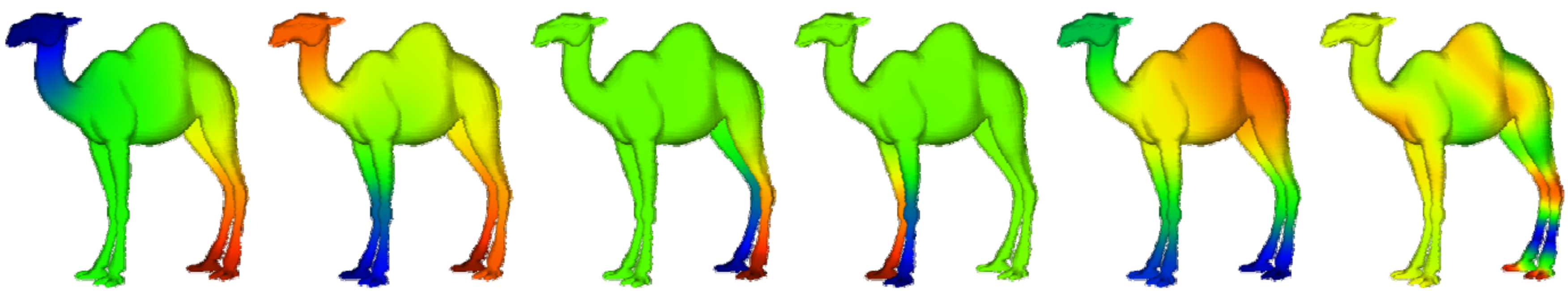
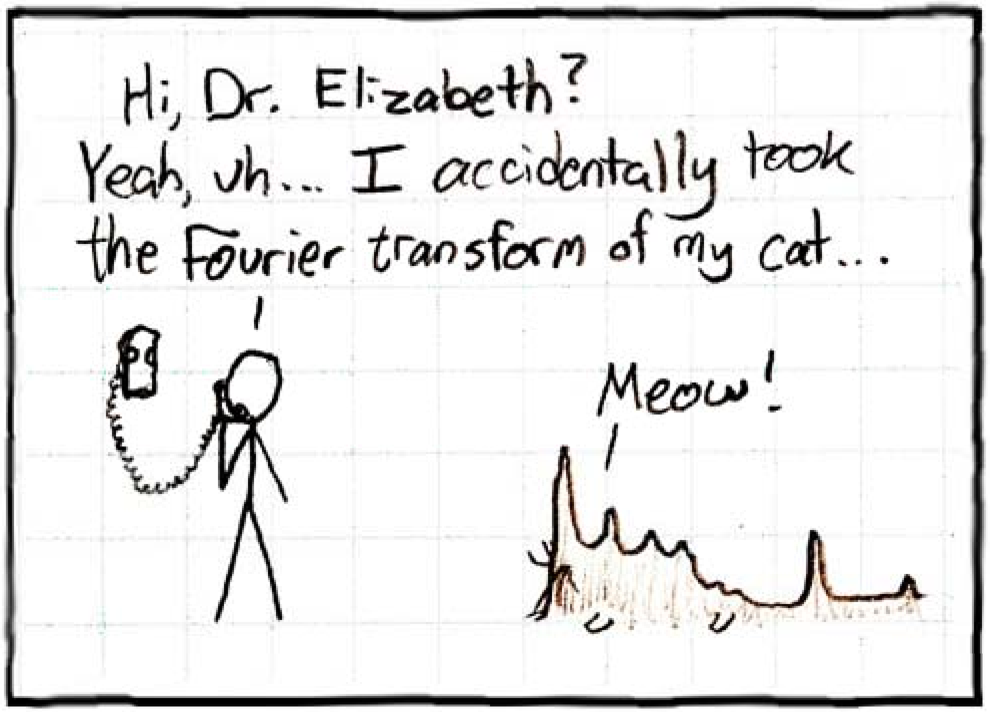
Spectral Algorithms I

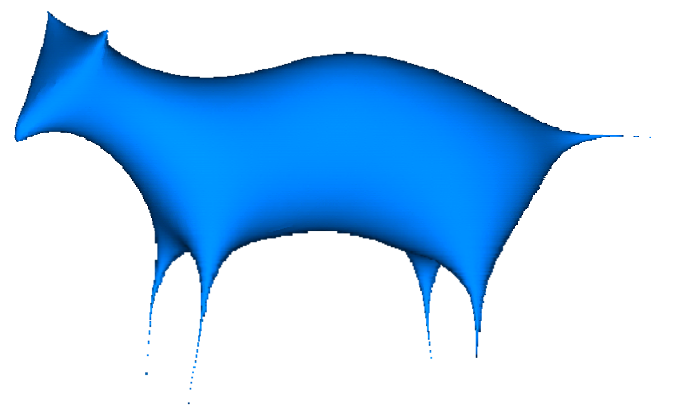


Slides based on “*Spectral Mesh Processing*” Siggraph 2010 course

Why Spectral?

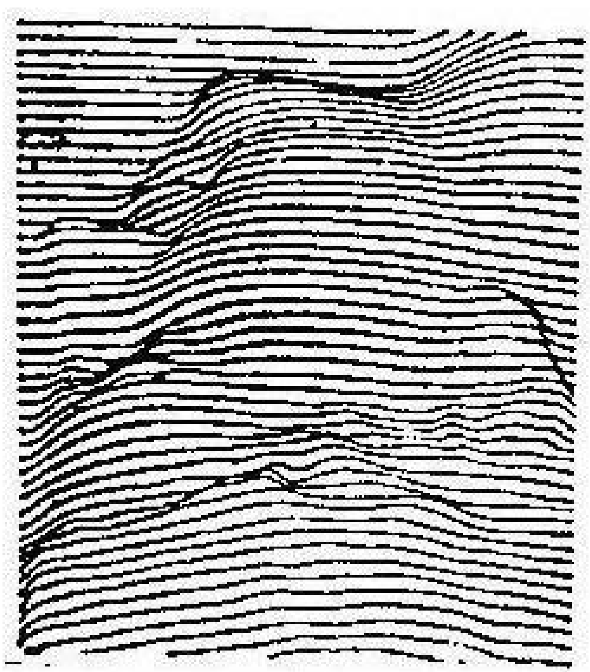
A different way to look at functions on a domain

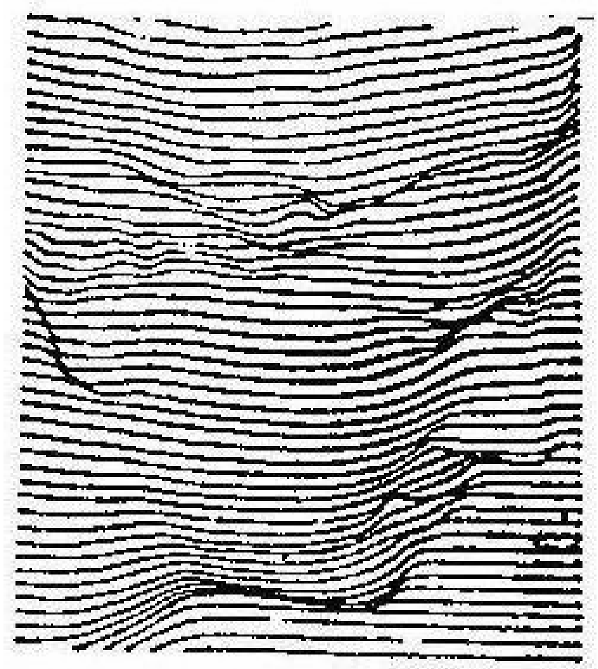




Why Spectral?

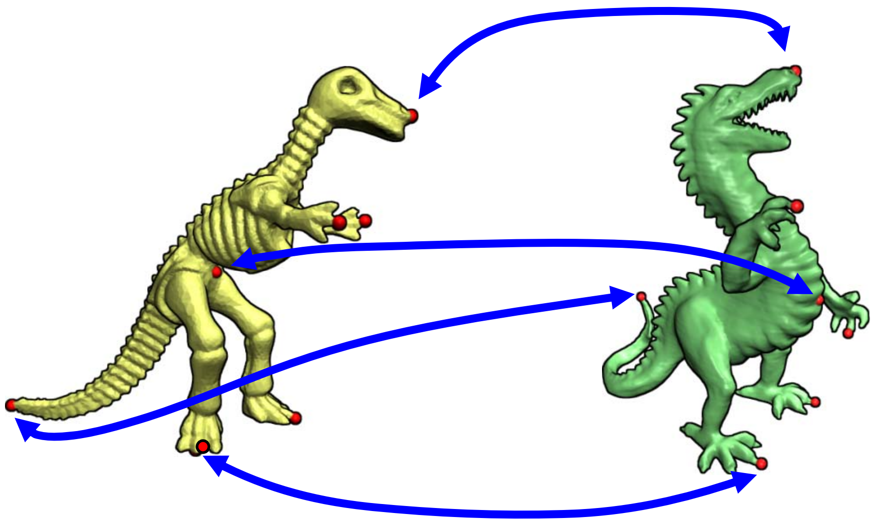
Better representations lead to simpler solutions



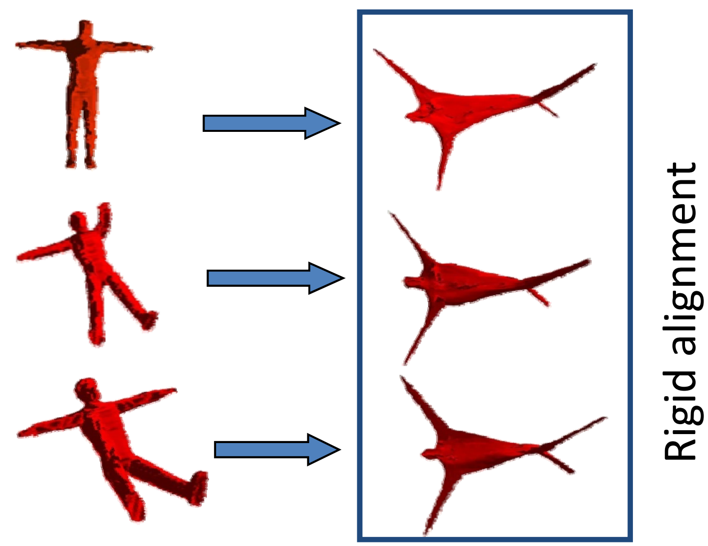


A Motivating Application

Shape Correspondence



- **Rigid alignment easy - different pose?**
- **Spectral transform** normalizes shape pose



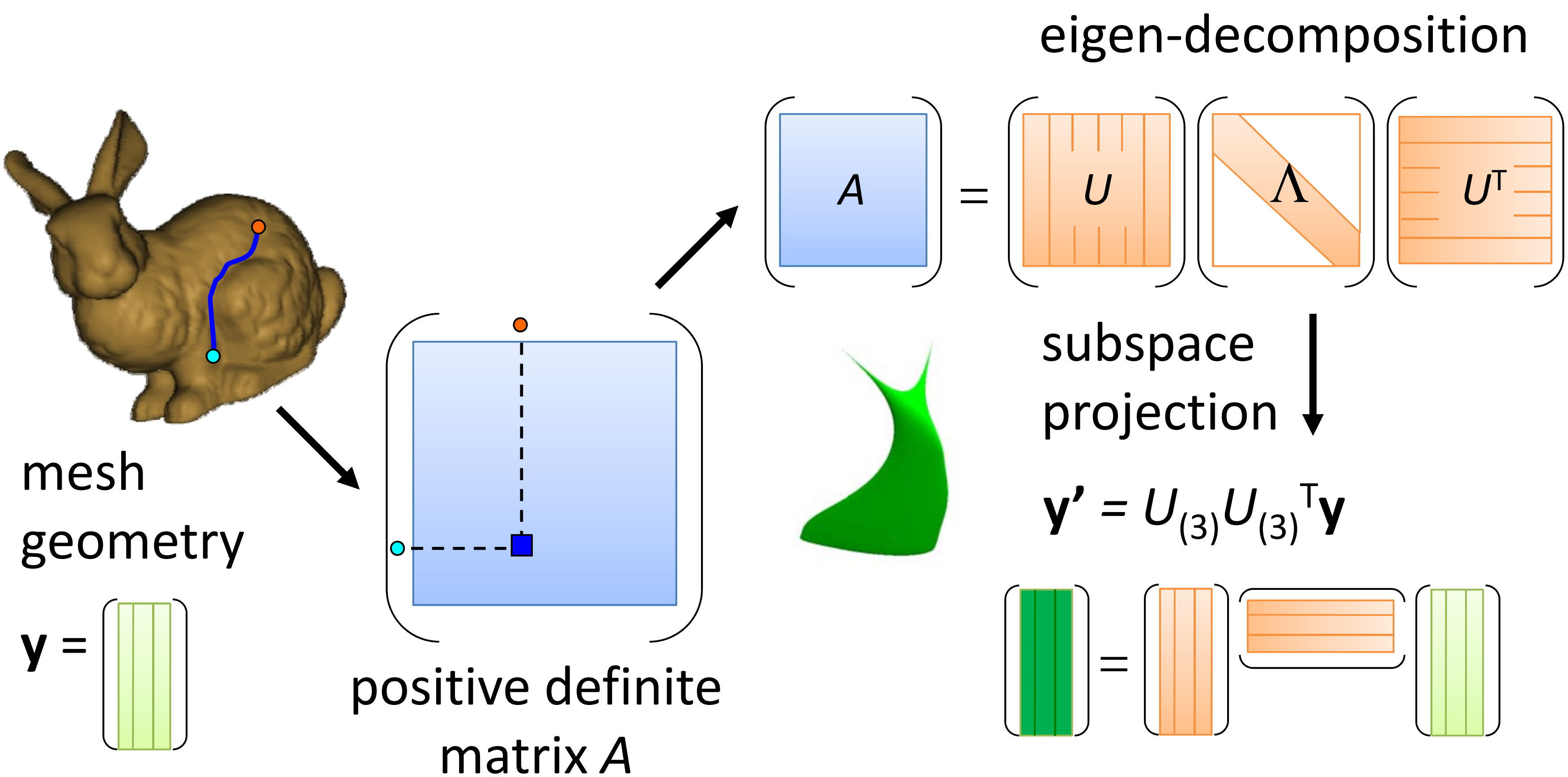
Spectral Geometry Processing

Use *eigen-structure*
of “well behaved” *linear operators*
for geometry processing

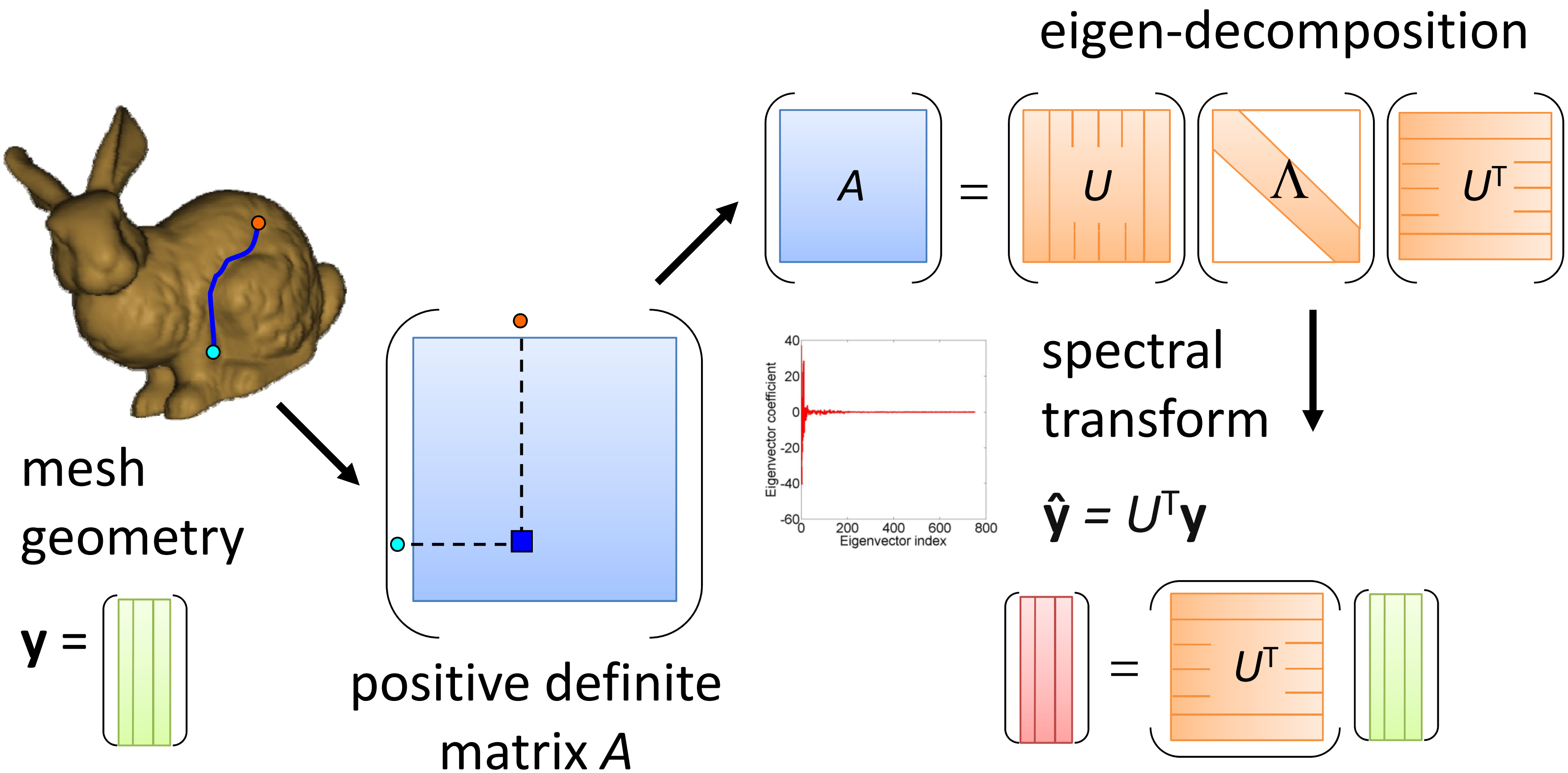
Eigen-structure

- Eigenvectors and eigenvalues $\boxed{A\mathbf{u} = \lambda\mathbf{u},\ \mathbf{u} \neq 0}$
- Diagonalization or eigen-decomposition $\boxed{A = U\Lambda U^{\mathrm{T}}}$
- Projection into eigen-subspace $\boxed{\mathbf{y}' = U_{(k)}U_{(k)}{}^{\mathrm{T}}\mathbf{y}}$
- DFT-like spectral transform $\boxed{\hat{\mathbf{y}} = U^{\mathrm{T}}\mathbf{y}}$

Eigen-structure



Eigen-structure



Classification of Applications

- Eigenstructure(s) used
  - Eigenvalues: **signature** for shape characterization
  - Eigenvectors: form **spectral embedding** (a transform)
  - Eigenprojection: also a transform — **DFT-like**

- Dimensionality of spectral embeddings
  - 1D: mesh sequencing
  - 2D or 3D: graph drawing or mesh parameterization
  - Higher D: clustering, segmentation, correspondence

- Mesh operator used
  - Laplace Beltrami, distances matrix, other
  - Combinatorial vs. geometric
  - 1^{st}-order vs. higher order
  - Normalized vs. un-normalized

Operators?

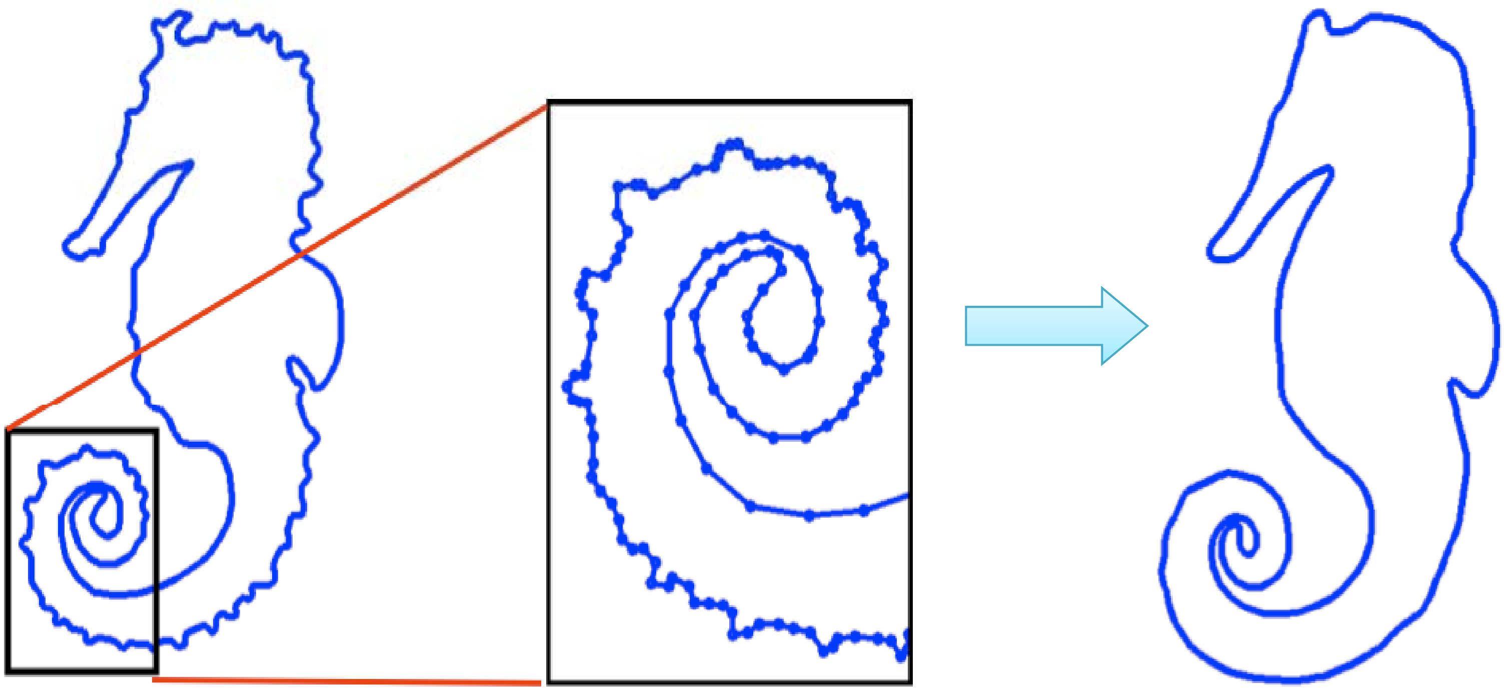
- **Best**
  - Symmetric positive definite operator: $x^TAx > 0$ for any x
- **Can live with**
  - Semi-positive definite ($x^TAx \geq 0$ for any x)
  - Non symmetric, as long as eigenvalues are real and positive e.g.: $L = DW$, where W is SPD and D is diagonal.
- **Beware of**
  - Non-square operators
  - Complex eigenvalues
  - Negative eigenvalues

Spectral Processing - Perspectives

- Signal processing
  - Filtering and compression
  - Relation to discrete Fourier transform (DFT)
- Geometric
  - Global and intrinsic
- Machine learning
  - Dimensionality reduction

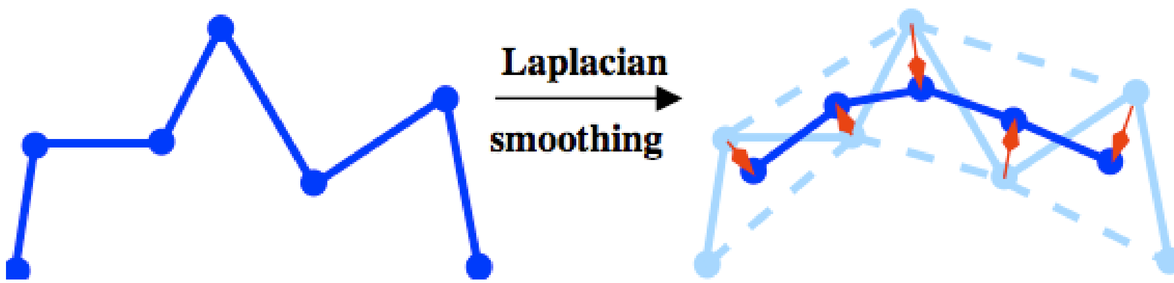
The smoothing problem

Smooth out rough features of a contour (2D shape)



Laplacian smoothing

Move each vertex towards the centroid of its neighbours

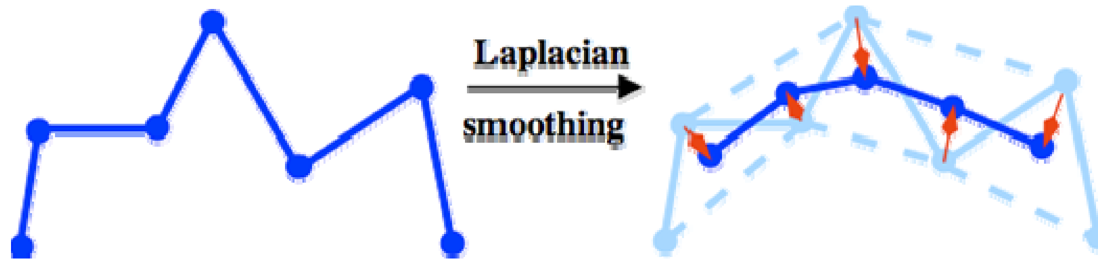


Here:

- Centroid = midpoint
- Move half way

Laplacian smoothing and Laplacian

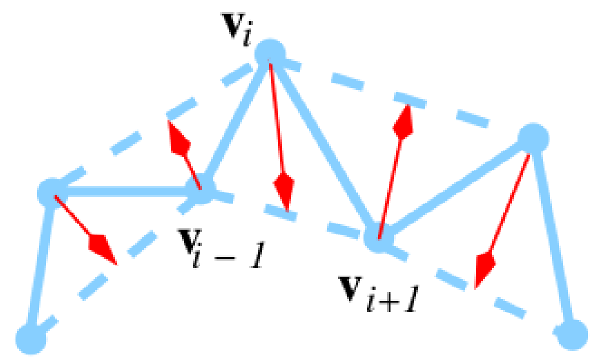
- Local averaging



$$\hat{\mathbf{v}}_i = \frac{1}{2}\left[\frac{1}{2}(\mathbf{v}_{i-1}+\mathbf{v}_i)\right] + \frac{1}{2}\left[\frac{1}{2}(\mathbf{v}_i+\mathbf{v}_{i+1})\right] = \boxed{\frac{1}{4}\mathbf{v}_{i-1} + \frac{1}{2}\mathbf{v}_i + \frac{1}{4}\mathbf{v}_{i+1}}$$

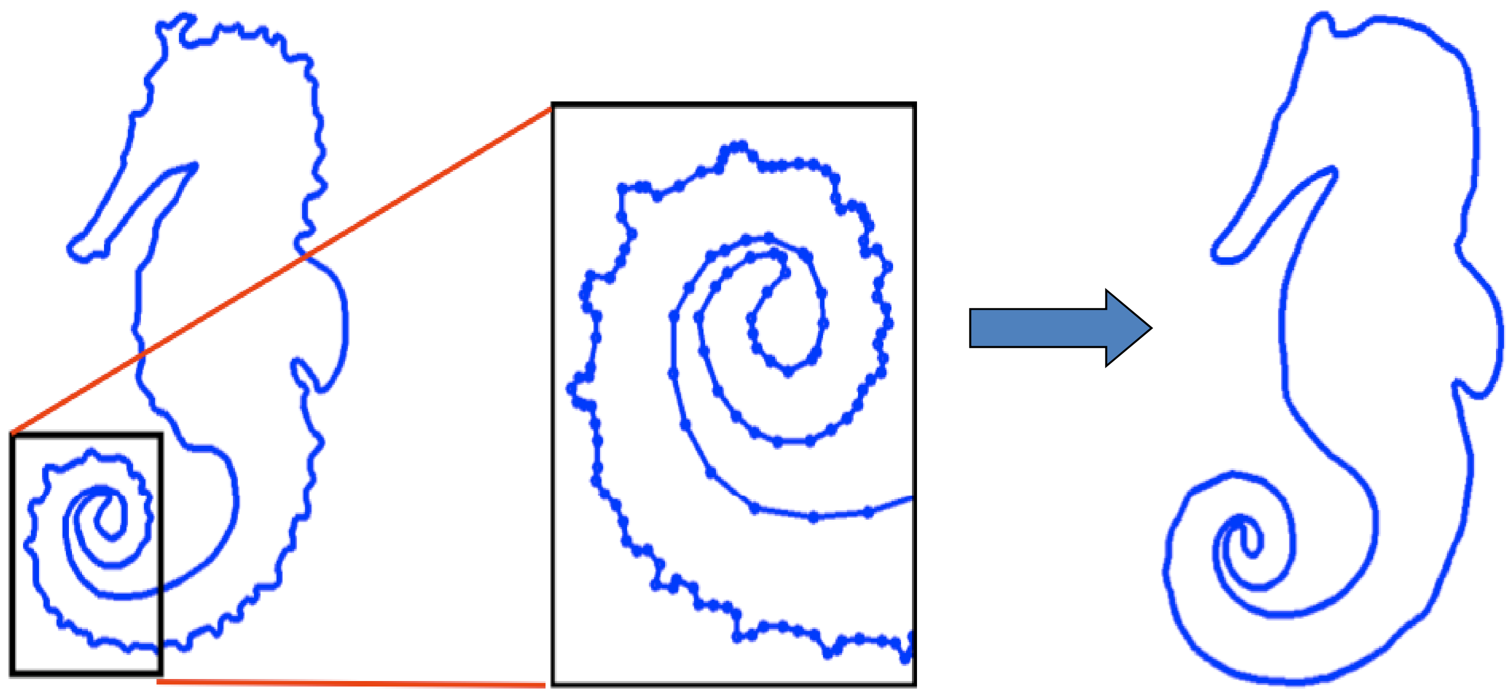
- **1D discrete Laplacian**

$$\delta(\mathbf{v}_i) = \frac{1}{2}(\mathbf{v}_{i-1}+\mathbf{v}_{i+1}) - \mathbf{v}_i$$



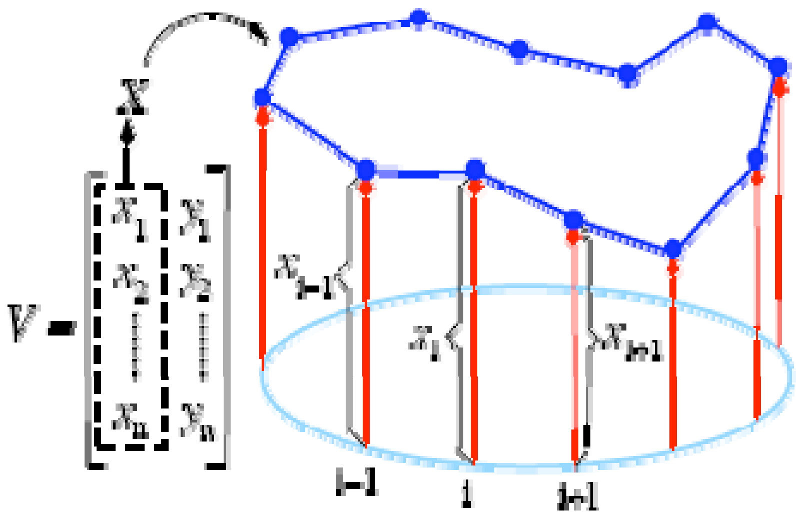
Smoothing result

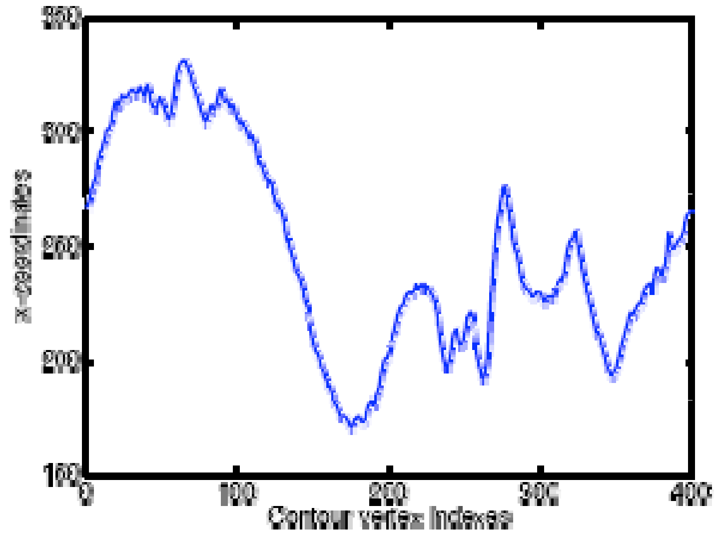
- Obtained by 10 steps of Laplacian smoothing



Signal representation

- Represent a contour using a discrete periodic 2D signal





x-coordinates of the seahorse contour

Laplacian smoothing in matrix form

$$\hat{X} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{n-1} \\ \hat{x}_n \end{bmatrix} = \boxed{\begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \dots & \dots & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = SX.$$

Smoothing operator

x component only
y treated same way

1D discrete Laplacian operator

$$\delta(X) = LX = \boxed{\begin{bmatrix} 1 & -\frac{1}{2} & 0 & \dots & \dots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \dots & \dots & 0 & -\frac{1}{2} & 1 \end{bmatrix}} X.$$

Smoothing and Laplacian operator $S = I - \frac{1}{2}L.$

Spectral analysis of signal/geometry

Express signal *X* as a linear sum of eigenvectors

$$X = \sum_{i=1}^{n} \mathbf{e}_i \tilde{x}_i = \begin{bmatrix} E_{11} \\ E_{21} \\ \vdots \\ E_{n1} \end{bmatrix} \tilde{x}_1 + \ldots + \begin{bmatrix} E_{1n} \\ E_{2n} \\ \vdots \\ E_{nn} \end{bmatrix} \tilde{x}_n = \begin{bmatrix} E_{11} & \ldots & E_{1n} \\ E_{21} & \ldots & E_{2n} \\ \vdots & \vdots & \vdots \\ E_{n1} & \ldots & E_{nn} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = E\tilde{X}.$$

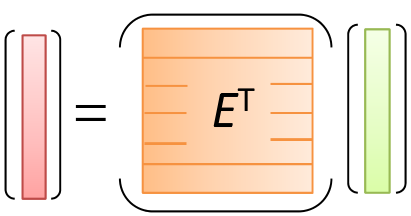
$$\boxed{\tilde{x}_i = \mathbf{e}_i^{\mathrm{T}} \cdot X}$$

Project *X* along eigenvector

DFT-like spectral transform

$X \longrightarrow \tilde{X}$

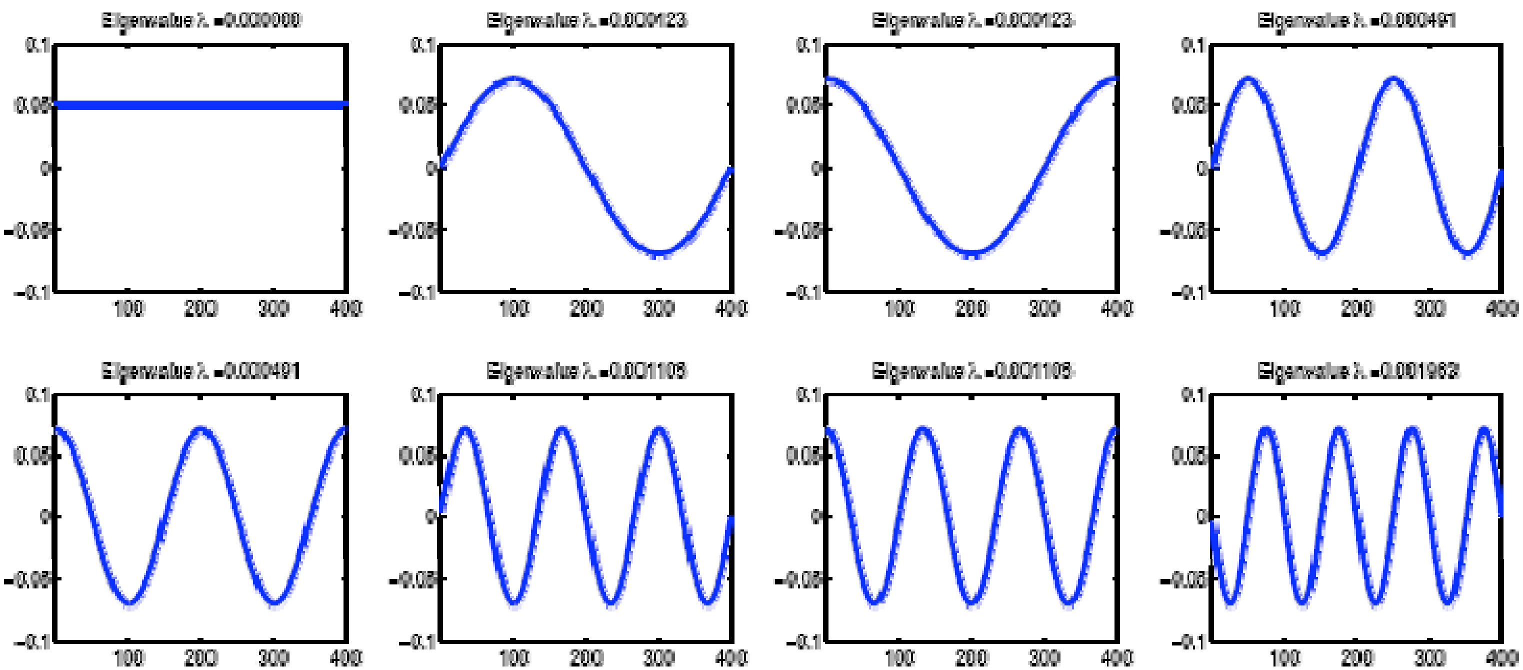
Spatial domain　　Spectral domain



$$\tilde{X} = E^{\mathrm{T}} X$$

Plot of eigenvectors

First 8 eigenvectors of the 1D periodic Laplacian



More oscillation as eigenvalues (frequencies) increase

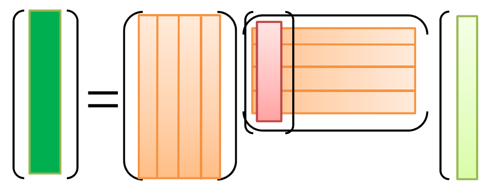
Relation to Discrete Fourier Transform

- Smallest eigenvalue of L is zero
- Each remaining eigenvalue (except for the last one when n is even) has multiplicity 2
- The plotted real eigenvectors are not unique to L
- **One particular set of eigenvectors of L are the DFT basis**
- Both sets exhibit similar oscillatory behaviours w.r.t. frequencies

Reconstruction and compression

- Reconstruction using *k* leading coefficients

$$X^{(k)} = \sum_{i=1}^{k} \mathbf{e}_i \tilde{x}_i, \qquad k \le n.$$

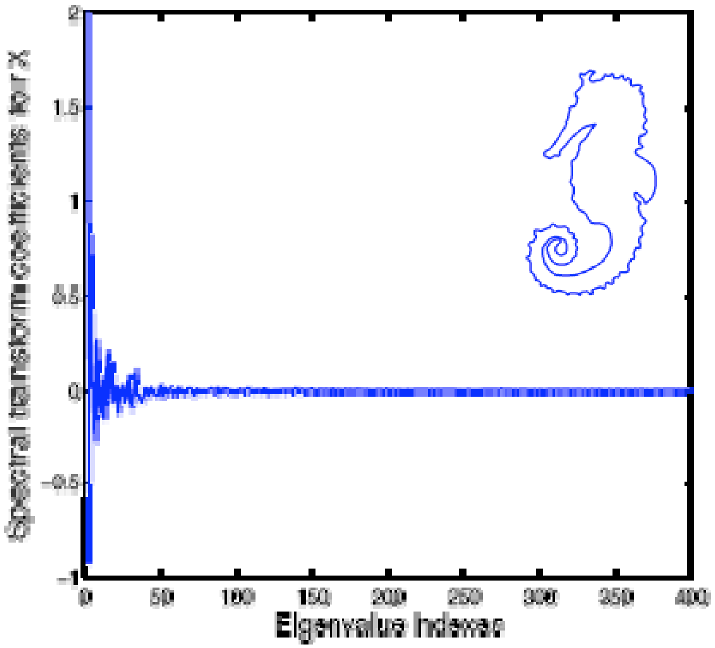


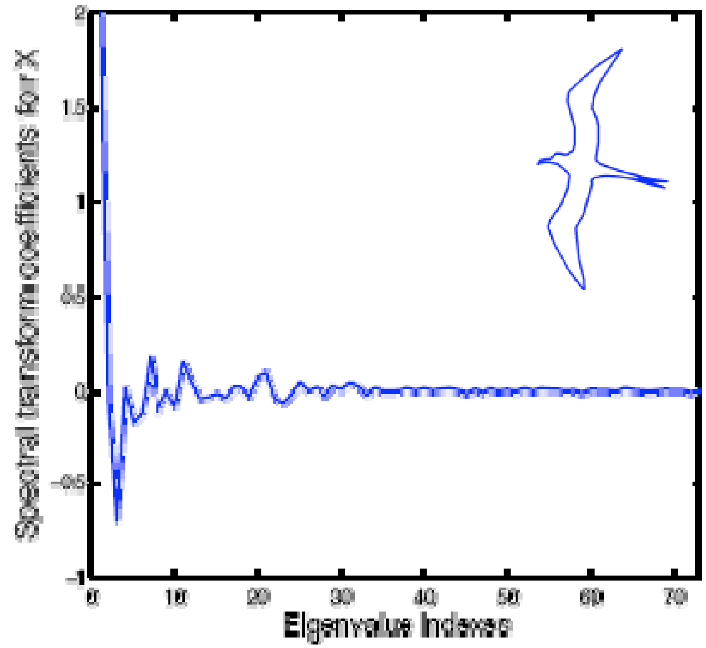
- A form of **spectral compression** with info loss given by L_2

$$||X - X^{(k)}|| = ||\sum_{i=k+1}^{n} \mathbf{e}_i \tilde{x}_i|| = \sqrt{\sum_{i=k+1}^{n} \tilde{x}_i^2}$$

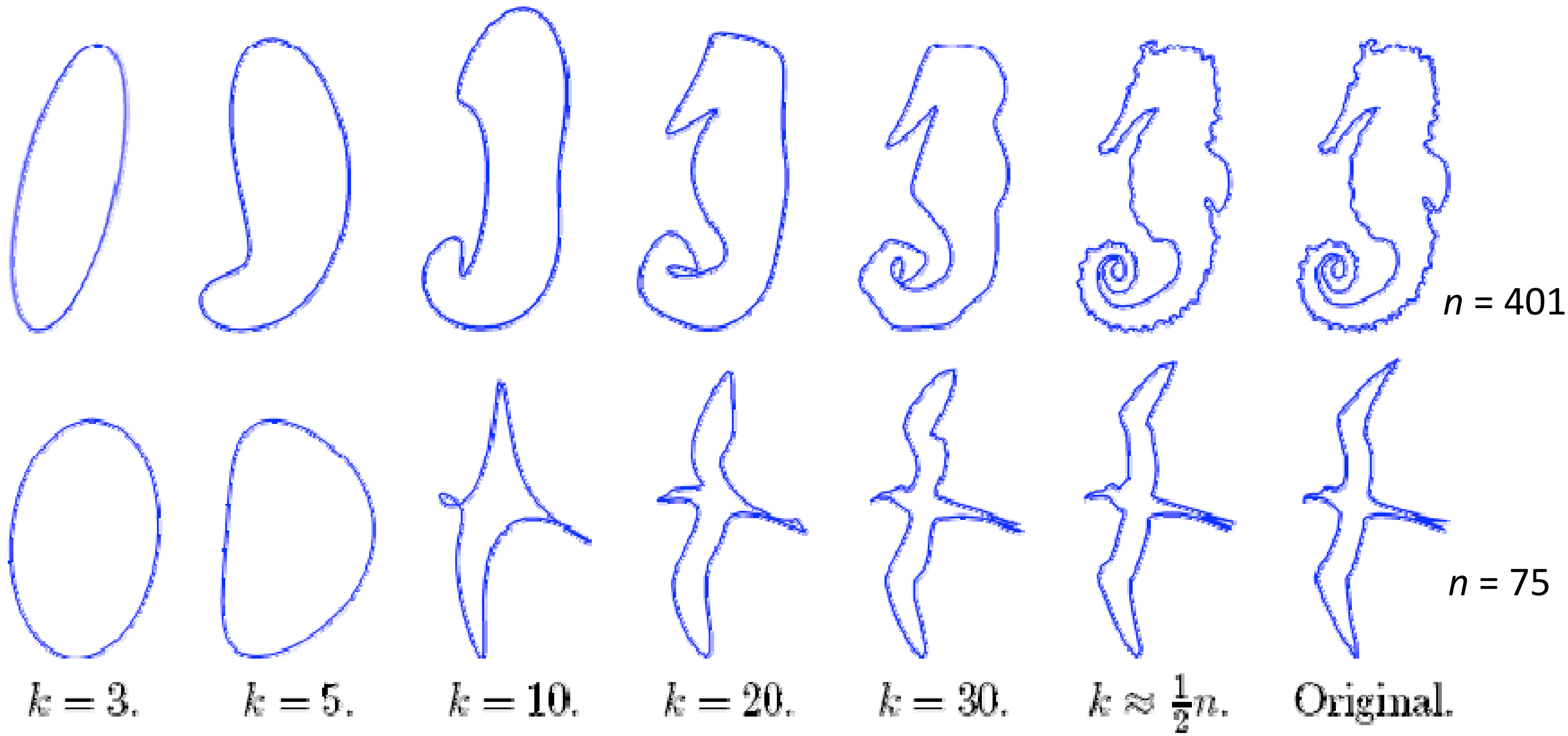
Plot of spectral transform coefficients

- Fairly fast decay as eigenvalue increases





Reconstruction examples



Laplacian smoothing as filtering

- Recall the Laplacian smoothing operator

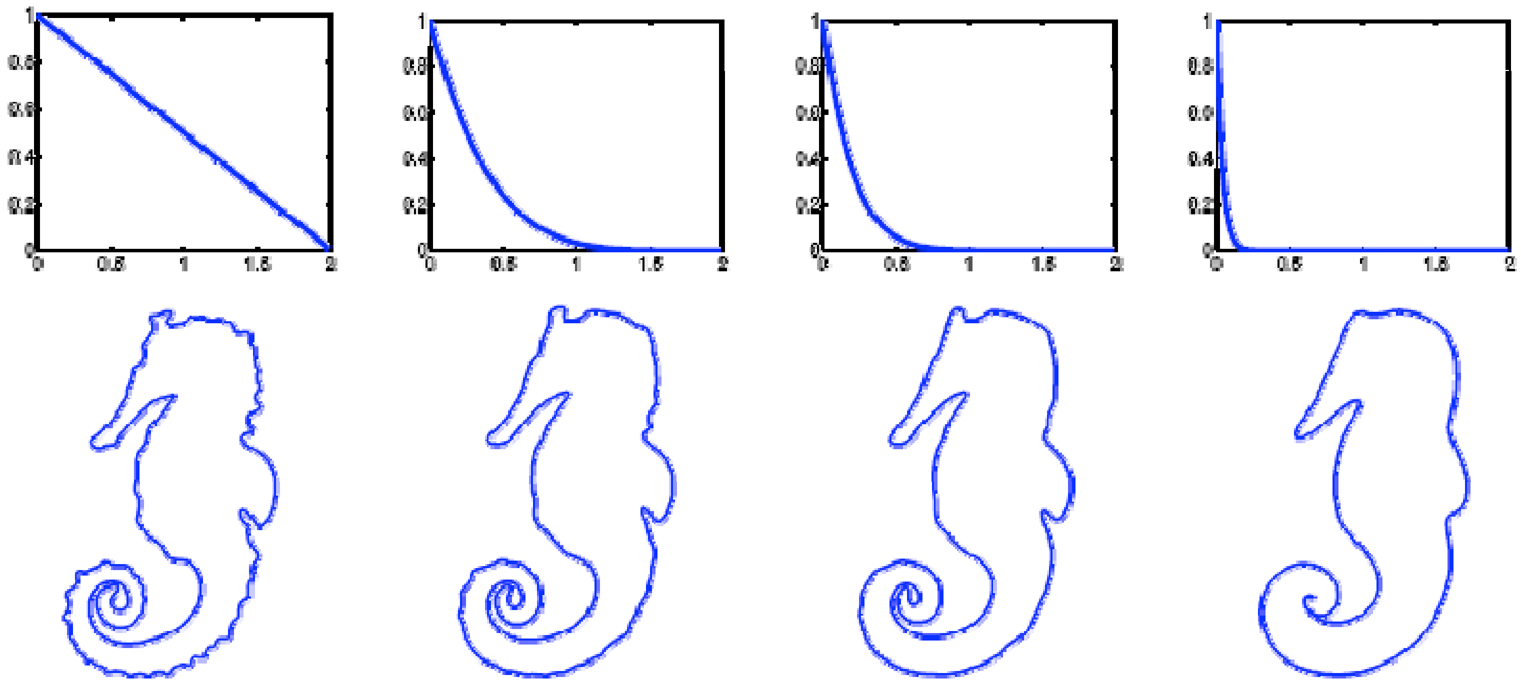
$$S = I - \frac{1}{2}L.$$

- Repeated application of *S*

$$X^{(m)} = S^m X = (I - \frac{1}{2}L)^m X = \sum_{i=1}^{n} (I - \frac{1}{2}L)^m \mathbf{e}_i \tilde{x}_i = \sum_{i=1}^{n} \mathbf{e}_i \boxed{(1 - \frac{1}{2}\lambda_i)^m} \tilde{x}_i.$$

A filter applied to
spectral coefficients

Examples



$m = 1$ $m = 5$ $m = 10$ $m = 50$

Filter: $f(\lambda) = (1 - \frac{1}{2}\lambda)^m$

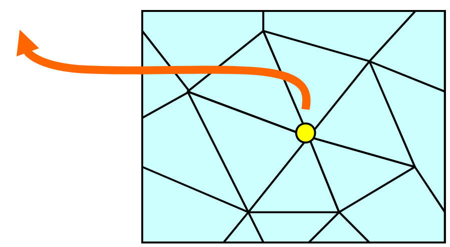
Computational issues

- No need to compute spectral coefficients for filtering
  - Polynomial (e.g., Laplacian): matrix-vector multiplication
- Spectral compression needs explicit spectral transform
- Efficient computation [Levy et al. 08]

Towards spectral **mesh** transform

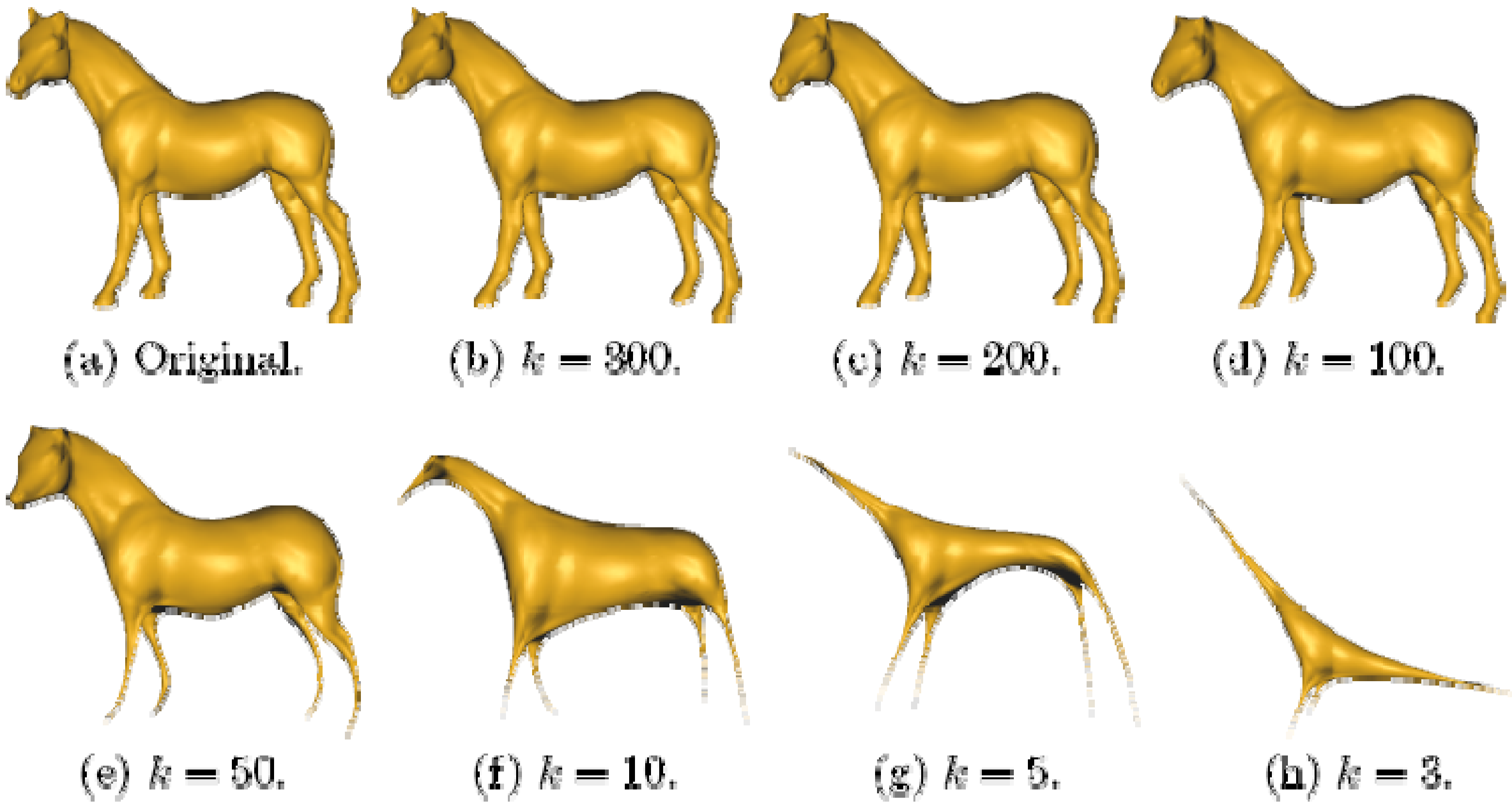
- Signal representation
  - Vectors of *x*, *y*, *z* vertex coordinates

(x, y, z)



- Laplacian operator for meshes
  - Encodes connectivity and geometry
  - Combinatorial: **graph Laplacians** and variants
  - Discretization of the continuous Laplace-Beltrami operator
- The same kind of spectral transform and analysis

Spectral Mesh Compression



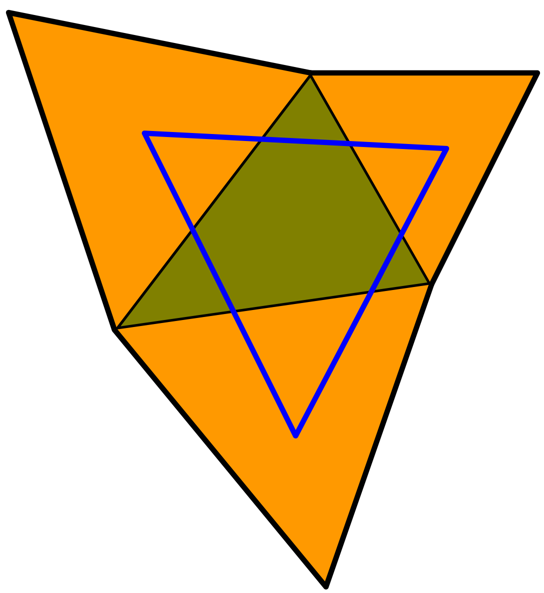
Spectral Processing - Perspectives

- Signal processing
  - Filtering and compression
  - Relation to discrete Fourier transform (DFT)
- Geometric
  - Global and intrinsic
- Machine learning
  - Dimensionality reduction

A geometric perspective: classical

Classical Euclidean geometry

- Primitives not represented in coordinates
- Geometric relationships deduced in a pure and self-contained manner
- Use of axioms

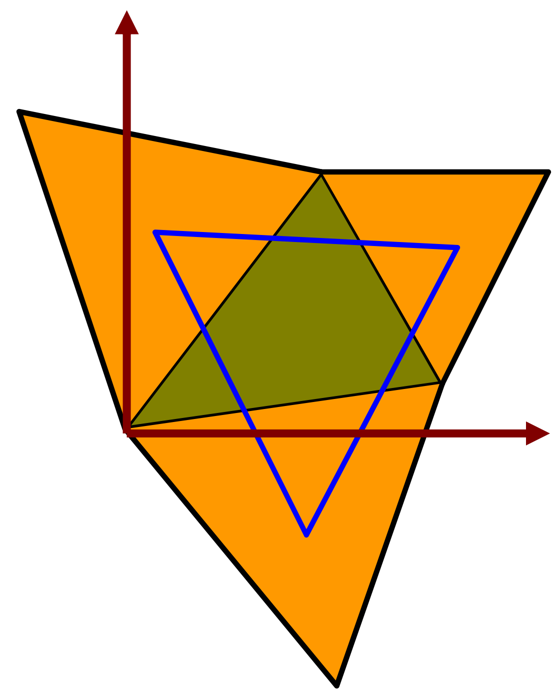


A geometric perspective: analytic

Descartes' analytic geometry

– Algebraic analysis tools introduced

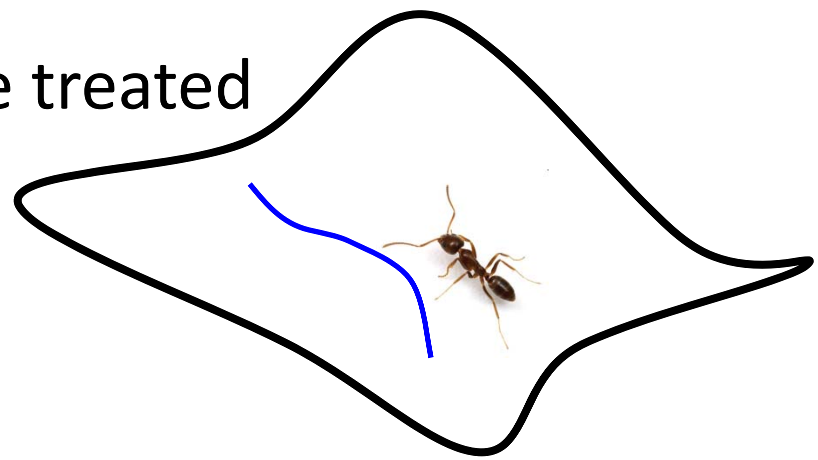
– Primitives referenced in global frame —— **extrinsic** approach



Intrinsic approach

Riemann’s intrinsic view of geometry

- Geometry viewed purely from the surface perspective
- Metric: “distance” between points on surface
- Many spaces (shapes) can be treated simultaneously: **isometry**



Spectral methods: intrinsic view

Spectral approach takes the intrinsic view

- **Intrinsic geometric/mesh information** captured via a linear mesh operator
- **Eigenstructures** of the operator present the intrinsic geometric information in an **organized manner**
- Rarely need all eigenstructures, **dominant** ones often suffice

Capture of global information

(Courant-Fisher) Let $S \in \Re^{n \times n}$ be a symmetric matrix. Then its eigenvalues $\lambda_1 \leq \lambda_2 \leq .... \leq \lambda_n$ must satisfy the following,

$$\lambda_i = \min_{\substack{\|\mathbf{v}\|_2 = 1 \\ \mathbf{v}^{\mathrm{T}}\mathbf{v}_k = 0,\ \forall 1 \leq k \leq i\text{-}1}} \mathbf{v}^{\mathrm{T}} S \mathbf{v}$$

where $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_{i-1}$ are eigenvectors of S corresponding to the smallest eigenvalues λ_1, λ_2 , …, λ_{i-1}, respectively.

Interpretation

$$\lambda_i = \min_{\substack{\|\mathbf{v}\|_2=1 \\ \mathbf{v}^{\mathrm{T}}\mathbf{v}_k=0,\ \forall 1\le k\le i\text{-}1}} \mathbf{v}^{\mathrm{T}} S \mathbf{v} \rightarrow \frac{\mathbf{v}^{\mathrm{T}} S \mathbf{v}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}$$

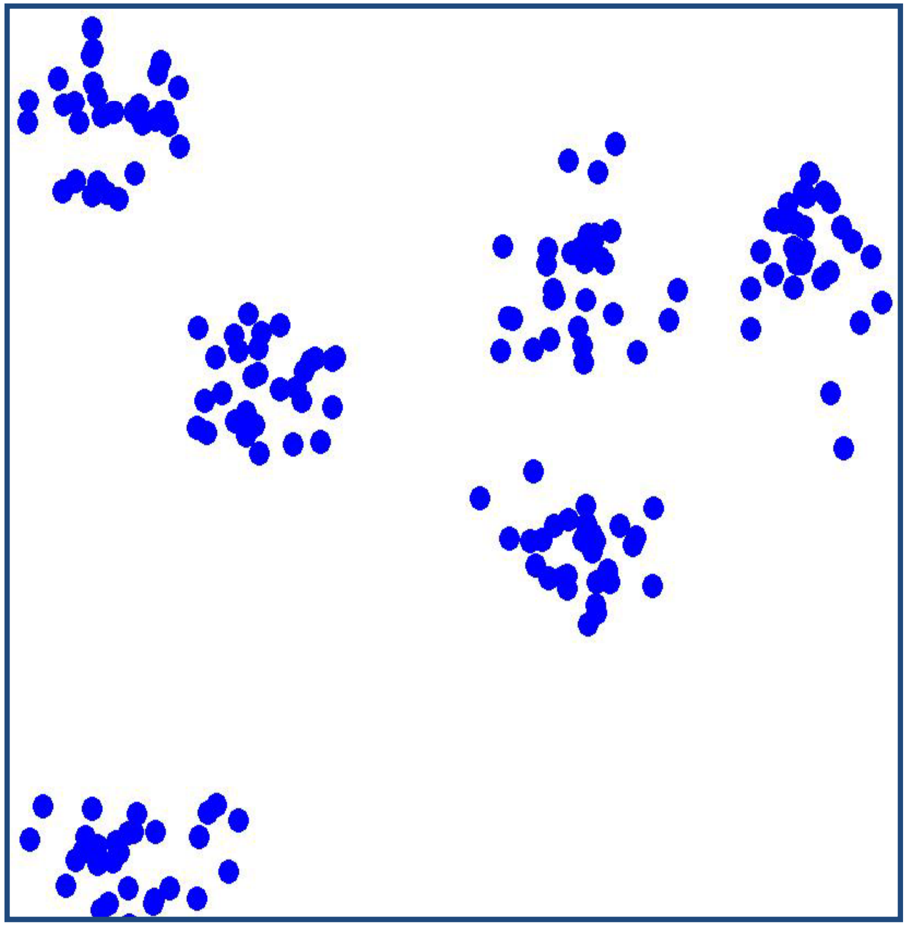
Rayleigh quotient

- Smallest eigenvector minimizes the Rayleigh quotient
- *k*-th smallest eigenvector minimizes Rayleigh quotient, among the vectors orthogonal to all previous eigenvectors
- Solutions to **global optimization** problems

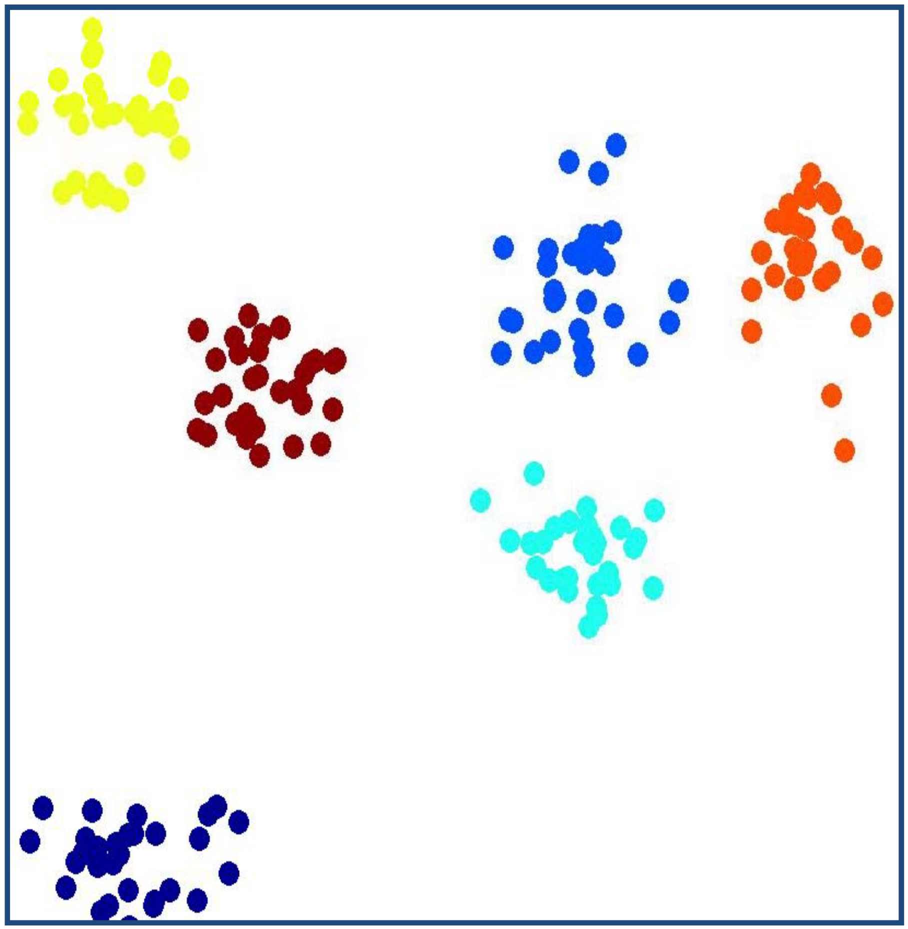
Use of eigenstructures

- Eigenvalues
  - Spectral graph theory: graph eigenvalues closely related to almost all major global graph invariants
  - Have been adopted as **compact global shape descriptors**
- Eigenvectors
  - Useful extremal properties, e.g., heuristic for NP-hard problems — normalized cuts and sequencing
  - Spectral embeddings capture global information, e.g., clustering

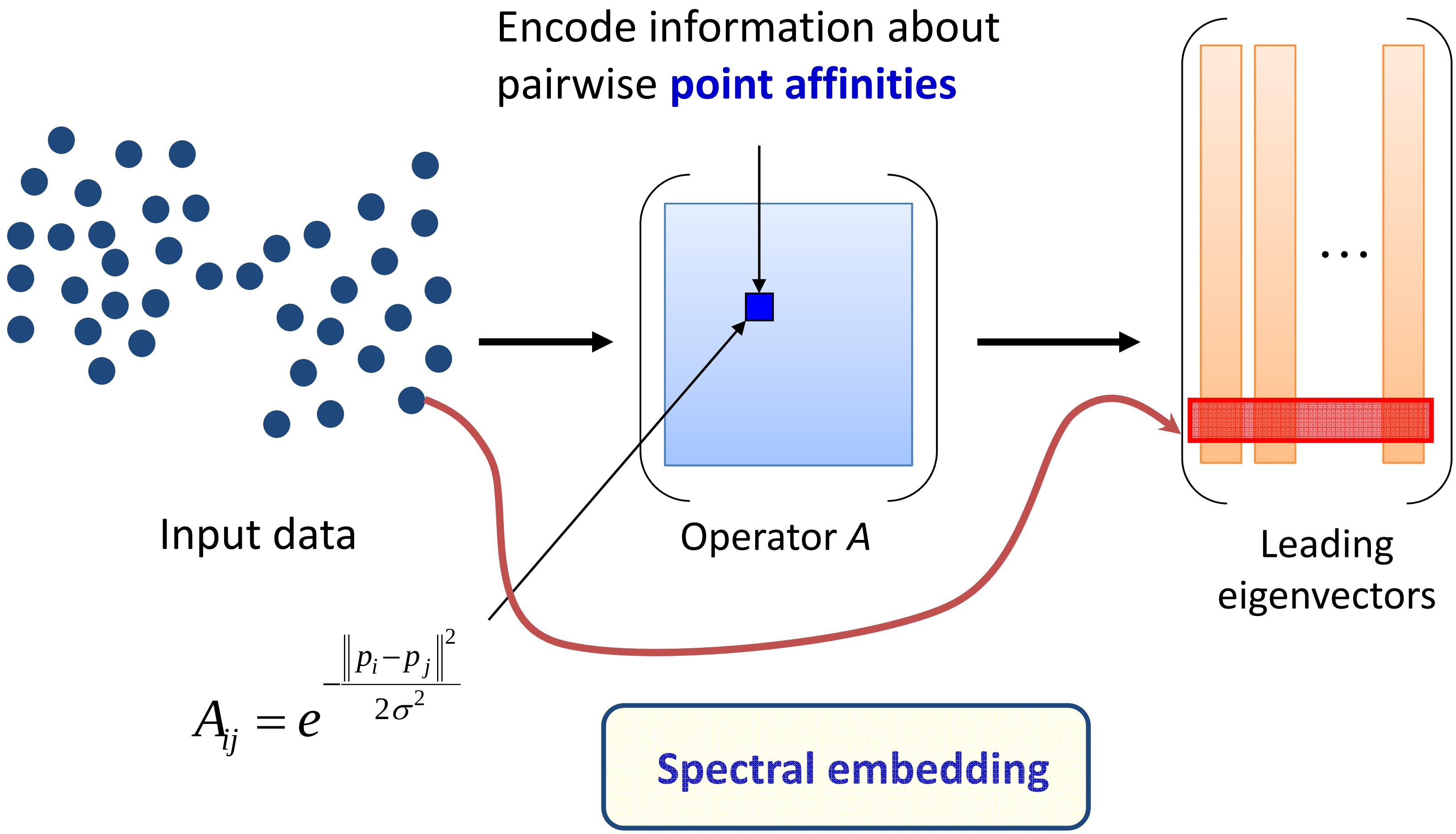
Example: clustering problem



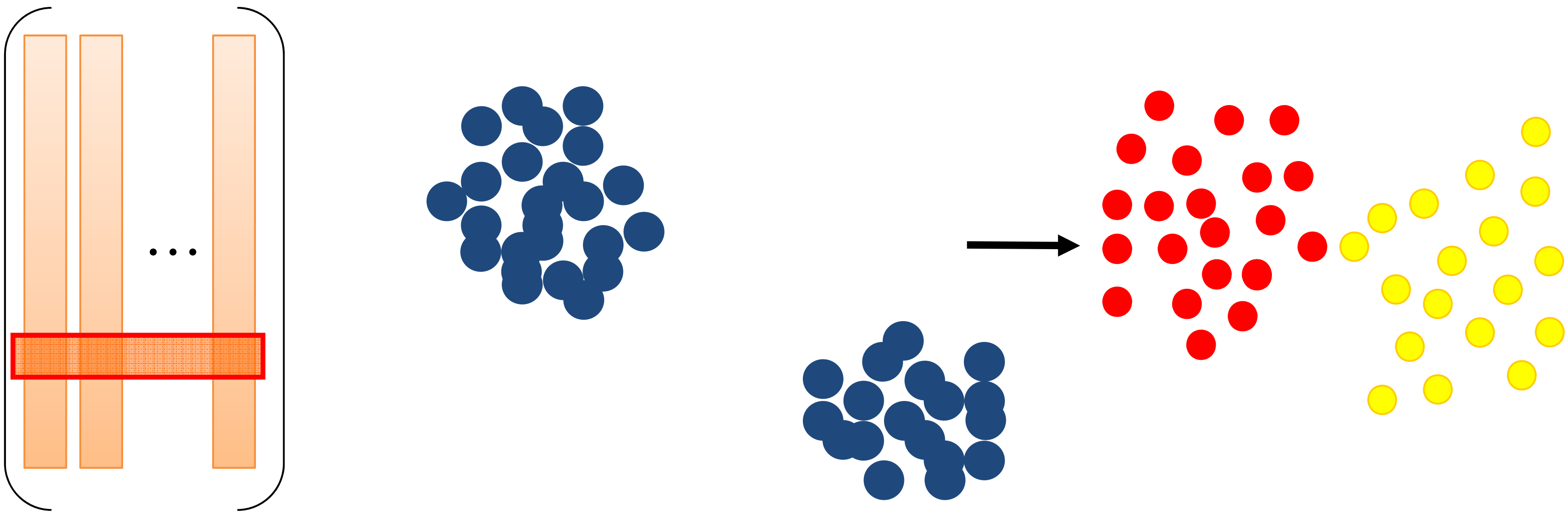
Example: clustering problem



Spectral clustering



Spectral clustering

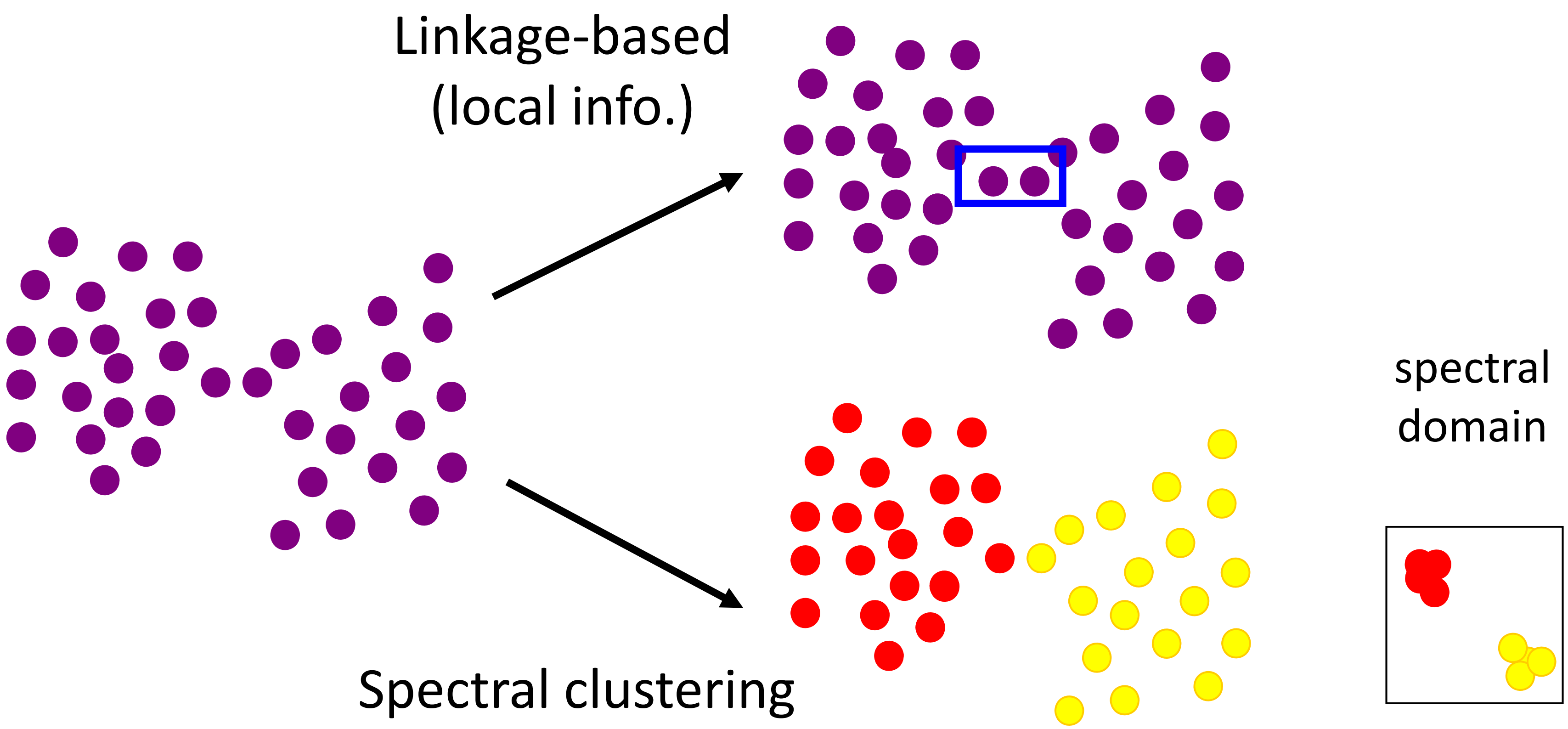


eigenvectors

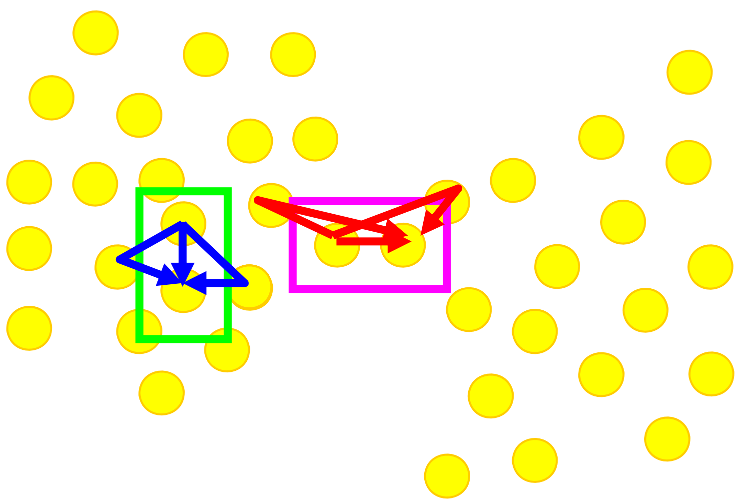
In spectral domain

Perform any clustering (e.g., *k*-means) in spectral domain

Why does it work this way?



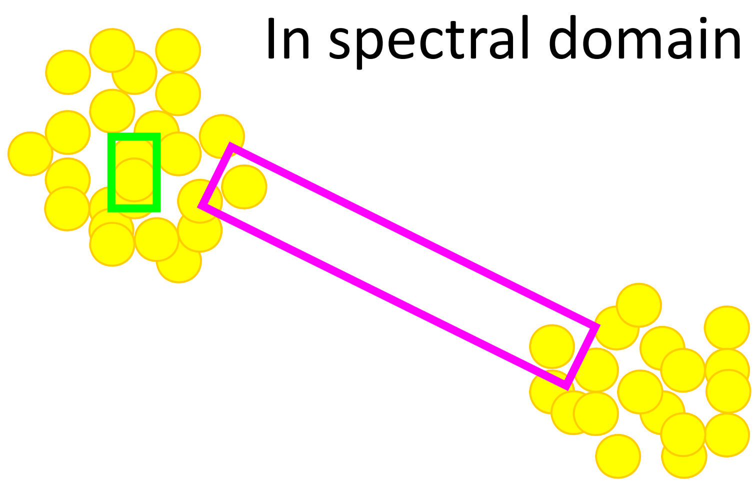
Local vs. global distances



Would be nice to cluster according to c_{ij}

- A **good distance**: Points in same cluster closer in transformed domain
- Look at **set of shortest paths — more global**
- **Commute time distance** c_{ij} = expected time for random walk to go from i to j and then back to i

Local vs. global distances



Commute time and spectral

- Eigen-decompose the graph Laplacian K

$$K = U \Lambda U^{\mathsf{T}}$$

- Let K' be the **generalized inverse** of K,

$$K' = U \Lambda' U^{\mathsf{T}},$$

$\Lambda'_{ii} = 1/\Lambda_{ii}$ if $\Lambda_{ii} \neq 0$, otherwise $\Lambda'_{ii} = 0$.

- Note: the Laplacian is singular

Commute time and spectral

- Let z_i be the i-th row of $U\Lambda'^{1/2}$ — the spectral embedding
  - Scaling each eigenvector by **inverse square root of eigenvalue**
- Then

$$||z_i - z_j||^2 = c_{ij}$$

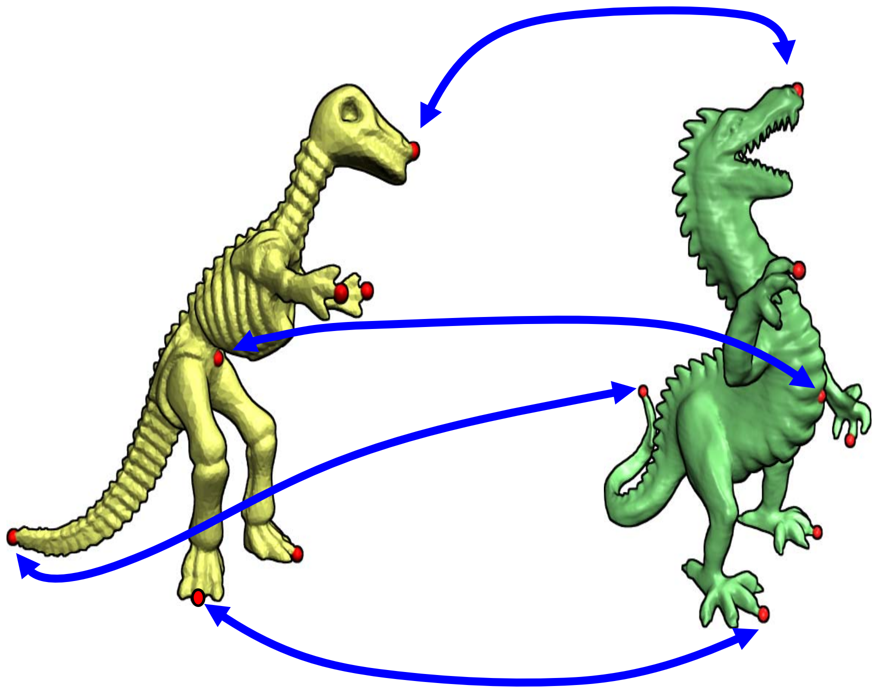
the commute time distance

[Klein & Randic 93, Fouss et al. 06]

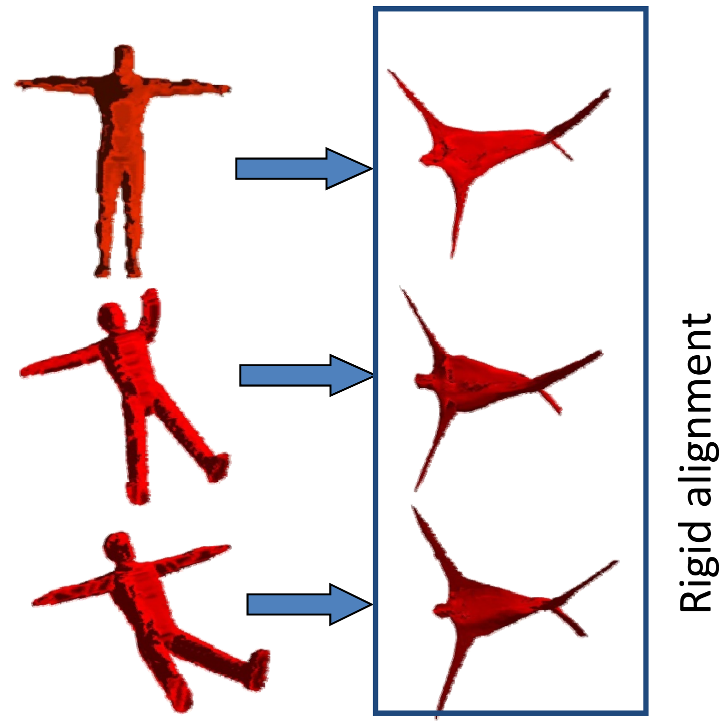
- Full set of eigenvectors used, but select first k in practice

Example: intrinsic geometry

Our first example: correspondence



Spectral transform to handle shape pose

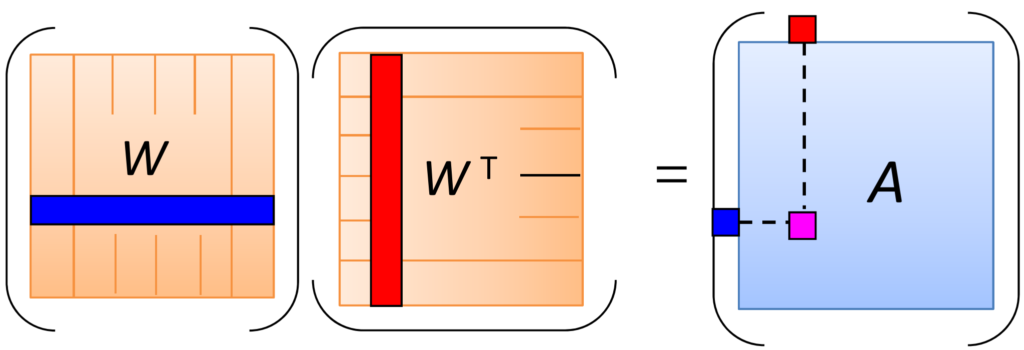


Spectral Processing - Perspectives

- Signal processing
  - Filtering and compression
  - Relation to discrete Fourier transform (DFT)
- Geometric
  - Global and intrinsic
- Machine learning
  - Dimensionality reduction

Spectral embedding

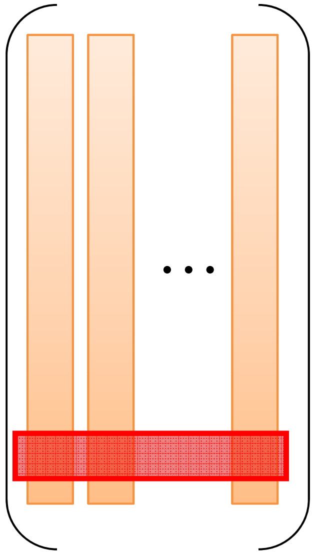
- Spectral decomposition $\boxed{A = U\Lambda U^{\mathrm{T}}}$
- Full spectral embedding given by scaled eigenvectors (each scaled by squared root of eigenvalue) completely captures the operator



$$W = U\Lambda^{1/2}$$

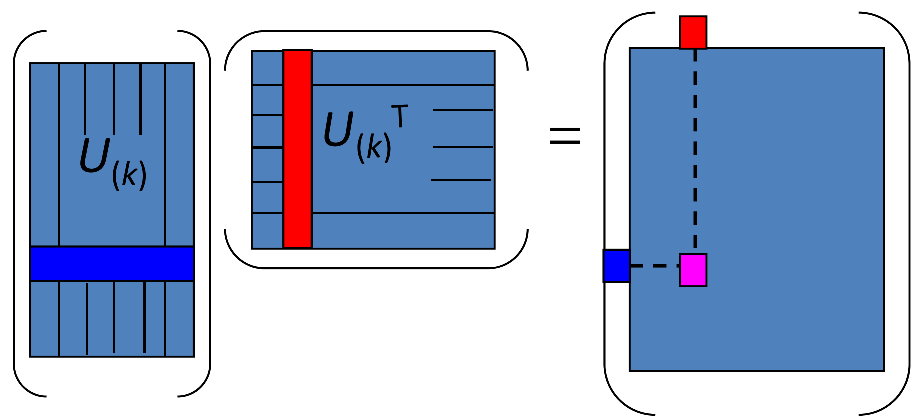
Dimensionality reduction

- Full spectral embedding is high-dimensional
- Use few dominant eigenvectors —— dimensionality reduction
  - **Information-preserving**
  - **Structure enhancement** (Polarization Theorem)
  - Low-D representation: **simplifying solutions**



Eckard & Young: Info-preserving

- $A \in \Re^{n \times n}$: symmetric and positive semi-definite
- $U_{(k)} \in \Re^{n \times k}$: leading eigenvectors of A, scaled by square root of eigenvalues
- Then $U_{(k)}U_{(k)}^{\mathsf{T}}$: **best rank-*k* approximation** of A in Frobenius norm



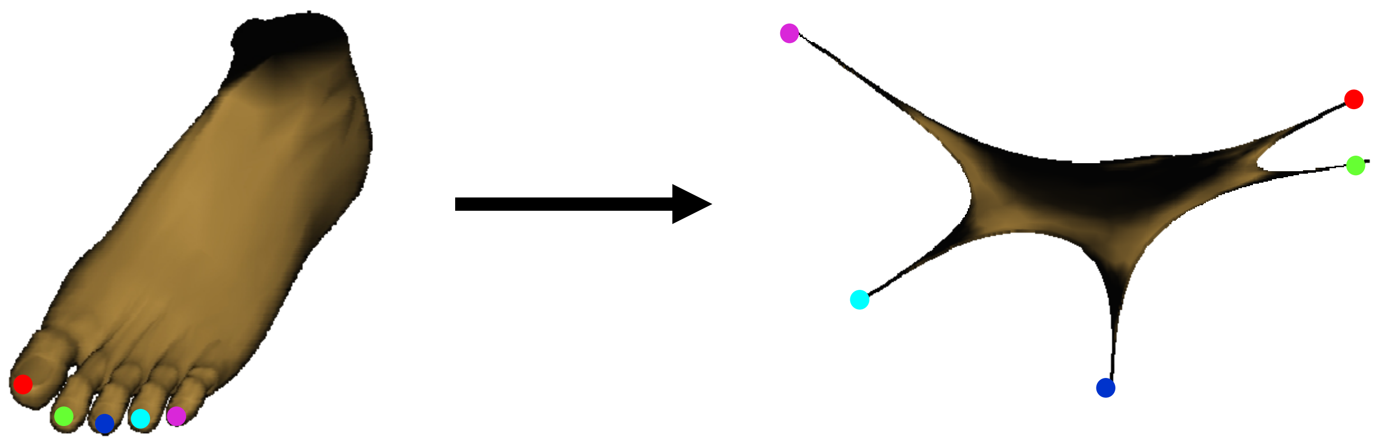
Brand & Huang: Polarization Theorem

Theorem 5.6 (Polarization Theorem) Denote by $S_{(k)} = X_{(k)}X_{(k)}^{\top}$ the best rank-k approximation of S with respect to the Frobenius norm, where $X_{(k)}$ is as defined in Theorem 5.5. As S is projected to successively lower ranks $S_{(n-1)}, S_{(n-2)}, \ldots, S_{(2)}, S_{(1)}$, the sum of squared angle-cosines,

$$s_k = \sum_{i \neq j} (\cos \theta_{ij}^{(k)})^2 = \sum_{i \neq j} \left(\frac{\mathbf{x}_i^{(k)\top} \mathbf{x}_j^{(k)}}{\|\mathbf{x}_i^{(k)}\|_2 \cdot \|\mathbf{x}_j^{(k)}\|_2} \right)^2$$

is strictly increasing, where $\mathbf{x}_i^{(k)}$ is the i-th row of $X_{(k)}$.

Low-dim $\rightarrow$ simpler problems

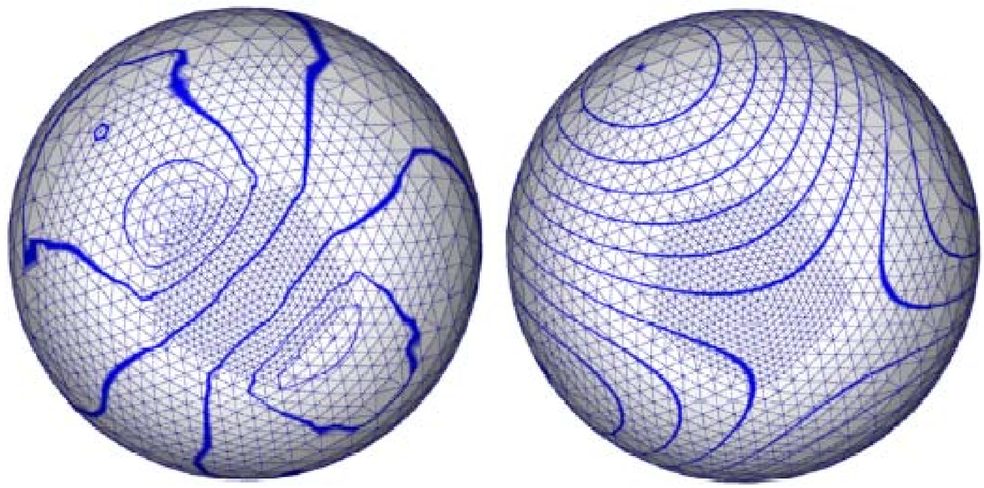


- Mesh projected into the eigenspace formed by the first two eigenvectors of a mesh Laplacian
- Reduce 3D analysis to contour analysis [Liu & Zhang 07]

Challenges - Not quite DFT

- Basis for DFT is fixed given *n*, e.g., regular and easy to compare (Fourier descriptors)

- Spectral mesh transform is operator-dependent

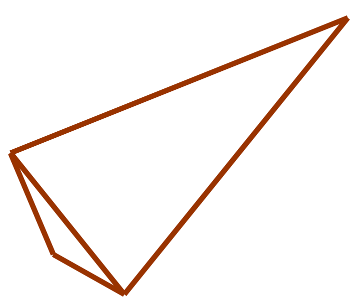


Which operator to use?

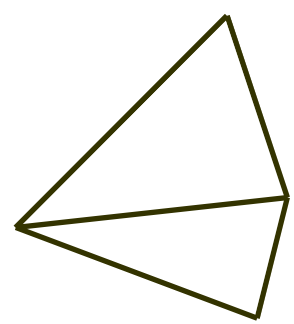
Different behavior of eigen-functions on the same sphere

Challenges - No free lunch

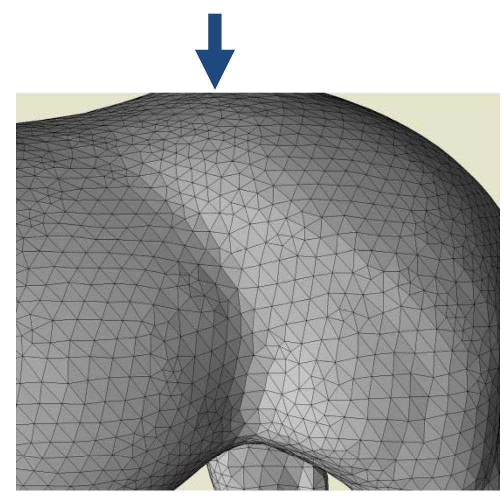
- No mesh Laplacian on general meshes can satisfy a list of all desirable properties
- Remedy: use nice meshes — **Delaunay** or **non-obtuse**



Delaunay but obtuse



Non-obtuse



Additional issues

- Computational issues: FFT vs. eigen-decomposition
- Regularity of vibration patterns lost
  - Difficult to characterize eigenvectors, eigenvalue not enough
  - Non-trivial to compare two sets of eigenvectors — how to pair up?

Conclusion

Use eigen-structure of “well-behaved” linear operators for geometry processing

Solve problem in a different domain via a spectral transform

Fourier analysis on meshes

Captures global and intrinsic shape characteristics

Dimensionality reduction: effective and simplifying