Department of Computer Science
Kasetsart University

Lab 9 Image Restoration

Asst.Prof. Dr. Pakaket Wattuya

Comparison of using inverse and Wiener filter to restore the image degraded by motion blur and additive Gaussian noise.

Motion noisy image (len = 145 ; theta = 45)

Motion noisy image

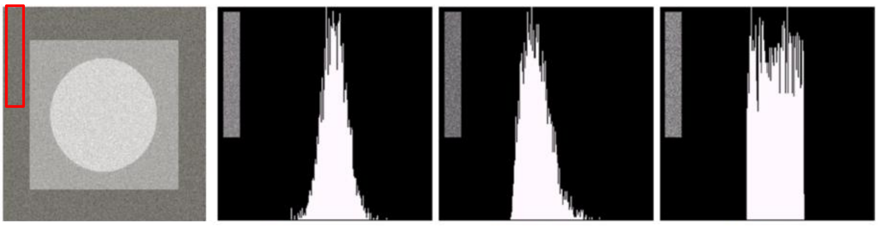
Image Restoration (Deconvolution)

MATLAB functions	Example
`J = deconvwnr(I,PSF)`	Deconvolves image I using the **Wiener filter** algorithm with no estimated noise. In the absence of noise, a Wiener filter is equivalent to an **ideal inverse filter**. • **PSF** is the point-spread function (PSF) with which I was convolved. `PSF = fspecial('motion', len, theta);`
`J = deconvwnr(I,PSF,NSR)`	Deconvolves image I using the **Wiener filter** algorithm, returning deblurred image J. • **NSR** is the noise-to-signal power ratio of the additive noise. $H_w(u,v) = \frac{H^*(u,v)}{\lvert H(u,v)\rvert^2 + \left[S_{vv}(u,v)/S_{ff}(u,v)\right]}$

MATLAB functions	Example
`J = imnoise(I,type,parameters)`	Add noise of a given type to the intensity image I.
`patch = roipoly(I)`	**Create Polygon Interactively** Displays the grayscale or RGB image I in a figure window and creates an interactive polygon selection tool associated with the image.
`noise_hist = imhist(I)`	Calculates the histogram for the grayscale image I
`v = statmoments(noise_hist,n)`	Computes up to the nth statistical central moment of a histogram whose components are in vector: v(1) = mean, v(2)= variance, v(3)= 3rd moment, ... v(n)= nth moment.

Estimate Noise-to-signal power ratio ($S_{vv}(u, v) / S_{ff}(u, v)$)

To estimate the parameters of the PDF from small patches of reasonably constant background intensity.



Gaussian Rayleigh Uniform

STEP 1: Estimate Noise Parameters by Image Observation

1.1 Select a part of image with reasonably constant background intensity.

```
patch = roipoly(g);
```

1.2 Compute histogram of the image patch.

```
noise_hist = imhist(g(patch));
imhist(g(patch)); % to show the histogram
```

1.3 Compute mean and variance of noise.

```
noise_stat = statmoments(noise_hist,2);
```

STEP 2: Compute the power spectrum of the noise ($S_{vv}(u, v)$)

2.1 Create a Gaussian noisy image with `noise_stat`

```
approx_noise = imnoise(zeros(size(g)),'gaussian',0, noise_stat(2));
```

2.2 Estimating $S_{vv}(u,v) = |N(u,v)|^2$, the power spectrum of the noise

```
Svv = abs(fft2(approx_noise)) .^ 2;
```

STEP 3: Compute $\hat{f}$ which is the estimation of ideal image f (for estimating $S_{ff}(u, v)$)

2.1 Estimate $\hat{f}$ which is the estimation of ideal image f using a Gaussian smoothing function

2.2 Estimating $S_{ff}(u,v) = |F(u,v)|^2$, the power spectrum of the undegraded image.

```
Sff = abs(fft2(double(f_hat))).^ 2;
```

STEP 4: Compute the noise-to-signal power ratio ($S_{vv}(u, v) / S_{ff}(u, v)$)

```
NSR = Svv ./ Sff;
```

STEP 5: Wiener filtering

```
PSF = fspecial('motion', len, theta);
im_out = deconvwnr(g, PSF, NSR);
```

Wiener filter :

$$H_w(u,v) = \frac{H*(u,v)}{|H(u,v)|^2 + [S_{vv}(u,v)/S_{ff}(u,v)]}$$