

**ספריות הטכניון**  
*The Technion Libraries*

**בית הספר ללימודים מוסמכים ע"ש ארוין וג'ואן ג'ייקובס**  
*Irwin and Joan Jacobs Graduate School*

©  
**All rights reserved to the author**

*This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only. Commercial use of this material is completely prohibited.*

©  
**כל הזכויות שמורות למחבר/ת**

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיצו באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

# **Metric transformation through local linear maps: Application to frame field generation**

**Alexey Shevlyakov**



# **Metric transformation through local linear maps: Application to frame field generation**

Research Thesis

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical Engineering

**Alexey Shevlyakov**

Submitted to the Senate  
of the Technion — Israel Institute of Technology Elul  
5776 Haifa September 2016



This research was carried out under the supervision of Prof. Mirela Ben-Chen, in the Faculty of Computer Science.

The generous financial help of the Technion and the Lady Davis foundation is gratefully acknowledged.



# Contents

## List of Figures

<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Related work . . . . .	4
<b>2 Mathematical preliminaries</b>	<b>7</b>
2.1 Riemannian metric . . . . .	7
2.2 Connection . . . . .	7
2.2.1 Levi-Civita connection . . . . .	8
2.2.2 Geodesics . . . . .	8
2.3 Frame fields and N-RoSy fields . . . . .	8
<b>3 Metric transformation</b>	<b>11</b>
3.1 Parallel transport map . . . . .	11
3.2 Metric transformation . . . . .	11
3.3 Frame driven deformation . . . . .	13
3.4 Computing geometric quantities . . . . .	15
3.5 Deformation Energy . . . . .	16
3.5.1 Constructing Energy Matrix . . . . .	16
3.5.2 Compatibility constraints . . . . .	17
<b>4 Application</b>	<b>19</b>
4.1 Curvature shape . . . . .	19
4.1.1 Metric distortion . . . . .	24
4.1.2 Optimisation . . . . .	28
4.2 Computing smooth N-RoSy fields . . . . .	29
4.2.1 Change of metric and smooth vector fields . . . . .	30
4.2.2 Results . . . . .	31
4.3 Conformal parametrisation . . . . .	33
<b>5 Conclusion</b>	<b>37</b>

**Hebrew Abstract**

# List of Figures

1	left – 4-RoSy vector, right – general frame . . . . .	9
2	Linear transformation $\mathbf{W}$ acting on a face . . . . .	14
3	Incompatible linear transformations . . . . .	15
4	Examples of curvature shapes . . . . .	20
5	Quality of the cross-field . . . . .	22
6	Quality of the cross-field. Histogram . . . . .	23
7	Metric distortion . . . . .	27
8	Metric distortion. No constraints. Histogram . . . . .	28
9	Metric distortion. Constraints . . . . .	29
10	$u^{-1} \circ h^{-1}$ . . . . .	31
11	Aligned vector field . . . . .	32
12	Smooth cross field with constraints . . . . .	33
13	Distribution of Gaussian curvature . . . . .	34
14	An approximate embedding of the flat torus in $\mathbb{R}^3$ found with [BCGB08]	35
15	Gaussian curvature . . . . .	35
16	Metric distortion [PPTSH14] . . . . .	36
17	Conformal distortion . . . . .	36



# Abstract

Generic frame fields are important for many applications in computer graphics such as texture mapping and quadrangulation, however, unlike their simpler version – cross fields, they are hard to design. One category of approaches relies on creating an intermediate metric in which the frame field becomes a cross field. The vast majority of existing approaches achieve it by deforming the original surface in the Euclidean space. It's a known fact that not all Riemannian metrics can be accommodated by the three-dimensional Euclidean space, thus some authors seek solutions in higher dimensional Euclidean spaces, thus increasing the complexity of the problem. Intrinsic approaches, designing a suitable Riemannian metric directly have been introduced recently, however, they rely on a complicated formulation and numerical approximations. Moreover, they require developing new tools for quadrangulating the surface in this new metric since the surface is curved and not a polytope. We propose an intrinsic approach that obviates the necessity of an embedding and has a simple formulation. We compute a new Riemannian metric that results from warping a surface by local linear transformations and impose a compatibility constraint to ensure that the resulting space is a metric space and a polytope. This allows for a larger space of solutions than is possible to accommodate in the three- dimensional Euclidean space as well as a simple formulation. We show how to adapt existing methods for designing smooth vector fields for this framework and demonstrate the robustness of the algorithm relatively to extrinsic approaches.



# Chapter 1

## Introduction

Frame field design is an important topic in computer graphics. Frame fields have numerous applications, among them quadrangulation ([BLP<sup>+</sup>13]) and texture synthesis ([LH06]). However, designing generic frame fields is a challenging task. Many approaches rely on deformation of the original metric where the original frame field defined by features or user input becomes a cross-field thus reducing the problem to a simpler and more researched one of designing a smooth cross-field and then applying known tools for quadrangulation and mapping it back to the original shape. Such a metric is usually found by deforming a surface in the Euclidean space. It's a known fact that not all Riemannian metrics can be accommodated by the three dimensional Euclidean space ([Whi57]) thus the three dimensional space may be too restrictive. Some authors seek a solution using higher dimensional Euclidean spaces, but it introduces additional complexity. Intrinsic approaches, introduced recently design the Riemannian metric directly, however the Riemannian metric achieved by such methods does not yield a polyhedron. Therefore such approaches require developing new tools to work with this kind of objects. Finally, they have complicated formulations and rely on numerical approximations.

We propose a new intrinsic method of designing a Riemannian metric. We compute a new Riemannian metric that results from applying a linear transformation to every face and enforce a compatibility constraint to preserve the topology of the original manifold. Our metric change results in another polyhedral surface which allows us to utilise the well known Euclidean connection induced by the embedding of adjacent faces into the 2-dimensional Euclidean space. We apply our approach to a subset of the general problem of designing frame fields – we design orthogonal frame fields. However, this is a very popular application since it's often required to design frame fields conforming with principal directions which are an example of orthogonal frame fields. However it does not mean that our approach is limited to orthogonal frame fields. We show that our approach outperforms the state of the art approaches in certain settings and show that it succeeds at achieving a metric that is impossible to achieve by extrinsic approaches. Our approach also has the advantage of simplicity – since the intermediate shape is a

polytope it's possible to apply all existing methods of computer graphics that don't rely on extrinsic information.

## 1.1 Related work

Approaches to frame field design can be broadly divided into extrinsic and intrinsic the former using an embedding in an ambient space and the latter dealing only with the metric. It's a well known fact that not all 2-dimensional manifolds admit embedding in a 3-dimensional Euclidean space, thus several authors proposed using embeddings in higher dimensional Euclidean spaces. [KMZ10] proposed to utilise the 6-dimensional Euclidean space to design a metric driving the parametrisation. [PPTSH14] proposed to design a cross field on an intermediate surface constructed in such a way that a frame field on the original surface is taken to a cross field under the transformation relating the two shapes. The authors solve a quadratic optimisation problem over  $\mathbb{R}^3$ . This approach has a simple formulation, however, there are several limitations. There is no guarantee that the resulting surface will preserve its topology, it's also possible that a surface will have self-intersections. Another problem is that one Riemannian manifold defined by the same metric can have different embeddings. Finally, not all metrics are possible to embed in  $\mathbb{R}^3$  which limits the space of solutions to the minimisation problem. In [JFH<sup>+</sup>15] it's proposed to design a Riemannian metric  $g$  defined on each face in which the target frame field becomes a cross field and satisfies user constraints. This approach is intrinsic as ours, however there are important differences. Since the intermediate step for designing a smooth frame field on the original surface involves designing a smooth cross field (or, more generally, a N-Rosy field) on the intermediate surface it's necessary to approximate the Levi-Civita connection. The authors express the rotation angle representing the connection from Cartan structure equations and then estimate the necessary quantities at mid-edges using the known metric values associated to faces. However, they do not construct any "real" object in the sense that a polytope is. That is, they don't create a metric space, rather a sampling of a metric space with metric values only defined on faces. Thus, should there be a need to apply other methods of computer graphics further approximations would be needed. For instance, it's not clear how to construct geodesics on such an object (geodesics can be constructed as the straightest lines but it's impossible to construct them for every point of the surface) since it's not sufficient to define a metric per each face unless the surface is a polytope, which it's generally not, according to the authors. The authors of [JFH<sup>+</sup>15] also have to define their own Quadrangulation technique since most existing techniques are designed to work with simplicial manifolds. In contrast, we construct an object that is a polytope and therefore it's possible to apply any existing intrinsic methods of computer graphics to this object such as [BZK09], which is valuable since polytopes are the most studied and well behaved discrete objects and are ubiquitous in computer graphics. At the same time we shed the necessity to carry the extrinsic information.





## Chapter 2

# Mathematical preliminaries

In this chapter we will briefly define the notions we use for the purpose of our work for self-containedness. This introduction is not rigorous and omits many important nuances which are not relevant to our work.

### 2.1 Riemannian metric

A Riemannian metric tensor is a 2-tensor field  $g$  that is symmetric and positive definite (i.e.  $g(X, Y) = g(Y, X)$  and  $g(X, X) > 0$  for  $X \neq 0$ ). The Riemannian metric generalises the inner product in the Euclidean space. In other words it's a function taking two vectors  $\mathbf{v}, \mathbf{w} \in T_p M$  and assigning a scalar value  $g(\mathbf{v}, \mathbf{w})$  at every point of a manifold. A smooth manifold equipped with a Riemannian metric is called a Riemannian manifold.

### 2.2 Connection

We will only consider linear connections and will just call them connections. A connection on a manifold ([Spi70]) is a map

$$\nabla : TM \times TM \rightarrow TM$$

with the following properties

1.  $\nabla_X Y$  is linear over  $C^\infty(M)$ :

$$\nabla_{fX_1+gX_2} Y = f\nabla_{X_1} Y + g\nabla_{X_2} Y$$

for  $f, g \in C^\infty(M)$ ;

2.  $\nabla_X Y$  is linear over  $R$  in  $Y$ :

$$(aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2$$

for  $a, b \in \mathbb{R}$ ;

3.  $\nabla$  satisfies the following product rule:

$$\nabla_X(fY) = f\nabla_X Y + (Xf)Y$$

for  $f \in C^\infty(M)$ .

### 2.2.1 Levi-Civita connection

The Levi-Civita connection is a unique connection with the following properties

1. Preserves the metric:  $\nabla g = 0$
2. It's torsion free:  $\nabla_X Y - \nabla_Y X = [X, Y]$ , where  $[X, Y]$  is the Lie bracket of the vector fields  $X$  and  $Y$ .

We will not discuss the torsion of a connection, instead we will show that our method in the end utilises the standard Euclidean connection which is also a Levi-Civita connection.

### 2.2.2 Geodesics

Geodesics can be defined as locally shortest paths or straightest paths between two points on a manifold. However for simplicial manifolds such as ones used in computer graphics these two notions differ. It's not possible to define a shortest geodesic passing through a spherical vertex ([PS06]). We will avoid this conundrum by ignoring geodesics passing through vertices – we don't need them for our purposes. It will leave us with a familiar notion of geodesic in the Euclidean space – straight line. Indeed, if a curve does not pass through a vertex then it passes through a subset of a polyhedron isometric to the Euclidean space through a chain of hinge maps – isometric flattenings of adjacent triangles.

## 2.3 Frame fields and N-RoSy fields

A N-RoSy field stands for "rotationally symmetric field" and is a collection of vectors rotated by a multiple of  $2\pi/N$ . Following [PPTSH14] we define a frame field  $F_p$  at a point  $p$  as an ordered set of four vectors  $\{\mathbf{v}, \mathbf{w}, -\mathbf{v}, -\mathbf{w}\} \in T_p M$ . A frame field  $F$  is a collection of frames  $f_p$ . A cross-field is a frame field with all the vectors constituting the frame of equal norm. A cross-field is a special case of N-RoSy field with  $N = 4$ .

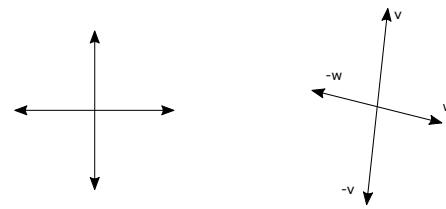


Figure 1: left – 4-RoSy vector, right – general frame



## Chapter 3

# Metric transformation

As introduced in [PPTSH14], we attempt to find a frame-driven transformation of a surface that takes a certain frame field to a cross field. The core idea of the work is to embed each face of a surface into its own 2-dimensional Euclidean space with a conveniently chosen basis and apply a linear transformation in terms of the picked bases to drive the metric change prescribed by the original frame field.

### 3.1 Parallel transport map

Since we consider only simplicial complexes, we can embed any 2-dimensional simplex in  $\mathbb{R}^2$  and endow it with a basis of our choice  $\mathbf{b}$ . Any two adjacent faces can be isometrically embedded in the Euclidean plane via what's colloquially known as the "hinge map" that can be obtained by unfolding two adjacent triangles. This allows us to define a connection as the ordinary Euclidean connection. This connection was formalised in [KCPS13] as the Levi-Civita connection. Consider  $T_p M$  and  $T_q N$  tangent spaces at points  $p$  and  $q$  on a manifold  $M$ . Fix a basis  $X_p$  and  $X_q$  in each face, then the connection will be a rotation aligning these bases. This connection can be represented as a rotation matrix  $\mathbf{R}$ . Suppose we have a vector  $\mathbf{u} \in T_p M$  expressed in the basis  $X_p$  and we want to compare it to a vector  $\mathbf{v} \in T_q M$ . The parallel transport map  $P : T_p M \rightarrow T_q M$  is given by

$$P := \mathbf{R}\mathbf{u} - \mathbf{v} \quad (3.1)$$

### 3.2 Metric transformation

We will consider a metric transformation  $h$  acting on a piece-wise linear manifold  $M$  which is represented by a simplicial complex. We assume that our simplicial complex consists of triangular simplices and is a manifold, that is a neighbourhood of every point is homeomorphic to a topological disk. We denote the Riemannian metric  $g$  and  $\bar{g}$  in the case when it's equivalent to the Euclidean metric, we will use this notation whenever we speak of the Riemannian metric in the continuous case or when we mean the totality

of matrices we assign to each face of a surface. We will use  $\mathbf{G}$  for the Riemannian metric ( $\overline{\mathbf{G}}$  for Euclidean) whenever we speak about a local assignment of a matrix representing the total metric  $g$  (or  $\bar{g}$ ) at a certain point. As introduced before, the Levi Civita connection depends on the metric and therefore changes of the metric will alter the Levi-Civita connection. Recall, that the Levi-Civita connection is in fact the ordinary Euclidean connection realised through embedding of faces in  $\mathbb{R}^2$ . As discussed above, a metric is a 2-tensor  $g$  assigned to every face of a manifold  $M$ . Practically it's a symmetric positive-definite matrix. Starting with the initial metric  $\bar{g}$  we obtain a new metric  $g$  by applying a piece-wise linear homeomorphism  $h : TM \rightarrow TM$ .

Consider the ordinary euclidean metric, instantiated by a matrix  $\overline{\mathbf{G}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  in each face. A bilinear form  $\mathbf{v}\overline{\mathbf{G}}\mathbf{u}^\top = \mathbf{v}\mathbf{u}^\top$  parallel transported across an edge  $l_{ij}$  adjoining faces  $i$  and  $j$  will be given as

$$\nabla_{ij}\bar{g}(\mathbf{v}, \mathbf{u}) = \mathbf{v}\mathbf{u}^\top - \mathbf{v}\mathbf{R}^\top\mathbf{R}\mathbf{u}^\top = \mathbf{v}(\mathbf{I} - \mathbf{R}^\top\mathbf{R})\mathbf{u}^\top = \mathbf{v}(\mathbf{I} - \mathbf{I})\mathbf{u}^\top = 0, \quad (3.2)$$

where  $\mathbf{R}$  is a rotation matrix.  $\nabla_{ij}\bar{g}$  is a covariant derivative of the metric  $\bar{g}$ . A homeomorphism  $h(M)$  can be realised as a collection of *compatible* local homeomorphisms  $h_i$  expressed in local coordinates of each face as a matrix  $\mathbf{W}_i$  which acts on each tangent vector space as a linear transformation. The compatibility condition expresses the fact that an edge should be equal to itself when represented in different bases so that the resulting space remains a metric space. Assuming  $\mathbf{W}_i$  and  $\mathbf{W}_j$  in adjacent faces are not equal their common edge expressed in different bases will generally differ:  $\mathbf{W}_i l_{ij}^{\mathbf{b}_i} \neq \mathbf{W}_j l_{ij}^{\mathbf{b}_j}$ . This violates the property that a metric should assign a unique value to a path between two points. The covariant derivative of the new metric  $g = h(\bar{g})$  will take the form

$$\nabla_{ij}g(\mathbf{v}, \mathbf{u}) = \mathbf{v}\mathbf{W}_i^\top\mathbf{W}_i\mathbf{u}^\top - \mathbf{v}\mathbf{W}_j^\top\mathbf{R}^\top\mathbf{R}\mathbf{W}_j\mathbf{u}^\top = \mathbf{v}(\mathbf{W}_i^\top\mathbf{W}_i - \mathbf{W}_j^\top\mathbf{W}_j)\mathbf{u}^\top. \quad (3.3)$$

Since generally  $\mathbf{W}_i$  does not equal  $\mathbf{W}_j$  this connection does not preserve the inner product (Subsection 2.2.1) and therefore is not compatible with the metric. Not only that, the parallel transport map also changes. But we can observe that the new manifold  $N = h(M)$  is another polyhedron homeomorphic to  $M$ . Observe that  $\mathbf{W}_i^\top\mathbf{v} = h_i(\mathbf{v}) = \mathbf{v}_N \subset TN = h(TM)$ . Thus we can define a new Levi-Civita connection and a new parallel transport map in this new metric the same way we defined it previously. For vectors  $\mathbf{u}_N = \mathbf{W}^\top\mathbf{u}$  and  $\mathbf{v}_N = \mathbf{W}^\top\mathbf{v}$  the parallel transport map will be defined as

$$P := \mathbf{R}\mathbf{u}_N^\top - \mathbf{v}_N$$

and the respective connection will become

$$\nabla_{ij}g(\mathbf{v}, \mathbf{u}) = \mathbf{v}_N\mathbf{u}_N^\top - \mathbf{v}_N\mathbf{R}^\top\mathbf{R}\mathbf{u}_N^\top = \mathbf{v}_N(\mathbf{I} - \mathbf{R}^\top\mathbf{R})\mathbf{u}_N^\top = \mathbf{v}_N(\mathbf{I} - \mathbf{I})\mathbf{u}_N^\top = 0,$$

Thus we can have the same Euclidean connection in the new manifold, which will be defined by the same rotation matrix. In terms of applications to designing smooth vector fields this will mean that a vector field that is considered smooth on a manifold  $N$  will not necessarily be smooth when pulled back to  $M$ , however, the per-face alignment with features will be preserved since the vectors will be pulled back by the same homeomorphism  $h_i$ .

### 3.3 Frame driven deformation

The first application of our method is to implement a frame-driven deformation as in [PPTSH14] but without using an embedding in the ambient  $\mathbb{R}^3$ . The advantage of this approach is that it allows to use a larger space of metrics that might be impossible to accommodate by  $\mathbb{R}^3$ . The Nash embedding theorem establishes that any smooth Riemannian manifold  $M$  of dimension  $m$  can be isometrically embedded in the Euclidian space of dimension  $n \geq m + 1$ . Any Riemannian manifold of class  $C^1$  can be embedded into  $\mathbb{R}^3$  as shown in [Nas56]. However, this embedding is fraught with difficulties – it requires an infinite number of elements. Perhaps it's possible to embed  $C^0$  surfaces in  $\mathbb{R}^3$  but as we show later, practical implementations of embedding algorithms cannot provide a solution to this problem. We will reformulate the problem set in [PPTSH14] to suit our framework and apply it to the case of an orthogonal but anisotropic frame field represented by principal curvature directions. Consider a simplicial manifold  $(M, g_M)$ , where  $g_M$  is the standard Euclidean metric represent by the identity matrix on every face. Let  $F \subset TM$  be a frame field on  $M$ . We want to find a new metric  $g_N$  on a simplicial manifold  $N$  such that  $F_N = h(F_M)$  is a crossfield. We will embed each face into a Euclidean plane:  $T_i \hookrightarrow \mathbb{R}^2$  and endow each such Euclidean plane with a basis  $\mathbf{b} = \{\mathbf{e}_1, \mathbf{e}_2\}$ , such that the basis  $\mathbf{b}$  is aligned with principal curvature directions. This allows us to formulate the problem of finding a metric  $g$  in the following way. For each face of  $N$  we want to find a matrix  $\mathbf{G} = \begin{pmatrix} \frac{1}{l_1^2} & 0 \\ 0 & \frac{1}{l_2^2} \end{pmatrix}$ , where  $l_1^2$  and  $l_2^2$  are the norms of the vectors in the frame field  $F$ . The resulting shape is called *curvature shape* – a shape obtained by a change of metric that results in morphing a frame field into a cross field. A quadratic form  $\mathbf{v}\mathbf{G}\mathbf{v}^\top$  can be expressed as

$$\mathbf{u}\mathbf{u}^\top = (\mathbf{W}\mathbf{v}^\top)^\top(\mathbf{W}\mathbf{v}) = \mathbf{v}\mathbf{W}^\top\mathbf{W}\mathbf{v}^\top = \mathbf{v}\mathbf{G}\mathbf{v}^\top$$

Thus we can view metric transformation as a piece-wise linear transformation

$$h : TM \rightarrow TM$$

$$\mathbf{v} \longmapsto \mathbf{W}\mathbf{v}^\top$$

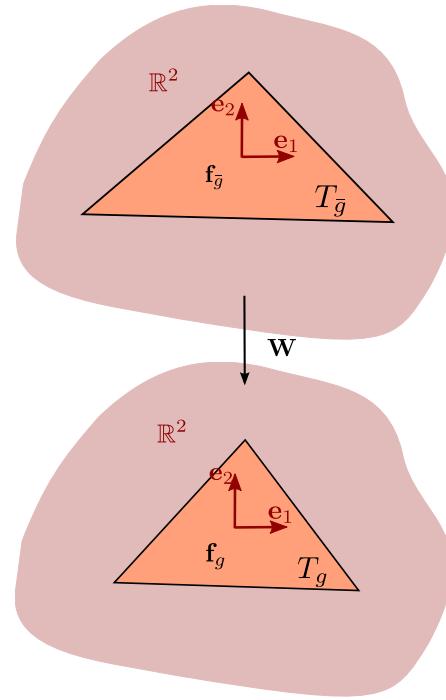


Figure 2: Linear transformation  $\mathbf{W}$  acting on a face

We, however, must pay attention to preserve the topology, since it might so happen that two incompatible transformation  $\mathbf{W}_i$  and  $\mathbf{W}_j$  in adjacent faces produce a tear, as discussed in Section 3.2. In order to ensure it we, using the previously established rules of parallel transport, will impose a strict constraint equating the norm of each edge represented in bases of adjacent faces

$$\mathbf{l}_{ij}(\mathbf{W}_i^\top \mathbf{W}_i - \mathbf{W}_j^\top \mathbf{W}_j)\mathbf{l}_{ij}^\top = 0$$

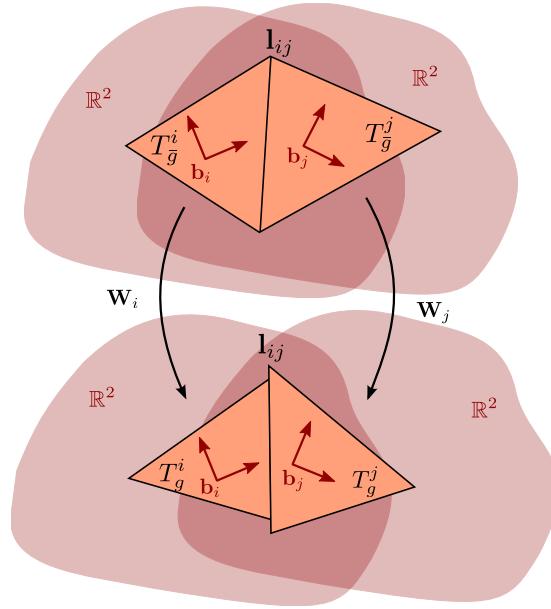


Figure 3: Linear transformations are incompatible. The same edge has a different metric.

### 3.4 Computing geometric quantities

The majority of geometric methods are developed for meshes in the Euclidean space, however, many of them do not in fact need to rely on an ambient space since they are intrinsic, thus computing any intrinsic geometric quantities on a mesh affected by a metric transformation is exactly the same as computing them on any other mesh, this allows us to freely use other tools developed for computing geometric quantities on meshed embedded in the Euclidean space. We will give expressions of geometric quantities computed on a transformed shape to show how this metric change affects them.

**Gaussian curvature.** Gaussian curvature is defined on a polyhedron by the following formula

$$K_G = 2\pi - \sum_j \alpha_j$$

where  $\alpha_j$  are angles formed by edges connected the vertex. Under the effect of  $h$  it transforms into

$$K_G^h = 2\pi - \sum_j h_j(\alpha_j)$$

In a particular basis it can be expressed as

$$K_G^h = 2\pi - \sum_j \arccos \left( \frac{\langle \mathbf{W}_j^\top \mathbf{e}_j^1, \mathbf{W}_j^\top \mathbf{e}_j^2 \rangle}{\|\mathbf{W}_j^\top \mathbf{e}_j^1\| \cdot \|\mathbf{W}_j^\top \mathbf{e}_j^2\|} \right)$$

**Cotangent weights Laplacian** This is a widely used discretisation suggested in [MDSB01] given by the formula

$$\frac{1}{2} \sum (\cot \alpha_{ij} + \cot \beta_{ij})$$

where  $\alpha$  and  $\beta$  are angles opposite the edge  $ij$ . We have shown above how the angles change under a linear transformation  $h$ . Generally we can extend every notion and every method that does not require extrinsic information to our setting.

### 3.5 Deformation Energy

The target transformation  $\mathbf{W}^* = \begin{pmatrix} \frac{1}{l_1^2} & 0 \\ 0 & \frac{1}{l_2^2} \end{pmatrix}$  that morphs the original surface into curvature shape (recall that we seek for a new metric and thus are not interested isometries) is generally impossible to achieve while still preserving the topology and ensuring the surface remains a metric space. Thus we will need to restrict the space of solutions by applying a constraint ensuring the local transformations  $W_i$  are compatible in the sense discussed previously.

$$\begin{cases} E(\mathbf{G}) = \sum_{j=1}^n (\|\mathbf{G}_j - \mathbf{G}_j^*\|), \\ \text{s.t. } \mathbf{l}_{ij}(\mathbf{G}_i - \mathbf{G}_j)\mathbf{l}_{ij}^\top = 0, \end{cases} \quad (3.4)$$

where  $\mathbf{G}^*$  is the ideal solution and  $\|\cdot\|$  – Frobenius norm,  $n$  is the number of faces. We got some of our results using only these constraints, however generally we might need to restrict the metric to being positive-definite (we don't always impose this constraint because the opstimisation problem results in a positive definite metric without constraints) or the identity in certain regions. For this task we can use several approaches which will be discussed in greater detail later when we talk about the results of our optimisation. These approaches include, supplying an initial solution for the optimisation algorithm, adding weights to the deformation energy, adding additional linear constraints on the metric or applying lower and upper bounds to the metric.

#### 3.5.1 Constructing Energy Matrix

We use the MATLAB built-in function Quadprog to solve problem (3.4). The Quadprog function is designed to solve problems of the form  $E = \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{f}^\top \mathbf{x}$ . We can formulate problem (3.4) as following

$$E(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^\top (\mathbf{x} - \mathbf{x}^*) = \mathbf{x}^\top \mathbf{x} - 2\mathbf{x}^\top \mathbf{x}^* + \mathbf{x}^{*\top} \mathbf{x}^*,$$

where  $\mathbf{x}$  is a  $2N_{faces}$  by 1 vector the first half of which is composed of  $\mathbf{G}_i(1, 1)$  and the second half of  $\mathbf{G}_i(2, 2)$  and  $\mathbf{x}^*$  is the vector of ideal values. We drop the constant term, hence further when we discuss the energy of the cross-field the energy is negative.

### 3.5.2 Compatibility constraints

The compatibility constraint is of the form

$$\mathbf{A}\mathbf{x}^\top = \mathbf{0},$$

where  $\mathbf{A}$  is a  $N_{edges}$  by  $2N_{faces}$  matrix and  $\mathbf{0}$  is the zero vector. To see how to construct the matrix  $\mathbf{A}$  consider two adjacent triangles  $T_i$  and  $T_j$  sharing a common edge. Let  $T_i$  in the most general case have a metric  $\mathbf{G}_i = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$  and  $T_j$  have a metric  $\mathbf{G}_j = \begin{pmatrix} d & e \\ e & f \end{pmatrix}$ .

The norm of the common edge in  $T_i$  will be given as  $\mathbf{y}^\top \mathbf{G}_i \mathbf{y}$ , where  $\mathbf{y} = \{y_1, y_2\}$  are coordinates of the common edge in the basis of  $T_i$ . In  $T_j$  the norm of the common edge will be expressed as  $\mathbf{z}^\top \mathbf{G}_j \mathbf{z}$ . These norms have to be equal, hence we can write

$$\mathbf{y}^\top \mathbf{G}_i \mathbf{y} - \mathbf{z}^\top \mathbf{G}_j \mathbf{z} = 0$$

Substituting the values for  $\mathbf{G}_i$  and  $\mathbf{G}_j$  and expanding the above expression we get

$$ay_1^2 + 2by_1y_2 + cy_2^2 - (dz_1^2 + 2ez_1z_2 + fz_2^2) = 0 \quad (3.5)$$

It's a linear combination of the unknown values of the metric  $a, b, c, d, e, f$ . Hence we can find a certain matrix  $\mathbf{A}_{ij}$  of size  $1 \times 6$ , such that (3.5) will be rewritten as  $\mathbf{A}_{ij}\mathbf{x}^\top = 0$  where  $\mathbf{x} = \{a, b, c, d, e, f\}$ . The total number of constraints will be equal to the number of edges. The total size of the matrix  $\mathbf{A}$  will be  $N_{edges} \times 3N_{faces}$ . In this particular work we are only concerned with orthogonal frame fields and hence  $b = e = 0$ . Equation (3.5) takes the form

$$ay_1^2 + cy_2^2 - (dz_1^2 + fz_2^2) = 0$$

and the total size of the matrix  $\mathbf{A}$  becomes  $N_{edges} \times 2N_{faces}$ .



# Chapter 4

# Application

## 4.1 Curvature shape

First, we will show how our method compares to [PPTSH14]. Since our method does not employ an Euclidean embedding we cannot visualise the results in the Euclidean space, instead we will use several metrics to assess the quality of the solution. It's natural to test how the immediate output of both algorithms compare. Recall that an ideal transformation takes a frame-field and morphs it into a cross-field, therefore we will use the measure of anisotropy of the resulting "cross-field" as a means to assess the quality of the transformation. We will set up the algorithms to conform with the original frame field on all faces of a mesh – this is an optimisation problem without constraints except the compatibility constraint. This frame field is not necessarily smooth everywhere (there are singularities on surfaces whose genus is not zero) nor everywhere anisotropic, however this provides us with a means to asses the flexibility of the algorithms.

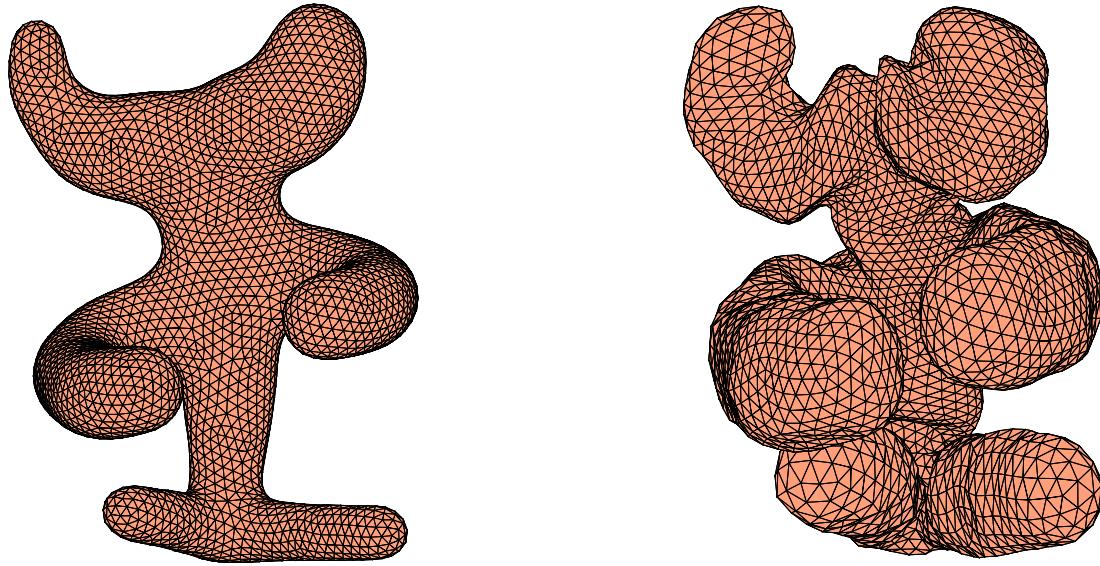
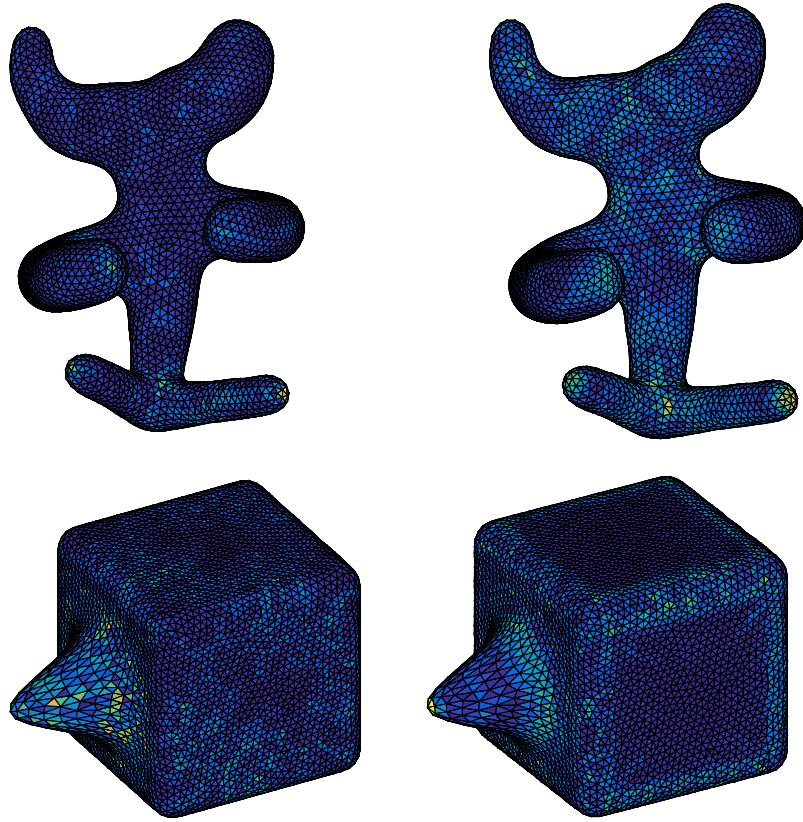
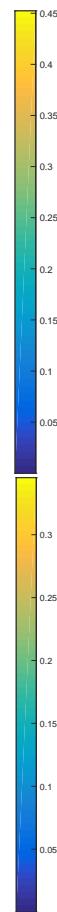


Figure 4: Examples of a surface (left) and its corresponding curvature shape (right)

Consider a frame consisting of vectors  $\{\mathbf{v}, \mathbf{w}, -\mathbf{v}, -\mathbf{w}\}$ . Then the following norm will measure the quality of the frame defined on each face

$$q = \frac{\|\mathbf{v}\| - \|\mathbf{w}\|}{\|\mathbf{v}\|}$$

We will plot this measure on the original surfaces for clarity, since embeddings of curvature shape can have very large variations of size of elements as well as self-intersections which would obscure the results. Also, our algorithm does not have a reference to the Euclidean space by design and hence such curvature shape cannot be visualised in a 3-dimensional Euclidean space.



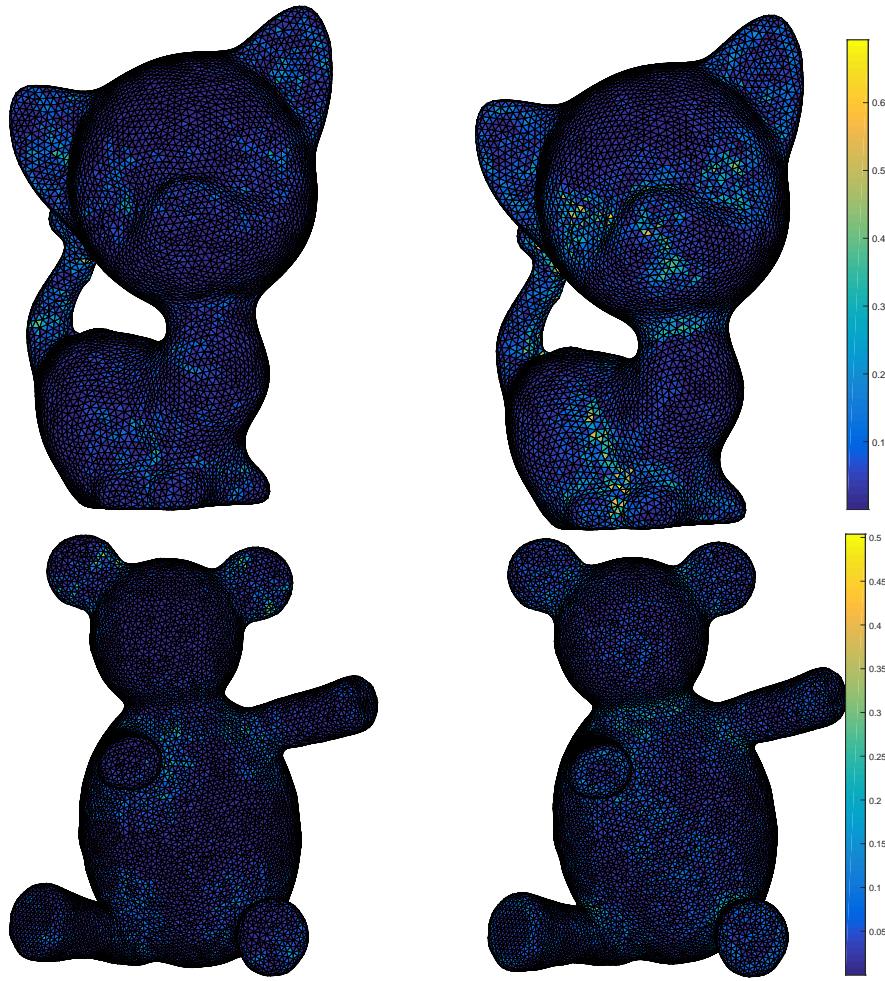


Figure 5: Quality of the cross-field. Left – ours, right – [PPTSH14].

In Figure 5 we visualised the quality measure by plotting it on the original shape. We can see that our method produces a vector field that is characterised by being more cross-like than that in [PPTSH14]. This can be explained by the fact that not all metrics can be realised in the Euclidean space and therefore the Euclidean space imposes additional implicit constraints. Another interesting feature to note is that on the cube the irregularities of the cross field seem to group around extrinsic features such as sharp creases which are indistinguishable from the metric point of view. This effect is noted in [JTPSH15] and utilised for designing quad meshes aligned with edges. Therefore, sometimes extrinsic information can play a useful role in frame field design. If we construct histograms (Figure 4.1), where along the horizontal axis we plot the values of the quality of a cross field discussed above and along the vertical axis the proportion of crosses falling in respective intervals, we can see that for most cases our methods gives more crosses with a low distortion, while the method proposed in [PPTSH14] shows more distortion.

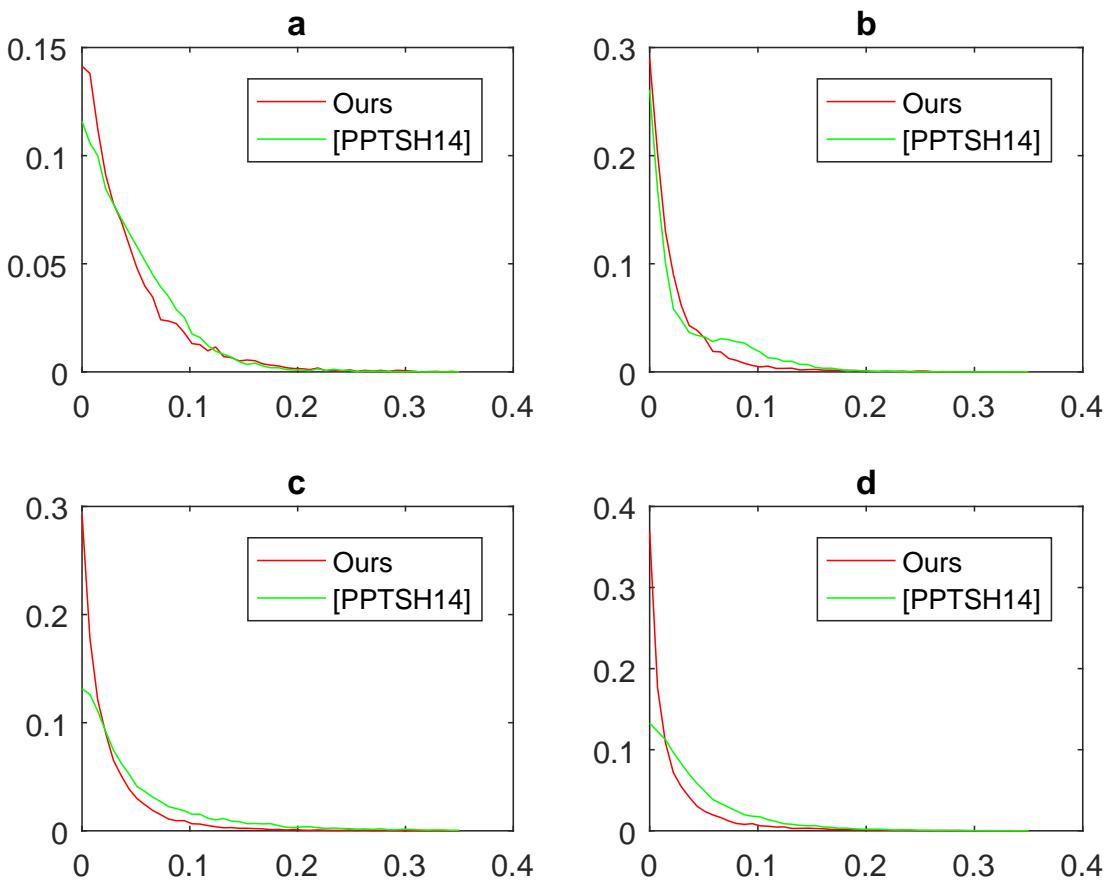


Figure 6: Quality of the cross-field. The proportion of crosses falling into each quality interval is depicted along the vertical axis, quality of a cross (lower is better) – along the horizontal axis. a) moomoo, b) cube, c) cat, d) teddy.

Finally, we attempted to "mimic" the results in [PPTSH14] with our method. That is, given an original surface  $M$  and a curvature surface  $N_{ext}$  achieved by the extrinsic algorithm we find a transformation  $h$  realised in terms of our approach, such that  $h(M) = N_{ext}$ . We can successfully solve this problem with the error of around  $10^{-6}$  relative to the average edge length, which can be considered negligible. We then measure the energy of the original function we intended to minimise substituting both our intrinsic solution and the solution for the extrinsic method. Below we give a table of the values for the extrinsic and intrinsic energy levels for the intrinsic and extrinsic solutions. Our energy levels are consistently lower than those for the extrinsic solution.

Table 4.1: Energy values for the extrinsic and intrinsic approaches achieved by (3.4)

	moomoo	cube	cat	teddy
# of vertices	8334	5256	9447	24996
Ours	-0.58	-3.04	-2.89	-3.62
[PPTSH14]	-0.56	-3.01	-2.75	-3.57

#### 4.1.1 Metric distortion

To bring all three methods discussed in this thesis to one common denominator we will devise another measure to compare them. Recall that in our formulation we are looking for a matrix  $\mathbf{G} = \mathbf{W}\mathbf{W}^\top$  defined on each face representing a Riemannian metric  $g$ . Assuming we have an initial frame field denoted by  $F = \{l_1\mathbf{d}_1, l_2\mathbf{d}_2, -l_1\mathbf{d}_1, -l_2\mathbf{d}_2\}$ , where  $\mathbf{d}_1, \mathbf{d}_2$  are unit vectors, the ideal solution would yield a metric  $g$  that takes  $F$  to a crossfield  $C = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_1, \mathbf{d}_2\}$  on the curvature shape, thus the optimal solution

$\mathbf{G}_{opt} = \begin{pmatrix} \frac{1}{l_1^2} & 0 \\ 0 & \frac{1}{l_2^2} \end{pmatrix}$ . The actual solution would be

$$\mathbf{G} = \begin{pmatrix} \frac{1}{s_1^2} & 0 \\ 0 & \frac{1}{s_2^2} \end{pmatrix}.$$

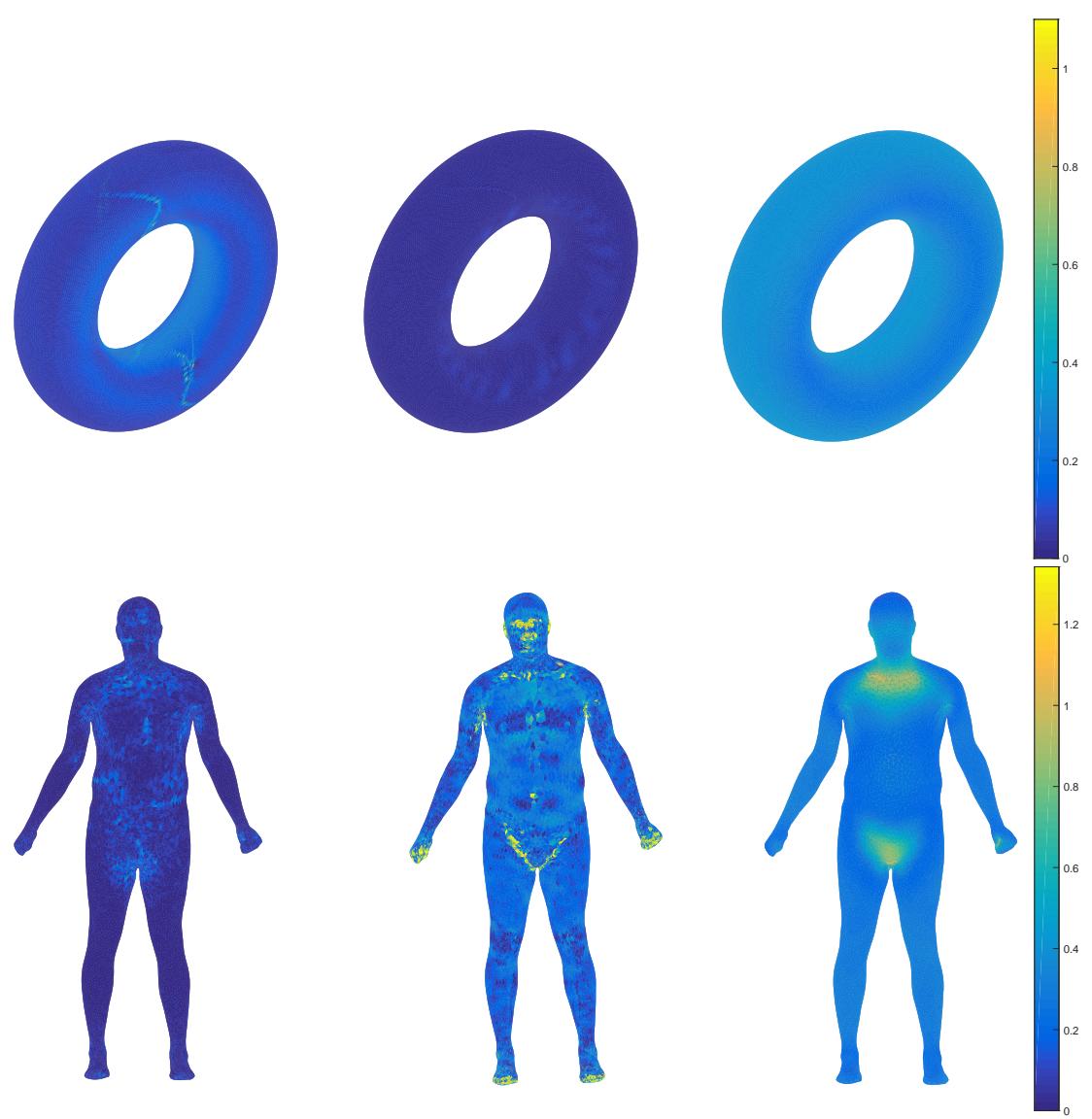
We propose the following metric to measure distortion

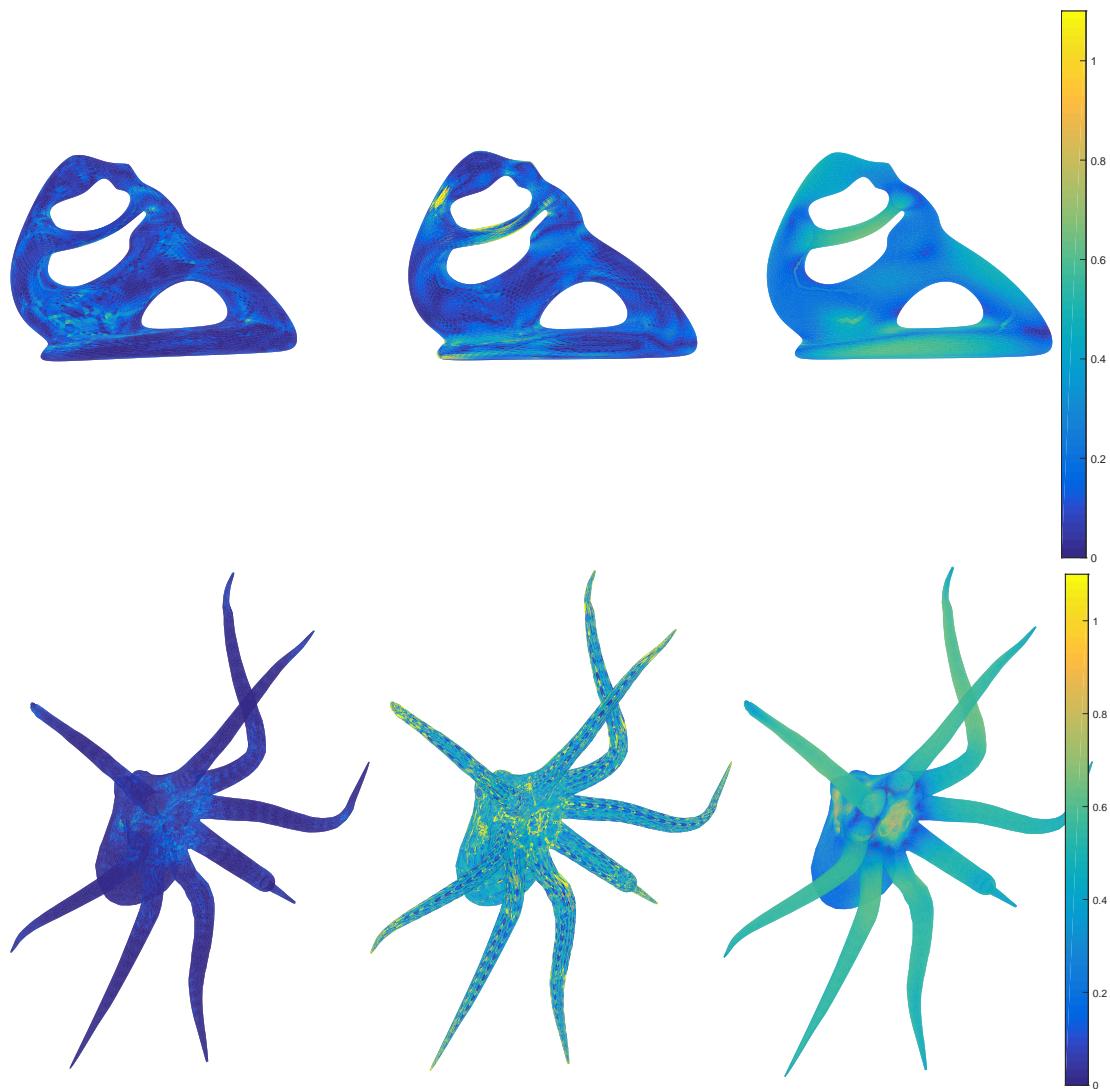
$$q = \frac{\|\mathbf{G} - \mathbf{G}_{opt}\|}{\|\mathbf{G}_{opt}\|} = \frac{\|\mathbf{W}\mathbf{W}^\top - \mathbf{W}_{opt}\mathbf{W}_{opt}^\top\|}{\|\mathbf{W}_{opt}\mathbf{W}_{opt}^\top\|},$$

where  $\|\cdot\|$  is the Frobenius norm. We can estimate the metric achieved by the algorithms in [PPTSH14] and [JFH<sup>+</sup>15] by looking at the output frame field. If the algorithm achieves the perfect metric then the following identity holds

$$F = h_{opt}^{-1}(C)$$

In particular, in the case of an original orthogonal anisotropic frame field  $F = \{l_1\mathbf{d}_1, l_2\mathbf{d}_2, -l_1\mathbf{d}_1, -l_2\mathbf{d}_1\}$  and a resulting frame field  $\bar{F} = \{s_1\mathbf{d}_1, s_2\mathbf{d}_2, -s_1\mathbf{d}_1, -s_2\mathbf{d}_2\}$  we can write  $h(\bar{F}) = \mathbf{W}\bar{F} = \begin{pmatrix} \frac{1}{s_1} & 0 \\ 0 & \frac{1}{s_2} \end{pmatrix} F = \bar{C}$ . Since different algorithm output differently scaled frame fields we will account for it by finding a global scalar multiple which results in lowest distortion computed as described above.





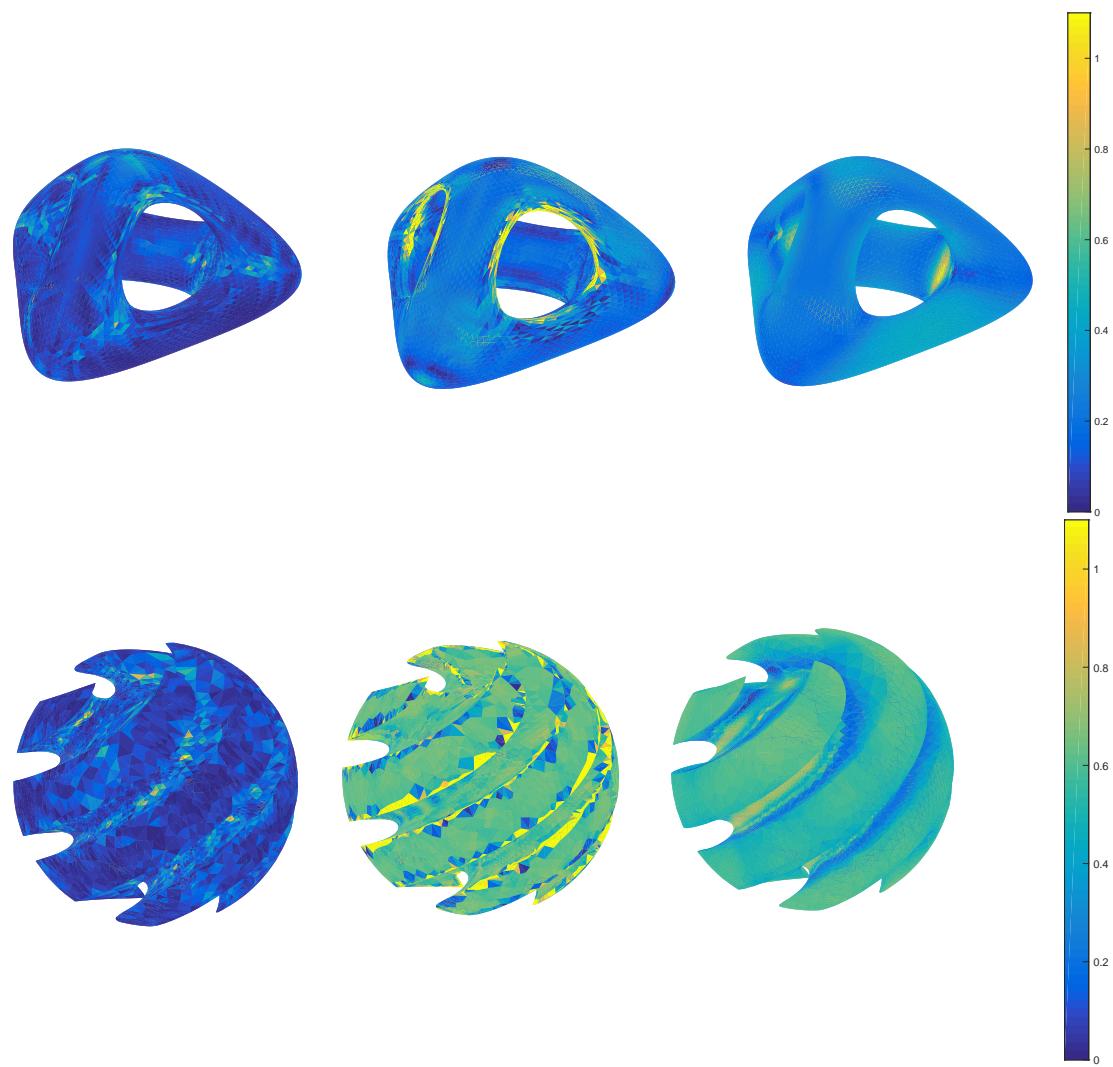


Figure 7: Metric distortion: our result – left, [PPTSH14] – middle, [JFH<sup>+</sup>15] – right

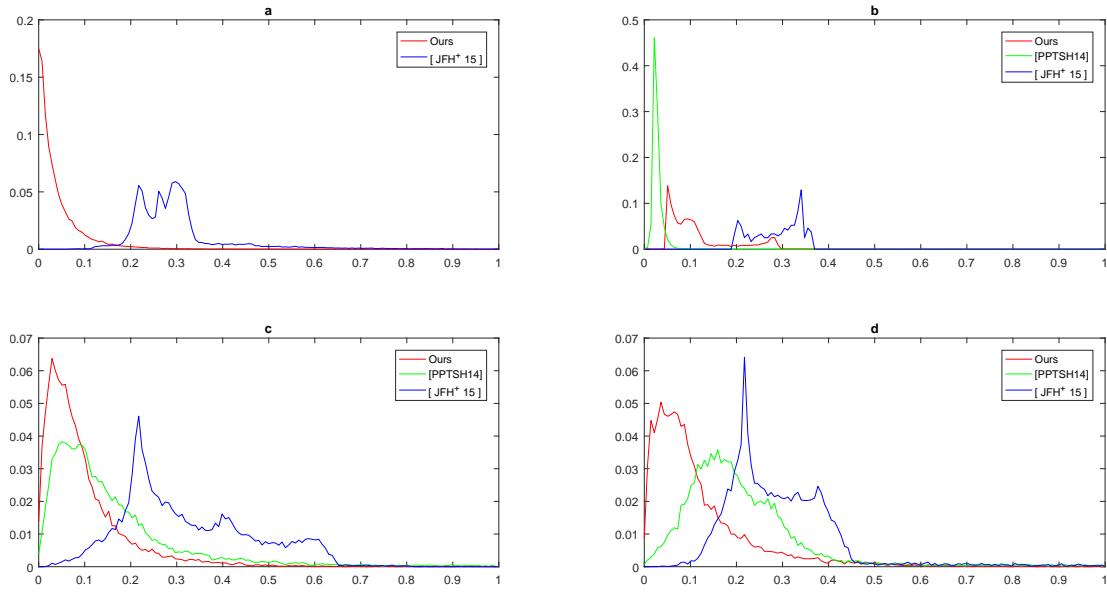


Figure 8: Metric distortion: a) Human, b) Torus, c) Sculpture d) Genus 3 surface. The graph for [PPTSH14] in a) is not plotted due to a high distortion

#### 4.1.2 Optimisation

We have shown the results when metric is unconstrained on an entire surface (complies with a given frame field), however for many practical applications we would like to constrain the metric in a situation when principal directions do not make much sense, such as in spherical regions. For this purpose we will constrain the optimisation function to output an identity transformation in regions when we don't need our resulting frame field to comply with the original frame-field. We used Quadprog in Matlab to solve the optimisation problem and the nature of constraints affects the quality of the solution to a great extent. We attempted to implement constraints in several ways:

1. By providing an upper and lower bounds on elements of the metric in the regions of interest
2. By imposing linear constraints of the form  $\mathbf{Ax} = \mathbf{b}$
3. By setting the optimal transformation to identity in the restricted region and applying weights.

We found the latter option, combined with providing an initial point achieved by running the algorithm without constraints provides the best results. We also achieved a lower distortion than [JFH<sup>+</sup>15]. In Figure 9 it can be seen that in the areas where we seek a metric conforming with the original frame-field we get a lower distortion than [JFH<sup>+</sup>15]. In the restricted areas the distortion is expected to be high since we are not looking for a metric conforming with the original frame field. Thus it's possible to achieve good

results, however they are not guaranteed with the given solver. For some larger shapes we were not able to achieve better results – merely introducing constraints, even if they in fact do not restrict anything worsens the results. We think it's likely due to the flaws of our optimisation process rather than the algorithm's inherent properties.

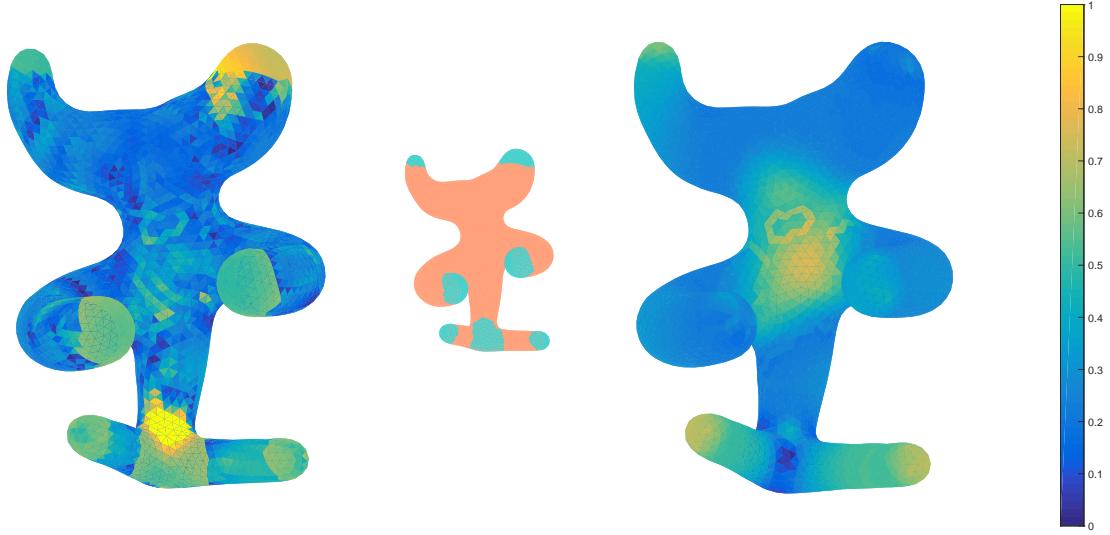


Figure 9: Metric distortion: our result – left, [JFH<sup>+</sup>15] – right. Constraints are shown in blue in the middle.

## 4.2 Computing smooth N-RoSy fields

In this section we will describe how to compute a smooth vector field on a surface affected by the metric change described earlier. Our approach is based on [KCPS13]. Again, we will assign one piece of data per face, however, there are no obstacles to other intrinsic approaches. N-RoSy fields is a generalisation of cross-fields where N vectors rotated by an integer multiple of  $\frac{2\pi}{n}$  are assigned to a point. We will denote  $\iota := \sqrt{-1}$  and we will reserve  $i$  and  $j$  for indices of faces. Given a chosen unit basis vector  $X_p \in T_p M$  every vector can be expressed as a complex function

$$c : \mathbb{C} \rightarrow \mathbb{C}$$

$$z \longmapsto z e^{2k\pi/n},$$

for  $k, n \in \mathbb{N}$ . A map  $u := z^n$  maps an N-vector to a single vector. The inverse function can recover the original vectors. The Levi-Civita connection, which is a rotation is expressed as

$$r_{pq} = e^{\iota\theta_{pq}}.$$

Then the parallel transport map takes form

$$\nabla(z_p X_p) := e^{\iota \theta_{pq} z_p X_q}$$

The difference between two vectors is given by

$$\|e^{\iota \theta_{pq}} z_q - z_q\|^2$$

The difference of two N-vectors can be written as

$$\|(e^{\iota \theta_{pq}} z_p)^n - (z_q)^n\|^2 = \|e^{\iota n \theta_{pq}} z_p^n - (z_q)^n\|^2 = \|e^{r_{pq}} u_p - u_q\|^2,$$

where  $r_{pq}$  is the connection for N-vector fields.

Finally we construct an energy matrix

$$\mathbf{M} = \mathbf{L}^\top \mathbf{L},$$

where  $\mathbf{L}$  is a  $N_{edges}$  by  $N_{faces}$  matrix. The matrix  $\mathbf{L}$  is a sparse matrix where for each row  $i$  there is a transport coefficient  $e^{\iota \theta_{jk}}$  in  $j$ s element of the row if we don't apply alignment constraints and 1 if we apply alignment constraints – this is a slight departure from [KCPS13] since we apply constraints on the connection. In order to align with the features we will have to use a global rotation after recovering the smooth vector field. The  $ks$  element of the row equals  $-1$ . Multiplying this matrix by a vector  $\mathbf{u}$ , which is obtained through the map  $u$  of an  $N$ -RoSy field, we get a vector  $\mathbf{L}^\top \mathbf{u}$ , representing the differences between vectors in neighbouring faces with respect to a given connection. Squaring this vector, we obtain the desired energy

$$E\mathbf{u} = \mathbf{u}^\top \mathbf{L}^\top \mathbf{L} \mathbf{u} = \mathbf{u}^\top \mathbf{M} \mathbf{u}.$$

Obtaining the minimum of this energy is a singular value problem, see [KCPS13] for details.

#### 4.2.1 Change of metric and smooth vector fields

The metric change described earlier can easily be translated into the language of complex functions. Recall that the linear transformation  $\mathbf{W}$  is anisotropic scaling. A vector  $\mathbf{u} = \mathbf{W}\mathbf{v}^\top$  can be represented as a complex number  $a e^{\iota \psi}$ ,  $a \in \mathbb{R}$ . With this in mind, we deduce that under a metric transformation  $h$ , the difference between two vectors

$$\nabla_{ij}^h = \|(e^{\iota \theta_{pq}} a_p e^{\iota \psi_p} z_p)^n - (a_q e^{\iota \psi_j} z_q)^n\|^2 = \|e^{r_{pq}} a_p e^{\iota n \psi_p} u_p - a_q e^{\iota n \psi_q} u_q\|^2 = \|e^{r_{pq}} v_p - v_q\|^2$$

where  $v = (h(z))^n$ . For an ideally smooth vector field this difference is equal to zero, and as we can see in Figure 10, generally smoothness is not preserved under the transformation  $h$  as well as its inverse. If a vector field is perfectly aligned with

some features (Subsection 4.2.2, this alignment is obviously preserved under a linear transformation  $h$  (and its inverse). Figure 10 shows the result of the transformation  $u^{-1} \circ h^{-1}$  applied to a smooth vector field.

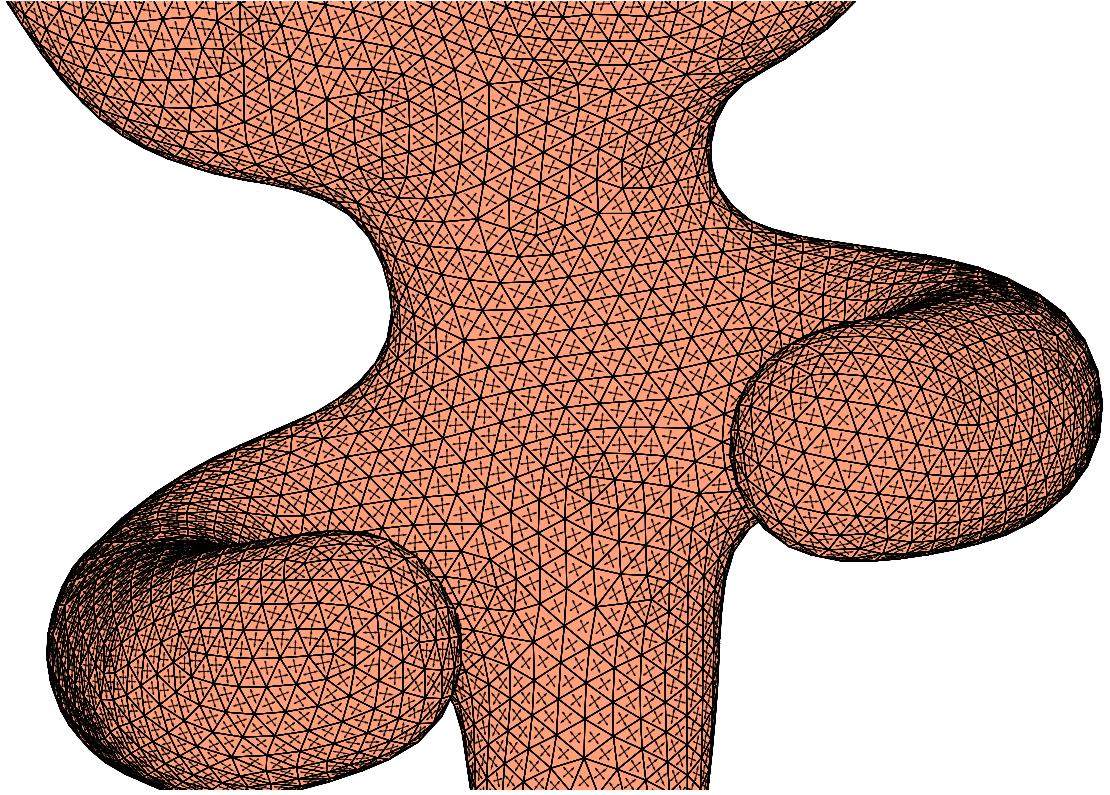


Figure 10:  $u^{-1} \circ h^{-1}$

#### 4.2.2 Results

In this section we will illustrate some of the properties of our method. Consider the map we discussed in Section 4.2:  $u := z^n$  that maps an  $N$ -vector to a single vector. We design a smooth vector field on a surface. To recover the smooth  $N$ -vector field from a smooth vector field we just need to take the inverse of the map  $u$ . However if we take  $h^{-1} \circ u^{-1}$ , generally, we get a non-orthogonal  $N$ -vector field where the notion of smoothness is not defined as illustrated by Figure 10. The non-orthogonality appears where the vector field is not precisely aligned with the basis (which is defined by principle directions) and the linear transformation  $h^{-1}$  skews the original  $N$ -RoSy field. In regions where we want to interpolate the original frame field we restrict the optimisation problem (3.4) to designing the identity metric  $\mathbf{G} = Id$ . Then the linear transformation  $h = Id$ . Thus when pulled onto the original surface by  $h^{-1} \circ u^{-1} = Id \circ u^{-1}$  the cross field will remain unaffected. In regions where conformity with the features is required, the cross field is constrained to align with the original frame field and  $h^{-1}$  preserves the orthogonality

only changing the magnitudes of the vectors in the cross field. This leads us to another property. If a vector field  $F_N$  is aligned with the basis then  $h^{-1} \circ u^{-1}$  is aligned with the basis as well. Subsection 4.2.2 show a pullback of a vector field, perfectly aligned with the basis (features).

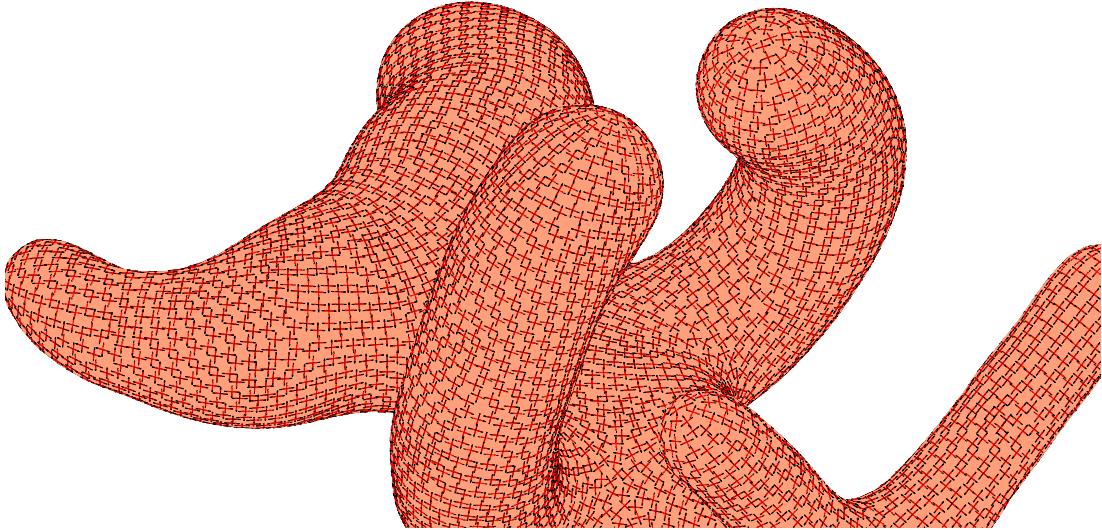


Figure 11: A vector field on a curvature shape perfectly aligned with principal curvature directions (red) and pulled back onto the original surface – black. Principle curvature directions – red

In light of the above, to design a frame field that is conforming with given features (PCDs) and interpolates them in the areas where such features are not defined, as discussed in Section 3.3 we design a metric  $g$  driven by the original frame field in regions of interest and is an identity on the rest of the surface. Then we design a smooth cross-field as explained in Section 4.2 restricting it in areas with features. We get a cross-field that is aligned with features where they are present and smooth on the rest of the surface. When such a vector field is pulled back onto the original surface aligned crosses transform into the original frame field to the extent the metric  $g$  we found admits, while the cross field transformed by the identity remains a smooth cross-field. Below are the results achieved by our method and [JFH<sup>+</sup>15]. Note that where constraints on our metric are present (refer to Figure 9), that is where we design a smooth unaligned frame field, the vector field is smooth and is aligned with the original frame field (pcds) on the rest of the surface. The amount of metric distortion shown in Figure 9 corresponds to the difference between the magnitude of the frame field we design and the original frame field.

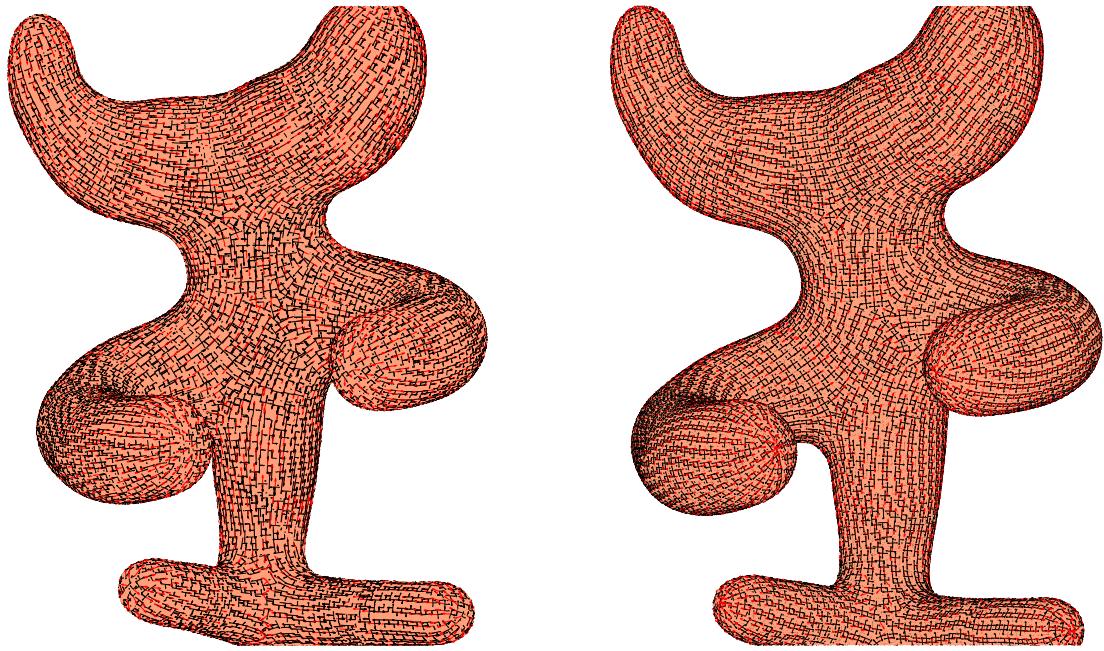


Figure 12: A smooth cross field (black) pulled back onto the original surface. Left – ours, right – [JFH<sup>+</sup>15]. We can observe that the framefield conforms with the PCDs (red) where it’s required and interpolates them where conformity is not required. See Figure 9.

### 4.3 Conformal parametrisation

Conformal parametrisation is another area where our algorithm can be applied with some advantages over [JFH<sup>+</sup>15] and [PPTSH14]. Our method is less restrictive than [PPTSH14] in terms of metric and at the same time has all the benefits of a real polytope in contrast to [JFH<sup>+</sup>15]. In this section we will show that our metric transformation is enough to accommodate a conformally flat mapping – we will not develop a new method to find a conformal transformation, merely will show that our framework subsumes this class of problems if an input in the form of a frame field is provided. Following [BCGB08] we will seek a conformal transformation that prescribes zero Gaussian curvature everywhere. Since it’s only feasible for manifolds of Euler characteristic 0, we will find a conformally flat mapping of a torus. We will adapt our method to the solution described in [BCGB08]. Consider a metric as a collection of values assigned to edges. These values completely specify the metric structure of a manifold. A conformal Riemannian metric in the continuous case can be achieved by scaling the original metric by a scaling function  $e^{2\phi}$ . The relation between the Gaussian curvature of the original surface and its conformal mapping is described by the following differential equation

([SR94])

$$\Delta\phi = K_G^0 - e^{2\phi} K_G^C,$$

where  $\Delta$  is the Laplace-Beltrami operator of the manifold,  $K_G^0$  is the original Gaussian curvature and  $K_G^C$  is Gaussian curvature of the conformal map. In the case where the conformal map is flat this equation reduces to

$$\Delta\phi = K_G^0$$

Proceeding as in [BCGB08] we find a corresponding scaling function  $\phi$  for each edge. The next step is to find such a linear transformation  $h$  represented by matrices  $\mathbf{W}$  in each face, that for each edge  $\mathbf{l}_{ij}$  adjoining faces  $i$  and  $j$ ,  $\mathbf{W}_i \mathbf{l}_{ij}^\top = e^{2\phi} \mathbf{l}_{ij}^\top$ , where  $\phi$  is found with [BCGB08]. To find an approximate embedding in  $\mathbb{R}^3$  of the conformally flat torus endowed with the metric we found with [BCGB08] we will combine our method with [PPTSH14]. We apply the inverse  $h^{-1}$  to a cross-field on the original shape and feed the resulting frame field to the algorithm in [PPTSH14]. Since the algorithm attempts to find a metric warping this frame field into the original cross-field, we expect to get an approximation of the conformally flat metric as a result. The linear transformation found by our metric gives a virtually exact conformally flat metric. We plot the conformal distortion as a vector  $\mathbf{W}_i(1, 1)^\top$  in every face. If the transformation is conformal the vector scales along the diagonal (unless it's an eigenvector – then it scales even if the transformation is not conformal). The conformal distortion seen in Figure 17 is explained by the fact that the original algorithm has some baseline distortion. [PPTSH14] cannot accommodate the metric and introduces additional distortion. We do not how to apply the method in [JFH<sup>+</sup>15] within this framework since it does not construct a polytope.

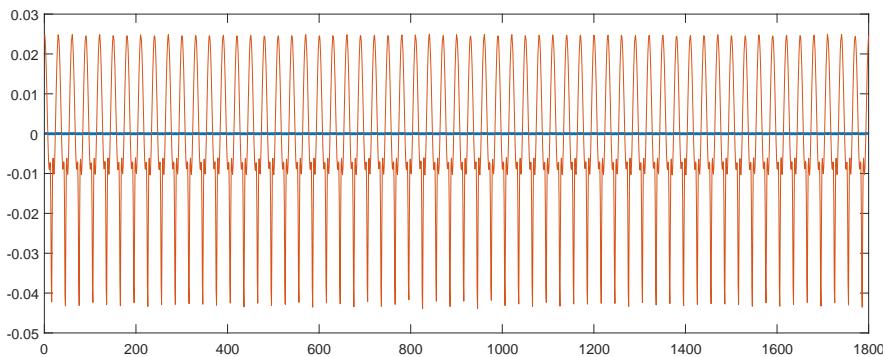


Figure 13: Distribution of Gaussian curvature: blue – intrinsic, red – embedding

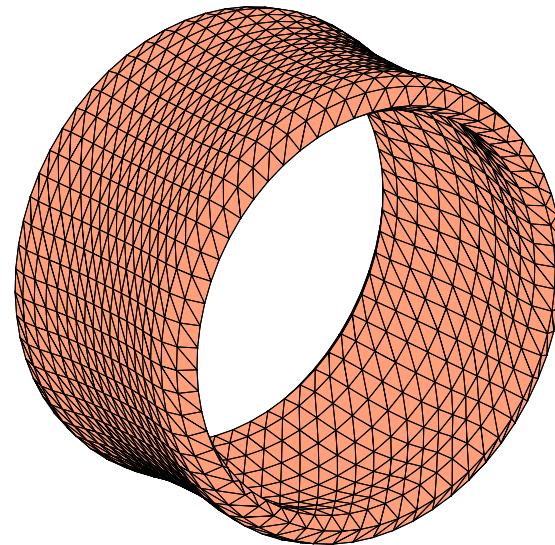


Figure 14: An approximate embedding of the flat torus in  $\mathbb{R}^3$  found with [BCGB08]

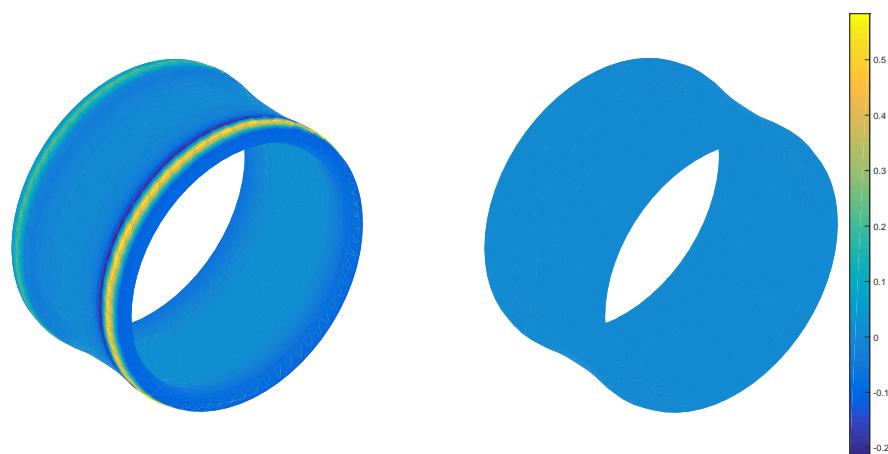


Figure 15: Gaussian curvature: [PPTSH14] – left, ours – right.

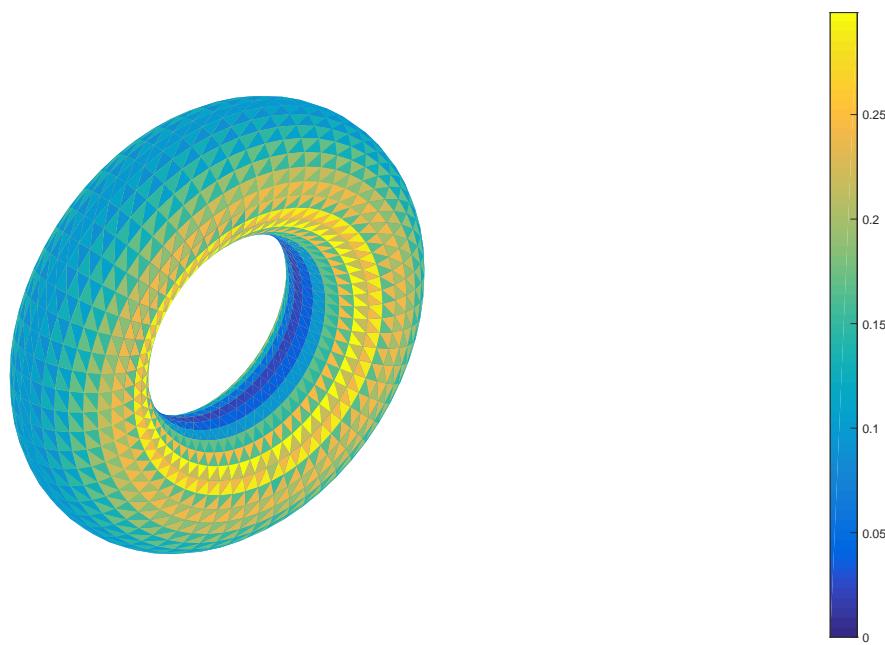


Figure 16: Metric distortion [PPTSH14]

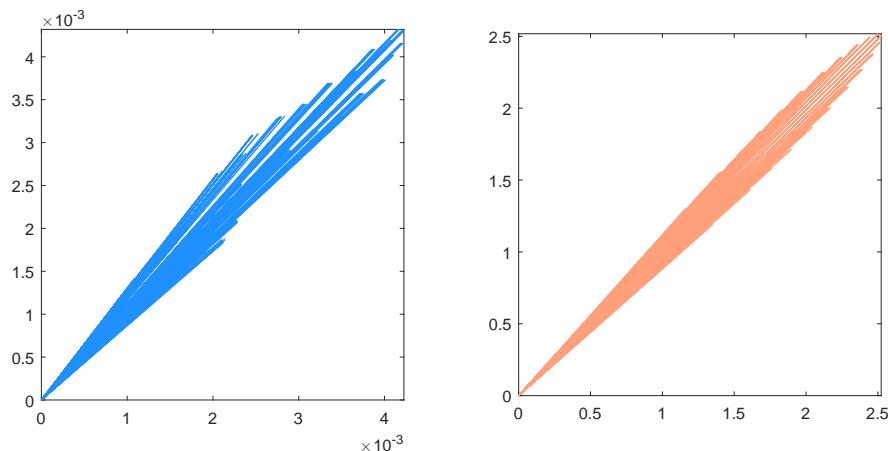


Figure 17: Conformal distortion: ours – right, [PPTSH14] – left

## Chapter 5

# Conclusion

We proposed a simple and efficient algorithm for frame field design. We showed that in many instances our algorithm outperforms state of the art methods yet has a simple formulation which makes it compatible with other existing computer graphics tools. However, just as in [PPTSH14] we cannot guarantee that the mesh we get as an intermediate step of our algorithm is of high quality – it's possible that some input frame fields can lead to close to degenerate elements. Another problem, we encountered, though it's likely not inherent to our algorithm but rather to the solver, is that it's hard to find an optimal solution. We noticed that certain constraints on the optimisation problem can lead to poor results. Finally, we only explored the metric transformations driven by orthogonal frame fields, however there are no obstacles to applying our algorithm to more general frame fields.



# Bibliography

- [BCGB08] Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum*, volume 27, pages 449–458. Wiley Online Library, 2008.
- [BLP<sup>+</sup>13] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-Mesh Generation and Processing: A Survey. In *Computer Graphics Forum*, volume 32, pages 51–76. Wiley Online Library, 2013.
- [BZK09] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)*, 28(3):77, 2009.
- [JFH<sup>+</sup>15] Tengfei Jiang, Xianzhong Fang, Jin Huang, Hujun Bao, Yiyi Tong, and Mathieu Desbrun. Frame field generation through metric customization. *ACM Transactions on Graphics (TOG)*, 34(4):40, 2015.
- [JTPSH15] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA)*, 34(6), November 2015.
- [KCPS13] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. Globally optimal direction fields. *ACM Trans. Graph.*, 32(4), 2013.
- [KMZ10] Denis Kovacs, Ashish Myles, and Denis Zorin. Anisotropic quadrangulation. *Proceedings - 14th ACM Symposium on Solid and Physical Modeling, SPM'10*, pages 137–146, 2010.
- [LH06] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics (TOG)*, 25(3):541–548, 2006.
- [MDSB01] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Citeseer, 2001.

- [Nas56] John Nash. The imbedding problem for riemannian manifolds. *The Annals of Mathematics*, 63(1):20, 1956.
- [PPTSH14] Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. Frame fields: Anisotropic and non-orthogonal cross fields. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 33(4):134:1–134:11, 2014.
- [PS06] Konrad Polthier and Markus Schmies. *Straightest geodesics on polyhedral surfaces*. ACM, 2006.
- [Spi70] Michael Spivak. *A comprehensive introduction to differential geometry. Vol. 2*. Waltham, Mass., 1970.
- [SR94] Yau S.-T. Schoen R. *Lectures on Differential Geometry*. Intl Press, 1994.
- [Whi57] H. Whitney. *Geometric Integration Theory*. Princeton University Press, 1957.

שטרנספורמציות לינאריות אלה תהינה קומפקטיביות ותשמרנה את הטופולוגיה של היריעה. אז אנו מראים שטרנספורמציה כזו מובילה לפוליטופ חדש, הומוורפי למקורו ומשתמש באותו קשר Levi-Civita. ואז אנו מגדירים ופותרים בעית אופטימיזציה שמחפש מטריקה אופטימלית, המתאימה לשדה הקטורים המקורי. אנו מראים שהגישה שלנו נותנת תוצאות יותר טובות בחלק מהמקרים. אנו מישמים את הגישה שלנו לעיצוב שדות פריאמיים ומדגימים שהגישה שלנו יכולה להתאים לספקטורות רחב יותר של מטריקות מאשר Panozzo et. al. עם פורמלציה יותר פשוטה לעומת Jiang et. al. מה שהופך אותה מתאימה ליישום כל הכלים של גרפיקה ממוחשבת המתאים לפוליטופים שאינם נסמכים על מידע חיצוני.

## תקציר

עיצוב שדות וקטוריים הוא נושא חשוב בגרפיקה ממוחשבת. לשדות וקטוריים יש שימושים רבים, ביניהם קוואדרנגולציה ופרמטריזציה של רשת משולשים. גישות לעיצוב שדות וקטוריים ניתנות בגודל לחולקה לגישות חיצונית (אקסטריניסטיות) או אינטראיניסטיות, הראשונה משתמשת בשיכון מרחב המקיים, והאחרונה עוסקת רק במטריקה.\_CIDOU, לא כל יריעה דו מימדית ניתנת לשיכון למרחב אוקלידי תלת מימדי, ולכן כמה מחברים הציעו שיבוצים למרחבים אוקלידיים מסדר גבוה יותר. Kovacs et. al. הציעו להשתמש במרחב אוקלידי 6 מימדי לעיצוב מטריקה לפרמטריזציה. Panozzo et. al. הציע לעצב שדה צלבים על משטחBINIIM שבינוי כך ששדה פריאמיים על המשטח המקורי מועבר לשדה צלבים תחת הטרנספורמציה הנוגעת לשתי הצורות. המחברים פותרים בעית אופטימיזציה קוודראטית מעל  $\mathbb{R}^3$ . לגישה זו יש פורמלציה פשוטה, אך ישן כמה מגבלות. אין ערובה שימושה בתוצאה ישר את הטופולוגיה שלו, וגם ישנה אפשרות שהוא יכול להיות שיכונים שונים. לבסוף, לא כל המטריקות יכולות להיות משוכנות ב  $\mathbb{R}^3$  מה שמנבל את מרחב הפתרונות בעית המינימיזציה. ב. Jiang et. al. מוצע להשתמש במטריקה רימנית מוגדרת על כל פאה שבו שדה הפריאמיים הופך לשדה צלבים ומספק את מגבלות המשטח. גישה זו היא אינטראיניסטיבית כשלנו, אך ישנים הבדלים חשובים. הויאל צעד הביניים לעיצוב שדה פריאמיים חלק על המשטח המקורי מציריך לעצב שדה צלבים חלק (או יותר בכלליות, שדה N-Rosy) על משטח הביניים, הכרחי לקרב את קשר ה Levi-Civita. המחברים מבטאים את זיות הסיבוב המבטה את הקשר ממשוואות מבנה Cartan ואז מעריכים את הגדים הנדרשים באמצעות הקשות, בהשתמש בערכים מטריים הקשורים לפאות. אך הם אינם יוצרים אובייקט "אמתית" באותו מבן שפוליטופ הוא אובייקט אמיתי. כמובן, הם לא יוצרים מרחב מטרי, אלא דגימה של מרחב מטרי עם ערכים מטריים שرك מוגדרים על הפאות. לכן, אם יש צורך להשתמש בגרפיקה ממוחשבת, יש צורך בקירובים נוספים. לדוגמה, לא ברור איך לבנות עקומות גאודזיות על אובייקט כזה הויאל זה לא מספיק להגדיר מטריקה על כל פאה אלא אם האובייקט הוא פוליטופ, ובדרך כלל הוא אינו כזה, לפי המחברים. לדוגמה, Jiang et. al. מגדירים טכנית קוואדרנגולציה משלהם. לעומת זאת, אנו בונים אובייקט שהוא פוליטופ ולכן אפשר לישם כל שיטה אינטראיניסטיבית קיימת של גרפיקה ממוחשבת, למשל Bommes et. al., מה שימושי מאד, הויאל ופוליטופים הם אובייקטים דיסקרטיים שנחקרו ונלמדו רבות ונפוצים בגרפיקה ממוחשבת. אנו גם נפטרים מה צורך לשאת את המידע החיצוני שבדרך כלל בא עם פוליטופים.

בעובדה זו אנו מציעים את הדרך הבאה לעצב מטריקה המתאימה לשדה פריאמיים נתון. ראשית, אנו משכנים כל פאה למרחב אוקלידי משלה ומעניקים למרחב זה בסיס נוח. בהשתמש בגישה של Knoppel et. al. אנו משתמשים בקשר Levi-Civita. אנו מראים שניי מטריקה ניתנת להשגה על ידי טרנספורמציה לנארית לכל פאה המוגדרת בסיס המתאים. אנו מראים שאנו מכך



המחקר בוצע בהנחייתו של פרופסור מירלה בר-חן, בפקולטה למדעי המחשב.

אני מודה לטכניון על התמיכה הכספית הנדיבת בהשתלמותי.



# שינוי מטריקה עם מיפויים לינאריים לוקליים, עם אפליקציה לייצור שדות וكتוריים

חיבור על מחקר

לשם מילוי תפקידי של הדרישות לקבלת התואר  
מגיסטר למדעים במדעי החישול

אלכסיי שבליקוב

הוגש לשנת הטעניון – מכון טכנולוגי לישראל  
אלול תשע"ו ספטמבר 2016 חיפה



**שינויי מטריקה עם מיפויים לינאריים  
לוקליים, עם אפליקציה לייצור שדות  
וקטוריים**

**אלכסי שבליקוב**