Local Enhancement and Spatial Domain Filtering

The histogram processing methods, discussed in the second session, are global. These transformation functions are designed according to the gray-level distribution over the entire gray levels of an image. They are good when we want to enhance the entire range of gray levels and are not suitable for enhancing details over small areas. These small areas have negligible influence on designing the global transformation function. In this worksheet, our objective is to study the transformation functions that are designed for small areas of gray levels in an image.

Enhancement Based on Local Histogram Processing

In spatial domain filtering square or rectangular neighborhood (block) is defined and the histogram in the local block is computed. The design of transformation functions is based on the local gray-level distribution. Then the local enhancement transformation function e.g. histogram equalization or specification method is used to generate the transformation function, which performs the gray level mapping for each pixel in the block. The center of the block is moved to an adjacent pixel location and repeat this process is repeated until we reach the last pixel in the image.

Since the block only shifts one pixel each time the local histogram can be updated each time without re-computing the histogram over all pixels in the new block. If utilizing a non-overlapping region shift, the processed image usually has an undesirable *checkerboard* effect.

Fig. 7.1 shows the difference between global histogram and local histogram processing. Fig. 7.1(a), Fig. 7.1(b),and Fig. 7.1(c) show noised and blurred image, output from applying global histogram, and the result from local histogram processing, respectively. We can see significant visual enhancement in Fig. 7.1(c) over Fig. 7.1(b). In Fig. 7.1(b) the noise

contents were also enhanced. While in Fig. 7.1(c) their effect was minimized due to the use of a 7 x 7 neighborhood which has too small an influence on the global histogram specification.

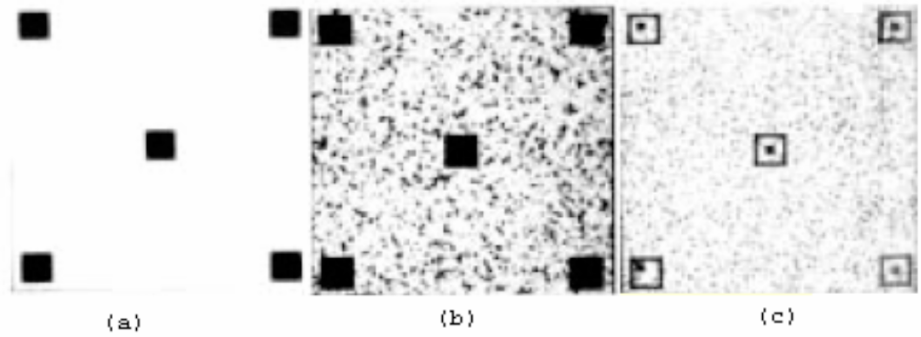

Figure 7.1: (a) Input noised and blurred image (b) Output image after global histogram processing (c) Output image after local histogram processing

Local Enhancement Based on Spatial(Mask) Filtering

In this section, we will focus on spatial filtering and its contribution to the enhancement of an image. A sub-image Fig. 7.3(b) called *filter*, *mask*, *kernel*, *template*, or *window* is masked with input image Fig. 7.3(a) as in Eq. 1. The values in the window are called filter coefficients. Based on the filter coefficient we can classify spatial filters as linear filters(LF), nonlinear filters(NLF), low pass filters(LPF), and high pass filters(HPF) based on spatial filtering.

$$g(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} \omega(i,j).f(x+i,y+j) \tag{1}$$

Specifically, the response of a 3 x 3 mask with a sub-image with gray levels z_1, z_2,z_2,...,z_9 is given in Eq. 2

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i eq : shortmask \tag{2}$$

Mask operation near the image border some part of the masks is located outside the image plane; to handle this problem we use

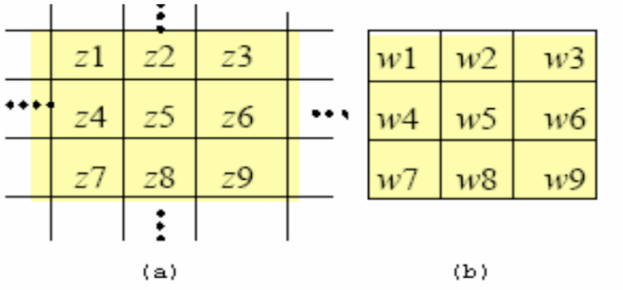

Figure 7.3: Spatial Mask with w1, w2 etc as filter coefficients

1. Discard the problem pixels (e.g. 512 x 512 input becomes 510 x 510 output image if the mask is 3 x 3)

2. Zero padding at the first and last rows and columns. So, the 512 x 512 image become equal to the 514 x 514 intermediate output image. To get the final output 512 x 512 image the first and last rows and columns are discarded.

Smoothing Spatial Filtering

Smoothing filters are also called LPF. Because it attenuates or eliminates high-frequency components that characterize edges and sharp details in an input image. It results in blurring or noise reduction. Blurring is usually used in preprocessing steps, e.g., to remove small details from an image prior to object extraction or to bridge small gaps in lines or curves. These filters are based on neighborhood averaging and in general M x M masks with the same filter coefficients, Eq. 3 are used to design an LPF. Some filters have weighted masks. To preserve the nearest neighborhood

effect they are weighted with some weighting factor. These filters produce some undesirable edge blurring.

$$w_i = \frac{1}{M^2}; 1 \leq w_i \leq M \qquad (3)$$

Order Statistic Nonlinear Filters

The order statistic filters are nonlinear filters whose response is based on ordering (ranking) the pixels contained in the image area surrounded by the filter, and then the center value is replaced with the value determined by the ranking result. Based on ranking criteria we can classified order statistic filters into three types.

Median(50th Percentile) Filter

The transformation function of the median filter is given in Eq. 4. Median filters are useful in a situation where impulse noise, salt and pepper noise. The median, ξ, of masked neighbors, $z_k(x,y)$, is calculated by ranked in ascending order of gray levels. Ultimately, half of the pixels are above the median, ξ, and half are below ξ. And finally assigned to the output pixel $R(x,y)$, at (x,y).

$$R(x,y) = \xi(z_k(x,y)|k = 1,2,...,9) \qquad (4)$$

Generally, The transfer function of the median filter forces the output gray levels to be more similar to the neighbors. If the isolated group of pixels has area $A \leq \frac{n^2}{2}$ is eliminated by the n x n median filter. Conversely, the larger cluster is less affected by it.

Min (0th Percentile) Filter

The transformation function of the *min filter* is given by Eq. 5.

$$R_k(x,y) = min(z_k|k = 1,2,3,...,9) \qquad (5)$$

These filters are applied, similar to the *median* filters, on input images to result in masked intermediate outputs $z_k(x,y)$ but the ranking and assigning criteria are different from median filters. The assignment of minimum value, to the output $z_k(x,y)$, in the neighborhood makes it useful to remove *salt noise*.

Max (100th Percentile) Filter

The transformation function of the *min filter* is given by Eq. 6.

$$R_k(x,y) = max(z_k|k = 1,2,3,...,9) \qquad (6)$$

These filters are applied, similar to *min* filters, on input images to result in masked intermediate outputs $z_k(x,y)$ but the ranking and assigning criteria are different from both *mean* and *median* filters. The assignment of maximum value, in the neighborhood, to the output $z_k(x,y)$ makes them useful to remove *pepper noise*.

Activity No.1

Take the image of your choice and take out hidden objects in the background using the techniques discussed above.

Hint for Activity No.1

Apply a 3 x 3 moving average filter on the input image and detect areas of higher contrast like edges and then use the co-efficient in $g(x,y)$ to produce the output image.

Activity No. 2

Take the input image and apply the smoothing filter mask of order: *i*− 3 x 3, *ii*− 5 x 5, *iii*− 9 x 9, and *iv*− 15 x 15. Sketch the output image and comment on your results.

Activity No. 3

Take a noised image and prove which *order statistic filter* is best.

Hint for Activity No. 3

Read an image and add *salt* and *pepper* noise with it using *imnoise*() function. And then write your code to implement Eq. 4,5, and 6.