

report

Problem 1

Run mean_shift_cluster.m

Problem 2

Run tractor_segmentation.m

2mean-shift

1. Static the color histogram
2. Calculate norm of every two picture
3. Set a radix threshold. Calculate the distance of the histograms between them under the threshold
4. Calculate the vector of each point to center. And add them all.
5. Move the center to this vector
6. Record the new center. And check if it can converge
7. If two centers are closed enough, then, merge them to one cluster

3Tractor Segmentation

1. First, deal with the tractor. Segment tractor with photoshop.
2. Set the rest of the picture to color white
3. Use k=5 kmeans method to train the picture
4. If the color of the pixel is white, use first kmeans method to train
5. Otherwise, use second kmeans method to train
6. At last, we can get 4 cluster center from training phase
7. Scan the new picture, we can find the closest cluster of the pixel.
8. Set the pixels who are not close to tractor clusters white.

4super param

Problem 1

Red Is the final param

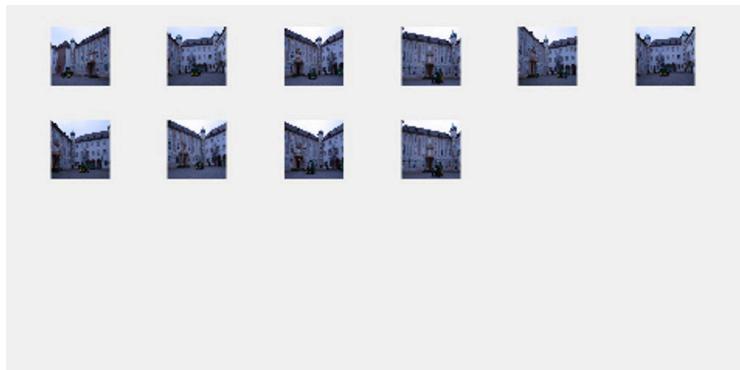
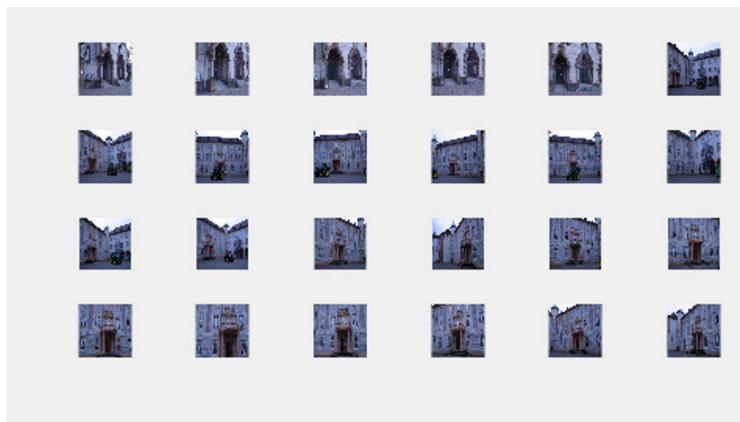
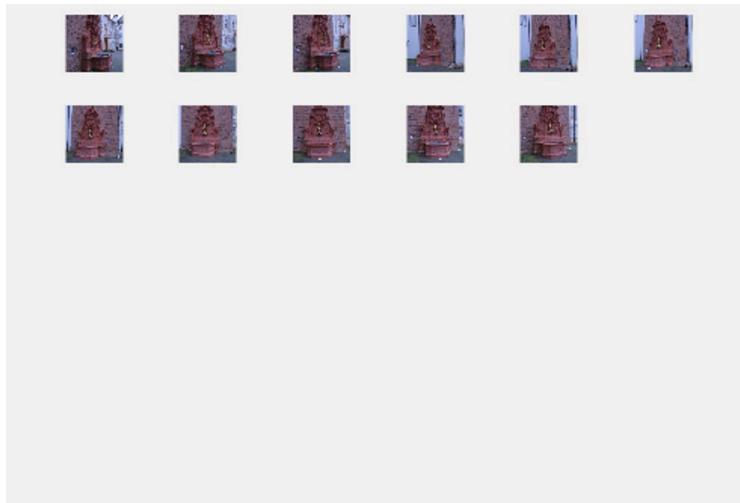
- picture size
 - 256*256
- color bin
 - 2
 - 4
 - 8
 - 16
- radix
 - 10000
 - 5000
 - 10000
 - 12000 some of them are not included.
 - 15000 seems great
 - 20000 too big
- converge threshold
 - 10
 - 100 not works too much
- Merge threshold
 - 10
 - 100
 - 10000 not works too much

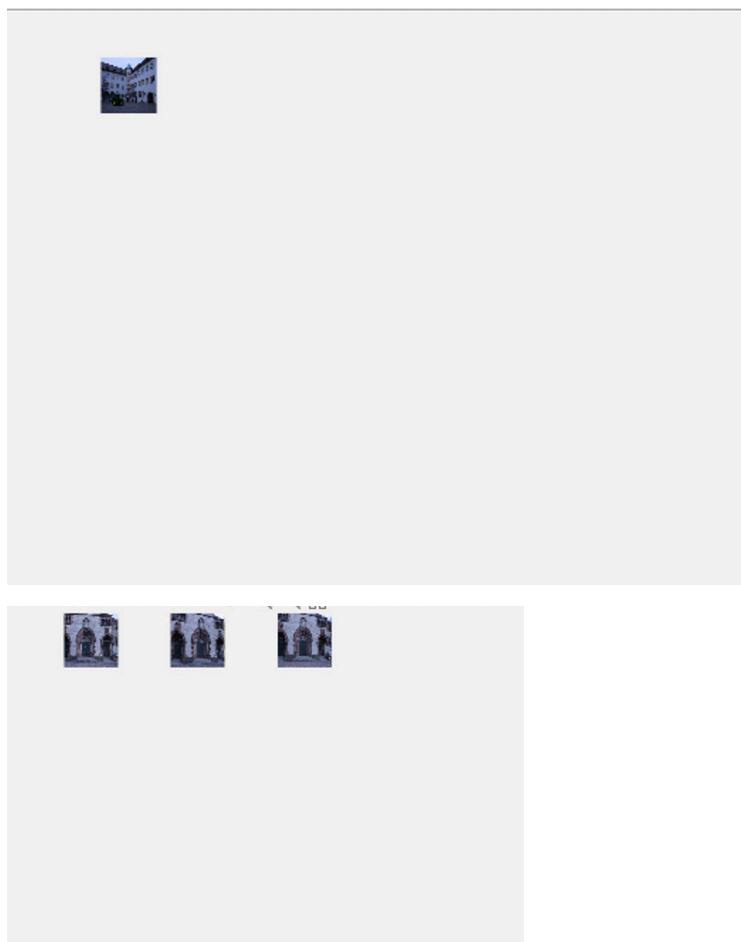
Problem 2

- picture size
 - 256*256
- Kmeans centers
 - 4 works great
- Kmeans iters
 - 4

5test result

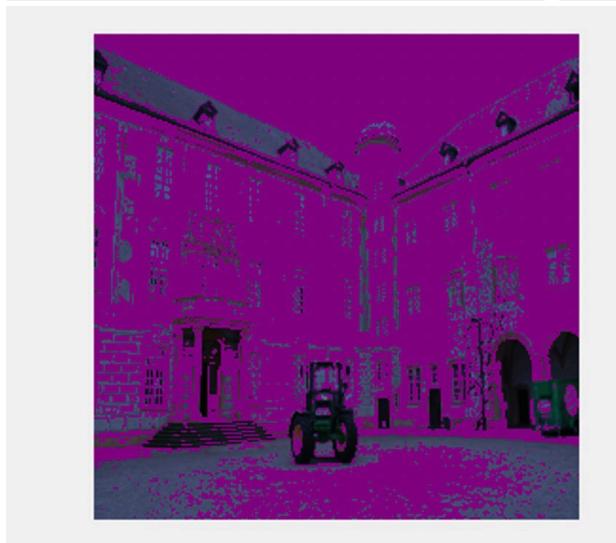
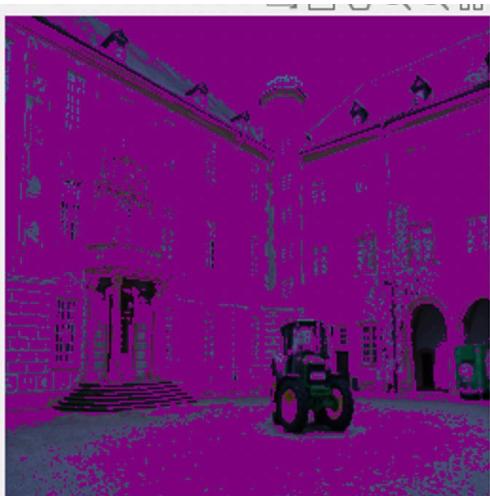
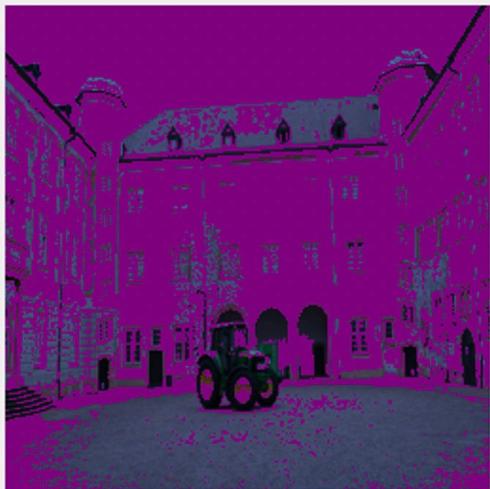
Problem 1





Problem 2





```

1 % read images
2 Files = dir(fullfile('pic2\*\*.jpg'));
3 LengthFiles = length(Files);
4
5 len = 256;
6 wid = 256;
7 Imgs = cell(1,LengthFiles);
8
9 bin = 8;
10 RGB_unit = 256 / bin;
11
12 % color histogram
13 X = zeros(bin,bin,bin,LengthFiles);
14
15 for i = 1:LengthFiles
16     Img = imread(strcat('pic2\\\',Files(i).name));
17     Img = imresize(Img,[len,wid]);
18     Imgs{i}=Img;
19     for j = 1:len
20         for k = 1:wid
21             R = fix(single(Img(j,k,1))/single(RGB_unit))+1;
22             G = fix(single(Img(j,k,2))/single(RGB_unit))+1;
23             B = fix(single(Img(j,k,3))/single(RGB_unit))+1;
24             temp = X(R, G, B, i);
25             X(R, G, B, i) = temp+1;
26         end
27     end
28 end
29
30 [clus] = mean_shift(X);
31 parents = unique(clus);
32 for i = 1:size(parents)
33     figure(i);
34     count = 1;
35     for j = 1:size(clus)
36         if clus(j) == parents(i)
37             im = Imgs{j};
38             subplot(6,6,count),imshow(im);
39             count = count + 1;
40         end
41     end
42 end
43
44
1 function [clusters, centers] = kmeans_code(X, k, iter)
2 X = single(X);
3 len = size(X, 1);
4 dim = size(X, 2);
5 points = rand(k, dim);
6 points = single(points);
7
8 % 1. sample k number
9 samples = randsample(len,k);
10
11 % 2. ensure centers are existed points
12 for i = 1:k
13     points(i, :) = X(samples(i), :);
14 end
15
16 % 3. iterate
17 for iter = 1:iters
18     % vector with label
19     XLabel = [X ones(len, 1)];
20     % 4. scan each vector
21     for i = 1:size(XLabel, 1)
22         minDist = norm(XLabel(i, 1:dim).'- points(1, :).');
23         minJ = 1;
24         for j = 1:size(points, 1)
25             dist = norm(XLabel(i, 1:dim).'- points(j, :).');
26             if dist <= minDist
27                 minJ = j;
28                 minDist = dist;
29             end
30         end
31         XLabel(i, dim + 1) = minJ;
32     end
33
34     % 5. group by point index
35     for i = 1:k
36         set(:, :, i) = {XLabel(XLabel(:, dim + 1) == i, 1:dim)};
37     end
38
39     % 6. renew points
40     for i = 1:k
41         avg = mean(set{i}, 1);
42         points(i, :) = avg(1:dim);

```

```

function [clus] = mean_shift(X)
% init
nums = size(X, 4);
dim = size(X, 2);
radix = 15000;
clusters = [];
centers = zeros(dim,dim,dim,nums);
end_threshold = 100;
close_threshold = 10000;
belong_to = zeros(nums,1);

% deal each picture
for i = 1:nums
    if belong_to(i)>0
        continue
    end
    converged = false;
    center = X(:,:,:,:i);
    while ~converged
        dis = zeros(nums,1);
        vec = zeros(dim, 1);
        % caculate each distance of other picture
        for j = 1:nums
            flat_x = reshape(X(:,:,:,:j),[],1).';
            flat_c = reshape(center,[],1).';
            dis(j) = norm(flat_x-flat_c);
            if dis(j) < radix
                % update shift vector
                vec = center - X(:,:,:,:j);
                % update cluster index
                belong_to(j) = i;
            end
        end
        % check if converged
        center = center + vec;
        flat_v = reshape(vec,[],1).';
        if norm(flat_v) < end_threshold
            converged = true;
        end
        centers(:,:,:,:i)=center(:,:,:,:);
    end
end

%merge the close cluster
for i=1:nums
    flat_center_i = reshape(centers(:,:,:,:belong_to(i)),[],1).';
    for j=i+1:nums
        if belong_to(i)==belong_to(j)
            continue
        end
        bt = belong_to(j);
        flat_center_j = reshape(centers(:,:,:,:bt),[],1).';
        if norm(flat_center_i-flat_center_j)<close_threshold
            belong_to(j)=i;
        end
    end
end
clus = belong_to;
end

```

```

1 % 1. deal
2 len = 256;
3 wid = 256;
4 castle = imread('pic1\castle-P19 (8).jpg');
5 castle = imresize(castle,[len,wid]);
6 tractor = imread('pic1\tractor.jpg');
7 tractor = imresize(tractor,[len,wid]);
8
9 %imshow(skyTrain)
0 %imshow(skyTrainNoSky)
1 clatle_bg = [];
2 tra_fg = [];
3 white = [255 255 255];
4 skyIndex = 1;
5 noSkyIndex = 1;
6
7 % 2. train
8 for i = 1:size(castle,1)
9     for j = 1:size(castle,2)
0
1         if all(tractor(i,j,:)==white)
2             clatle_bg(skyIndex,:,:)=castle(i,j,:);
3             skyIndex = skyIndex + 1;
4         else
5             tra_fg(noSkyIndex,:,:)=castle(i,j,:);
6             noSkyIndex = noSkyIndex + 1;
7         end
8     end
9 end
0
1 % 3. kmeans
2 [cluster1, center1] = kmeans_code(clatle_bg, 4, 4);
3 [cluster2, center2] = kmeans_code(tra_fg, 4, 4);
4
5 % 4. classification
6 for index = 1:5
7     img = imread(fullfile('pic1\', strcat('castle-P19 (' , num2str(index), ') .jpg')));
8     img = imresize(img,[len,wid]);
9     for i = 1:size(img,1)
0         for j = 1:size(img,2)
1             dis1=99999;
2             dis2=99999;
3
4                 for k=1:size(center1)
5                     dis1 = min(norm(double(img(i,j)).' - center1(k).'), dis1);
6                 end
4
7                 for k=1:size(center1)
8                     dis2 = min(norm(double(img(i,j)).' - center2(k).'), dis2);
9                 end
4
5                 if dis1 < dis2
6                     img(i,j,:)=[128 0 128];
7                 end
8             end
9         end
5
6         % 5. paint
55         figure(index)
56         imshow(img)
57     end
58

```