Contents


```
clear all;
close all;
```

Question 1. Part A

```
% The overall contrast of the system, from capture to display, is
% reduced because the overall luminance is raised to the power of 2/3
% (which is less than 1). Therefore at the output of the system we have an
% effective luminance of 1000^(2/3) = 100. Therefore, a display with a
% minimum contrast of 100:1 is enough.

contrast_display = 1000^(2/3);
display(['The contrast of the display must be ' num2str(contrast_display) ':1']);
```

```
The contrast of the display must be 100:1
```

Question 1. Part B

```
hdr_memorial = hdrread('hw1_data/hw1_memorial.hdr');
hdr_atrium = hdrread('hw1_data/hw1_atrium.hdr');

gray_memorial = rgb2gray(hdr_memorial);
gray_atrium = rgb2gray(hdr_atrium);

figure; imshow(gray_memorial);
figure; imshow(gray_atrium);

% We can see that certain portions of the images are either completely
% saturated or completely black. It is hard to see any detail in these
% portions of the images. The parts of the images that are easy to see are
% the parts that are somewhere in the middle of the contrast range, that
% havn't been clipped by the contrast of the display.
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```

Question 1. Part C

```
gamma_memorial = 0.4;
gamma_atrium = 0.5;

gray_memorial_gamma = gray_memorial .^ gamma_memorial;
gray_atrium_gamma = gray_atrium .^ gamma_atrium;

figure; imshow(gray_memorial_gamma);
figure; imshow(gray_atrium_gamma);
```

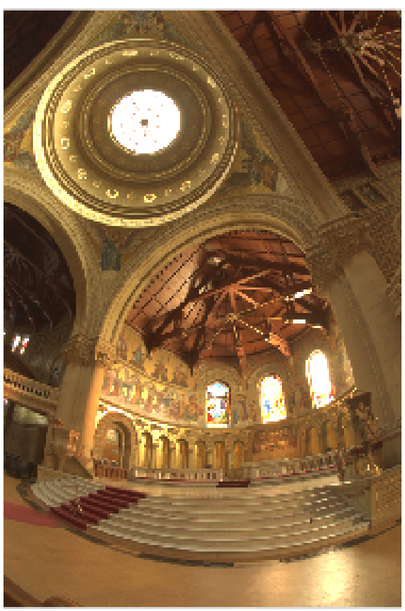
```
display(['The gamma for the memorial scene is : ' num2str(gamma_memorial)]);
display(['The gamma for the atrium scene is : ' num2str(gamma_atrium)]);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
The gamma for the memorial scene is : 0.4
The gamma for the atrium scene is : 0.5
```

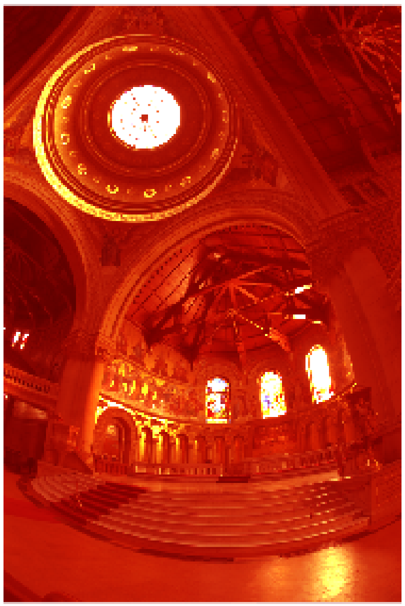
Question 1. Part D

```
% Apply same gamma correction to each channel
for i = 1:3
   hdr_memorial_gamma(:,:,i) = hdr_memorial(:,:,i) .^ gamma_memorial;
   hdr_atrium_gamma(:,:,i) = hdr_atrium(:,:,i) .^ gamma_atrium;
end

figure;
subplot(1,2,1); imshow(hdr_memorial_gamma);
```

```
subplot(1,2,2); imshow(hdr_atrium_gamma);
title('Same gamma correction to each color channel');

% Apply different gamma to different channels
gamma_memorial = [0.2, 1.0, 1.0];
gamma_atrium = [1.0, 0.2, 1.0];

for i = 1:3
   hdr_memorial_gamma(:,:,i) = hdr_memorial(:,:,i) .^ gamma_memorial(i);
   hdr_atrium_gamma(:,:,i) = hdr_atrium(:,:,i) .^ gamma_atrium(i);
end

figure;
subplot(1,2,1); imshow(hdr_memorial_gamma);
subplot(1,2,2); imshow(hdr_atrium_gamma);
title('Different gamma correction for each color channel');

% When you apply different gamma corrections to different channels you
% essentially squeeze the luminances of the different color channels
% differently. The end result is that you get mis colored images. As you
% can see in the images below, when we reduce the contrast of the red
% channel significantly the image becomes more red. The reason for this is
% that large red pixel values get squeezed down to roughly the same values
% as small ones, and when the color channels are normalized and combined to
% create the full color image, there will be much more "high" value red
% pixels on the scale of 0 to 1, thus making the images red.
```



Same gamma correction to each color channel



Different gamma correction for each color channel

Published with MATLAB® R2014b

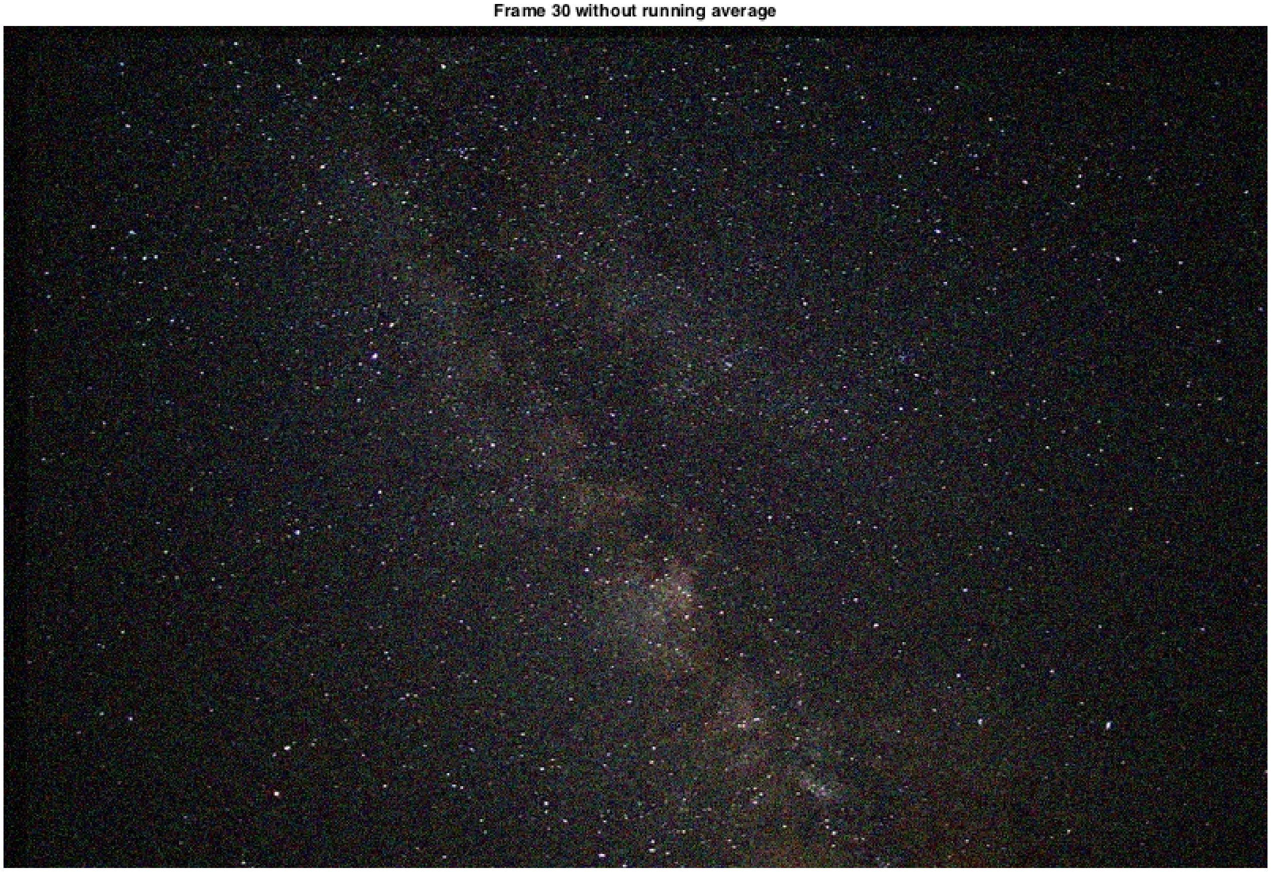
Contents

Question 2. Part A

```
clear all;
close all;

vidSky1 = VideoReader('hw1_data/hw1_sky_1.avi');
vidSky2 = VideoReader('hw1_data/hw1_sky_2.avi');

numFrames1 = get(vidSky1, 'NumberOfFrames');
numFrames2 = get(vidSky2, 'NumberOfFrames');

% Read in 1st frame
fAvg1(:,:,:,1) = im2double(read(vidSky1, 1));
fAvg2(:,:,:,1) = im2double(read(vidSky2, 1));

for i = 2 : numFrames1
   frame1 = im2double(read(vidSky1, i));
   fAvg1(:,:,:,i) = (i-1)/i * fAvg1(:,:,:,i-1) + 1/i * frame1;

   frame2 = im2double(read(vidSky2, i));
   fAvg2(:,:,:,i) = (i-1)/i * fAvg2(:,:,:,i-1) + 1/i * frame2;

   if( i == 30)
       frame1_30 = frame1;
       frame2_30 = frame2;
   end
end

figure; imshow(fAvg1(:,:,:,30));
title('Frame 30 with running average');
figure; imshow(frame1_30);
title('Frame 30 without running average');

figure; imshow(fAvg2(:,:,:,30));
title('Frame 30 with running average');
figure; imshow(frame2_30);
title('Frame 30 without running average');

% From the images below we can see that computing the running average of
% the frames significantly reduces the noise in the video. This is most
% noticible in sky_2, the video with the moon, where in the un-processed
% video we see a lot of noise surrounding the moon. In the processed video,
% we see that all of that nosie is removed. This is less obvious in sky_1,
% the video of the stars, because the noise was blended into the stars.

% However, we can see that because we did not align the images we
% introduced a significant amount of blurring into the video. This is very
% obvious in sky_2, where the edges of the moon are blurry, as well as in
% sky_1 where we see "shooting stars" almost.
```

Frame 30 with running average

Frame 30 without running average



Frame 30 with running average

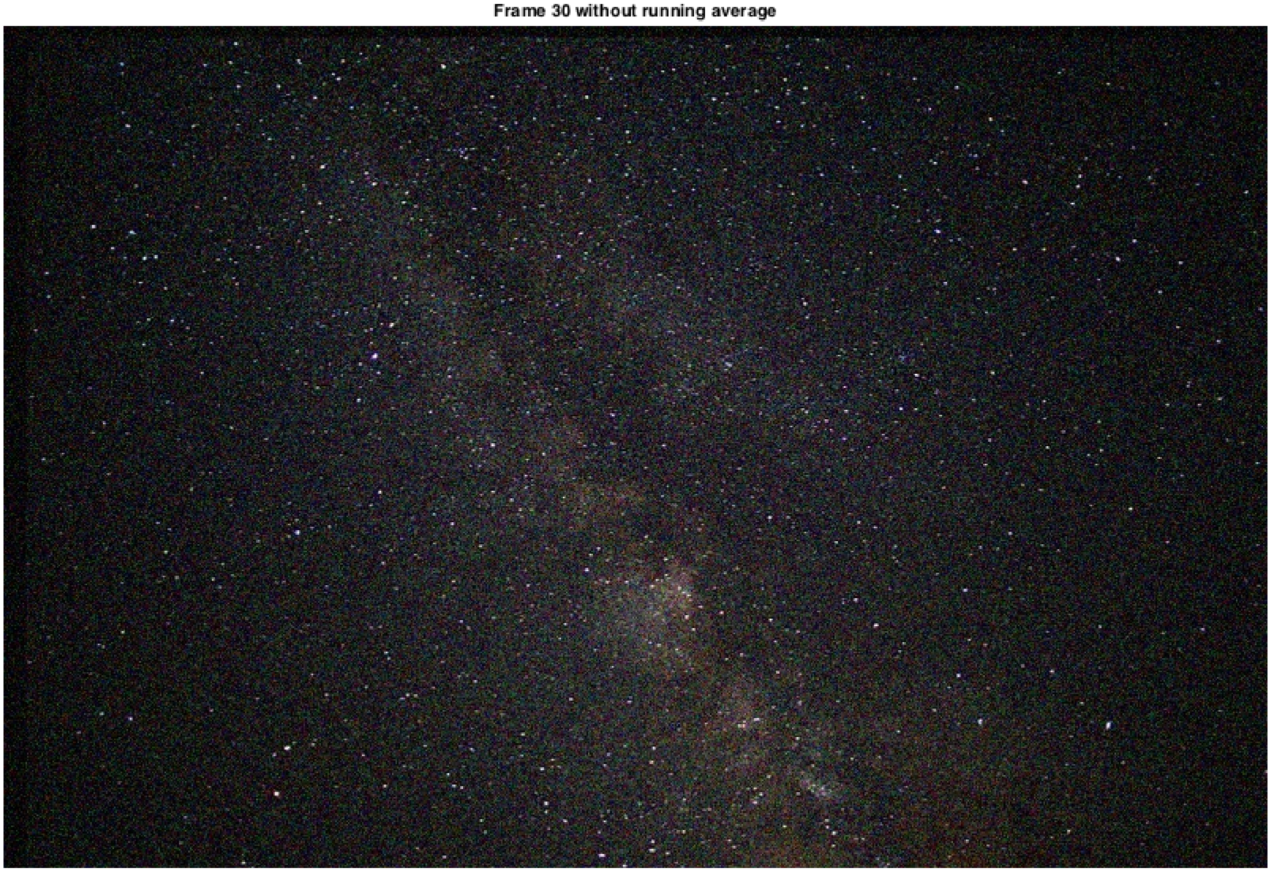
Frame 30 without running average

Question 2. Part B

```
% Read in 1st frame
fAvg_align1(:,:,:,1) = im2double(read(vidSky1, 1));
fAvg_align2(:,:,:,1) = im2double(read(vidSky2, 1));

for i = 2 : 30
   frame1 = im2double(read(vidSky1, i));
   fAvg_align1(:,:,:,i) = (i-1)/i * fAvg_align1(:,:,:,i-1) + 1/i * Align(frame1, fAvg_align1(:,:,:,i-1));

   frame2 = im2double(read(vidSky2, i));
   fAvg_align2(:,:,:,i) = (i-1)/i * fAvg_align2(:,:,:,i-1) + 1/i * Align(frame2, fAvg_align2(:,:,:,i-1));

   if( i == 30)
       frame1_30 = frame1;
       frame2_30 = frame2;
   end
end

figure; imshow(fAvg_align1(:,:,:,30));
title('Frame 30 with running average and alignment');
figure; imshow(frame1_30);
title('Frame 30 without running average');

figure; imshow(fAvg_align2(:,:,:,30));
title('Frame 30 with running average and alignment');
figure; imshow(frame2_30);
title('Frame 30 without running average');

% We can see that the sharpness of the image is preserved, while the noise
% is effectively removed. The alignment tracks the movement of the camera
% and removes the bluriness.
```

Frame 30 with running average and alignment

Frame 30 without running average



Frame 30 with running average and alignment

Frame 30 without running average

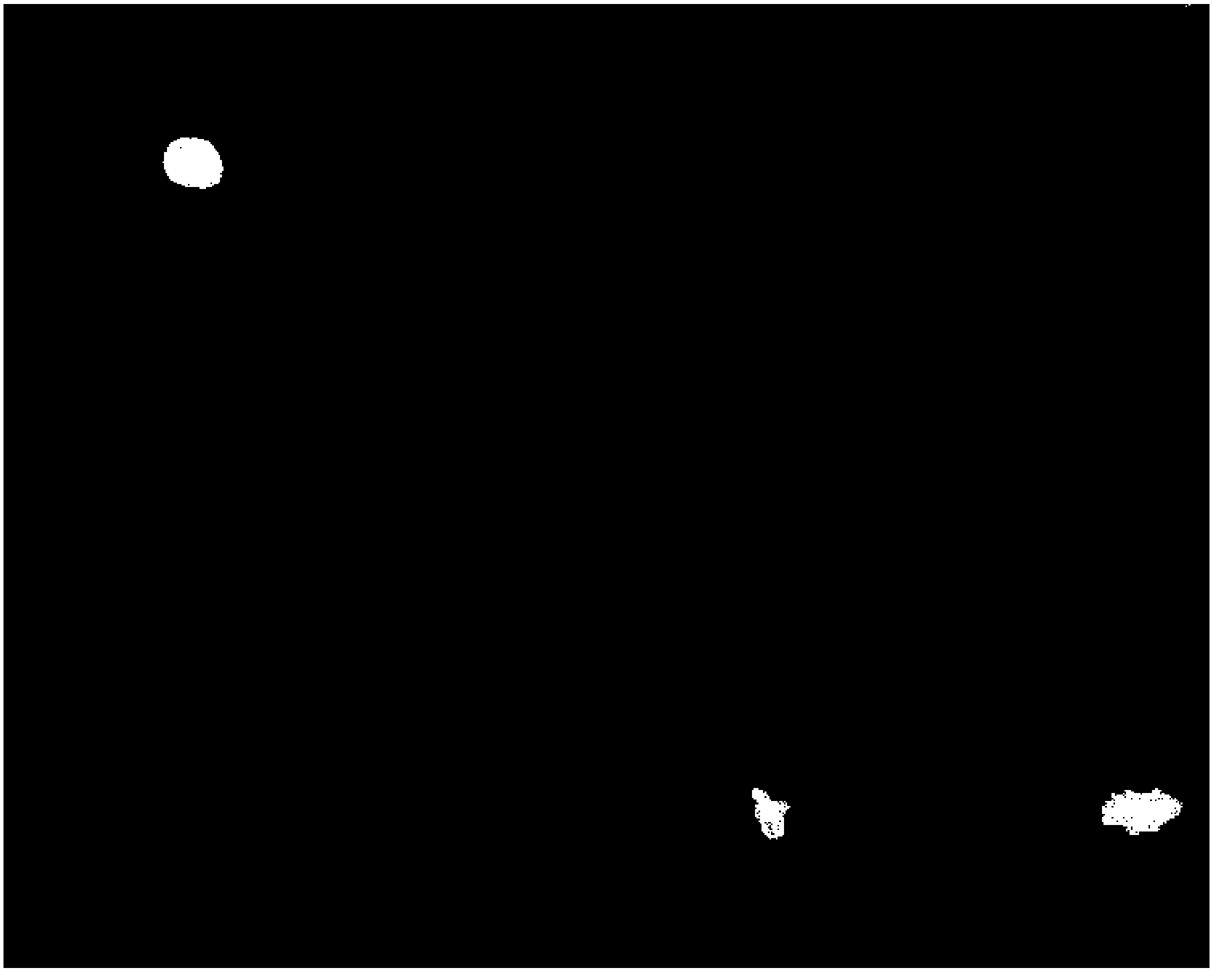
Published with MATLAB® R2014b

Question 3

```
clear all;
close all;

paint1_ref = im2double(imread('hw1_data/hw1_painting_1_reference.jpg'));
paint1_tamp = im2double(imread('hw1_data/hw1_painting_1_tampered.jpg'));
paint2_ref = im2double(imread('hw1_data/hw1_painting_2_reference.jpg'));
paint2_tamp = im2double(imread('hw1_data/hw1_painting_2_tampered.jpg'));

threshold = 0.02;

% Painting 1
[height, width] = size(paint1_ref);

paint1_tamp_aligned = Align(paint1_tamp, paint1_ref);
paint1_diff = abs(paint1_ref - paint1_tamp_aligned);
binary = rgb2gray(paint1_diff);
binary = binary > threshold;

figure;
imshow(binary);
title('Binary difference for painting 1');

% Painting 2
[height, width] = size(paint2_ref);

paint2_tamp_aligned = Align(paint2_tamp, paint2_ref);
paint2_diff = abs(paint2_ref - paint2_tamp_aligned);
binary = rgb2gray(paint2_diff);
binary = binary > threshold;

figure;
imshow(binary);
title('Binary difference for painting 2');
```

Binary difference for painting 1

Binary difference for painting 2

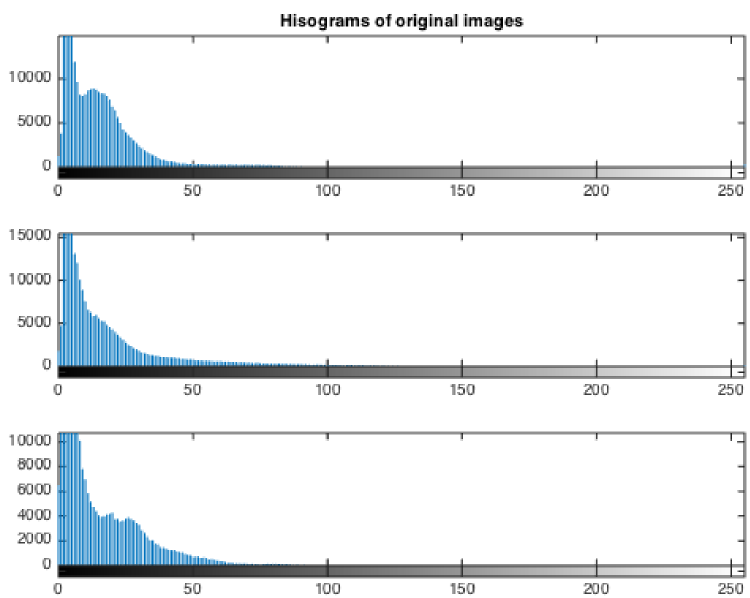


Published with MATLAB® R2014b

Contents

Question 4. Part A

```
clear all;
close all;

road1 = imread('hw1_data/hw1_dark_road_1.jpg');
road2 = imread('hw1_data/hw1_dark_road_2.jpg');
road3 = imread('hw1_data/hw1_dark_road_3.jpg');

figure;
subplot(3,1,1); imhist(road1);
title('Hisograms of original images');
subplot(3,1,2); imhist(road2);
subplot(3,1,3); imhist(road3);

% We can see that for road_1 there is a large peak around grayscale level 5
% and then a second smaller peak around the grayscale level 10. This second
% peak is due to the dark gray (but not quite black) pixels near the bottom
% of the image. This is probably due to the lights of the car illuminating
% the road. After these two peaks there is a drop in the histogram.

% For road_2, there is a single peak around the grayscale level 5 and then
% a gradual decrease.

% For road_3, there is a very large peak near the very bottom of the
% grayscale levels. This is due to the fact that almost the entire image is
% black. It has the fewest bright spots of the three images.
```



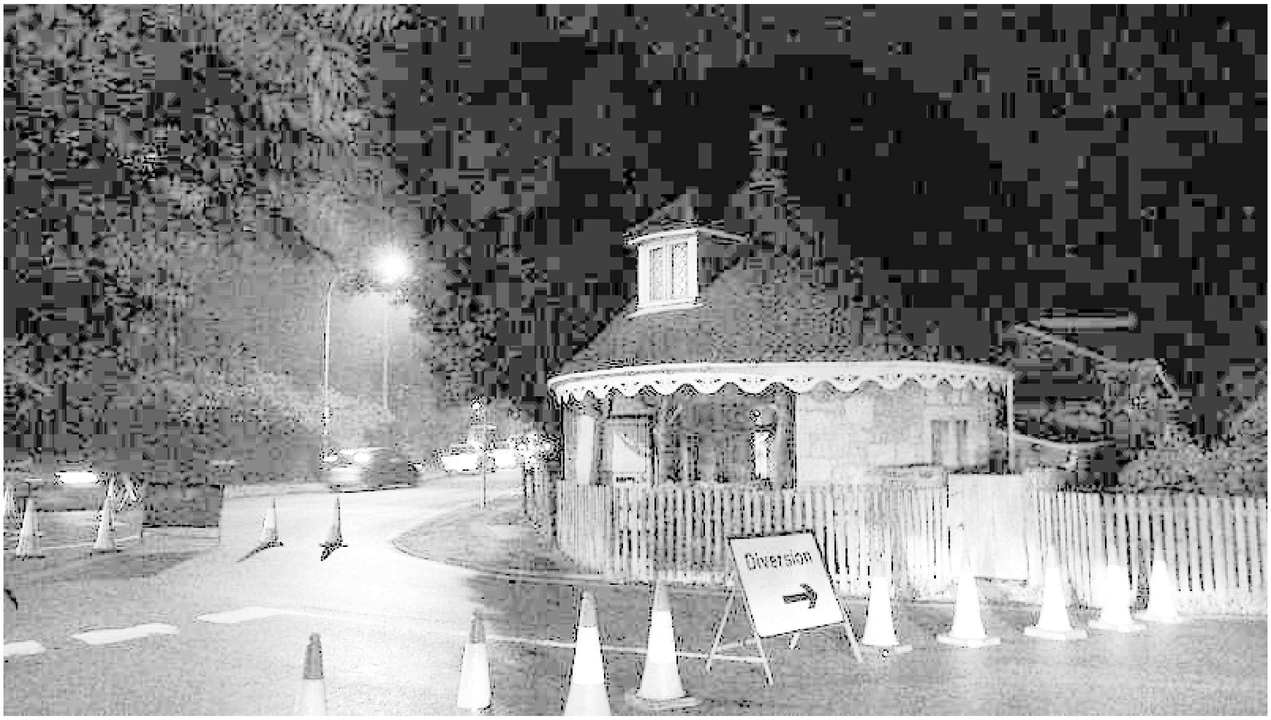
Question 4. Part B

```
road1_ghist = histeq(road1);
road2_ghist = histeq(road2);
road3_ghist = histeq(road3);

% figure;
% subplot(3,2,1); imshow(road1); title('Original Image');
% subplot(3,2,3); imshow(road2);
% subplot(3,2,5); imshow(road3);
% subplot(3,2,2); imshow(road1_ghist); title('With global histogram equalization');
% subplot(3,2,4); imshow(road2_ghist);
```

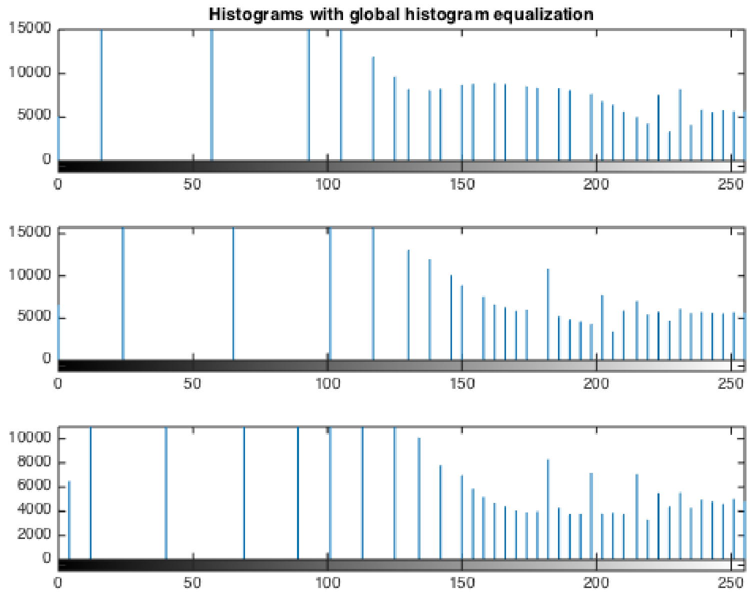
```
% subplot(3,2,6); imshow(road3_ghist);
figure; imshow(road1_ghist); title('With global histogram equalization');
figure; imshow(road2_ghist); title('With global histogram equalization');
figure; imshow(road3_ghist); title('With global histogram equalization');

figure;
subplot(3,1,1); imhist(road1_ghist);
title('Histograms with global histogram equalization');
subplot(3,1,2); imhist(road2_ghist);
subplot(3,1,3); imhist(road3_ghist);


% We can see that with equalization a lot of detail becomes visible. For
% example, in road_1 we can see the trees and the houses in front of the
% car. In road_2 we can see the edges of the house better, as well as the
% sign towards the left side of the image. In road_3 we can clearly see the
% building on the left as well as the building on the right. Both of these
% buildings were very hard to make out in the original image.

% However, we see a lot of patches showing up in the image equalized
% images. This is due to the fact that in histogram equalization, bins
% cannot be "split". That is, the number of non-zero bins cannot increase,
% and therefore there will be jumps between different grayscale values, not
% a gradual transition like we would expect.
```

With global histogram equalization

With global histogram equalization



With global histogram equalization



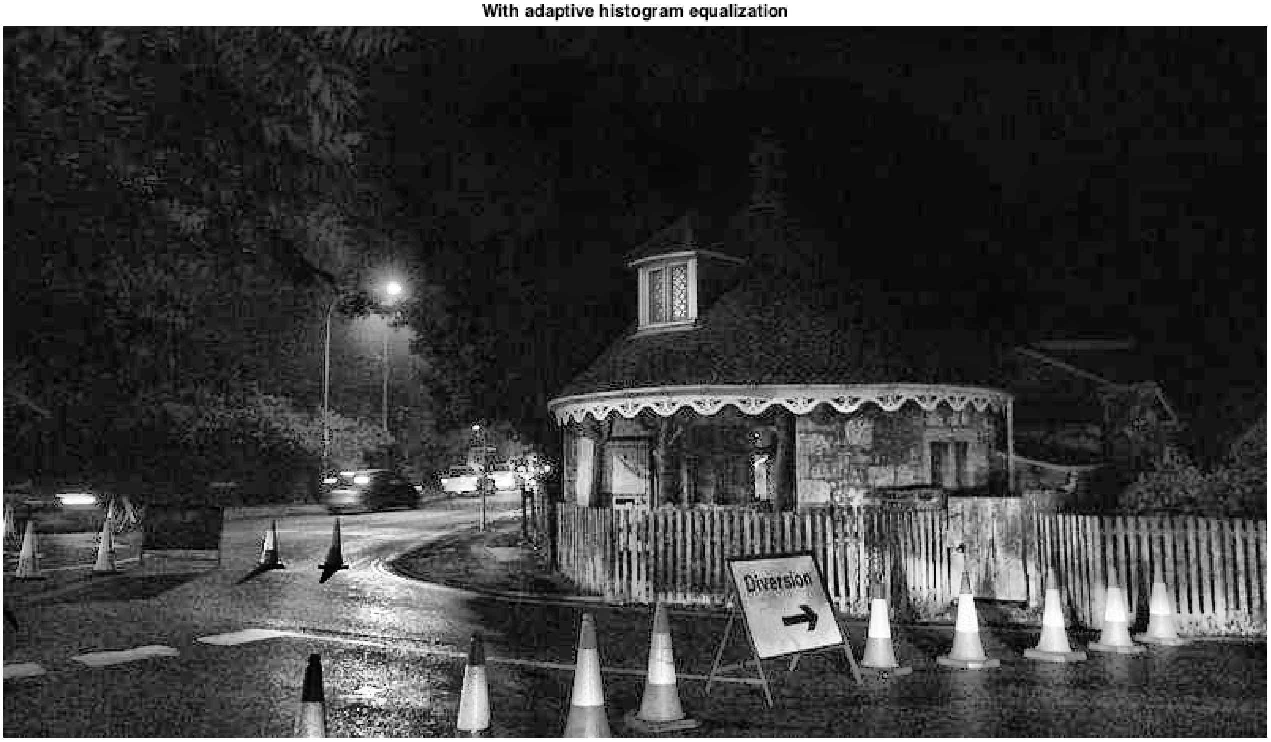
Question 4. Part C

```
road1_tiles = 16; road1_lim = 0.02;
road2_tiles = 16; road2_lim = 0.02;
road3_tiles = 16; road3_lim = 0.015;

road1_ahist = adapthisteq(road1, 'NumTiles', [road1_tiles, road1_tiles], 'ClipLimit', road1_lim);
road2_ahist = adapthisteq(road2, 'NumTiles', [road2_tiles, road2_tiles], 'ClipLimit', road2_lim);
road3_ahist = adapthisteq(road3, 'NumTiles', [road3_tiles, road3_tiles], 'ClipLimit', road3_lim);

display(['For Road_1: tiles = ' num2str(road1_tiles) 'x' num2str(road1_tiles) ' , Clipping limit = ' num2str(road1_lim) ]);
display(['For Road_2: tiles = ' num2str(road2_tiles) 'x' num2str(road2_tiles) ' , Clipping limit = ' num2str(road2_lim) ]);
display(['For Road_3: tiles = ' num2str(road3_tiles) 'x' num2str(road3_tiles) ' , Clipping limit = ' num2str(road3_lim) ]);

% figure;
% subplot(3,2,1); imshow(road1_ahist); title('With adaptive equalization');
% subplot(3,2,3); imshow(road2_ahist);
% subplot(3,2,5); imshow(road3_ahist);
% subplot(3,2,2); imshow(road1_ghist); title('With global equalization');
% subplot(3,2,4); imshow(road2_ghist);
% subplot(3,2,6); imshow(road3_ghist);

figure; imshow(road1_ahist); title('With adaptive histogram equalization');
figure; imshow(road2_ahist); title('With adaptive histogram equalization');
figure; imshow(road3_ahist); title('With adaptive histogram equalization');

figure;
subplot(3,1,1); imhist(road1_ahist);
title('Histograms with adaptive histogram equalization');
subplot(3,1,2); imhist(road2_ahist);
subplot(3,1,3); imhist(road3_ahist);


% We can see that when we apply adaptive histogram equalization as opposed
% to global histogram equalization we are able to pull out detail from the
% images without adding noise and "patchiness" to the image. We can still
% see the trees and the house in road_1, the edges of the roof and the road
% sign in road_2, and the buildings on the left and right in road_3.
```

```
For Road_1: tiles = 16x16 , Clipping limit = 0.02
For Road_2: tiles = 16x16 , Clipping limit = 0.02
For Road_3: tiles = 16x16 , Clipping limit = 0.015
```

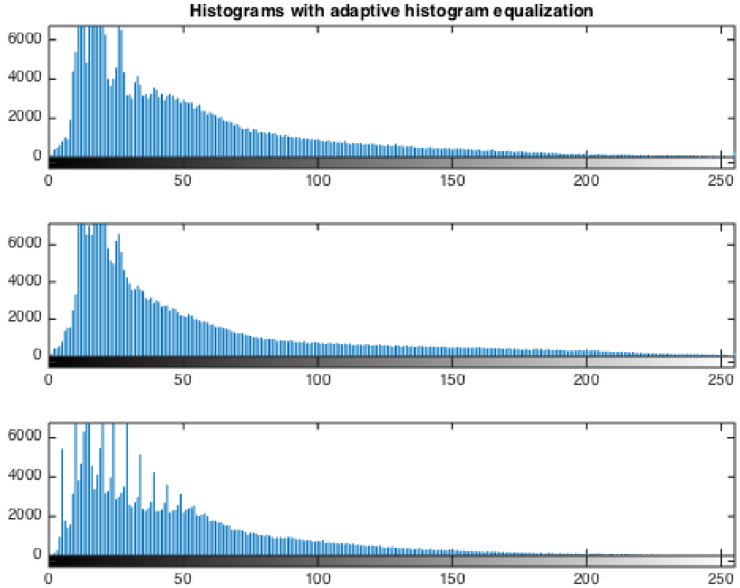
With adaptive histogram equalization

With adaptive histogram equalization



With adaptive histogram equalization

Histograms with adaptive histogram equalization



Published with MATLAB® R2014b