# Bones, Teeth and Animation

Ingrid Daubechies

May 10, 2016

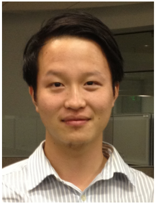
Green Family Lecture, IPAM, UCLA

# Collaborators

Rima Alaifari
ETH Zürich

Doug Boyer
Duke

Ingrid Daubechies
Duke



Tingran Gao
Duke

Yaron Lipman
Weizmann

Roi Poranne
ETH Zürich

Jesús Puente
J.P. Morgan

Robert Ravier
Duke

# Machine Learning, Fibre Bundles and Biological Morphology

Ingrid Daubechies    Tingran Gao

Department of Mathematics
Duke University

Feb 11, 2016

# Animation

- 3-dimensional Animation relies on computer graphics

# Animation

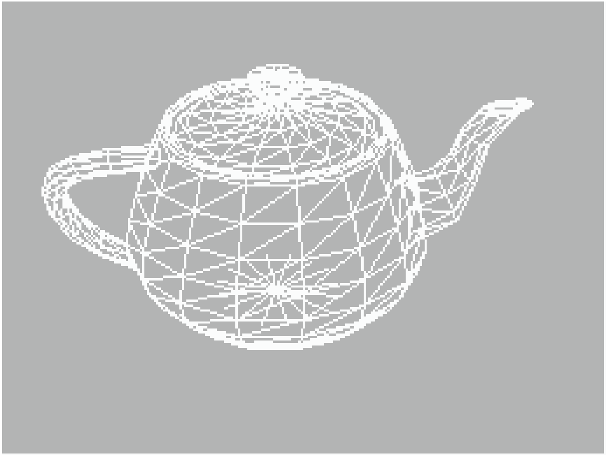
- 3-dimensional Animation relies on computer graphics
- computer graphics uses 3-dimensional mesh models

# 3D meshes in computer graphics

In the beginning was the TEAPOT

# 3D meshes in computer graphics

TEAPOT:



# 3D meshes in computer graphics

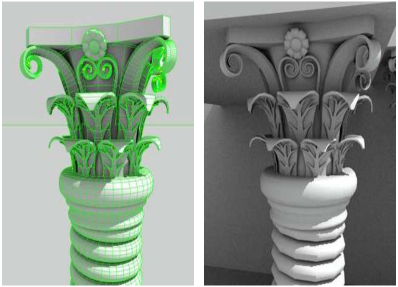
TEAPOT:

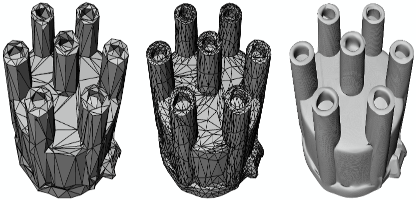
# 3D meshes in computer graphics

3D meshes became much more sophisticated over the years

# 3D meshes in computer graphics

# 3D meshes in computer graphics





# 3D meshes in computer graphics

# 3D meshes in computer graphics

# 3D meshes: not only for design from scratch

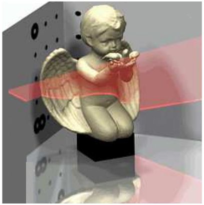
Scan existing objects

# 3D meshes: not only for design from scratch

Scan existing objects

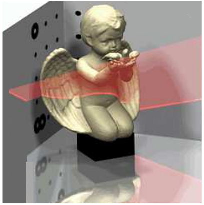
# 3D meshes: not only for design from scratch

Scan existing objects

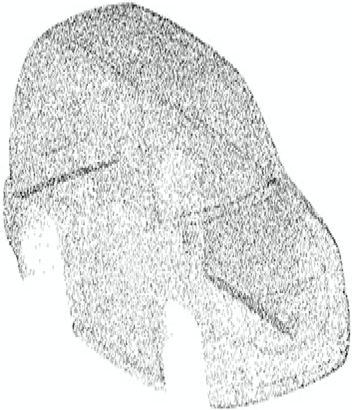
 

# 3D meshes: not only for design from scratch

Scan existing objects

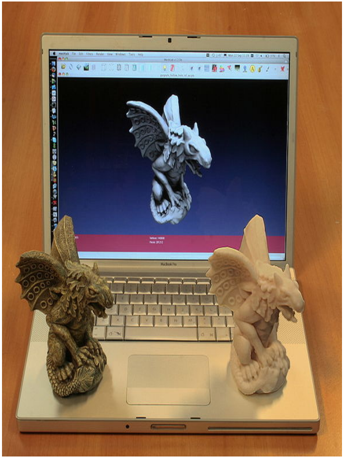


- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
-
-
-
-

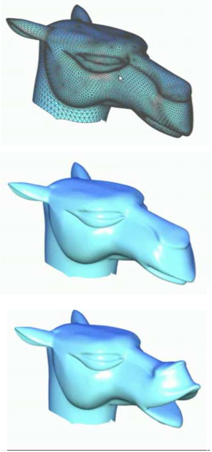


# surfaces from scans

- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
-
-
-
-

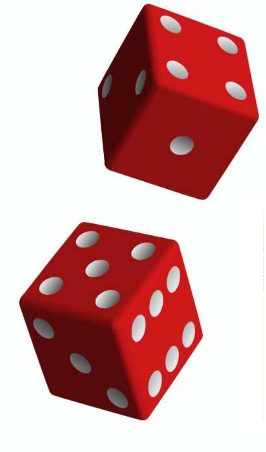


- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
- edit triangulated surfaces
- 
- 
- 



# surfaces from scans

- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
- edit triangulated surfaces
- recognize identical surfaces?
-
-



# surfaces from scans

- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
- edit triangulated surfaces
- recognize identical surfaces?
- or deformations of each other?
-

# surfaces from scans

- point cloud $\longrightarrow$ triangulation (Delauney triangulation)
- edit triangulated surfaces
- recognize identical surfaces?
- or deformations of each other?
- quantify difference?


# Reference points

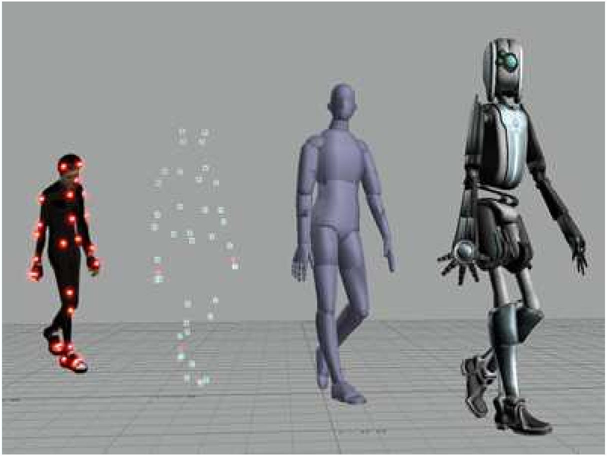
Animating “humanoid” characters requires reference points

# Reference points

Animating “humanoid” characters requires reference points

# Reference points

Animating "humanoid" characters requires reference points



# Summary of “problems”

We need to be able to:

- recognize when two point clouds correspond to the same surface or to two similar surfaces
- quantify how different two surfaces are from each other (or how similar to each other)
- find correspondence points for similar surfaces

# It all started with a conversation with biologists....

Doug Boyer

Jukka Jernvall

More Precisely: biological morphologists

↓

Study Teeth & Bones of

extant & extinct animals

still live today    fossils

First: project on “complexity” of teeth

First: project on “complexity” of teeth
Then: find automatic way to compute Procrustes distances between surfaces — without landmarks

First: project on “complexity” of teeth
Then: find automatic way to compute Procrustes distances between surfaces — without landmarks

Landmarked Teeth $\longrightarrow$

$$d^2_{Procrustes}\left(S_1, S_2\right) = \min_{R \text{ rigid tr.}} \sum_{j=1}^{J} \left\| R\left(x_j\right) - y_j \right\|^2$$

First: project on "complexity" of teeth
Then: find automatic way to compute Procrustes distances between surfaces — without landmarks

Landmarked Teeth $\longrightarrow$

$$d^2_{Procrustes}\left(S_1, S_2\right) = \min_{R \text{ rigid tr.}} \sum_{j=1}^{J} \left\| R\left(x_j\right) - y_j \right\|^2$$

Find way to compute a distance that does as well, for biological purposes, as Procrustes distance, based on expert-placed landmarks, automatically?

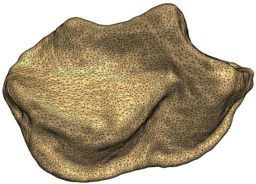
First: project on "complexity" of teeth
Then: find automatic way to compute Procrustes distances between surfaces — without landmarks

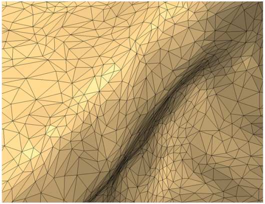
Landmarked Teeth $\longrightarrow$

$$d^2_{Procrustes}(S_1, S_2) = \min_{R \text{ rigid tr.}} \sum_{j=1}^{J} \left\| R(x_j) - y_j \right\|^2$$

Find way to compute a distance that does as well, for biological purposes, as Procrustes distance, based on expert-placed landmarks, automatically?

examples: finely discretized triangulated surfaces

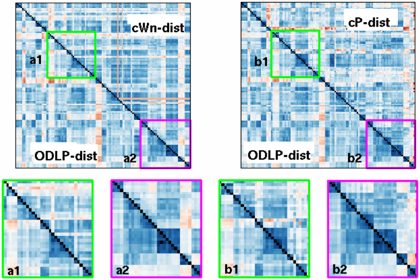


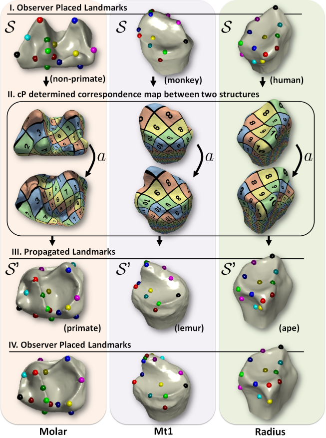


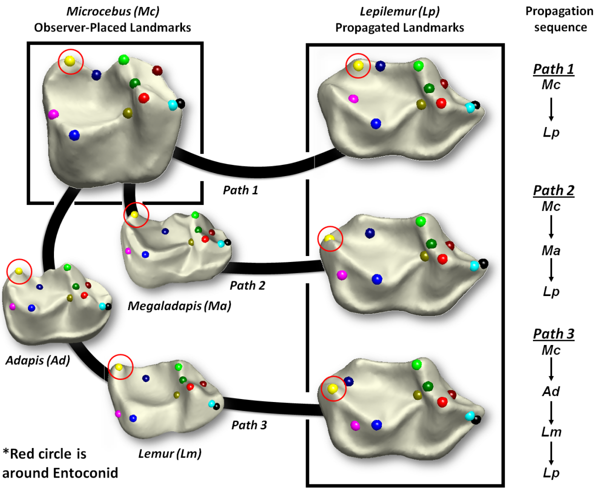
# We defined 2 different distances

$d_{\mathrm{cWn}}(S_1, S_2)$: conformal flattening
comparison of neighborhood geometry
optimal mass transport

$d_{\mathrm{cP}}(S_1, S_2)$: continuous Procrustes distance






Even mistake made by $d_{\mathrm{cP}}$ were similar to biologists' mistakes

small distances between $S_1, S_2 \longrightarrow$ OK maps

larger distances $\longrightarrow$ not OK

# Biologists' "wish list" changed...

... as they learned our language and saw our methods

- mappings more important to them than distances ($\longrightarrow$ discussion of variability in individuals or between species, locally)
- no *holonomy*!

# Biologists' "wish list" changed...

... as they learned our language and saw our methods

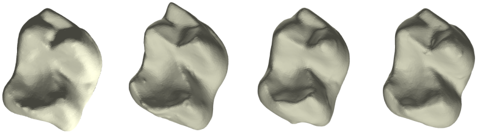
- mappings more important to them than distances ($\longrightarrow$ discussion of variability in individuals or between species, locally)
- no *holonomy*!

Our formulation of problem changed too

Tingran Gao $\longrightarrow$ reformulate as connection on fibre bundle
$+$ horizontal diffusion

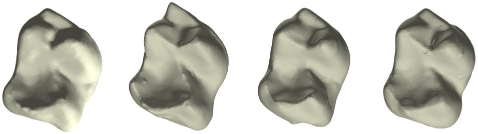
# Even before this...

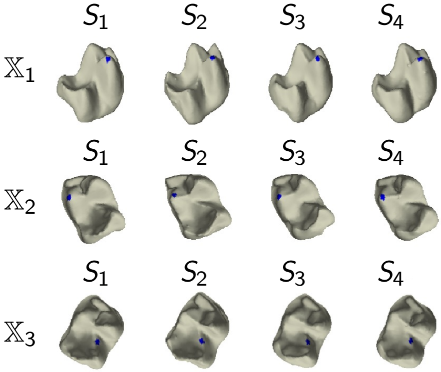
biological content in large concatenated matrix



# Even before this...

biological content in large concatenated matrix





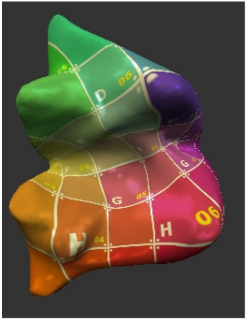

$$
\begin{aligned}
\min \quad & \|M\mathbb{X} - \mathbb{X}\|_2^2 + \lambda \|\mathbb{X}\|_1 \\
\text{s.t.} \quad & \|\mathbb{X}\|_2 = 1.
\end{aligned}
$$

Resulting minimizers $\mathbb{X}$ supported on <u>union</u> of 4 surfaces

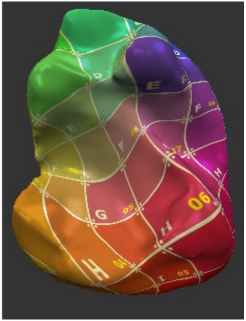
# Use the Information in the Maps!

$$d_{\mathrm{cP}}\left(S_1, S_2\right) = \inf_{\mathcal{C}\in\mathcal{A}(S_1,S_2)} \inf_{R\in\mathbb{E}(3)} \left( \int_{S_1} \| R\left(x\right) - \mathcal{C}\left(x\right) \|^2 \, d\mathrm{vol}_{S_1}\left(x\right) \right)^{\frac{1}{2}}$$

$S_1$



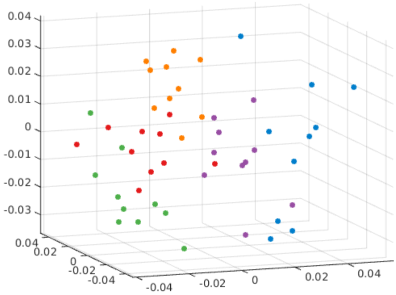
$\xrightarrow[f_{12}]{d_{12}}$



$S_2$

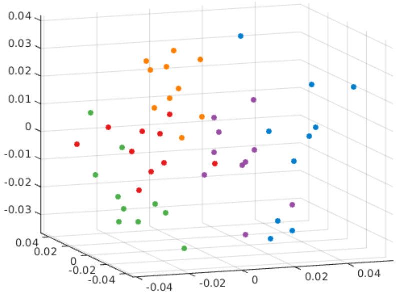
# Learning from Distances

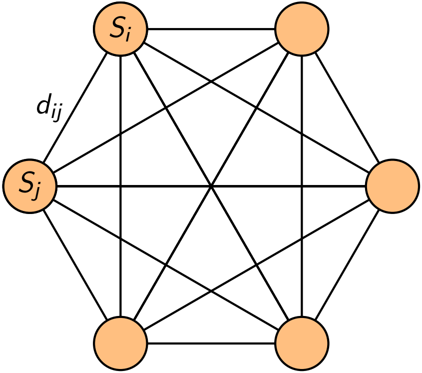
$$\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{pmatrix}$$



# Learning from Distances

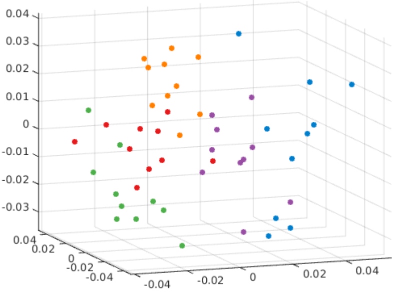
$$\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{pmatrix}$$

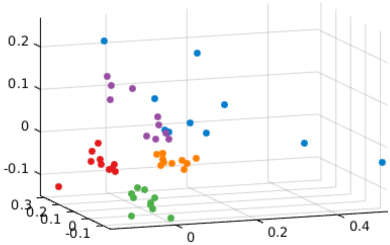


# Learning from Distances

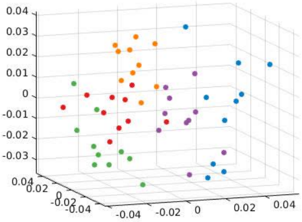
$$\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{pmatrix}$$



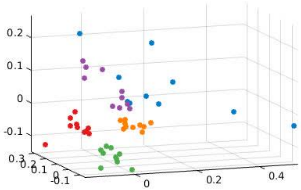




Diffusion Distance

# MDS for CPD & DD

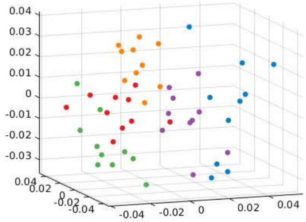



CPD

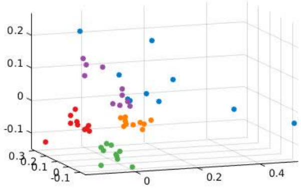



DD

# MDS for CPD & DD

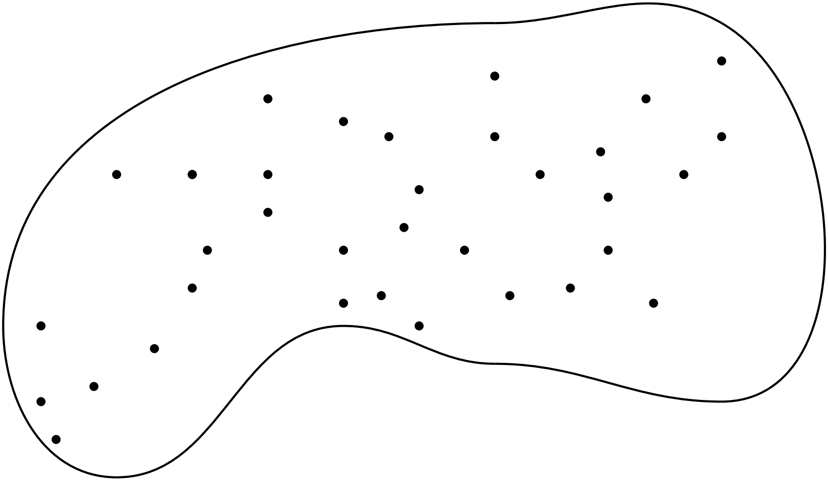



CPD

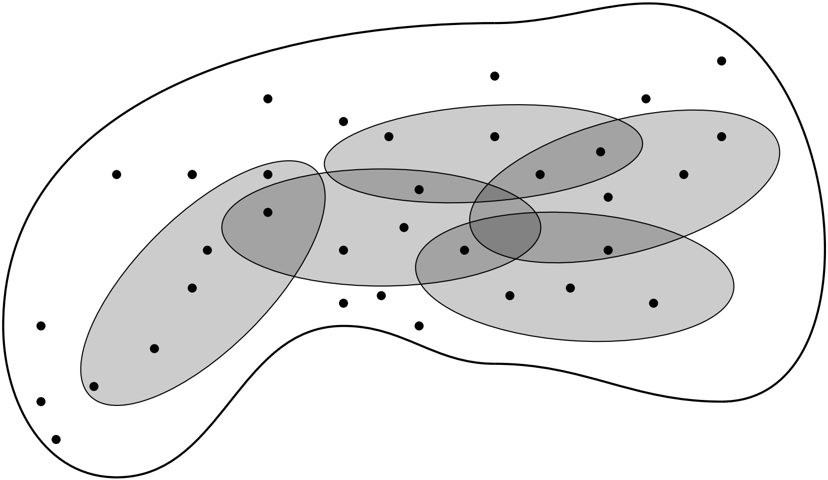



DD

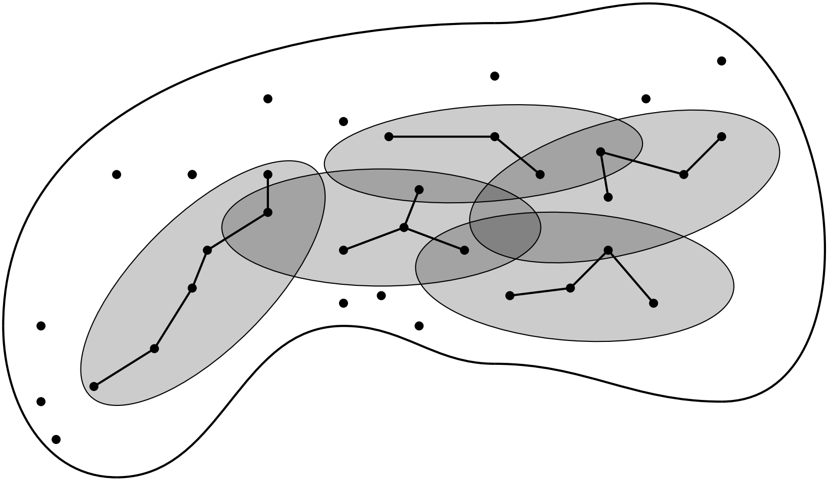
# Diffusion Maps: “Knit Together” Local Geometry



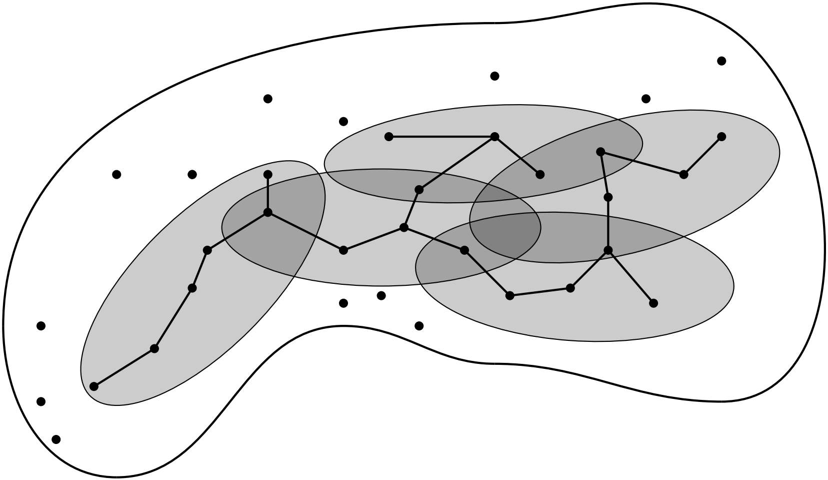
# Diffusion Maps: "Knit Together" Local Geometry



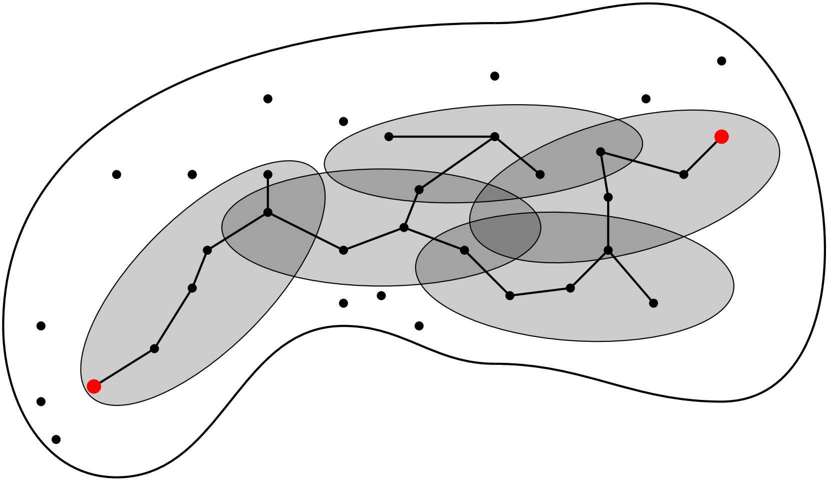
# Diffusion Maps: "Knit Together" Local Geometry



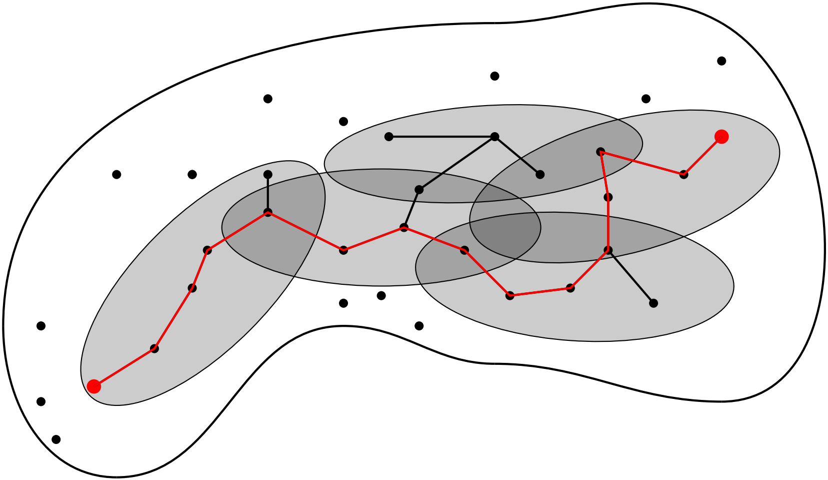
# Diffusion Maps: “Knit Together” Local Geometry



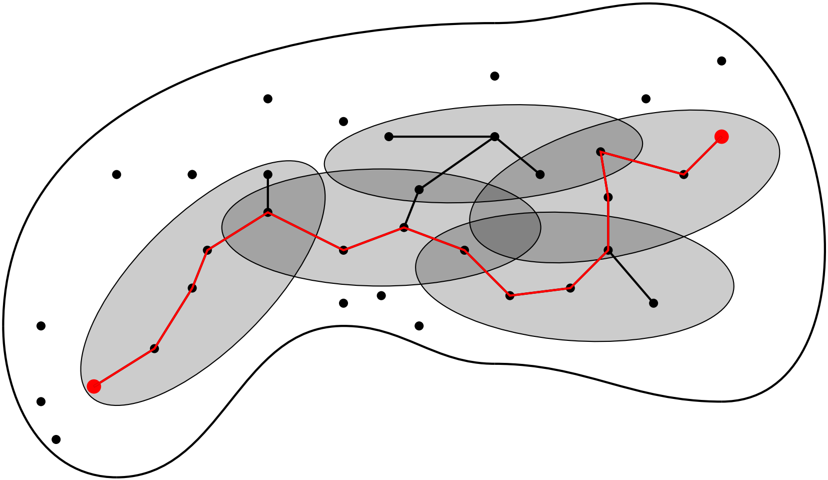
# Diffusion Maps: “Knit Together” Local Geometry



# Diffusion Maps: “Knit Together” Local Geometry

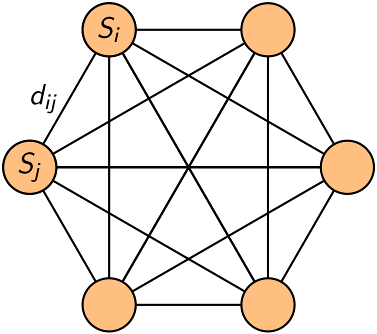


# Diffusion Maps: "Knit Together" Local Geometry



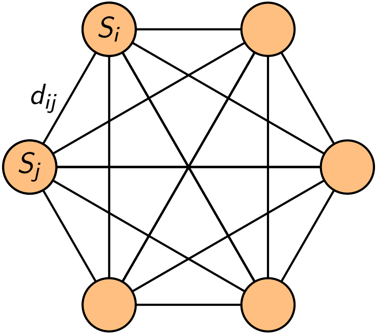
**Small distances are much more reliable!**

# Diffusion Maps: "Knit Together" Local Geometry




- $P = D^{-1}W$ defines a random walk on the graph

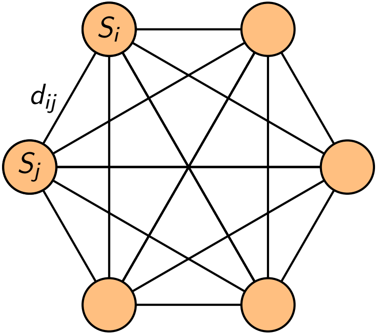
# Diffusion Maps: "Knit Together" Local Geometry




- $P = D^{-1}W$ defines a random walk on the graph
- Solve eigen-problem

$$Pu_j = \lambda_j u_j, \ j = 1, 2, \cdots, m$$

# Diffusion Maps: "Knit Together" Local Geometry




- $P = D^{-1}W$ defines a random walk on the graph
- Solve eigen-problem

$$Pu_j = \lambda_j u_j,\ j = 1, 2, \cdots, m$$

and represent each individual shape $S_j$ as an $m$-vector

$$\left(\lambda_1^{t/2} u_1(j), \cdots, \lambda_m^{t/2} u_m(j)\right)$$

# Diffusion Distance (DD)

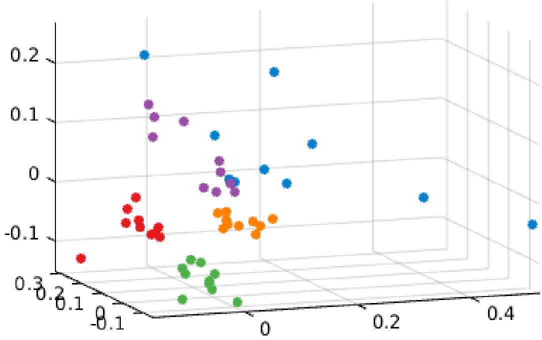
Fix $1 \leq m \leq N$, $t \geq 0$,

$$\boxed{D_m^t\left(S_i, S_j\right) = \left(\sum_{k=1}^{m} \lambda_k^t \left(u_k\left(i\right) - u_k\left(j\right)\right)^2\right)^{\frac{1}{2}}}$$

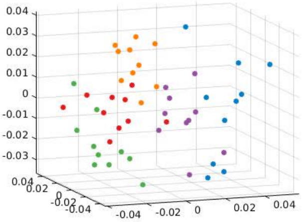
# Diffusion Distance (DD)

Fix $1 \leq m \leq N$, $t \geq 0$,

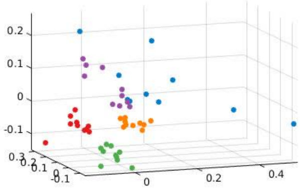
$$D_m^t\left(S_i, S_j\right) = \left(\sum_{k=1}^{m} \lambda_k^t \left(u_k\left(i\right) - u_k\left(j\right)\right)^2\right)^{\frac{1}{2}}$$



# MDS for CPD & DD

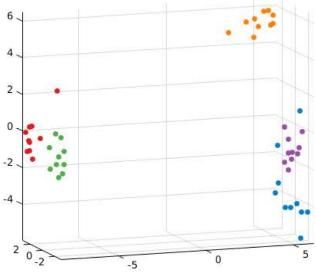



CPD

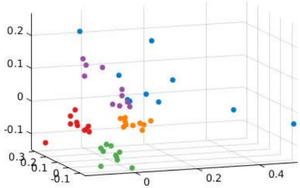



DD

# Even Better: More Information!

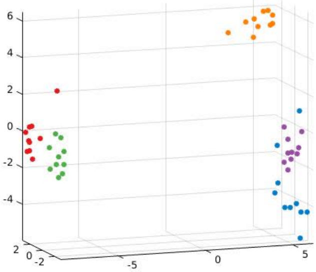



HBDD

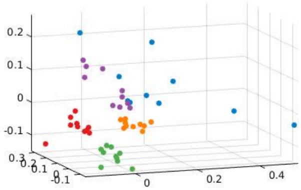



DD

# Even Better: More Information!




HBDD

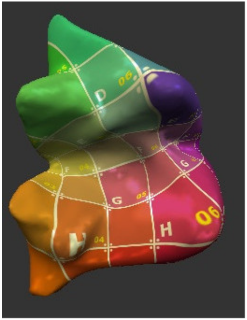



DD

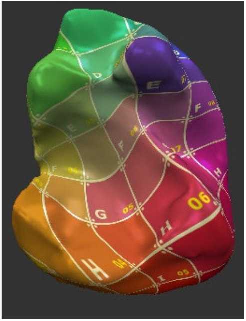
# Use the Information in the Maps!

$$D\left(S_1, S_2\right) = \inf_{\mathcal{C} \in \mathcal{A}(S_1, S_2)} \inf_{R \in \mathbb{E}(3)} \left( \int_{S_1} \left\| R\left(x\right) - \mathcal{C}\left(x\right) \right\|^2 d\mathrm{vol}_{S_1}\left(x\right) \right)^{\frac{1}{2}}$$

$S_1$

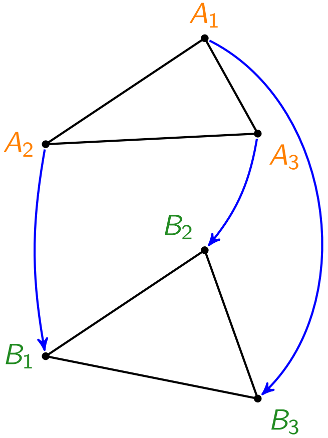


$\xrightarrow[f_{12}]{d_{12}}$

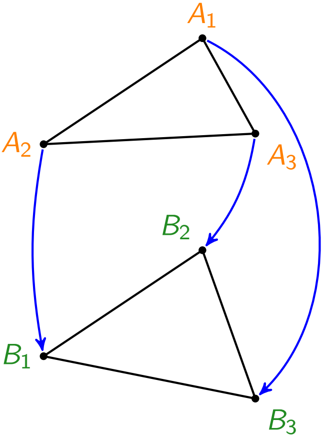


$S_2$

# Correspondences Between Triangular Meshes

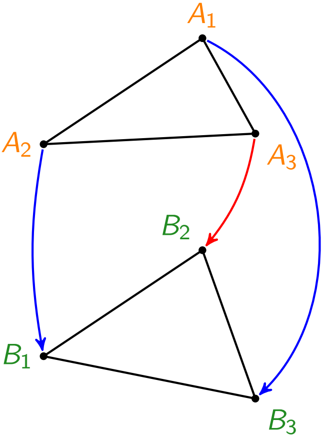


# Correspondences Between Triangular Meshes



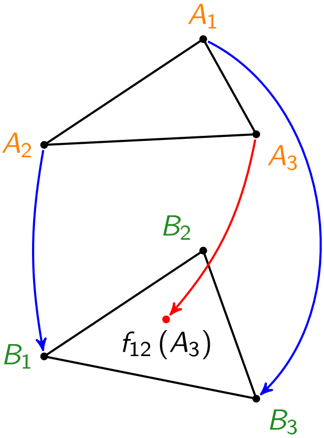

$$
\begin{array}{cc}
 & \xrightarrow{\qquad\qquad S_2 \qquad\qquad} \\
 & \begin{array}{ccccc} \cdots & B_1 & B_2 & B_3 & \cdots \end{array} \\
S_1 \Big\downarrow \begin{array}{c} \vdots \\ A_1 \\ A_2 \\ A_3 \\ \vdots \end{array} &
\begin{pmatrix}
 & \vdots & \vdots & \vdots & \\
\cdots & 0 & 0 & 1 & \cdots \\
\cdots & 1 & 0 & 0 & \cdots \\
\cdots & 0 & 1 & 0 & \cdots \\
 & \vdots & \vdots & \vdots &
\end{pmatrix}
\end{array}
$$

# Correspondences Between Triangular Meshes



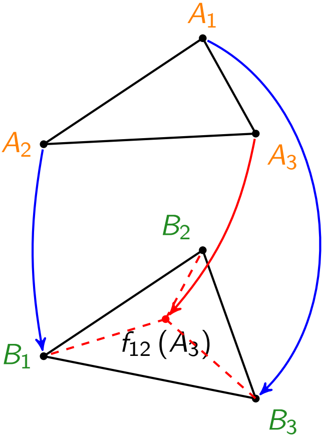

$$
\begin{array}{c c c}
 & & S_2 \longrightarrow \\
 & & \begin{array}{ccccc} \cdots & B_1 & B_2 & B_3 & \cdots \end{array} \\
S_1 \downarrow & \begin{array}{c} \vdots \\ A_1 \\ A_2 \\ A_3 \\ \vdots \end{array} & \begin{pmatrix} & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 1 & \cdots \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 0 & \boxed{1} & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{pmatrix}
\end{array}
$$

# Correspondences Between Triangular Meshes




$$
\begin{array}{ccc}
 & & \overset{S_2}{\longrightarrow} \\
 & & \begin{array}{ccccc} \cdots & B_1 & B_2 & B_3 & \cdots \end{array} \\
S_1 \downarrow & \begin{array}{c} \vdots \\ A_1 \\ A_2 \\ A_3 \\ \vdots \end{array} & \begin{pmatrix} & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 1 & \cdots \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 0 & \boxed{?} & 0 & \cdots \\ & \vdots & \vdots & \vdots & \end{pmatrix}
\end{array}
$$

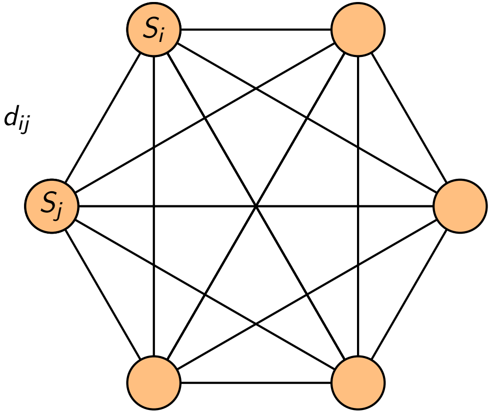
# Correspondences Between Triangular Meshes



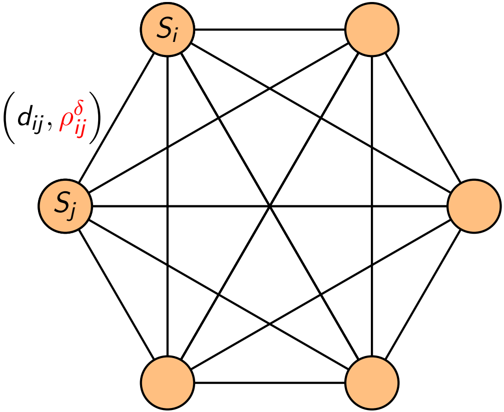

$$\begin{array}{ccc} & & \xrightarrow{\quad s_2 \quad} \\ & & \begin{array}{ccccc} \cdots & B_1 & B_2 & B_3 & \cdots \end{array} \\ s_1 \downarrow & \begin{array}{c} \vdots \\ A_1 \\ A_2 \\ A_3 \\ \vdots \end{array} & \begin{pmatrix} & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 1 & \cdots \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 0.91 & 0.95 & 0.88 & \cdots \\ & \vdots & \vdots & \vdots & \end{pmatrix} \end{array}$$

$$\rho_{12}^{\delta}(r, s) = \exp\left(-\frac{\left\| f_{12}(A_r) - B_s \right\|^2}{\delta}\right)$$

# Distance Graph



# *Augmented* Distance Graph



# *Horizontal Random Walk* on a *Fibre Bundle*

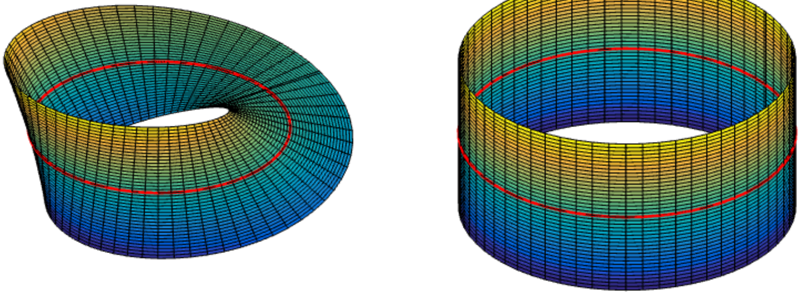
Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold

# *Horizontal Random Walk* on a *Fibre Bundle*

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

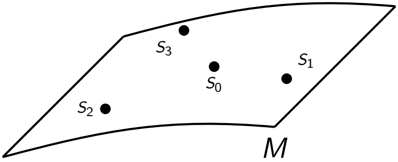
- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

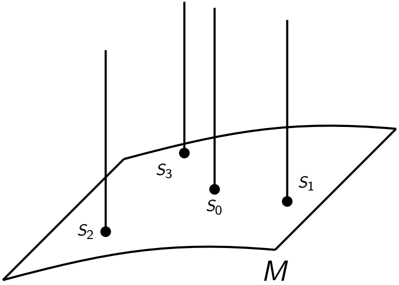


# *Horizontal Random Walk* on a *Fibre Bundle*

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

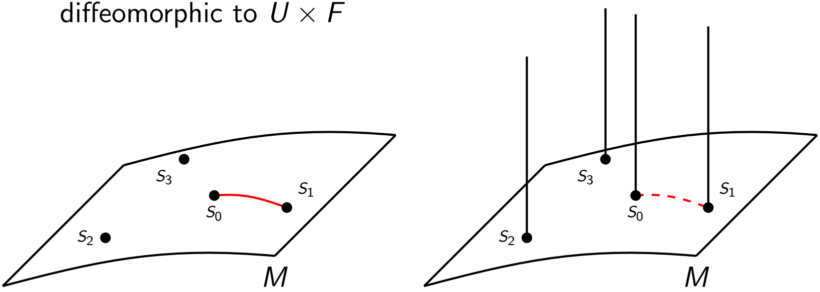


# Horizontal Random Walk on a Fibre Bundle

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

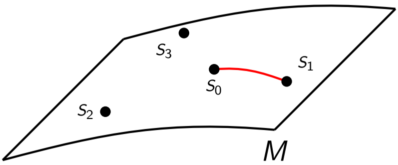
- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

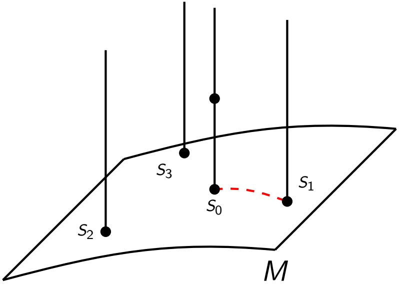


# Horizontal Random Walk on a Fibre Bundle

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

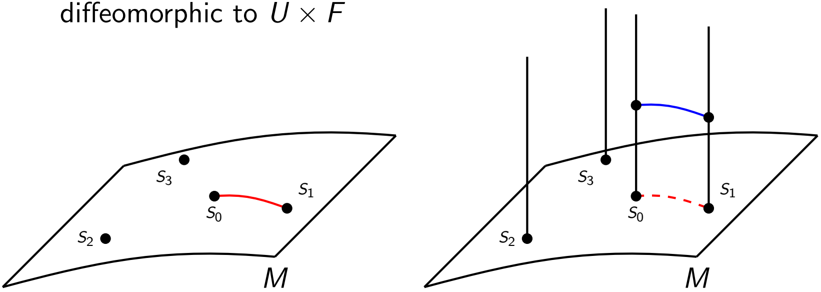


# *Horizontal Random Walk* on a *Fibre Bundle*

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

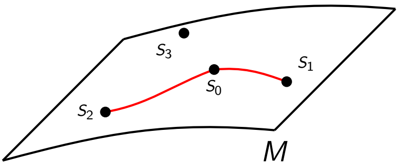
- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

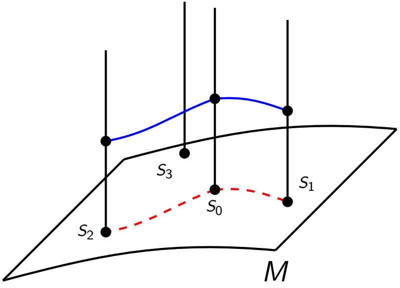


# Horizontal Random Walk on a Fibre Bundle

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$

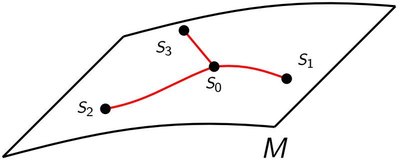


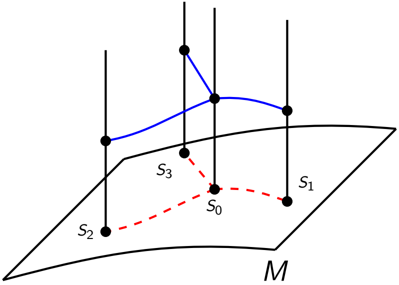



# *Horizontal Random Walk* on a *Fibre Bundle*

Fibre Bundle $\mathscr{E} = (E, M, F, \pi)$

- $E$: *total* manifold
- $M$: *base* manifold
- $\pi : E \to M$: smooth surjective map (*bundle projection*)
- $F$: *fibre* manifold
- *local triviality*: for "small" open set $U \subset M$, $\pi^{-1}(U)$ is diffeomorphic to $U \times F$



# Towards *Horizontal* Diffusion Maps

Diffusion Maps

$$D^{-1} W u_k = \lambda_k u_k, \quad 1 \leq k \leq N$$

$$D^{-1} \begin{pmatrix} & & & \vdots & \\ & & & \vdots & \\ \cdots & \cdots & e^{-d_{ij}^2/\epsilon} & & \cdots \\ & & & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ u_k(j) \\ \vdots \end{pmatrix} = \lambda_k \begin{pmatrix} \vdots \\ \vdots \\ u_k(j) \\ \vdots \end{pmatrix}$$

# Towards *Horizontal* Diffusion Maps

Horizontal Diffusion Maps

$$\mathcal{D}^{-1}\mathcal{W}u_k = \lambda_k u_k, \quad 1 \leq k \leq \kappa$$

$$\mathcal{D}^{-1}\begin{pmatrix} & & \vdots & \\ & & \vdots & \\ \cdots & \cdots & e^{-d_{ij}^2/\epsilon}\rho_{ij}^{\delta} & \cdots \\ & & \vdots & \end{pmatrix}\begin{pmatrix} \vdots \\ \vdots \\ u_{k[j]} \\ \vdots \end{pmatrix} = \lambda_k \begin{pmatrix} \vdots \\ \vdots \\ u_{k[j]} \\ \vdots \end{pmatrix}$$

# Towards *Horizontal* Diffusion Maps

Horizontal Diffusion Maps

$$\mathcal{D}^{-1}\mathcal{W}u_k = \lambda_k u_k, \quad 1 \leq k \leq \kappa$$

$$\mathcal{D}^{-1}\begin{pmatrix} & & \vdots & \\ & & \vdots & \\ \cdots & \cdots & \boxed{e^{-d_{ij}^2/\epsilon}\rho_{ij}^{\delta}} & \cdots \\ & & \vdots & \end{pmatrix}\begin{pmatrix} \vdots \\ \vdots \\ \boxed{u_{k[j]}} \\ \vdots \end{pmatrix} = \lambda_k \begin{pmatrix} \vdots \\ \vdots \\ \boxed{u_{k[j]}} \\ \vdots \end{pmatrix}$$

# Towards *Horizontal* Diffusion Maps

Horizontal Diffusion Maps

$$\mathcal{D}^{-1}\mathcal{W}u_k = \lambda_k u_k, \quad 1 \leq k \leq \kappa$$

$$\mathcal{D}^{-1}\begin{pmatrix} & & \vdots & \\ & & \vdots & \\ \cdots & \cdots & e^{-d_{ij}^2/\epsilon}\rho_{ij}^{\delta} & \cdots \\ & & \vdots & \end{pmatrix}\begin{pmatrix} \vdots \\ \vdots \\ u_{k[j]} \\ \vdots \end{pmatrix} = \lambda_k \begin{pmatrix} \vdots \\ \vdots \\ u_{k[j]} \\ \vdots \end{pmatrix}$$

Horizontal Diffusion Maps: For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\kappa_j \times m$ *matrix*

$$\left(\lambda_1^{t/2}u_{1[j]}, \cdots, \lambda_m^{t/2}u_{m[j]}\right)$$

# Diffusion Maps vs. Horizontal Diffusion Maps

Diffusion Maps: For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as an $m$-dimensional *vector*

$$\left(\lambda_1^{t/2} u_1\left(j\right), \cdots, \lambda_m^{t/2} u_m\left(j\right)\right)$$

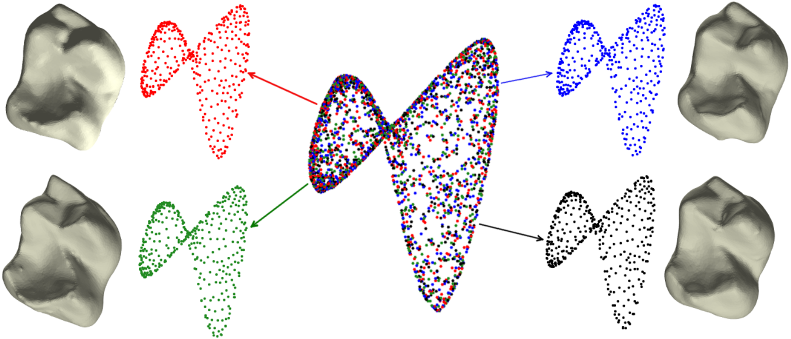
Horizontal Diffusion Maps: For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\kappa_j \times m$ *matrix*

$$\left(\lambda_1^{t/2} u_{1[j]}, \cdots, \lambda_m^{t/2} u_{m[j]}\right)$$

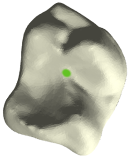
# HDM: Application in Geometric Morphometrics

1. Global Registration
2. Automatic Landmarking
3. Species Classification

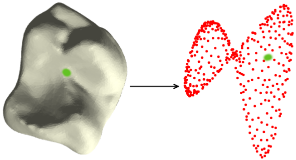
# 1. Global Registration



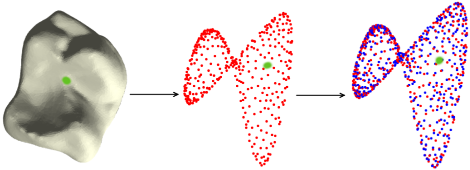
# 1. Global Registration



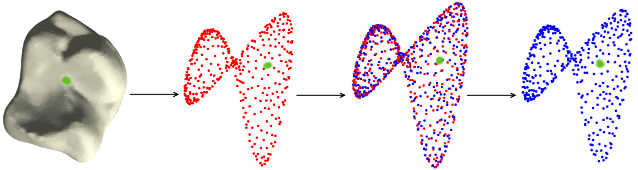
# 1. Global Registration



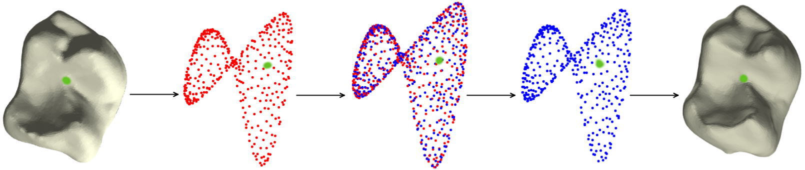
# 1. Global Registration



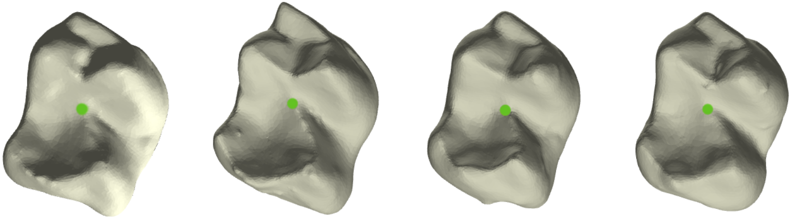
# 1. Global Registration



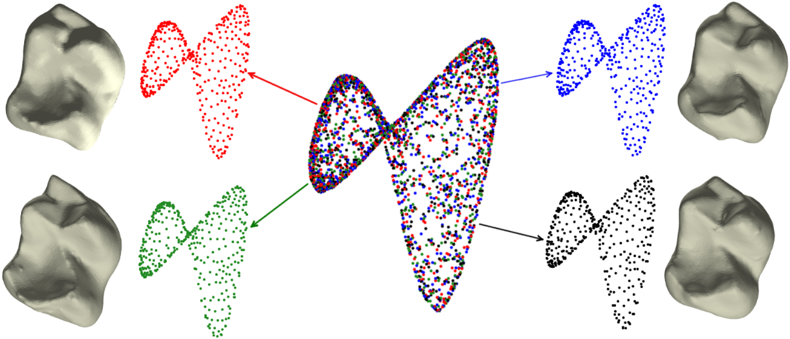
# 1. Global Registration



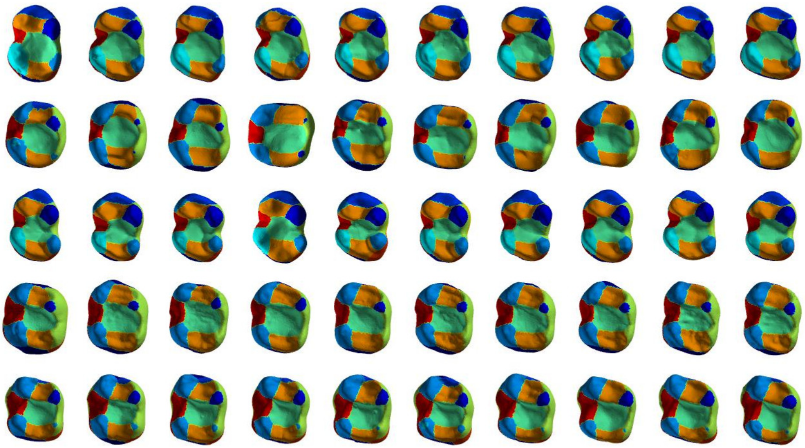
# 1. Global Registration



# 2. Automatic Landmarking: *Spectral Clustering*



# 2. Automatic Landmarking: *Spectral Clustering*



# 3. Species Classification: HBDM & HBDD

Horizontal Diffusion Maps (HDM): For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\kappa_j \times m$ *matrix*

$$\left(\lambda_1^{t/2} u_{1[j]}, \cdots, \lambda_m^{t/2} u_{m[j]}\right)$$

# 3. Species Classification: HBDM & HBDD

Horizontal Diffusion Maps (HDM): For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\kappa_j \times m$ *matrix*

$$\left(\lambda_1^{t/2} u_{1[j]}, \cdots, \lambda_m^{t/2} u_{m[j]}\right)$$

Horizontal Base Diffusion Maps (HBDM): For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\binom{m}{2}$-dimensional *vector*

$$\left(\lambda_\ell^{t/2} \lambda_k^{t/2} \left\langle u_{\ell[j]}, u_{k[j]} \right\rangle\right)_{1 \leq \ell < k \leq m}$$

# 3. Species Classification: HBDM & HBDD

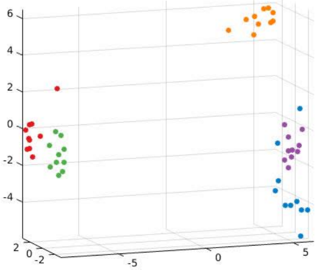
Horizontal Base Diffusion Distance (HBDD): For fixed $1 \leq m \leq \kappa$, $t \geq 0$,

$$D_{HB}^{t}\left(S_i, S_j\right) = \left(\sum_{1 \leq \ell < k \leq m} \lambda_\ell^t \lambda_k^t \left(\left\langle u_{\ell[i]}, u_{k[i]}\right\rangle - \left\langle u_{\ell[j]}, u_{k[j]}\right\rangle\right)^2\right)^{\frac{1}{2}}$$

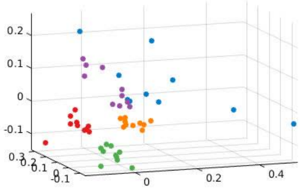
Horizontal Base Diffusion Maps (HBDM): For fixed $1 \leq m \leq \kappa$, $t \geq 0$, represent $S_j$ as a $\binom{m}{2}$-dimensional *vector*

$$\left(\lambda_\ell^{t/2} \lambda_k^{t/2} \left\langle u_{\ell[j]}, u_{k[j]}\right\rangle\right)_{1 \leq \ell < k \leq m}$$

# 3. Species Classification: HBDM & HBDD

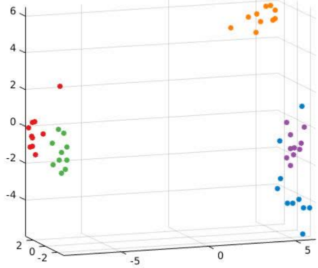



HBDD

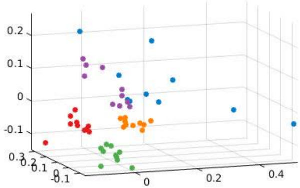



DD

# 3. Species Classification: HBDM & HBDD

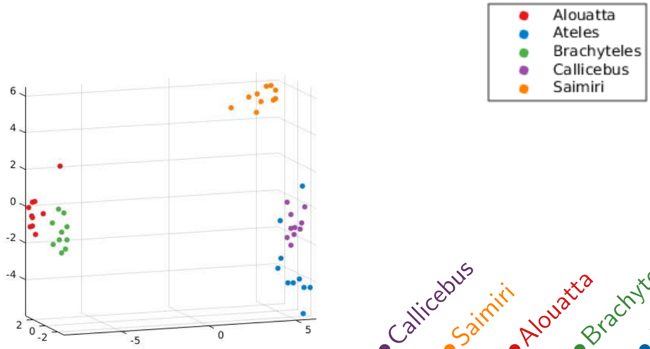



HBDD

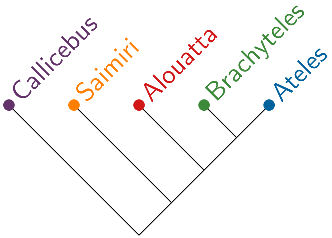



DD

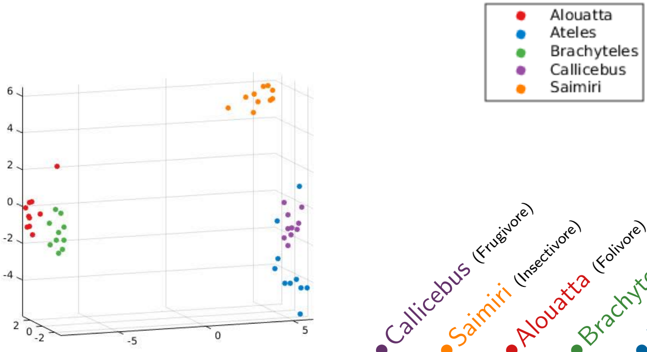
# 3. Species Classification: HBDM & HBDD



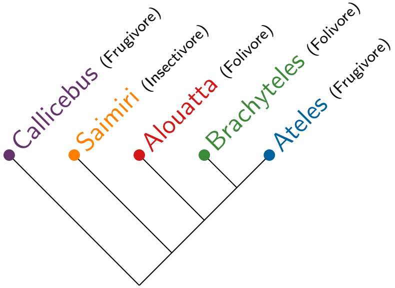

HBDD



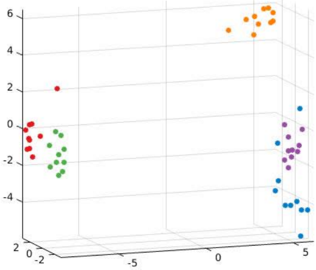
# 3. Species Classification: HBDM & HBDD




HBDD

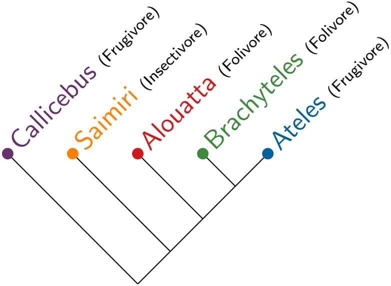


# 3. Species Classification: HBDM & HBDD





HBDD




*"Form Follows Function"*