Problem set 1

SSE@TJU, Fall 2019

1. Denoising for Astrophotography

Amateur astrophotographers often set up static cameras pointed toward particular regions of the night sky and record for an extended period of time. Among the handouts, you can find two videos hw1_sky_1.avi and hw1 sky_2.avi, which contain two recordings of the night sky each lasting a few minutes. Low light levels cause the video frames to be noticeably noisy.

a) To generate a single denoised image from each video, compute a running average of the frames f^t $(t = 1, 2, ...)$ in the video without frame alignment, according to the following update rule:

$$f^1_{average} = f^1$$
$$f^t_{average} = \frac{t-1}{t} f^{t-1}_{average} + \frac{1}{t} f^t, t = 2, 3, \ldots$$

To access the video frames in MATLAB, the following code can be used:

```
vidobj = VideoReader(video.avi);
numFrames = get(vidobj, NumberOfFrames);
for i = 1 : numFrames
frame = im2double(read(vidobj, i));
end
```

Display and submit $f^t_{average}$ at $t = 30$ for each video. Comment on how effectively the noise is reduced and how much the sharp features are blurred by the averaging operation.

b) Now, compute a running average of the frames with frame alignment, according to the following update rule:

$$f^1_{average} = f^1$$
$$f^t_{average} = \frac{t-1}{t} f^{t-1}_{average} + \frac{1}{t} Align(f^t, f^{t-1}_{average}), t = 2, 3, \ldots$$

Here, $Align(f, g)$ aligns frames f and g by minimizing the mean squared difference over a set of horizontal and vertical shifts. The following MATLAB code shows an example of how to horizontally and vertically shift a frame:

```
dx = 1; % pixels
dy = -1; % pixels
A = [1 0 dx; 0 1 dy; 0 0 1];
tform = maketform(affine, A.);
[height, width, channels] = size(frame);
frameTform = imtransform(frame, tform, bilinear, XData, [1 width], YData, [1 height], ...
                                                        FillValues, zeros(channels, 1));
```

Display and submit $f^t_{average}$ at $t = 30$ for each video. Compare to the result in (a) and comment on how effectively noise is reduced while sharp features are better preserved.

2. Nighttime Road Contrast Enhancement

The visibility of lane markings, road signs, and obstacles on the roads is significantly reduced at nighttime. To assist drivers in dark conditions, we can perform contrast enhancement on images captured by the cars front-facing camera and display the enhanced images to the driver. On the handouts webpage, you can find three images captured at different times on different roads: hw1_dark_road_1.jpg, hw1_dark_road_2.jpg, and hw1_dark_road_3.jpg.

For each image, please perform the following operations and submit the required results.

(a) Plot and submit the histogram (MATLAB function: imhist) of the original images grayscale values. Briefly comment on the shape of each histogram.

(b) Apply global histogram equalization to the original image (MATLAB function: histeq is not allowed; just implement it by yourself and compare yours with Matlab's 'histeq'). Display and submit the modified image. Plot and submit the histogram of the modified images grayscale values. Comment on visually desirable/undesirable regions in the modified image.

c) Apply locally adaptive histogram equalization to the original image (MATLAB function: adapthisteq). Display and submit the modified image. Plot and submit the histogram of the modified images grayscale values. Choose and report the number of tiles and the clipping limit for attaining higher contrast while avoiding the generation of noisy regions and the amplification of nonuniform lighting effects. Comment on the subjective quality of the modified image compared to the result in (b).

d) Apply a γ-nonlinearity mapping to each image to perform contrast enhancement, show the new image, and submit the displayed image. For each image, find and report a value of γ that allows you to see more details.

Note: use latex to prepare your submission and moreover please include relevant MATLAB code in your submission.