Optimal direction fields with alignments

Nihal Poosa
nihalp@bu.edu

Prithviraj Khelkar
pkhelkar@bu.edu

Mahaveer Bonagiri
mahaveer@bu.edu

Abstract

The paper, Globally Optimal Direction Fields 2013 [2], provides a way to find smooth directional fields where the singularities show up in natural places by the energies defined in the paper. In this course project, we try to add new singularities by providing alignment vectors (as specified in the paper) around a singularity. These alignment vectors have been obtained from the Trivial Connections on Discrete Surfaces 2010 [1]. We have implemented, in the framework provided by the authors of the Globally Optimal Direction Fields [2], a way to provide any random vectors as alignments while the framework currently only implements curvature and boundary alignments for line fields and cross fields. We have also implemented line fields and cross fields where the alignments from Trivial Connections 2010 paper [1], which are single vectors, are converted to line fields or cross fields by rotating appropriately in the tangent space and these line fields and cross fields are provided as alignments for the line fields and cross fields that are to be computed, in a polyscope [3] visualizer.

1. Background

Before we describe the methods used to obtain the results, we provide an overview of the parts of both the papers, Globally Optimal Direction Fields 2013 [2] and Trivial Connections 2010 [1], that are important to the current project and the parameters which are used in those papers.

1.1. Globally Optimal Direction Fields [2] overview

This paper finds the optimal direction fields with singularities at natural positions based on the energies specified in the paper. In this paper to obtain the vector fields to a given alignment vectors we need to solve the equation:

$$(A - \lambda_t M)\tilde{u} = Mq$$

The parameter λ above shouldn't be confused with the t which is used to describe the alignment energies. An important thing to note here is that even though the parameter t intuitively specifies which energy is given more weight among the smoothness energy and the alignment energy, it is not explicitly controllable and cannot be inputted to the algorithm as such.

Instead the parameter λ can be inputted to the algorithm to solve the equation, and the parameter t behaves like so:

$$\lambda \in (-\infty, \lambda_1) :: t \in (-1, 0)$$

where λ_i = smallest eigen value of A. We can only change the input parameter λ and see what the corresponding t becomes. Another important thing to note is that this paper works on vertex based vector fields unlike the other paper, Trivial Connections 2010 [1].

1.2. Trivial Connections on Discrete Surfaces [1] overview

This paper finds the smooth face based vector fields on a mesh from a given set of vertices as singularities and their corresponding index values. The sum of all the index values should be equal to the Euler Characteristic of the mesh according to the Hopf-Poincaré Theorem. Point to be noted would be that the resulting vector fields from this paper are face based unlike the Globally Optimal Vector Fields 2013 [2] implementation whose vectors are vertex based.

2. Methodology

2.1. Alignment Fields around singularities

We take the vertices where we want the singularities to be, decide on the index values for those singularities and make sure these sum up to Euler Characteristic of the given mesh. We then obtain face base vector fields [1] based for this mesh. We then take the vector fields only around the singularities as alignments. We obtain these vector fields around the singularities by getting the faces that are at a particular distance from the singularity (here distance is the number of vertices to singularity, and is not to be confused with geometric distance). We obtain this distance using breadth first search around the singularity where the vertices are nodes and edges are edges of the graph. Figure 1 is an example of Trivial Connections where the distance to the singularity is 5.

Now we need to convert the face based vector fields of the faces attached to a vertex to a complex number that can

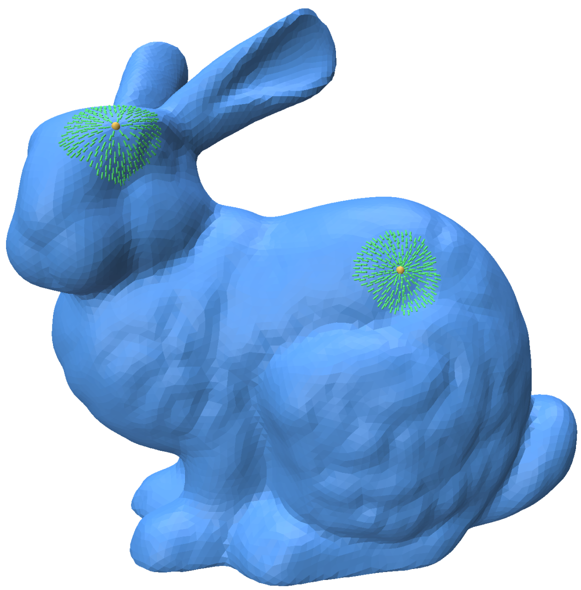


Figure 1. 5 Rings alignments on bunny mesh

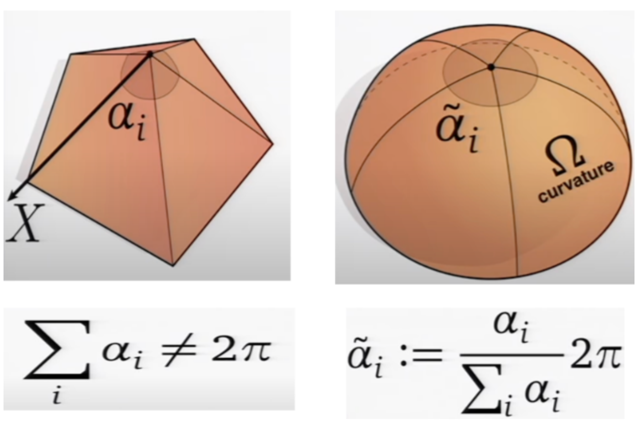


Figure 2. curvature

be used in [2] as an alignment field. For that we first convert the face vector field on every face attached to a vertex to a complex number by finding the angle with the basis vector in the tangent space of the hemisphere, after the current pyramid shape is bent into a hemisphere. Basically all the angles in the current pyramid around the vertex are scaled to 2π.
To do this, first we find the face vector's angle with one of the edges, then add it to the sum of angles of that edge going over all the faces that come in between to the basis vector edge and finally scaling it to 2π using

$$\tilde{\alpha}_i := \frac{\alpha_i}{\sum_i \alpha_i} 2\pi$$

Figure 2 is an example of conversion from pyramid to hemisphere.

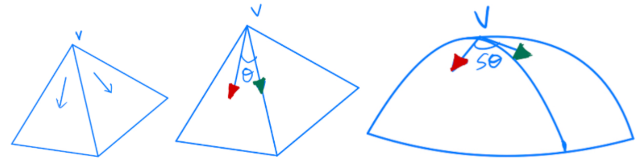


Figure 3. Face to Edge

Figure 3 is an example of converting the angle of a face vector to the edge vector of the face into the hemisphere.

Now that we have the angle in the tangent space or in the hemisphere from the face vector to the basis vector, the complex representation of the face vector becomes e to the power $i*(scaledangle)$. To find the complex representation of the vector on the vertex we just average all the complex representations of vectors on all the faces attached to it by the areas of the faces. Now we have the complex number for each vertex which represents the alignment field and we solve the above equation

$$(A - \lambda_t M)\tilde{u} = Mq$$

where q are the alignments that we just calculated.

2.2. Implementing cross fields and line fields

For this we raise the alignment complex number to the degree that we want (2 for line fields and 4 for cross fields) and solve the equation with the new raised complex numbers as q. After obtaining the $\tilde{u}$ we find the nth roots of the complex numbers at each vertex and these nth roots become the n-vectors at each vertex. We've only done this for line fields and cross fields because the other n-vectors don't make much sense in general and are a little more difficult to visualize as of now.

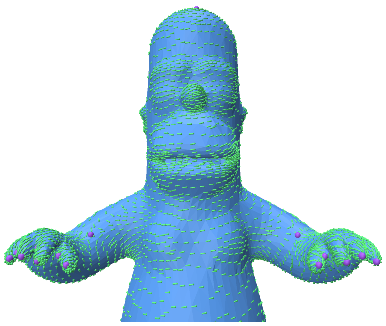
3. Results

3.1. Alignment

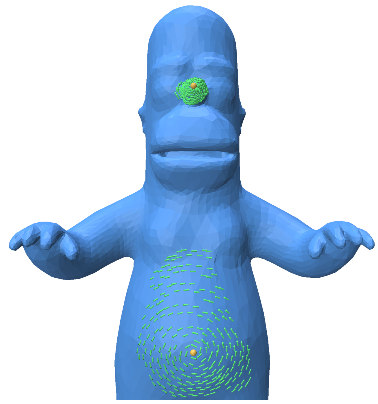
Figure 4a of generated vector fields of the Homer mesh without alignments, Figure 4b the alignments from [1] with a distance of 5 from the singularities and Figure 4c the generated vector fields with these as alignments respectively.

Figure 5 is a summary of the example. As you can see, it created new singularities on the nose and on the belly (but it removed the singularities on the upper arms, more in the drawbacks section).

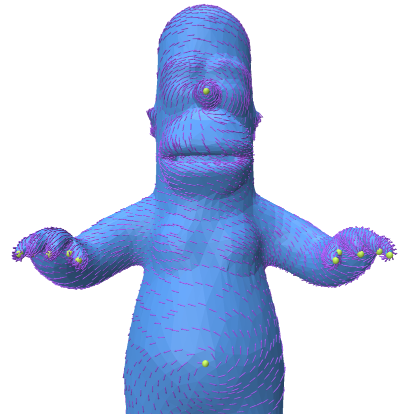
Figure 6 is another example of the Stanford bunny with a distance of 5 nodes from singularities from Trivial Connections. Left is the alignment vector field, right is the generated vector field. It added the singularities along with the already present singularities (the ones on the ears which is slightly visible in the right image).



(a) Homer mesh without any alignments



(b) Homer mesh with 2 singularities, n-rings



(c) Homer mesh with alignment, $\lambda = -100$

Figure 4. Add alignments on homer

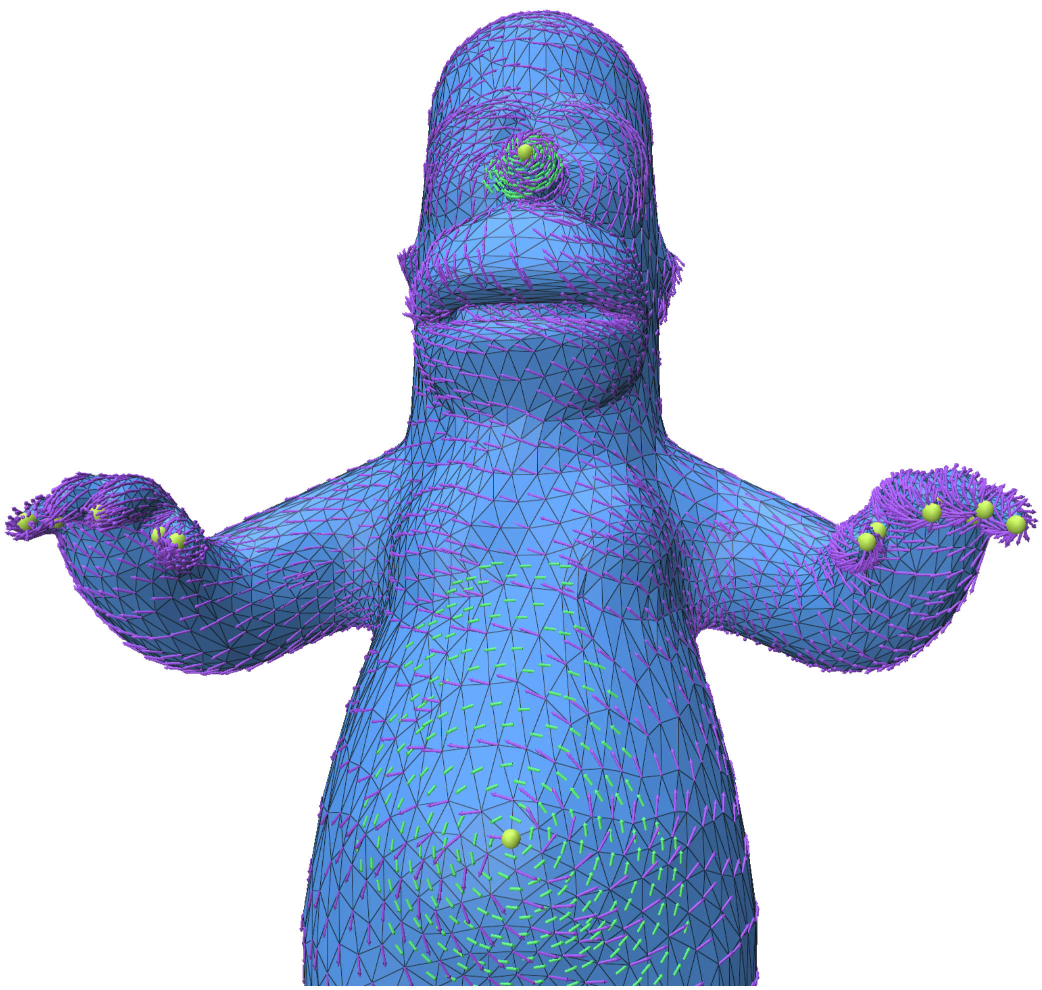
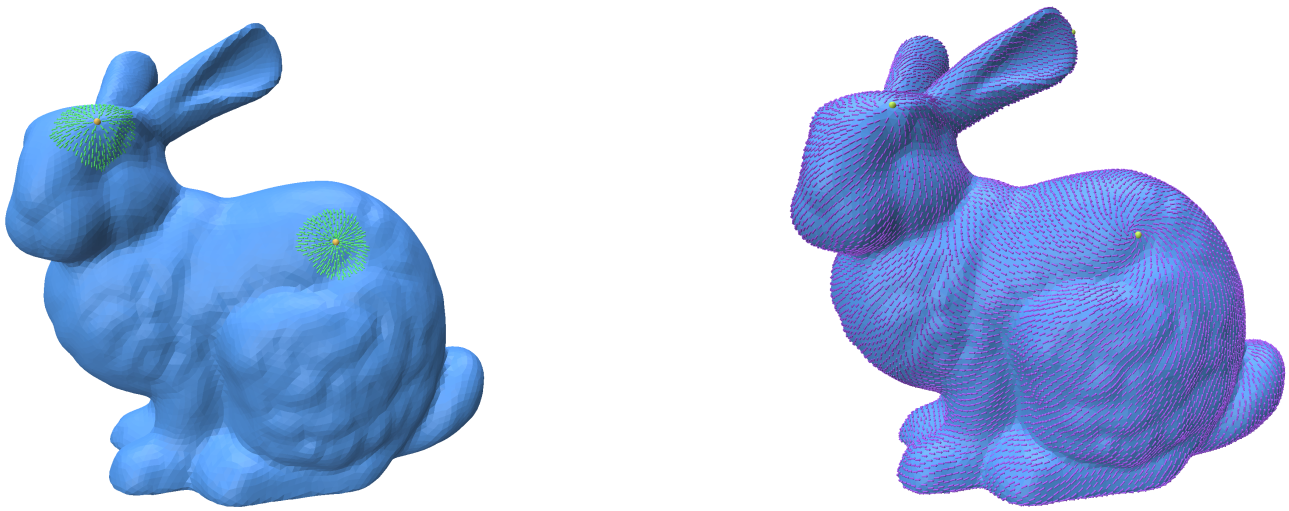


Figure 5. Homer mesh with alignment and aligned fields over-layed



(a) 5-Rings alignment fields on bunny mesh

(b) Bunny aligned. $\lambda = -400$

Figure 6. Example on Bunny mesh

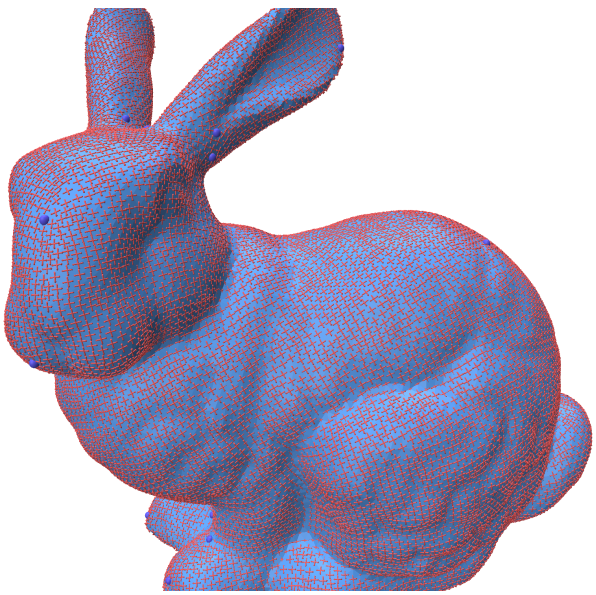


Figure 7. Smooth Cross Fields

3.2. Cross-field Alignment

Below is an example of generated cross fields from stanford bunny mesh where the alignment is provided for the entire mesh. Figure 7 is without the alignments, 8 is the alignments and the 9 is the generated aligned cross fields.

As you can see, it did try to create singularities on the eye and on the side of the mesh (but it created more singularities, more in the drawbacks section).

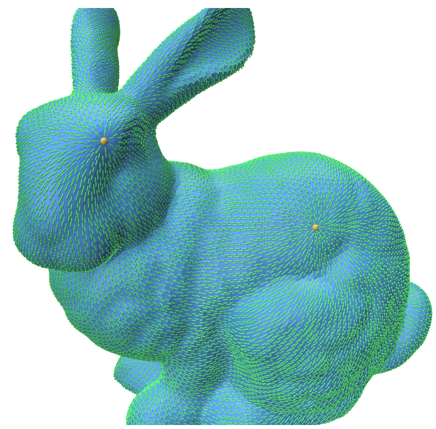


Figure 8. Full alignment fields on bunny mesh

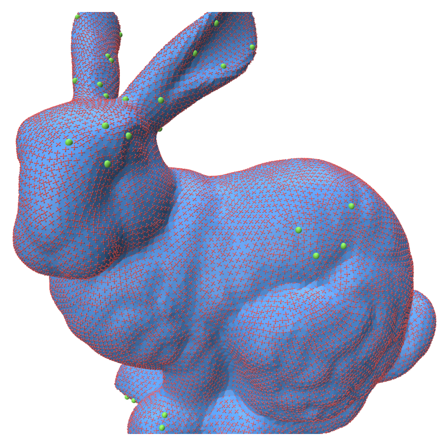
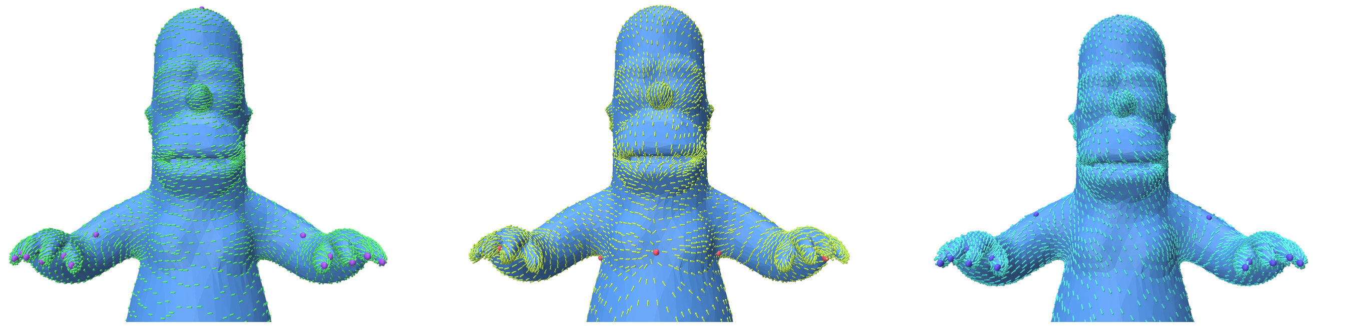


Figure 9. Aligned Cross Fields



(a) s=0

(b) s=1

(c) s=-1

Figure 10. Homer, no alignments, varying smoothness

3.3. Smoothness

Figure 10 is an example of the homer mesh without any alignments, with $s = 0$, $s = 1$ and $s = -1$ respectively.

As you can see it does indicate what the [2] mentions about smoothness and singularities, ie, Quadratic Smoothness Energies. Figure 10a creates more straight lines (on the face) with more number of singularities (on the fingers). In 10b, we can see that the vector fields have become less straight (both on the face and the body) and the number of singularities have gone down (on the fingers). For 10c, the vectors are more straight (on the face and on the body) but the number of singularities didn't go down (on the fingers). The number of singularities didn't increase though, they were exactly the same. We need a better understanding of when exactly the number of singularities go down and when they do not.

4. Drawbacks

Adding the alignment fields from [1] isn't always guaranteeing a singularity. Sometimes the energy of the alignment isn't enough.

Below is an example of where the alignment index we specified was -2 (Figure 11), but it created four singularities with 0.5 index (Figure 12). This could be because the obtained energy parameter t=0.142, might not be enough to enforce the alignment.

Figure 14 is an example where the yellow are the specified singularities and red is the singularity that got formed near the specified singularity.

So to solve this we increase the alignment energy by decreasing the λ. This brings us to the second and more important drawback.

Figure 14 is the same example but with reduced λ. It adds new singularities in places that we didn't even specify. But it brings one of the singularities near to the specified singularity (eye in this example).

The same thing can be observed in the bunny cross field

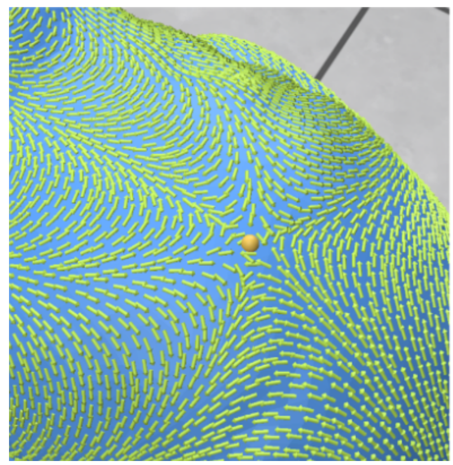


Figure 11. Singularity with index = -2

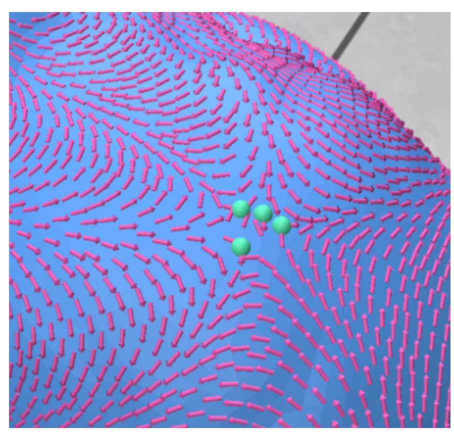


Figure 12. After alignment. s=1, $\lambda = -500$

example from the results section. It added new singularities near the eyes and the side of the bunny. Similarly, for the homer it removed the singularities on the upper arms.

The possible reason for this is that the energy is minimized only if it creates new singularities. If adding singu-

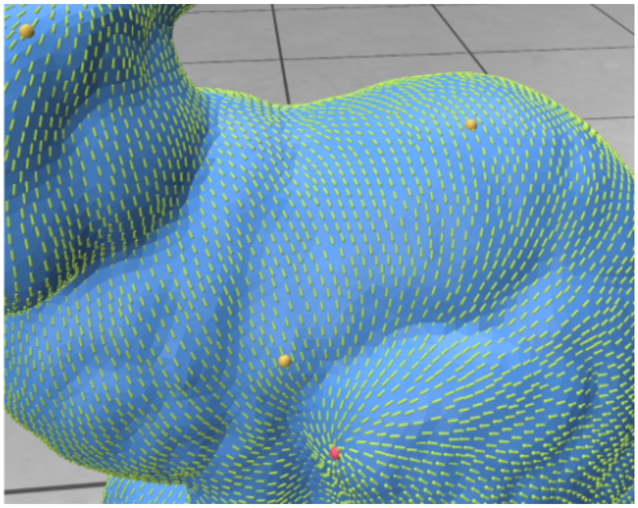


Figure 13. Misaligned Singularities

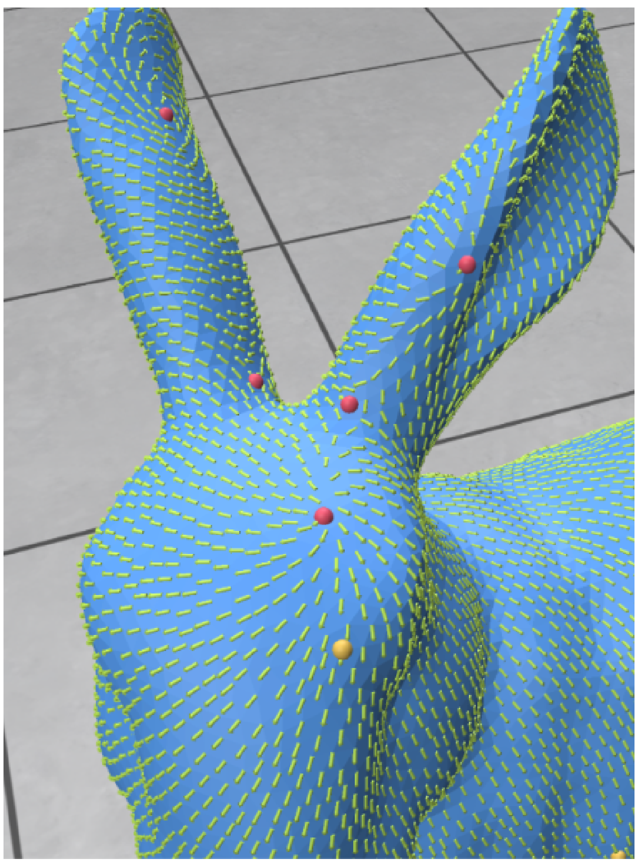


Figure 14. Increased Energy Alignment

larities changes the minimum energy in such a way that new singularities are created, we need to find a totally different approach to add new singularities.

Instead of increasing the alignment energy parameter λ one can specify the alignments all over the mesh. That is what we tested with the cross fields example of bunny mesh in the results section, but it did add more singularities near eyes, ears and on the side of the bunny.

Another possible reason could be that we're adding singularities which is changing the sum of index values of all singularities. In such cases the energies will enforce that the sum of index values of all singularities become equal to Euler Characteristic by adding new singularities which balance out new singularities which we added.

To solve this problem, we could've first get [2] fields on the mesh, then take all the singularities with one of the positions of the singularities changed (but the index shouldn't be changed) and provide it as the input to [1]. Then take the alignment fields near the singularities and provide it as input back to [2].

5. Conclusion

In conclusion, the project presents an extension of the work done in the paper Globally Optimal Direction Fields [2] by adding new singularities through the use of alignment vectors. The alignment vectors are obtained from the Trivial Connections on Discrete Surfaces paper [1], and the implementation allows for any random vectors to be used as alignments, providing a more flexible and customizable approach.

All the visualizations are presented using polyscope [3].

6. Acknowledgments

To Prof. Edward Chien for giving us the opportunity to work on this project.
To Rahul Mitra for his support throughout the semester.

References

[1] Keenan Crane, Mathieu Desbrun, and Peter Schröder. Trivial connections on discrete surfaces. *Computer Graphics Forum (SGP)*, 29(5):1525–1533, 2010. 1, 2, 5, 6

[2] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. Globally optimal direction fields. *ACM Trans. Graph.*, 32(4), 2013. 1, 2, 5, 6

[3] Nicholas Sharp et al. Polyscope, 2019. www.polyscope.run. 1, 6