Spectral Algorithms I

Slides based on “Spectral Mesh Processing” Siggraph 2010 course
Why Spectral?

A different way to look at functions on a domain

Hi, Dr. Elizabeth?

Yeah, uh... I accidentally took the Fourier transform of my cat...

Meow!
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: $Au = \lambda u$, $u \neq 0$

- Diagonalization or eigen-decomposition: $A = U\Lambda U^T$

- Projection into eigen-subspace: $y' = U_{(k)}U_{(k)}^T y$

- DFT-like spectral transform: $\hat{y} = U^T y$
Eigen-structure

Eigen-decomposition

\[ A = \begin{pmatrix} U & \Lambda & U^T \end{pmatrix} \]

Subspace projection

\[ y' = U_{(3)} U_{(3)}^T y \]

Mesh geometry

\[ y = \begin{bmatrix} \text{positive definite matrix } A \end{bmatrix} \]
Eigen-structure

\[ \text{mesh geometry} \]

\[ \mathbf{y} = \begin{pmatrix} \end{pmatrix} \]

\[ \text{positive definite matrix } \mathbf{A} \]

\[ \mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \]

\[ \text{eigen-decomposition} \]

\[ \hat{\mathbf{y}} = \mathbf{U}^T \mathbf{y} \]

\[ \text{spectral transform} \]
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
  – 1st-order vs. higher order
  – Normalized vs. un-normalized
Operators?

• **Best**
  – Symmetric positive definite operator: \( x^T A x > 0 \) for any \( x \)

• **Can live with**
  – Semi-positive definite (\( x^T A x \geq 0 \) for any \( x \))
  – Non symmetric, as long as eigenvalues are real and positive
    e.g.: \( L = D W \), 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

\[ \hat{v}_i = \frac{1}{2} \left[ \frac{1}{2} (v_{i-1} + v_i) \right] + \frac{1}{2} \left[ \frac{1}{2} (v_i + v_{i+1}) \right] = \frac{1}{4} v_{i-1} + \frac{1}{2} v_i + \frac{1}{4} v_{i+1} \]

- **1D discrete Laplacian**

\[ \delta(v_i) = \frac{1}{2} (v_{i-1} + v_{i+1}) - v_i \]
Smoothing result

• Obtained by 10 steps of Laplacian smoothing
Signal representation

• Represent a contour using a discrete periodic 2D signal

\[ V = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \]

x-coordinates of the seahorse contour
Laplacian smoothing in matrix form

\[ \hat{X} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{n-1} \\ \hat{x}_n \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 & \cdots & \cdots & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \cdots & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \vdots & \cdots & \cdots & \vdots & \ddots & \vdots \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = S \hat{X}. \]

- Smoothing operator
- \( x \) component only
- \( y \) treated same way
1D discrete Laplacian operator

\[ \delta(X) = LX = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \cdots & \cdots & 0 & -\frac{1}{2} & 1 \end{bmatrix} X. \]

Smoothing and Laplacian operator

\[ S = I - \frac{1}{2}L. \]
Spectral analysis of signal/geometry

Express signal $X$ as a linear sum of eigenvectors

$$X = \sum_{i=1}^{n} e_i \tilde{x}_i = \begin{bmatrix} E_{11} & \cdots & E_{1n} \\ E_{21} & \cdots & E_{2n} \\ \vdots & \ddots & \vdots \\ E_{n1} & \cdots & E_{nn} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = E \tilde{X}.$$ 

Project $X$ along eigenvector

$$\tilde{x}_i = e_i^T \cdot X$$

DFT-like spectral transform

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

Spatial domain $X$ \hspace{2cm} Spectral domain $\tilde{X}$
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

$$X^{(k)} = \sum_{i=1}^{k} e_i \tilde{x}_i, \quad k \leq n.$$ 

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

$$||X - X^{(k)}|| = \left|\sum_{i=k+1}^{n} e_i \tilde{x}_i \right| = \sqrt{\sum_{i=k+1}^{n} \tilde{x}_i^2}.$$
Plot of spectral transform coefficients

- Fairly fast decay as eigenvalue increases
Reconstruction examples

$n = 401$

$k = 3.$  $k = 5.$  $k = 10.$  $k = 20.$  $k = 30.$  $k \approx \frac{1}{2}n.$  Original.

$n = 75$
Laplacian smoothing as filtering

• Recall the Laplacian smoothing operator

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

• Repeated application of \( S \)

\[ X^{(m)} = S^m X = (I - \frac{1}{2}L)^m X = \sum_{i=1}^{n} (I - \frac{1}{2}L)^m e_i \tilde{x}_i = \sum_{i=1}^{n} e_i (1 - \frac{1}{2}\lambda_i)^m \tilde{x}_i. \]

A filter applied to spectral coefficients
Examples

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

- 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

(a) Original.
(b) $k = 300$.
(c) $k = 200$.
(d) $k = 100$.
(e) $k = 50$.
(f) $k = 10$.
(g) $k = 5$.
(h) $k = 3$. 
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 $S \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Then its eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ must satisfy the following,

$$\lambda_i = \min_{\|v\|_2 = 1} v^T S v$$

$$v^T v_k = 0, \ \forall 1 \leq k \leq i-1$$

where $v_1, v_2, \ldots, v_{i-1}$ are eigenvectors of $S$ corresponding to the smallest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{i-1}$, respectively.
Interpretation

\[ \lambda_i = \min_{\|v\|_2 = 1} \frac{v^T S v}{v^T v} \]

- 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

Encode information about pairwise point affinities

Spectral embedding

Input data

$A_{ij} = e^{-\frac{(p_i - p_j)^2}{2\sigma^2}}$

Operator $A$

Leading eigenvectors
Spectral clustering

in spectral domain (e.g., $k$-means) in spectral domain
Why does it work this way?

Linkage-based (local info.)

Spectral clustering
Local vs. global distances

• 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 $i$

Would be nice to cluster according to $c_{ij}$
Local vs. global distances

In spectral domain
Commute time and spectral

- Eigen-decompose the graph Laplacian $K$

  \[ K = U \Lambda U^T \]

- Let $K'$ be the \textit{generalized inverse} of $K$,

  \[ K' = U \Lambda' U^T, \]

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

- 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 \textit{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

Rigid alignment
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 \( A = U\Lambda U^T \)
- 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

- $A \in \mathbb{R}^{n \times n}$: symmetric and positive semi-definite
- $U_{(k)} \in \mathbb{R}^{n \times k}$: leading eigenvectors of $A$, scaled by square root of eigenvalues
- Then $U_{(k)} U_{(k)}^\top$: best rank-$k$ approximation of $A$ in Frobenius norm
Theorem 5.6 (Polarization Theorem) Denote by $S(k) = X(k)X(k)^T$ 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,

$$s_k = \sum_{i \neq j} (\cos \theta_{ij}^{(k)})^2 = \sum_{i \neq j} \left( \frac{x_i^{(k)^T}x_j^{(k)}}{||x_i^{(k)}||_2 \cdot ||x_j^{(k)}||_2} \right)^2$$

is strictly increasing, where $x_i^{(k)}$ 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 eigenfunctions 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
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.