## Introduction

In this lab, the students had to implement the K-means algorithm for clustering unlabeled data. In unsupervised learning, natural clusters within unlabeled data samples (i.e. with no categorical information) may be identified using an iterative learning process. When the functional form of the underlying probability densities of the data are assumed to be known, the only thing that must be learnt is the value of an unknown parameter vector. One elementary but popular approximate method that performs the above is the k-means clustering algo- rithm. The goal of the k-means clustering algorithm is to identify k mean vectors or cluster centres within the given unlabeled data. n the k-means clustering algorithm, we begin with randomly initializing the mean vectors (k cluster centers) and then assigning the data points to the nearest cluster by computing the Euclidean distance. Once all the data points are assigned to one of the k clusters, the mean vectors of the k clusters are recomputed. The process is repeated until there is no change observed in the recomputed mean vectors of the k clusters

## Results

### Part 1: For c = 2

**Table 1: Initial Mean Values**

| [R, G, B] |
|---|
| [33.6532  243.8143  15.2438] |
| [240.2229  146.6782  59.8689] |

**Table 2 : Final Mean (Mu) values:**

| | | |
|---|---|---|
| [136.3930 | 91.4700 | 93.6969] |
| [162.1561 | 196.6642 | 216.1052] |

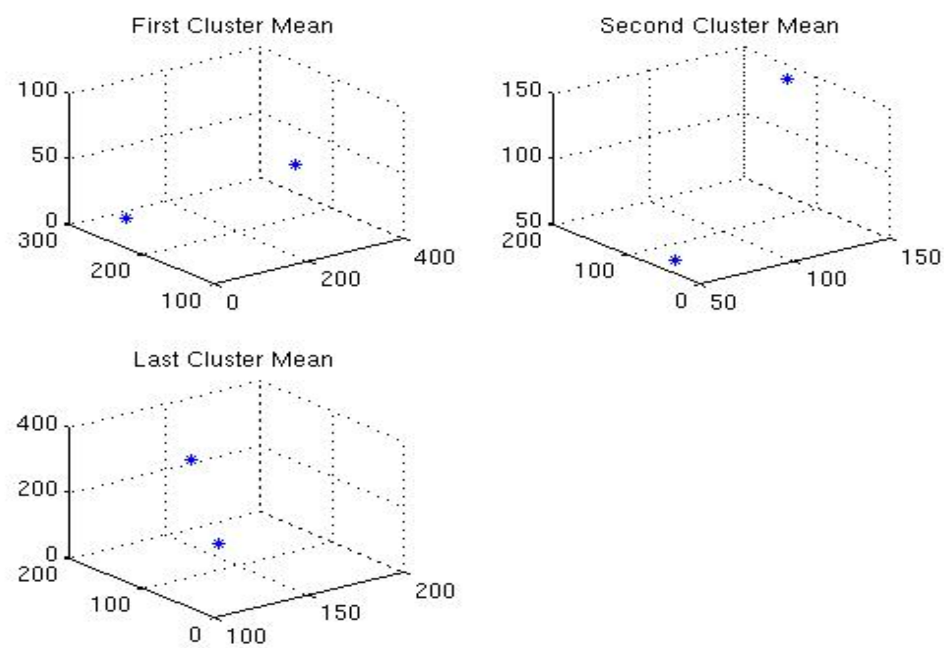


Figure 1: Cluster Mean For 3 Stages

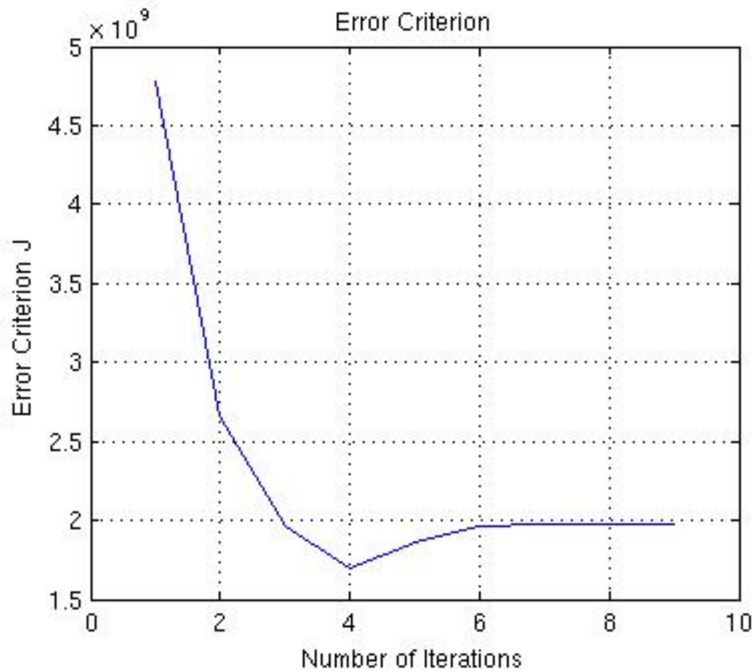


Figure 2: Error Criterion

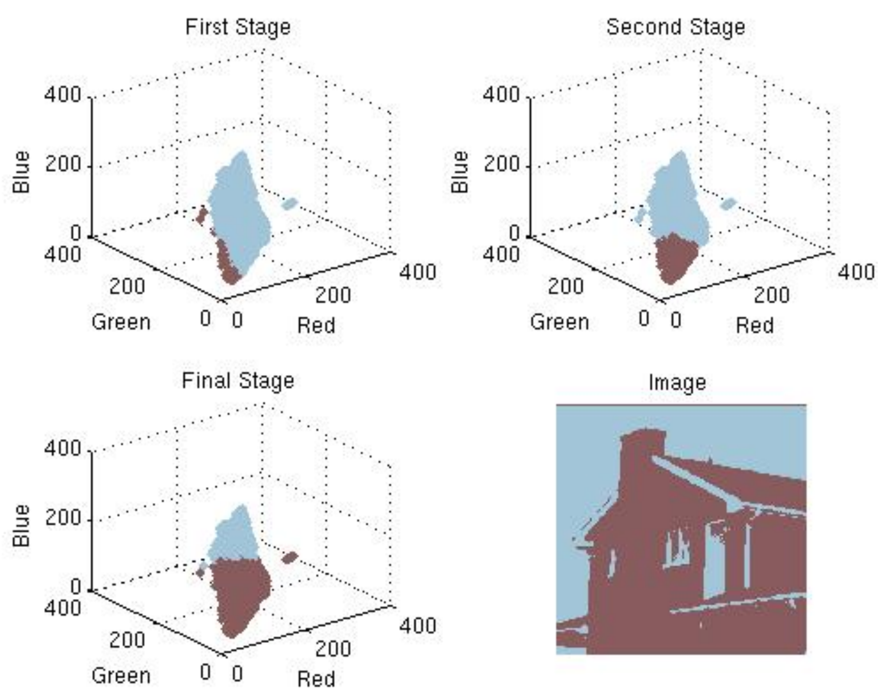


Figure 3: Data Samples in RGB Space with Image

## Part 2: c = 5

### First Run

Table 3: Initial Mean Values

[93.9636  197.8067  129.6697]
[159.5327  124.1319  130.2467]
[198.9580  111.1439  208.4951]
[20.6871  113.9299  202.6820]
[236.9934   78.1191  164.3011]

Table 4: End Mean Values

[140.2193  154.5575  159.5643]
[166.4945  106.3712   96.4801]
[90.7688   54.9372   71.6931]
[163.7433  199.6181  220.1403]
[119.8934   91.6135  105.3736]

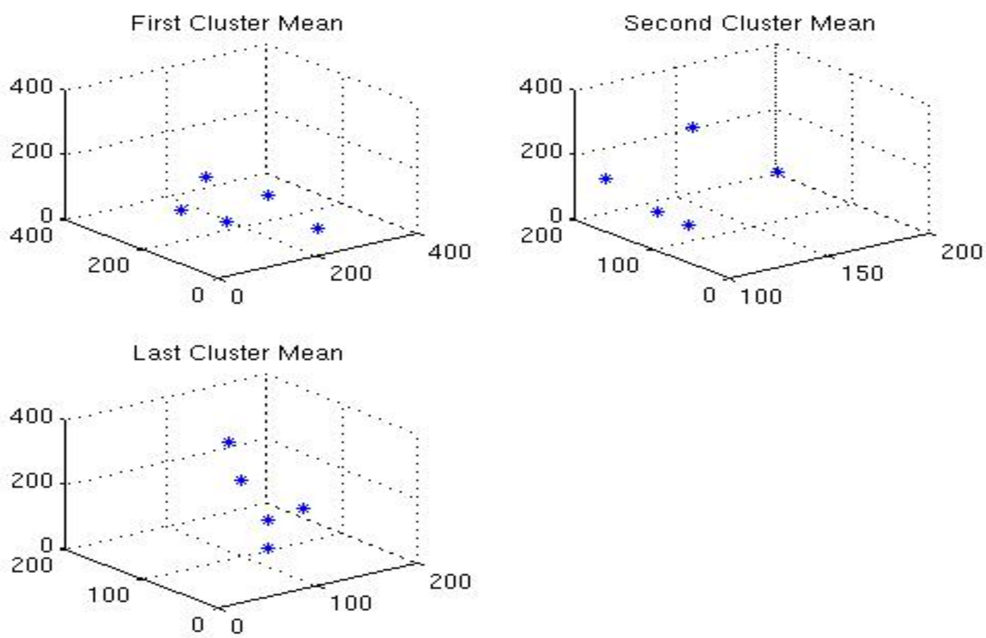


Figure 4: Cluster Mean For 3 Stages

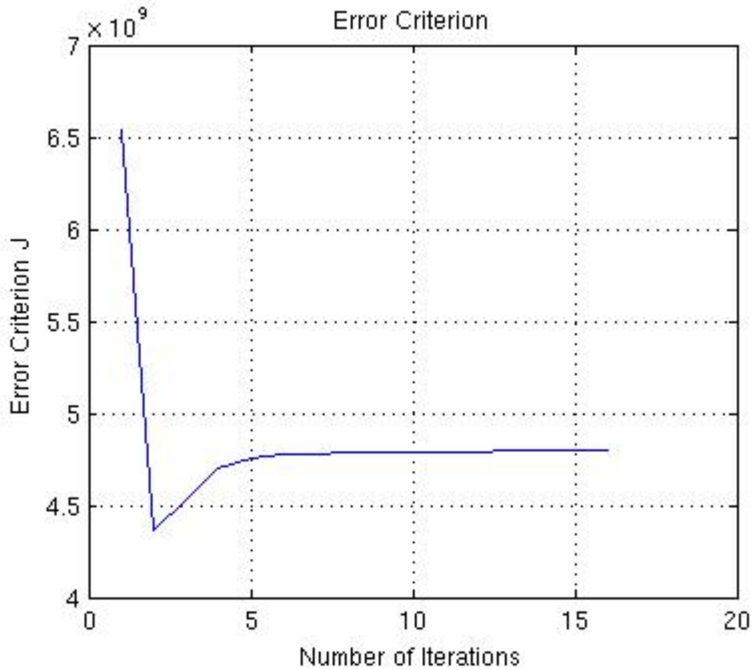


Figure 5: Error Crieterion J

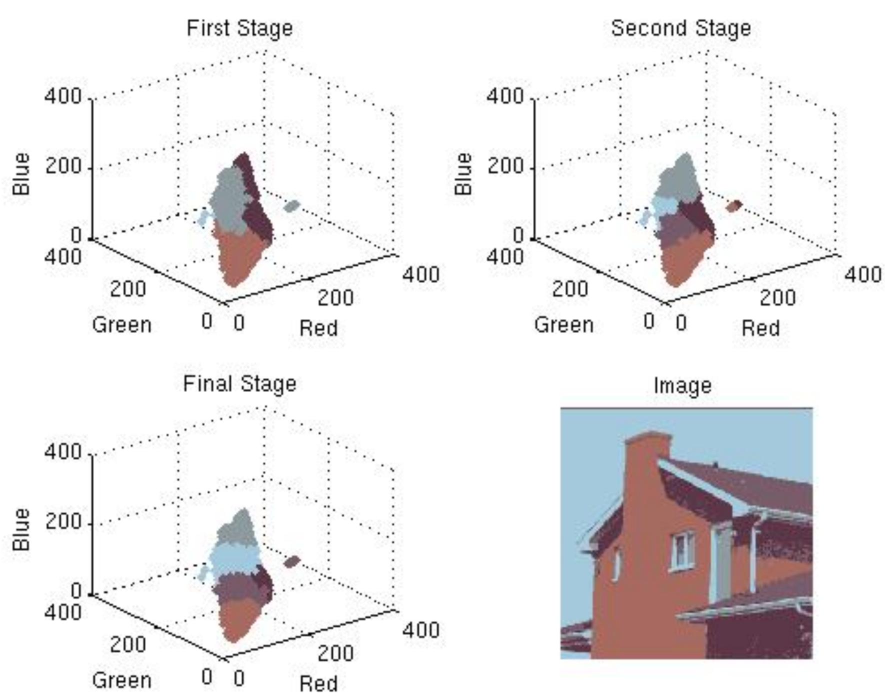


Figure 6: Data Samples in RGB Space with Image

**Run 2**

Table 5: Initial Mean Values

| | | |
|---|---|---|
| [225.7178 | 85.5160 | 166.7081] |
| [232.8881 | 173.3306 | 126.0144] |
| [203.0269 | 34.8211 | 198.6582] |
| [25.1716 | 183.9130 | 182.3345] |
| [66.7772 | 27.2243 | 230.4487] |

Table 6: Final Mean Values

| | | |
|---|---|---|
| [159.2820 | 105.3212 | 98.2036] |
| [163.7582 | 199.5677 | 220.0477] |
| [159.7422 | 0 | 222.8516] |
| [138.0886 | 151.8591 | 157.1567] |
| [96.3836 | 63.3228 | 77.8244] |

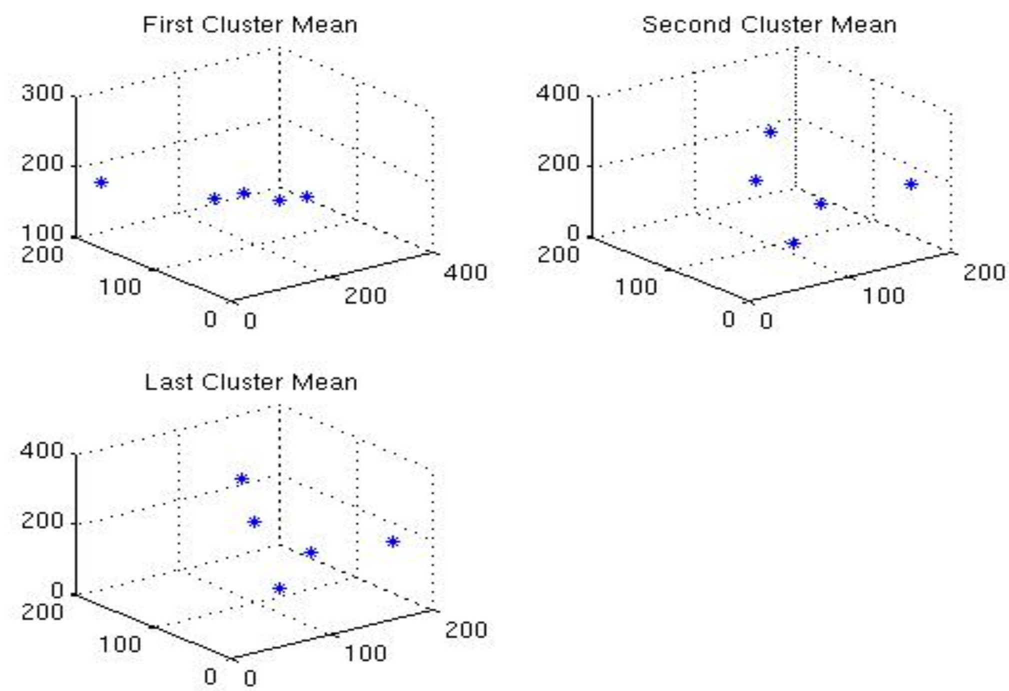


Figure 7: Cluster Mean For 3 Stages

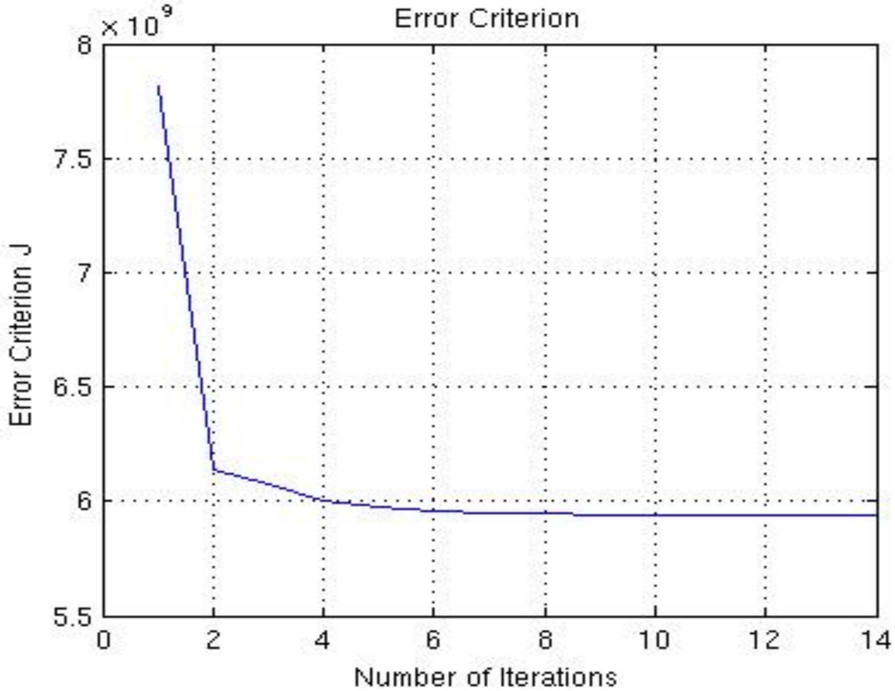


Figure 8: Error Crieterion J

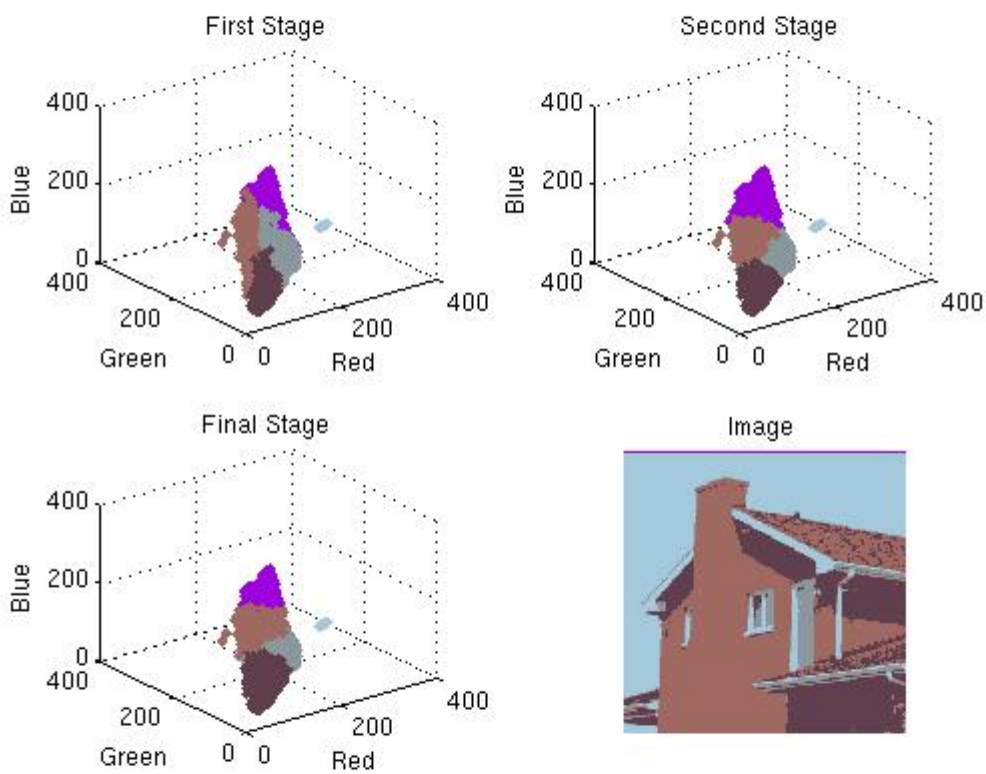


Figure 9: Data Samples in RGB Space with Image

**Part 3: Xie-Beni (XB) Index**

The Xie-Beni index for c=2 and c=5 were found to be as follows:

XB(2) = 0.2014
XB(5) = 0.0832 (First Run)
XB(5) = 0.0697 (Second Run)

Since the initial mean values for $_i$ are constantly changing, the Xie-Beni index will always be changing. As seen by the two runs done while c = 5 it was noticed that the XB values were reasonable and are very similar. By having more clusters it was also confirmed that the XB values were smaller meaning it performed better.

# Conclusion

In conclusion, the K-means algorithm is a great tool for cluster analysis in data mining. It is very accurate in classifying the unlabelled data sets. This can be verified by comparing the original image to the reformed image, looking at the Xie-Beni index or analysing the cluster components in the distribution graph. However the performance needs to be improved. When applied with a large input sample such as 65000 pixels of a house, it takes a large amount of processing power and time. Although no optimization was done, the program still takes a couple of minutes to perform.

# References


[1] N. Zhang, "ELE888/EE8209 { Intelligent Systems (2015) { Student Lab Manual," Department of Electrical and Computer Engineering, Ryerson University, Toronto, Ontario, April 5. 2015.


# Appendix

```
//Kmeans
function kmeans(c)
%Algorithm takes k (c=k) means and classifies data into clusters around
%different mean points
I=imread('house.tiff');
imshow(I);
[M,N,D]=size(I);
X=reshape(I,M*N,3);
x=double(X);
figure;
plot3(x(:,1),x(:,2),x(:,3),'.')
xlim([0 255]);
ylim([0 255]);
zlim([0 255]);
hold on; grid;
title('All pixels in RGB');
xlabel('Red');
ylabel('Green');
zlabel('Blue');

mu=zeros(2,3); %Data has 3 axes: R,G,B
Mus=zeros(2,3);
```

```
Mucount=1;

disp('Initial Mu values');
mu=rand(c,3)*255 %Initialize random numbers between 0 and 255

for i=1:c
    Mus(i,:)=mu(i,:);
    Mucount=Mucount+1;
end

iteration=0;
delta_mu=1;
jpt_count=1;
f=@(a,b) (a-b).^2;
jpt=0;
jps=0;

while(delta_mu>0)
    v=zeros(length(x),2);
    jp=zeros(length(x),2);
    index=0;
    for j=1:c
        v(:,j)=squeeze(sqrt(sum(bsxfun(f,x,mu(j,:)),2)));
        jp(:,j)  = sum( bsxfun(f,x,mu(j,:)) ,2);
        jps(j) = sum(jp(:,j));
    end

    jpt(jpt_count) = sum(jps);
    jpt_count = jpt_count+ 1;
    [minv,index]=min(v,[],2);
    [cluster,count]=MinIndex(x,index);

    if(iteration==0) %Get cluster mean at first stage
        cluster1=cluster;
        count1=count; %Get cluster mean at second stage
    end
    if(iteration==1)
        cluster2=cluster;
        count2=count;
    end
    meanV=zeros(1,3);
    delta_mu=0;

    for i=1:c
        meanV=mean(cluster(1:count(i)-1,(i*3)-2:(i*3)));
        Mus(Mucount,:)=meanV;
        Mucount=Mucount+1;
        delta_mu=delta_mu+abs(mu(i,:)-meanV);
        %delta_mu will remain zero if no mu change from previous iteration
        mu(i,:)=meanV;
    end
    iteration=iteration+1;
end
figure;
plot(1:iteration,jpt); %Plot the error criterion
grid;
xlabel('Number of Iterations');
ylabel('Error Criterion J');
title('Error Criterion');
```

```
%% Find the Xie-Beni Index
XieBeni=0;
for k=1:size(x,1)
    for j=1:c
        if index(k,1)==j
            XieBeni=XieBeni+v(k,j)/min(norm(mu-repmat(mu(c,:),c,1)));
        end
    end
end

disp('Xie-Beni Index');
XieBeni=XieBeni/size(x,1)

%% Plot first stage clusters
figure;
subplot(2,2,1);
colour=zeros(1,3);
for i=1:c
    colour(i,:)= Mus(length(Mus)-i,:)/255;
    plot3(cluster1(1:count1(i)-1,(i*3)-2),cluster1(1:count1(i)-1,(i*3)-
1),cluster1(1:count1(i)-1,(i*3)),'.','Color',colour(i,:));
    hold on;
end
xlabel('Red');
ylabel('Green');
zlabel('Blue');
title('First Stage');
grid;
%% Plot second stage clusters
subplot(2,2,2);
colour=zeros(1,3);
for i=1:c
    colour(i,:)= Mus(length(Mus)-i,:)/255;
    plot3(cluster2(1:count2(i)-1,(i*3)-2),cluster2(1:count2(i)-1,(i*3)-
1),cluster2(1:count2(i)-1,(i*3)),'.','Color',colour(i,:));
    hold on;
end
xlabel('Red');
ylabel('Green');
zlabel('Blue');
title('Second Stage');
grid;
%% Plot final stage clusters
subplot(2,2,3);
colour=zeros(1,3);
for i=1:c
    colour(i,:)= Mus(length(Mus)-i,:)/255;
    plot3(cluster(1:count(i)-1,(i*3)-2),cluster(1:count(i)-1,(i*3)-
1),cluster(1:count(i)-1,(i*3)),'.','Color',colour(i,:));
    hold on;
end
xlabel('Red');
ylabel('Green');
zlabel('Blue');
title('Final Stage');
grid;

%% Display the image w/ dominant colours
```

```
for i=1:length(index)
    image(i,:)=mu(index(i),:);
end

Ilabeled=reshape(image,M,N,3);
subplot(2,2,4);
imshow(uint8(Ilabeled));
title('Image');

%% Plot the cluster means

figure;
subplot(2,2,3)
plot3(Mus(length(Mus)-c:length(Mus),1),Mus(length(Mus)-
c:length(Mus),2),Mus(length(Mus)-c:length(Mus),3),'*','Color',[ 0 0 1 ])
title('Last Cluster Mean');
grid;

subplot(2,2,1)
plot3(Mus(1:c,1),Mus(1:c,2),Mus(1:c,3),'*','Color',[ 0 0 1 ])
title('First Cluster Mean');
grid;

subplot(2,2,2)
plot3(Mus(c+1:c*2,1),Mus(c+1:c*2,2),Mus(c+1:c*2,3),'*','Color',[ 0 0 1 ])
title('Second Cluster Mean');
grid;

%% Display any data

disp('Final Mean values');
mu
end
```

//MinIndex

```
function [group,counter] = MinIndex(x,index)
   group= zeros(2,3);

   for i=1:max(index)

        counter(i) = 1;
   end
   for w=1:length(x)

       k = abs(((index(w))*3)-2);
       j = (index(w))*3;
       group(counter(index(w)),k:j) = x(w,:);
       counter(index(w)) = counter(index(w)) + 1;

   end

end
```