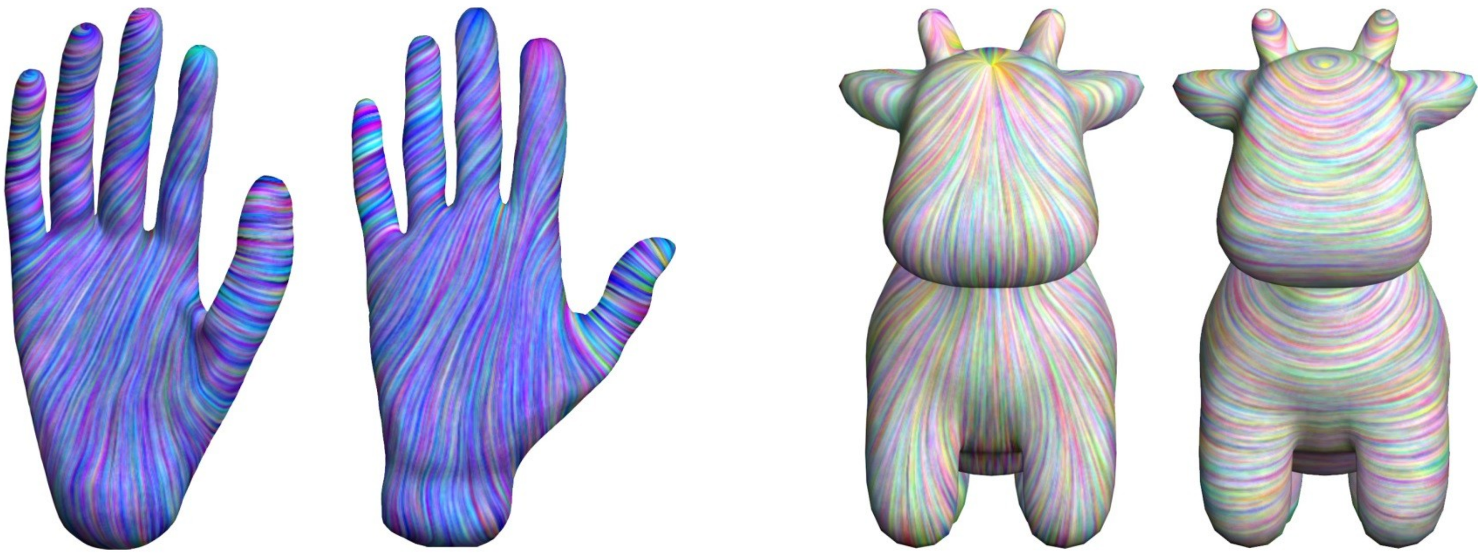
An Operator Approach to Tangent Vector Field Processing

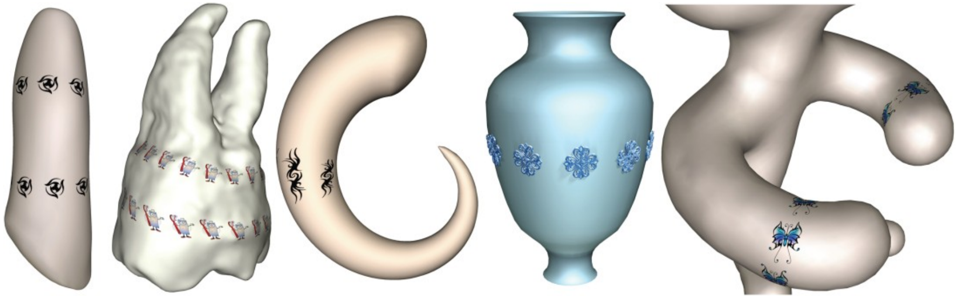


Omri Azencot
Technion

Joint work with: Mirela Ben-Chen, Frederic Chazal and Maks Ovsjanikov

Motivation

Pattern Generation

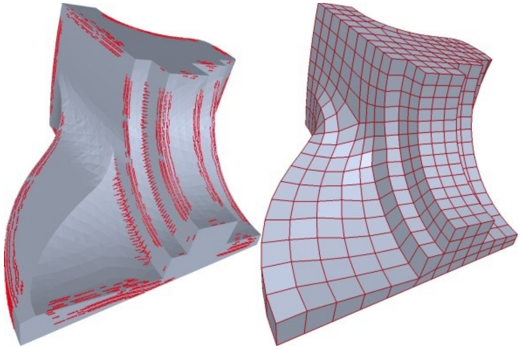


[Ben-Chen et al. 10]

Texture Synthesis

[Fisher et al. 07]

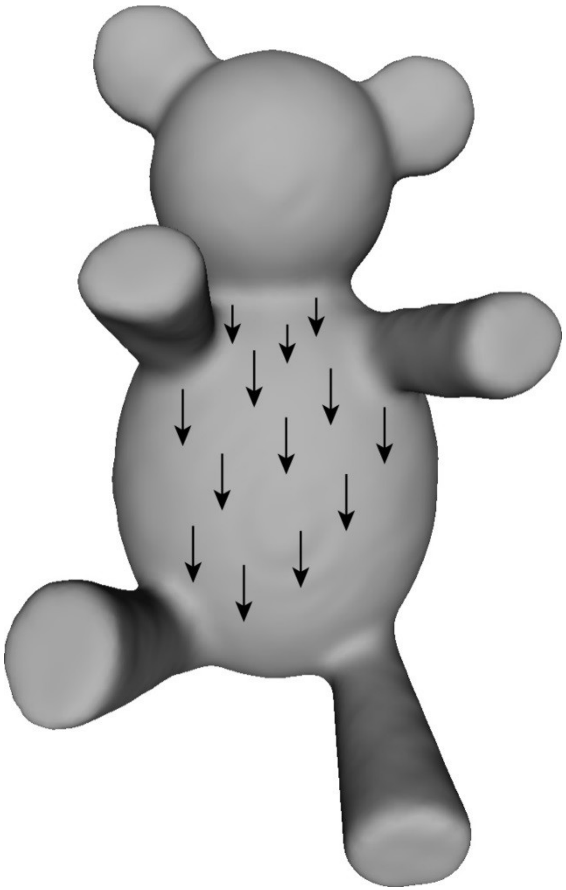
Quad Remeshing



[Bommes et al. 09]

What is a Vector Field (VF)?

Vector field V



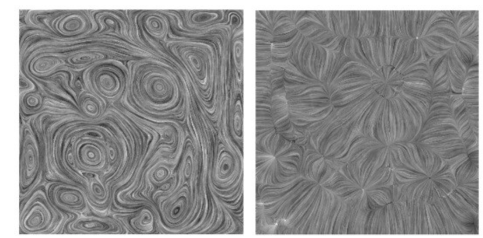
Flow ϕ_V^t

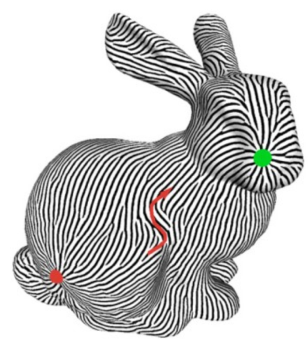
Representing Vector Fields

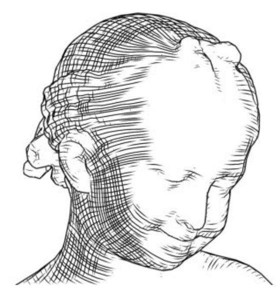
- A powerful toolbox with ability to pose:
  - Low-level constraints, e.g. singularities
  - High-level constraints, e.g. symmetry
- Relate between vector fields and mappings, e.g. flow
- An efficient and robust optimization framework

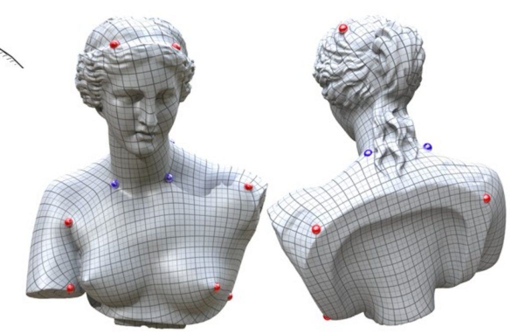
Previous Work

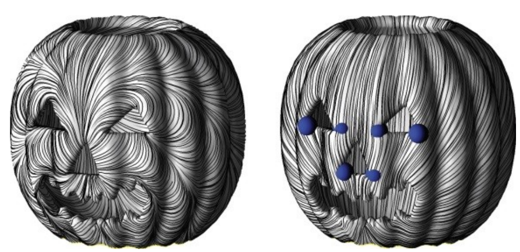
- Tangent vector per simplex
  - [Polthier et al. 03]
  - [Tong et al. 03]
- DEC
  - [Fisher et al. 07]
- N-RoSy fields
  - [Palacios et al. 07]
  - [Ray et al. 09]
  - [Crane et al. 10]











Our Approach

- Represent VFs using operators: $\mathrm{V} \leftrightarrow D_V$
- D_V acts on smooth functions defined on M
- A common view in differential geometry

geometry

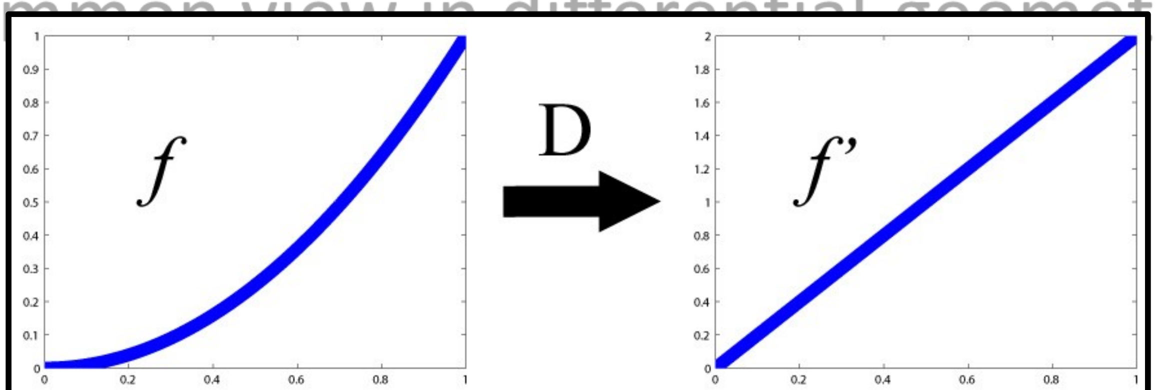
Our Approach

- Represent VFs using operators: $V \leftrightarrow D_V$

The **derivative** of scalar functions is an operator:

$$\frac{df}{dx}(x_0) = \lim_{t \to 0} \frac{f(x_0 + t) - f(x_0)}{t} \iff \frac{\boldsymbol{df}}{\boldsymbol{dx}}(\boldsymbol{x_0}) = \boldsymbol{D}(\boldsymbol{f})(\boldsymbol{x_0})$$

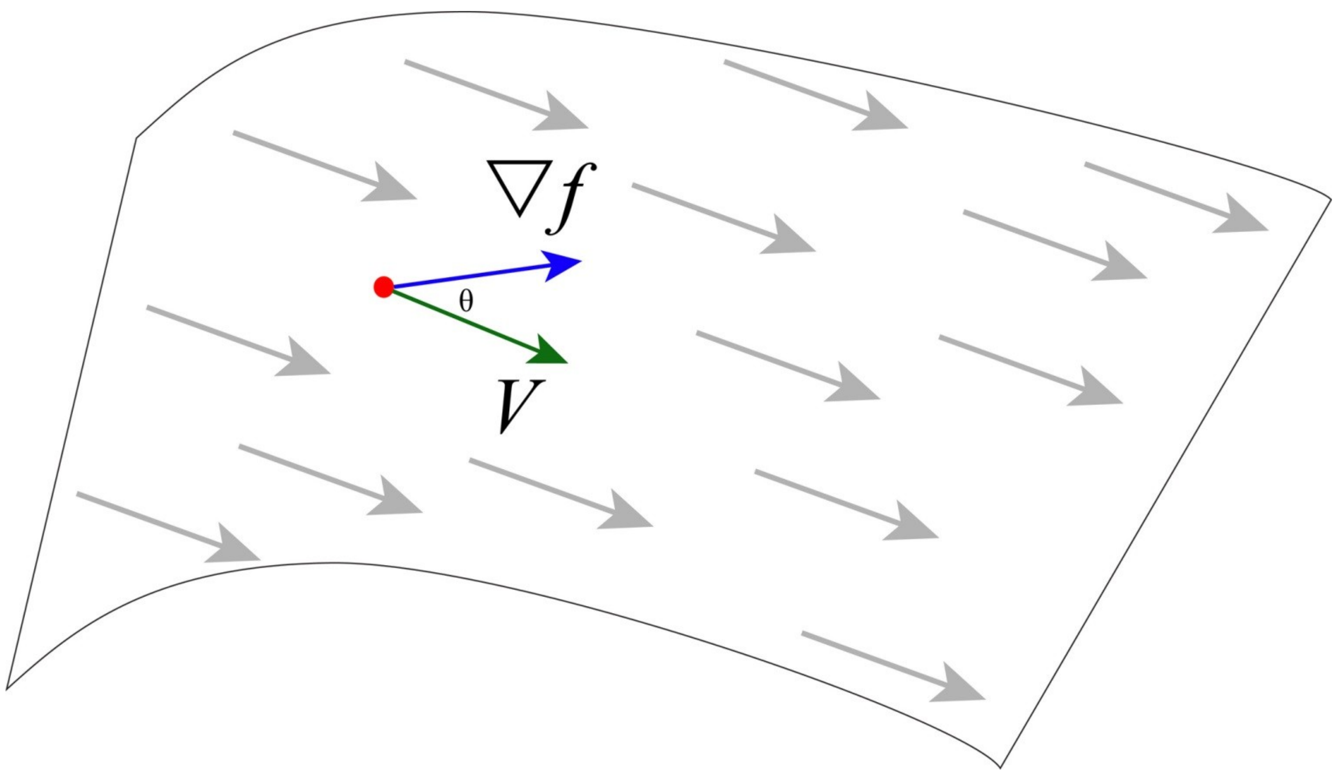
- A common view in differential geometry



Representing Vector Fields

Using Functional Vector Fields (**FVFs**):

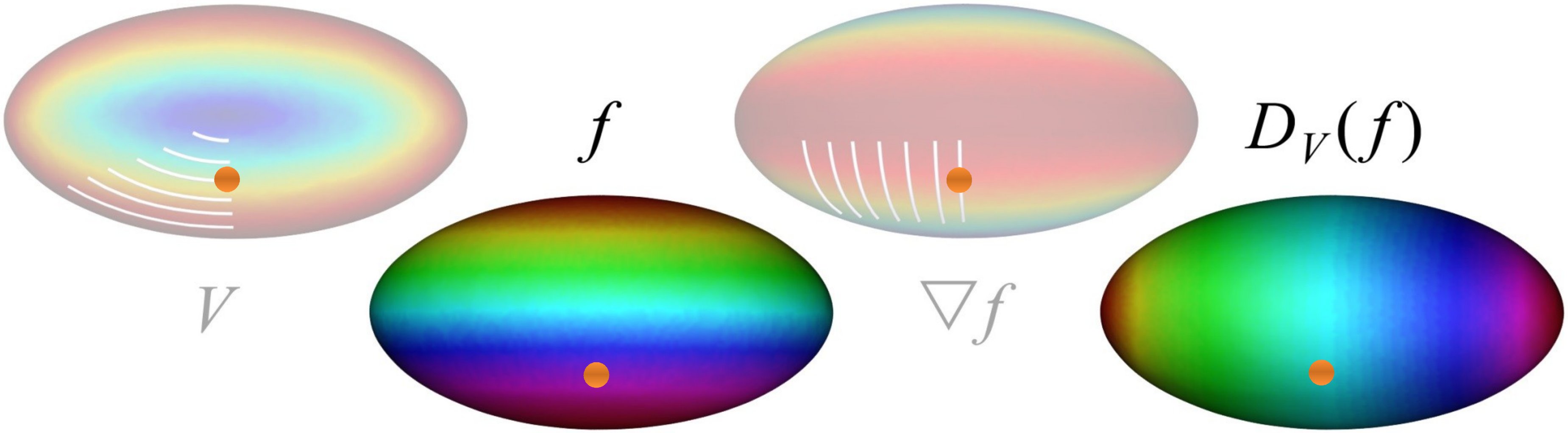
$$D_V(f) = < V, \nabla f >$$



FVFs
An example

Using Functional Vector Fields (**FVF**s):

$$D_V(f) = < V, \nabla f >$$



FVFs

Properties

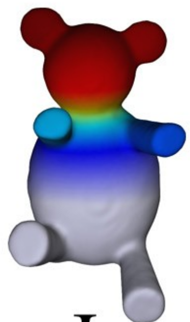
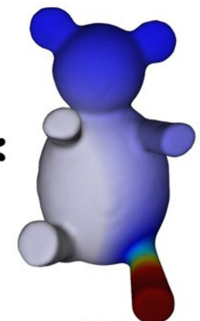
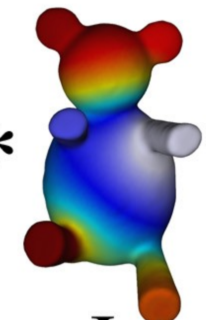
- D_V is an FVF if and only if it fulfills:
  - Linearity:
    $$D_V(\alpha f + \beta g) = \alpha D_V(f) + \beta D_V(g)$$
  - The product rule:
    $$D_V(fg) = f D_V(g) + g D_V(f)$$
- V can be reconstructed from D_V
- D_V is also called the **covariant derivative**

derivative

Matrix Representation

How do FVFs look like?

- Basis $\mathbf{\Phi}$ for the function space:

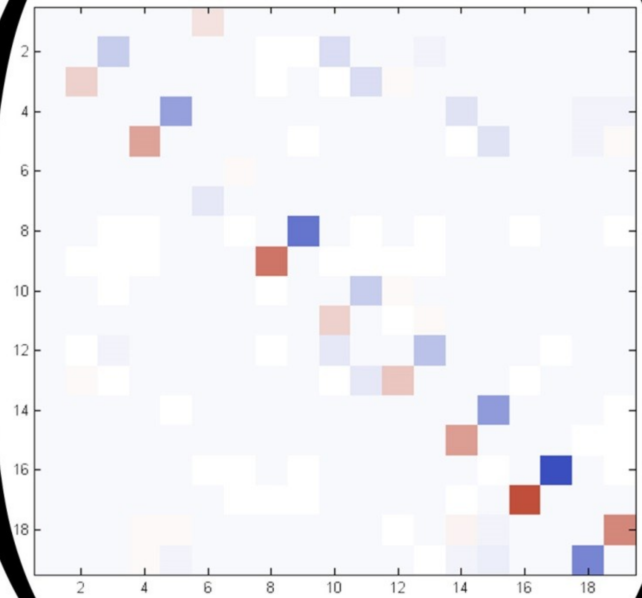
$$f = F_1 * \phi_1 + F_2 * \phi_2 + F_3 * \phi_3 + \ldots$$

- Laplace-Beltrami eigenfunctions

Matrix Representation

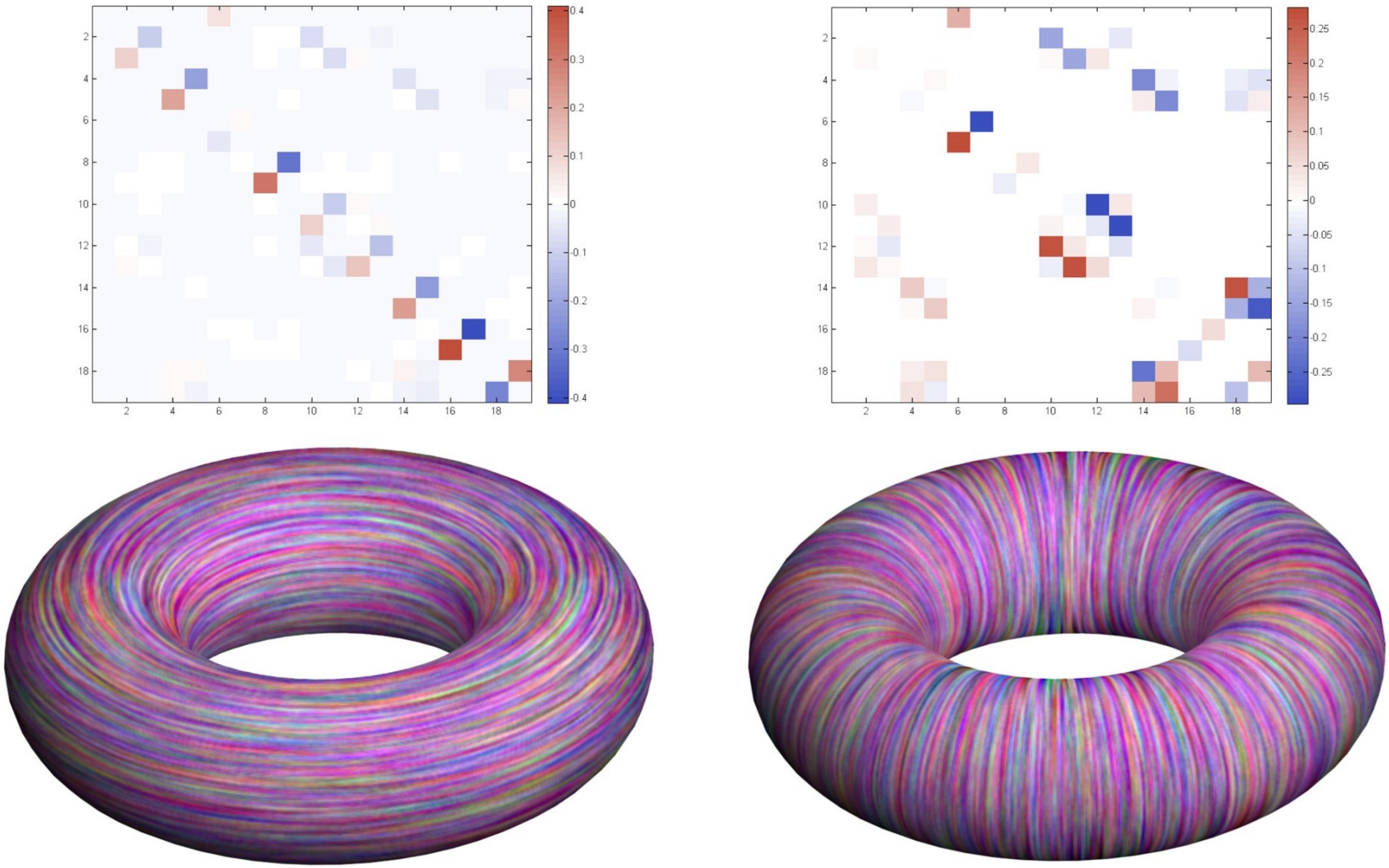
$$g = D_V(f) \quad \Leftrightarrow \quad G = DF$$

$$\begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ \cdots \\ \\ \end{pmatrix} = \left(\; D \;\right) \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ \cdots \\ \\ \end{pmatrix}$$



$$G \qquad\qquad D \qquad\qquad F$$

VFs & FVFs

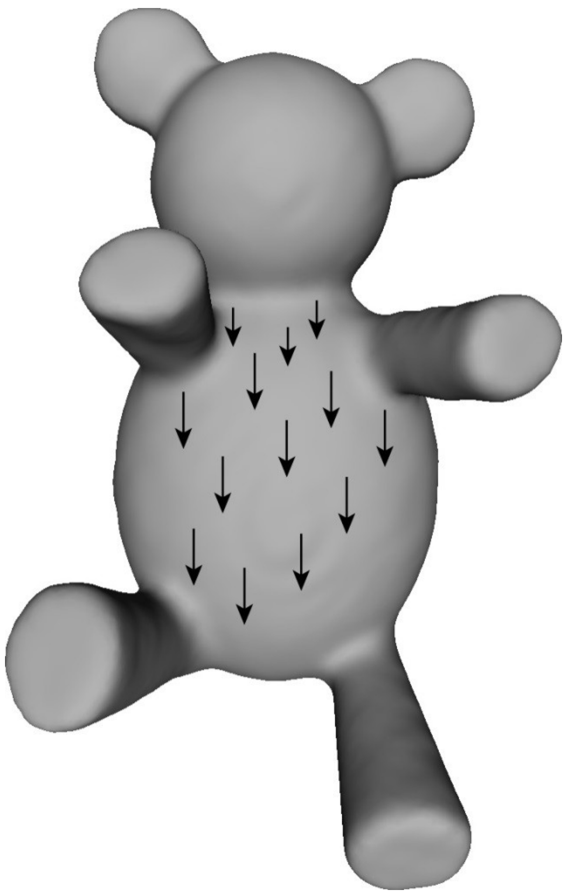

2 4 6 8 10 12 14 16 18
0.4 0.3 0.2 0.1 0 -0.1 -0.2 -0.3 -0.4
0.25 0.2 0.15 0.1 0.05 0 -0.05 -0.1 -0.15 -0.2 -0.25

What are Operators Good For?

- Composition
- Algebraic properties
- Spectral decomposition

What is a Vector Field (VF)?

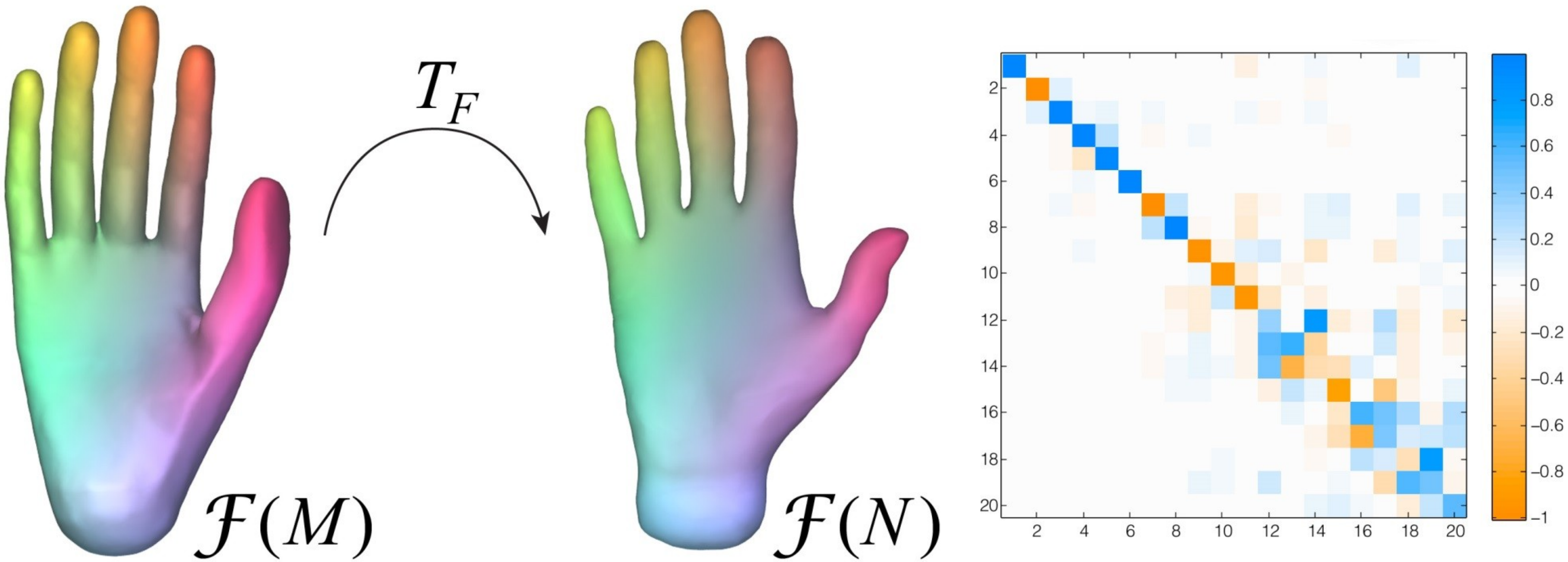
Vector field V



Flow ϕ_V^t

Relation to Functional Maps

Given a pair of shapes and a map $T: N \to M$,



The **Functional Map** [OBCS*12] of T is defined by:

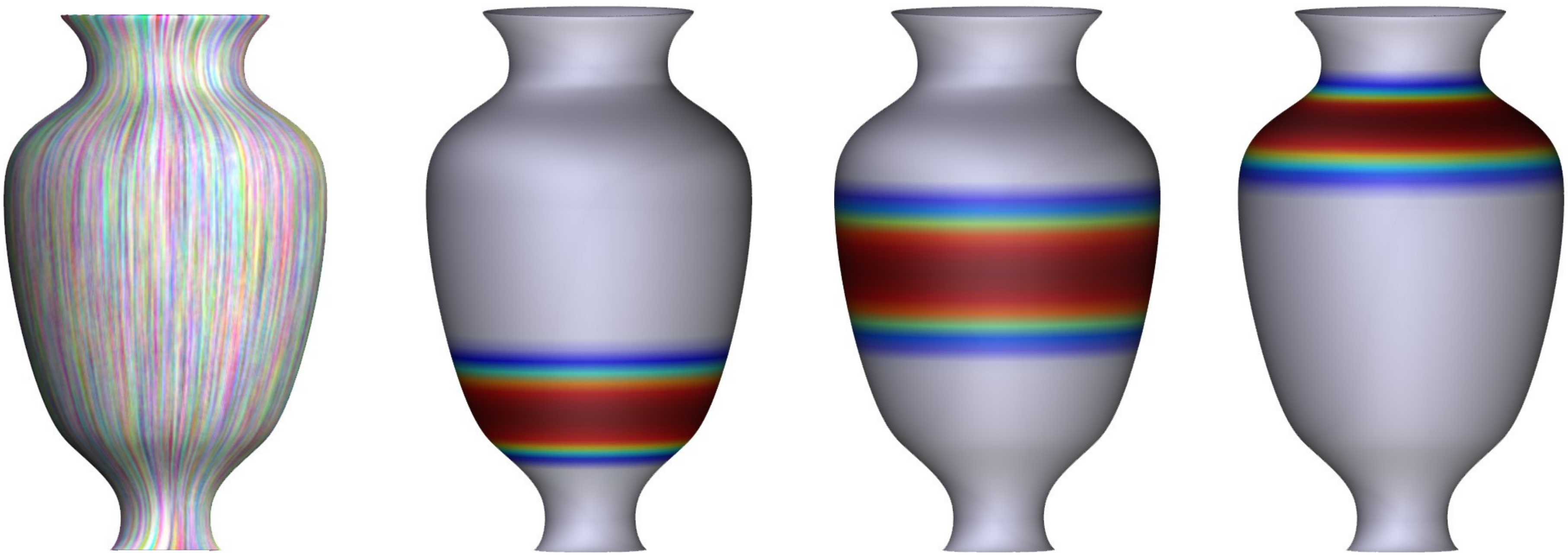
$$g = T_F(f) = f \circ T \quad f \in \mathcal{F}(M), g \in \mathcal{F}(N).$$

Relation to Functional Maps

Flowing a Function

The **flow** $\boldsymbol{\phi}_V^t$ is a self-map and its functional map T_F^t is:

$$T_F^t = \exp(tD_V)$$



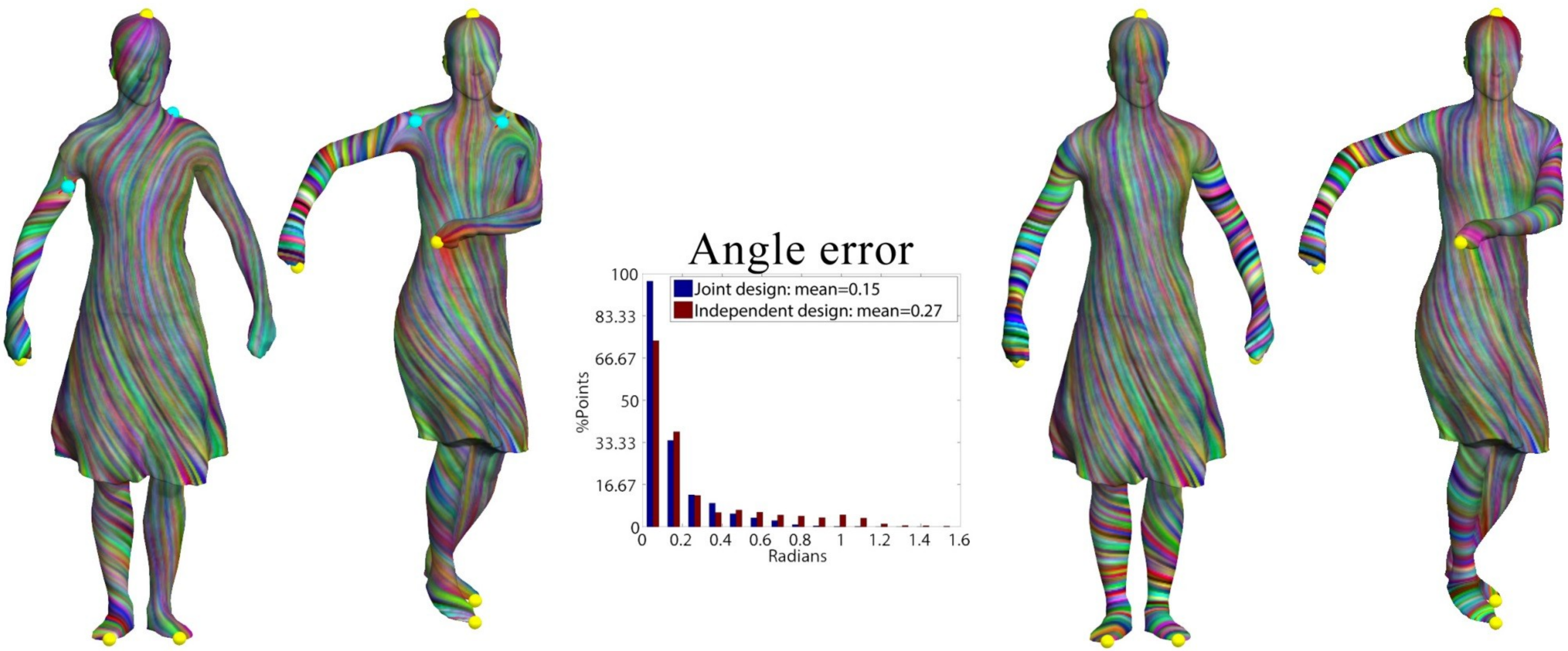
V $\quad t_0 \quad t_1 \quad t_2$

Relation to Functional Maps

Vector Field Transportation

Using a bijective map $T: N \rightarrow M$ we can **transport** VFs:

$$D_{V_2} = T_F \circ D_{V_1} \circ (T_F)^{-1}$$



Independent design

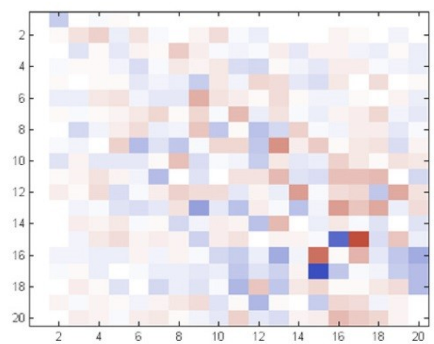
Joint design

Designing FVFs

- **<u>Problem</u>**: Not all matrices are FVFs!

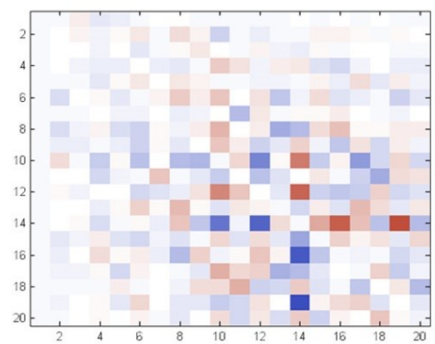
- Solution: Use a basis for the tangent vector fields

$$D_V = a_1 *$$



D_{ψ_1}

$$+ a_2 *$$



D_{ψ_2}

$$+ \ldots$$

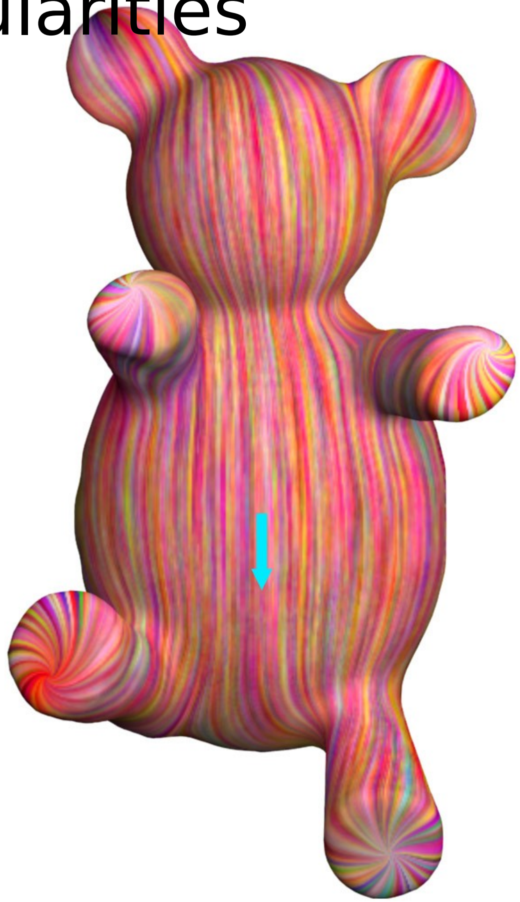
Vector Field Design

- $\boldsymbol{V} = \sum \boldsymbol{a}_i \boldsymbol{\psi}_i \iff \boldsymbol{D}_V = \sum \boldsymbol{a}_i \boldsymbol{D}_{\psi_i}$
- In practice we optimize for $(\boldsymbol{a}_i)$
- We can prescribe:
  - Low-level constraints, e.g. singularities
  - High-level constraints, e.g. symmetry
- We solve a **linear system** of equations:

$$\boldsymbol{W}\boldsymbol{a} = \boldsymbol{c}$$

Directional Constraints

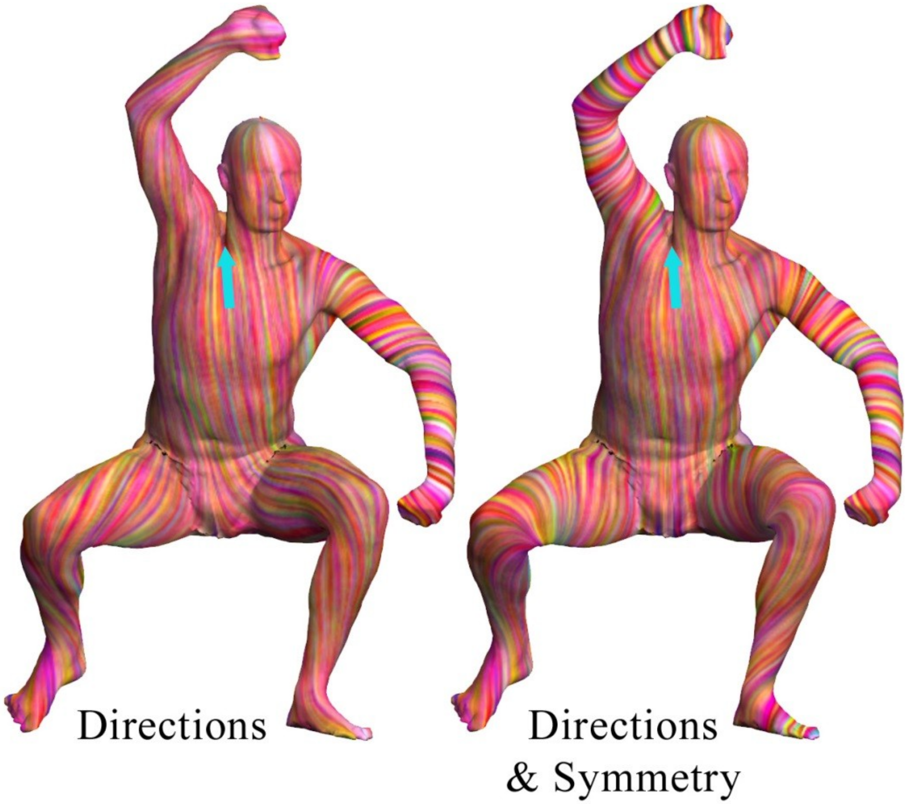
Low-level constraints: Directions and singularities



Symmetric VFs

Symmetric VFs are easily generated using a self-map S:

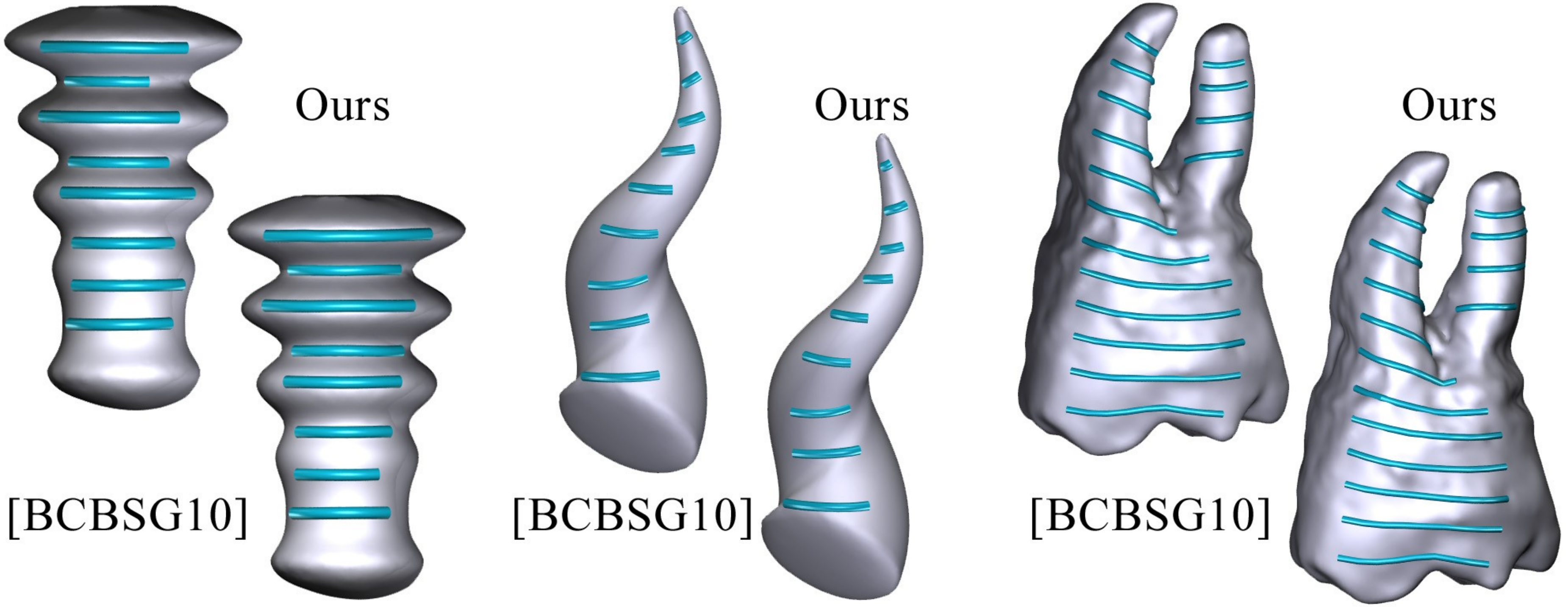
$$|\boldsymbol{D_V} \circ \boldsymbol{S_F} - \boldsymbol{S_F} \circ \boldsymbol{D_V}| = \boldsymbol{0}$$



KVFs

- VF with continuous isometric flows is **Killing**.
- A high-level linear constraint:

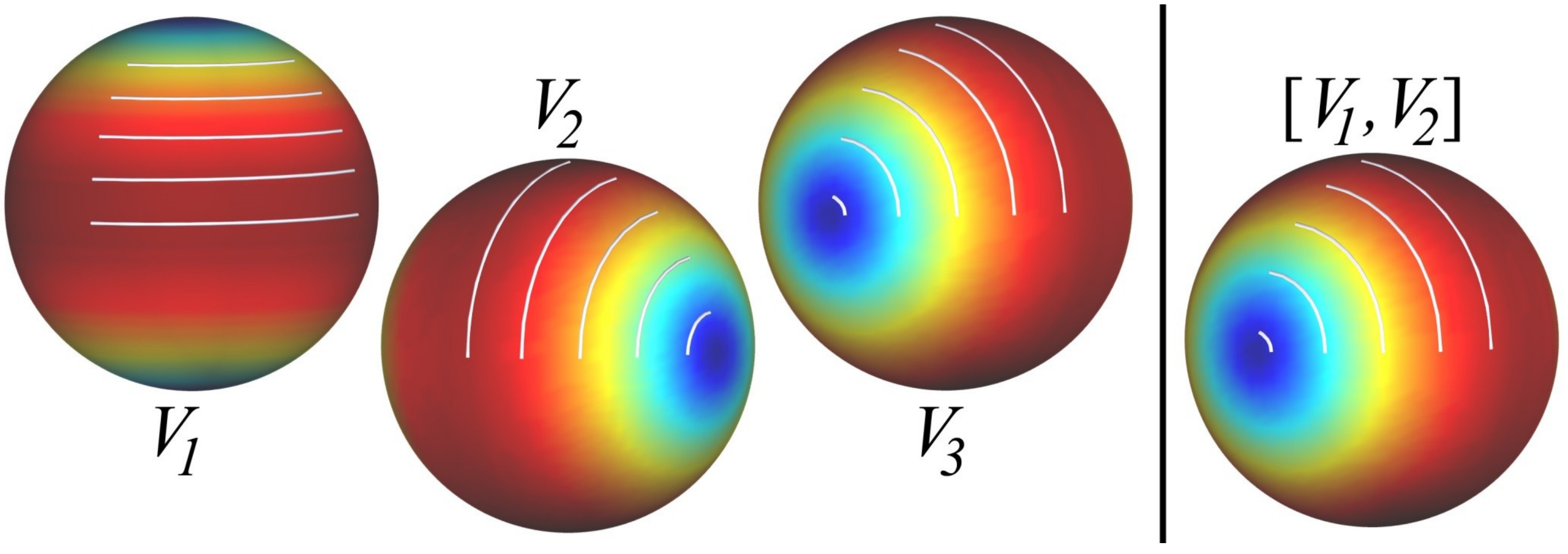
$$|\boldsymbol{D_V} \circ \boldsymbol{L} - \boldsymbol{L} \circ \boldsymbol{D_V}| = \mathbf{0}$$



The Lie Bracket

•

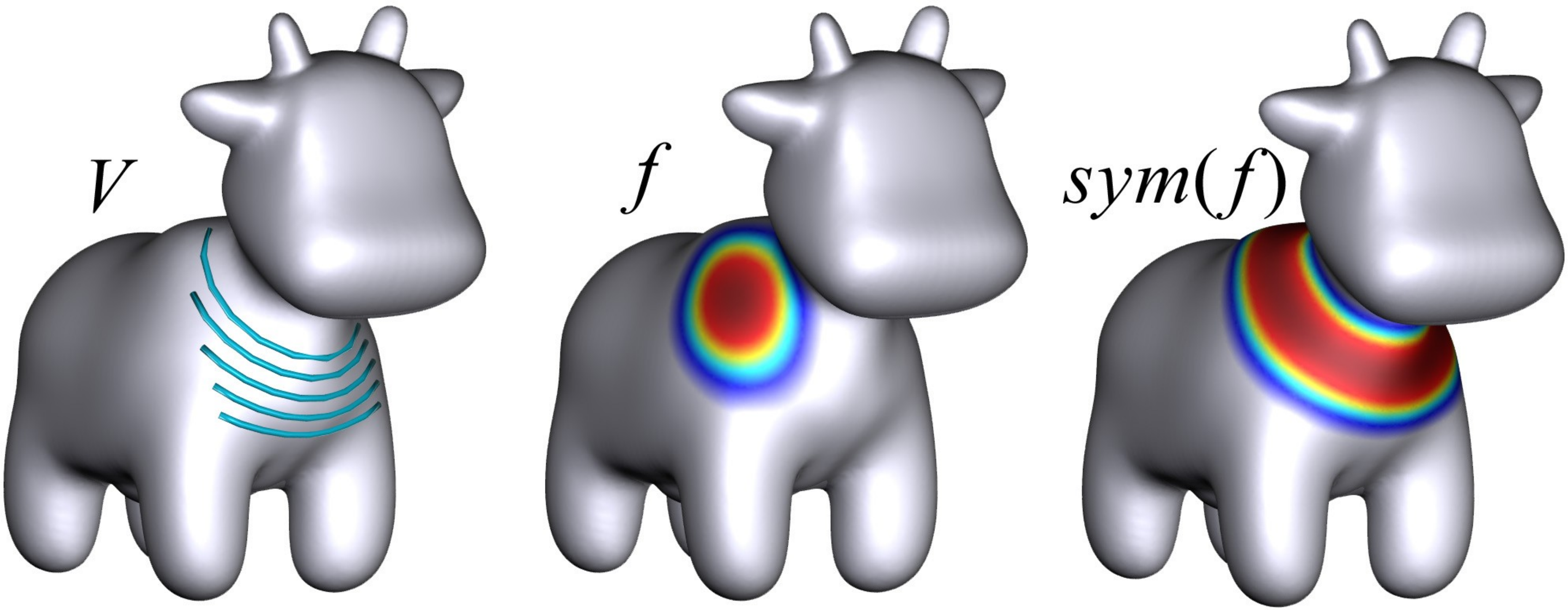
$$[V, U] = D_V D_U - D_U D_V$$



Function Symmetrization

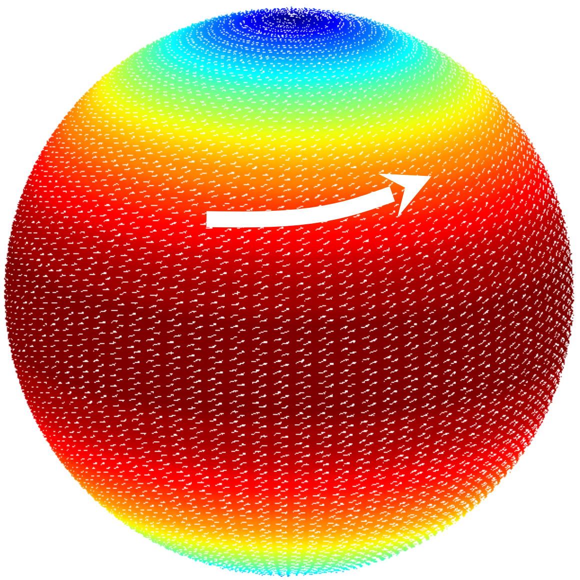
$\mathbf{ker}(\boldsymbol{D_V})$ of a KVF $\boldsymbol{V}$ holds a basis for the **symmetric functions**:

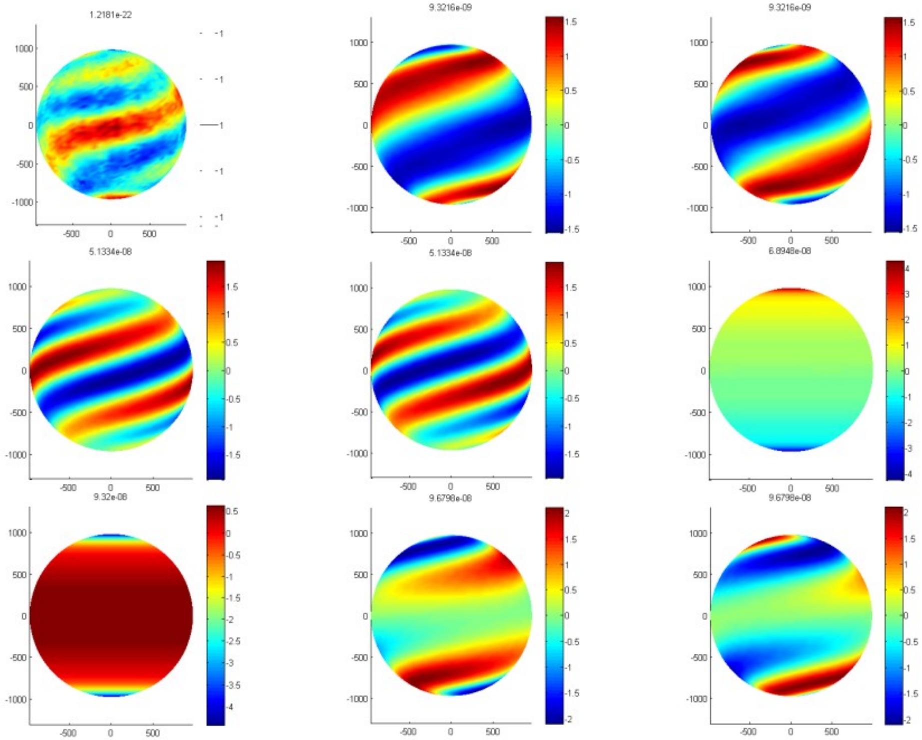
functions:



$\mathbf{sym}(\cdot)$ is a projection of $\boldsymbol{f}$ onto $ker(D_V)$.

Detecting Singularities





Future Work

- Covariant derivative of vector fields
- Vector field simplification
- Joint design on multiple shapes
- Align directions with feature lines
- Design conformal Killing vector fields
- Mesh parameterization using FVFs
- Representation of N-RoSy fields
- …

Conclusions

- A representation of tangent vector fields as operators
- A powerful toolbox:
  - Multiple constraints into a linear system
  - Relation between vector fields and mappings
- Representation using operators.. What's next?