Nonlinear Filters

Imaging Lab 4 - May, 2017

Kamyar Nazeri
Student ID: 100633486

1. Salt & Pepper Noise, Mean and Median Filters

To load an image, add 5% *Salt & Pepper* noise and apply *Mean* and *Media* filters, we have created a Matlab function (*Process.m*) which takes the ideal image, resizes it to the specified value, adds noise, filters and computes statistics.
Figure 1 shows the *Slat & Pepper* noise on an image and Mean/Median filters applied on the noisy image:

Figure 1: Salt & Pepper noise, the mean filtered image and the median filtered image

2. Mean and Median filters: different image sizes

We use Matlab's *imresize* command to create 4 versions of the original noise-free image with the following sizes:

500x500 1000x1000 2000x2000 4000x4000

The following table shows the runtime of each filter and RMSE/SNR for each filtered image:

	Mean Filter			**Median Filter**		
	Runtime(s)	**RMSE**	**SNR**	**Runtime(s)**	**RMSE**	**SNR**
500x500	0.003286	0.0371	46.9587	0.005668	0.0272	53.1623
2000x2000	0.010378	0.0115	56.6133	0.014682	0.0062	68.8205
2000x2000	0.053689	0.0038	64.6637	0.059251	0.0011	88.7440
4000x4000	0.190688	0.0017	67.229	0.210134	0.0002	107.6343

Table 1: Mean/Median filters statistics on different image sizes

The following shows the Matlab functions to calculate SNR, RSME and a function (Process) to resize an image to the specified value, add noise, filter and compute statistics:

```
function snr = SNR(denoised, ideal)
    % Calculates Signal-to-Noise ratio.
    % Input: Filtered image (denoised). Ideal image (ideal)
    % Output: Scalar snr specifying the SNR of a filter
    snr = 20*log(norm(double(ideal), 'fro')/(norm(double(ideal) -
       double(denoised), 'fro')));
end
```

Listing 1: Matlab SNR Function

```
function rmse = RMSE(denoised, ideal)
    % Calculates RMS error.
    % Input: Filtered image (denoised). Ideal image (ideal)
    % Output: Scalar rmse specifying the RMSE of a filter
    [m,n] = size(ideal);
    rmse = 1/(m*n)*norm( double(denoised)-double(ideal),'fro');
end
```

Listing 2: Matlab RMSE Function

```
function [noisy, mean, mean_rmse, mean_snr, median, median_rmse,
   median_snr] = Process(img, size)
    % Calculates Lab4 statistics.
    % Input: Image (img). Size of the image (size)

    resized = imresize(img, size);
    median = resized;
    noisy = imnoise (resized, 'salt & pepper', 0.05);
    w = fspecial('average', [5,5]);

    display(['Calculating mean filter ', num2str(size(1))]);
    tic; mean = imfilter(noisy, w, 'replicate'); toc

    display(['Calculating median filter ', num2str(size(1))]);
    tic;
    for i=1:3
        median(:,:,i) = medfilt2(noisy(:,:,i), [5,5]);
    end;
    toc

    resized_gs = rgb2gray(resized);
    mean_gs = rgb2gray(mean);
    median_gs = rgb2gray(median);

    mean_rmse = RMSE(mean_gs, resized_gs);
    mean_snr = SNR(mean_gs, resized_gs);

    median_rmse = RMSE(median_gs, resized_gs);
    median_snr = SNR(median_gs, resized_gs);
end
```

Listing 3: Matlab function to resize an image to the specified value, add noise, filter and compute statistics

Runtime vs. Image Size Visualization

Figure 2 shows the runtime vs. the image size plot for mean and median filters:

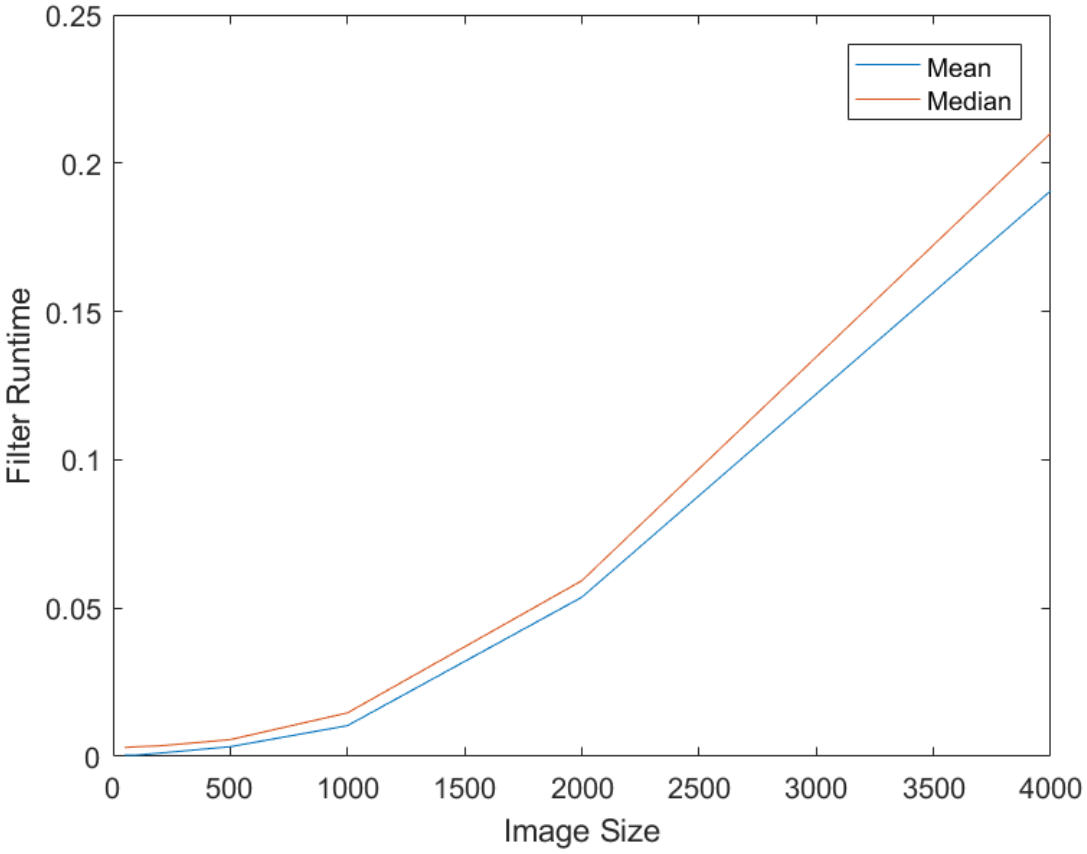

Figure 2: Runtime vs. the image size for mean and median filters

Conclusions

- The median filter does a better job on removing salt & pepper noise as one can visually compare images with mean and median filters in the *figure 1*.
- The SNR results in the *Table 1* corroborates previous arguments, note that Signal-Noise-Ratio for Median filter is much higher compared with the results of the Mean filter.
- The RMSE error value is decreased with the increase in the size of the image; this is mainly because we applied only 5% noise on all the sample images, and 5% noise in smaller images means more noise on the image hence larger error value.
- As one can see in *figure 2*, the *Median* filter is generally slower than the *Mean* filter, and the difference in run-time becomes larger with larger image sizes.