EIE4512 - Digital Image Processing – Homework #1

Instructor: Prof. Zhen Li

Due on 04/10/2019 17:00 pm

Written exercises

problem 1

Show that the Fourier transform of the 2-D continuous sine function

$$f(x, y) = A\sin(u_0x + v_0y)$$

is the pair of conjugate impulses

$$F(u, v) = -j\frac{A}{2}\left[\delta\left(u - \frac{u_0}{2\pi}, v - \frac{v_0}{2\pi}\right) - \delta\left(u + \frac{u_0}{2\pi}, v + \frac{v_0}{2\pi}\right)\right]$$

[*Hint:* Use the continuous version of the Fourier transform $F(\mu, \nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t, z)\, e^{-j2\pi(\mu t + \nu z)}\, dt\, dz$, and express the sine in terms of exponentials.]

problem 2

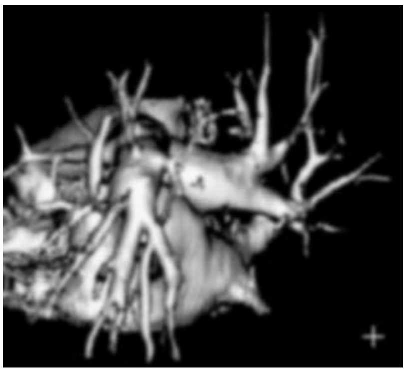
Consider a linear, position-invariant image degradation system with impulse response

$$h(x - \alpha, y - \beta) = e^{-[(x-\alpha)^2 + (y-\beta)^2]}$$

Suppose that the input to the system is an image consisting of a line of infinites-imal width located at $x = a$, and modeled by $f(x, y) = \delta(x - a)$, where δ is an impulse. Assuming no noise, what is the output image $g(x, y)$?

problem 3

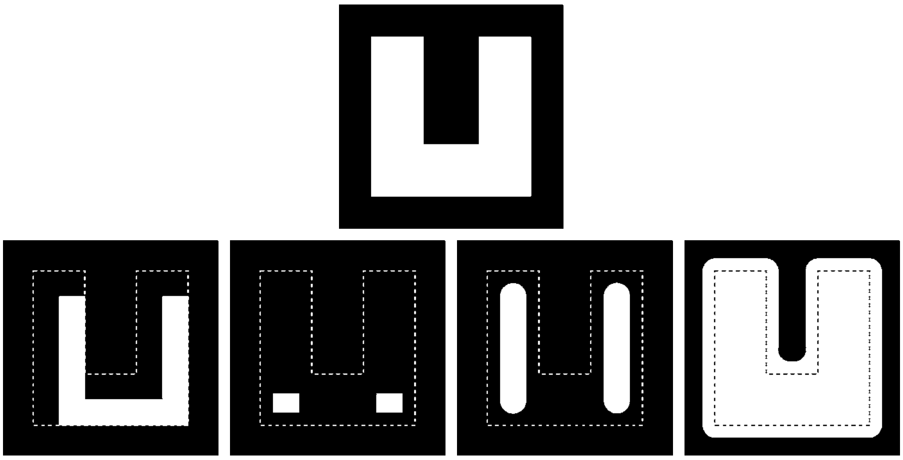
The image shown is a blurred, 2-D projection of a volumetric rendition of a heart. It is known that each of the cross hairs on the right bottom part of the image was 3 pixels wide, 30 pixels long, and had an intensity value of 255 before blurring. Provide a step-by-step procedure indicating how you would use the information just given to obtain the blurring function H(u, v).



(Original image courtesy of G.E. Medical Systems.)

problem 4

With reference to the image shown, give the structuring element and morphological operation(s) that produced each of the results shown in images (a) through (d). Show the origin of each structuring element clearly. The dashed lines show the boundary of the original set and are included only for reference. Note that in (d) all corners are rounded.



(a) (b) (c) (d)

problem 5

Prove the validity of the duality expressions $(A \bullet B)^c = (A^c \circ \hat{B})$ and $(A \circ B)^c = (A^c \bullet \hat{B})$.

problem 6

Sketch the result of applying the hit-or-miss transform to the image and structuring element shown. Indicate clearly the origin and border you selected for the structuring element.

Image

Structuring element

Matlab exercises

problem 1

In this problem, we study how to recognize the characters on a license plate using morphological image processing. Please use the two images pro1_license_plate_clean.png, pro1_license_plate_noisy.png, and pro1_character_templates.zip, which contains 36 separate images of the alphanumeric characters "ABC…XYZ0123456789" in the same font as the "EIBLT8H" written on the license plate.

pro1_license_plate_clean.png

pro1_license_plate_noisy.png

(a) Binarize the clean license plate image, the noisy license plate image, and the template images, so that the large characters in the middle of the plate appear as white and the background appears as black. Choose the threshold by applying Otsu's method (https://www.mathworks.com/help/images/ref/graythresh.html) on the clean license plate image, and use the same threshold for all binarization operations in this problem. Submit the binarized clean and noisy license plate images and the binarized template images.

(b) Perform character detection by erosion in the binarized clean license (MATLAB function: `imerode`). To eliminate the effects of slight mismatches along the character boundaries, compute a slightly eroded template which is erode(template, 3x3 square), and then use the eroded template as the SE in the erosion detector. For each template that generates a nonzero detection result, dilate the eroded license plate image (MATLAB function: `imdilate`) by the template for visualization purposes and submit the dilation result, like shown in the following example for the template "B". Which characters are detected by this approach? Are there any false positive detections, and if so, why?

(c) Repeat Part B, except use a hit-miss filter in place of erosion for the detector (MATLAB function: `bwhitmiss`). Use the same SE as in Part B for the foreground SE. For the background SE, use the difference dilate(template, 5x5 square) – dilate(template, 3x3 square) which extracts a thin outline around the character. Comment on the advantages of the hit-miss detector compared to the erosion detector.

(d) Repeat Part C, except replace the clean license plate image by the noisy license plate image. Comment on how the noise affects the hit-miss detector's accuracy.

(e) Repeat Part C, except replace the clean license plate image by the noisy license plate image and replace the hit-miss filter by the minimum of two rank filters: rank-filter(binary image, p_1, SE_1) and rank-filter(NOT[binary image], p_2, SE_2). Here, SE_1 and SE_2 are the same foreground and background SEs used in Part C, and p_1 and p_2 indicate the two ranks. Note that if $p_1 = p_2 = 1$, then the minimum of the two rank filters is the same as the hit-miss filter. Choose and report the ranks p_1 and p_2 that enable correct detection of the characters in the noisy license plate without false positives. The MATLAB function for rank filtering is `ordfilt2`.

Note: Please attach relevant MATLAB code.

problem 2

Sharpness Enhancement by Dilation and Erosion

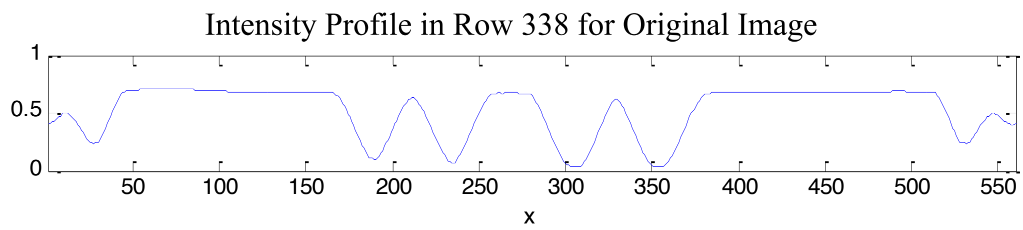
In this problem, we use an iterative grayscale morphological image processing algorithm to enhance the sharpness of structures in a blurry image.

(a) Please use the image pro2_road_sign_school_blurry.jpg. This image shows a school crossing road sign taken by an out-of-focus camera. Apply 10 iterations of the following algorithm. Design and report your structuring element. Submit the resulting image after 10 iterations. Comment on which features in the image have been made sharper. (MATLAB functions: `imdilate, imerode, strel`).

```
Im := Input Image
For Iteration = 1:NumIterations
      Im_d = dilate(Im, W)    % Note that this is grayscale dilation
      Im_e = erode(Im, W)     % or erosion with structuring element W
      Im_h = 0.5(Im_d + Im_e)

      % Perform the following test for each pixel
      If Im ≥ Im_h
            Im := Im_d
      Else
            Im := Im_e
      End
End
```

(b) Submit plots of the intensity profile in row 338 after iterations 1, 2, …, 6. For example, the following is a plot of the intensity profile in row 338 in the original image. Comment on how the algorithm iteratively changes the intensity profile.

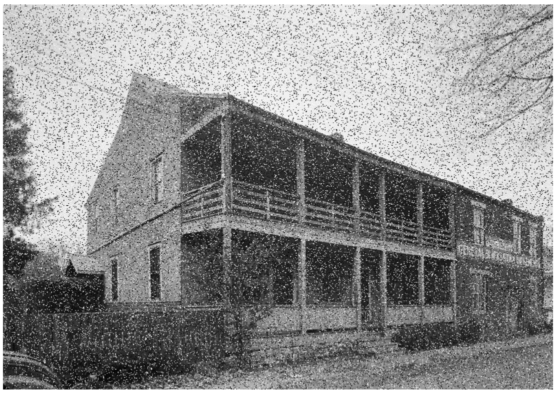
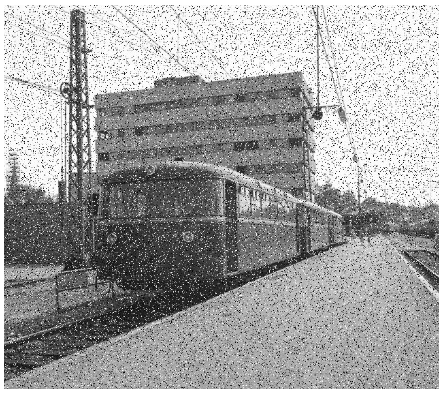


Note: Please attach relevant MATLAB code.

problem 3

Noise Reduction by Median Filtering

In this problem, we study how to use different types of median filtering to reduce salt-and-pepper noise in vintage photographs.

(a) Please use the images pro3_building.jpg and pro3_train.jpg . Apply median filtering with a 3x3 window and a 5x5 window (MATLAB function: `medfilt2`). Display and submit the resulting images. For each window size, comment on how effectively the noise is reduced while sharp edges and features in the image are preserved.

(b) Given a set of input values $f_1, f_2, \cdots, f_N$ and weights $w_1, w_2, \cdots, w_N$, weighted median filtering repeats the value f_i by w_i times and then computes the median of all the repeated values:

$$g = median\left(w_1 \otimes f_1, w_2 \otimes f_2, \cdots, w_N \otimes f_N\right)$$
$$w \otimes f = \underbrace{f, f, \cdots, f}_{w \text{ times}}$$

Implement a weighted median filter with the following 5x5 window of weights, where the brackets around 4 indicate the center of the sliding window:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & [4] & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

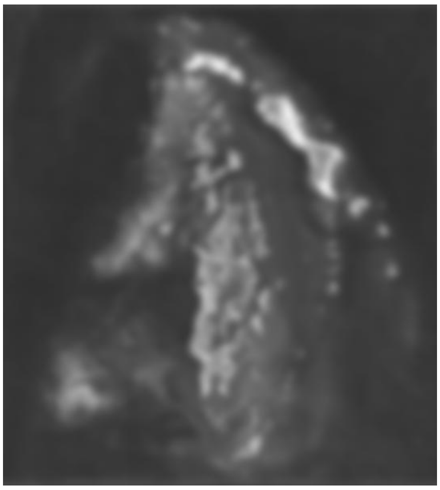
Apply your weighted median filter to each image. Display and submit the resulting image. Compare to the results from part (a) in terms of how effectively noise is reduced while sharp edges and features are preserved.

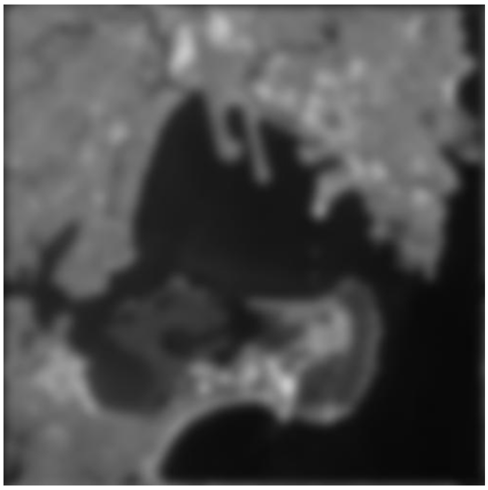
Note: Please attach relevant MATLAB code.

problem 4

Wiener Filtering of Satellite Images

Satellite images can be degraded by atmospheric turbulence, which we try to model as linear, shift-invariant blurring with an isotropic Gaussian impulse response (with unknown standard deviation) plus additive white noise (with unknown variance). Please use pro4_satellite_1_degraded.tiff and pro4_satellite_2_degraded.tiff .





(a) Design and implement a method to estimate the standard deviation of the additive white noise directly from the degraded image. For each image, report the estimated noise standard deviation, assuming image intensity values in the range [0,1].

(b) Design and implement a method to estimate the standard deviation of the Gaussian impulse response directly from the degraded image. For each image, report the estimated Gaussian standard deviation.

(c) Perform inverse filtering on each image, using the estimated Gaussian impulse response from Part B (function: `deconvwnr` with a noise estimate of 0). Submit the inverse filtered image. Comment on the quality of the restored image.

(d) Perform Wiener filtering on each image, using the estimated noise standard deviation from Part A and the estimated Gaussian impulse response from Part B (function: `deconvwnr`). Submit the Wiener filtered image. Comment on the quality of the restored image and compare to the result from Part C.

Note: Please include relevant MATLAB code.