EE475 HW#3 **Fall 2019**

Adaptive Linear Filtering

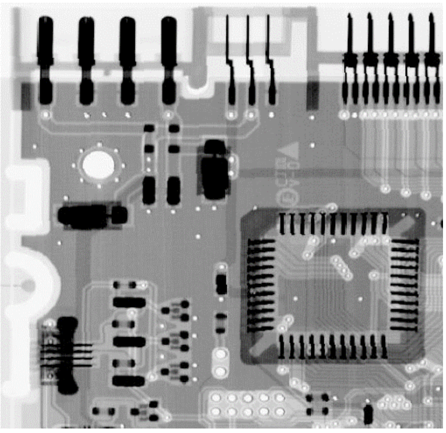
We would like the filter to attenuate the noise, but at the same time not to blur the image details. Let's do adaptive filtering to mitigate Gaussian noise in local neighborhood, centered at xy and of size nxn. Let μ_{xy} be the local mean, σ_{xy}^2 be the signal variance in the neighborhood, σ_n^2 the noise variance over all the image, f(x,y) the contaminated signal, g(x,y) the filtered signal. Consider the formula

$$g(x,y) = f(x,y) + \frac{\sigma_n^2}{\sigma_{xy}^2}\left(f(x,y) - \mu_{xy}\right)$$

If $if\ \frac{\sigma_n^2}{\sigma_{xy}^2} > 1, then\ set\ \frac{\sigma_n^2}{\sigma_{xy}^2} = 1.$

Contaminate the Circuit-board image with zero-mean Gaussian noise having variance $\sigma_n^2 = 1024$. Compare the image detail (e.g., legs of circuit components) preserving ability and noise removal performance of the following filters:

a) Filter the noisy image with 7x7 arithmetic mean filter
b) Filter it with the 7x7 geometric mean filter
c) Filter it with the above 7x7 adaptive linear filter
d) In general, you do not know the noise variance. How would you estimate it?



Adaptive Median Filter

Median filter is effective in removing impulsive noise. But it collapses if the impulse noise density is above 20% and also for large windows can distort image details. Consider the following adaptive median filter that tries to remove salt-and-pepper noise, to smooth other types of noise and to reduce distortion on the image content. The standard median filter replaces every pixel with its median; this filter either does the median substitution or preserves the original value.

z_{min}: minimum intensity value in the window
z_{max}: maximum intensity value in the window
z_{med}: median intensity value in the window

The algorithm is as follows:

A: If $z_{min} < z_{med} < z_{max}$, go to B
Else increase the window size

If window size ≤ max window size repeat A
Else, output z_{med}

B: If $z_{min} < f(x,y) < z_{max}$, output f(x,y)
Else output z_{med}

In the loop, start with 3x3 window and proceed till window size 7x7

a) Contaminate the Pentagon image with 25% salt and 25% pepper noise.
b) Filter the image with 7x7 median filter
c) Filter the image with the adaptive median filter where the loop starts with 3x3 window and proceeds till size 7x7
d) Compare and comment on the noise removal and image detail (e.g., cars in the carpark) preservation capability of the two filters

Notch filtering

The Moon and the Car images have been contaminated by a patterned interference.

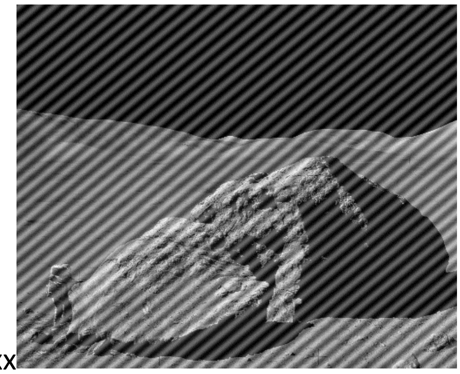
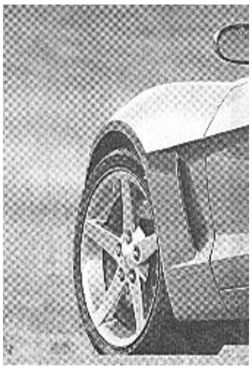
a) Plot the spectrum of the image
b) Threshold the spectrum to identify the frequencies of the spectral peaks due to interference
c) Apply Gaussian notch filter to separate the interference pattern and the original image; plot them separately.

One can build a notch filter, i.e., a band-stop filter by starting with a low-pass filter

$H_{LP}(u,v)$: $H_{HP}(u,v) = 1 - H_{LP}(u,v)$. For example, ideal HP filter
$H_{HP}(u,v) = \begin{Bmatrix} 0 & if\ D(u,v) \le D_o \\ 1 & if\ D(u,v) > D_o \end{Bmatrix}$, where D_o is the parameter to adjust the bandwidth

and D(u,v) is the Euclidean distance from the center of MxN rectangle, i.e.,
$D(u,v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$. The Gaussian high-pass filter then has the form
$H(u,v) = 1 - exp\{-D^2(u,v)/2D_o^2\}$.

The notch filter is simply $H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v)H_{-k}(u,v)$ where Q is the number of observed peak pairs $\{(u_k, v_k), (-u_k, -v_k)\}_{k=1}^{Q}$, and $D(u,v) = \sqrt{\left(u - \frac{M}{2} - u_k\right)^2 + \left(v - \frac{N}{2} - v\right)^2}$.

xx 

Heat Maps

Consider the Rose image. Contaminate it with salt-and-pepper noise with increments of 3%, from 3% to 30%. Try to remove the noise with a median filter with window sizes 3x3, 5x5, 7x7, 9x9, 11x11, 13x13. To show the improvements, if any, brought in by the median filter plot the image quality index SSIM before and after median filtering as a function of window size using a heat map.

Note 1:
In scientific research, we are often confronted with the problem of presenting a large quantity of numerical results, e.g., thousands of numbers. Just listing the numbers is not the best approach. Heat map is a visualization tool that at one look will tell the essence of all that bunch of numbers (http://en.wikipedia.org/wiki/Heat_map).

Plot the noise intensity, i.e., salt-and-pepper percentage on the horizontal axis and the filter size on the vertical axis. For the heatmap, you can use matlab's imagesc, colormap and colorbar functions. The heat should be proportional to the performance; in our case, the lower the S&P percentage, the better it is. Set blue to good performance, red to poor performance. For example, use colormap(flipud(jet)).

Note 2:

There are several metrics to measure image quality and the performance of image processing algorithms. The most common one is the Peak-Signal-to-Noise-Ratio

$PSNR = 10\log_{10}\dfrac{255^2}{\sum\sum\left[J(m,n)-I(m,n)\right]^2}$ where J and I are, respectively, the processed and the input images. SNR or PSNR is criticized for not reflecting subjective evaluations. Also PSNR needs a reference image. Another index, more reflective of human judgement and also one that does not need a referencve image is SSIM: Structural Similarity Index Measure:

https://en.wikipedia.org/wiki/Structural_similarity

https://www.mathworks.com/help/images/image-quality-metrics.html?requestedDomain=www.mathworks.com