(a)

```
function NL = CreateImageGraph(U)

[rows, cols]=size(U);
n = rows*cols;

sigdist = 150;
sigint = 0.002;
W = zeros(n);

for i=1:rows
    for j=1:cols
       current = U(i,j);
       if (i-1)>0
           top = U(i-1,j);
           W(j+(i-1)*cols,j+(i-2)*cols) = exp(-1/sigdist)...
               *exp(-(current-top)^2/sigint);
       end
       if (i-1)>0 && (j-1)>0
           topleft = U(i-1,j-1);
           W(j+(i-1)*cols,(j-1)+(i-2)*cols) = exp(-2/sigdist)...
               *exp(-(current-topleft)^2/sigint);
       end
       if (i-1)>0 && (j+1)<=cols
           topright = U(i-1,j+1);
           W(j+(i-1)*cols,(j+1)+(i-2)*cols) = exp(-2/sigdist)...
               *exp(-(current-topright)^2/sigint);
       end
       if (j-1)>0
           left = U(i,j-1);
           W(j+(i-1)*cols,(j-1)+(i-1)*cols) = exp(-1/sigdist)...
               *exp(-(current-left)^2/sigint);
       end
       if (j+1)<=cols
           right = U(i,j+1);
           W(j+(i-1)*cols,(j+1)+(i-1)*cols) = exp(-1/sigdist)...
               *exp(-(current-right)^2/sigint);
       end
       if (i+1)<=rows
           bottom = U(i+1,j);
           W(j+(i-1)*cols,j+i*cols) = exp(-1/sigdist)...
               *exp(-(current-bottom)^2/sigint);
       end
       if (i+1)<=rows && (j-1)>0
           bottomleft = U(i+1,j-1);
           W(j+(i-1)*cols,(j-1)+i*cols) = exp(-2/sigdist)...
               *exp(-(current-bottomleft)^2/sigint);
       end
       if (i+1)<=rows && (j+1)<=cols
```

```
            bottomright = U(i+1,j+1);
            W(j+(i-1)*cols,(j+1)+i*cols) = exp(-2/sigdist)...
                *exp(-(current-bottomright)^2/sigint);
        end
    end
end

D = diag(sum(W));
NL = sparse(eye(n) - D^(-1/2)*W*D^(-1/2));
```

(b)

```
%
% CS475/675: Assignment 4
%
%   Cell image segmentation
%


%
% Read in a block from a cell image
%
U = imread('cellimage.tif');
U = U(90:190,190:290);
U = double(U);
U = U/max(U(:));


%
% Create the normalized graph Laplacian from image U
% You will need to implement this function for ***part (a).***

NL = CreateImageGraph(U);




%
% Perform normalized spectral clustering
%

%*** Provide your code here for part (b)! ***
K=9;
```

```
p=zeros(size(NL,1),K);
[V, D] = eigs(NL, K, 'smallestabs');
for i=1:size(V,1)
    p(i,:)=V(i,:)/norm(V(i,:));
end

index = kmeans(p,K,'replicates',20);
%result should be the variable 'index' produced by Matlabs kmeans command,
%i.e., a vector of length m*n containing the cluster index for each pixel



%
% Extract segments for the expected cell region in a simple way
%
Clusters = reshape(index,size(U,1),size(U,2));
Clusters = Clusters';
Disk = fspecial('disk',floor(size(U,1)/2));
Disk = Disk>0;

Cell = zeros(size(U));
for k=1:K
    seg_size = nnz(Clusters==k);
    overlap = (Clusters==(Disk*k));
    in_size = nnz(overlap);
    if in_size == seg_size
        Cell = Cell + (Clusters==k);
    end
end
Cell = 2*(Cell-0.5);


%
% Visualize segmentation results
%
figure(1);

%input image
subplot(1,3,1);
imshow(U,[]);

%generated clusters
subplot(1,3,2);
imshow(Clusters,[]);

%segmented result
subplot(1,3,3);
imshow(U,[]);
hold on;
contour(Cell,[0 0],'r', 'linewidth', 1.5);
```

```
hold off;
```