



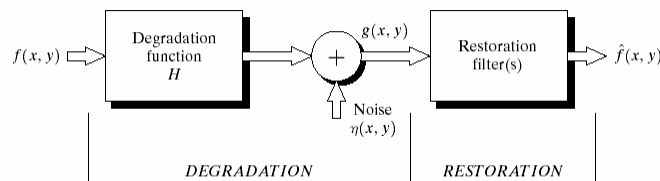
## بازیافت تصویر (Image Restoration)

- شباهت و هم پوشانی زیادی بین بحث بازیافت تصویر و بهبود کیفیت تصویر (Enhancement) وجود دارد. هدف بازیافت تصویر اولیه از تصویری است که در اثر عواملی کیفیت آن کاهش یافته است. به طور معمول در بازیافت تصویر اطلاعاتی در مورد نوع اعوجاج تصویر داشته و هدف مدلسازی آن و اعمال فرآیند معکوس برای برطرف کردن اثر آن است.
- روشهای بهبود کیفیت تصویر، سعی دارند تا با اعمال پردازشهایی و با در نظر گرفتن سیستم بینایی انسان، نمایش بهتری از تصویر را برای چشم فراهم کنند.
- افزایش کنتراست تصویر با افزایش محدوده دینامیکی شدت روشنایی جزو پردازشهای Enhancement و برطرف کردن اثر تار شدن تصویر با فیلتر Sharpening بازیافت تصویر محسوب می شود.



## بازیافت تصویر (Image Restoration)

- مدلسازی کاهش کیفیت تصویر (Degradation) و فرآیند بازیافت:
- همانطور که در شکل نشان داده شده است، عامل کاهش کیفیت تصویر بعنوان تابعی در نظر گرفته شده که به همراه نویز اضافه شونده، تصویر تخریب شده را تشکیل می دهند.



A model of the image degradation/ restoration process.

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$



## بازیافت تصویر (Image Restoration)

• با داشتن تصویر تخریب شده،  $g(x,y)$ ، و داشتن اطلاعاتی در مورد تابع تنزل دهنده کیفیت (Degradation function) یعنی  $H$  و نویز اضافه شونده، تخمینی از تصویر اولیه به دست می آید. هر قدر این اطلاعات بیشتر باشد، تصویر تخمینی به تصویر اولیه نزدیکتر خواهد بود.

• اگر  $H$  خطی و تغییر ناپذیر در حوزه مکانی باشد، تصویر تخریب شده در حوزه spatial با رابطه زیر به دست می آید:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

• تابع  $h(x,y)$  گاهی اوقات **PSF: Point Spread Function** نامیده می شود.

• مدل فوق در حوزه فرکانس با رابطه زیر بیان می شود:

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

• تابع  $H(u,v)$  گاهی اوقات **OTF: Optical Transfer Function** نامیده می شود.



## بازیافت تصویر (Image Restoration)

•  $h(x,y)$  و  $H(u,v)$  جفت تبدیل فوریه بوده و در جعبه ابزار پردازش تصویر، توابع **psf2otf** و **otf2psf** برای تبدیل آنها به یکدیگر استفاده می شوند.

• در این فصل ابتدا اثر تابع  $H$  را صرفنظر کرده و فقط اثر نویز را بررسی می کنیم. سپس اثر این تابع و نویز را با هم بررسی خواهیم کرد.

• مدل نویز

• در جعبه ابزار از تابع **imnoise** برای افزودن نویز به تصویر استفاده می شود:

$$g = \text{imnoise}(f, \text{type}, \text{parameters})$$

$F$  تصویر ورودی و  $\text{type}$  و  $\text{parameters}$  مشخصات نویز هستند. این تابع قبل از افزودن نویز به تصویر، آنرا به صورت **double** و به بازه  $[0 \ 1]$  تبدیل می کند. این مسئله در افزودن نویز بایستی در نظر گرفته شود. مثلاً به منظور افزودن نویز گوسی با میانگین  $m$  و واریانس  $\text{var}$  به تصویری با فرمت **uint8**، برای استفاده از تابع **imnoise** میانگین به **255** و واریانس به مربع آن تقسیم می شوند.



## بازیافت تصویر (Image Restoration)

• فرمتهای متداول در تابع imnoise:

$g = \text{imnoise}(f, \text{'gaussian'}, m, \text{var})$

نویز گوسی با میانگین  $m$  (پیش فرض صفر) و واریانس  $\text{var}$  (پیش فرض 0.01) به تصویر  $f$  اضافه می کند.

$g = \text{imnoise}(f, \text{'salt \& pepper'}, d)$

نویز فلل-نمکی با چگالی  $d$  که نشان دهنده درصد پیکسلهای تخریب شده است به تصویر  $f$  اضافه می کند.  $d * \text{numel}(f)$  تعداد پیکسلهای در معرض نویز قرار گرفته را نشان می دهد. مقدار پیش فرض آن 0.05 است.

$g = \text{imnoise}(f, \text{'speckle'}, \text{var})$

نویز متخلخل را که نویز ضربی است با رابطه  $g = f + f * n$  به تصویر  $f$  اضافه می کند. که  $n$  نویز تصادفی با توزیع یکنواخت و میانگین صفر و واریانس  $\text{var}$  است. مقدار پیش فرض آن 0.04 است.



## بازیافت تصویر (Image Restoration)

• نویز متناوب Periodic Noise:

• این نوع نویز عمدتاً از دستگاههای الکتریکی یا الکترو مکانیکی در زمان تصویر برداری (Image Acquisition) ایجاد می گردد. این نوع نویز اغلب در حوزه فرکانس حذف می گردد. پارامترهای این نوع نویز اغلب با تحلیل طیف تبدیل فوریه مشخص می گردند.

• بازیافت تصویر در حضور فقط نویز -حوزه مکان:

• با توجه به مدل کاهش کیفیت تصویر، فقط در حضور نویز داریم:

$$g(x, y) = f(x, y) + \eta(x, y)$$

• در این قسمت بازیافت تصویر با حضور نویز و در حوزه مکانی بررسی می گردد. تعدادی فیلتر مکانی برای کاهش نویز معرفی می گردد.



## بازیافت تصویر (Image Restoration)

❖ تعدادی از فیلتر های حذف نویز درحوزه مکان:  
Sxy نشان دهنده همسایگی m\*n در تصویر تخریب شده g است.

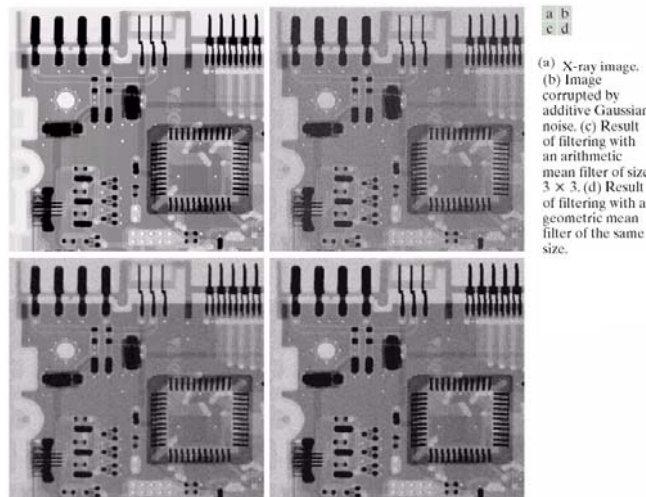
Filter Name	Equation	Comments
Arithmetic mean	$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$	Implemented using IPT functions $w = \text{fspecial}('average', [m, n])$ and $f = \text{imfilter}(g, w)$ .
Geometric mean	$\hat{f}(x, y) = \left[ \prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$	This nonlinear filter is implemented using function <code>gmean</code> (see custom function <code>spfilt</code> in this section).
Harmonic mean	$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s, t)}}$	This nonlinear filter is implemented using function <code>harmean</code> (see custom function <code>spfilt</code> in this section).
Contra-harmonic mean	$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q}$	This nonlinear filter is implemented using function <code>charmean</code> (see custom function <code>spfilt</code> in this section).
Median	$\hat{f}(x, y) = \text{median} \{g(s, t)\}_{(s,t) \in S_{xy}}$	Implemented using IPT function <code>medfilt2</code> : $f = \text{medfilt2}(g, [m, n])$ .
Max	$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\}$	Implemented using IPT function <code>ordfilt2</code> : $f = \text{ordfilt2}(g, m*n, \text{ones}(m, n))$ .
Min	$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$	Implemented using IPT function <code>ordfilt2</code> : $f = \text{ordfilt2}(g, 1, \text{ones}(m, n))$ .
Midpoint	$\hat{f}(x, y) = \frac{1}{2} \left[ \max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right]$	Implemented as 0.5 times the sum of the max and min filtering operations.
Alpha-trimmed mean	$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g(s, t)$	The $d/2$ lowest and $d/2$ highest intensity levels of $g(s, t)$ in $S_{xy}$ are deleted. $g(s, t)$ denotes the remaining $mn - d$ pixels in the neighborhood. Implemented using function <code>alphatrim</code> (see custom function <code>spfilt</code> in this section).

6/7/11

7



## بازیافت تصویر (Image Restoration)



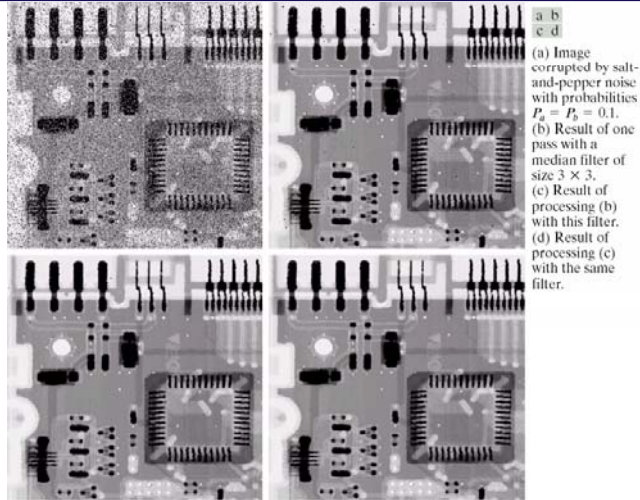
6/7/2011

Digital Image Processing - Seyedarabi

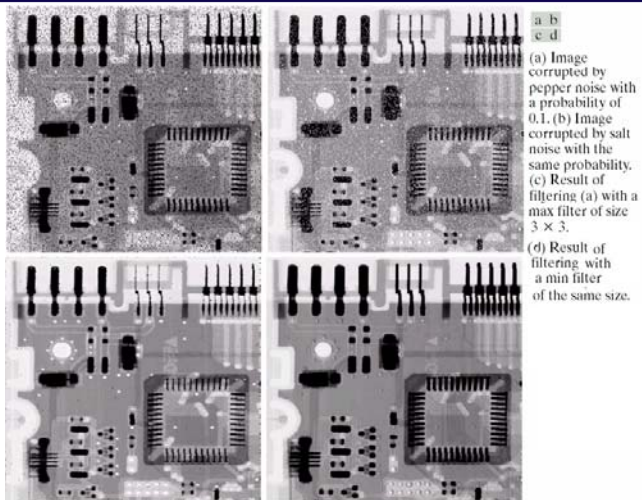
8



## بازیافت تصویر (Image Restoration)



## بازیافت تصویر (Image Restoration)

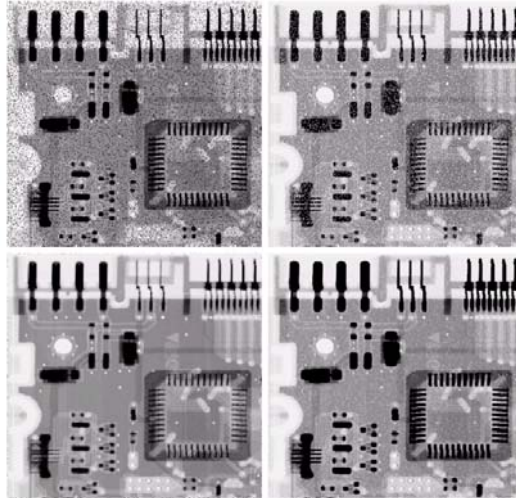




## بازیافت تصویر (Image Restoration)

a b  
c d

**FIGURE 5.8**  
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a  $3 \times 3$  contraharmonic filter of order 1.5. (d) Result of filtering (b) with  $Q = -1.5$ .



6/7/2011

Digital Image Processing - Seyedarabi

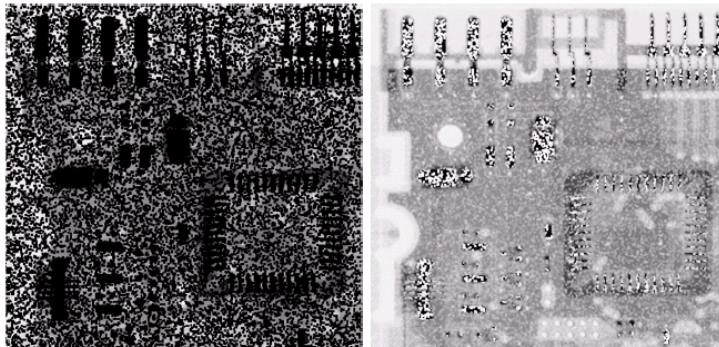
11



## بازیافت تصویر (Image Restoration)

a b

**FIGURE 5.9** Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size  $3 \times 3$  and  $Q = -1.5$ . (b) Result of filtering 5.8(b) with  $Q = 1.5$ .



6/7/2011

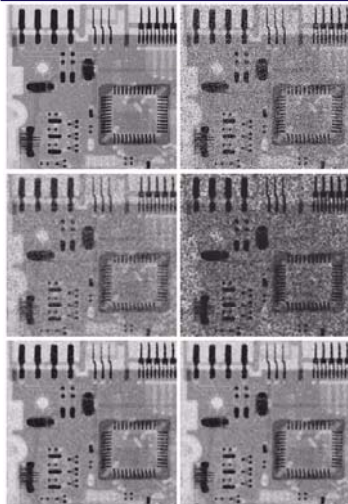
Digital Image Processing - Seyedarabi

12





## بازیافت تصویر (Image Restoration)



**FIGURE 5.12** (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a  $5 \times 5$ . (c) arithmetic mean filter, (d) geometric mean filter, (e) median filter, and (f) alpha-trimmed mean filter with  $d = 5$ .



## بازیافت تصویر (Image Restoration)

### • فیلتر notch :

• نویزهای پریودی که چون دارای طیف محدودی هستند، برای حذف این نوع نویز و جلوگیری از آسیب به طیف سیگنال اصلی می توان از فیلتر notch استفاده کرد. این فیلتر طوری طراحی می گردد که در حالت ایده آل، فقط یک فرکانس یا محدوده فرکانسی کوچکی را که مربوط به ناحیه فرکانسی نویز پریودیک است حذف کرده و بقیه را عبور دهد.

• فیلتر notch ایده آل:

$$H(u, v) = \begin{cases} 0 & \text{if } D_1(u, v) \leq D_0 \text{ or } D_2(u, v) \leq D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u, v) = [(u - M/2 - u_0)^2 + (v - N/2 - v_0)^2]^{1/2}$$

$$D_2(u, v) = [(u - M/2 + u_0)^2 + (v - N/2 + v_0)^2]^{1/2}$$



## بازیافت تصویر (Image Restoration)

• فیلتر notch باترورث:

$$H(u, v) = \frac{1}{1 + \left[ \frac{D_0^2}{D_1(u, v)D_2(u, v)} \right]^n}$$

• فیلتر notch گوسی:

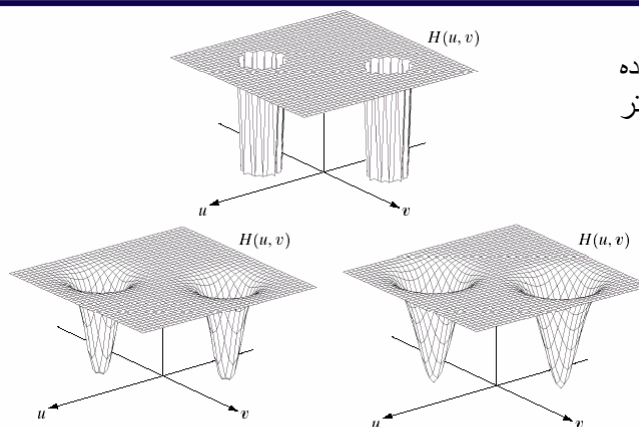
$$H(u, v) = 1 - e^{-\frac{1}{2} \left[ \frac{D_1(u, v)D_2(u, v)}{D_0^2} \right]}$$

- $(u_0, v_0)$  و به طور متقارن  $(-u_0, -v_0)$  محل notch ها هستند.  $D_0$  نیز شعاع آنهاست.
- اگر  $u_0=v_0=0$  باشد هر سه فیلتر فوق تبدیل به فیلتر بالا گذر می گردند.



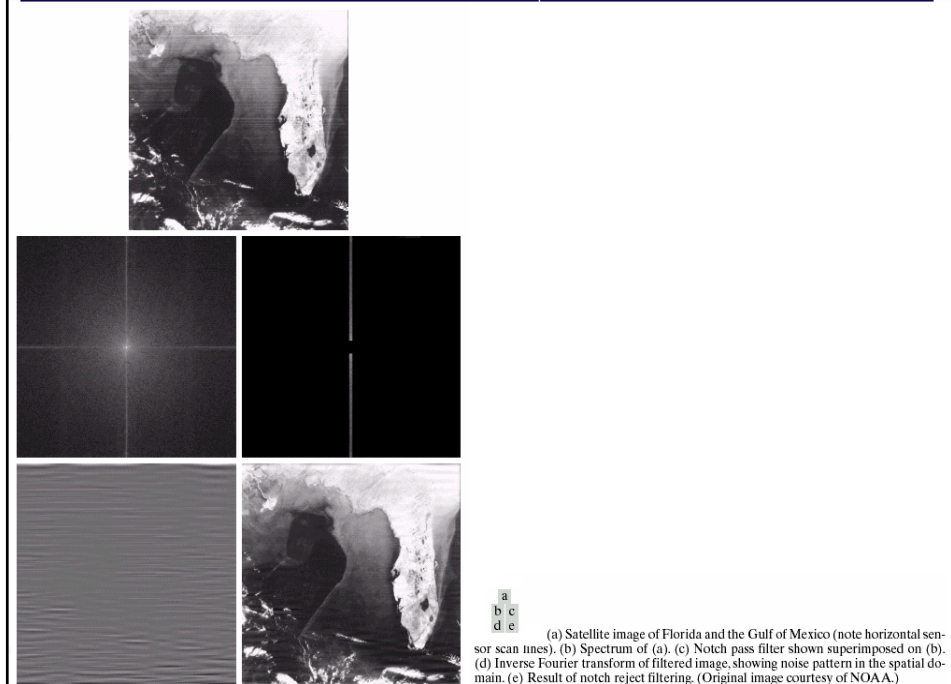
## بازیافت تصویر (Image Restoration)

• با توجه به شکل، استفاده از تابع `fftshift` در فیلتر کردن ضروری است.



a Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian  
b c notch (reject) filters.





## بازیافت تصویر (Image Restoration)

### • مدلسازی تابع Degradation:

• یکی از عوامل کاهش کیفیت تصویر، تار شدن تصویر و به خصوص در اثر حرکت سنسور در مرحله تصویر برداری است (Motion Blur).

• در محیط Matlab با دستور زیر می توان PSF آنرا به دست آورد:

```
PSF = fspecial('motion', len, theta)
```

• در این رابطه len طول برحسب پیکسل (پیش فرض 9) و theta زاویه حرکت بر حسب درجه (پیش فرض صفر) است.

• با دستور زیر می توان اثر Motion Blur را با PSF به دست آمده شبیه سازی کرد:

```
g = imfilter(f, PSF, 'circular');
```

• مدل Degradation را با افزودن نویز می توان تکمیل کرد:

```
g = g + noise;
```

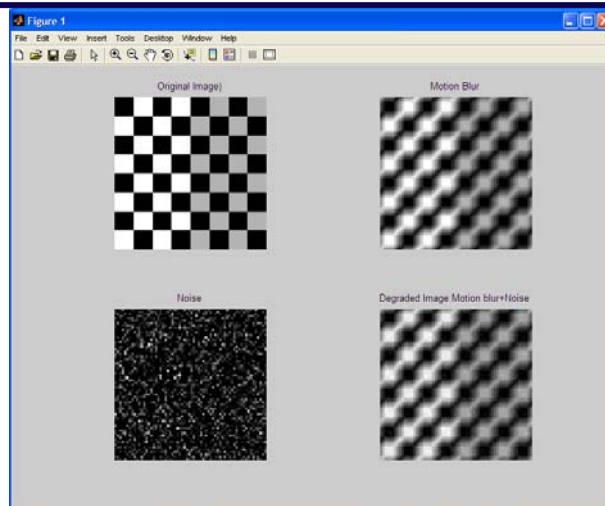


## بازیافت تصویر (Image Restoration)

```
f=checkerboard(8);
PSF=fspecial('motion',7,45);
gb=imfilter(f,PSF,'circular');
noise=imnoise(zeros(size(f)),'gaussian',0,.001);
g=gb+noise;
subplot(2,2,1);imshow(f,[]);title('Original Image');
subplot(2,2,2);imshow(gb,[]);title('Motion Blur');
subplot(2,2,3);imshow(noise,[]);title('Noise');
subplot(2,2,4);imshow(g,[]);title('Degraded Image:Motion
blur+Noise');
```



## بازیافت تصویر (Image Restoration)





## بازیافت تصویر (Image Restoration)

• روش فیلتر کردن معکوس:

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

• اساس این روش رابطه مقابل است:

در این رابطه با تقسیم تبدیل فوریه تصویر کاهش کیفیت یافته بر تبدیل فوریه تابع Degradation و گرفتن عکس تبدیل فوریه تخمینی از تصویر به دست می آید.

• این روش حتی اگر تابع Degradation به طور کامل مشخص باشد، نمی تواند تصویر اولیه را به طور کامل آشکار کند. چرا که نویز ماهیت تصادفی داشته و تابع آن قابل پیش بینی نیست.

• فیلتر ویلر (Wiener Filtering)

• این روش یکی از بهترین و قدیمی ترین روشها برای تخمین تصویر اولیه است.  
• در این روش تخمینی از تصویر به دست می آید که تابع خطای آماری زیر را کمینه

$$e^2 = E\{(f - \hat{f})^2\}$$

کند:



## بازیافت تصویر (Image Restoration)

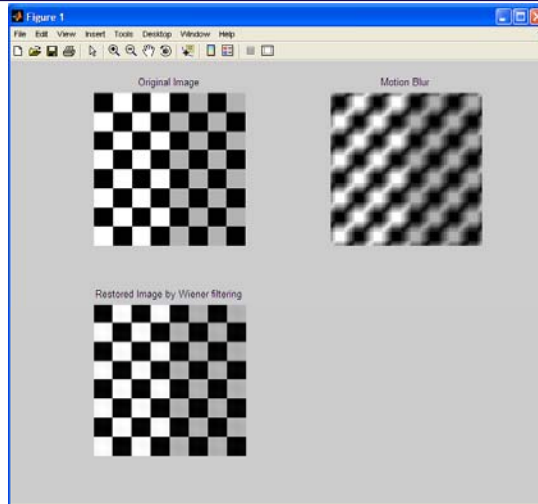
• در محیط Matlab با دستور زیر می توان تصویر را با داشتن PSF بازسازی کرد:  
مثال: `fr=deconvwnr(g,PSF)`

```
f=checkerboard(8);
PSF=fspecial('motion',7,45);
g=imfilter(f,PSF,'circular');
fr=deconvwnr(g,PSF);
subplot(2,2,1);imshow(f,[]);title('Original Image');
subplot(2,2,2);imshow(g,[]);title('Motion Blur');
subplot(2,2,3);imshow(fr,[]);title('Restored Image by Wiener filtering');
```

• در این مثال اثر نویز در نظر گرفته نشده است.



## بازیافت تصویر (Image Restoration)



## بازیافت تصویر (Image Restoration)

### • Blind Deconvolution

• یکی از مشکل ترین بحث ها در بازیافت تصویر، تخمین مناسب PSF هنگامی که معلوماتی در مورد آن نداریم می باشد. این روش Blind Deconvolution نامیده می شود.

### • تبدیلات هندسی:

• تابع `imtransform` و `maketform` برخی تبدیلات هندسی روی تصویر مانند تغییر مقیاس، دوران و ... اعمال می کنند.



## بازیافت تصویر (Image Restoration)

Type	Affine Matrix, T	Coordinate Equations	Diagram
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = w$ $y = z$	
Scaling	$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = s_x w$ $y = s_y z$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = w\cos\theta - z\sin\theta$ $y = w\sin\theta + z\cos\theta$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = w + \alpha z$ $y = z$	
Shear (vertical)	$\begin{bmatrix} 1 & \beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = w$ $y = \beta w + z$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \delta_x & \delta_y & 1 \end{bmatrix}$	$x = w + \delta_x$ $y = z + \delta_y$	

جدول زیر تبدیلات تعریف شده در تابع `maketform` و `imtransform` را نشان می دهد:

6/7/2011

Digital Image Processing - Seyedarabi

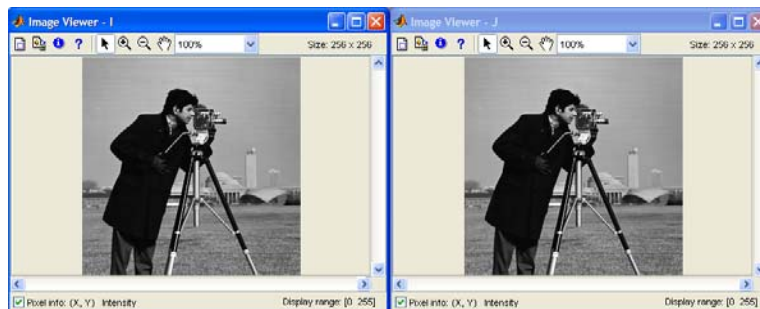
25



## بازیافت تصویر (Image Restoration)

Identity•

```
I = imread('cameraman.tif');
tform = maketform('affine',[1 0 0; 0 1 0; 0 0 1]);
J = imtransform(I,tform);
imview(I), imview(J)
```



6/7/2011

Digital Image Processing - Seyedarabi

26



## بازیافت تصویر (Image Restoration)

### Scaling •

```
I = imread('cameraman.tif');
tform1 = maketform('affine',[.5 0 0; 0 .5 0; 0 0 1]);
tform2 = maketform('affine',[2 0 0; 0 2 0; 0 0 1]);
J = imtransform(I,tform1);
K = imtransform(I,tform2);
imview(I), imview(J),imview(K)
```



6/7/2011

Digital Image Processing - Seyedarabi

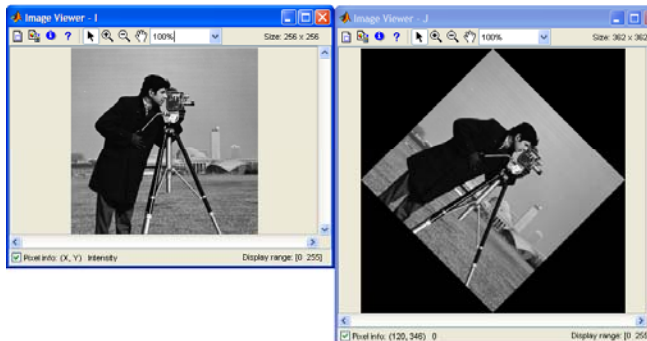
27



## بازیافت تصویر (Image Restoration)

### Rotation •

```
I = imread('cameraman.tif');
tform = maketform('affine',[.707 .707 0; -.707 .707 0; 0 0 1]);
J = imtransform(I,tform);
imview(I), imview(J)
```



6/7/2011

Digital Image Processing - Seyedarabi

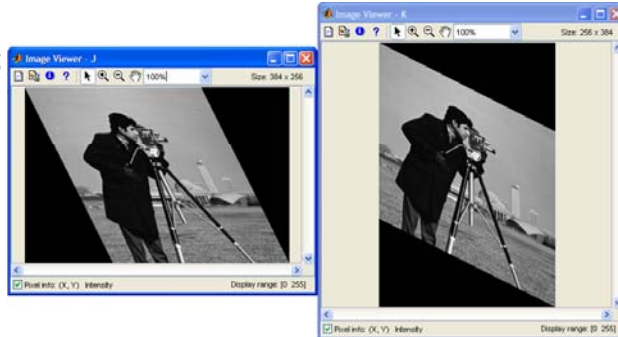
28



## بازیافت تصویر (Image Restoration)

### Vertical and Horizontal Shear •

```
I = imread('cameraman.tif');
tform1 = maketform('affine',[1 0 0;.5 1 0; 0 0 1]);
tform2 = maketform('affine',[1 .5 0;0 1 0; 0 0 1]);
J = imtransform(I,tform1);
K = imtransform(I,tform2);
imview(J), imview(K)
```



6/7/2011

Digital Image Processing - Seyedarabi

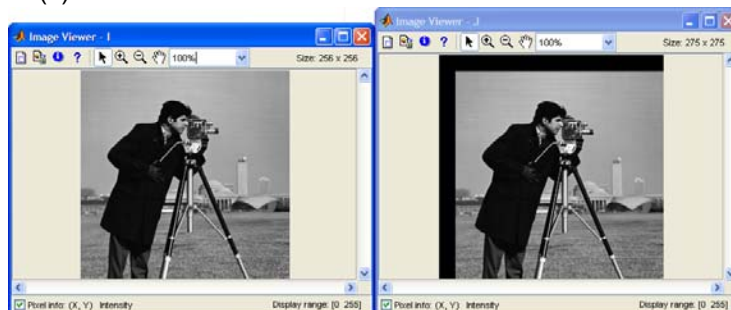
29



## بازیافت تصویر (Image Restoration)

### Translation •

```
I = imread('cameraman.tif');
tform = maketform('affine',[1 0 0;0 1 0; 20 20 1]);
J = imtransform(I,tform,'XData',[1,275],'YData',[1,275]);
imview(I), imview(J)
```



6/7/2011

Digital Image Processing - Seyedarabi

30