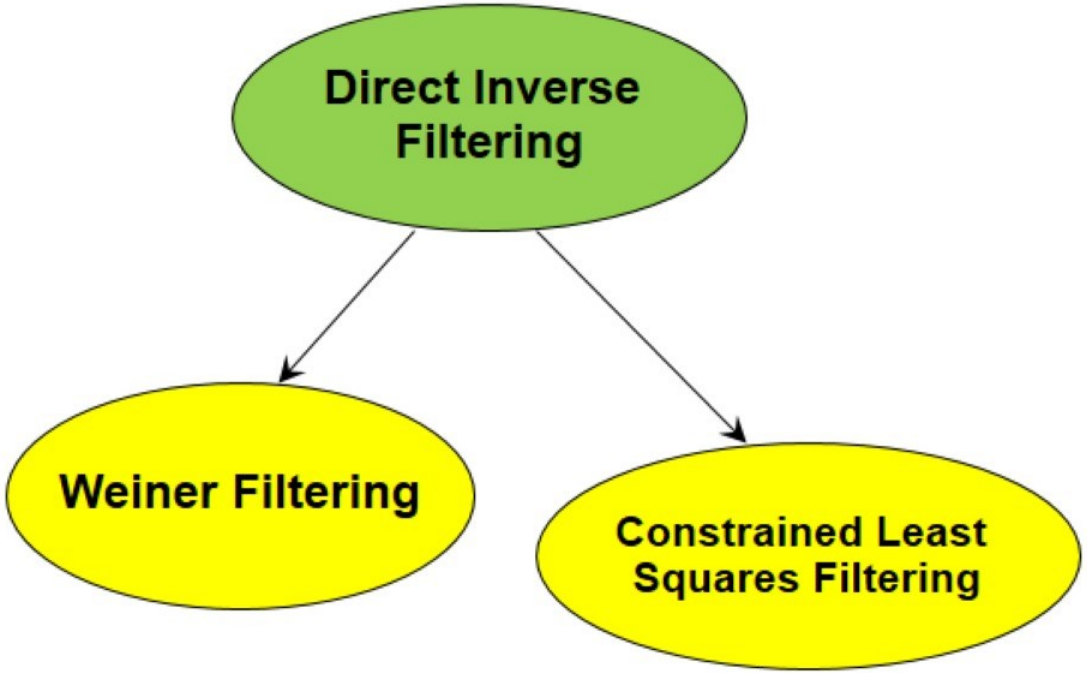
Image Processing

Image Restoration (Part II)

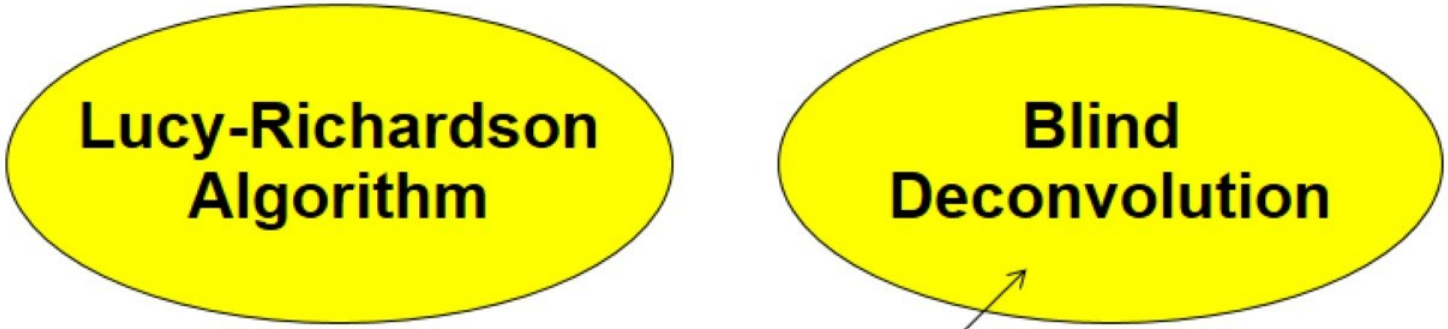
Pattern Recognition and Image Processing Laboratory (Since 2012)

Approaches for Image Restoration



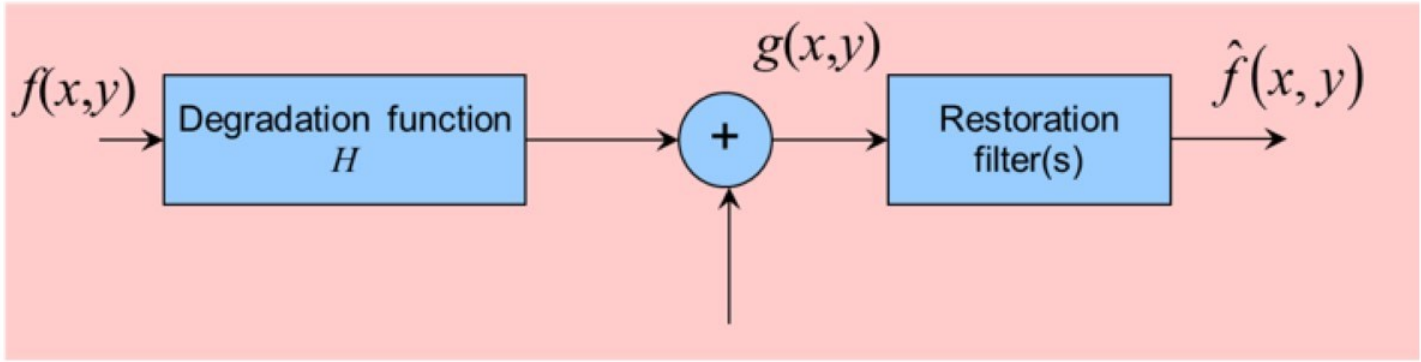
Note: - These approaches are linear image restoration.
- PSF (Point Spread Function) is available.

Approaches for Image Restoration



Note: - These approaches are nonlinear image restoration.
- PSF is NOT available.

Direct Invert Filtering



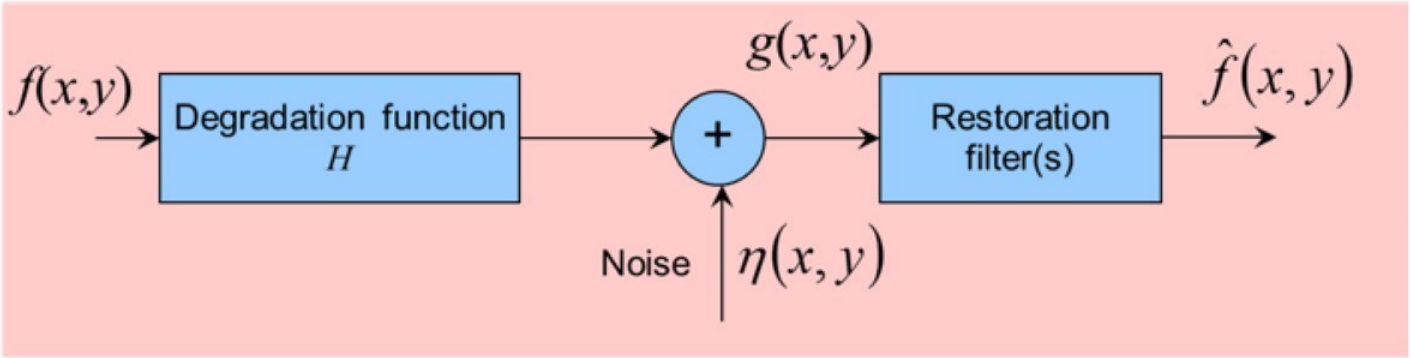
Degradation Eq

$$G(u,v) = H(u,v)F(u,v)$$

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Restoration Eq

Direct Invert Filtering



Degradation Eq

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Restoration Eq

Wiener Filtering

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

‘0’

$$\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}$$

Direct Inverse Filtering

Wiener Filtering

```
>> fr = deconwnr(g, PSF) % Direct Inverse Filter
>> fr = deconwnr(g, PSF, NSPR) % Parametric Weiner Filter
>> fr = deconwnr(g, PSF, NACORR, FACORR) % Weiner Filter with
                                          % Autocorrelation

>> degrad5_5 % See demonstration
```

Constrained Least Squares Filtering

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

'0'

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Direct Inverse F iltering

Iterative Nonlinear Restoration Using the Lucy-Richardson Algorithm

```
>> degrad5_9 % See demonstration
```

Blind Deconvolution

One of the most difficult problems in image restoration is obtaining a suitable estimation of the PSF to use in restoration algorithm.

Blind Deconvolution

Image restoration methods that are NOT based on specific knowledge of the PSF are called “blind deconvolution” algorithm.

Blind Deconvolution

```
>> degrad5_10  % See demonstration
```


The end of
part II

```
% degrad5_5.m

% ----------  Modeling the Degradation Function ----------------

clear all
close all

f = checkerboard(8);
figure, imshow(im2uint8(mat2gray(f)));

PSF = fspecial('motion', 7, 45);
gb = imfilter(f, PSF, 'circular');
figure, imshow(im2uint8(mat2gray(gb)));

noise = imnoise(zeros(size(f)), 'gaussian', 0, 0.001);
figure, imshow(im2uint8(mat2gray(noise)));

g = gb + noise;
%g = gb;

figure, imshow(im2uint8(mat2gray(g)));

figure, imshow(pixeldup(f, 8), []);

% --- Restoration by Using Direct Inverse Filter Technique
%     via deconvwnr function   ------------------------------

fr1 = deconvwnr(g, PSF);
figure, imshow(im2uint8(mat2gray(fr1)));

Sn = abs(fft2(noise)).^2;               % noise power spectrum
nA = sum(Sn(:))/prod(size(noise));      % noise average power
Sf = abs(fft2(f)).^2;                   % image power spectrum
fA = sum(Sf(:))/prod(size(f));          % image average power
R = nA/fA;

fr2 = deconvwnr(g, PSF, R);
figure, imshow(im2uint8(mat2gray(fr2)));

% --- Restoration by Using Autocorrelation function -----------

NCORR = fftshift(real(ifft2(Sn)));
ICORR = fftshift(real(ifft2(Sf)));
fr3 = deconvwnr(g, PSF, NCORR, ICORR);
figure, imshow(im2uint8(mat2gray(fr3)));
```

```
% ---- Restoration by Using Constrained Least Squares (Regularized)
%      Filtering    ---------------------------------------

fr4 = deconvreg(g, PSF, 2);            % noise power is
approximately 2
figure, imshow(im2uint8(mat2gray(fr4)));

fr5 = deconvreg(g, PSF, 0.4, [1e-7 1e7]);
figure, imshow(im2uint8(mat2gray(fr5)));
```

```
% degrad5_9.m

% ----------  Iterative Nonlinear Restoration Using the
%             Lucy-Richarson Algorithm  --------------------

clear all
close all

f = checkerboard(8);
figure, imshow(im2uint8(mat2gray(f)));
figure, imshow(pixeldup(f, 8), []);

PSF = fspecial('motion', 7, 10);
SD = 0.01;
g = imnoise(imfilter(f, PSF), 'gaussian', 0, SD^2);
figure, imshow(g);

DAMPAR = 10*SD;
LIM = ceil(size(PSF, 1)/2);
WEIGHT = zeros(size(g));
WEIGHT(LIM + 1:end - LIM, LIM + 1:end - LIM) = 1;
NUMIT = 20;
fr = deconvlucy(g, PSF, NUMIT, DAMPAR, WEIGHT);
figure, imshow(pixeldup(fr, 8))
figure, imshow(fr)
```

```
% degrad5_10.m

%  ------------ Blind Deconvolution -----------------

clear all
close all

f = checkerboard(8);
%figure, imshow(pixeldup(f, 8), []);
figure, imshow(f);

PSF = fspecial('gaussian', 5, 45);
%imshow(pixeldup(PSF, 73), []);
figure, imshow(PSF);
SD = 0.05;
g = imnoise(imfilter(f, PSF), 'gaussian', 0, SD^2);
figure, imshow(g);

INITPSF = ones(size(PSF));
NUMIT = 100;
DAMPAR = 10*SD;
LIM = ceil(size(PSF, 1)/2);
WEIGHT = zeros(size(g));
WEIGHT(LIM + 1:end - LIM, LIM + 1:end - LIM) = 1;
[fr, PSFs] = deconvblind(g, INITPSF, NUMIT, DAMPAR,
WEIGHT);
figure, imshow(fr);
```

Image Processing

Workshop on Image Restoration (Part II)

Pattern Recognition and Image Processing Laboratory (Since 2012)

Workshop on Image Restoration (Part II)

1. ให้เขียน MATLAB Script เพื่อสร้าง PSF ในรูปแบบของ “motion” อย่างน้อย 3 รูปแบบ
   โดยดูตัวอย่างได้จาก degrad5_5.m
2. ให้เขียน MATLAB Script เพื่อนำผลลัพธ์จากข้อ 1 ทั้ง 3 รูปแบบใส่ในภาพ lena.bmp
   โดยดูตัวอย่างได้จาก degrad5_5.m
3. ให้เขียน MATLAB Script เพื่อกู้คืนภาพจาก “motion blur” จากข้อ 2
   โดยดูตัวอย่างได้จาก degrad5_5.m