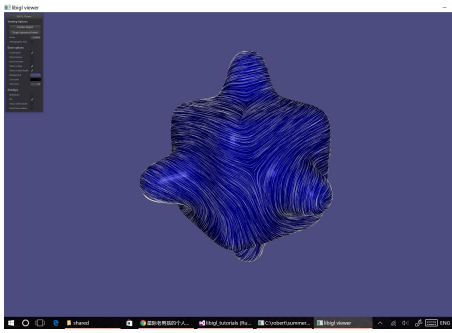


Figure 9: p=0.1,energy=143330

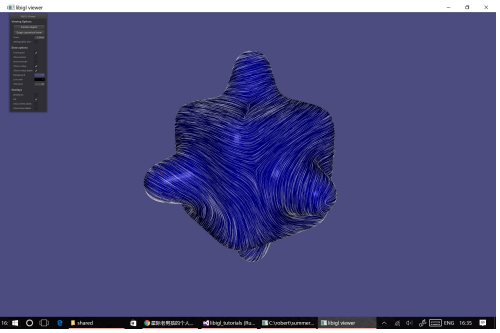


Figure 11: p=0.5,energy=593.934

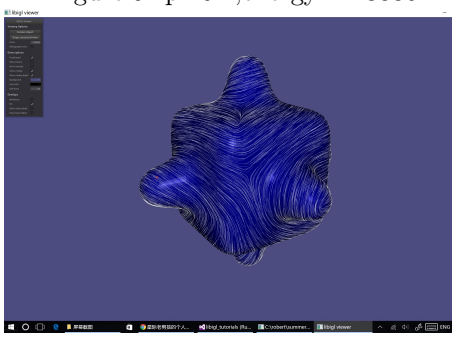


Figure 10: p=0.9,energy=338.956

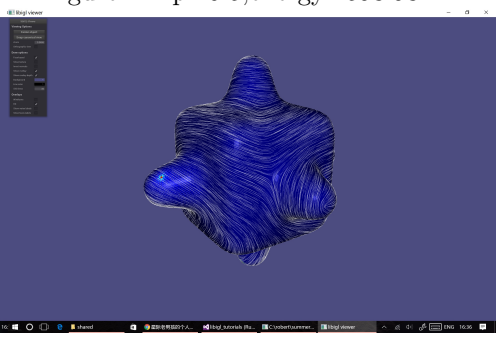


Figure 12: p=2,energy=3398.07

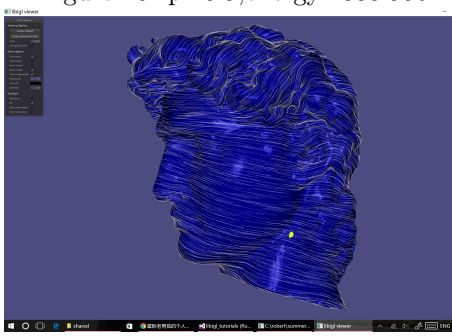


Figure 13: p=0.1,energy=163316

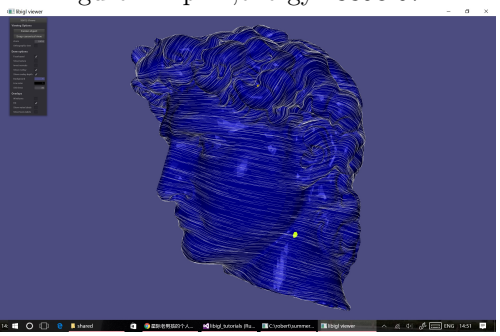


Figure 15: p=0.5,energy=651.608

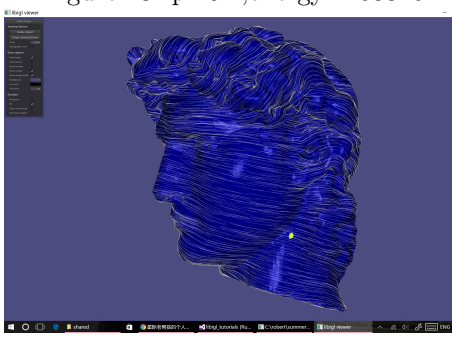


Figure 14: p=0.9,energy=359.255

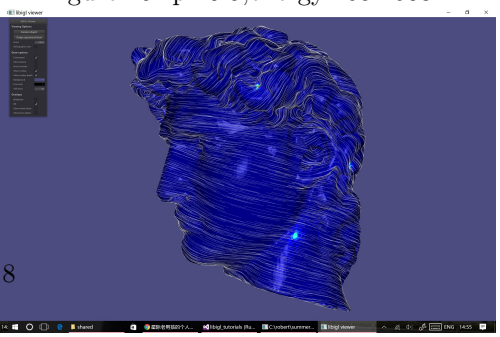


Figure 16: p=2,energy=1900.17

For L^p situation with IRLS, We find that when regression times up to 20, the energy tend to stable, so in our test we fix the regression times to 20.
From the figures above we proved that most of our suggestion is true, with L^p norm, when the regression times goes up, smoothness energy on the surface is concentrated on the constraint faces,and total energy is decreased exponentially, while L^2 norm will distribute smoothness energy around the constraint surfaces. which is represent by color on each figure. for L^p norm, when p rise up, total energy of smoothness is going down.
We only generate 2-vectorfield and it's symmetric, which is represented by white streamline on each figure, there is a large difference between L2 and the others, but surprisingly there is almost no difference for p smaller than one. There also seems to be a lot less singularities in the L1 results, and the fields is more straight. For more examples,please go to: test result

5 issues/future work

5.1 problems

We use IRLS to minimize the L^p norm, but this arouse a problem, the length of the vectorfield will become shorter and shorter after each iteration. currently we normalize the vectorfield after we generate it for visualization. It seems like the method we apply simply just minimize the energy by decrease it to nearly zero, and only the direction remain. so before this problem being solved, it's not good for parametrization.
All the figure in this report uses streamline to visualize the polyvector, But there is a bug inside this code, when the example we loaded is big, our we reset the constraints for too many times, the code may crash. Currently I'm using another way to visualize the vector. which is not streamline, it's bug-free. but the quality of the vector field is not as good as the one with streamline.
The code is only used for test and demonstration for L^pnorm with 2-polyvector field, when n is not equals to 2, the cold may crash. see test code here

5.2 future work

Like we mentioned before, We want to explore how different norms affect curl of the field and its singularity structure. So next we may want to extract the singularities for this kind of polyvector fields. But as we mentioned in 5.1, we will need to add some additional terms that preserve the magnitude of the roots so that it could be used for quadrilateral remesh, parametrization or other applications.

6 reference

http://libigl.github.io/libigl/tutorial/tutorial.html
http://cs.nyu.edu/~panozzo/papers/n-polyvector-fields.pdf

https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares