MATH444 HW7

Jiasen Zhang

1 Problem 1

The given compressed image is shown in Figure 1. Since the given data only takes about 0.15% of the full image data, the given image is almost empty only with some sparse points in the image. That means we can't recover all the information of the full image.

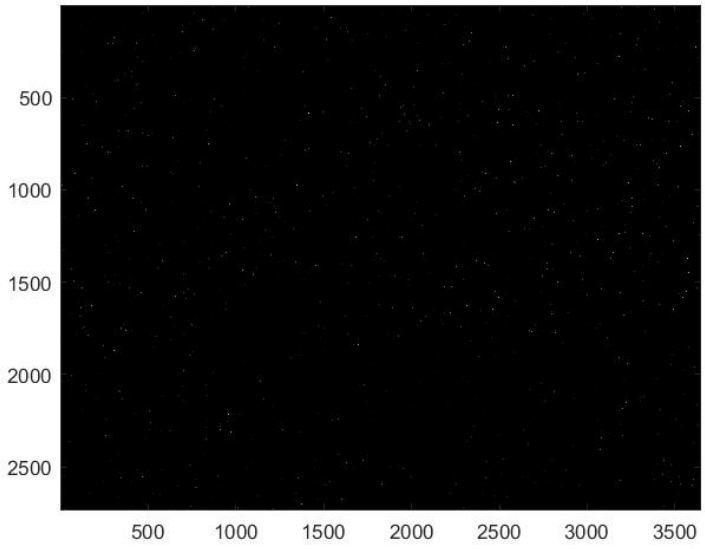

Figure 1: The given image.

Using tree regression to reconstruct the image, the best result is shown in Figure 2, in which all the leaves are pure. In Figure 2 we can see a white house and a white tower near the sea. We can recognize the blue sky with clouds, the blue sea and green trees.

Figure 2: The approximate image.

Figure 3 shows different stages of reconstruction.

Figure 3

2 Matlab Code

```
clear all;clc;
tic

load MysteryImage.mat;

% show the given data as an image
I0=zeros(m,n,3);
for i=1:15000
    I0(rows(i),cols(i),1)=vals(i,1);
    I0(rows(i),cols(i),2)=vals(i,2);
    I0(rows(i),cols(i),3)=vals(i,3);
end
%imagesc(I0);

row0 = 1:m;
col0 = 1:n;
y = zeros(m,n,3); % the approximated image


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initialize the tree
% structure R
R(1).x = vals;
R(1).row = rows; % coordinate of sampled pixels
R(1).col = cols; % coordinate of sampled pixels
R(1).row0 = row0; % coordinate of image
R(1).col0 = col0; % coordinate of image
R(1).split_idx = [];
R(1).leftR = []; % left child
R(1).rightR = []; % right child
m=1;

depth=16;

figure(1);
for k=1:depth

for i=1:m
    % if it's a mixed leaf
    if isempty(R(i).leftR) && length(R(i).row)>1
        s = find_s(R(i).row,R(i).x);
        R(i).leftrow = R(i).row(R(i).row≤s);
        R(i).rightrow = R(i).row(R(i).row>s);
        R(i).leftcol = R(i).col(R(i).row≤s);
        R(i).rightcol = R(i).col(R(i).row>s);
        xleft=R(i).x(R(i).row≤s,:);
        xright=R(i).x(R(i).row>s,:);
        %
        c1 = mean(R(i).x(R(i).row≤s,:),1);
        c2 = mean(R(i).x(R(i).row≥s,:),1);
        %
        for j=1:3
            y(min(R(i).row0):floor(s),R(i).col0,j)=c1(j);
            y(ceil(s):max(R(i).row0),R(i).col0,j)=c2(j);
        end

        R(i).split_idx = s;

        if k<depth
            R(i).leftR = m+1;
            R(i).rightR = m+2;
            R(m+1).row = R(i).leftrow;
            R(m+1).col = R(i).leftcol;
            R(m+2).row = R(i).rightrow;
```

```
            R(m+2).col = R(i).rightcol;
            R(m+1).x = xleft;
            R(m+2).x = xright;

            R(m+1).row0 = (min(R(i).row0):floor(s));
            R(m+1).col0 = R(i).col0;
            R(m+2).row0 = (ceil(s):max(R(i).row0));
            R(m+2).col0 = R(i).col0;

            m=m+2;
        end
    end
end
imagesc(y);drawnow;

for i=1:m
    % if it's a mixed leaf
    if isempty(R(i).leftR) && length(R(i).col)>1
        s = find_s(R(i).col,R(i).x);
        R(i).leftcol = R(i).col(R(i).col<=s);
        R(i).rightcol = R(i).col(R(i).col>s);
        R(i).leftrow = R(i).row(R(i).col<=s);
        R(i).rightrow = R(i).row(R(i).col>s);
        xleft=R(i).x(R(i).col<=s,:);
        xright=R(i).x(R(i).col>s,:);
        %
        c1 = mean(R(i).x(R(i).col<=s,:),1);
        c2 = mean(R(i).x(R(i).col>=s,:),1);
        %
        for j=1:3
            y(R(i).row0,min(R(i).col0):floor(s),j)=c1(j);
            y(R(i).row0,ceil(s):max(R(i).col0),j)=c2(j);
        end

        R(i).split_idx = s;

        if k<depth
            R(i).leftR = m+1;
            R(i).rightR = m+2;
            R(m+1).row = R(i).leftrow;
            R(m+1).col = R(i).leftcol;
            R(m+2).row = R(i).rightrow;
            R(m+2).col = R(i).rightcol;
            R(m+1).x = xleft;
            R(m+2).x = xright;

            R(m+1).row0 = R(i).row0;
            R(m+1).col0 = (min(R(i).col0):floor(s));
            R(m+2).row0 = R(i).row0;
            R(m+2).col0 = (ceil(s):max(R(i).col0));

            m=m+2;
        end
    end
end
imagesc(y);drawnow;

fprintf('depth = %d \t m = %d\n',k,m);
end


toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% function for finding s
function s = find_s(t,x)
% t: coordinate of x
```

```
% x: data value
% s: split value
p=length(t);
[t, idx] = sort(t); % sort t and x
x(:,:) = x(idx,:);
sigma=zeros(1,p-1);
F=zeros(1,p-1);
for j=1:p-1
    sigma(j)=(t(j)+t(j+1))/2;
    idx1=(t≤sigma(j));
    idx2=(t>sigma(j));
    c1 = mean(x(idx1,:),1);
    c2 = mean(x(idx2,:),1);
    FL = sum(sum((x(idx1,:)-c1).^2)); % sum of square of 2-norm
    FR = sum(sum((x(idx2,:)-c2).^2)); % sum of square of 2-norm
    F(j)=FL+FR; % MSE
end
s=sigma(F==min(F));
s = (s(1)+s(end))/2;
end
```