Spectral Transform

Group 4

March 19, 2019

Overview

- Intro and Basic of Spectral Transform
-
-
-

Intro

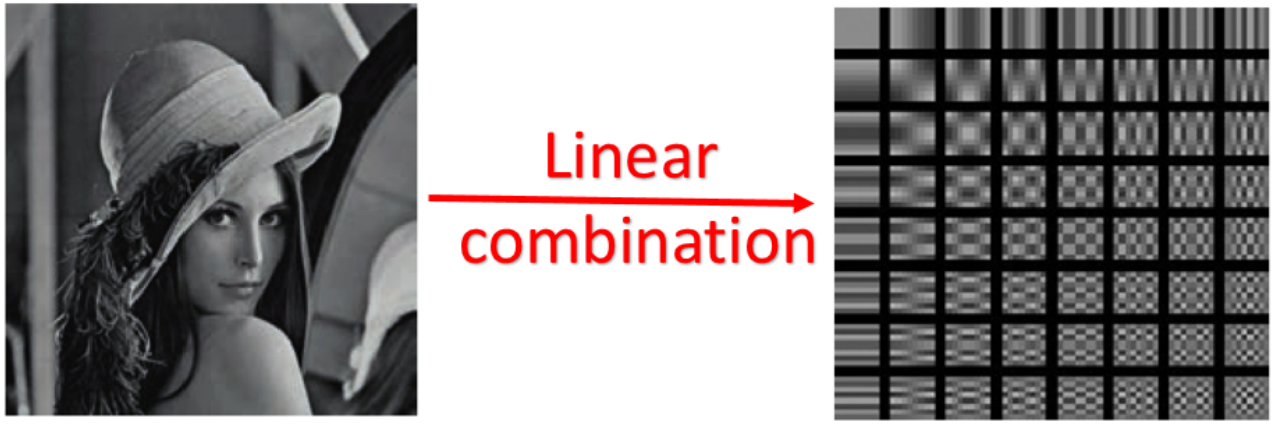
How to understand ”Spectral Transform”?

- Consider Fourier series
- $f(x) = \frac{a_0}{2} + \sum_{-\infty}^{+\infty} a_k cos(kx)$
- Infinite dimensional linear space , whose base ”vectors” differ in frequency:$[cos(x), cos(2x), \ldots]$
- And the Coordinates under this space determine the specific f(x):$[a_0, a_1, \ldots]$

Intro

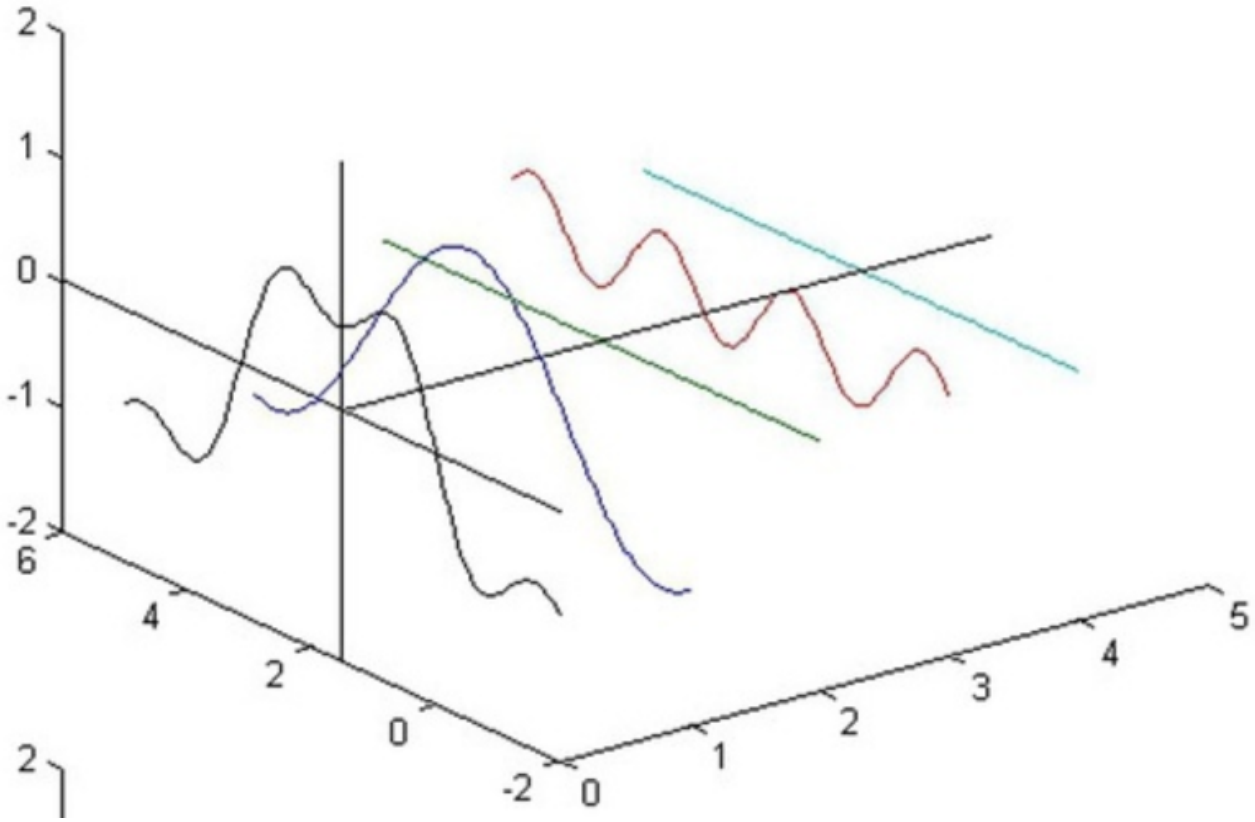
From Real Space to Discrete Space

- Consider matrix in discrete space
- We can find a "matrix base" of $n * n$ matrix space:$[M_1, M_2 \ldots, M_{n^2}]$
- $M_{image} = \sum_{k=1}^{n^2} a_k M_k$



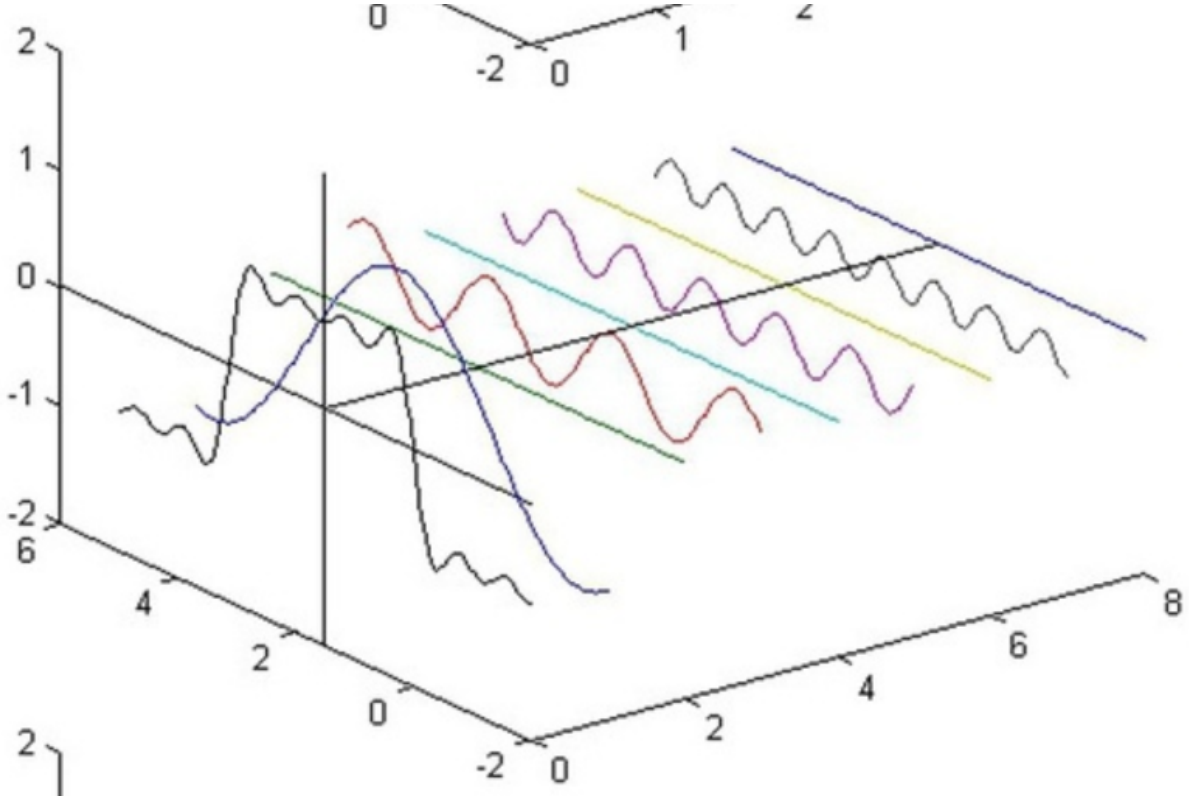
Intro

To understand the ”High/Low frequency information”



Intro

To understand the ”High/Low frequency information”



Intro

How can we utilize ”Spectral Transform”?

- Eg: image compression
- $M_{compression} = \sum_{k=1}^{t(t<n^2)} a_k M_k$
- Drop out some certain a_k helps to compress image while keep most of ”feature”

Original 80% 90% 95% 98% 99%

Intro

Think about that

- Notice that $M_k$just depends on n ,it's no need to save the M_k
- Before compression: $M_{image} = \sum_{k=1}^{n^2} a_k M_k$ so we need to keep n^2 numbers ($a_k, k \in [1, n^2]$)
- After compression: $M_{compression} = \sum_{k=1}^{t(t<n^2)} a_k M_k$ we only need to keep $t$numbers ($a_k, k \in [1, t]$)

Intro

Difference between ”Spectral Transform” and SVD?

- SVD: $M = \sum_{i=1}^{n} \sigma_i \vec{\mu^T}_i \vec{\lambda}_i$ define $\vec{\mu^T}_i \vec{\lambda}_i$ as S_i(a single-rank matrix)
- ”Spectral Transform” : $M = \sum_{i=1}^{n^2} \sigma_i M_i$
- S_i depends on the specific matrix ,while M_i is not.

Intro

Two practical problems we focus on

- Removing the rough features or noise(fig2)
- Compression under certain data loss(fig1)

fig1

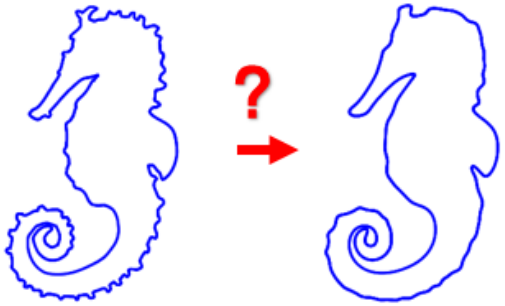


fig2

Prob1—-Removing the rough features

Eg:seahorse

- The contour is represented as a sequence of 2D points

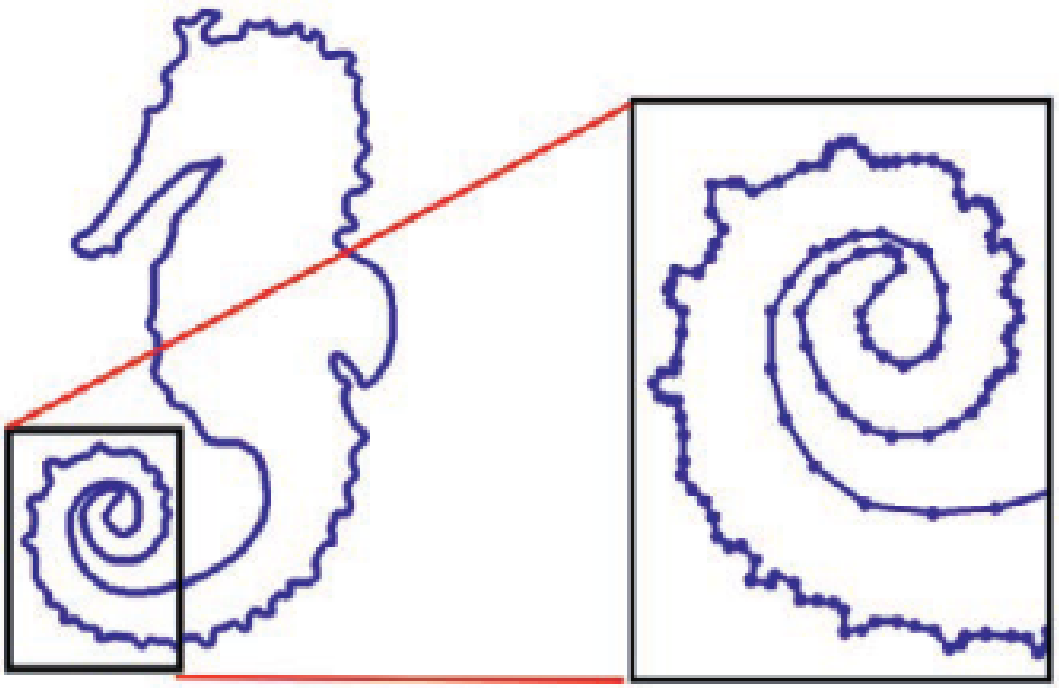


Figure: A seahorse shape with rough features

Prob1—-Removing the rough features

Approach:Midpoint-smoothing

- Midpoint-smoothing:$v_i = \frac{1}{2}(v_i + v_{i-1})$

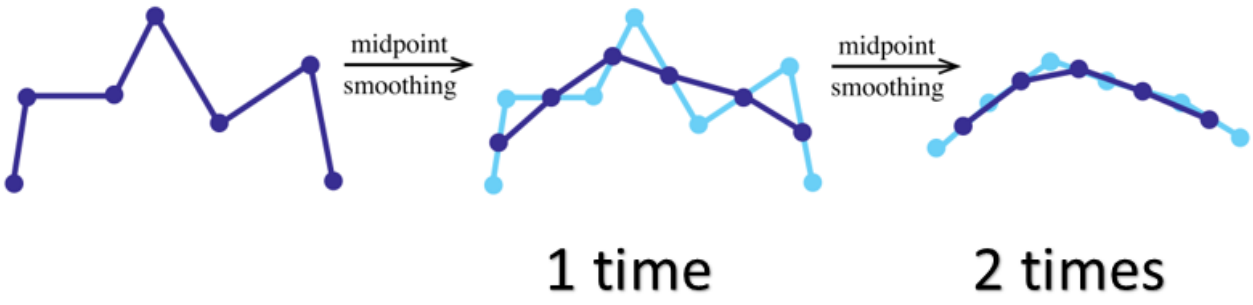
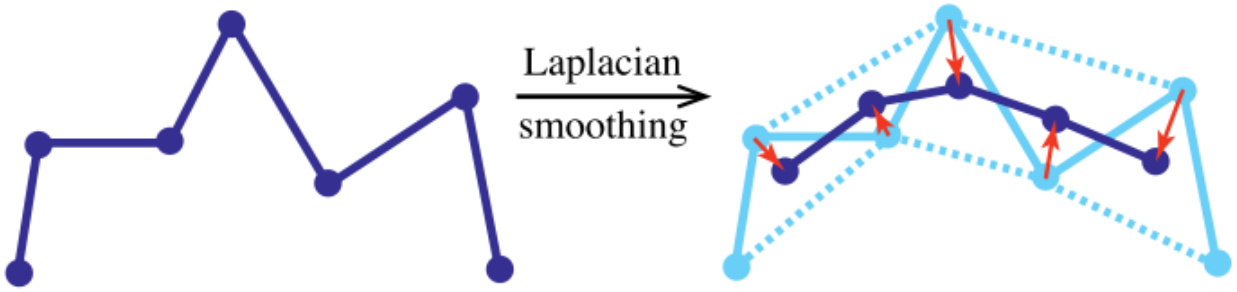


Figure: Results after 2 times of midpoint-smoothing

Prob1—-Removing the rough features

Definition:Laplacian smoothing

- After 2 times Midpoint-smoothing , define *local averaging* as $v_i = \frac{1}{2}(\frac{1}{2}(v_i + v_{i-1}) + \frac{1}{2}(v_i + v_{i+1}))$ which is referred to as *Laplacian smoothing*



(b) Laplacian smoothing.

Figure: Visualization of Laplacian smoothing

Prob1—-Removing the rough features

Additional Definition: *1D discrete Laplacian*

- $\delta(v_i) = \frac{1}{2}(v_{i-1} + v_{i+1}) - v_i$
- Called *1D discrete Laplacian*

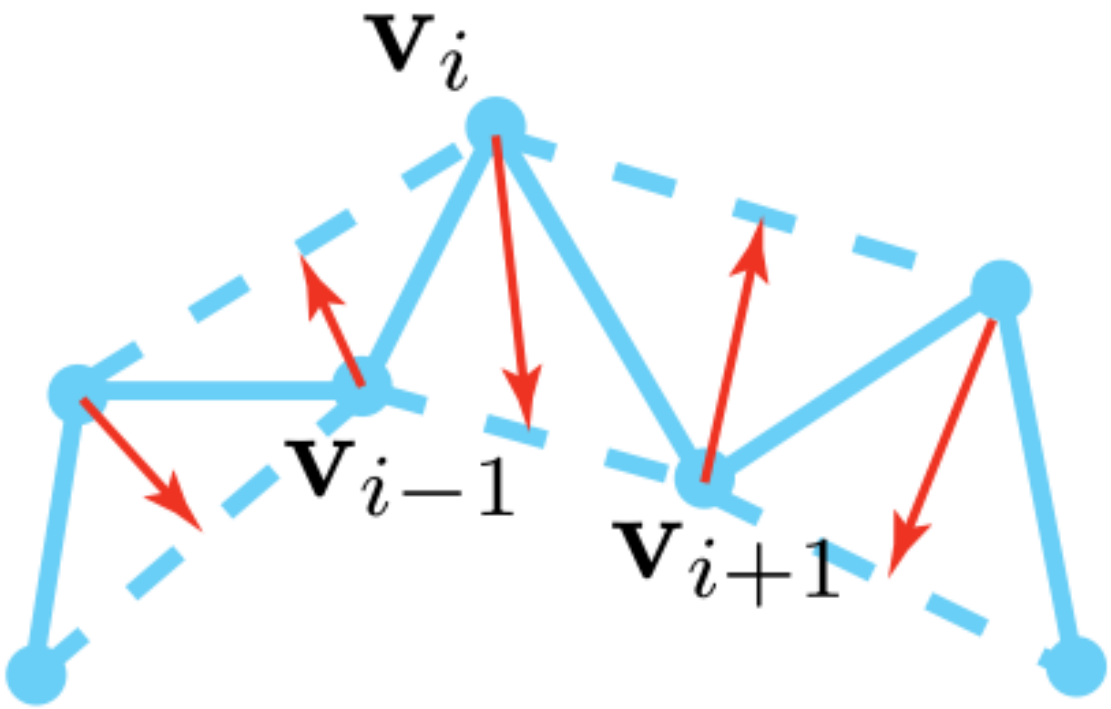


Figure: 1D discrete Laplacian

Prob1—-Removing the rough features

Subquestions about Laplacian smoothing

- Why can Laplacian smoothing remove the rough features (or noise) from the data ?
- What happens if we apply Laplacian smoothing in the limit ?

Prob1—-Removing the rough features

Algebraical Formulation—-*Signal x*

- Let each vertex-of-"seahorse" v_i's coordinates be $[x_i, y_i]$
- consider $\vec{x}^T = [x_1, x_2 \ldots, x_n]$

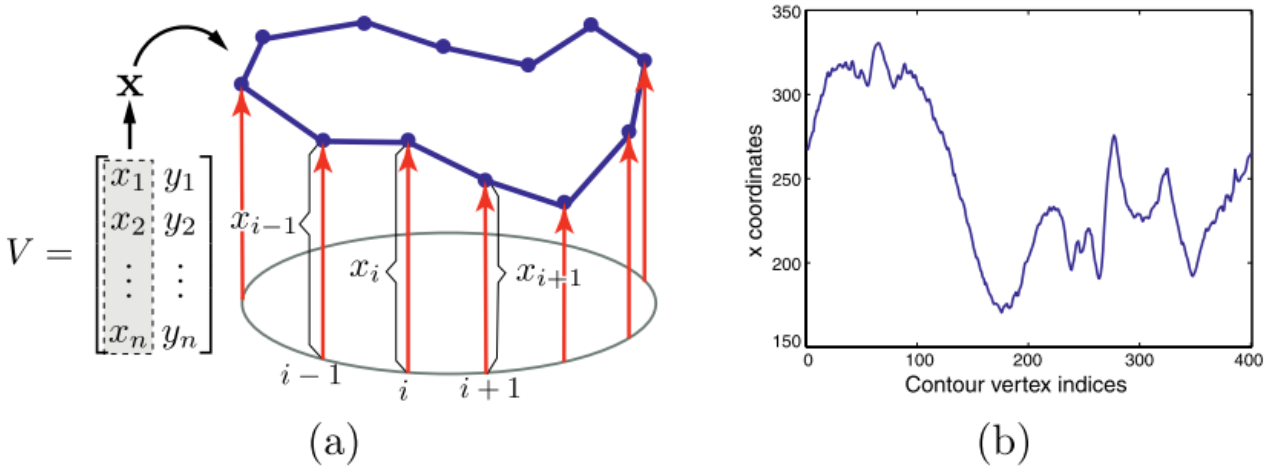


Figure: Visualization of x-coordinates

Prob1—-Removing the rough features

Algebraical Formulation—-*Laplacian smoothing*

- Let S denote the *Laplacian smoothing* to signal $\vec{x}$, consider the matrix of S
- Notice that: $S(x_i) = \frac{1}{4}x_{i-1} + \frac{1}{2}x_i + \frac{1}{4}x_{i+1}$

$$S = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \dots & \dots & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

Figure: Matrix of S

Prob1—-Removing the rough features

Algebraical Formulation—-*1D discrete Laplacian operator*

- Let L denote the *1D discrete Laplacian operator* to signal $\vec{x}$, consider the matrix of L
- Notice that:$\delta(v_i) = \frac{1}{2}(v_{i-1} + v_{i+1}) - v_i$

$$\delta(\mathbf{x}) = L\mathbf{x} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & \dots & \dots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \dots & \dots & 0 & -\frac{1}{2} & 1 \end{bmatrix} \mathbf{x}.$$

Figure: Matrix of L

Prob1—-Removing the rough features

Analyze the *1D discrete Laplacian operator*

- Notice that $S = I - \frac{1}{2}L$, now we consider eigenvalue:$[\lambda_1, \lambda_2 \dots]$ and eigenvectors:$[e_1, e_2 \dots]$ of L
- Then signal x can be denoted by $[e_1, e_2 \dots]$

$$\mathbf{x} = \sum_{i=1}^{n} \tilde{x}_i \, \mathbf{e}_i = \begin{bmatrix} | & | & & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix} = E\,\tilde{\mathbf{x}}.$$

Figure: Express x as a linear combination of the eigenvectors

Prob1—-Removing the rough features

We call this a spectral transform

- convert x to $\widetilde{x}$ under the base of eigenvectors is called *spectral transform*
- Why is it special? You just express x form different base vectors.

Prob1—-Removing the rough features

Why do we call this a spectral transform?

- Then the magic happens!
- Ps: e_i is sorted by ”frequency”

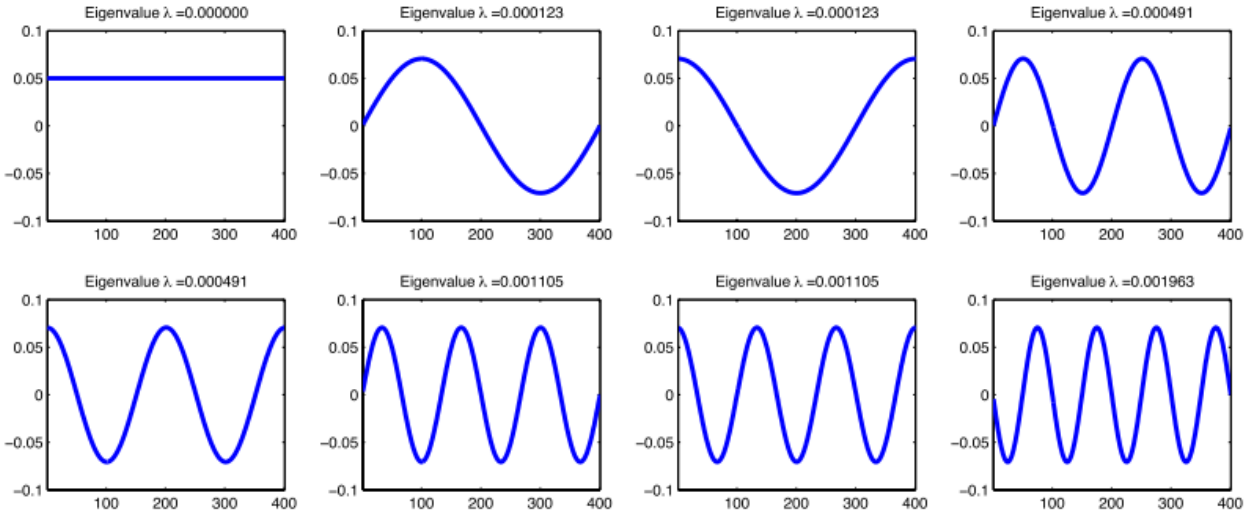


Figure: Magic eigenvectors of matrix L

Prob1—-Removing the rough features

Look back on Fourier series and High/Low frequency information

- $x = \sum_{i=1}^{n} \widetilde{x}_i e_i$
- $f(x) = \frac{a_0}{2} + \sum_{-\infty}^{+\infty} a_k cos(kx)$
- $e_i \sim cos(kx)$
- Ps: e_i is sorted by ”frequency”
- $\widetilde{x}_i \sim a_k$

Prob1—-Removing the rough features

Back to Prob:How to remove the rough features?

- Lemma1:$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$
- Lemma2:Only $\lambda_i = 0$

$$\mathbf{x}^{\langle m \rangle} = S^m \mathbf{x} = \left(I - \frac{1}{2}L\right)^m \mathbf{x} = \sum_{i=1}^{n} \left(I - \frac{1}{2}L\right)^m \mathbf{e}_i \, \tilde{x}_i$$
$$= \sum_{i=1}^{n} \mathbf{e}_i \left(1 - \frac{1}{2}\lambda_i\right)^m \tilde{x}_i \, .$$

Prob1—-Removing the rough features

Back to Prob:How to remove the rough features?

- First row: filter plots, $f(\lambda) = (1 - \frac{1}{2}\lambda)^m$with $m = 1, 5, 10, 50$
- Second row: corresponding results of Laplacian smoothing on the seahorse

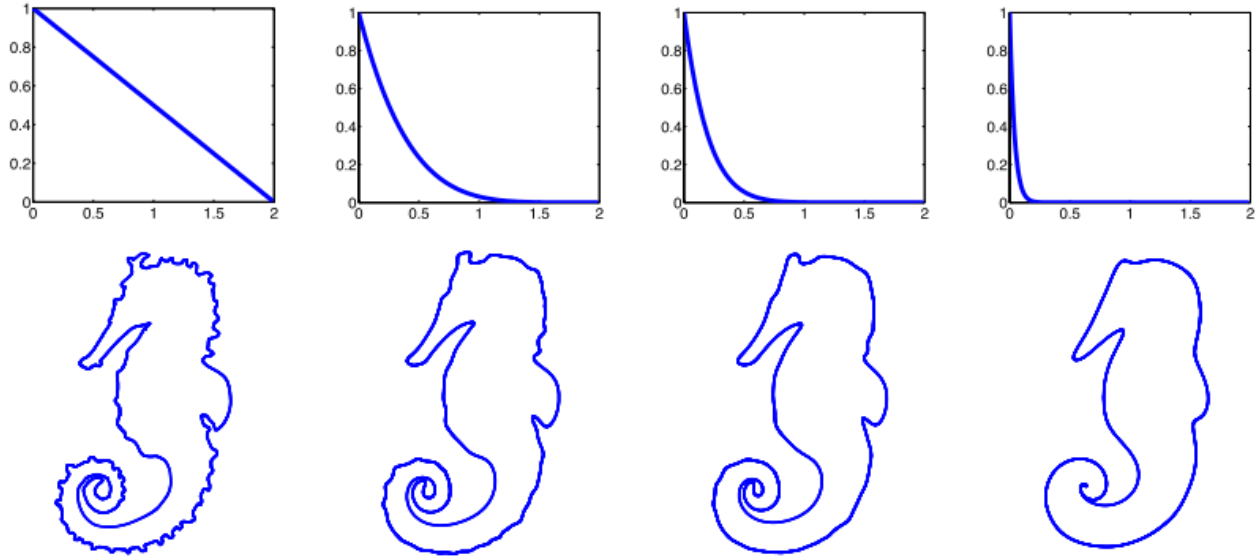


Figure:

Prob1—-Removing the rough features

Back to Prob:Apply Laplacian smoothing to the limit?

- Lemma1:$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$
- Lemma2:Only $\lambda_1 = 0$
- Lemma3:$\lambda_i \in [0, 2]$
- Lemma4:$_i = [\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots]^T$

$$\mathbf{x}^{\langle m \rangle} = S^m \mathbf{x} = \left(I - \frac{1}{2}L\right)^m \mathbf{x} = \sum_{i=1}^{n} \left(I - \frac{1}{2}L\right)^m \mathbf{e}_i \, \tilde{x}_i$$
$$= \sum_{i=1}^{n} \mathbf{e}_i \left(1 - \frac{1}{2}\lambda_i\right)^m \tilde{x}_i \, .$$

Prob1—-Removing the rough features

Back to Prob:Apply Laplacian smoothing to the limit?

- If $\lambda_i \neq 0$ then $(1 - \frac{1}{2}\lambda_i)^m \Rightarrow 0$
- So $x^{<m>} \Rightarrow e_1\widetilde{x}_1 = [\overline{x}, \overline{x}, \ldots] = [\frac{1}{n}\sum_i^n x_i, \frac{1}{n}\sum_i^n x_i, \ldots]$
- Which means all x_i will converge to one point:*centroid*

$$\mathbf{x}^{\langle m \rangle} = S^m \mathbf{x} = \left(I - \frac{1}{2}L\right)^m \mathbf{x} = \sum_{i=1}^n \left(I - \frac{1}{2}L\right)^m \mathbf{e}_i \, \tilde{x}_i$$
$$= \sum_{i=1}^n \mathbf{e}_i \left(1 - \frac{1}{2}\lambda_i\right)^m \tilde{x}_i \, .$$

Prob2—-Compression under certain data loss

Based on former Analysis

- Consider $x^{(k)} = \sum_{i=1}^{k} \widetilde{x}_i e_i, k < n$ as compression of signal $\vec{x}$
- While $x = \sum_{i=1}^{n} \widetilde{x}_i e_i$
- quantify the information loss by measuring the l2 error :

$$\left\|\vec{x} - \vec{x}_{(k)}\right\| = \left\|\sum_{i=k+1}^{n} \widetilde{x}_i e_i\right\| = \sqrt{\sum_{i=k+1}^{n} \widetilde{x}_i^2}$$

Prob2—-Compression under certain data loss

Based on former Analysis

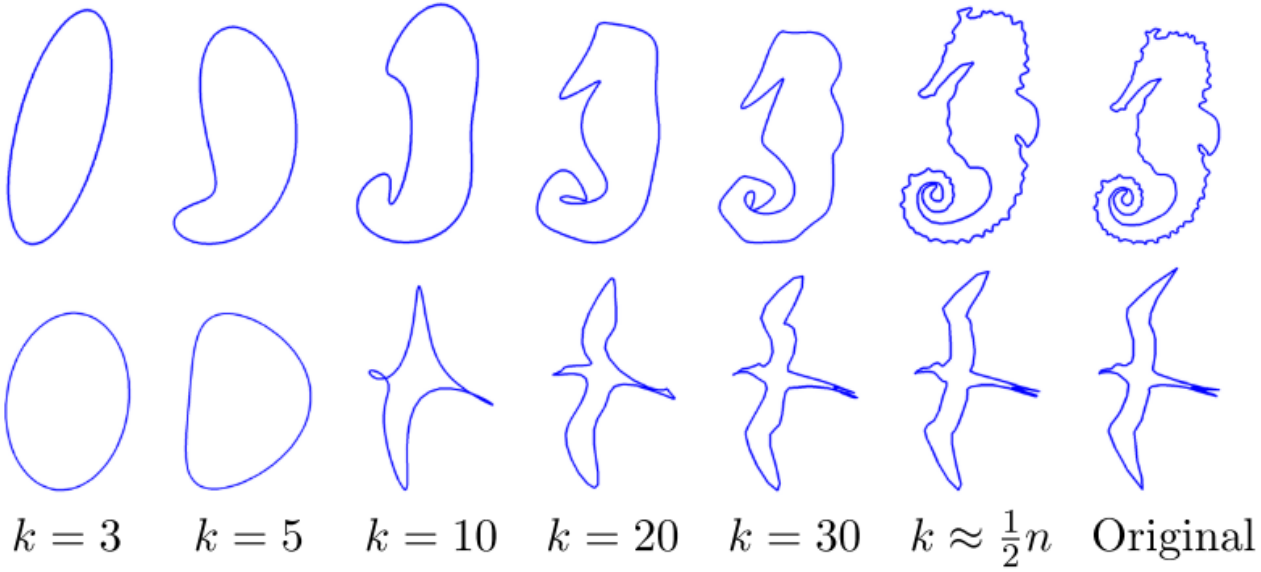


Figure: Shape reconstruction via Laplacian-based spectral analysis.

Prob2—-Compression under certain data loss

Based on former Analysis

- To quantify the performance of this kind of compression, consider plotting $\widetilde{x}$'s coefficients

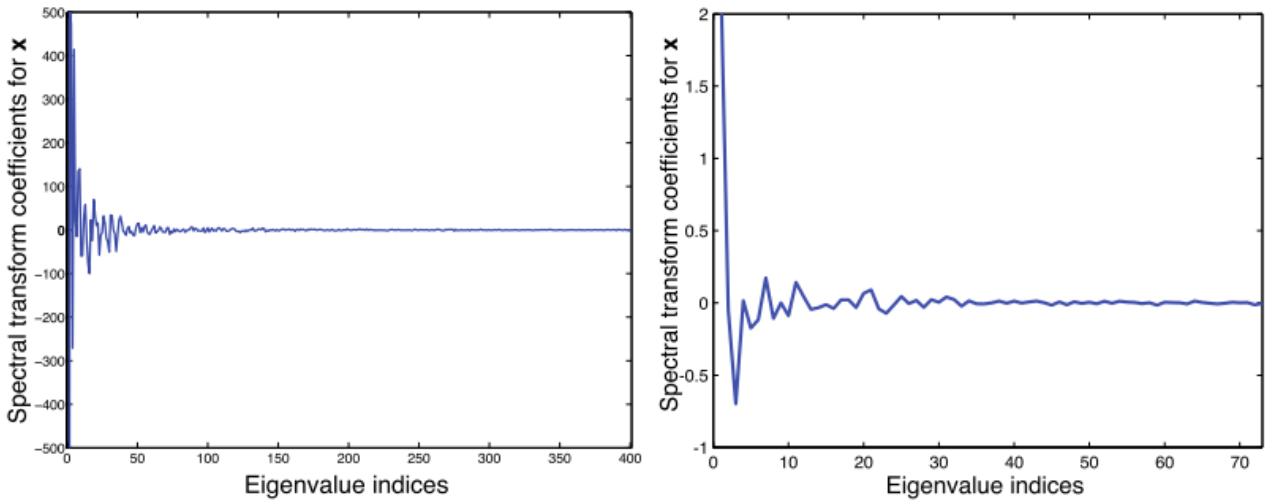


Figure: Plot of spectral transform coefficients for the x component of a contour.

Prob2—-Compression under certain data loss

Based on former Analysis

- A signal whose energies are concentrated in the low-frequency end can be effectively compressed at a high compression rate

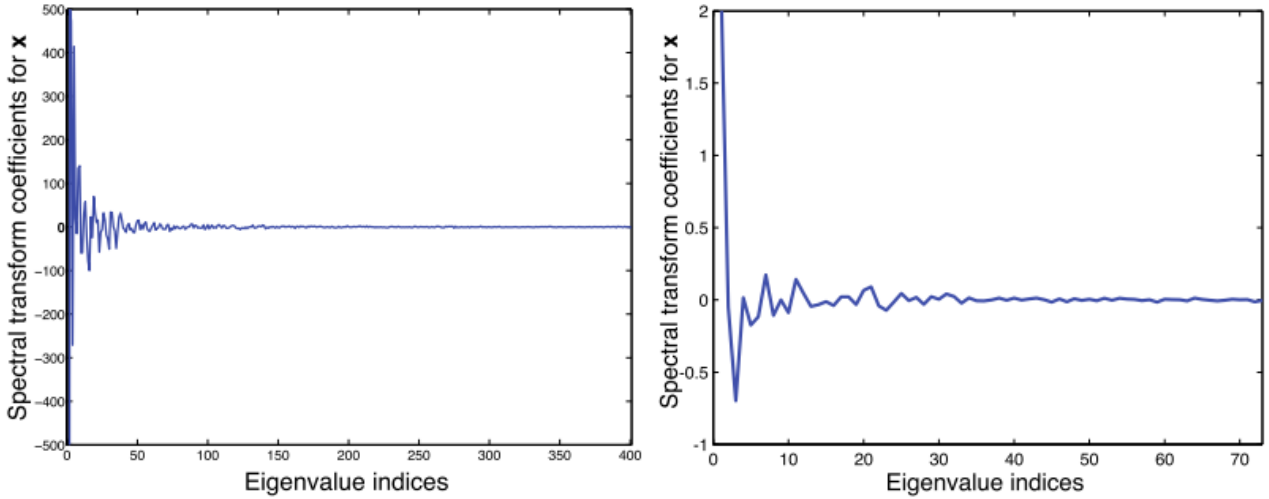


Figure: Notice the low-frequency concentration of x signal

Comparison between SVD and Spectral Transform

In the field of image compression

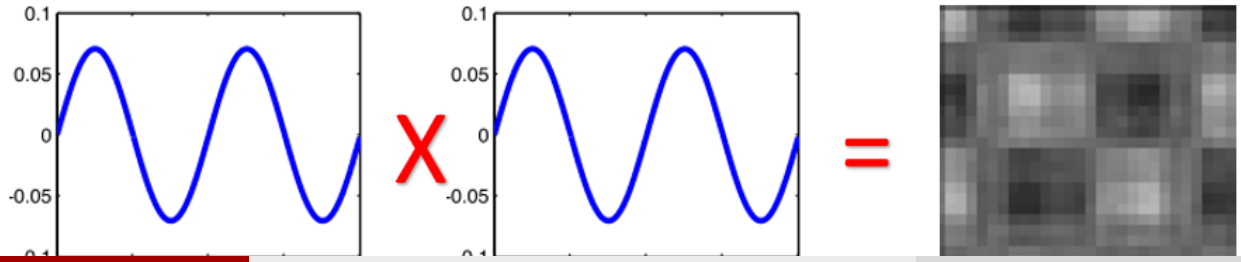
- Totally same as spectral transform , SVD can do imgae compression : $M = \sum_{i=1}^{k} \sigma_i M_i, k < n$

Figure: Image compression by SVD

Comparison between SVD and Spectral Transform

To quantify the efficiency of two method

- SVD: $M = \sum_{i=1}^{k} \sigma_i M_i, k < n$, so its compression rate is $R(k) = \frac{n^2}{k(2n+1)}$ $(M_i = \vec{\mu^T}_i \vec{\lambda}_i)$
- Spectral Transform: $M = \sum_{i=1}^{k} \widetilde{\sigma}_i M_i, k < n$, so its compression rate is $R(k) = \frac{n^2}{k}$
- ps:M_i is 2D-Spectral-Transform's base matrix ,in fact is should be written as $M_{i,j} = e_j e_i^T$



The Discrete Laplacian Operator

from continue to discrete

- the continue Laplacian is easier to understand
- in 3D Cartesian coordinates:
  $$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$
  Example: $\nabla^2 V = -\frac{\rho}{\varepsilon_0}$
- in 1D case, the Laplacian operator is just the second order derivative operator

The Discrete Laplacian Operator

from continue to discrete

- the continue Laplacian is easier to understand
- in 3D Cartesian coordinates:
  $\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$
  Example: $\nabla^2 V = -\frac{\rho}{\varepsilon_0}$
- in 1D case, the Laplacian operator is just the second order derivative operator
- discrete case:
  $\nabla^2 f_i = (f_{i+1} - f_i) - (f_i - f_{i-1}) = f_{i+1} + f_{i-1} - 2f_i = -2Lf_i$

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- solving euqation:
  $Lv = \lambda v$

  where $L = \begin{bmatrix} 1 & -\frac{1}{2} & \dots & -\frac{1}{2} \\ -\frac{1}{2} & 1 & \dots & 0 \\ \vdots & & \dots & \vdots \\ -\frac{1}{2} & 0 & \dots & 1 \end{bmatrix}$ is a Symmetric Matrices
- in equivalent, $Lv_i = v_i - \frac{1}{2}(f_{i-1} + f_{i+1}) = \lambda v_i$

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- solving euqation:
  $Lv = \lambda v$
- in equivalent, $Lv_i = v_i - \frac{1}{2}(f_{i-1} + f_{i+1}) = \lambda v_i$
- Periodic boundary conditions:
  $v_0 = v_n,\ v_{n+1} = v_1$

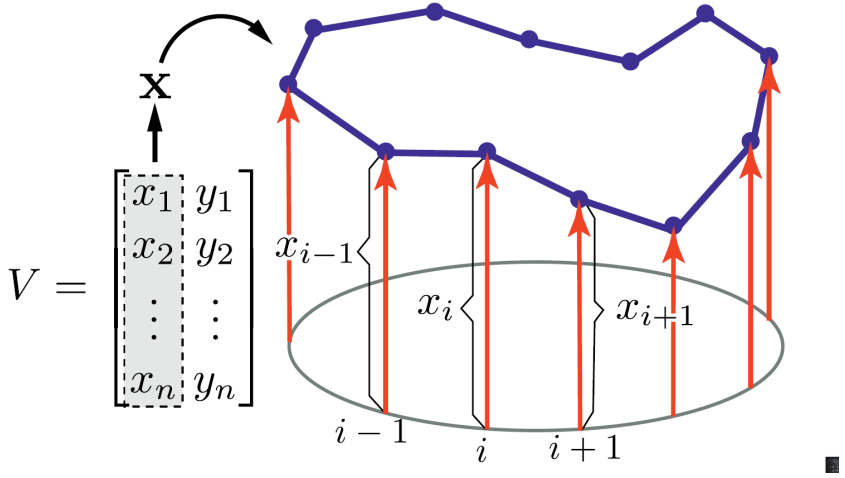


Figure: Periodic boundary

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- solving euqation:
  $Lv = \lambda v$
- in equivalent, $Lv_i = v_i - \frac{1}{2}(f_{i-1} + f_{i+1}) = \lambda v_i$
- Periodic boundary conditions:
  $v_0 = v_n,\ v_{n+1} = v_1$
- the continue case:
  $\frac{d^2 f}{dx^2} = \lambda f$, where f is a periodic function
  $\Rightarrow f = \cos\sqrt{-\lambda}x \ or\ f = \sin\sqrt{-\lambda}x$

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- solving euqation:
  $Lv = \lambda v$
- in equivalent, $Lv_i = v_i - \frac{1}{2}(f_{i-1} + f_{i+1}) = \lambda v_i$
- Periodic boundary conditions:
  $v_0 = v_n,\ v_{n+1} = v_1$
- the continue case:
  $\frac{d^2 f}{dx^2} = \lambda f$, where f is a periodic function
  $\Rightarrow f = \cos\sqrt{-\lambda}x\ or\ f = \sin\sqrt{-\lambda}x$
- the eigenvector might be a trigonometric vector:
  $v_k = [1, \omega_n^k, ..., \omega_n^{(n-1)k}]^T$
  where $\omega = e^{i\frac{2\pi}{n}}$ to fit the periodic boundary condition

The Discrete Laplacian Operator

the Eigenvalue and Eignvector

- Exam it!:

$$
\begin{aligned}
(Lv)_j &= \sum L_{ji} v_i \\
&= v_j - \frac{1}{2}(v_{j-1} + v_{j+1}) \\
&= \omega_n^{kj}[1 - \frac{1}{2}(\omega_n^k + \omega_n^{-k})] \\
&= v_j(1 - \cos\frac{2\pi k}{n})
\end{aligned}
\tag{1}
$$

The Discrete Laplacian Operator

the Eigenvalue and Eignvector

- Exam it!:

$$
\begin{aligned}
(Lv)_j &= \sum L_{ji} v_i \\
&= v_j - \frac{1}{2}(v_{j-1} + v_{j+1}) \\
&= \omega_n^{kj}[1 - \frac{1}{2}(\omega_n^k + \omega_n^{-k})] \\
&= v_j(1 - \cos\frac{2\pi k}{n})
\end{aligned} \tag{1}
$$

- no need to check for boundary situation

The Discrete Laplacian Operator

the Eigenvalue and Eignvector

- Exam it!:

$$
\begin{aligned}
(Lv)_j &= \sum L_{ji} v_i \\
&= v_j - \frac{1}{2}(v_{j-1} + v_{j+1}) \\
&= \omega_n^{kj}[1 - \frac{1}{2}(\omega_n^k + \omega_n^{-k})] \\
&= v_j(1 - \cos\frac{2\pi k}{n})
\end{aligned} \tag{1}
$$

- no need to check for boundary situation
- $\lambda_k = 1 - \cos\frac{2\pi k}{n}$

  notice that $\lambda_k = \lambda_{n-k}$, which means there are only $\frac{n}{2}$ eigenvalues

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- Why?

$v_k = \begin{bmatrix} 1 \\ \omega_n^k \\ \vdots \\ \omega_n^{(n-1)k} \end{bmatrix}$, $v_{-k} = v_{n-k} = \begin{bmatrix} 1 \\ \omega_n^{-k} \\ \vdots \\ \omega_n^{-(n-1)k} \end{bmatrix}$ are both the eigenvalue λ_k's eigenvectors, the dimension of eigenspace is 2.

The Discrete Laplacian Operator

the Eigenvalue and Eigenvector

- Why?

$$v_k = \begin{bmatrix} 1 \\ \omega_n^k \\ \vdots \\ \omega_n^{(n-1)k} \end{bmatrix}, v_{-k} = v_{n-k} = \begin{bmatrix} 1 \\ \omega_n^{-k} \\ \vdots \\ \omega_n^{-(n-1)k} \end{bmatrix}$$ are both the eigenvalue λ_k's eigenvectors, the dimension of eigenspace is 2.

- eigenvector in real space:

$$\alpha_k = \frac{1}{2}(v_k + v_{-k}) = \left[1, \quad \cos\frac{2\pi k}{n} \quad \dots \quad \cos\frac{2\pi k(n-1)}{n}\right]^T$$

$$\beta_k = \frac{1}{2i}(v_k - v_{-k}) = \left[1, \quad \sin\frac{2\pi k}{n} \quad \dots \quad \sin\frac{2\pi k(n-1)}{n}\right]^T$$

Discrete Fourier transform

- Fourier transform is another important transformation concerning frequency
- using the vector v_k defined as above:
  $v_k = [1, \omega_n^k, ..., \omega_n^{(n-1)k}]^T$
- Define: $F = [v_0, v_1, ..., v_{n-1}]$
  we have $F^{-1} = \frac{1}{n}[v_0, v_{-1}, ..., v_{-(n-1)}]^T$

Discrete Fourier transform

- Fourier transform is another important transformation concerning frequency
- using the vector v_k defined as above:
  $v_k = [1, \omega_n^k, ..., \omega_n^{(n-1)k}]^T$
- Define: $F = [v_0, v_1, ..., v_{n-1}]$
  we have $F^{-1} = \frac{1}{n}[v_0, v_{-1}, ..., v_{-(n-1)}]^T$
- the DFT is defined as:
  $\tilde{x}_j = (F^{-1}x)_j = \frac{1}{n}\sum_k x_k (v_j)_k = \frac{1}{n}\sum_k x_k e^{\frac{i2\pi jk}{n}}$
- $x = F\tilde{x} = \sum_j \tilde{x}_j v_j$, from low frequency to high frequency

Compare between LT and DFT

- We can decompose the matrix L using F:
  $L = F^{-1} diag(\lambda_0, ..., \lambda_{n-1}) F$
  so $Lx = F^{-1} diag(\lambda_0, ..., \lambda_{n-1}) Fx$, to change the proportion of coefficient on every frequency
- the laplacian transform is a composition of Fourier transform and coefficient adjustment

Spectral analysis on meshes

- A triangle mesh with n vertices is represented as $\mathcal{M} = (\mathcal{G}, \mathcal{P})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ models the mesh graph, with $\mathcal{V}$ denoting the set of mesh vertices and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ the set of edges, and $\mathcal{P} \in R^{n \times 3}$ represents the geometry of the mesh, given by a matrix of 3D vertex coordinates.
- meshes are more complicated than 1-D situation. Because it has more complex relation between vertices.

Spectral analysis on meshes

- W is the adjacency matrix

$$W_{i,j} = \begin{cases} 1 & if\ (i,j) \in \mathcal{E} \\ 0 & otherwise. \end{cases}$$

- D is a diagonal matrix

$$D_{i,j} = \begin{cases} d_i = |\mathcal{N}(i)| & if\ i = j \\ 0 & otherwise. \end{cases}$$

- $\mathcal{N}(i)$ is the set of immediate neighbor vertices of i

Spectral analysis on meshes

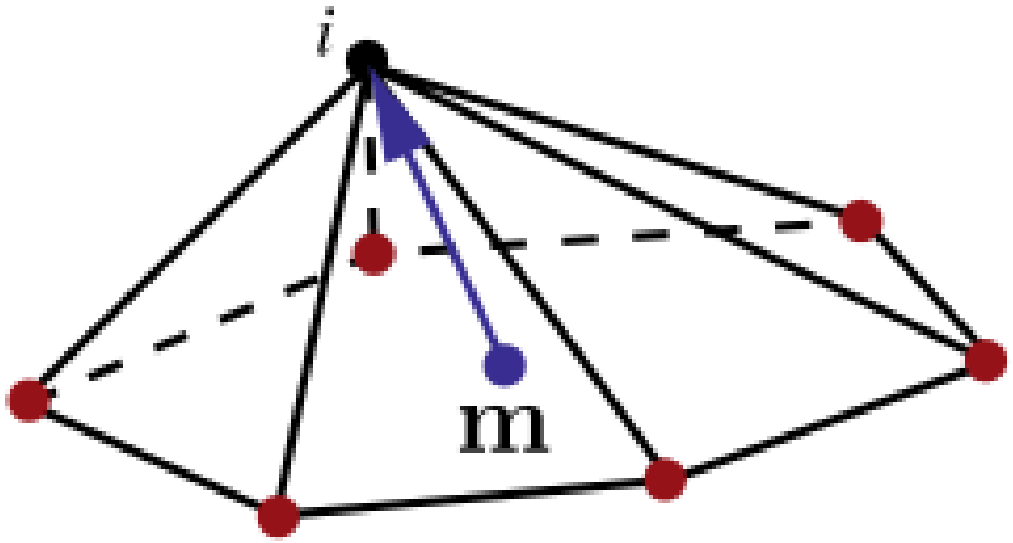


Figure: Laplacian coordinate

Spectral analysis on meshes

- Similarly, we define Laplacian coordinate as:

$$\delta(x_i) = x_i - \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} x_j$$

- we define our mesh Laplacian matrix T as

$$T = I - D^{-1}W$$

Spectral mesh transform

- T is the same as 1D case for L that we saw earlier. Let E be the matrix whose columns are the eigenvectors.
- Denote by $\mathbf{e}_1, \cdots, \mathbf{e}_n$ the normalized eigenvectors of T, corresponding to eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$, and let E be the matrix whose columns are the eigenvectors.

$$x^{(k)} = \sum_{i=1}^{k} \tilde{x}_i e_i$$

Spectral mesh transform

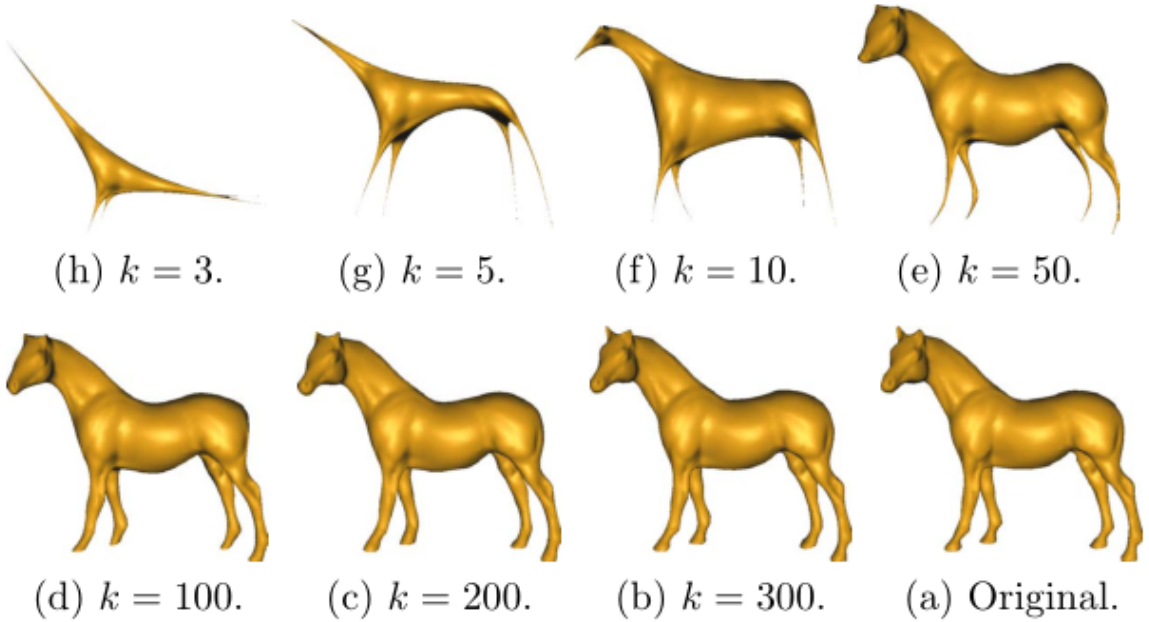


Figure: Laplacian Result

Spectral mesh transform

Also colors...

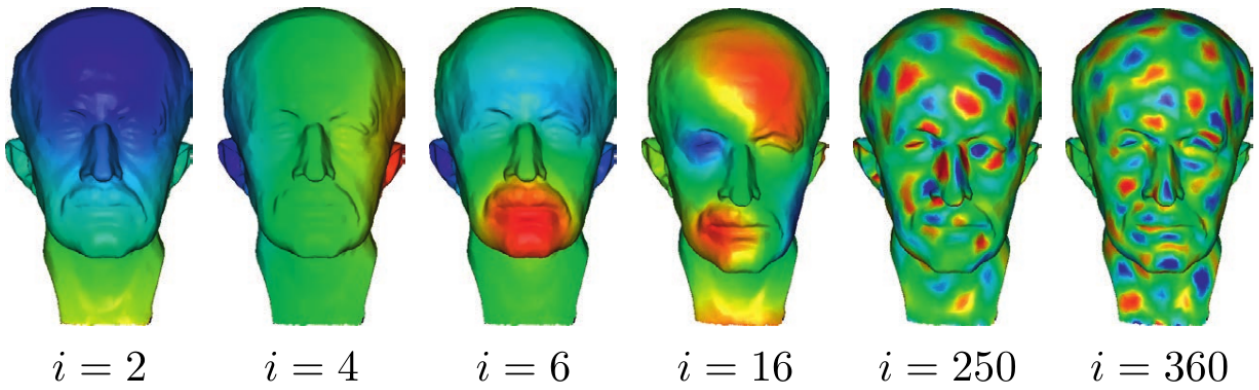


Figure: Laplacian Result

Advantages

- T is depended on adjacency matrix.
- Different Laplacian operator performs well in different surface.(e.g. L^{cot})
- Laplacian operator may calculate slowly. But there are some method to spped up this process, e.g., via spectral shift and invert [VL08],algebraic multi-grid methods, or settling for approximated results [ZvKD10]