POLAR DECOMPOSITION OF SQUARE MATRICES

SHIZUO KAJI (YAMAGUCHI UNIVERSITY / JST PRESTO)

「情報セキュリティにおける数学的方法とその実践」
2016年12月21日　北海道大学

WHAT IS POLAR DECOMPOSITION?

$z \in \mathbb{C}^{\times}$ can be decomposed uniquely as $z = ru$ where $r > 0, |u| = 1$

$$r \in \mathbb{R}_{>0} \simeq \mathbb{R}, \quad u \in S^1$$

Similarly,

$g \in G$: Lie group can be decomposed uniquely as

$g = VU$ where

$V \in \mathbb{R}^n$: contractible

$U \in K_G$: maximal compact

POLAR DECOMPOSITION OF A MATRIX

We focus on G=GL(n, R): the group of real nxn-invertible matrices

V: positive definite ⇔ $x^T V x > 0$ for any $x \neq 0$

$A = VU$ where $V \in SPD(n)$: symmetric positive definite matrix

$U \in O(n)$: orthogonal matrix

$U^T U = E$

PROPERTIES OF THE ORTHOGONAL COMPONENT

A,B ∈ GL(n, R) $AB^T = VU$: Polar decomposition

Theorem

$$|A - UB|_F \leq |A - U'B|_F \leq |A + UB|_F, \quad \forall U' \in O(n)$$

where $|C|_F := \sqrt{\mathrm{tr}(CC^T)} = \sqrt{\sum_{i,j} c_{i,j}^2}$ is the Frobenius norm

The U is the best approximating orthogonal transform

PROOF

- It is enough to look at the case when B=E.
- Let's minimise $$|A-U|_F^2 = \mathrm{tr}(A-U)(A-U)^T$$ under $$UU^T = E$$
- plug in a Lagrange multiplier Λ, which we can take as symmetric
$$(\star)\quad \mathrm{tr}\left((A-U)(A-U)^T + \Lambda(UU^T - E)\right)$$
- By differentiating by U, $$-2(A-U) + 2\Lambda U = 0$$
- and so $A = (E+\Lambda)U$
- $V = E + \Lambda$ is SPD since it is the second derivative of (☆)
- Assertion follows from the uniqueness of the polar decomposition

PROPERTIES OF THE SPD COMPONENT

$$A = VU \quad \begin{array}{r} V \in SPD(n): \\ U \in O(n): \end{array}$$

- The eigenvalues of V are the *singular values* of A
- $V = \sqrt{AA^T}$
- $V^2 = AA^T$

  is called the covariance matrix (up to a scalar depending on the convention)

  when the mean of columns is zero

  It is important in data analysis as the matrix amalgamates correlation among rows

TWO REMARKS

REMARK: FOR SINGULAR/NON-SQUARE MATRICES

- First, apply the QR-decomposition (Gram-Schmidt) to obtain

$$A = A' N$$

  where A' is a regular upper-triangular matrix
  and N a matrix with orthonormal rows.
- Then, for the polar decomposition $A' = VU$,
  we obtain $A = V(UN)$,
  which we regard as the polar decomposition of A

REMARK: LEFT AND RIGHT POLAR DECOMPOSITIONS

- The order of the two factors matter:

$$A = VU = UV' \qquad V, V' \in SPD(n), U \in O(n)$$

- the O(n) component is same
- but SPD(n) components may differ
- A is normal ⇔ V=V’

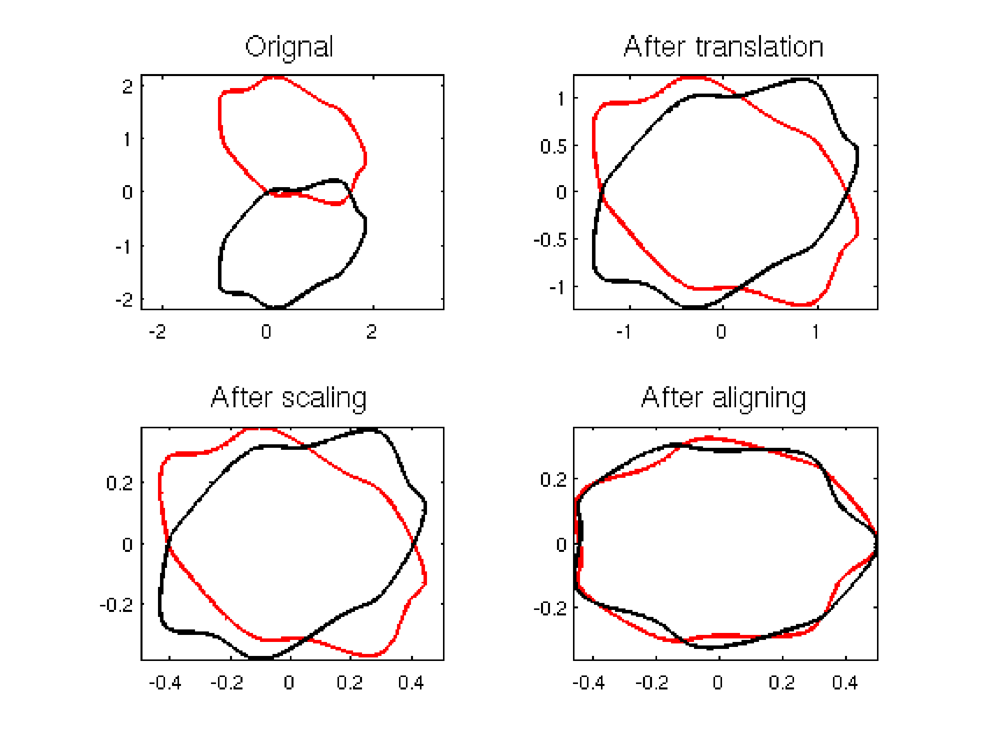
APPLICATIONS IN DATA ANALYSIS

WHITENING

- $A \in R^{n\times m}$: data (each column represents a sample and each row a random variable)
- Correlation between variables is amalgamated in $A A^T$
- *Whitening* is a linear transform (change of basis) W such that the rows of WA have no correlation (white noise); that is, $(WA)(WA)^T = E$
- If A=VU is the polar decomposition,

$$(V^{-1}A)(V^{-1}A)^T = V^{-1}AA^TV^{-1} = E$$

DATA ALIGNMENT



Given two shapes (vector data sets), align them using scaling and rotation.

An important pre-process for coordinate free data analysis

DATA ALIGNMENT (PROCRUSTES PROBLEM)

$$A, B \in \mathbb{R}^{n\times m}$$

We want to find U ∈ O(n) and c ∈ R such that $|A - cU B|_F$ is minimised

Thm First, translate P and Q so that the means of the row vectors become zero.

Decompose $AB^T = VU$

Then, U and $c = \mathrm{tr}(V)/\mathrm{tr}(BB^T)$ gives the solution

DISTANCE BETWEEN POINT CLOUDS

Measure how different two data sets (indexed and of a fixed size) A and B are up to scaling and rotation.

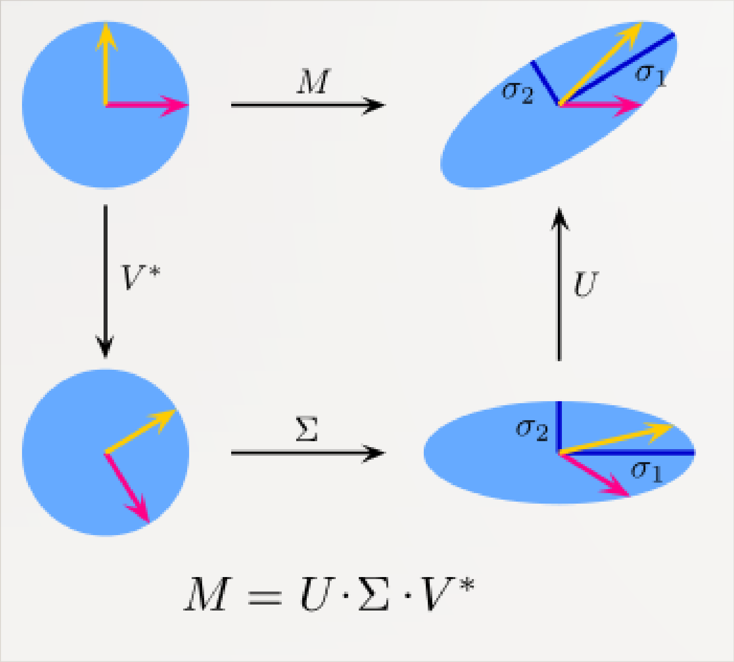
$$A, B \in \mathbb{R}^{n \times m}$$

$$\min_{c,U} |A - cUB|_F$$

serves as a good distance between point clouds.
It can be computed by the previous theorem.

SINGULAR VALUE DECOMPOSITION (SVD)

- SVD of A $A = P\Sigma Q^T$ $P, Q \in O(n)$



$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$$

$\sigma_i > 0$ singular values

$$\Longleftrightarrow$$

$$A = (P\Sigma P^T)(PQ^T)$$

is the polar decomposition

APPLICATIONS OF SVD

$$A = P\Sigma Q^T$$

- pseudo inverse $A^+ = P\Sigma^+ Q^T$

Ax=b => A^+b is the least norm solution when there is a solution
$x=A^+b$ minimises $|Ax-b|^2$ when there is no solution
(least square solution)

- matrix approximation by low rank matrix: (equivalent to PCA)

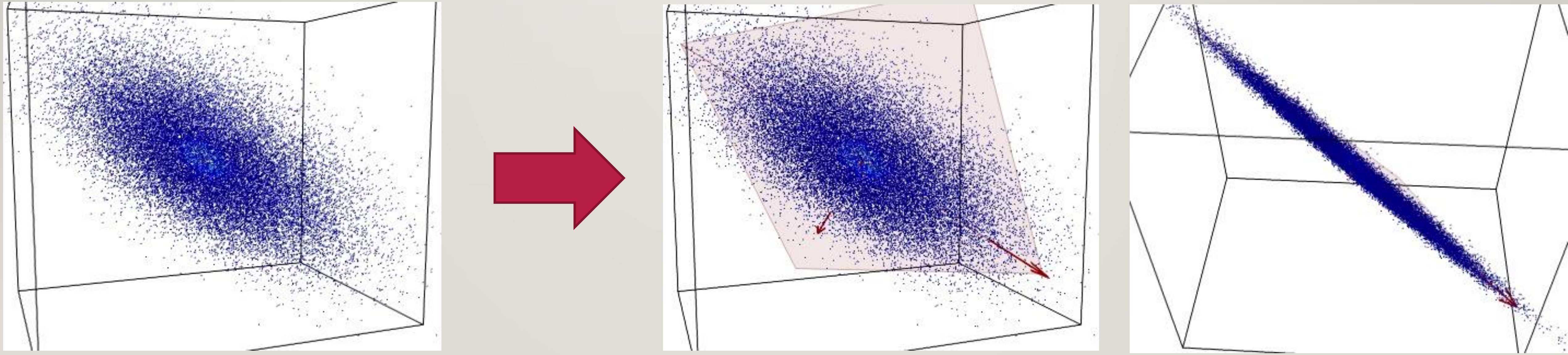
setting lower singular values to zero, one obtains the best approximation in terms of the Frobenius norm

AQ=PΣ gives the components

PRINCIPAL COMPONENT ANALYSIS (PCA)

Dimension reduction technique

"Find a linear subspace of dim=n such that
the projected data loses as little as possible information"



COMPUTATION OF POLAR DECOMPOSITION

BY SVD

- SVD of A $$A = P\Sigma Q^T \qquad P, Q \in O(n)$$
$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$$

$\Longleftrightarrow$ $A = (P\Sigma P^T)(PQ^T)$ is the polar decomposition

- PROS: numerically stable (many good algorithms for SVD)
- CONS: SVD is expensive and not always available

DIAGONALISATION

$$A = VU \quad \begin{array}{r} V \in SPD(n) : \\ U \in O(n) : \end{array}$$

$$V = \sqrt{AA^T}$$

can be computed by diagonalising the symmetric matrix A A^T:

$$AA^T = Q\Sigma Q^T$$

All the diagonal entries of Σ (singular values) are positive, so take their square roots to have

$$V = \sqrt{AA^T} = Q\sqrt{\Sigma}Q^T, \quad U = V^{-1}A$$

→ diagonalisation is expensive

HIGHAM'S ITERATIVE METHOD

Thm
$$A_0 = A$$
$$A_{k+1} = (A_k + A_k^{-1})/2$$
$$\lim_{k\to\infty} A_k = U$$

computes the orthogonal factor of A = UV

and converges quadratically.

(can be accelerated by scaling)

Then, V = A U^T

Proof: When A is diagonal, the iteration converges to E up to sign.

($x=(x+x^{-1})/2 \Rightarrow x^2=1$)

So $A=P\Sigma Q^T$ converges to $PQ^T=U$

KAJI-OCHIAI'S METHOD

Recall the Cartan decomposition:

$z \in \mathbb{C}^\times$ can be decomposed as $z = e^s e^{i\theta}$ where $s \in \mathbb{R} = L(\mathbb{R}_{>0})$

$i\theta \in i\mathbb{R} = L(S^1)$

Similarly for A∈GL(n, R)

$$A = \exp(X)\exp(Y)$$ where X: symmetric

$$Y \in \mathfrak{o}(n) := \{Y \mid Y^T = -Y\}$$

KAJI-OCHIAI'S METHOD

$$A = \exp(X)\exp(Y)$$

$$\boxed{A = VU \quad \begin{array}{r} V \in SPD(n) : \\ U \in O(n) : \end{array}}$$

$\Rightarrow$

$$V = \exp(X)$$
$$U = \exp(-X)A$$

where $X = \log(AA^T)/2$

OK, but how can we compute log and exp?

KAJI-OCHIAI'S METHOD

- Divide

$$\exp(X) = 1 + X + \frac{X^2}{2!} + \cdots$$

  by the characteristic polynomial (Carley-Hamilton) to obtain an (n-1)-degree polynomial f
- The coefficients of f are functions of eigenvalues of X.
- Same is true for log (and any conjugate invariant function)

EXPONENTIAL OF A SYMMETRIC MATRIX

Thm For a symmetric 3x3-matrix X,
(a similar formula holds for any size)

$$\exp(X) = c_1 I + c_2 X + c_3 X^2$$

where $$\begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 \\ 1 & \lambda_2 & \lambda_2^2 \\ 1 & \lambda_3 & \lambda_3^2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} e^{\lambda_1} \\ e^{\lambda_2} \\ e^{\lambda_3} \end{pmatrix}$$

λ_i: eigenvalues of X

CF. EXPONENTIAL OF AN ANTI-SYMMETRIC MATRIX

Rodrigues' theorem

For X: 3x3 satisfying $X^T = -X$

$$\exp(X) = I_3 + \frac{\sin\theta}{\theta}X + \frac{1-\cos\theta}{\theta^2}X^2 = I_3 + \mathrm{sinc}(\theta)X + \frac{1}{2}\left(\mathrm{sinc}\frac{\theta}{2}\right)^2 X^2$$

where $\theta = \sqrt{\dfrac{\mathrm{tr}({}^tXX)}{2}}$

Our argument can be used to prove this famous formula and its generalisation

COMPARISON OF COMPUTATION METHODS

- SVD: Reliable, but slow.
  Directly works for non-singular matrices
- Higham: Fast and widely used
- Kaji-Ochiai: Very fast when computing a lot of polar decompositions of fixed size matrices.
  Computes the Cartan decomposition as well.
  Numerically unstable for near singular matrices

CODES

MIT licensed C++ codes are available at

https://github.com/shizuo-kaji/AffineLib

which contain all four algorithms and more

APPLICATION IN GRAPHICS

Shape/Motion

- Analysis
- Recognition
- Deformation

SHAPE ANALYSIS

Find "distorted" parts

Piecewise linear map

$f: M_1 \rightarrow M_2$

=> $f|_T = VU$ polar decomp

and use $|V-E|$ as an indicator

SHAPE MATCHING

Very fast "simulation" of an elastic body

$M(t) = \{ x(t) \in R^3 \}$: elastic body

$F(t)$: external force

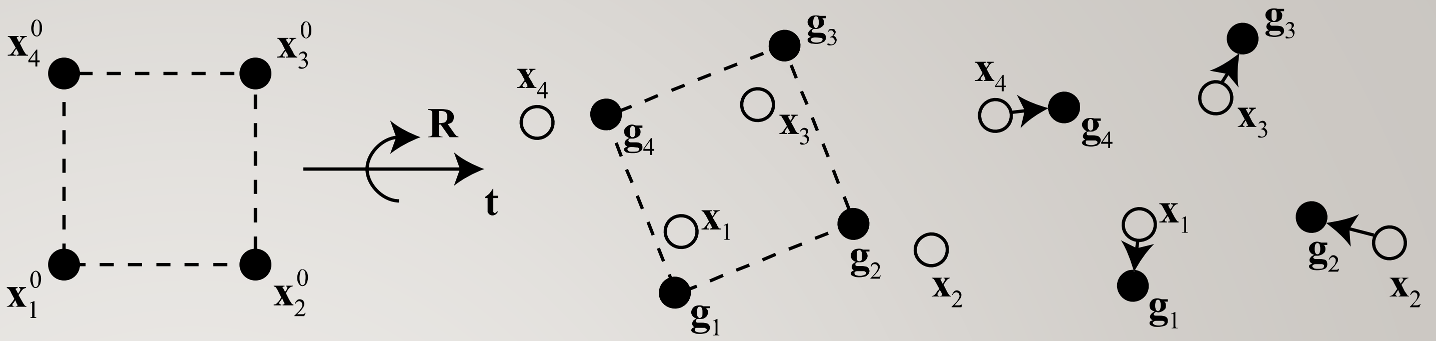
geometric constraints

$\Rightarrow$ $M(t+\Delta t)$

Video
https://www.youtube.com/watch?v=CClwiC37kks

Muller et al.
Meshless Deformations Based on Shape Matching
SIGGRAPH2005

SHAPE MATCHING

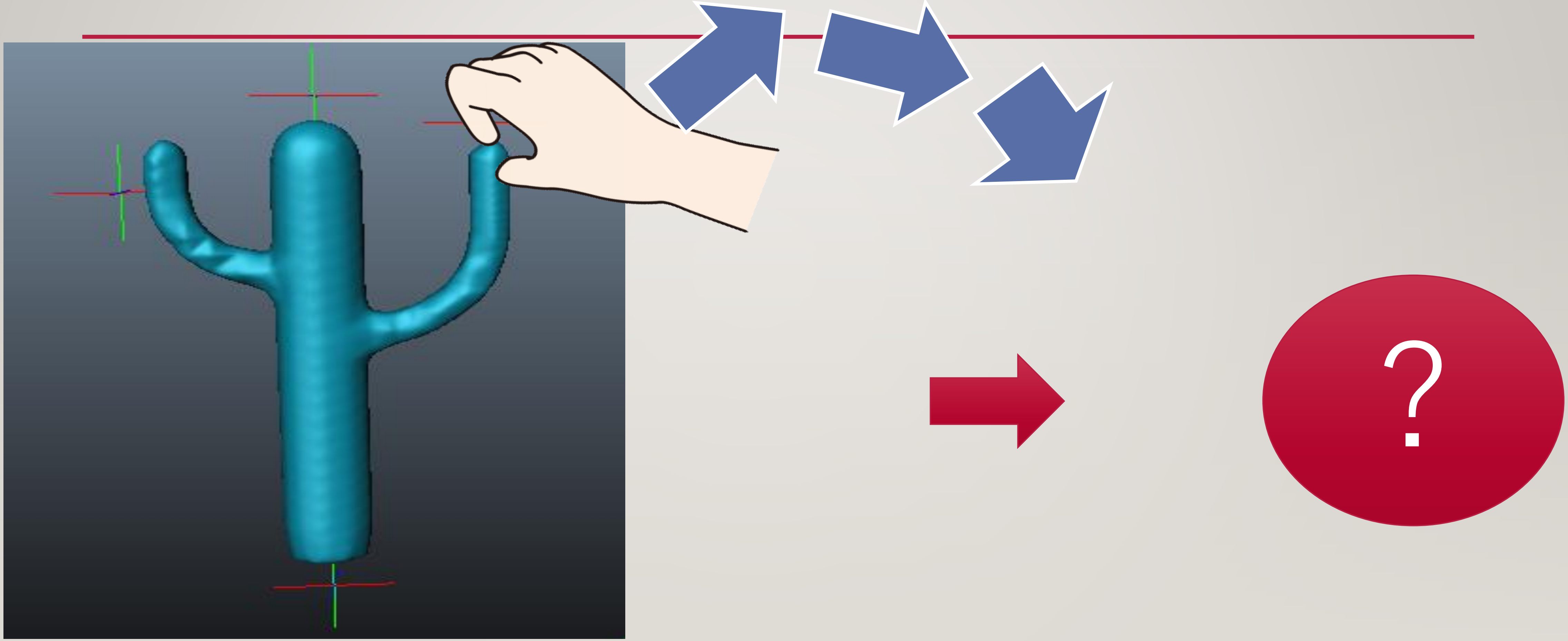


- Find $U \in SO(n)$ which minimises $|U\,M(0) - M(t)|$ by Polar decomp
- Define "elasticity" force at a point x by $c(U\,x(0) - x(t))$ for some constant c
- Update the speed of x by

$$x'(t+\Delta t) = d(x'(t) + F(t) + c(U\,x'(0) - x(t)))$$

where $d < 1.0$ is the damping coefficient

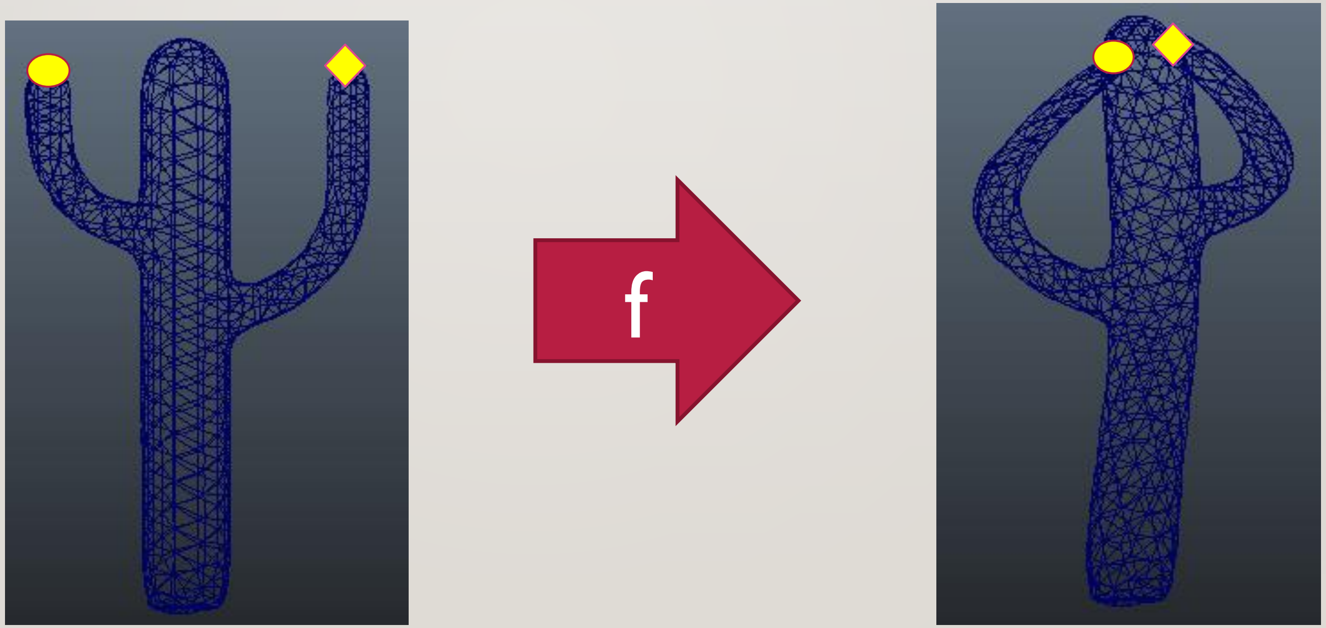
SHAPE MODELLING

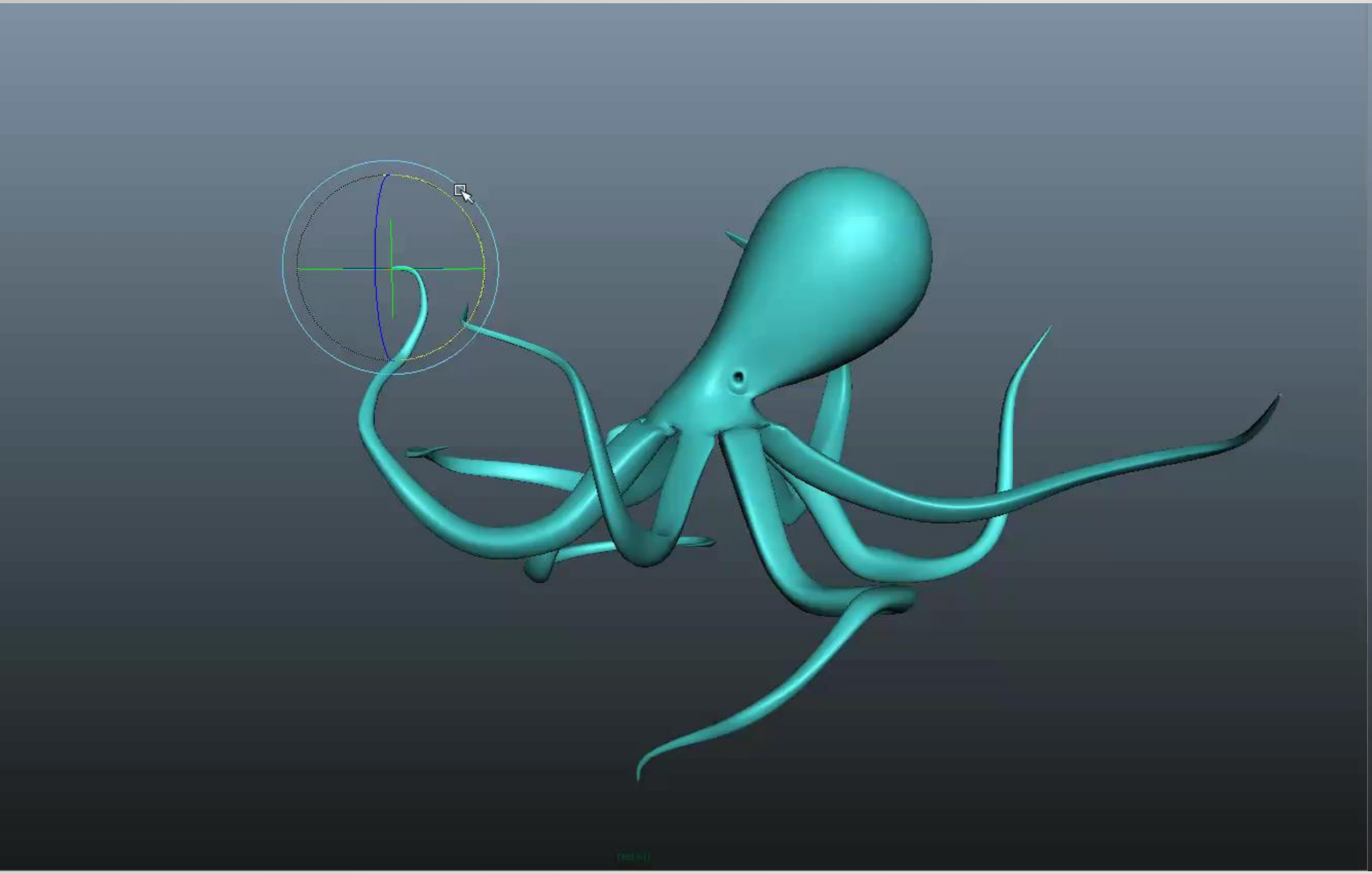


shape + user interaction => deformed shape

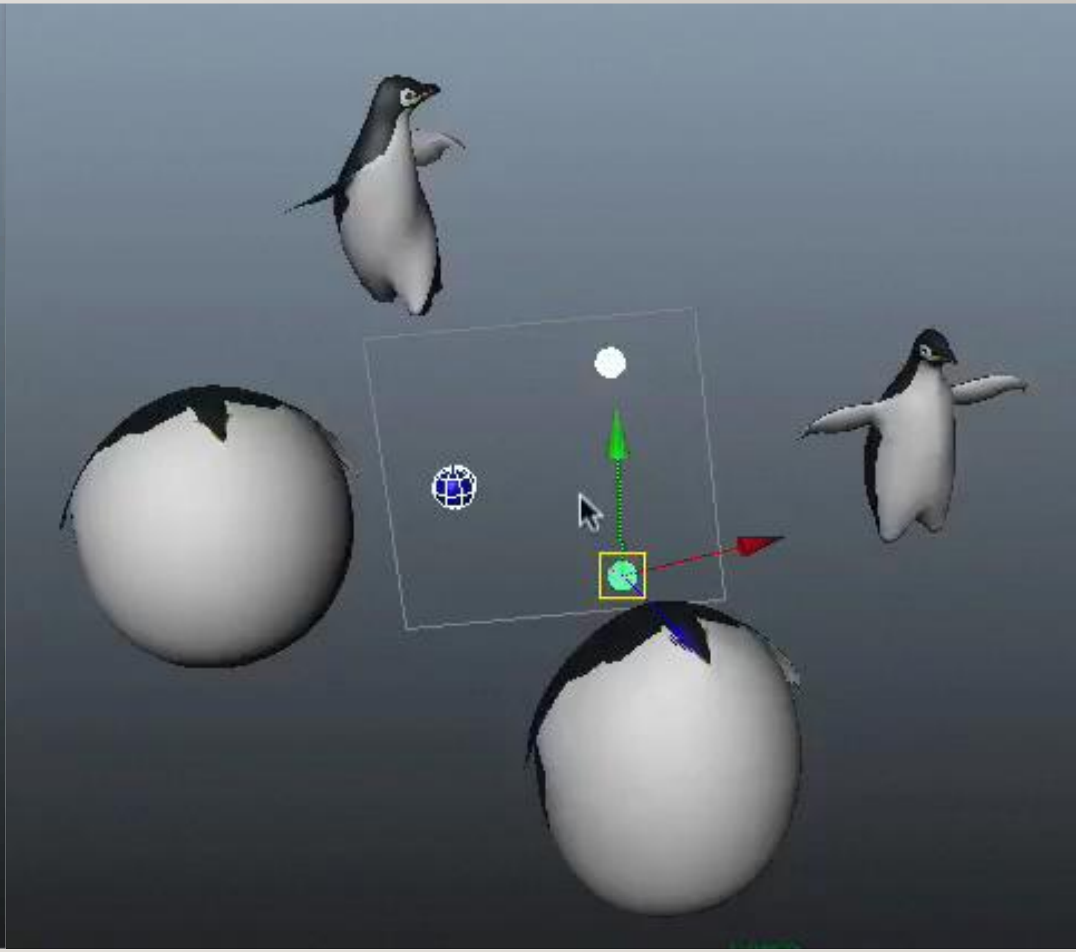
SHAPE MODELLING

Given a shape M and constraints, find a map $f: M \rightarrow R^3$





Kaji-Liu2014



Kaji2015

FIELD OF TRANSFORMATION

- First, construct a field A: M → GL(3; R) by solving

  the Laplace equations ΔU = 0, ΔV = 0   A=VU

  under A(some points) = constraints

- Then, find f which minimises

$$\int_M |\nabla f - A|^2 dM$$

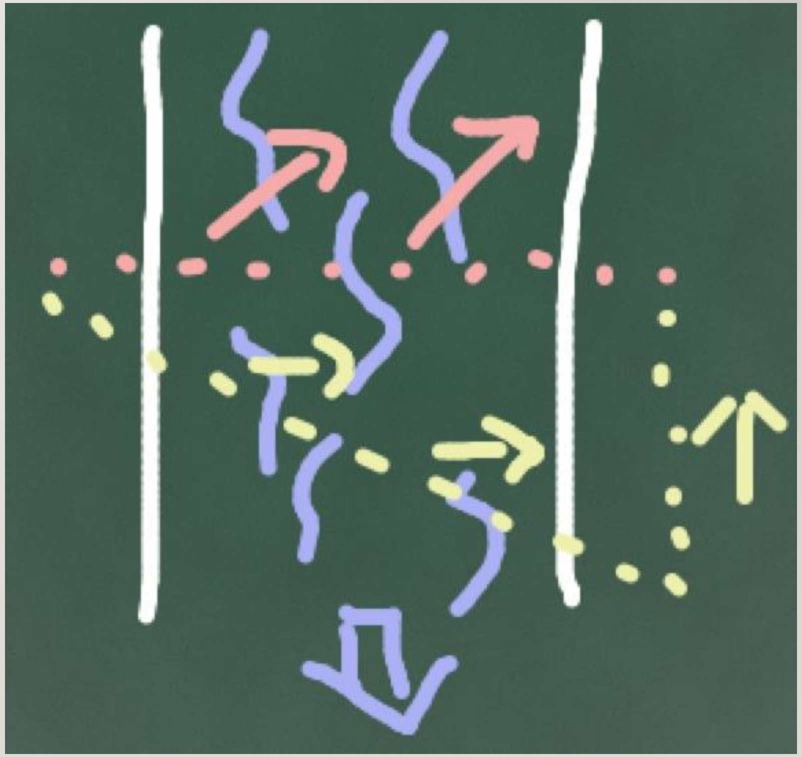
The solution is given by the curl free part of the Helmholtz-Hodge decomposition of A

WHY DECOMPOSE?

Mathematical reason

- We want an "easy" presentation of GL(n; R)
- Let's use Lie algebra
- The problem is that Lie correspondence is not surjective since GL(n; R) is not compact
- But decomposed factors are mapped surjectively by exponential

Intuitive/cognitive reason



Rotation doesn't cost

DEMO WITH LEAP MOTION

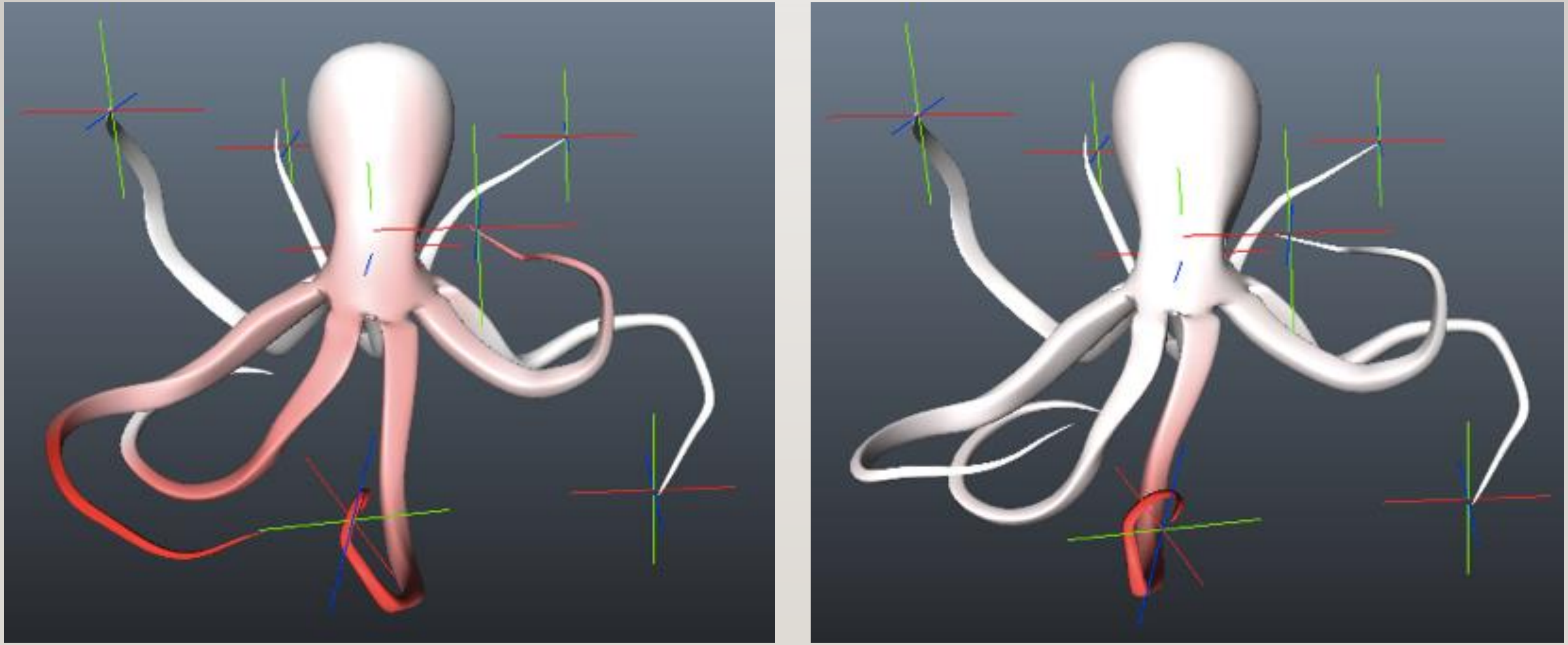
DISCRETE DIFFERENTIAL GEOMETRY

DDG discusses how to define Δ, ∇, $\int$ for discrete objects
and is getting popular in data sciences

Mantra:

- (meaningful) Big data in a high dimensional Euclidean space should lie on a manifold
  (dimension reduction)
- Geometry of the manifold tells a lot (curvature / intrinsic metric)
- Much of geometry is captured by the Laplacian

HARMONIC FIELD – Δ KNOWS THE GEOMETRY



THANK YOU!