Programming Assignment 9

NikhilTank: 13110070

1. Given a set of multi-exposure images of a static scene captured using a static camera, design an approach to generate HDR image of the scene and estimate the camera response function. Tone map the HDR image using bilateral filter for display.

Input images:

Into the Open (HDR) - ppw - 01.tif
> Shutter speed: 30s

Into the Open (HDR) - ppw - 02.tif
> Shutter speed: 15s

Into the Open (HDR) - ppw - 03.tif
> Shutter speed: 8s

Into the Open (HDR) - ppw - 04.tif
> Shutter speed: 4s

Into the Open (HDR) - ppw - 05.tif
> Shutter speed: 2s

Into the Open (HDR) - ppw - 06.tif
> Shutter speed: 1s

Into the Open (HDR) - ppw - 07.tif
> Shutter speed: 1/2s

Into the Open (HDR) - ppw - 08.tif
> Shutter speed: 1/4s

Into the Open (HDR) - ppw - 09.tif
> Shutter speed: 1/8s

Into the Open (HDR) - ppw - 10.tif
> Shutter speed: 1/15s

Into the Open (HDR) - ppw - 11.tif
> Shutter speed: 1/30s

Into the Open (HDR) - ppw - 12.tif
> Shutter speed: 1/60s

%% Matlab Code

```
clear all
clc
img1 = imread('ppw - 01.tif');
img1 = imresize(img1, 0.05);
img2 = imread('ppw - 02.tif');
img2 = imresize(img2, 0.05);
img3 = imread('ppw - 03.tif');
img3 = imresize(img3, 0.05);
img4 = imread('ppw - 04.tif');
img4 = imresize(img4, 0.05);
img5 = imread('ppw - 05.tif');
img5 = imresize(img5, 0.05);
img6 = imread('ppw - 06.tif');
img6 = imresize(img6, 0.05);
```

```
img7 = imread('ppw - 07.tif');
img7 = imresize(img7, 0.05);
img8 = imread('ppw - 08.tif');
img8 = imresize(img8, 0.05);
img9 = imread('ppw - 09.tif');
img9 = imresize(img9, 0.05);
img10 = imread('ppw - 10.tif');
img10 = imresize(img10, 0.05);
img11 = imread('ppw - 11.tif');
img11 = imresize(img11, 0.05);
img12 = imread('ppw - 12.tif');
img12 = imresize(img12, 0.05);
[x, y, c] = size(img1);

archive = {img1,img2,img3,img4,img5,img6,img7,img8,img9,img10,img11,img12};

B = [log(30); log(15);log(8);log(4);log(2);log(1);log(1/2);log(1/4);log(1/8);log(1/15);log(1/30);log(1/60)];


zmin = 0;
zmax = 255;

%% generating weighting function
w = zeros(256,1);
for i = 0:254
    if i <= 1/2*(zmin+zmax)
      w(i+1) = i - zmin;
    elseif i > 1/2*(zmin+zmax)
      w(i+1) = zmax - i;
    end
end

%% generating response funtion
l=5;
inew=zeros(1000,12);
for z =1:c
    for p =1:12
        iarc = archive{p}(:,:,z);
        iarc = iarc(:);
        inew(:,p) = iarc(1:1000);
    end
    g = gsolve(inew,B,l,w);
    G(:,z) = g;
end

%% getting HDR image
R=zeros(x,y,c);
D=zeros(x,y,c);
for z = 1:c
    for i = 1:x
        for j = 1:y
            for p =1:12
```

```
            R(i,j,z) = w(archive{p}(i,j,z)+1)*(G(archive{p}(i,j,z)+1,z)-B(p))+ R(i,j,z);
            D(i,j,z)=  w(archive{p}(i,j,z)+1) +  D(i,j,z);
        end
        R(i,j,z) = R(i,j,z)/D(i,j,z);
    end
  end
end
imwrite(R,'HDR.jpg');
```

OUTPUT

```
%% generating irradiance values
R = abs(R);
Inten = zeros(x,y);
for i = 1:x
   for j = 1:y
      Inten(i,j) = 1/61*(R(i,j,1)*20 + R(i,j,2)*40 + R(i,j,3));
   end
end

R_new = zeros(x,y,c);
L = zeros(x,y,c);
for z = 1:c
   for i = 1:x
      for j = 1:y
         R_new(i,j,z) = R(i,j,z)/Inten(i,j);
         L(i,j,z) = log2(Inten(i,j));

      end
   end
end

for k = 1:z
   for i = 1:x
      for j = 1:y
         if L(i,j,k)>1
            L(i,j,k)=1;
         end
```

```
            if L(i,j,k)<0
                L(i,j,k)=0;
            end
        end
    end
end

B = bfilter2(L_1, 5, [3, 0.1]);
B = im2uint8(B);
L = im2uint8(L);
D = L - B;
o = 1; %offset
s = 2; %scale

for z = 1:c
    for i = 1:x
        for j = 1:y
        B_1(i,j,z) = (B(i,j,z) - o)*s;
        end
    end
end

O = exp(B_1 + D);
tone = zeros(x,y,c);
for z = 1:c
    for i = 1:x
        for j = 1:y
        tone(i,j,z) = O(i,j,z)*R(i,j,z)/Inten(i,j);
        end
    end
end
 imread(tone, 'tone_HDR.jpg');
```

Reference:

[1] http://cs.brown.edu/courses/cs129/results/proj5/njooma/

[2] http://farbspiel-photo.com/learn/hdr-pics-to-play-with/hdr-pics-to-play-with-into-the-open