Geometric Modeling

Summer Semester 2012

Dimensionality Reduction, Intrinsic Geometry and Discrete Differential Operators

UNIVERSITÄT DES SAARLANDES

CLUSTER OF EXCELLENCE

max planck institut informatik

Dimensionality Reduction

Fitting Affine Subspaces

The following problem can be solved exactly:

- Best fitting line to a set of 2D, 3D points
- Best fitting plane to a set of 3D points
- In general: Affine subspace of $\mathbb{R}^d$, with dimension $d \leq m$ that best approximates a set of data points $\mathbf{x}_i \in \mathbb{R}^m$.

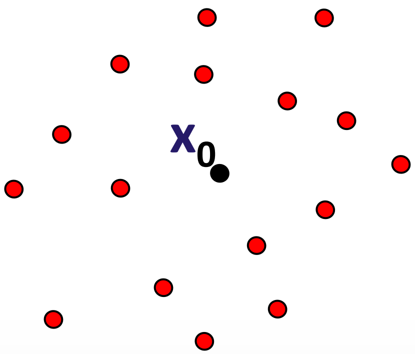
This will lead to the - famous - *principle component analysis (PCA)*.

Start: 0-dim Subspaces

Easy Start: The optimal 0-dimensional affine subspace

- Given a set $\mathbf{X}$ of n data points $\mathbf{x}_i \in \mathbb{R}^m$, what is the point $\mathbf{x}_0$ with minimum least square error to all data points?
- Answer: just the sample mean (average)...:

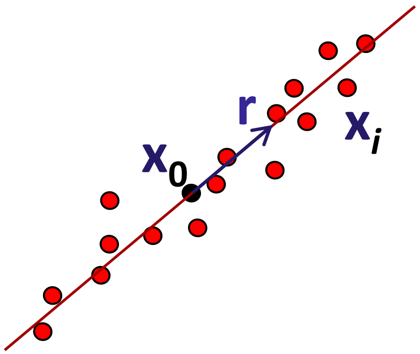
$$\mathbf{x}_0 = \mathrm{m}(\mathbf{X}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$



One Dimensional Subspaces...

Next:

- What is the optimal line (1D subspace) that approximates a set of data points $\mathbf{X}$?
- Two questions:
  - Optimum origin (point on the line)?
    - This is still the average
  - Optimum direction?
    - We will look at that next...
- Parametric line equation:



$$\mathbf{x}(t) = \mathbf{x}_0 + t.\mathbf{r} \qquad (\mathbf{x}_0 \in \mathbb{R}^m, \mathbf{r} \in \mathbb{R}^m, \|\mathbf{r}\| = 1)$$

Best Fitting Line

Result:

$$\sum_{i=1}^{n} dist(line, \mathbf{x}_i)^2 = -\mathbf{r}^{\mathrm{T}}\mathbf{S}\mathbf{r} + \text{const.}$$

with $\mathbf{S} = \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{x}_0)(\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}, \ \|\mathbf{r}\| = 1$

Eigenvalue Problem:

- $\mathbf{r}^{\mathrm{T}}\mathbf{S}\mathbf{r}$ **is a Rayleigh quotient**
- **Minimizing the energy: maximum quotient**
- **Solution: eigenvector with *largest* eigenvalue**

General Case

Fitting a d-dimensional affine subspace:

- $d = 1$: **line**
- $d = 2$: **plane**
- $d = 3$: **3D subspace**
- **...**

Simple rule:

- **Use the d eigenvectors with *largest eigenvalues* from the spectrum of S.**
- **Gives the (total) least-squares optimal subspace that approximates the data set $\mathbf{X}$.**

PCA Maximum Variance Formulation

Alternate Formulation:

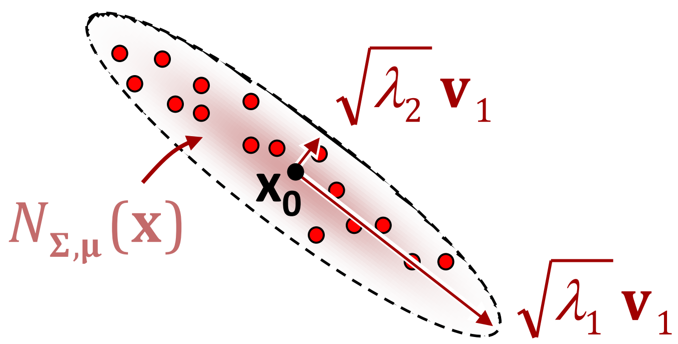
- Let $\mathbf{v} \in \mathbb{R}^m$ be the 1D subspace (with $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$), that maximize the variance of data $\mathbf{X} \in \mathbb{R}^m$.
- Each data point $\mathbf{x}_i$ is projected onto a scalar value $\mathbf{v}^{\mathrm{T}}\mathbf{x}_i$.
- The mean of projected data is $\mathbf{v}^{\mathrm{T}}\mathbf{x}_0$.
- The variance of the projected data is given by:
  $\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{v}^{\mathrm{T}}\mathbf{x}_0\right)^2 = \mathbf{v}^{\mathrm{T}}\mathbf{S}\mathbf{v}$; $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{x}_0)(\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}$
- The problem now reduces to

$$\mathbf{v}^* = \boldsymbol{arg\ max}\,\{\mathbf{v}^{\mathrm{T}}\mathbf{S}\mathbf{v} + \lambda(1 - \mathbf{v}^{\mathrm{T}}\mathbf{v})\}$$

- Solution:

Eigenvector of **S** with ***largest*** eigenvalue λ_1.

Statistical Interpretation



$$\boldsymbol{\mu} = \mathbf{x}_0 = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

$$\boldsymbol{\Sigma} = \frac{1}{n-1}\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{x}_0)(\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}$$

$$N_{\boldsymbol{\Sigma},\boldsymbol{\mu}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\det(\boldsymbol{\Sigma})^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Observation:

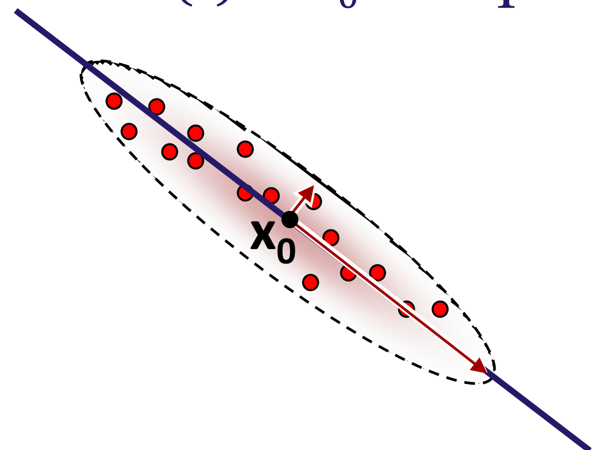
- $\frac{1}{n-1}\mathbf{S}$ is the covariance matrix of the data $\mathbf{X}$.
- PCA can be interpreted as fitting a Gaussian distribution and computing the main axes.

Applications

Fitting a line to a point cloud in $\mathbb{R}^2$:

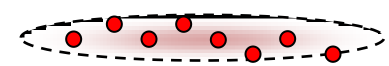
- Sample mean and direction of maximum eigenvalue

$\mathbf{x}(t) = \mathbf{x}_0 + t.\mathbf{v}_1$

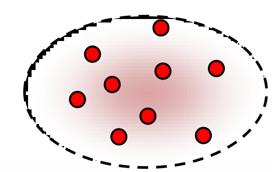


Plane Fitting in $\mathbb{R}^3$:

- Sample mean and the two directions of maximum eigenvalues
- Smallest eigenvalue
  - Eigenvector points in normal direction
  - Aspect ratio (λ_3 / λ_2) is a measure of "flatness" (quality of fit)



(λ_2 / λ_1) small



(λ_2 / λ_1) larger

Applications

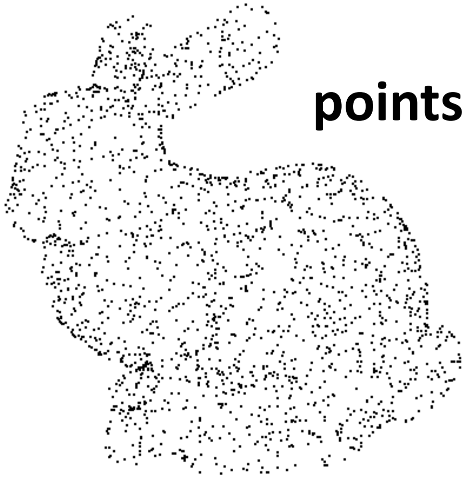
Application: Normal estimation in point clouds

- Given a set of points $\mathbf{X} \in \mathbb{R}^{\mathbf{3}}$ sampled from a smooth surface.
- We want to estimate Surface Normals.

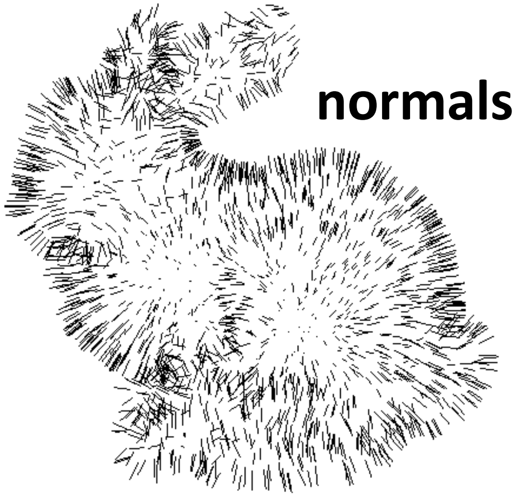
Algorithm:

- For each point, compute the k-nearest neighbors ($k = 20$).
- Compute a PCA (average, main axes) of these points.
- Eigenvector with smallest eigenvalue $\rightarrow$ normal direction.
- The other two eigenvectors $\rightarrow$ tangent vectors.

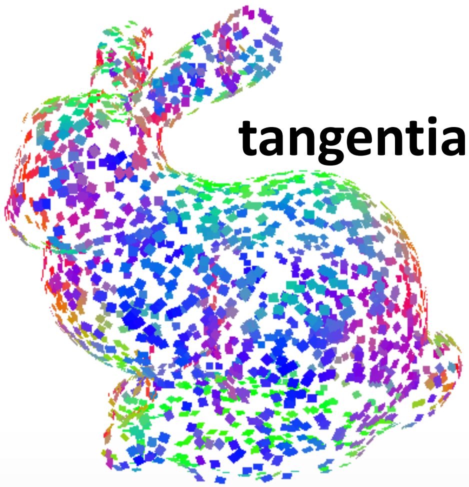
Example



points



normals



tangential frames

Dimensionality Reduction

Notations:

- $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_n], \quad \mathbf{x}_k \in \mathbb{R}^m$

- $\mathbf{R}$ is a $m \times m$ orthogonal matrix with $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}_{\boldsymbol{m}}$.

Projection: from $\mathbb{R}^m$ onto $\mathbb{R}^d$ removes $m - d$ rows of $\mathbf{R}^{\mathrm{T}}$ to obtain $\mathbf{Q}^{\mathrm{T}}$

- $\mathbf{Y} = \mathbf{Q}^{\mathrm{T}}\mathbf{X}$ with $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}_{\boldsymbol{d}}$.

Reconstruction: of $\mathbb{R}^m$ from $\mathbb{R}^d$ removes $m - d$ columns of $\mathbf{R}$ to obtain $\mathbf{Q}$

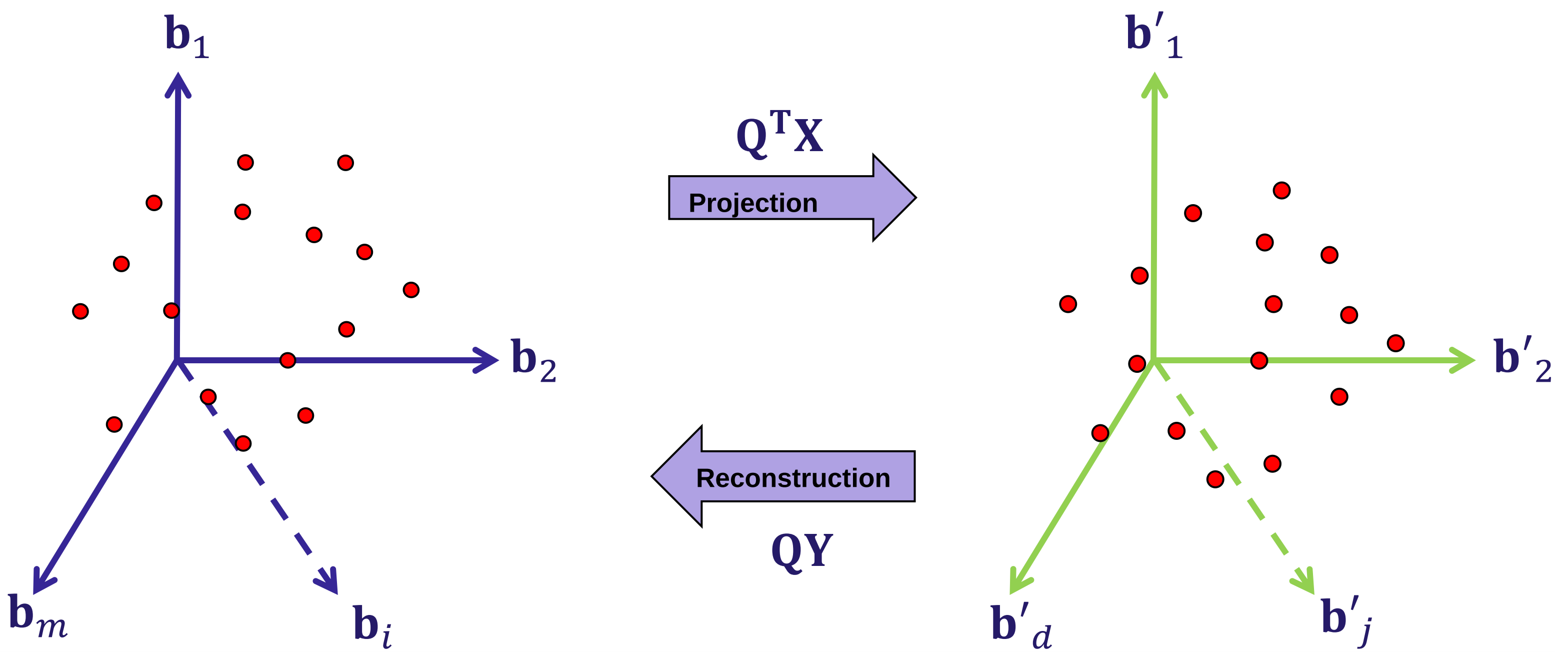
- $\mathbf{X} = \mathbf{Q}\mathbf{Y}$ with $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}_{\boldsymbol{d}}$.

$\det(\mathbf{R}) = +1$ implies a *Rotation* matrix.

Dimensionality Reduction

Idea: Projection of higher dimensional data to a lower dimensional subspace.

- $\mathbf{X} \in \mathbb{R}^m \rightarrow \mathbf{Y} \in \mathbb{R}^d$ **where** $\boxed{d \ll m}$ ***Information is LOST**



- ***Linear*** **Dimensionality Reduction using PCA computes** $\mathbf{Q} = \mathbf{VI}_d$.

Metric Multi Dimensional Scaling

PCA uses Covariance matrix

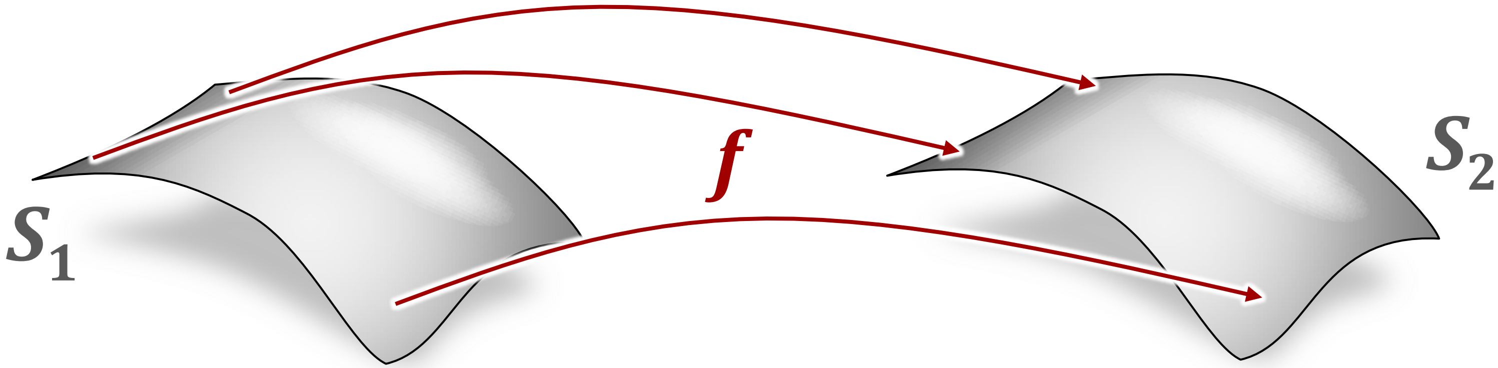
- $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^{\mathrm{T}}$ assuming *centered* data (*i.e.* zero mean).
- The data points are represented in a vector space.
- Dimension of $\mathbf{S}$ is $m \times m$.

MDS uses Gram Matrix (*dot products*)

- $\mathbf{G} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ captures (dis)similarity of data points.
- The data points are *not explicitly required*.
- Dimension of $\mathbf{G}$ is $n \times n$.
- Goal is to *embed* the data in d-dimensional space such that some metric is preserved.

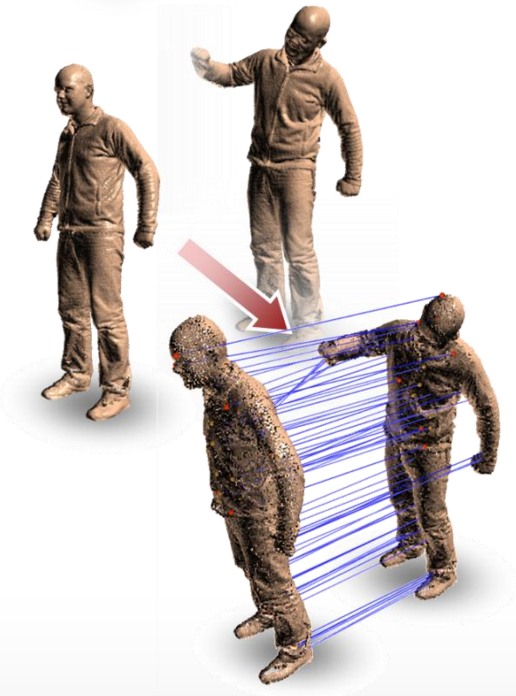
Part III: Intrinsic Geometry and Intrinsic Mappings

Scenario



Mapping between Surfaces

- Intrinsic view – only metric tensor
- Ignore isometric deformations
- Applications:
  - Deformable shape matching
  - Texture mapping (flat $\rightarrow$ 3D) "parametrization"



Differential Geometry Revisited

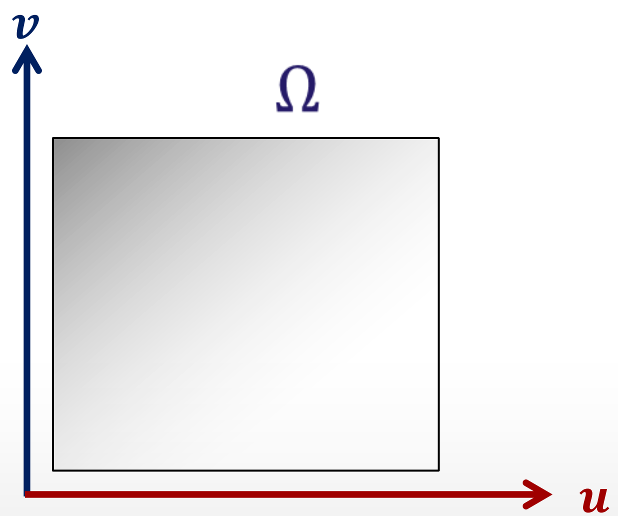
Parametric Patches

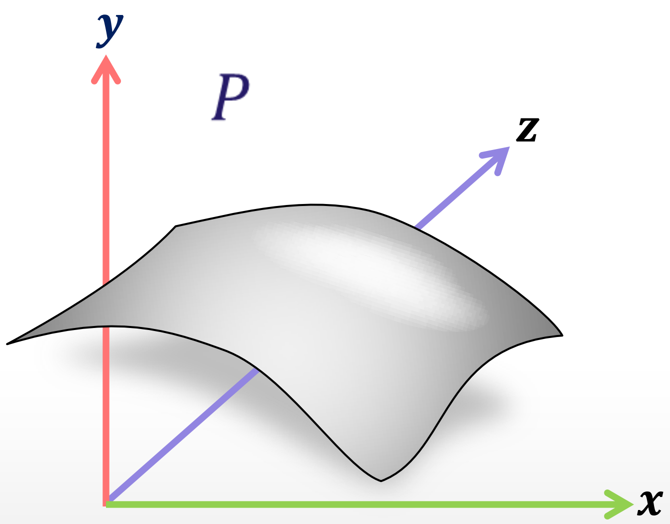
Parametric Surface Patches:

A smoothly differentiable function

$$f : \mathbb{R}^2 \supseteq \Omega \to \mathbb{R}^3 \text{ describes } P = f(\Omega), P \subseteq \mathbb{R}^3.$$

$$f(u,v) = \big(x(u,v), y(u,v), z(u,v)\big)$$

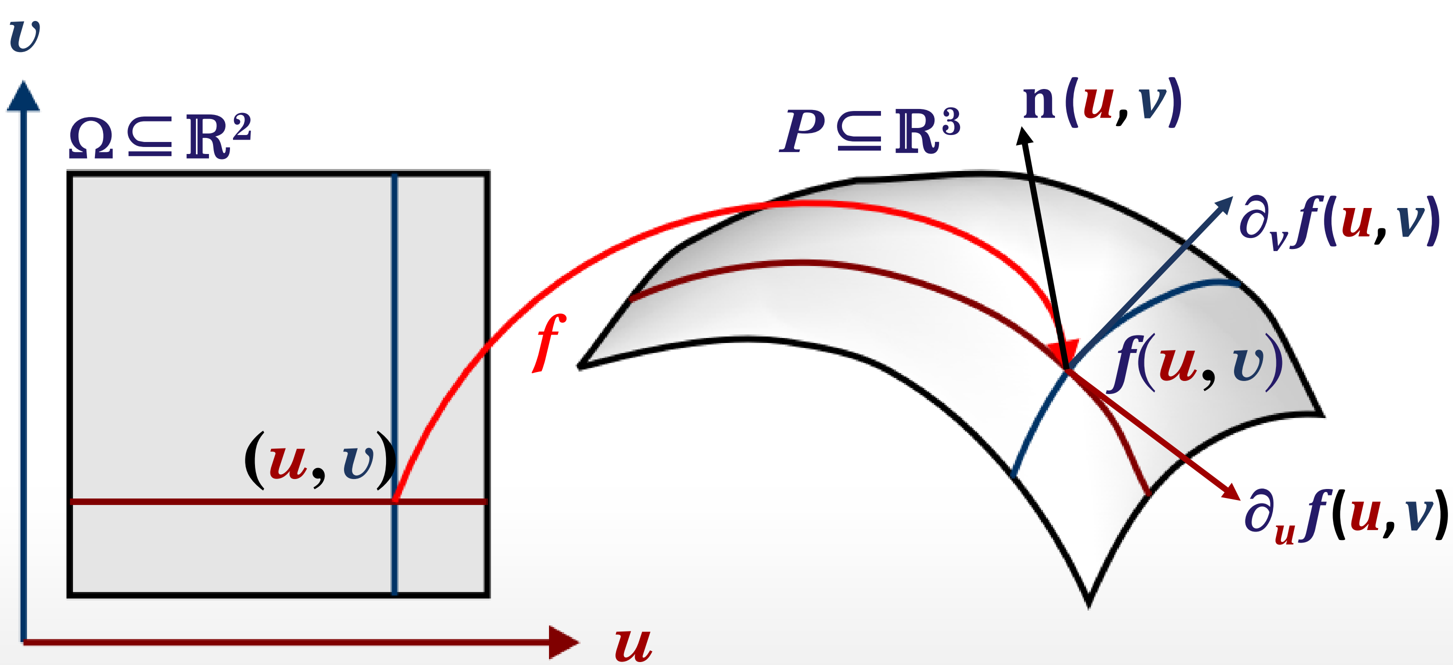




Parametric Patches

Function $f(u,v) \to \mathbb{R}^3$

- Canonical tangents: $\partial_u f(u,v),\ \partial_v f(u,v)$
- Normal: $\mathbf{n}(u,v) = \frac{\partial_u f(u,v) \times \partial_v f(u,v)}{\|\partial_u f(u,v) \times \partial_v f(u,v)\|}$

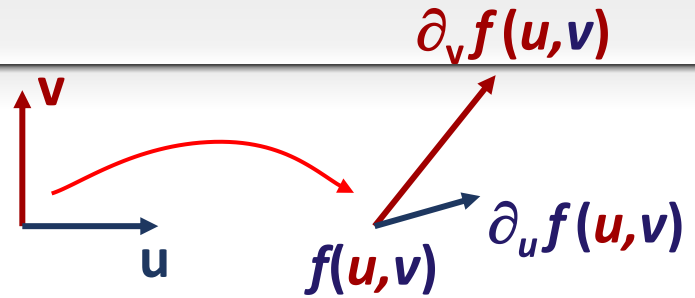


Fundamental Forms

Fundamental Forms:

- Describe the local parameterized surface.
- Measure...
  - ...distortion of length (first fundamental form)
  - ...surface curvature (second fundamental form)

First Fundamental Form



First Fundamental Form

- Also known as ***metric tensor***.
- It can be written as a 2×2 symmetric matrix:

$$\mathbf{I} = \begin{pmatrix} \partial_u f \partial_u f & \partial_u f \partial_v f \\ \partial_u f \partial_v f & \partial_v f \partial_v f \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

- The matrix is symmetric and positive definite (regular parametrization, semi-definite otherwise)
- Defines a *generalized scalar product* that measures lengths and angles *on the surface*.

Second Fundamental Form

Second Fundamental Form

- Also known as *shape operator* or *curvature tensor*.
- It can be written as a 2×2 symmetric matrix:

$$\mathbf{II} = \begin{pmatrix} \partial_{uu} f . \mathbf{n} & \partial_{uv} f . \mathbf{n} \\ \partial_{uv} f . \mathbf{n} & \partial_{vv} f . \mathbf{n} \end{pmatrix} =: \begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

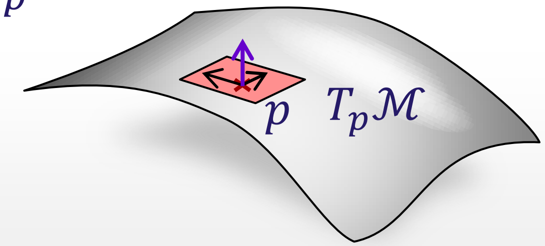
Eigen-analysis:

- Eigenvalues of *second fundamental form* for an *orthonormal tangent* basis are called *principal curvatures* κ_1, κ_2.
- Corresponding orthogonal eigenvectors are called *principal directions of curvature*.

How to build a metric

Tangent Space

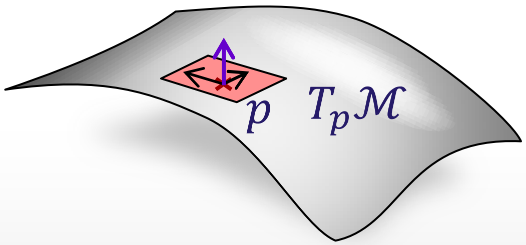
- For a d-dimensional manifold $\mathcal{M}$, at each point $p \in \mathcal{M}$ there exist a vector space $T_p\mathcal{M}$, called "*Tangent Space*".
- It consist of all tangent vectors to manifold at point p.
- A metric at point p is a function $g_p: T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ such that:
  - g_p is Bilinear : $g_p(a\boldsymbol{\alpha}_p + b\boldsymbol{\beta}_p, \boldsymbol{\gamma}_p) = a g_p(\boldsymbol{\alpha}_p, \boldsymbol{\gamma}_p) + b g_p(\boldsymbol{\beta}_p, \boldsymbol{\gamma}_p)$ and $g_p(\boldsymbol{\gamma}_p, a\boldsymbol{\alpha}_p + b\boldsymbol{\beta}_p) = a g_p(\boldsymbol{\gamma}_p, \boldsymbol{\alpha}_p) + b g_p(\boldsymbol{\gamma}_p, \boldsymbol{\beta}_p)$
  - g_p is Symmetric: $g_p(\boldsymbol{\alpha}_p, \boldsymbol{\beta}_p) = g_p(\boldsymbol{\beta}_p, \boldsymbol{\alpha}_p)$
  - g_p is Non-degenerated: $\boldsymbol{\beta}_p \mapsto g_p(\boldsymbol{\alpha}_p, \boldsymbol{\beta}_p), \ \forall \boldsymbol{\alpha}_p \neq \mathbf{0}$



How to build a metric

Inner Product Metric

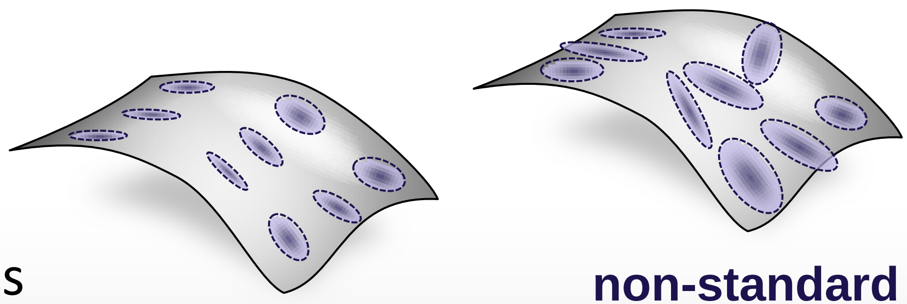
- Two tangent vector at point p can be defined as:
  - $\boldsymbol{\alpha_p} = \alpha_1 \partial_u f + \alpha_2 \partial_{\boldsymbol{v}} f$ and $\boldsymbol{\beta_p} = \beta_1 \partial_u f + \beta_2 \partial_{\boldsymbol{v}} f$
- The Inner product is defined as:
  - $g_p(\boldsymbol{\alpha_p}, \boldsymbol{\beta_p}) = \langle \boldsymbol{\alpha_p}, \boldsymbol{\beta_p} \rangle = [\alpha_1 \quad \alpha_2] \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$
  - It inherently uses the first fundamental form.
- $g_p(\boldsymbol{\alpha_p}, \boldsymbol{\beta_p})$ is *symmetric*, *bilinear* and *non-degenerated* and hence a metric.



Riemannian Manifolds

Riemannian Manifold

- Manifold topology, d-dimensional
  - We mostly focus on 2-manifolds, embedded in $\mathbb{R}^3$
- Real differentiable manifold.
- Local parametrization: tangent space
- Intrinsic metric (metric tensor everywhere)
  - Allows to define various geometric notions on manifold:
    - Angles
    - Lengths of curves
    - Areas (or volumes)
    - Curvature
    - Gradients of functions
    - Divergence of vector fields



standard metric

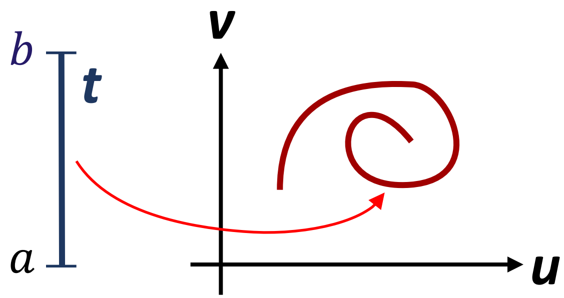
non-standard (pos-def.)

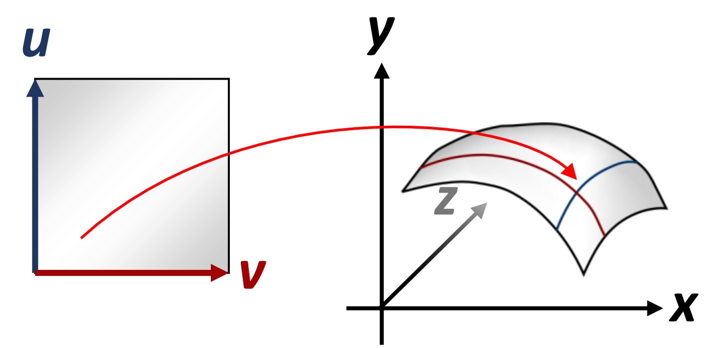
Riemannian Manifolds

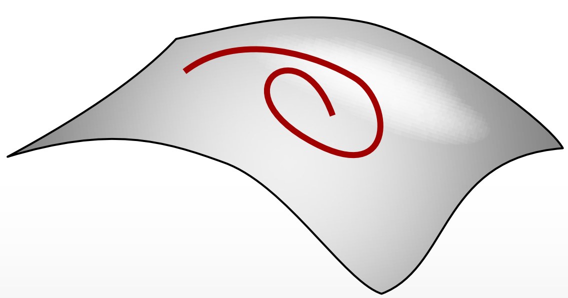
Arclength:

- Let $g(t) = (u, v)$, in an interval $[a, b]$.
- $f\big(u(t), v(t)\big)$ will trace out a parametric curve in P.

Arclength of parametric curve in interval $[a, b]$:







Riemannian Manifolds

Arclength:

- Let $g(t) = (u, v)$, in an interval $[a, b]$.
- $f\big(u(t), v(t)\big)$ will trace out a parametric curve in P.

Arclength of parametric curve in interval $[a, b]$:

$$
\begin{aligned}
s &= \int_a^b \left\| \frac{d}{dt} f\big(u(t), v(t)\big) \right\| dt \\
&= \int_a^b \| du\, \partial_u f + dv\, \partial_v f \| dt \\
&= \int_a^b \sqrt{du^2 \partial_u f . \partial_u f + 2du\, dv\, \partial_u f . \partial_v f + dv^2 \partial_v f . \partial_v f}\; dt \\
&= \int_a^b \sqrt{du^2 E \; + 2du\, dv\, F + dv^2 G}\; dt \\
&= \int_a^b \sqrt{\begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix}}\; dt
\end{aligned}
$$

Riemannian Manifolds

Surface Area:

- Apply integral transformation theorem:

$$\text{area}(P) = \iint_{\Omega} \|\partial_u f \times \partial_v f\| \, du \, dv$$

- Using Langrange's identity:

$$\text{area}(P) = \iint_{\Omega} \sqrt{(\partial_u f . \partial_u f)(\partial_v f . \partial_v f) - (\partial_u f . \partial_v f)^{2}} \, du \, dv$$

$$= \iint_{\Omega} \sqrt{EG - F^{2}} \, du \, dv$$

$$= \iint_{\Omega} \sqrt{\det \begin{bmatrix} E & F \\ F & G \end{bmatrix}} \, du \, dv$$

Riemannian Manifolds

Curvature:

- **Principle Curvature :** Eigenvalues of *second fundamental form* for an *orthonormal tangent* basis.

  κ_1, κ_2

- **Normal Curvature :** *norm of the projection of the derivative* $\frac{d\boldsymbol{T}}{ds}$ *on normal plane* $\mathbf{n}$.

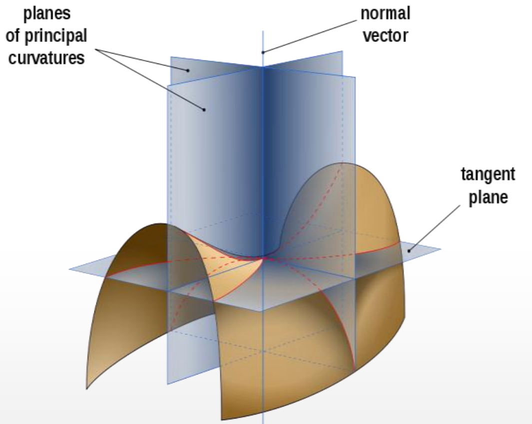
  $\kappa_N = \kappa(\mathbf{n}) = \mathbf{n}^T \begin{pmatrix} e & f \\ f & g \end{pmatrix} \mathbf{n}$ with $max(\kappa(\mathbf{n})) = \kappa_1$ and $min(\kappa(\mathbf{n})) = \kappa_2$

- **Mean Curvature :**

  $$H = \frac{1}{2}(\kappa_1 + \kappa_2) = \det\begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

- **Gaussian Curvature :**

  $$K = \kappa_1 \kappa_2 = \frac{1}{2}\mathrm{tr}\begin{pmatrix} e & f \\ f & g \end{pmatrix}$$



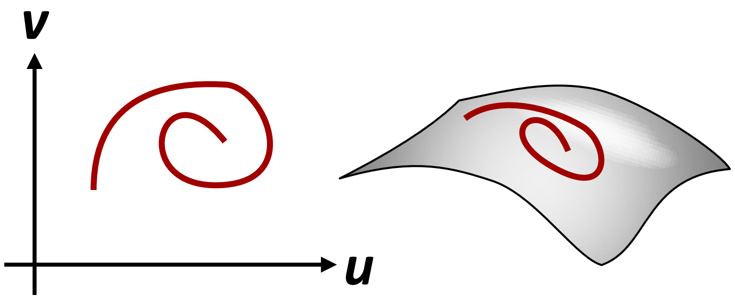
Source: Wikipedia

Riemannian Manifolds

Curvature:

- **Geodesic Curvature :** norm of the projection of the derivative $\frac{dT}{ds}$ on the tangent plane.
  - **It allows to distinguish inherent curvature of the curve in the *(u,v)* space from the curvature induced by mapping f in $\mathcal{M}$.**
- **Total Curvature :**

$$\kappa_T = \sqrt{\kappa_N^2 + \kappa_g^2}$$



- **For circles :** $\kappa_N = 1.$
- **For *Great* circles :** $\kappa_T = \kappa_N = 1$, $\kappa_g = 0$ *i.e. locally Flat e.g.* Earth
- **For *Small* circles of radius** r **:** $\kappa_g = \sqrt{1 - r^2/r}$

Riemannian Manifolds

Geodesics:

- Curves on a surface which minimize length between the end points are called *Geodesics.*

$$s = \int_a^b \sqrt{\begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix}}\, dt$$

$$energy = \int_a^b \begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} dt$$

- A path minimizing *energy* is just a geodesic parameterized by *arc length*.
- A curve on a surface with zero *geodesic curvature* is a geodesic.

Types of Mappings

Given: Riemannian Manifolds $\mathcal{M}_1, \mathcal{M}_2$

Consider: Functions $\psi: \mathcal{M}_1 \to \mathcal{M}_2$

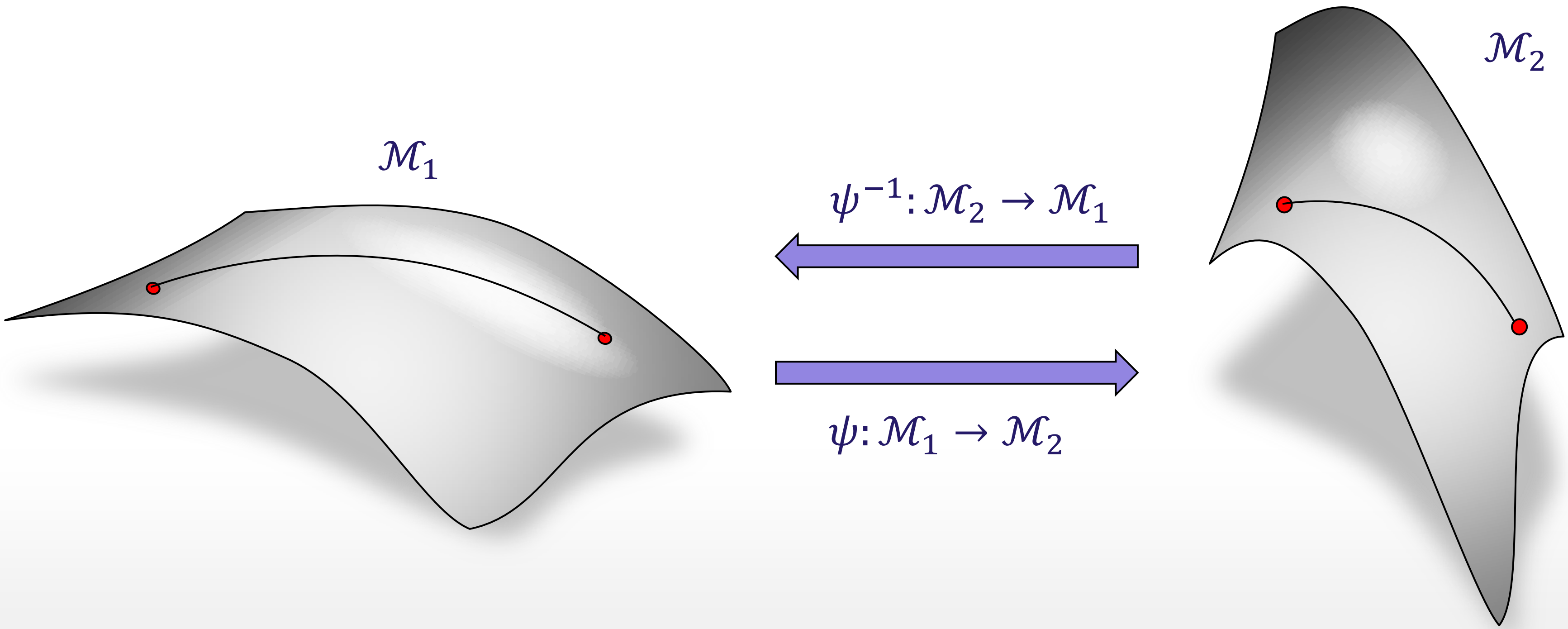
Important types of mappings:

- *Isometric*: preserves distances, angles and area
- *Conformal*: preserves (only) angles
- *Equi-areal (incompressible)*: preserves area

Isometric Mapping

Definition

- An mapping ψ between two surface patch $\mathcal{M}_1$ and $\mathcal{M}_2$ is an isometric mapping if it preserves distance on them.
- An isometric mapping is **symmetric** *i.e.* ψ^{-1} is also an isometry.

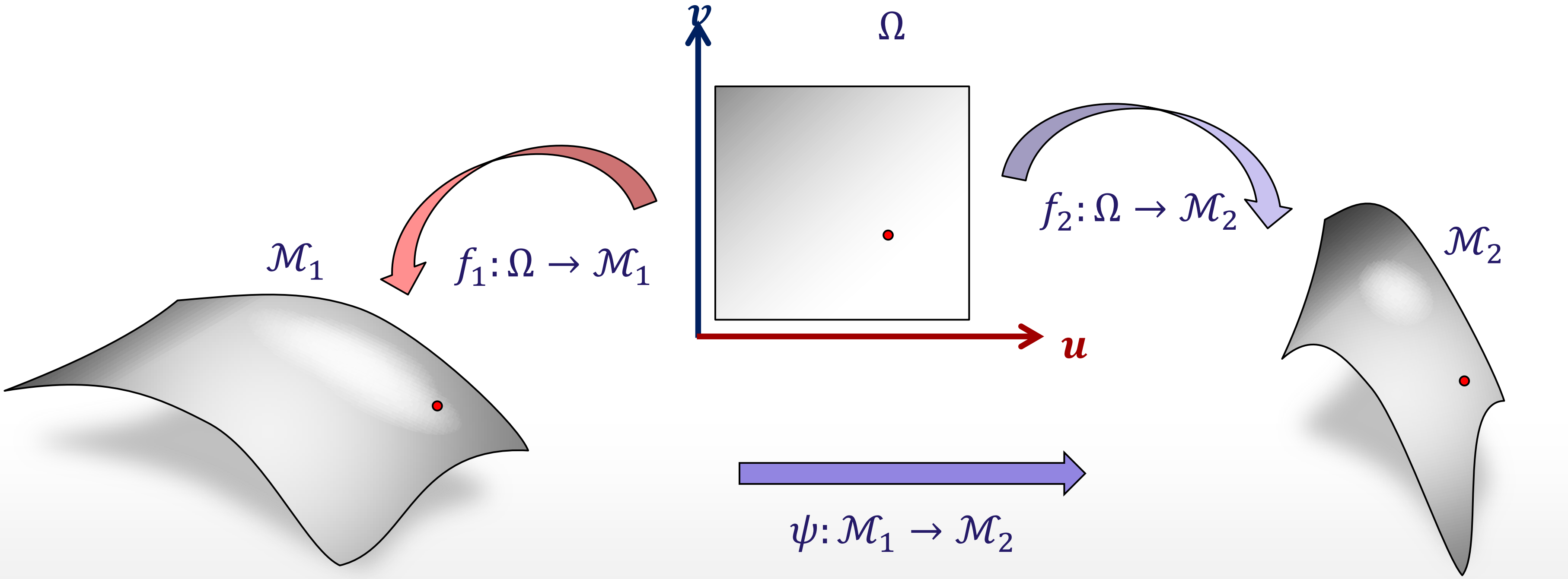


Isometric Mapping

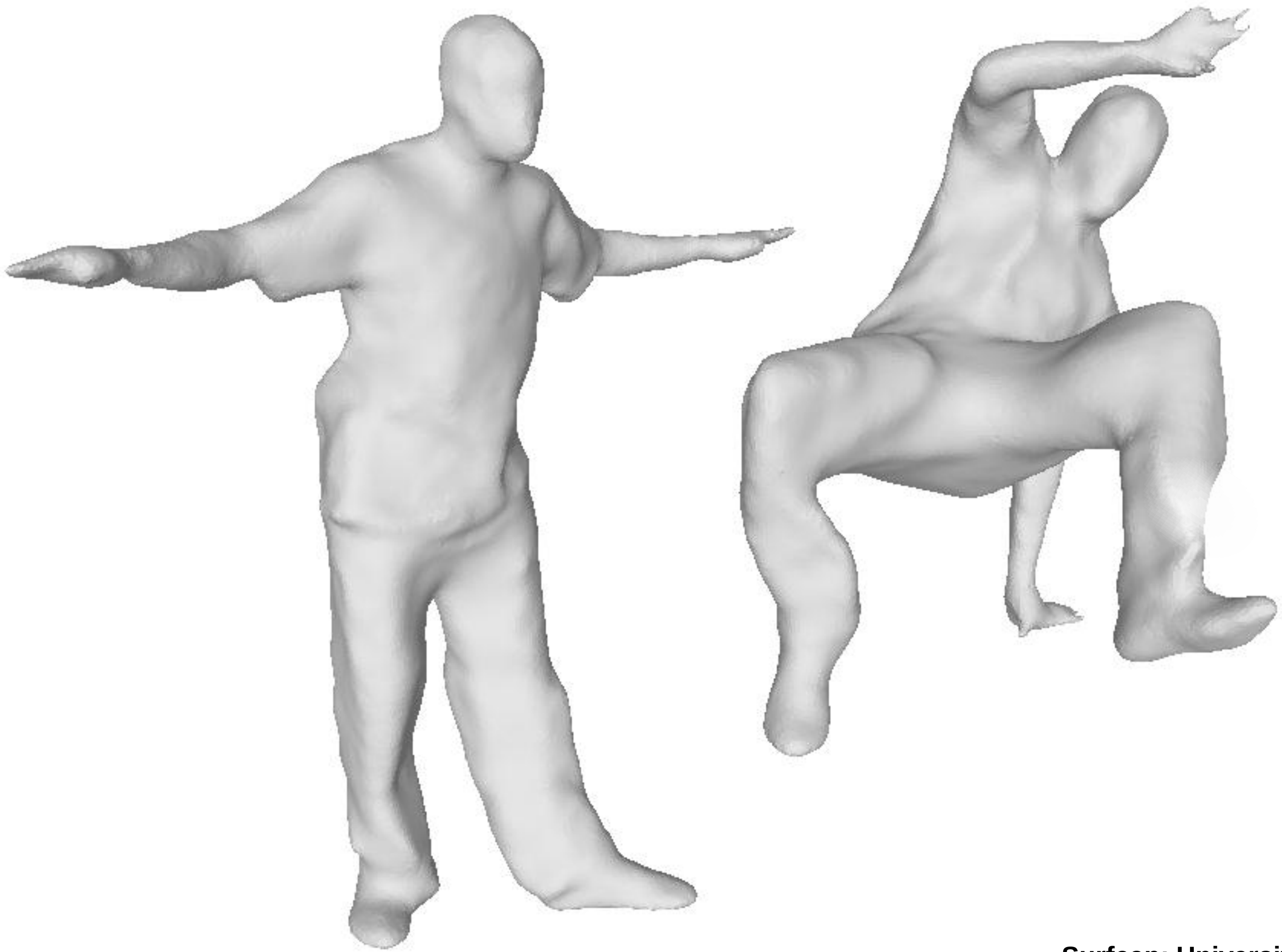
Isometric surfaces have same parametric domain

- If $\psi: \mathcal{M}_1 \to \mathcal{M}_2$ is an isometric mapping then both surfaces have same intrinsic parameterization. *i.e.*

$$f_1^{-1}(\mathcal{M}_1) = f_2^{-1}(\mathcal{M}_2) = \Omega(u, v)$$



Isometric Mapping

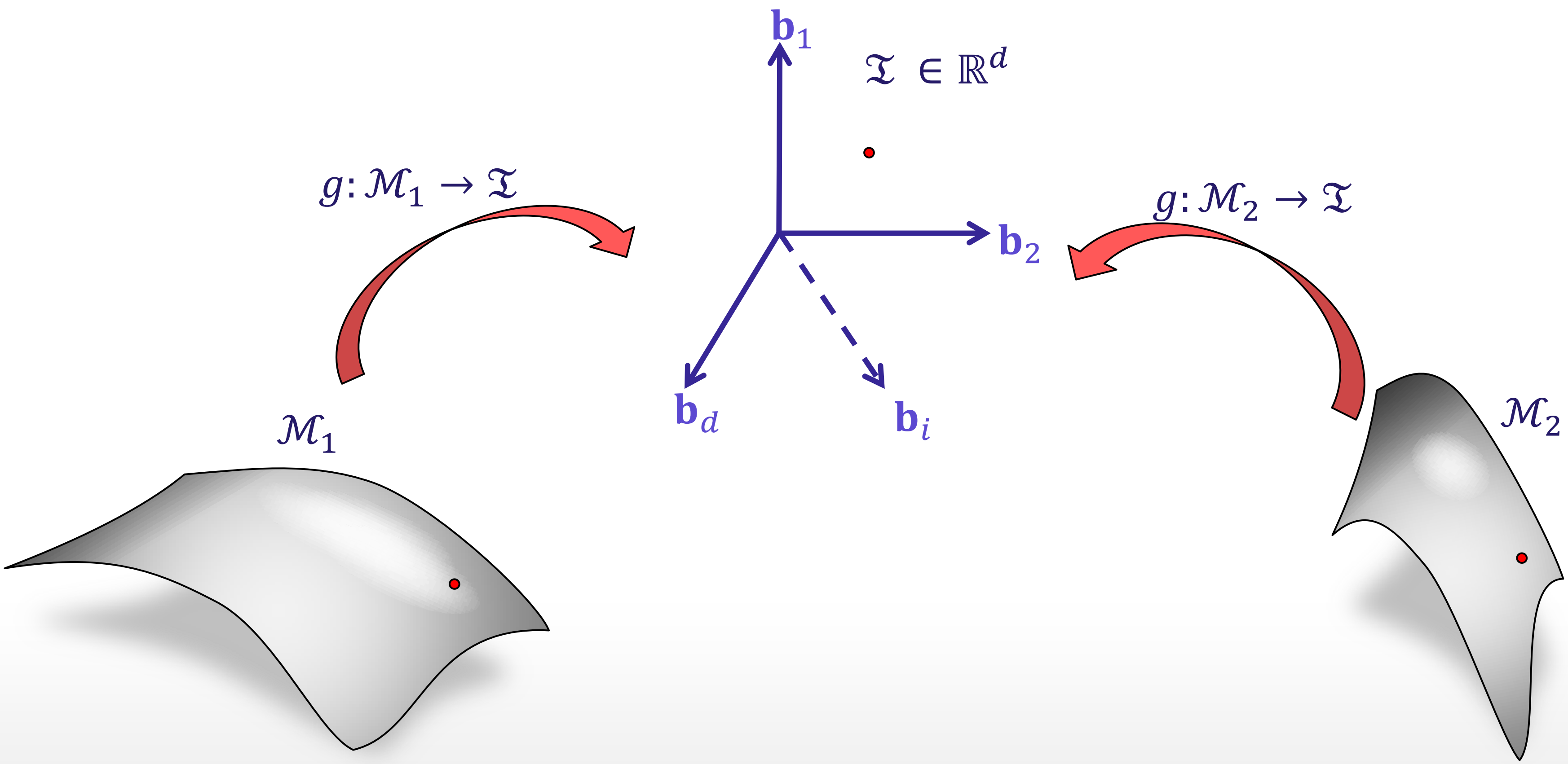


Surfcap: University of Surry

Isometric Mapping

To solve Isometry

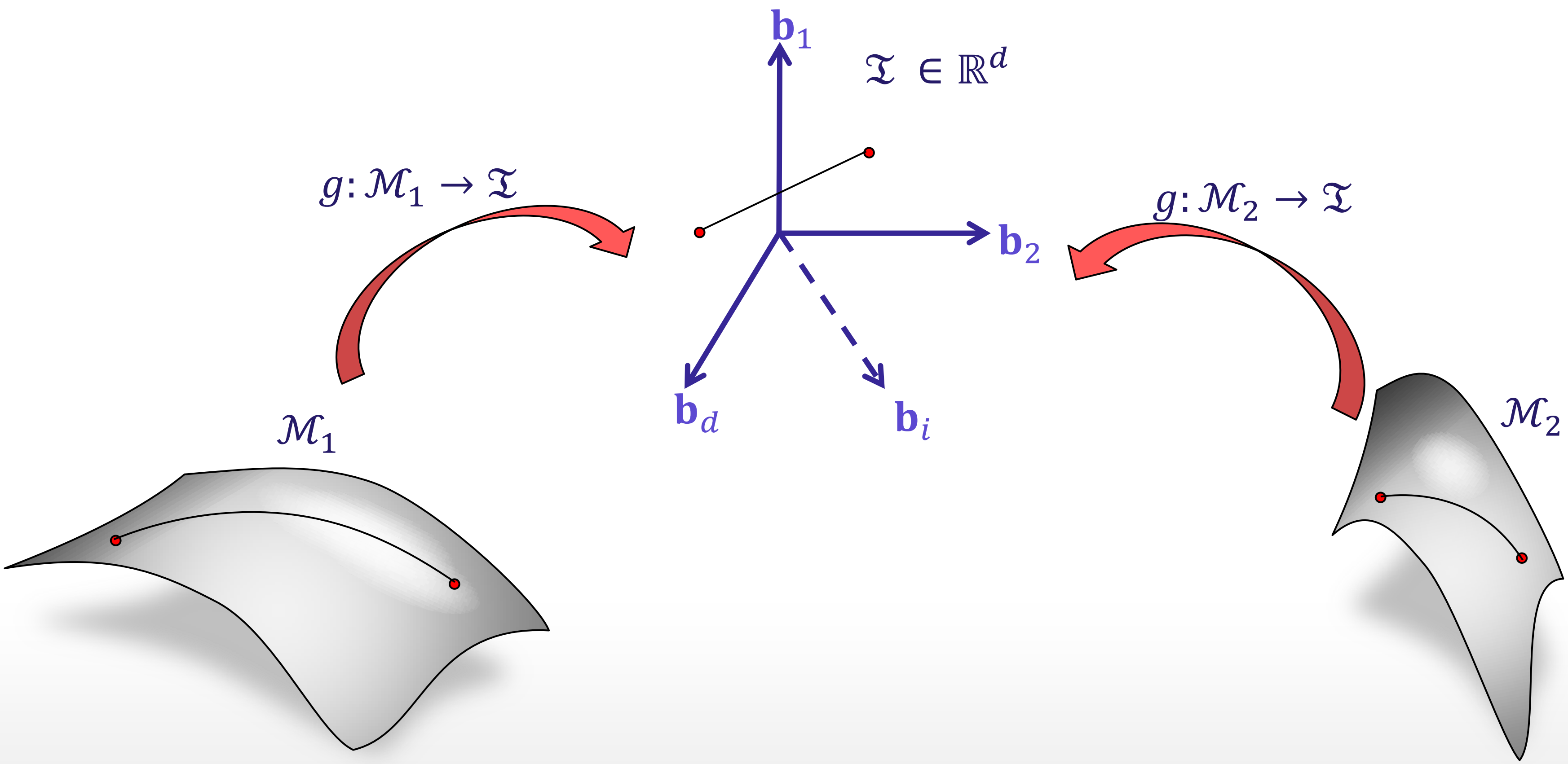
- **Either** find the explicit parameterization f_1 and f_2 to recover the implicit domain.
- **OR** find an intrinsic mapping g of $\mathcal{M}_1$ and $\mathcal{M}_2$ to an isometry invariant space $\mathfrak{T}$.



Isometric Mapping

Property of Isometry Invariant Space

- **Geodesic** distance between any pair of points in $\mathcal{M}_1$ and their images in $\mathcal{M}_2$ should be equivalent to **Euclidean** distance between their images in $\mathfrak{T}$.

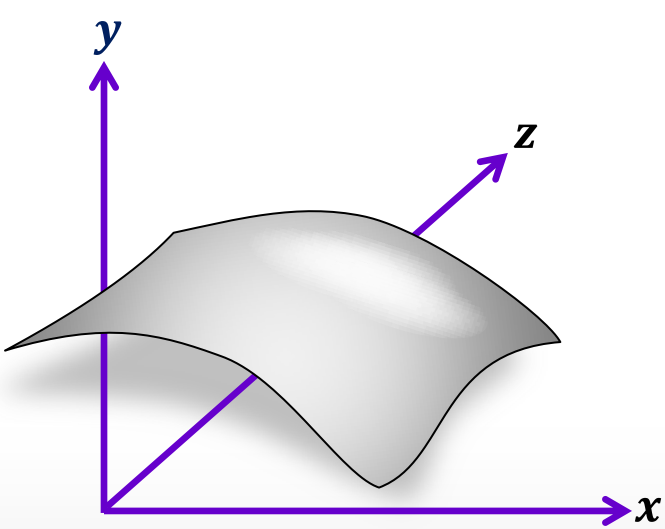


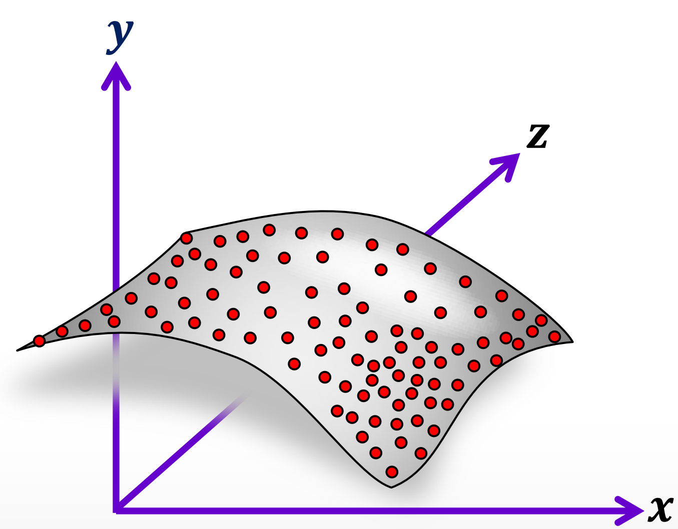
Spectral Graph Methods for 3D Shape Analysis

Discrete Manifold Representation

Surface Patches in Practice

- Most existing shape acquisition methods often yield noisy point clouds instead of a nice parametric surface representation.
- Finding a parameterization of complex real world object is practically infeasible.

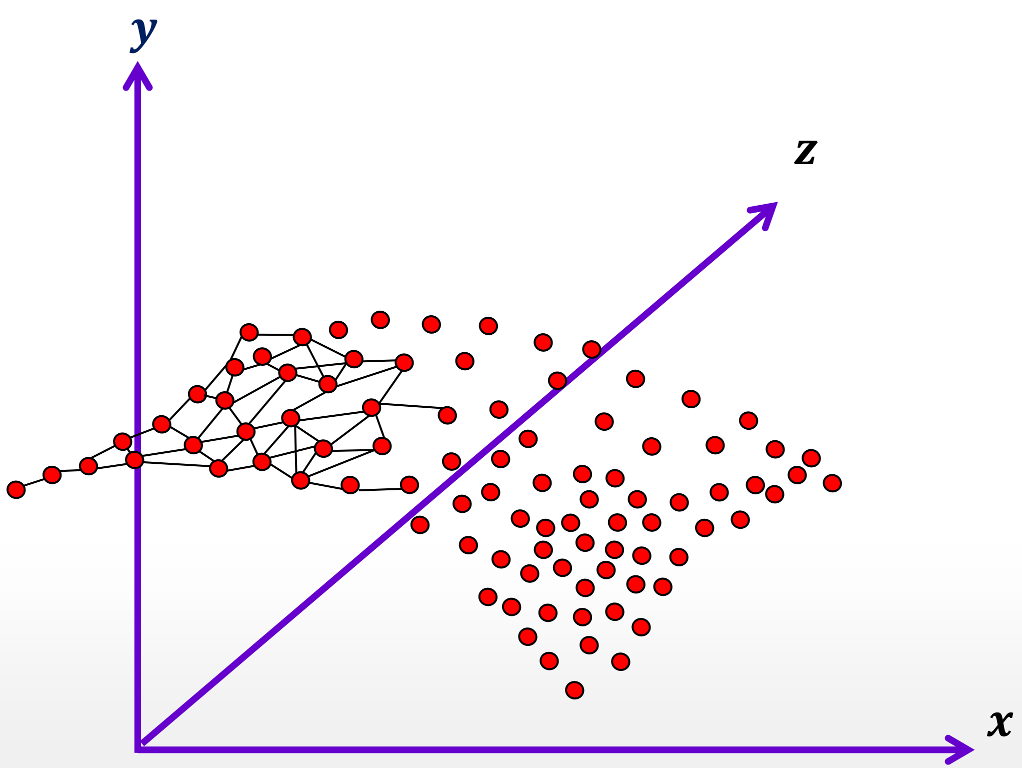




Graph Based Representation

Represent surface patch with an underlying locally connected graph structure

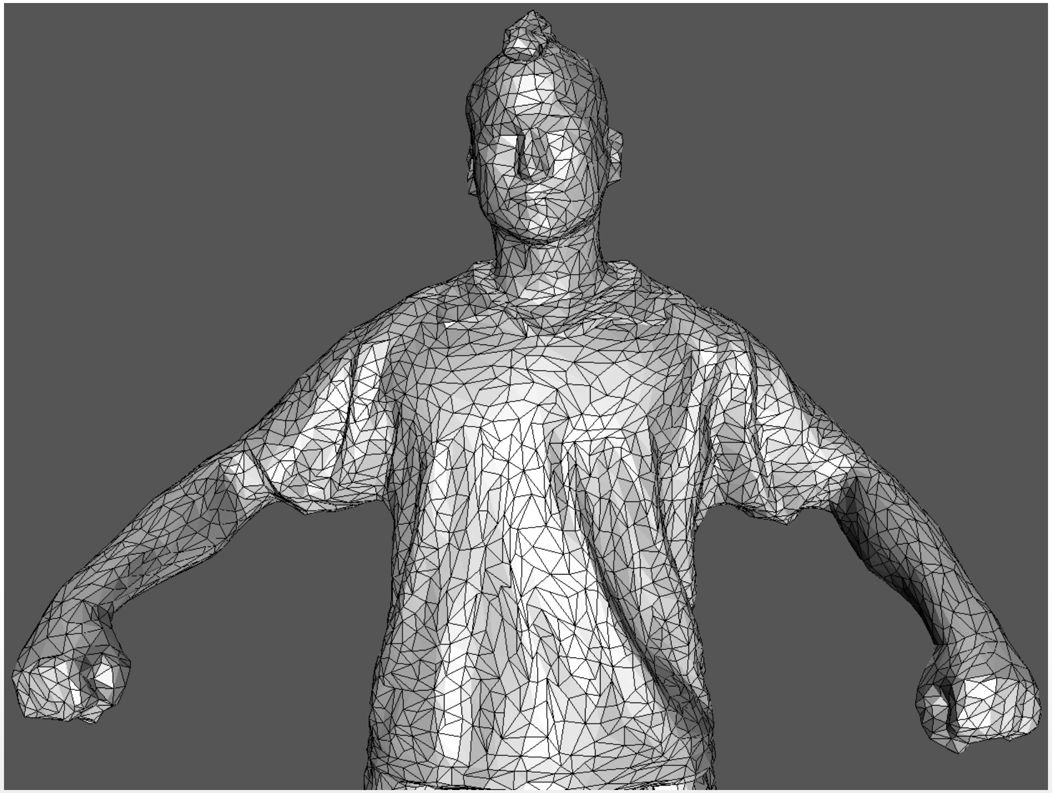
- Distances are assumed to be locally Euclidean.
- In practice we assume that Isometric transforms keep the **topology** (*i.e.* the *connectivity*) of underlying graph intact.



Graph Based Representation

Popular Representation

- **Mesh** representation of 3D object is traditionally popular.
- It enables a direct application of Graph based tools for shape analysis tasks.



Spectral Graph Theory (SGT)

Spectral Graph Theory analyze properties of graphs via eigenvalues and eigenvectors of various graph matrices.

- Builds on well studied algebraic properties of graph matrices.
- Provides a natural link between **differential operators** on continuous and discrete manifold representation.
- Allows to ***Embed*** a discrete manifold (graph) into an isometry invariant space.
- Provide intrinsic ***Spectral metric*** for isometry invariant distance computation.

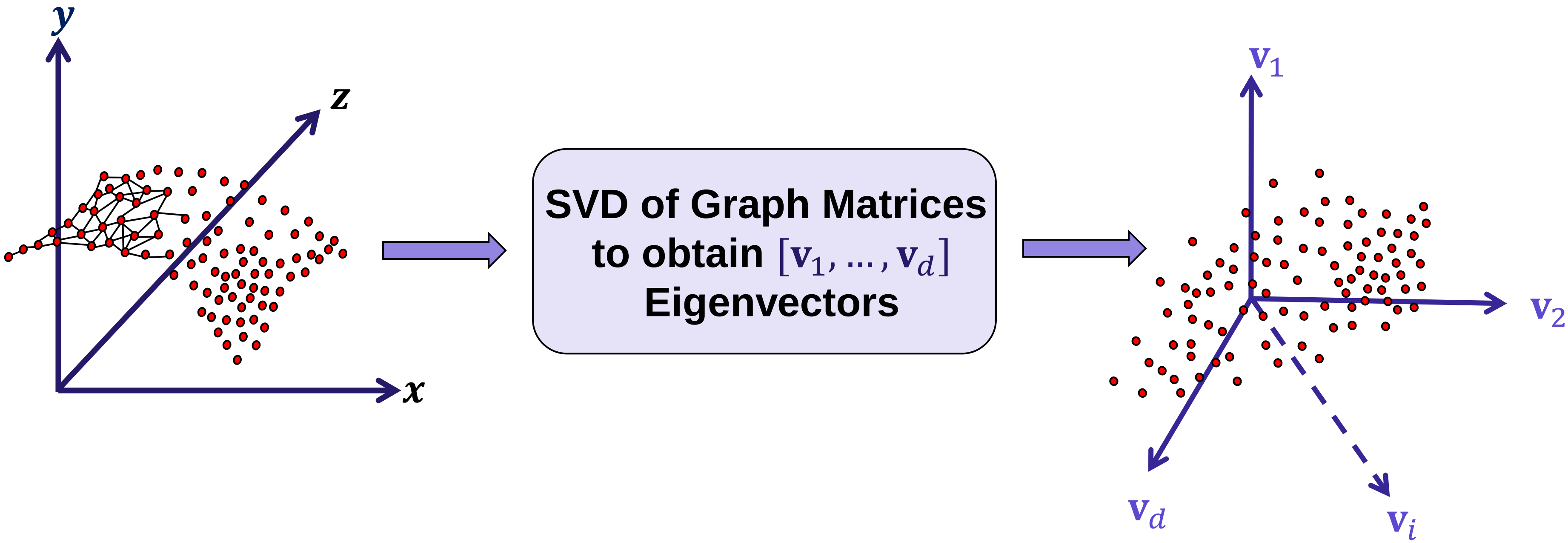
References:

- F. R. K. Chung. Spectral Graph Theory. 1997.
- M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation, 15, 1373{1396 (2003).
- U. von Luxburg. A Tutorial on Spectral Clustering. Statistics and Computing, 17(4), 395{416 (2007).*
- Software: http://open-specmatch.gforge.inria.fr/index.php.

Spectral Graph Theory (SGT)

Isometry Invariant Space

Spectral Embedding



Input $\mathbb{R}^3$ space

Output $\mathbb{R}^d$ space

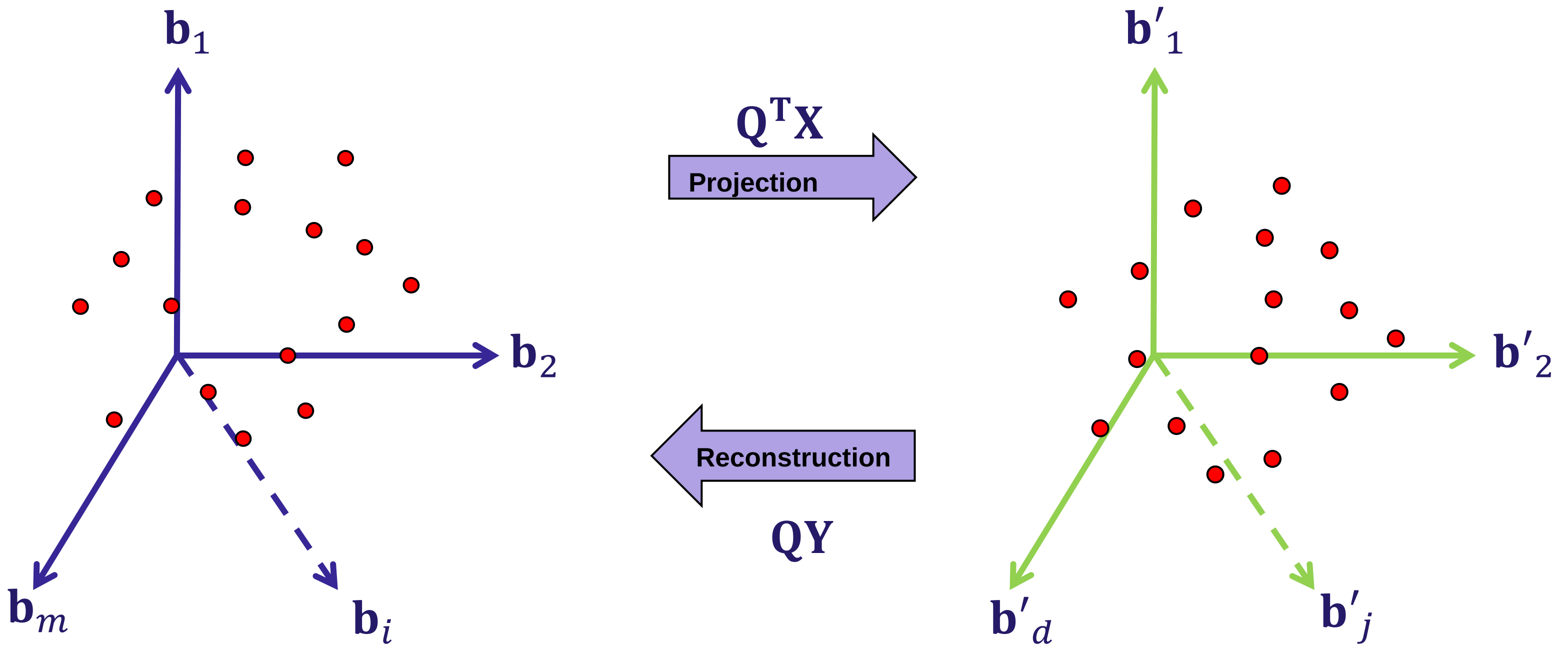
Geodesic distances in the input space $\mathbb{R}^3$ can be approximated by Euclidean distances in the new space $\mathbb{R}^d$ spanned by eigenvectors of certain Graph matrices.

* Typically $d \gg 3$, *i.e. we can now embed the surface in a higher dimensional space.*

Recall Dimensionality Reduction (DR)

Idea: Projection of higher dimensional data to a lower dimensional subspace.

- $\mathbf{X} \in \mathbb{R}^m \rightarrow \mathbf{Y} \in \mathbb{R}^d$ **where** $\boxed{d \ll m}$ ***Information is LOST**



- ***Linear*** **Dimensionality Reduction using PCA computes** $\mathbf{Q} = \mathbf{VI}_d$.

Difference between SGT and DR

In Dimensionality Reduction

- In linear DR (PCA) we consider eigendecomposition of Scatter/Covariance matrix of size ($m \times m$), where m is the original dimension of input data.
- Matrices here are typically **full** with all entries as non-zero.
- We find a least square fitting of the data.

In Spectral Graph Theory

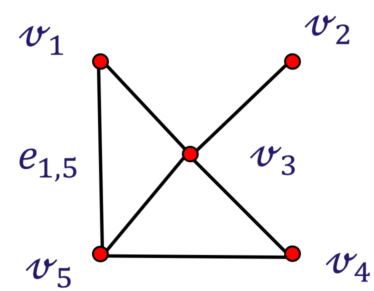
- We consider eigen-decomposition of various Graph matrices of size ($n \times n$) where n is the number of data points.
- Matrices here are typically **sparse** with very few entries as non-zero.
- We minimize a different criteria that preserve geodesic distances on the surface patch.

Graph Matrices

Basic Graph Notations

- Consider a simple graph with no loops and multiple edges $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$:
  - $\mathcal{V} = \{v_1, \dots, v_n\}$ as vertex set with its cardinality *i.e.* (number of vertices) $|\mathcal{V}| = n$.
  - $\mathcal{E} = \{e_{i,j}\}$ as edge set with each element $e_{i,j}$ representing an edge between two adjacent vertex $v_j \backsim v_i$.

Adjacency and Degree Matrix



A graph with 5 vertices and 6 edges.

- Adjacency matrix of graph $\mathcal{G}$ is a *symmetric* matrix :
  - $\mathbf{A} = [a_{i,j}]_{n\times n}$ with $a_{i,j} = \begin{cases} \omega_{i,j} & \text{if } e_{i,j} \in \mathcal{E} \\ 0 & \text{if } e_{i,j} \notin \mathcal{E} \\ 0 & \text{if } i = j \end{cases}$
  - $\omega_{i,j}$ is a weight assigned to each edge and in case of unweighted graph $\omega_{i,j} = 1$.

- Degree matrix of graph $\mathcal{G}$ is :
  - $\mathbf{D} = [diag(d_i)]_{n\times n}$ where $d_i = \sum_{\forall v_j \sim v_i} \omega_{i,j}$.
  - A graph $\mathcal{G}$ is ***connected*** if $d_i > 0, \forall i \in \{1{:}n\}$
  - A graph $\mathcal{G}$ is ***regular*** if $d_i = d_j, \forall i\, \forall j \in \{1{:}n\}$

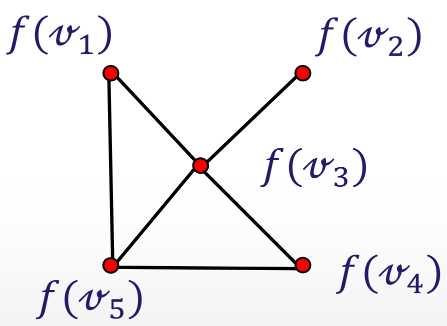
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

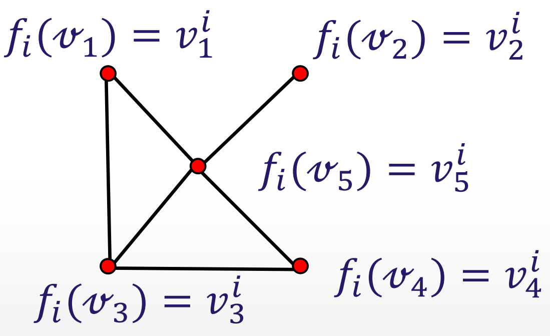
Discrete Functions on Graph

Definition

- Let f be a smooth real valued function on graph $\mathcal{G}$ such that $f\colon \mathcal{V} \to \mathbb{R}$.
  - It assign a real number to each node of the graph.
  - A discrete **vector** representation of a continuous function f is written as $\boldsymbol{f} \in \mathbb{R}^n$ where $\boldsymbol{f} = (f(v_1), \dots, f(v_n))^{\boldsymbol{T}}$.

- Eigen-decomposition of Adjacency matrix $\mathbf{A}$ yields n smooth eigenvectors that can be seen as discrete functions defined on graph vertices.
  - $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ with $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ as matrix of eigenvectors.
  - Hence $\mathbf{v}_i = (f_i(v_1), \dots, f_i(v_n))^{\boldsymbol{T}} = (v_1^i, \dots, v_n^i)^{\boldsymbol{T}}$



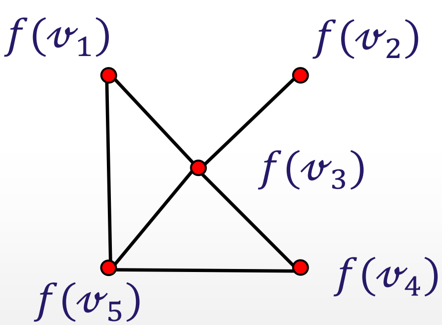


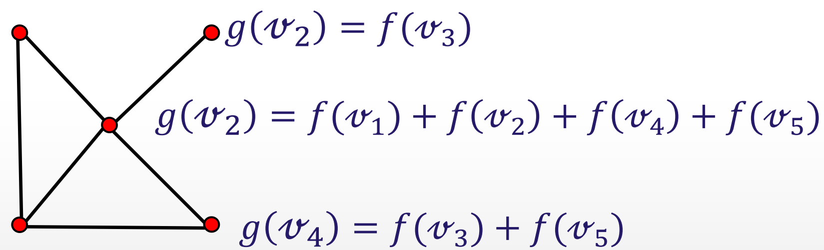
Graph Matrix as Discrete Operator

Adjacency Matrix as an Operator

- Adjacency matrix $\mathbf{A}$ can be viewed as an operator.
  - $\boldsymbol{g} = \mathbf{A}\boldsymbol{f}$ where $\boldsymbol{g}(v_i) = \sum_{\forall v_j \sim v_i} f(v_j)$
- In quadratic form we can write.
  - $\boldsymbol{f}^T\mathbf{A}\boldsymbol{f} = \sum_{e_{i,j}} f(v_i)f(v_j)$

$$\begin{bmatrix} g(v_1) \\ g(v_2) \\ g(v_3) \\ g(v_4) \\ g(v_5) \end{bmatrix} = \boldsymbol{g} = \mathbf{A}\boldsymbol{f} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} f(v_1) \\ f(v_2) \\ f(v_3) \\ f(v_4) \\ f(v_5) \end{bmatrix} = \begin{bmatrix} f(v_3) + f(v_5) \\ f(v_3) \\ f(v_1) + f(v_2) + f(v_4) + f(v_5) \\ f(v_3) + f(v_5) \\ f(v_1) + f(v_3) + f(v_4) \end{bmatrix}$$





Discrete Laplace Operator

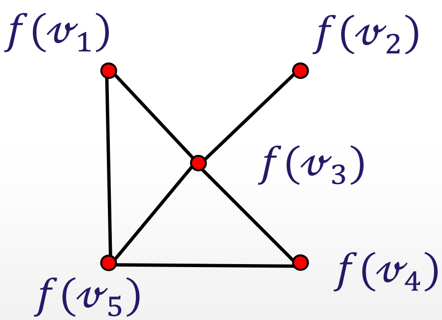
Laplacian Matrix

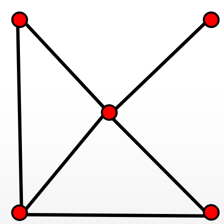
- Laplacian matrix $\mathbf{L}$ of graph $\mathcal{G}$ is a $n \times n$ *symmetric* matrix :

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

- This matrix can also be viewed as an operator :
  - $\boldsymbol{g} = \mathbf{L}\boldsymbol{f}$ where $\boldsymbol{g}(v_i) = \sum_{\forall v_j \sim v_i} \big(f(v_i) - f(v_j)\big)$

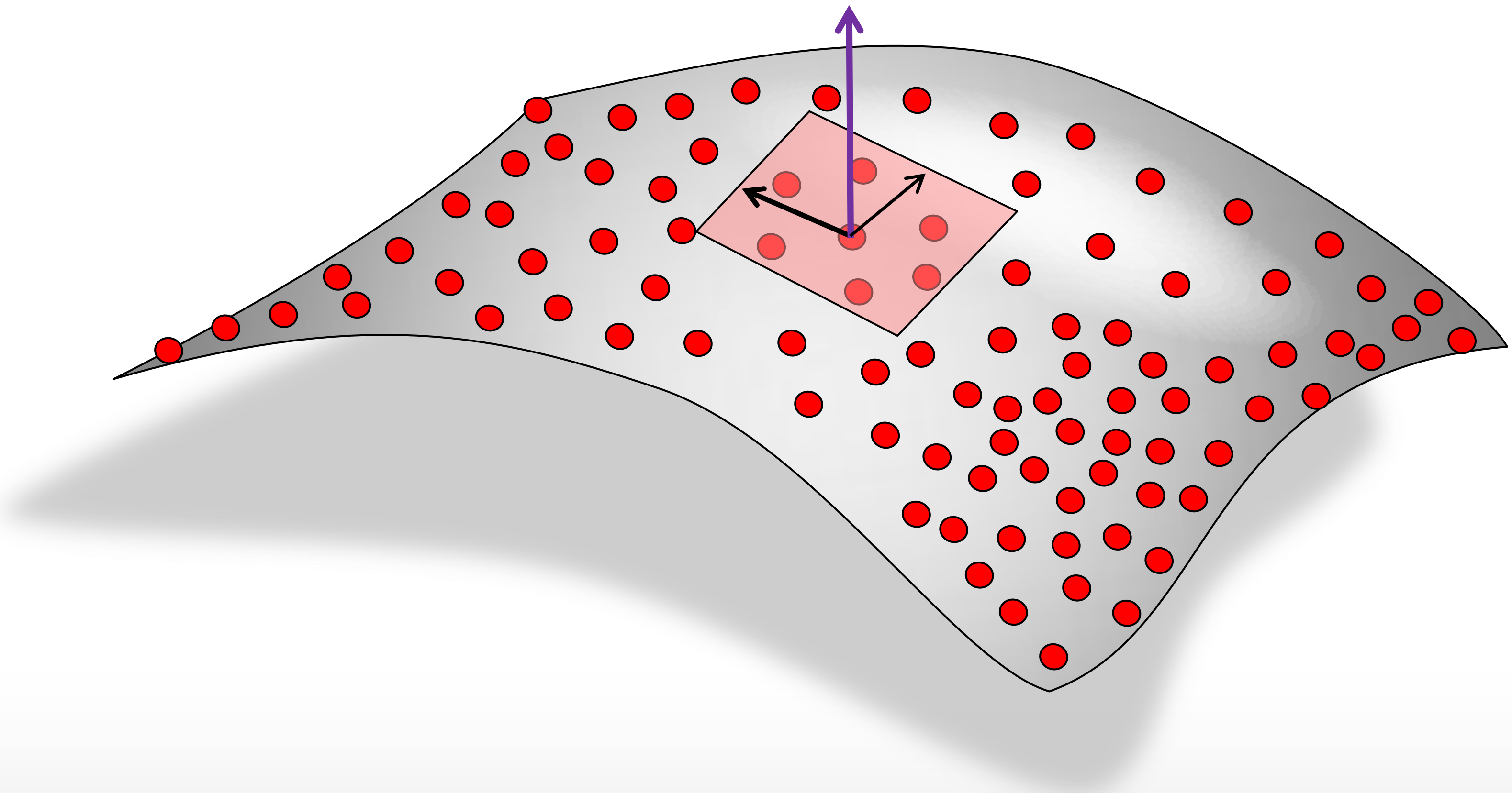
- In quadratic form we can write:
  - $\boldsymbol{f}^T \mathbf{L} \boldsymbol{f} = \sum_{\forall v_j \sim v_i} \big(f(v_i) - f(v_j)\big)^2$



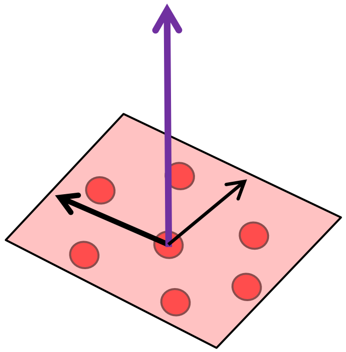


$g(v_2) = f(v_3) + f(v_5) - 2f(v_2)$

Continuous V/s Discrete Laplace



Local Parametrization



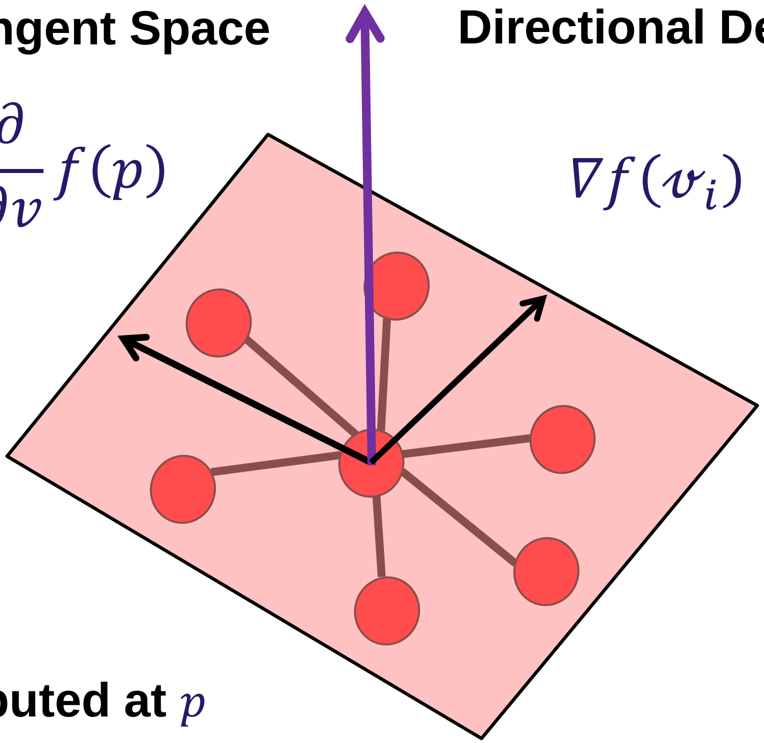
Continuous V/s Discrete Laplace

Derivative vector in Tangent Space

$$\nabla f(p) = \frac{\partial}{\partial u} f(p) + \frac{\partial}{\partial v} f(p)$$

Directional Derivative w.r.t. edge $e_{i,j}$

$$\nabla f(v_i) = f(v_i) - f(v_j)$$



Laplace operator computed at p

$$\Delta f(p) = \operatorname{div} \nabla f(p) = \nabla . \nabla f(p)$$

$$\Delta f(p) = \frac{\partial^2}{\partial u^2} f(p) + \frac{\partial^2}{\partial v^2} f(p)$$

Laplace operator computed at v_i

$$\Delta f(v_i) = \operatorname{div} \nabla f(v_i) = \nabla^T \nabla f(v_i)$$

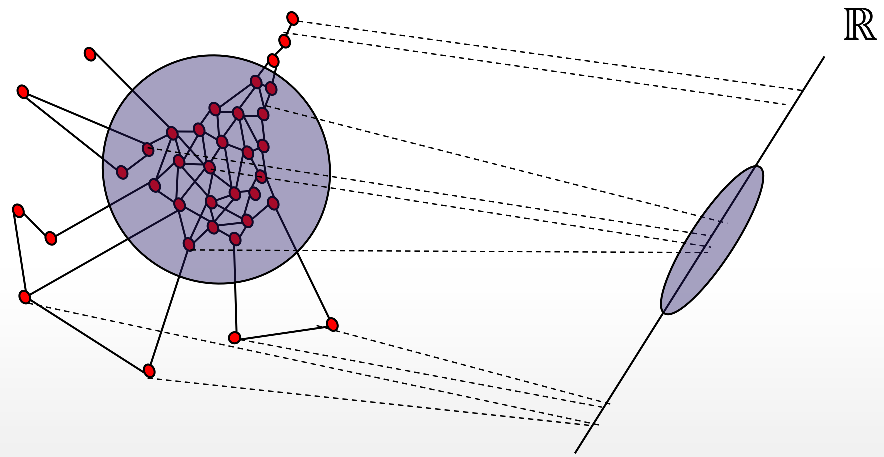
$$\mathbf{L} f(v_i) = \sum_{\forall j \sim i} f(v_i) - f(v_j)$$

Discrete Laplace Operator

Laplacian Operator

- Minimization of quadratic form $\boldsymbol{f}^T \mathbf{L} \boldsymbol{f} == \sum_{\forall v_j \sim v_i} \left(f(v_i) - f(v_j)\right)^2$ over $\boldsymbol{f}$ yield a smooth function that maps neighboring vertices of $\mathcal{G}$ together.
  - Hence it is desired that $f(v_i)$ and $f(v_j)$ are mapped closer on the real line.

- One important consequence of this is that if a graph is *not* ***regular*** (i.e. *not uniformly connected*) then a set of ***stongly*** connected vertices will be mapped closer as compare to set of ***weakly*** connected vertices.



Laplacian Eigenvectors

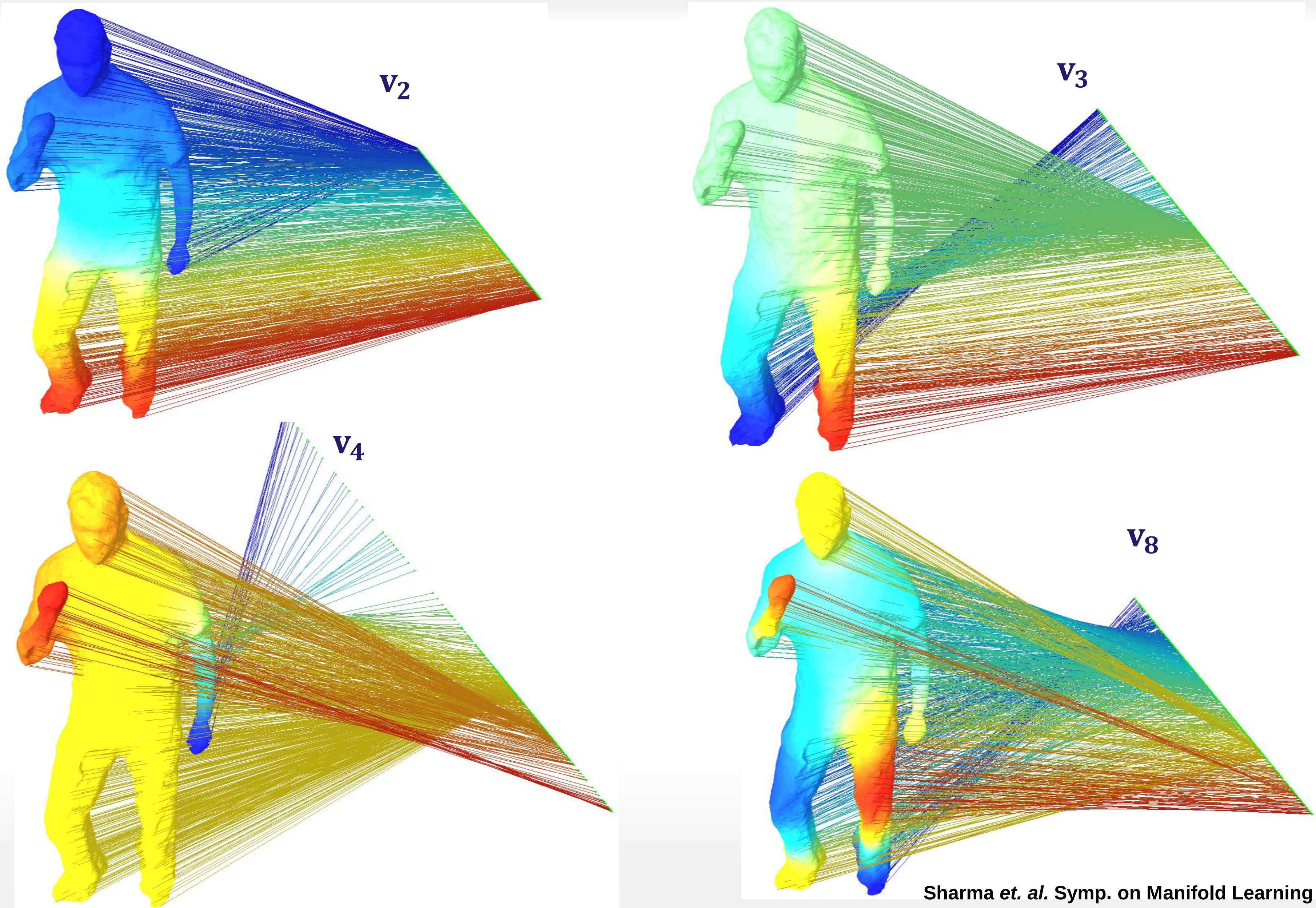
Laplacian Eigenvectors

- Minimization of quadratic form $\boldsymbol{f}^T \mathbf{L} \boldsymbol{f}$ over $\boldsymbol{f}$ can be written as **Rayleigh quotient** :

$$\min_{\boldsymbol{f}} \frac{\boldsymbol{f}^T \mathbf{L} \boldsymbol{f}}{\boldsymbol{f}^T \boldsymbol{f}}$$

- The solution of minimization is a family of smooth orthonormal functions *i.e.* eigen-functions of $\mathbf{L}$ matrix corresponding to increasing eigenvalues.
- $\mathbf{LV} = \mathbf{\Lambda V}$ with $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ and $\mathbf{\Lambda} = \mathbf{diag}(\lambda_1, \dots, \lambda_n)$ such that $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \dots \leq \lambda_n$.
- $\mathbf{L1} = \mathbf{0}$, $\lambda_1 = 0$ (If graph is connected). Trivial solution as *constant vector.*
- $\mathbf{Lv}_2 = \lambda_2 \mathbf{v}_2$ is called the Fiedler vector.
- $\forall i \in \{2{:}\,n\}$, $\mathbf{v}_i^{\mathrm{T}} \mathbf{1} = \mathbf{0}$ by orthonormal property of eigenvectors.
- Hence, $\sum_{j=1:n} \mathbf{v}_i(j) = 0$.

Laplacian Eigenvectors



Sharma *et. al.* Symp. on Manifold Learning 2009

Types of Laplace Discretization

$$\boldsymbol{g} = \mathbf{L}\boldsymbol{f} \text{ where } \boldsymbol{g}(v_i) = \sum_{\forall v_j \sim v_i} \omega_{ij}\big(f(v_i) - f(v_j)\big)$$

Binary Weighting (*Non Geometric*)

- Weights of adjacency matrix are set to 0 or 1, *i.e.* all edges are equally weighted. Also known as ***Umbrella*** operator.

Gaussian Weighting

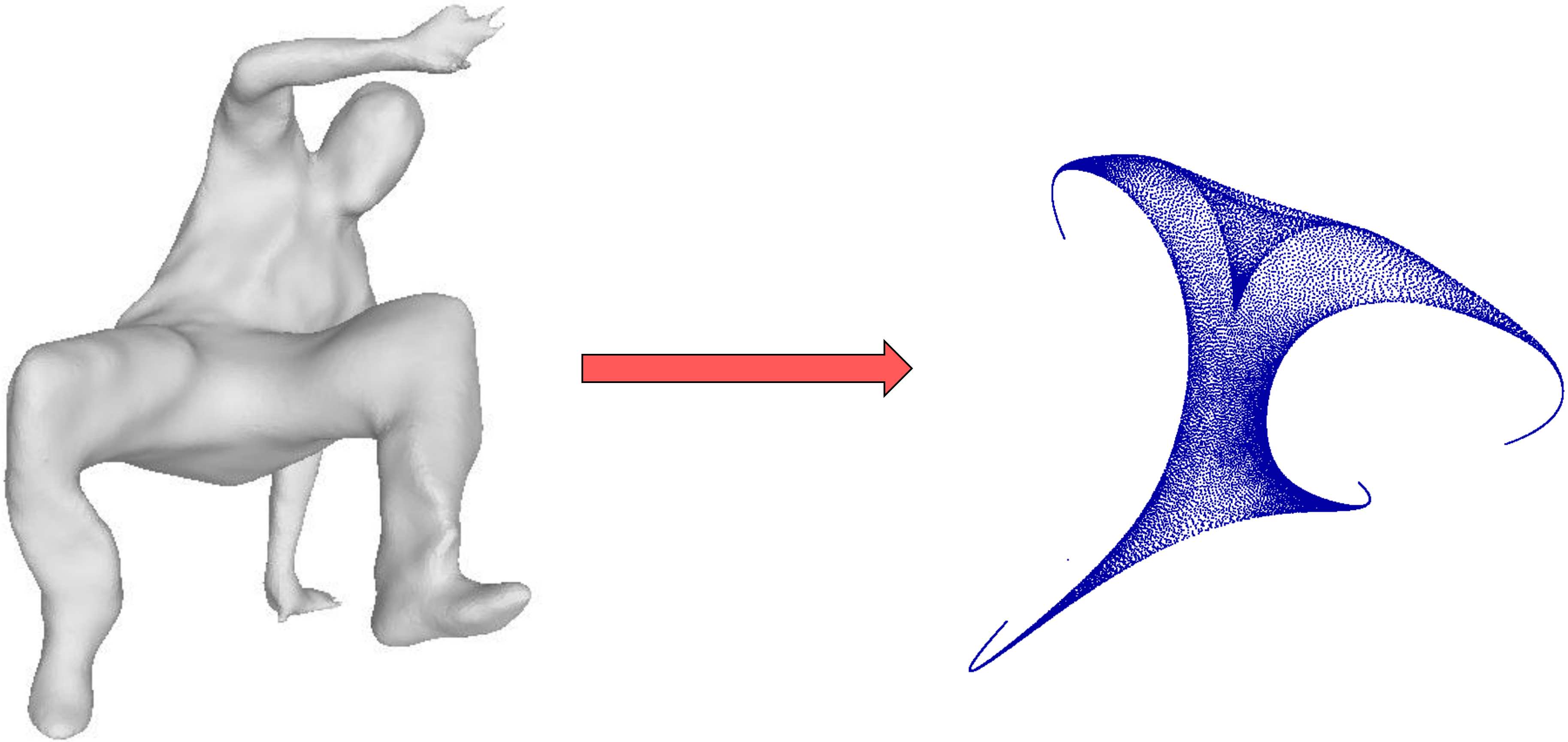
- Weights of adjacency matrix are set to $\mathcal{N}_{\mu,\sigma}\left(-\mathrm{Euclid}(v_i, v_j)\right)$, *i.e.* edges with small length are weighted more than the larger once.

Cotangent Weighting

- Weights of adjacency matrix are computed in terms of cotangent of angles of triangle in the triangulated mesh, *i.e.* areas with higher curvature are weighted *more*.

Discrete Laplace operators: No free lunch, M. Wardetzky, S. Mathur, F. Kälberer and E. Grinspun, In SGP 2007

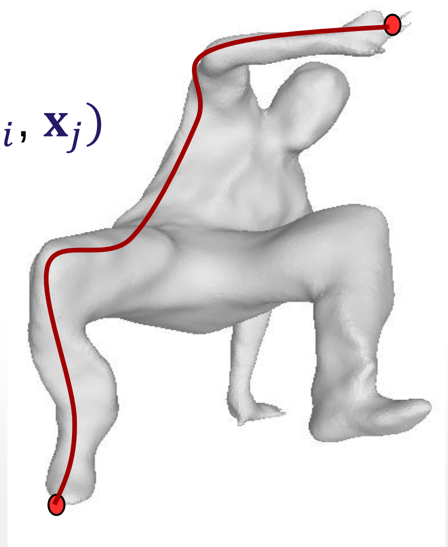
Laplacian Embedding

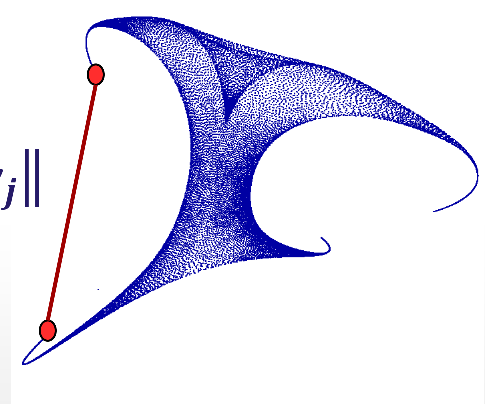


Spectral Metric

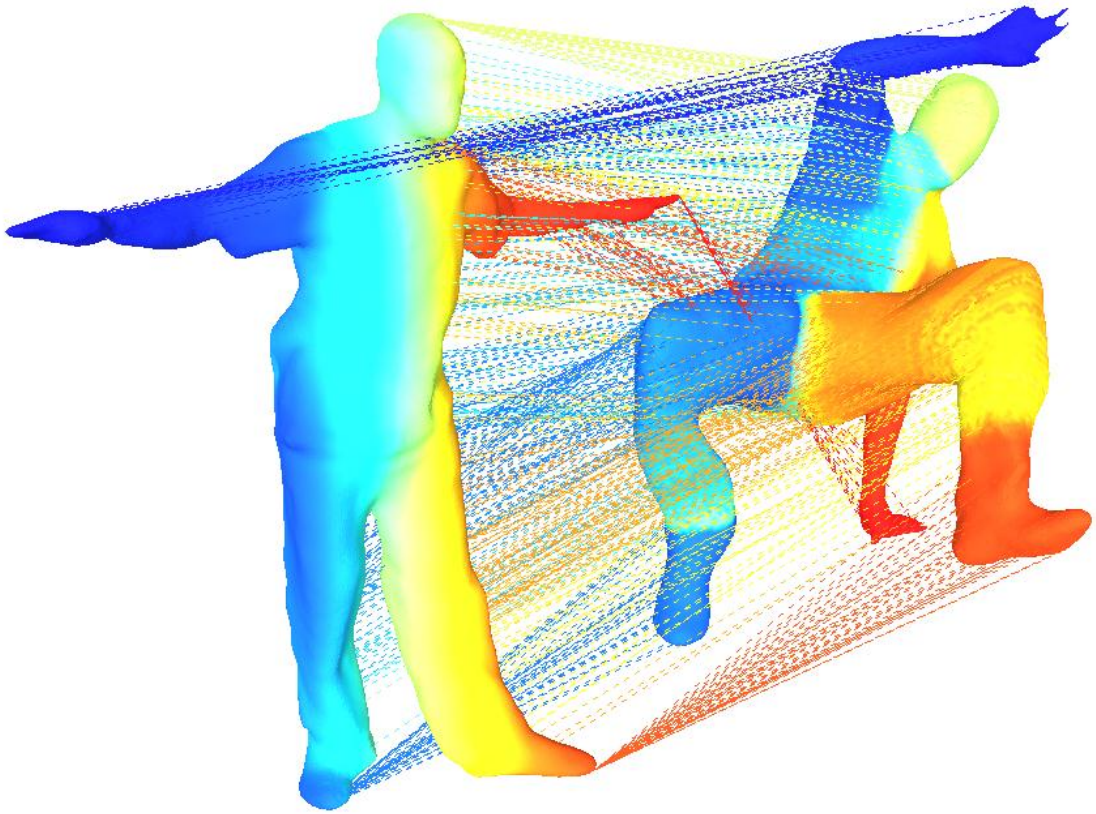
Distance computation on surface

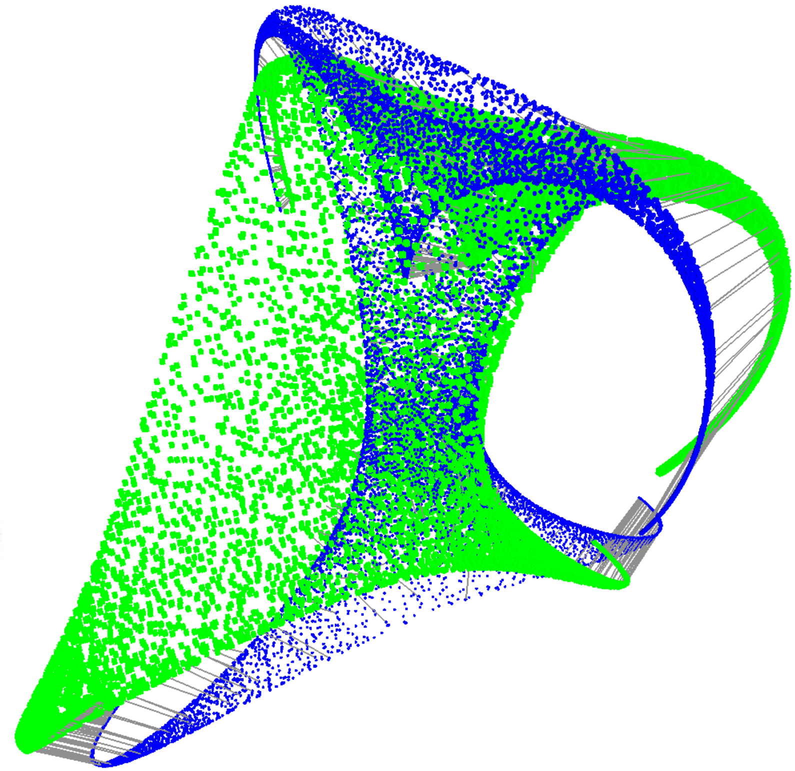
- Given a 3D shape represented as $\mathbf{X} \in \mathbb{R}^3$ and its d-dimensional embedding as $\mathbf{Y} = [\mathbf{v}_2, \dots, \mathbf{v}_{d+1}]^{\mathrm{T}}$.
- Each point $\mathbf{x}_i \in \mathbf{X}$ is represented as $\boldsymbol{y}_i = (\mathbf{v}_2(i), \dots, \mathbf{v}_{d+2}(i))^{\mathrm{T}}$.
- Geodesic$(\mathbf{x}_i, \mathbf{x}_j) \approx \|\boldsymbol{y}_i - \boldsymbol{y}_j\| = \sqrt{\sum_{k=2}^{d+1}(\mathbf{v}_k(i) - \mathbf{v}_k(j))^2}$.





Spectral Matching





Mateus *et. al.* CVPR 2008

Questions?