Digital Image Processing
Lectures 19 & 20

M.R. Azimi, Professor

Department of Electrical and Computer Engineering
Colorado State University

Spatial Operations

Unlike point operations, spatial operations involve manipulations on several pixels in a local neighborhood surrounding a given pixel. For linear operations, the process can be viewed as the convolution of the image with a 2-D finite impulse response (FIR) filter or a spatial mask. The coefficients of the filter can be chosen to perform a wide variety of tasks including noise removal or smoothing, edge detection and sharpening, and template matching and target detection.

Noise Removal & 2-D Low-pass Filtering

Any image is subject to noise and interference due to various sources such as sensor noise, film grain noise, channel noise, and speckle noise in synthetic aperture radar or sonar (SAR/SAS). The noise can corrupt an image either additively or multiplicatively depending on its source. Here, we introduce some *ad-hoc* spatial operations that don't use any *a priori* knowledge about the image and/or the noise properties for removing additive/multiplicative noise.

Spatial Low-Pass Filtering

Effective method for removing additive Gaussian noise from noisy images. Uses a linear 2-D FIR filter where each pixel in an image is replaced by the weighted sum of the neighboring pixels within the mask i.e.

$$y(m,n) = \sum_{k,l \in W} \sum h(k,l)x(m-k,n-l)$$

$x(m,n)$: input image, $y(m,n)$: output image, $h(k,l)'s$: filter coefficients or the weights, and W: a suitable mask. Note that for low-pass filtering $\sum\sum_{k,l \in W} h(k,l) = 1$. A common choice is $h(k,l) = \frac{1}{N_W}, \forall (k,l) \in W$ i.e. "spatial averaging" where N_W represents the number of pixels in the mask, W.

Figures 1(a)-(c) show three different 3×3 masks with different choices of coefficients. The choices of the filter coefficients and window size present a trade-off between noise removal ability and the edge smearing artifacts caused due to loss of high frequency information.

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

(a) Mask 1

0	$1/8$	0
$1/8$	$1/2$	$1/8$
0	$1/8$	0

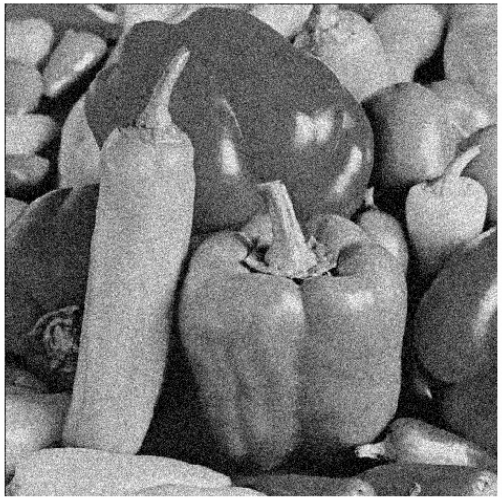
(b) Mask 2

0	$1/5$	0
$1/5$	$1/5$	$1/5$
0	$1/5$	0

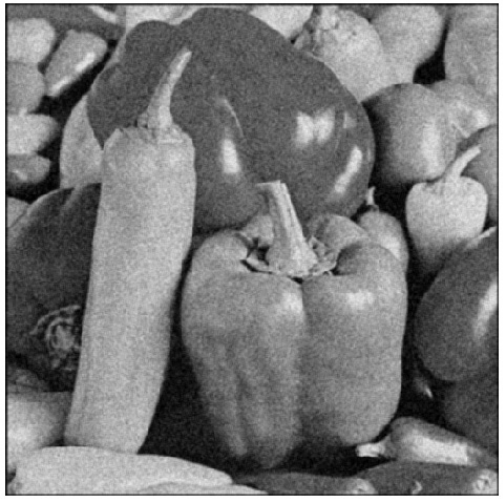
(c) Mask 3

Figure 1: Different Spatial Low-Pass Filtering Masks.

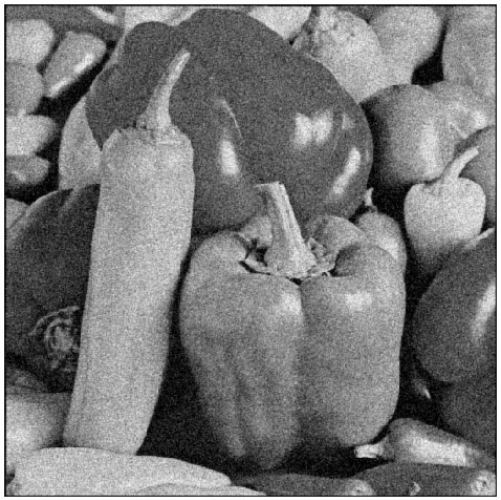
Fig. 2(a) shows noisy "Peppers" image corrupted by white Gaussian noise ($SNR = 6.6dB$). Figs. 2(b) & (c) show filtered images ($SNR_1 = 16dB$ and $SNR_2 = 12dB$) using Masks 1 and 2, respectively. Although noise removal ability of the first mask is better, some smearing artifacts are also more noticeable in Fig. 2(b). This can be explained by comparing magnitude responses of these filters in Figs 3(a) & (b). The first mask has a smaller bandwidth, hence better noise removal ability and more blurring artifacts.



(a) Noisy Pepper Image.

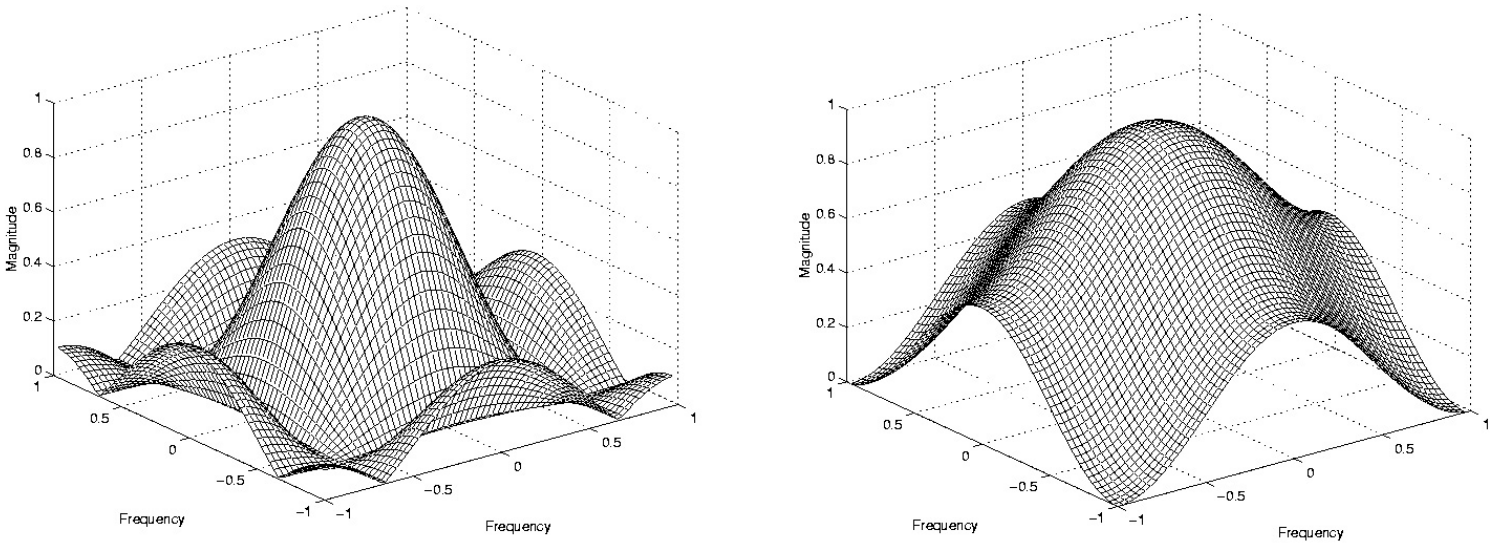


(b) Filtered Image-Mask 1.



(c) Filtered Image-Mask 2.

Figure 2: Noisy and Filtered Pepper Images.



(a) Mask 1 (b) Mask 2

Figure 3: Magnitude Responses for Masks 1 and 2.

Example 1: Consider the two 5-point masks in Fig. 1(c) with

$$y_1(m,n) = \frac{1}{5}[x(m,n)+x(m-1,n)+x(m+1,n)+x(m,n-1)+x(m,n+1)]$$

$$y_2(m,n) = \frac{1}{8}[4x(m,n)+x(m-1,n)+x(m+1,n)+x(m,n-1)+x(m,n+1)]$$

Compare their frequency responses in terms of their selectivity and noise removal ability.

Using 2-D DSFT

$$\begin{aligned} H_1(\Omega_1,\Omega_2) &= \frac{Y_1(\Omega_1,\Omega_2)}{X(\Omega_1,\Omega_2)} \\ &= \frac{1}{5}[1+e^{-j\Omega_1}+e^{j\Omega_1}+e^{-j\Omega_2}+e^{j\Omega_2}] \\ &= \frac{1}{5}[1+2cos\Omega_1+2cos\Omega_2] \end{aligned}$$

For the second one, $H_2(\Omega_1,\Omega_2)=\frac{1}{8}[4+2cos\Omega_1+2cos\Omega_2]$. Forming the cross-section of these 3-D plots with the plane $\Omega_1-\Omega_2=0$ gives

$$H_1(\Omega_1,\Omega_2)=\begin{cases} 1, & \Omega_1=\Omega_2=0 \\ \frac{1}{5}, & \Omega_1=\Omega_2=\frac{\pi}{2} \\ 0, & \Omega_1=\Omega_2=0.58\pi \end{cases}$$

$$H_2(\Omega_1,\Omega_2)=\begin{cases} 1, & \Omega_1=\Omega_2=0 \\ \frac{1}{2}, & \Omega_1=\Omega_2=\frac{\pi}{2} \\ 0, & \Omega_1=\Omega_2=\pi \end{cases}$$

i.e. Mask 3 has a shorter bandwidth in comparison with Mask 2 and thus it produces more blurring artifacts.

To determine the effect of the window size on the noise removal ability of the spatial filters consider a noisy image $z(m,n)$

$$z(m,n) = x(m,n) + \eta(m,n)$$

$x(m,n)$: noise-free image and $\eta(m,n)$: additive white Gaussian noise with $\mu_\eta = 0$ and variance, σ_η^2. Spatial averaging gives

$$y(m,n) = \frac{1}{N_W} \sum\sum_{k,l \in W} x(m-k, n-l) + \overline{\eta}(m,n)$$

$\overline{\eta}(m,n) = \frac{1}{N_W} \sum\sum_{k,l \in W} \eta(m-k, n-l)$ is the filtered version of $\eta(m,n)$ (or left-over noise) in the processed image. It can easily be shown that $\overline{\eta}(m,n)$ has also zero mean and variance $\overline{\sigma}_\eta^2 = \frac{\sigma_\eta^2}{N_W}$, i.e. noise power is reduced by factor N_W. Note that here it is assumed that the spatial averaging does not impact the image itself. However, increasing the size of the window increases the edge smearing artifacts as a result of reducing the bandwidth of the spatial filter. Figs. 4(a) & (b) show the results of applying spatial averaging filters of size 5×5 and 7×7, respectively.

(a) 5x5 Mask (b) 7x7 Mask

Figure 4: Filtered Images.

Median Filtering

Median filtering is a nonlinear filtering process primarily used to remove impulsive or "salt & pepper" type noise. Similar to the spatial filtering, median filter operation involves sliding a window encompassing an odd number of pixels. The center pixel in the window is then replaced by the *median* of the pixels within the window, i.e.

$$y(m,n) = median[x(m-k, n-l)], \quad k,l \in W$$

Finding the median value requires arranging pixel intensities in the window in increasing or decreasing order and picking the middle value. Generally, window size, N_W, is chosen to be an odd number to facilitate the selection of the median. Some of the key properties of the median filter are listed below.

Properties:

1. Nonlinear filter, i.e.

$$median\{x_1(m) + x_2(m)\} \neq median\{x_1(m)\} + median\{x_2(m)\}$$

2. Performs very well on images containing salt & pepper noise and impulsive noise, but performs poorly when noise is Gaussian.
3. When number of noise pixels in window $\geq \frac{N_W}{2}$, it performs poorly.
4. Median filtering does not cause prominent edge smearing artifacts. Nonetheless, as window size increases, median gets closer to the local mean and hence more smearing and edge distortion will be noticeable.

5. When N_W is odd, search for the median, i.e., $\frac{N_W+1}{2}$ largest value, requires

$$(N_W - 1) + (N_W - 2) + ... + \frac{N_W - 1}{2} = \frac{3 \times N_W^2 - 1}{8}$$

comparisons, e.g., for 3×3 and 5×5 windows, the number of comparisons are 30 and 224, respectively.

6. A 2-D median filter is a separable filter and hence can be implemented by successive 1-D median filters to row and columns.

7. To protect sharp corners and thin lines in the images, other shaped masks (e.g., cross-shaped) are preferred.

Figures 5 show the "Baboon" image corrupted with salt & pepper noise density of 40% and median filtering results using 3x3, 5x5 and 7x7 size windows. As can be observed, even for this level of noise median filter performs very well. Although the 3x3 median filter leaves behind some noise samples, due to relatively high density of the noise, it causes the least blurring artifacts amongst all the masks considered. The smearing is more evident in the hairy and whiskers areas with high spatial activity or rough texture.

The performance plot, which is the plot of the percent noise versus the variance of the leftover noise in the processed image, is given in Figure 6 for different window sizes. As expected, when noise density increases the ability of the filter to remove the noise deteriorates. This deterioration is more prominent for the smaller size windows.

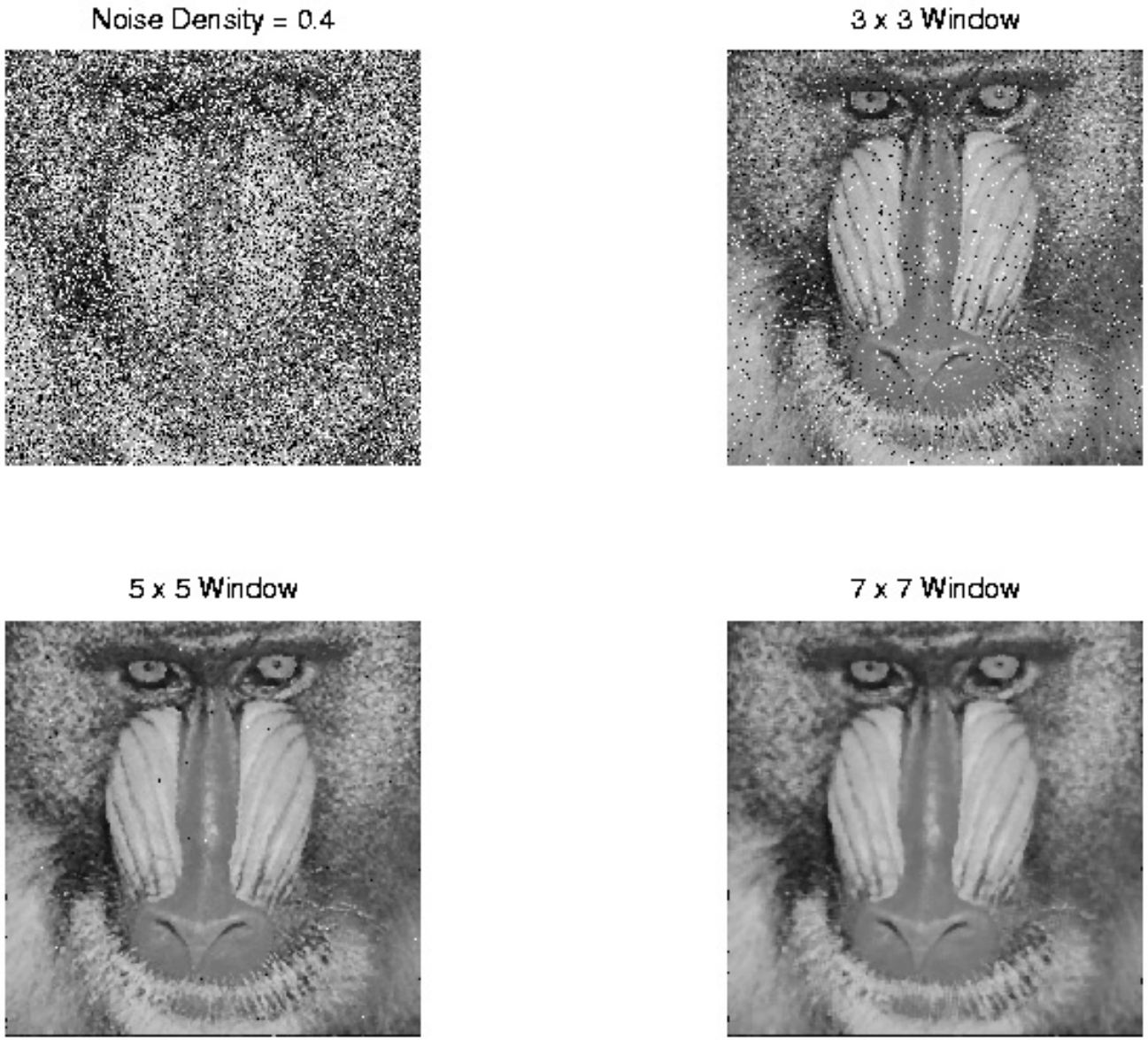


Figure 5: Noisy Baboon and Median Filtered Versions with Different Sizes.

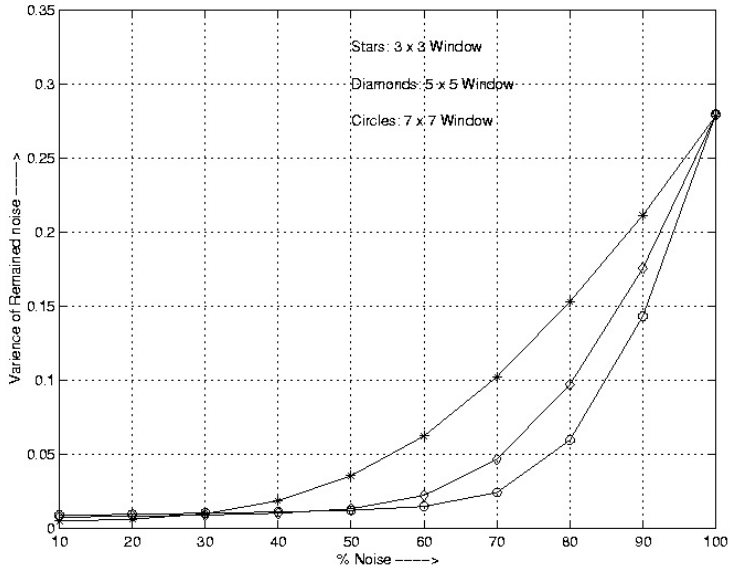


Figure 6: Performance plots for Median Filter with Different Sizes.

Example 1:

Consider a 1-D signal $x(m), m \in [0, M]$. Show that $\zeta(\hat{x})$ given below is minimized when $\hat{x} = median\{x(m)\}$

$$\zeta(\hat{x}) = \sum_{m=1}^{M} |x(m) - \hat{x}|$$

Let S_1 and S_2 be two subsets of indices m with $S_1 \bigcup S_2 = \{1, \cdots, M\}$ and cardinality N_1 and N_2 respectively, such that

$$\zeta(\hat{x}) = \sum_{m \in S_1} (x(m) - \hat{x}) + \sum_{m \in S_2} (-x(m) + \hat{x})$$

Now, taking partial wrt $\hat{x}$ gives $-N_1 + N_2 =$ i.e. $N_1 = N_2$. That is, the minimum of $\zeta(\hat{x})$ is achieved when $\hat{x}$ is in the midpoint or median of these data points.

Edge Extraction and 2-D High-pass Filtering (HPF)

Used to improve the visual appearance of images with less prominent edges. This is due to the fact that an image with accentuated edges is subjectively more pleasing to human eyes than an exact reproduction. This procedure is also referred to as "unsharp masking" in printing industry. Edge extraction is also an important part of typical image analysis systems that rely on contours or other edge-based features to discriminate different objects.

The basic idea is to detect the edges of an image using a HPF or a 2-D gradient operator. To sharpen or emphasize the edges, a fraction of this gradient or high-pass filtered image is then added to the original image. The gradient or high-pass image can also be generated by subtracting an unsharp or blurred or low-pass filtered image from the original image. In general, the unsharp masking operation can be represented by,

$$y(m,n) = x(m,n) + \lambda g(m,n),$$

$g(m,n)$ is a suitable gradient computed for the mask centered at (m,n) and $\lambda > 0$ is a proportionality constant. The pixel at (m,n) is an edge pixel if $g(m,n)$ exceeds a pre-specified threshold. This threshold can typically be decided by examining the histogram of the image $g(m,n)$.

The commonly used edge detection operators are the Roberts, Prewitt, Sobel and 2-D discrete Laplacian operators. These are briefly described below.

a. Roberts Operator

This simple operator is based upon using two 2x2 diagonal gradients $g_1(m,n) = x(m,n) - x(m+1,n+1)$ and $g_2(m,n) = x(m+1,n) - x(m,n+1)$ and combining the results using either absolute value $g(m,n) \equiv |g_1(m,n)| + |g_2(m,n)|$ or the square root operation as $g(m,n) = \{g_1^2(m,n) + g_2^2(m,n)\}^{1/2}$.

The corresponding masks are shown in Figures 7(a) and 7(b). The edge orientation with respect to the horizontal axis is given by $\theta(m,n) = tan^{-1}\frac{g_1(m,n)}{g_2(m,n)}$. Fig. 9(a) shows the result of applying the Roberts operator to the original Airplane image in Fig. 8. The major problem with this operator is its susceptibility to noise and small fluctuations in the image intensity. This is partly attributed to its small mask window size.

b. Prewitt and Sobel Operators

The Prewitt and Sobel operators are somewhat similar to the Roberts in the sense that they are based upon approximating the first derivatives and detecting edges with different orientations.

The results are typically combined using the square root operation. The 3x3 horizontal and vertical masks associated with these two operators are shown in Fig. 7. As can be seen, Sobel masks differ from those of Prewitt only by the way of the weights the north, south, west and east pixels are chosen. Figs. 9(b) and (c) show the edge detected images using these operators that are visually better than that of the Roberts in Fig. 9(a). The reason being the size of the masks in these cases are larger hence lesser sensitivity to noise and small perturbations.

c. Laplacian Operator

The Laplacian operator is based upon a 2-D discrete approximation of the second order derivatives of the image function. The idea is that although the first order derivative peaks where the transition or edges are located, the second order derivative will have a zero-crossing at these locations. The 3x3 mask in Figure 7(g) can be used to perform the 2-D Laplacian operation. The input/output relationship for this edge operator is $g(m,n) \equiv 4x(m,n) - [x(m-1,n) + x(m+1,n) + x(m,n-1) + x(m,n+1)]$.

Fig. 9(d) shows the result of applying the 3x3 Laplacian operator to the original "Airplane" image. The 2-D Laplacian operator is highly sensitive to noise and small spatial variations in a region even more so than the other operators. Additionally, due to the second derivative, this operator has a *double response* to edges.

Overall, these results indicate that among the gradient-based operators, Sobel performs better than the other ones. Also note that for high-pass filters we have, $\sum\sum_{k,l\in W} h(k,l) = 0$.

0	0	0
0	1	0
0	0	−1

(d) a

0	0	0
0	0	1
0	−1	0

(e) b

1	0	−1
1	0	−1
1	0	−1

(f) c

−1	−1	−1
0	0	0
1	1	1

(g) d

1	0	−1
2	0	−2
1	0	−1

(h) e

−1	−2	−1
0	0	0
1	2	1

(i) f

0	−1	0
−1	4	−1
0	−1	0

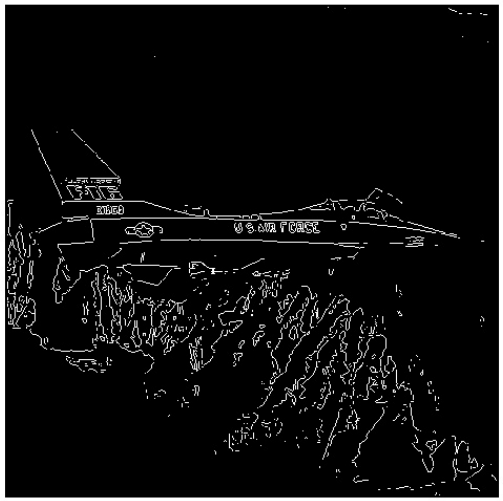
(j) g

Figure 7: Different Edge Detection Masks: (a) and (b) 2x2 Roberts Masks, (c) and (d) 3x3 Prewitt Masks, (e) and (f) 3x3 Sobel Masks (g) 3x3 Laplacian Mask.

Figure 8: Original Airplane

(a) Roberts

(b) Sobel



(c) Prewitt

(d) 3x3 Laplacian

Figure 9: Edge Maps for Four Edge Operators.

2-D Matched Filtering

In a number of applications such as motion and stereo correspondence and target detection it is often required to search for known objects or structures in the observed images. This 2-D "template matching" involves defining templates that closely represent the objects of interest to be detected and localized. The original image is then cross-correlated with the template. A match is found at the locations where the cross-correlation peaks and the amplitudes of the peaks exceed a chosen threshold. Since the cross-correlation operation is related to the convolution, the entire operation can be carried out in the spatial domain using spatial filtering with the specimen or templates, hence the term "matched filter".

Let us assume that a target template is denoted by $t(m,n), m,n \in W$ with W being the region of support of the template. Then, this template is spatially shifted by (k,l) and cross-correlated with the original image to yield

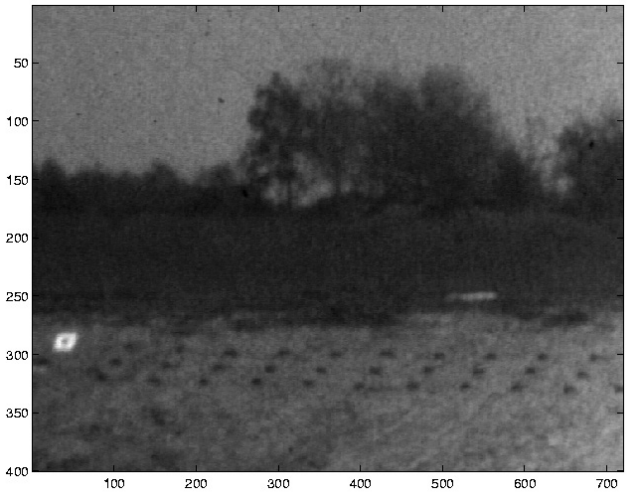
$$c_{x,t}(k,l) \triangleq \sum_m \sum_n x(m,n)t(m-k,n-l)$$

or

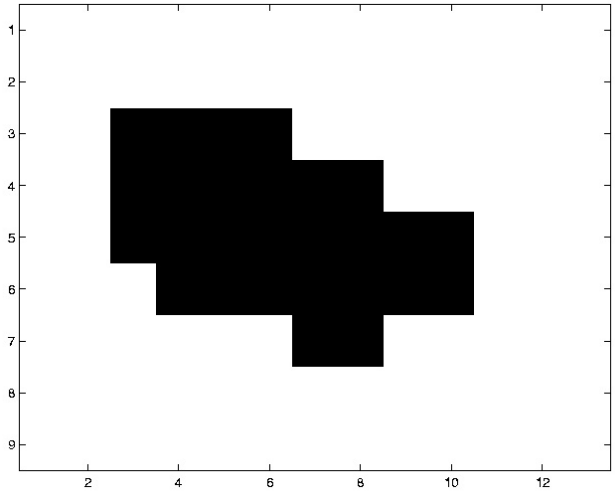
$$c_{x,t}(k,l) = x(k,l) * *t(-k,-l)$$

$c_{x,t}(k,l)$ is cross-correlation at the spatial location k,l and $**$ represents 2-D convolution. Clearly, the highest cross-correlation is attained when the unknown object in the image is a translated version of the template. However, in practice objects to be detected can be translated, scaled, rotated and distorted versions of the templates. In such cases, the standard template matching becomes less effective. A better approach is to employ feature-based matching methods.

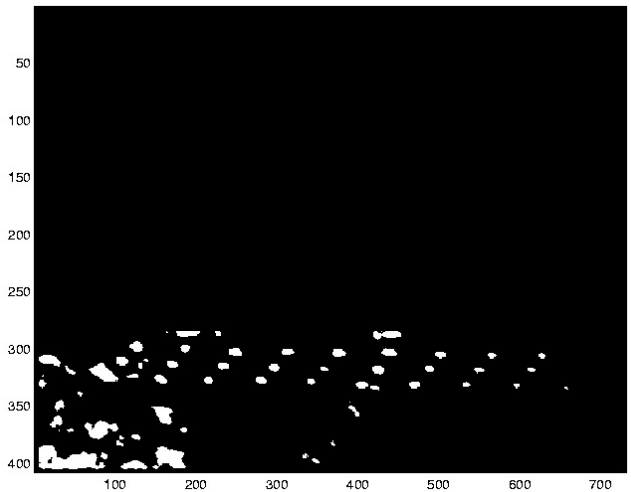
The minefield IR image in Fig. 10 is used here to detect the potential surface-laid targets. The image is first enhanced using histogram specification and then matched filtered with the chosen template in Fig. 10(b). The resultant image is shown in Fig. 10(c). Most targets are detected while the false alarm is kept to an acceptable level. Lowering the threshold will increase the detection rate, while at the same the incident of the false detections will also increase. It must be pointed out that a more careful choice of the template based upon some prior knowledge can substantially improve the detection rate while minimizing the false alarm rate.



(a) Original Minefield IR Image.



(b) Target Template.



(c) Matched Filtered Image.

Figure 10: Original and Matched Filtered Images of Minefield.

Point-Spatial Hybrid Operations

There are certain image processing tasks that involve the use of more than one operations. As an example, the combination of point and spatial operations can be useful in numerous image processing applications. An example is given below.

Multiplicative Noise Removal

In several applications such as synthetic aperture radar (SAR) imaging, the received images contain a multiplicative noise source known as "speckle". To remove speckle, one can use spatial filtering, after changing the multiplicative noise to additive one by taking the natural logarithm of the noisy image. Once the filtering is completed an inverse mapping (i.e. exponential) is then applied to convert the filtered images to the image domain.