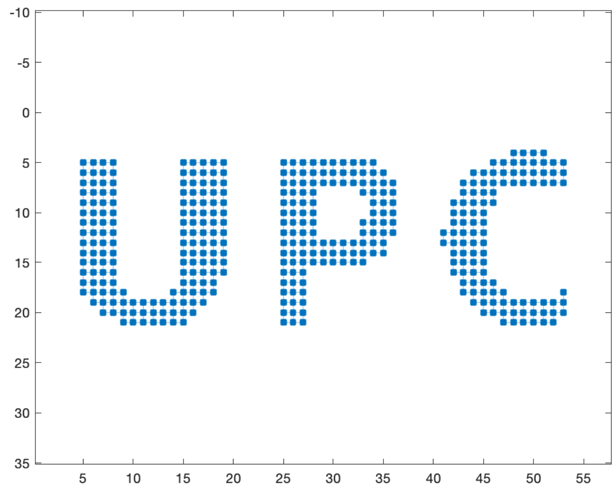
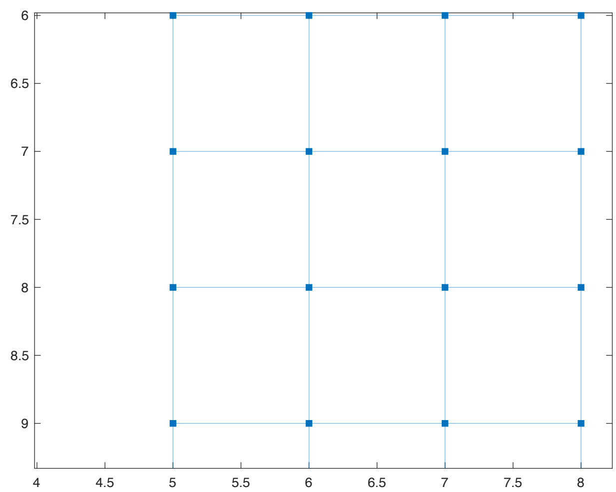
1. Load a png file (e.g. ‘upc.png’).

2. Build a regular grid.

```
bw=imread('upc.png');
bw=bw(:,:,1);
[i,j]=find(bw==0);
for ind=1:length(i)
bw(i(ind),j(ind))=1;
end;
[i,j]=find(bw>1);
for ind=1:length(i)
bw(i(ind),j(ind))=0;
end;
[g,nodenums] = binaryImageGraph(bw,4);
```

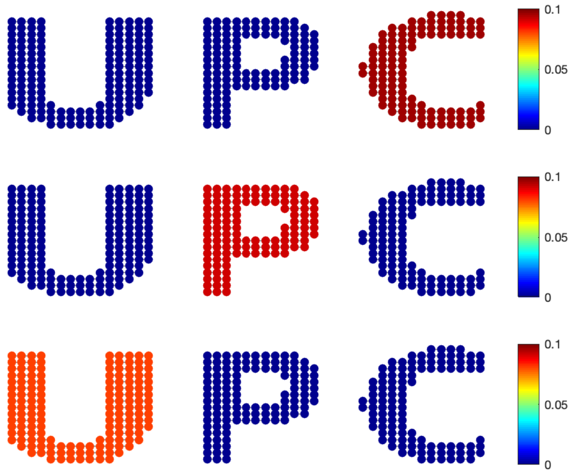


A zoom would look like



3. Spectral analysis of the undirected unweighted graph

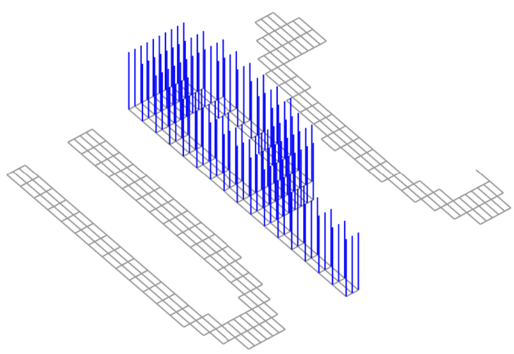
```
W=adjacency(g);
G=gsp_graph(W,[xcoor ycoor]);
G = gsp_compute_fourier_basis(G);
param.climits=[0 0.1];
figure(2)
subplot(311);gsp_plot_signal(G,G.U(:,1),param);
subplot(312);gsp_plot_signal(G,G.U(:,2),param);
subplot(313);gsp_plot_signal(G,G.U(:,3),param);
```



In this example, 3 eigenvectors associated to the nul eigenvalue -> Three communities.

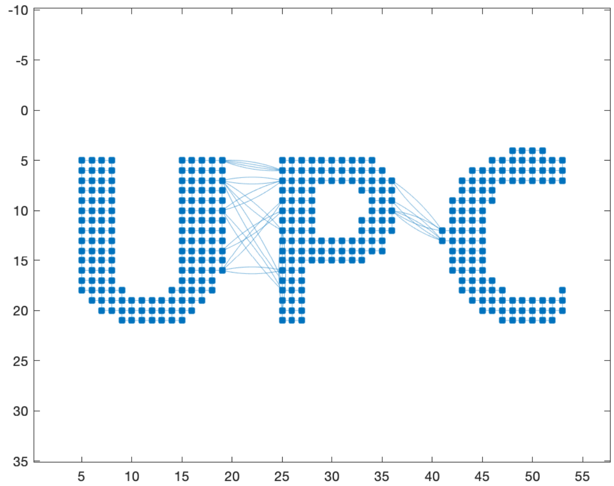
4. Representation of graph signals in 3D.

```
param.bar=1;
figure(3)
gsp_plot_signal(G,G.U(:,2),param)
```

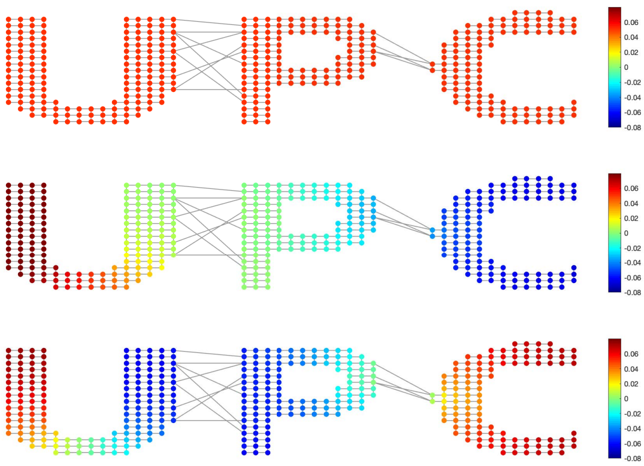


5. Here I add some extra edges (from U to P, and from P to C) to build a connected graph.

```
newedg=[144 157;144 157; 146 158; 146 163; 146 168 ; 149 158;
149 169; 153 161; 163 171; 155 160; 155 167;273 276; 273 277;
270 276; 272 277];
g=addedge(g,newedg(:,1),newedg(:,2),ones(1,size(newedg,1)));
xcoor = g.Nodes.x;
ycoor = size(nodenums,2)-g.Nodes.y; % Flip to proper plot.
figure(1);plotImageGraph(g)
```

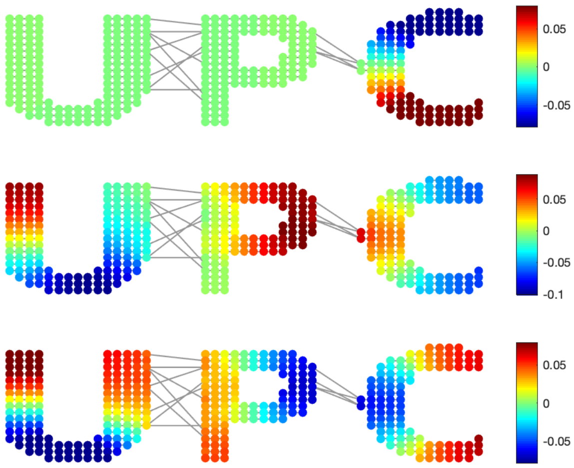



```
W=adjacency(g);
G=gsp_graph(W,[xcoor ycoor]);
G = gsp_compute_fourier_basis(G);
figure(4)
param.climits=[-0.08 0.08];
param.bar=0;
subplot(311);gsp_plot_signal(G,G.U(:,1),param);
subplot(312);gsp_plot_signal(G,G.U(:,2),param);
subplot(313);gsp_plot_signal(G,G.U(:,3),param);
```



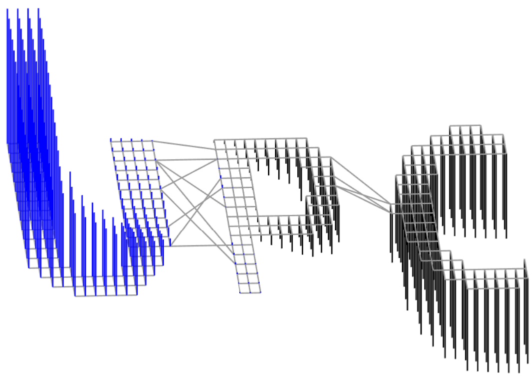
Now, the first 3 eigenvectors look like that. Notice the variation of the signal across the graph as the eigenvalue increases.

Eigenvectors 4, 5, 6

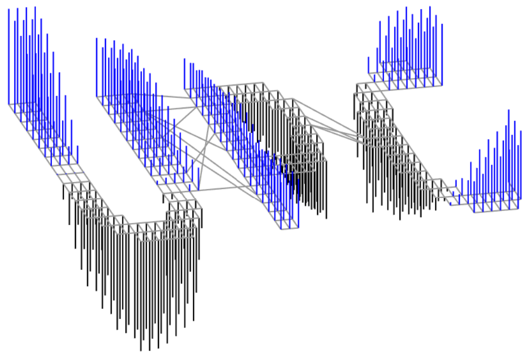


6. Second eigenvector as a graph signal (representation in 3D).

```
param.bar=1;
figure(6)
gsp_plot_signal(G,G.U(:,2),param)
```

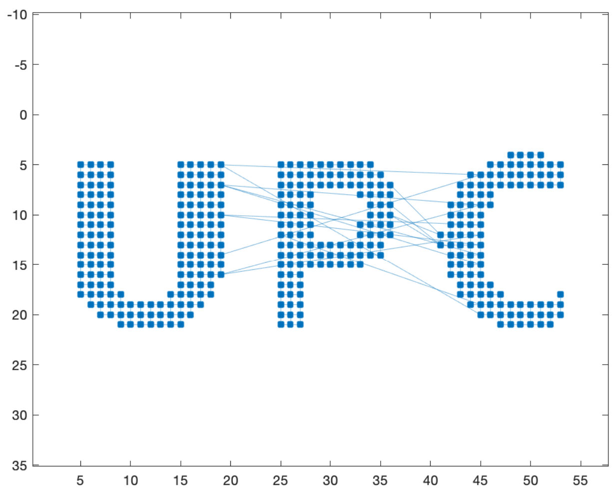


The same for the 6th eigenvector.



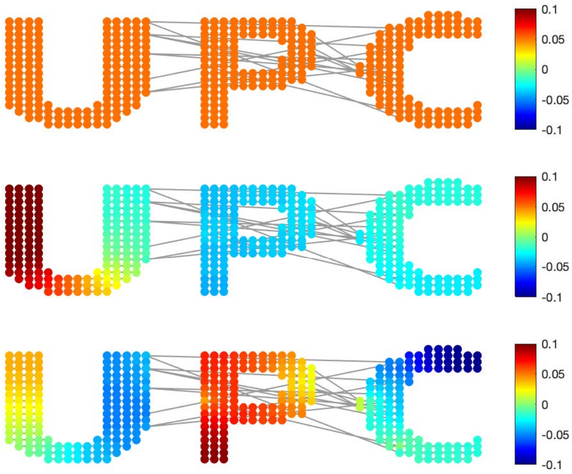
7. An additional test. Now U is connected to C, and C to P. Notice the change in the variation of the eigenvectors across the graph.

```
[g,nodenums] = binaryImageGraph(bw,4);
newedg=[144 312;144 326; 146 315; 146 320; 146 321 ; 149 317; 149
319; 153 312; 163 325; 155 315; 155 318;273 276; 273 277; 270 276;
272 277];
g=addedge(g,newedg(:,1),newedg(:,2),ones(1,size(newedg,1)));
xcoor = g.Nodes.x;
ycoor = size(nodenums,2)-g.Nodes.y; % Flip to proper plot.
figure(1);plotImageGraph(g)
```



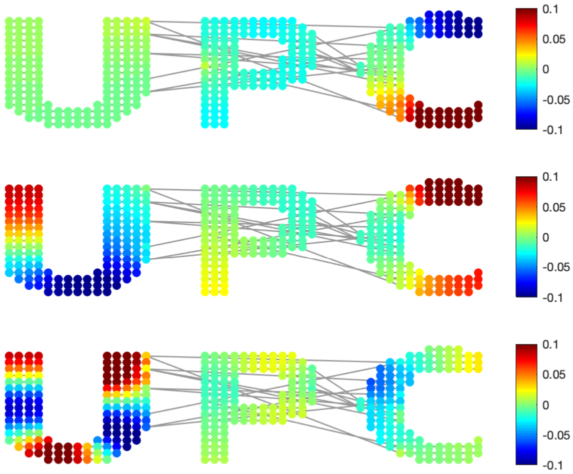

```
W=adjacency(g);
G=gsp_graph(W,[xcoor ycoor]);
G = gsp_compute_fourier_basis(G);
figure(4)
param.climits=[-0.1 0.1];
param.bar=0;
subplot(311);gsp_plot_signal(G,G.U(:,1),param);
subplot(312);gsp_plot_signal(G,G.U(:,2),param);
subplot(313);gsp_plot_signal(G,G.U(:,3),param);
```

The first 3 eigenvectors.

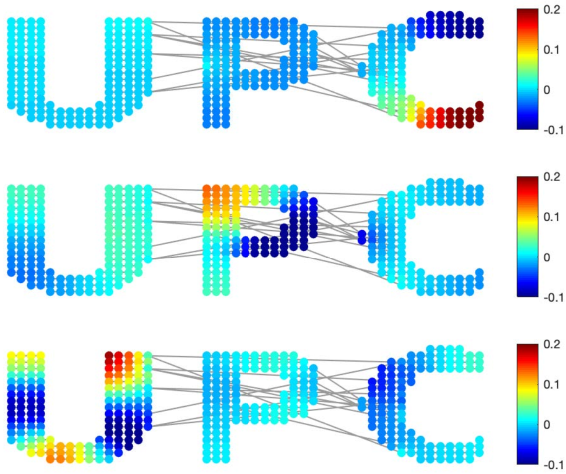



```
figure(5)
param.climits=[-0.1 0.2];
subplot(311);gsp_plot_signal(G,G.U(:,4),param);
subplot(312);gsp_plot_signal(G,G.U(:,8),param);
subplot(313);gsp_plot_signal(G,G.U(:,12),param);
```

Eigenvectors 4,5,6



Eigenvectors 4, 8,12 (pay attention to the range.)


0.2
0.1
0
-0.1
0.2
0.1
0
-0.1
0.2
0.1
0
-0.1