Tutorial 4 Image Restoration and Filtering

COMP 4421: Image Processing

March 1, 2016

Outline

- Noise Models
- Restoration in the Presence of Noise Only-Spatial Filtering
- Periodic Noise Reduction by Frequency Domain Filtering
- Restoration based on Degradation Function

Outline

- Noise Models
- Restoration in the Presence of Noise Only-Spatial Filtering
- Periodic Noise Reduction by Frequency Domain Filtering
- Restoration based on Degradation Function

Noise Models

- Gaussian Noise
- Rayleigh Noise
- Gamma Noise
- Exponential Noise
- Uniform Noise
- Impulse (Salt & Pepper) Noise

Noise Models

Matlab Code (add noise to image)

```
x=imread('camera.ras');
figure,imshow(x);
% salt & pepper noise added to image
y1=imnoise(x,'salt & pepper');
figure,imshow(y1);
y2=imnoise(x,'salt & pepper',0.2);
figure,imshow(y2);
g = imnoise(x, 'gaussian', 0, 0.01);
figure, imshow(g);
g = imnoise(x, 'gaussian', 0, 0.1);
figure, imshow(g);
```

Function imnoise( )

Function r = imnoise(f, type, parameters)

- Corrupt image f with noise specified in type and parameters
- Results returned in r
- Type include:

Value	Description
`'gaussian'`	Gaussian white noise with constant mean and variance
`'localvar'`	Zero-mean Gaussian white noise with an intensity-dependent variance
`'poisson'`	Poisson noise
`'salt & pepper'`	On and off pixels
`'speckle'`	Multiplicative noise

Salt & Pepper



Gaussian



Outline

- Noise Models
- Restoration in the Presence of Noise Only-Spatial Filtering
- Periodic Noise Reduction by Frequency Domain Filtering
- Restoration based on Degradation Function

Restoration in the Presence of Noise Only-Spatial Filtering

- Mean Filters
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contraharmonic mean filter
- Order-Statistics Filters
  - Median filter
  - Max and min filters
  - Midpoint filter
  - Alpha-trimmed mean filter
- Adaptive Spatial Filters
  - Adaptive, local noise reduction filter
  - Adaptive median filter

Restoration in the Presence of Noise Only-Spatial Filtering

Function f = spfilt(g, type, m, n, parameter)

- Performs spatial filtering
- Type include:
  - Mean Filters
    amean, gmean, hmean, chmean
  - Order-Statistics Filters
    median, max, min, midpoint, artimmed

Reference: Gonzalez R C, Woods R E, Eddins S L. Digital image processing using MATLAB[M]. Knoxville: Gatesmark Publishing, 2009.

Restoration in the Presence of Noise Only-Spatial Filtering

- Creating a pepper noise image

```
f = imread('lenna.jpg');
R = imnoise2('salt & pepper', M, N, 0.1, 0);
c = find(R == 0);
gp = f;
gp(c) = 0;
figure, imshow(gp);
```

- Filtering

```
fp = spfilt(gp, 'amean', 5, 5);
fpm = spfilt(gp, 'median', 5, 5);
fpmax = spfilt(gp, 'max',2,2);
figure,imshow(fp);
figure,imshow(fpm);
figure,imshow(fpmax);
```

Function imnoise2( )

Function R = imnoise2(type, M,N,a,b);

- Generates R of size M-by-N
- ‘a‘ and ‘b‘ are parameters
- Type include:
  Gaussian Noise; Rayleigh Noise; Gamma Noise; Exponential Noise; Uniform Noise; Salt & Pepper Noise; Lognormal Noise

Reference: Gonzalez R C, Woods R E, Eddins S L. Digital image processing using MATLAB[M]. Knoxville: Gatesmark Publishing, 2009.

Example

Image corrupted by pepper noise with probability 0.1. Result images with arithmetic mean filter, median filter, max filter respectively.

Periodic Noise Reduction by Frequency Domain Filtering

- Bandreject Filters
- Bandpass Filters
- Notch Filters
- Optimum Notch Filtering

Restoration based on Degradation Function

- Inverse filtering
- Minimum mean square error (Wiener) filtering
- Geometric mean filter

Degradation Model

- Spatial domain
  $g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$
- Frequency domain
  $G(x,y) = H(x,y)F(x,y) + N(x,y)$
- Problem Definition
  - Observe the degraded image $g(x,y)$
  - H function and N (noise) are either known or need to be estimated
  - Goal: restore the original image $f(x,y)$

Matlab built-in functions

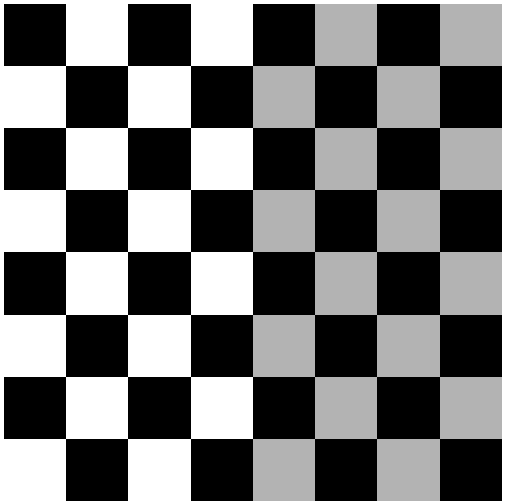
- PSF = fspecial('type', paras);
  - All kinds of spatial filters.
    Type includes 'average', 'disk', 'gaussian', 'motion'$\cdots$
  - Paras are corresponding parameters
  - PSF is the spatial domain filter created
  - Type help fspecial for more
- fr = deconvwnr(g, PSF, NSPR);
  - Wiener filtering: g is corrupted image, fr is restored one
  - PSF is the filter that degrades the image
  - NSPR: noise to signal power ratio
  - Type help deconvwnr for more

Inverse Filtering

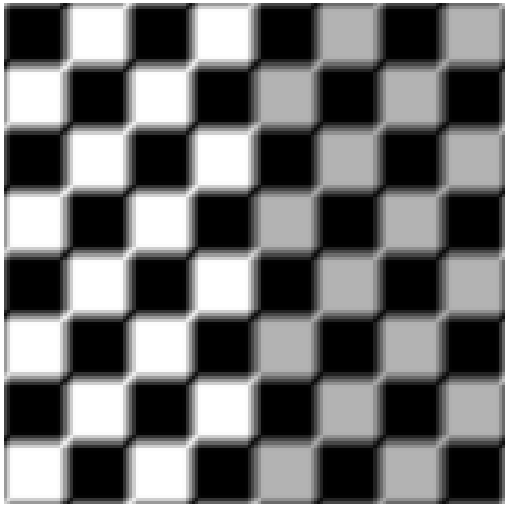
Matlab Code

```
f = checkerboard(20);
figure,subplot(1,3,1); imshow(f, [ ]); xlabel('original image')
PSF = fspecial('motion',7,45);
gb = imfilter(f,PSF,'circular');
subplot(1,3,2); imshow(gb, [ ]); xlabel('motion blurred')
fr = deconvwnr(gb,PSF);
subplot(1,3,3); imshow(fr, [ ]); xlabel('restored');
```

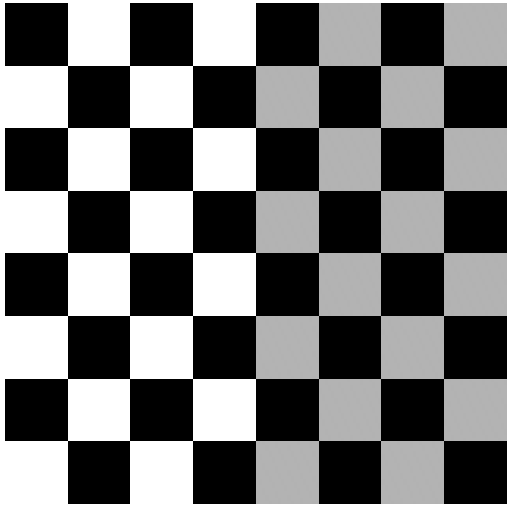
Inverse Filtering



original image



motion blurred

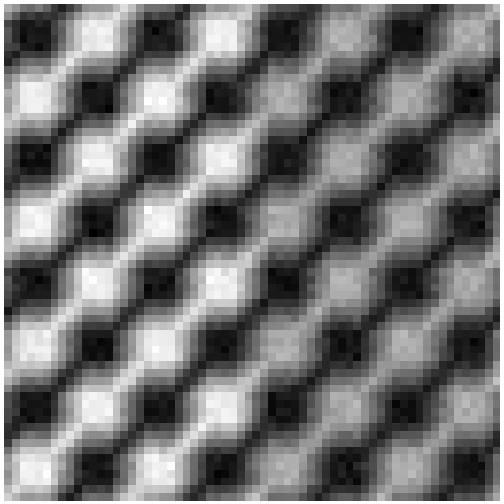


restored

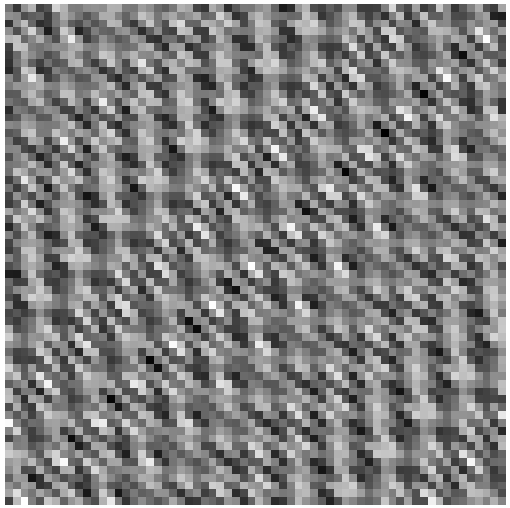
Wiener Filtering

```
f = checkerboard(8);
PSF = fspecial('motion',7,45);
gb = imfilter(f,PSF,'circular');
noise = normrnd(0,0.001^0.5,size(f));
gb = gb + noise;
figure,subplot(1,3,1); imshow(gb, [ ]); xlabel('noisy image');
fr1 = deconvwnr(gb, PSF);
subplot(1,3,2),imshow(fr1, [ ]); xlabel('inverse filtering');
Sn = abs(fft2(noise)).^2; %noise power spectrum
nA = mean(Sn(:)); % noise mean power spectrum;
Sf = abs(fft2(f)).^2; %image power spectrum
fA = mean(Sf(:)); %image mean power spectrum
K = nA/fA;
fr2 = deconvwnr(gb,PSF,K);
subplot(1,3,3); imshow(fr2,[]); xlabel('Wiener filtering');
```

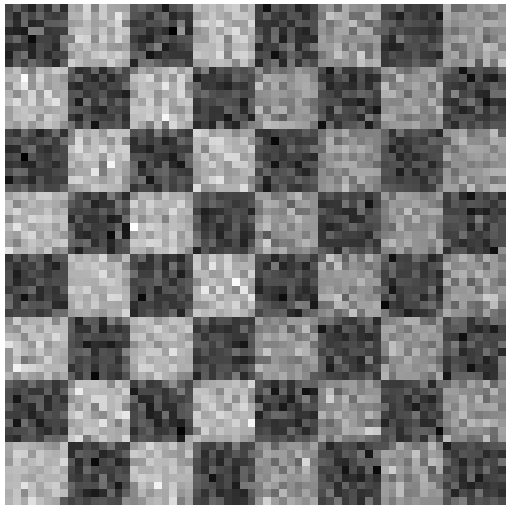
Wiener Filtering



noisy image



inverse filtering



Wiener filtering

Hints for Assignment 1

- Programming Section & Written Section
- Main routine: comp4421_assign1.m

Hints for Assignment 1

- Part 1: Create one gradient magnitude image. (10%)
- You need to complete the function in the file 'grad_mag_image.m'. The function 'grad_mag_image' takes a grayscale image of type uint8 as input and returns a gradient magnitude image of the same data type. $\text{mag}(\bigtriangledown f)$ at z_5 can be calculated with the following equation:

$$\begin{aligned}\text{mag}(\bigtriangledown f) \approx &|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + \\ &|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|\end{aligned}$$

- Hint: You can create one corresponding matrix Z_i for each element z_i in the mask. Then conduct operations on these matrices to create the gradient magnitude image.

Hints for Assignment 1

- Part 2: An additive Gaussian noise generator. (15%)
- You need to complete the implementation of an additive Gaussian noise generator in the gen_gauss_noise.m file. The routine gen_gauss_noise takes three numbers as input: size of an image along the X-axis and Y-axis, and the standard deviation of the Gaussian noise. You should generate an image of additive Gaussian noise with the given standard deviation. The data type of the output image should be double.
- Hint: You can find a MATLAB internal function to generate random numbers with Normal distribution, eg. randn() and normrnd()

Hints for Assignment 1

- Part 3: Arithmetic mean filter. (15%)
- You need to complete the implementation of the routine in the arithmetic_mean_filter.m file. The routine takes a noisy image (in grayscale format of type uint8) as input and attempts to remove the noise with an arithmetic mean filter. A 3×3 window is used to filter the noisy image. The output image type should be uint8.
- Hint: You may use the matlab internal function imfilter(), or calculate directly. imfilter(double(IM));

Arithmetic Mean Filter

- The arithmetic mean filtering process computes the average value of the corrupted image $g(x, y)$ in the area defined by S_{xy}.

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- This can also be implemented by using a mask with all the coefficients equal to $1/(mn)$.

Hints for Assignment 1

- Part 4:Wiener filtering with the power spectra of noise and un-degraded image. (15%)
- You need to complete the Wiener filter, suppose we know the power spectra of the noise S_η and the un-degraded image S_f. You need to complete the implementation in the wiener_filter_1.m file. The data type of the output image should be uint8. Do not use the internal MATLAB function 'wiener2' for implementation.
- Hint: $|H|^2 = conj(H).*H$

Wiener Filtering

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right] G(u,v)$$

Hints for Assignment 1

- Part 5:Wiener filtering with a constant K. (15%)
- You need to complete the Wiener filter, suppose we DO NOT know the power spectra of the noise S_η and the un-degraded image S_f. This routine employs the degradation function H_d (given in the main routine) and a constant $S \approx S_\eta / S_f$ to filter the given noisy image. You need to complete the implementation in the wiener_filter_2.m file. The data type of the output image should be uint8. Do not use the internal MATLAB function 'wiener2' for implementation.
- Hint: $|H|^2 = conj(H) .* H$

Wiener Filtering

- If neither N nor F is known, then the solution is to approximate

$$\frac{S_\eta(u,v)}{S_f(u,v)} = K$$

- The approximated, estimated image is given by

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

Thank you!