CSE 291 I00: Machine Learning for 3D Data
Homework 4

Qiaojun Feng
A53235331
Department of Electrical and Computer Engineering
University of California, San Diego
qjfeng@ucsd.edu

Fig. 1: chain graph

For this assignment we discover the property of specific graph's Laplacian matrix and its eigenvalues.

1. Laplacian of a chain

First we define a chain graph like a line segment with vertices on it. Each vertex is connected with two nearby vertices except for the two vertices at the front and the end. The degree of these vertices is 2. For the front and end vertex, their degree is 1. See Fig. 1 for reference.

Then the Laplacian matrix will look like this:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}$$

For a chain graph of 30 nodes, the Laplacian matrix is 30×30. The eigenvectors are in $\mathbb{R}^{30}$. We visualize the first 20 eigenvectors $v_1, \ldots, v_{20}$ through plotting function with 30 equal spacing point. The values are shown on the y-axis. See Fig. 3 for details.

We can see that these eigenvectors are somehow like the basis functions for Fourier transform. They show some periodic property with the fixed amplitude in most of them. The first eigenvector is the same at each place. The eigenvectors corresponding to larger eigenvalues have higher frequency.

2. Laplacian of a lattice

Similar to chain graph, we take lattice graph as the 2-d situation of chain graph. Most of the vertices have degree of 4, with vertices on the four edges having degree of 3 and four vertices on the four corner having degree of 2. For a a lattice of size 30 by 30, we index them from left to right from left to right. For the node n with degree of 4, it is connected with $[n-30, n-1, n+1, n+30]$. See Fig. 2 for reference.

Now the eigenvectors are in $\mathbb{R}^{30^2}$ and we would like to visualize each eigenvector as a image. See Fig. 4 5 for details.

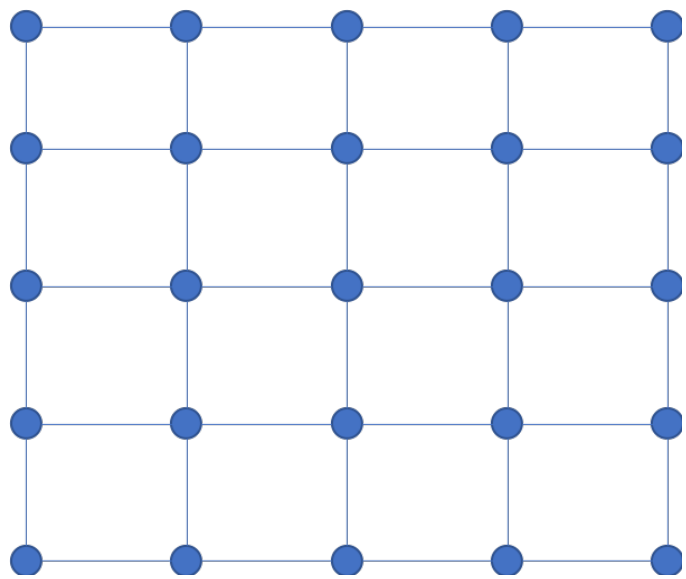

Fig. 2: lattice graph

Also the first eigenvector is the same at each place. The eigenvectors corresponding to larger eigenvalues have higher frequency.

3. Approximation of an image by laplaican bases

Now that we know the eigenvectors of a Laplacian matrix are orthogonal to each other. So they can be a basis of $\mathbb{R}^n$ where n is the dimension of the eigenvectors. Based on our knowledge of linear algebra, any vector in $\mathbb{R}^n$ can be represented as a linear combination of the basis vectors in such basis. Here comes the intuition of our approximation of an image using lattice graph laplacian bases.

We are going to approximate image with size 150×150 so the Laplacian matrix is of the size $150^2 \times 150^2$. Using the same method in 2 we can get 150^2 eigenvectors as basis vectors.

Define this orthogonal basis as $\mathcal{V} = \{v_1, \ldots, v_{150^2}\}$. To get the linear combination representation of arbitrary vector $a \in \mathbb{R}^{150^2}$ we need to calculating the projections of a on these basis vectors v_i

$$a = \sum_{i=1}^{N} c_i v_i$$

$$c_i = \frac{a \cdot v_i}{v_i \cdot v_i}$$

Here N is the number of eigenvectors we used to approximate the origin image.

Fig. 6 7 shows some approximation results. For a image of size 150×150 which is a 150^2-dimension vector, using only 30 eigenvectors can't perform well for the image approximation especially for some complex images as we chose to test. When we are using 300 eigenvectors, things are getting a little bit better. We can roughly see the object in the 300 eigenvectors reconstructed images.

REFERENCES

[1] L14_GraphLaplacian.key
[2] lect01-15.pdf

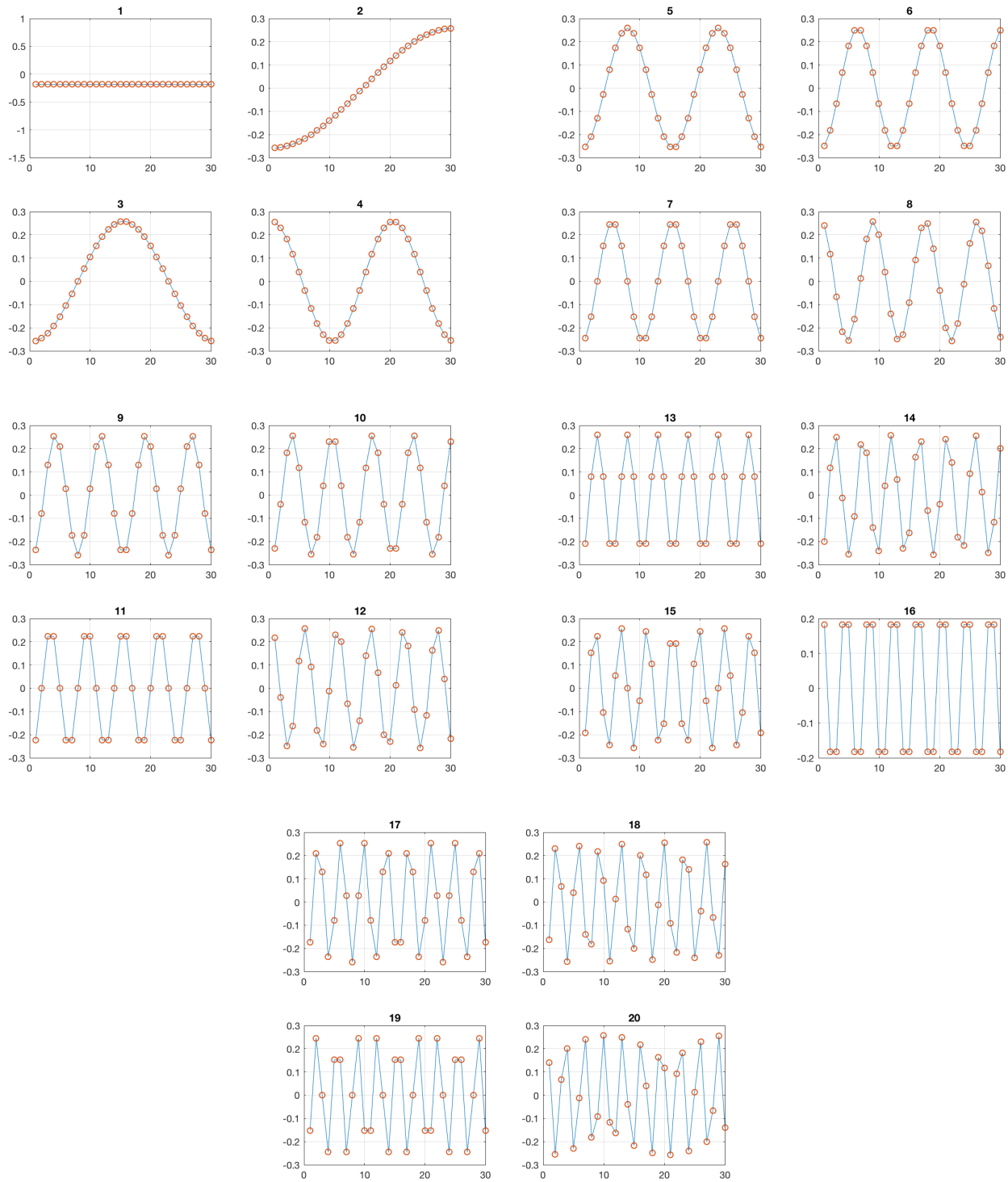

Fig. 3: top 20 Laplacian eigenvectors of a chain graph

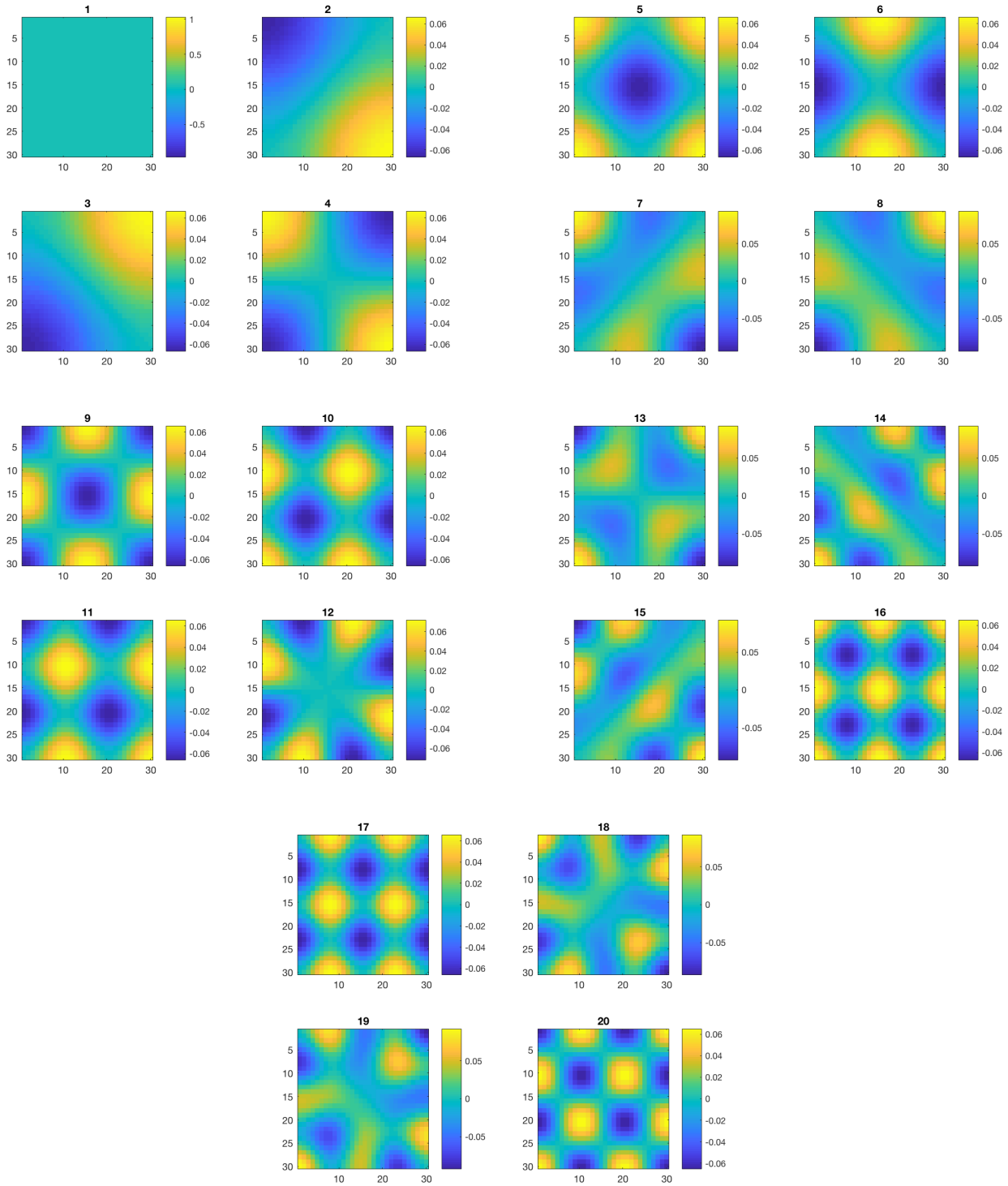

Fig. 4: top 20 Laplacian eigenvectors of a lattice graph

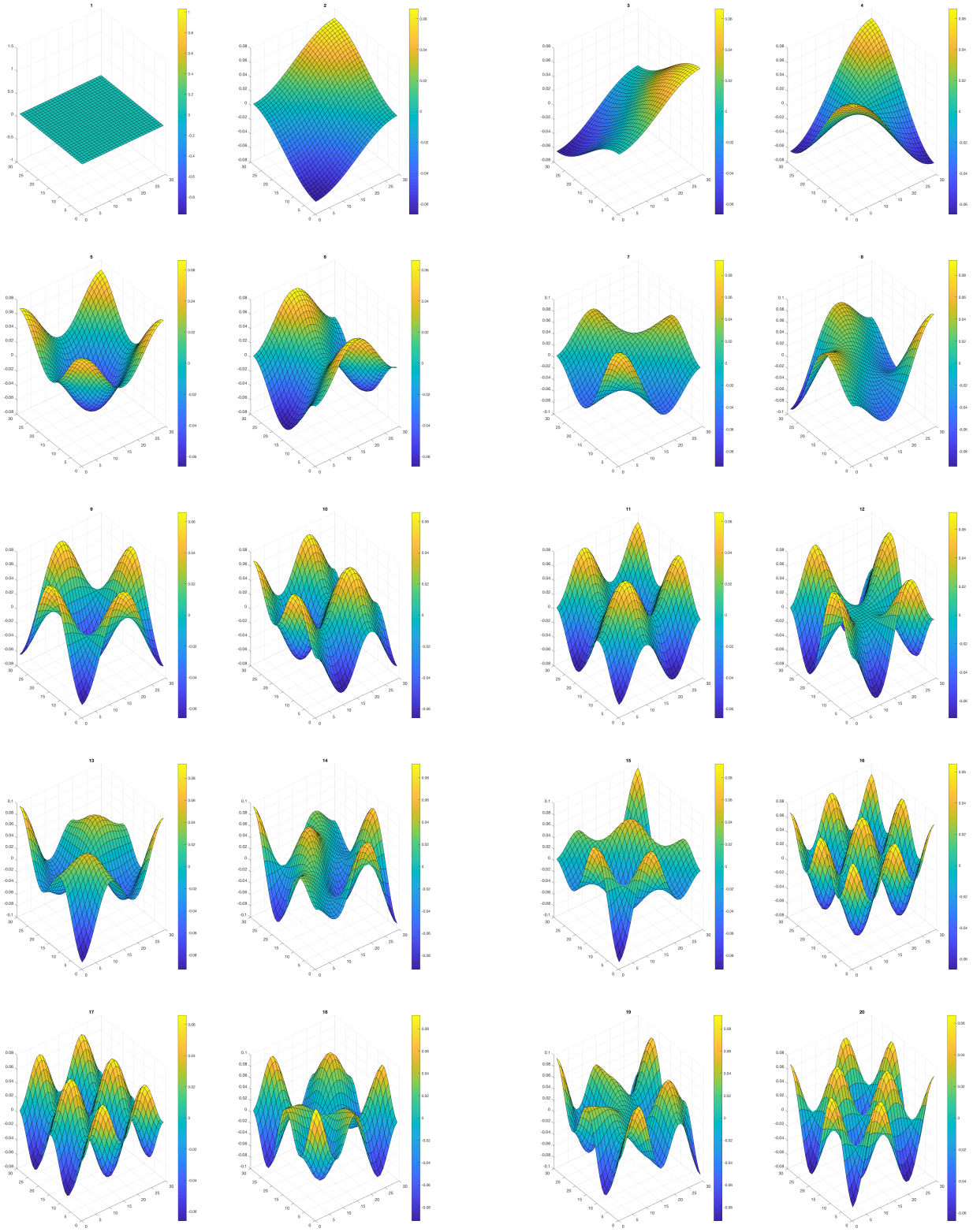

Fig. 5: top 20 Laplacian eigenvectors of a lattice graph

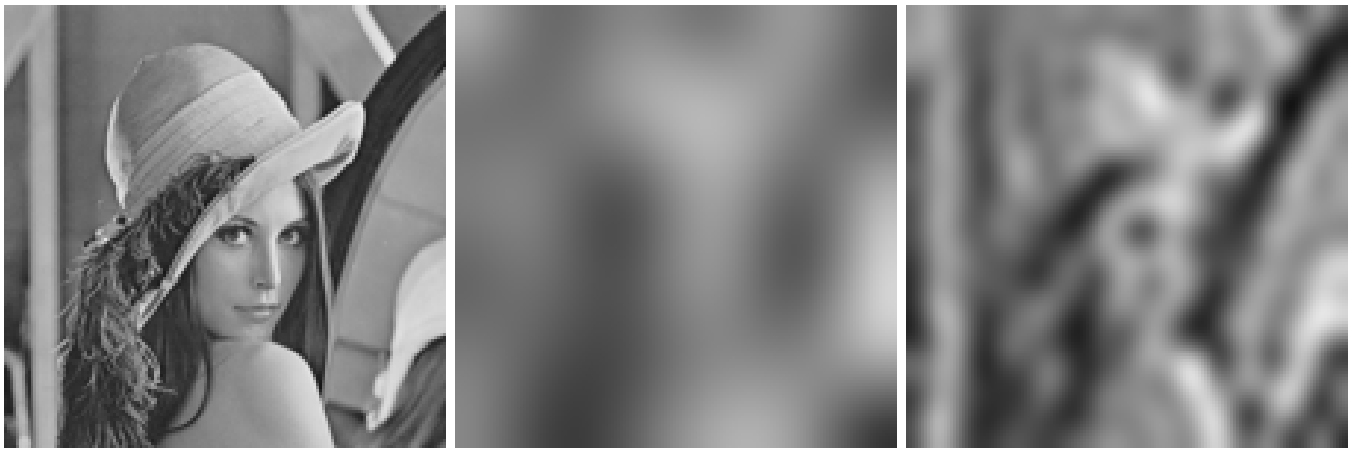

Fig. 6: Laplacian eigenvectors reconstruction: Lena
Left:origin Middle:30 eigenvectors Right:300 eigenvectors

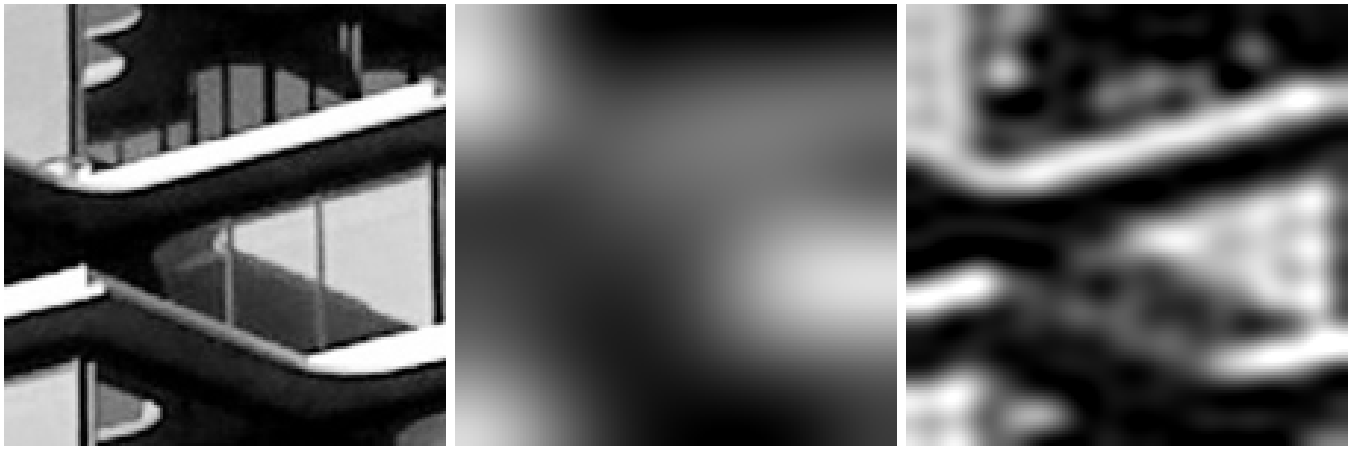

Fig. 7: Laplacian eigenvectors reconstruction: Geisel
Left:origin Middle:30 eigenvectors Right:300 eigenvectors

P.S. MATLAB codes

1. Laplacian of a chain

```
% a chain graph of 30 nodes
% Laplacian Matrix
a = diag(ones(1,29),1);
a = a + a';
la = diag(sum(a)) - a;
% eigenvector
[v,d] = eig(full(la));
d = diag(d);
% plot the first 20
for i =1:5
    figure;
    for j = 1:4
        % index = 31-((i-1)*4+j);
        index = (i-1)*4+j;
        subplot(2,2,j);
        plot(v(:,index));
        hold on;
        plot(v(:,index),'o');
        grid on;
        title(index);
    end
end
```

2. Laplacian of a lattice

```
% a lattice of 30 by 30
% Laplacian Matrix
a = zeros(900,900);
for i = 1:900
    neigs_i = q2_neighbor_lattice(i,30);
    for j = 1:length(neigs_i)
        a(i,neigs_i(j)) = 1;
    end
end
la = diag(sum(a)) - a;
% eigenvector
[v,d] = eig(full(la));
d = diag(d);
% plot the first 20
for i =1:5
    figure;
    for j = 1:4
        % index = 901-((i-1)*4+j);
        index = (i-1)*4+j;
        subplot(2,2,j);
        colormap(gray);
        imagesc(reshape(v(:,index),30,30));
        title(index);
    end
end

function neigs = q2_neighbor_lattice(index,size)
```

```
    neigs = [];
    if index-1>=1 && mod(index-1,size)~=0
        neigs = [neigs,index-1];
    end
    if index+1<=size^2 && mod(index+1,size)~=1
        neigs = [neigs,index+1];
    end
    if index-size>=1
        neigs = [neigs,index-size];
    end
    if index+size<=size^2
        neigs = [neigs,index+size];
    end
end
```

3. Approximation of an image by laplaican bases

```
% Laplacian Matrix
img_size = size(img,1);
node_num = img_size^2;
a = zeros(node_num,node_num);
for i = 1:node_num
    neigs_i = q2_neighbor_lattice(i,img_size);
    for j = 1:length(neigs_i)
        a(i,neigs_i(j)) = 1;
    end
end
la = diag(sum(a)) - a;
% eigenvector
[v,d] = eig(full(la));
d = diag(d);
% directly read the stored first 500 eigenvectors
% load('v_500.mat');
% v = v_500;
img = imread('geisel_2.png');
img_size = size(img,1);
n = 30;
% the projection on first n eigenvectors
% plot the first n
img_vec = double(reshape(img,img_size*img_size,1));
coeff = zeros(n,1);
for i =1:n
    coeff(i) = q3_projection(img_vec,v(:,i));
end
% linear combination of eigenvectors
img_vec_recon = zeros(size(img_vec));
for i =1:n
    img_vec_recon = img_vec_recon + coeff(i).*v(:,i);
end
figure;
%subplot(1,3,1)
imshow(imresize(img,5,'nearest'));
title('origin img')
figure;
%subplot(1,3,2)
img_recon = uint8(reshape(img_vec_recon,img_size,img_size));
imshow(imresize(img_recon,5,'nearest'));
```

```
title(sprintf('reconstructed with %d eigenvectors',n));
figure;
%subplot(1,3,3)
img_error = uint8(abs(double(img_recon)-double(img)));
imshow(imresize(img_error,5,'nearest'));
title('error img');

function aa = q3_projection(a,b)
    aa = sum(a.*b) / sum(b.^2);
end
```