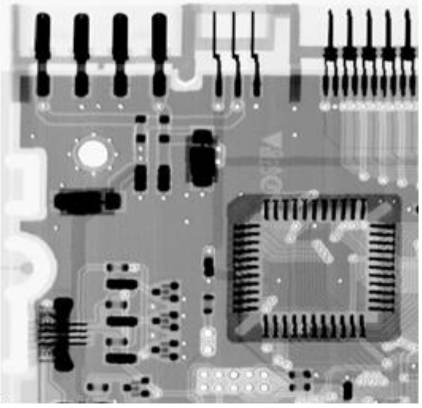
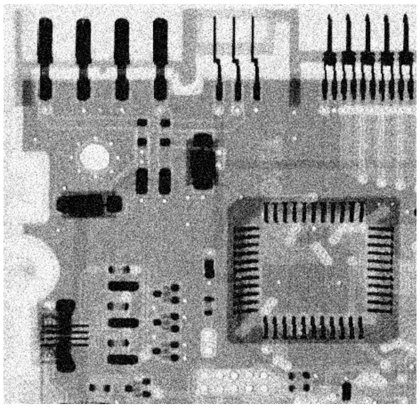
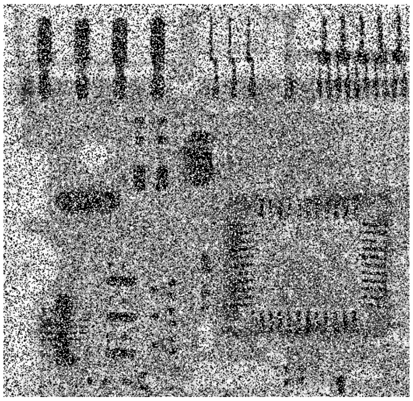
Project 4. Image Denoising and Wiener Filtering

1. **Adaptive local noise reduction (ALND) (5 points):** Given an original image (the left one, "circuit.bmp") and a noisy version (the middle one, "circuit_gauss001.bmp" which is corrupted by additive Gaussian noise of zero mean and variance 0.01 (the pixel values are 0-1), implement the adaptive noise reduction filter (Slides 17-18, Lecture 19) with different window sizes, and compared it with the arithmetic mean filter (*imboxfilt*) or Gaussian smoothing filter (*imgaussfilt*) of different window sizes. Evaluate image denoising performance for both ALND and the mean/Gaussian smoothing filters under different window sizes by using PSNR and SSIM discussed in Slides 7-10 of Lecture 23.


circuit.bmp


circuit_gauss001.bmp


circuit_spn05.bmp

2. **Adaptive Median filter (5 points):** Given a noisy image (the right one, "circuit_spn05.bmp") which is corrupted by salt and pepper noise with $P_a = P_b = 0.25$, implement an adaptive median filter (Slides 19-20, Lecture 19) that involves the adaptive window size to smooth impulse noise with larger probability and to preserve details. Compare the adaptive median filter with the regular fixed-window median filter under different fixed window sizes (*medfilt2).* Evaluate image denoising performance for both the adaptive median filter and the fixed-window median filters under different window sizes by using PSNR and SSIM discussed in Slides 7-10 of Lecture 23.

3. **Wiener filtering (10 points):** Given three images (*blur1.bmp*, *blur2.bmp*, *blur3.bmp*) which are degraded by the same motion blur ($a = b = 0.1$ and $T = 1$) and additive noise of different levels, you are expected to restore them with both inverse filtering and Wiener filtering. (1) You should try to implement inverse filtering first (with appropriate cut-off frequency values) (Slides 4-5, Lecture 22); (2) You should implement the Wiener filtering by adjusting the constant K value (Slide 10, Lecture 22) to obtain the best performance given the original image (*original.bmp*), e.g., the highest PSNR and SSIM. (3) For three degraded images with the same motion blur but different noise levels, please show and compare the restoration results by inverse filtering and Wiener filtering. (3) For the two noisy images (blur2.bmp and blur3.bmp), you need to study if pre-denoising is helpful. Which denoising filter should be used and why? You are allowed to use suitable Matlab functions for denoising, such as *medfilt2*, *imboxfilt*, or *imgaussfilt* prior to Inverse filtering or Wiener filtering. Discuss your findings and visualize experimental results with quantitative PSNR/SSIM results. (Note: You should use the SINC function in Matlab to create the transfer function $H(u, v)$ of the motion blur to avoid directly using the ratio of $\frac{sin(x)}{x}$).

Note: Two Matlab demo functions are provided for you to do this project. The first one "noiseimage.m" can be used to create noisy images of different noise distributions or different noise levels. The second one "degrad_t.m" is a demo code for inverse filtering to restore the different atmospheric turbulence degradations.