Image restoration

Image **restoration** techniques aim at modelling a degradation corrupting the image and inverting this degradation to correct the image so that it is as close as possible to the original.

© IE PŁ

Image restoration vs. image enhancement

Image enhancement

↓

Heuristic criteria; there is no image degradation model constructed, no mathematical criteria used

Image restoration

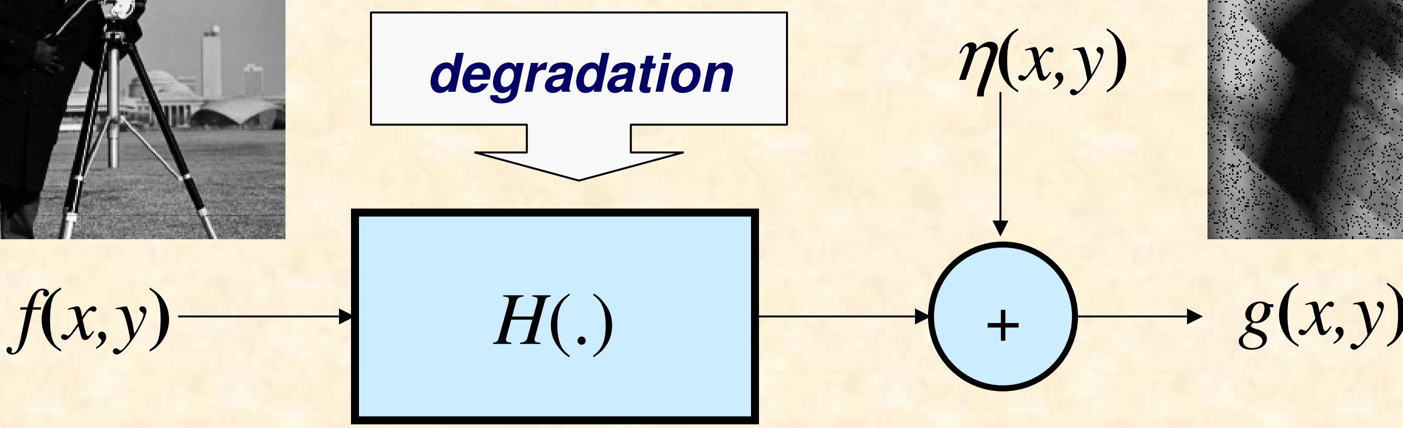
↓

Quantitative criterion used; image model and degradation model required

Image degradation model

Source image

Degraded image



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

Image degradation model

Assume $H(.)$ is linear and shift invariant:

$$g(x,y)=\int_{-\infty}^{\infty}\int f(\alpha,\beta)h(x-\alpha,y-\beta)\,d\alpha\,d\beta+\eta(x,y)$$

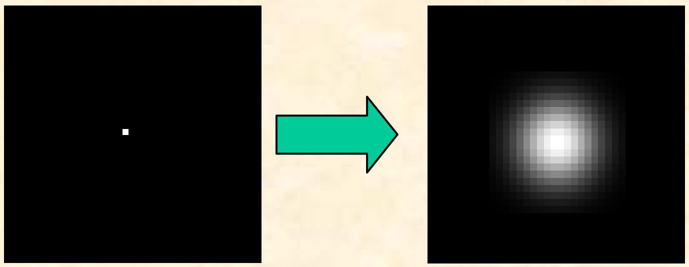
thus for the spectrum domain:

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

Estimating the degradation function

How to estimate the degradation function?

- build a mathematical model of the degradation (example given),
- reproduce the degradation process on a known image,



Degradation estimation example

Assume $f(x,y)$ is linearly shifted during exposure and $x_0(t)$, $y_0(t)$ are coordinates of motion. The degraded image is given by:

$$g(x,y) = \int_0^T f\left(x - x_0(t), y - y_0(t)\right) dt$$

Fourier transform of the degraded image:

$$G(u,v) = \int_{-\infty}^{\infty}\int g(x,y)\, exp\left[-j2\pi(ux + vy)\right] dx\, dy$$

Image restoration example

Hence:

$$G(u,v)=\int_{-\infty}^{\infty}\int\left[\int_0^T f\left(x-x_0(t),\, y-y_0(t)\right)dt\right]\exp\left[-j2\pi\left(ux+vy\right)\right]dx\,dy$$

By inverting the order of integration and using Fourier transform property of the displaced function :

$$G(u,v)=F(u,v)\cdot\underbrace{\int_0^T \exp\left[-j2\pi\left(ux_0(t)+vy_0(t)\right)\right]dt}_{H(u,v)}$$

Image restoration example

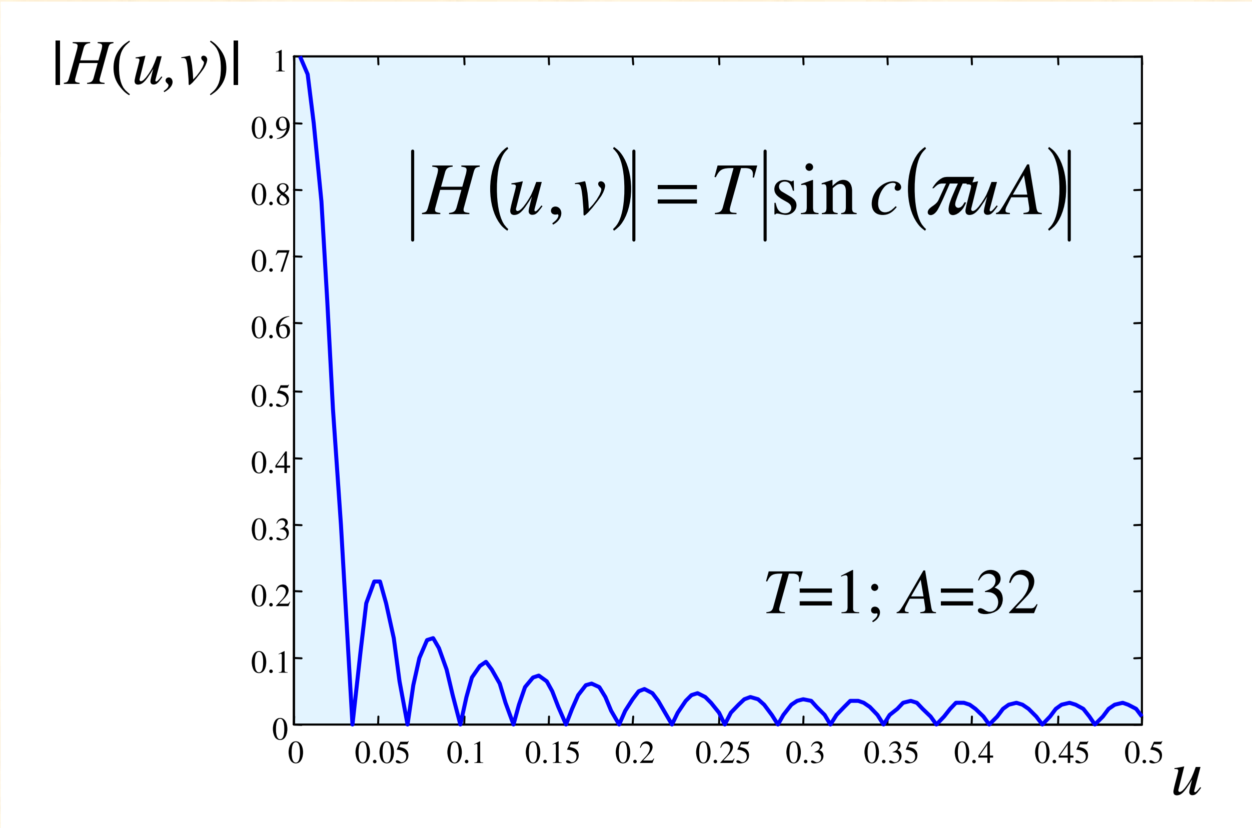
Assume the motion is parallel to X axis and it is a liner, uniform motion, i.e. $x_0 = At/T$ and $y_0 = 0$:

$$H(u,v) = \int_0^T \exp[-j2\pi u x_0(t)]dt = \int_0^T \exp\left[-\frac{j2\pi u A t}{T}\right]dt =$$

$$= \frac{T}{\pi u A}\sin(\pi u A)e^{-j\pi u A}$$

Note that $H(u,v)$ has zeros (!) for $u = i/A$, where i is an integer number.

Image restoration example



Zeros: $u=i/A=0.313, 0.626, \ldots..$

Image restoration example

$f(x,y)$

Degradation

$g(x,y)$

Source image
N*N =256*256

Degraded image,
shift: A=1/8*N=32

Inverse filtering

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

Spectrum of the source image can be estimated from:

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \cdot G(u,v) = F(u,v) + \frac{1}{H(u,v)} \cdot N(u,v)$$

Inverse filter

Numerical instability for u, v for which $H(u,v) \approx 0$; i.e., for zeros of $H(u,v)$ and frequencies for which $N(u,v) > F(u,v)$.

The Wiener filter

The Wiener filter is given by:

$$\hat{F}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma \dfrac{S_\eta(u,v)}{S_f(u,v)}} \cdot G(u,v) = W(u,v) \cdot G(u,v)$$

where: $S_f(u,v)$ - image power spectral density,
$S_\eta(u,v)$ - noise power spectral density.

$$|H(u,v)|^2 = H^*(u,v) \cdot H(u,v)$$

The Wiener filter

The parametric Wiener filter:

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma \frac{S_\eta(u,v)}{S_f(u,v)}} \cdot G(u,v)$$

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

- γ adjustable factor,
- for $S_\eta(u,v) \approx 0$ or $K \rightarrow 0$ the Wiener filter is an inverse filter,
- for $K \rightarrow \infty$ the Wiener filter becomes a low-pass filter.

The Wiener filter

If $S_f(u,v)$ and $S_\eta(u,v)$ are unknown, the Wiener filter can be approximated by the formula:

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

where $K \in \boldsymbol{R}^+$.

In practice, $K \sim \sigma^2$, where σ^2 is the noise variance that is easy to estimate.

Wiener filter spectral characteristics

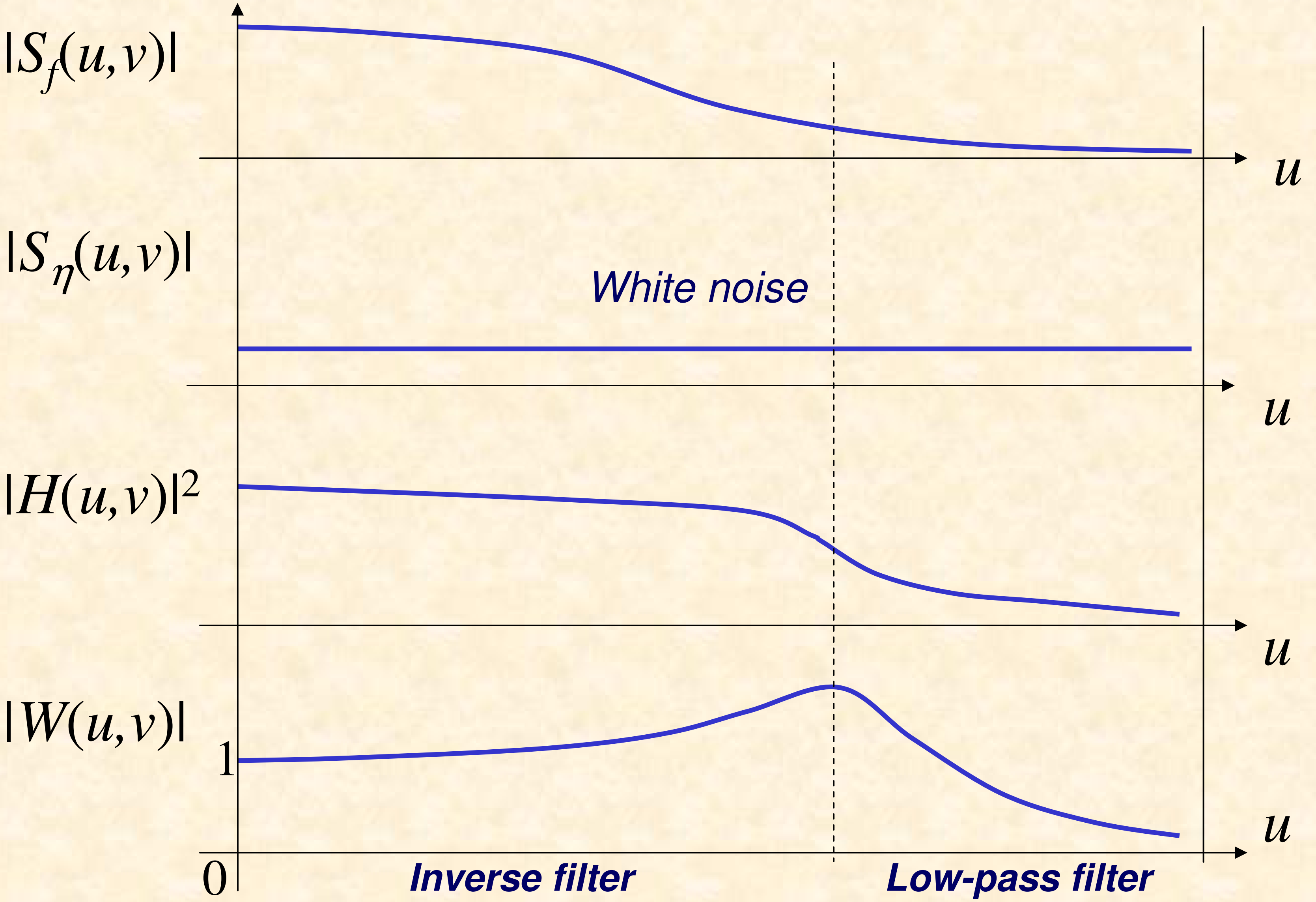
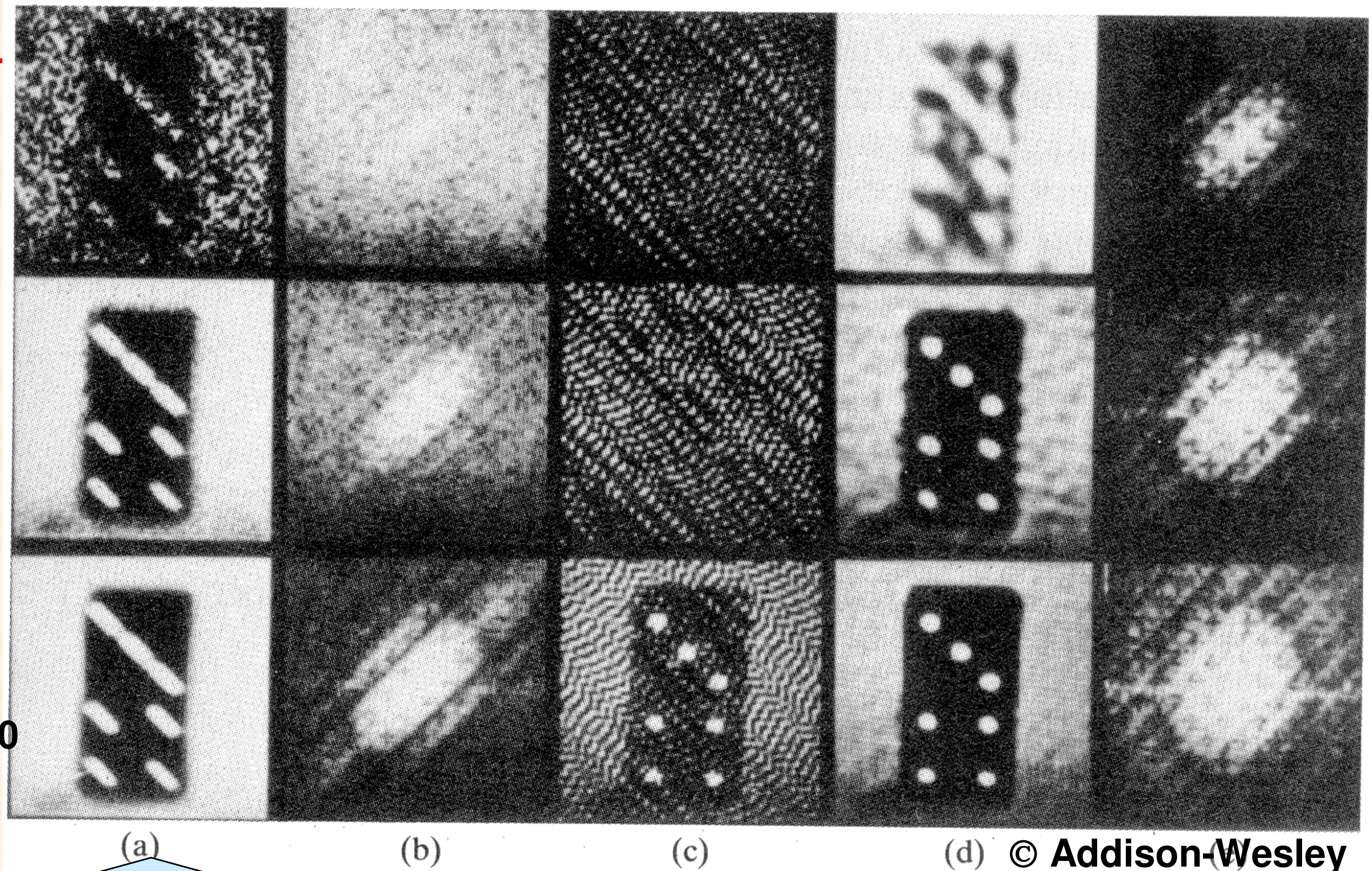


Image restoration examples

SNR=1

SNR=10

SNR=100



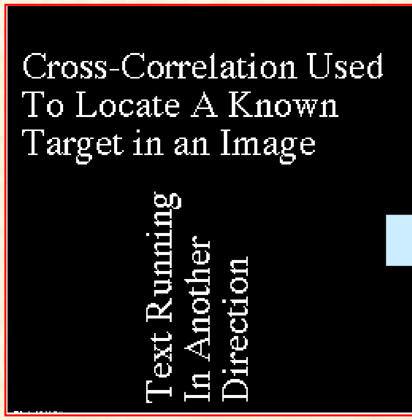
Degraded images

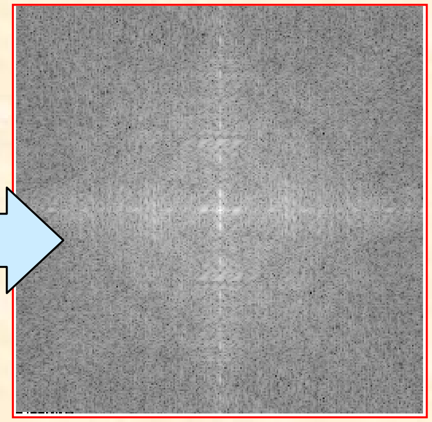
Inverse filtering

Wiener filtering

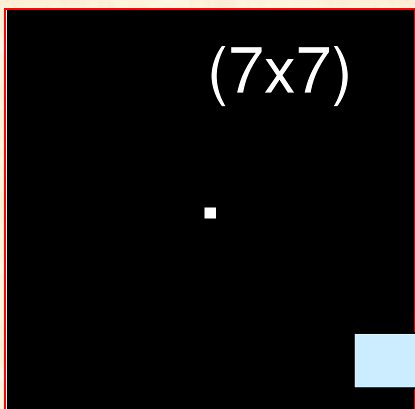
Image restoration examples

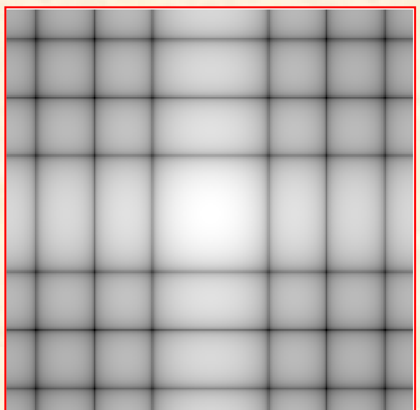
An image and its spectrum



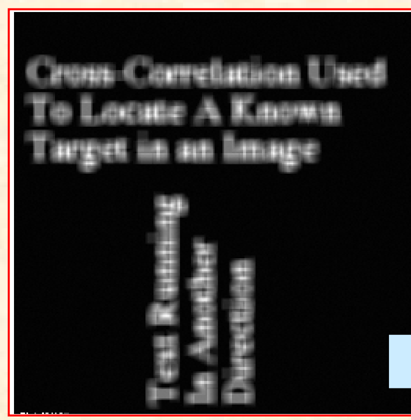


The filter and its spectrum





Degraded image and its spectrum



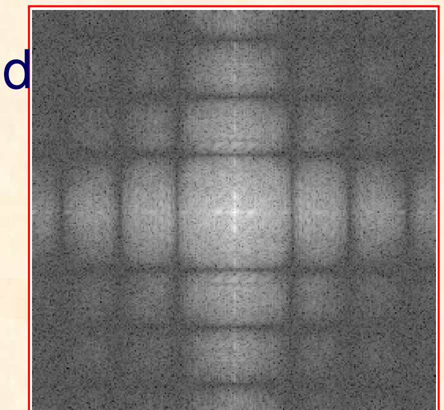
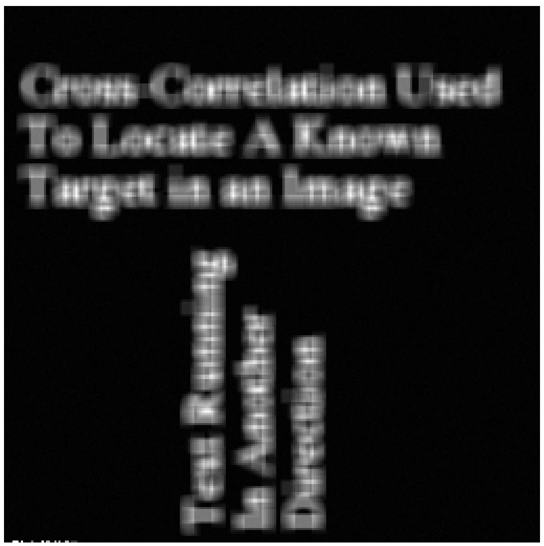
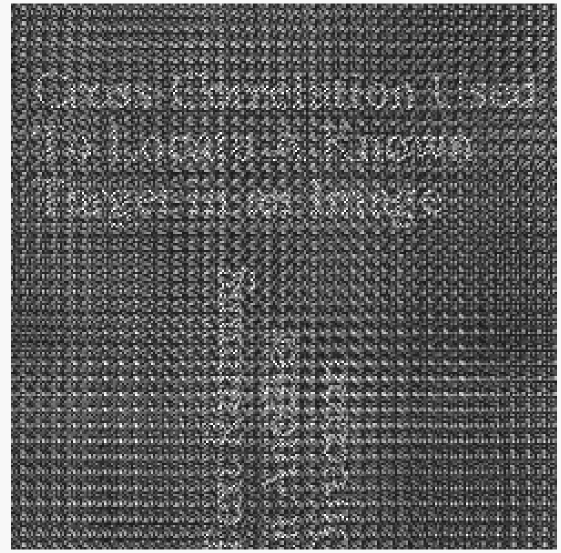


Image restoration examples

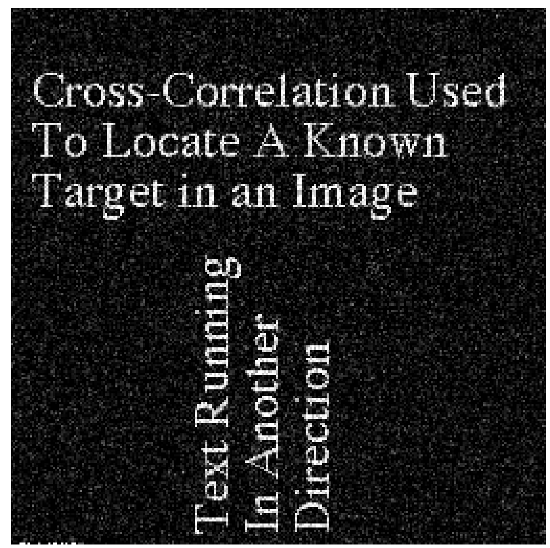


Degraded image
(blur+noise)

Inverse filter



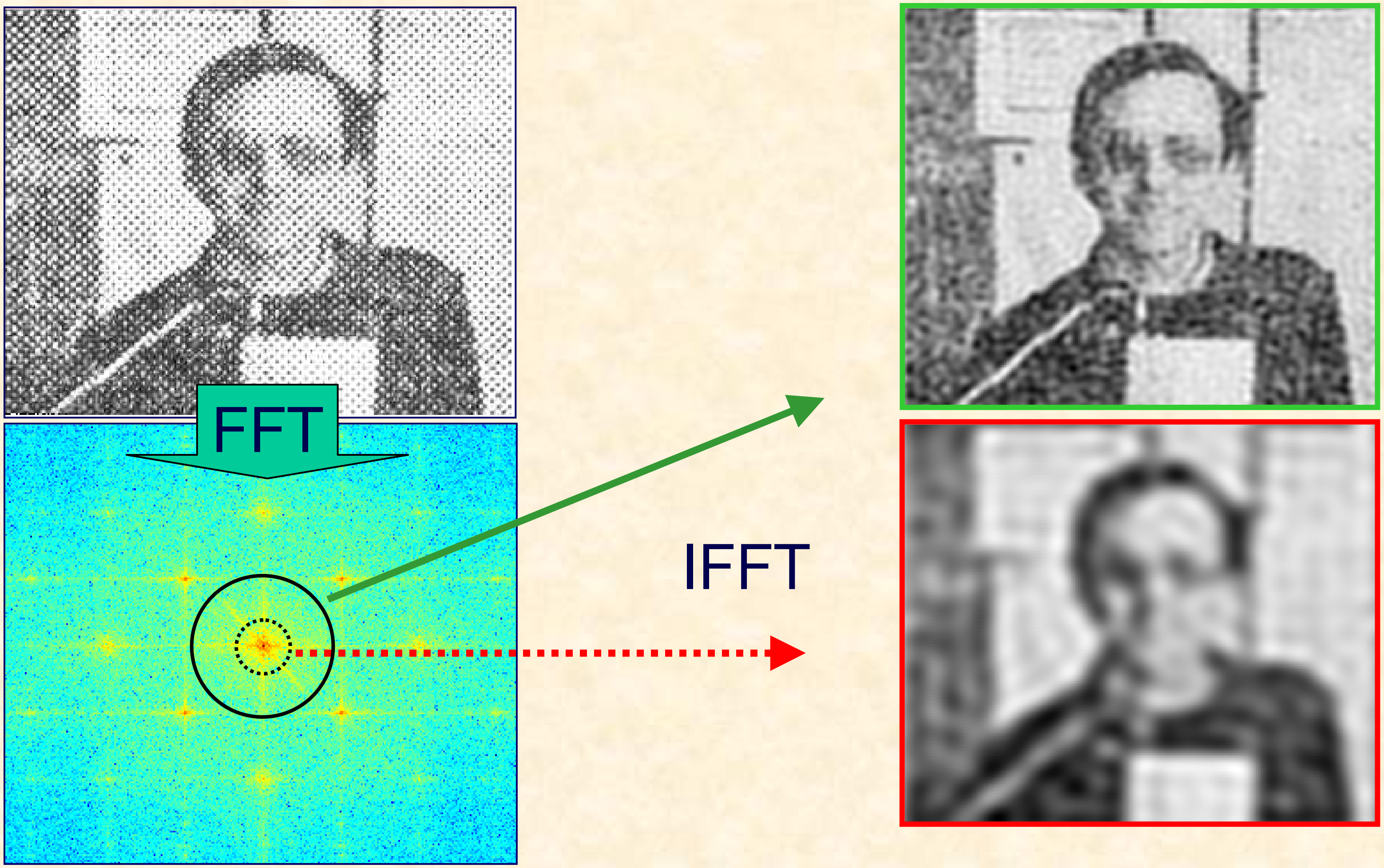
Wiener filter



MATLAB

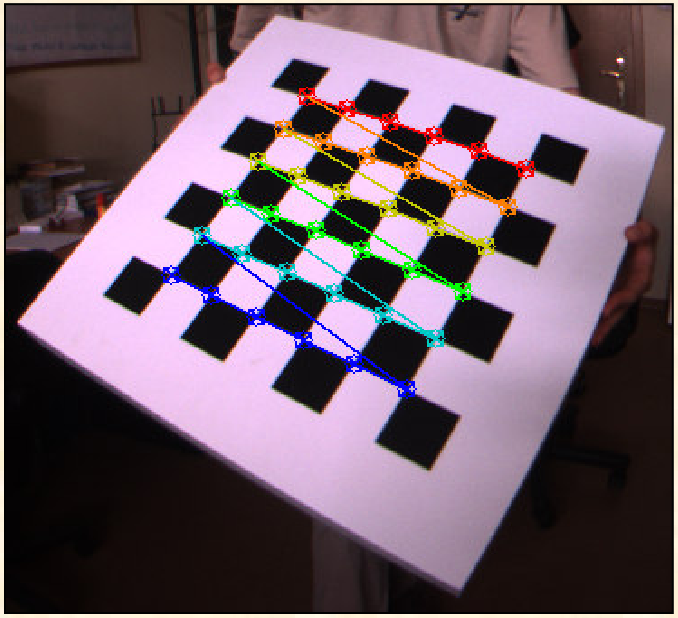
```
x=imread('text.tif');                 %load image
figure(1),imshow(x);                  %show image
N=256;
M=7;                                  % filter order
hg=fspecial('average',M);
hhg=zeros(N,N);
hhg(N/2-(M-1)/2:N/2+(M-1)/2, N/2-(M-1)/2:N/2+(M-1)/2)=hg;     %filter mask image
y=filter2(hg,x);                      %low-pass filtering
y=y+0.001*randn(256,256);
figure(2),imshow(y);

Y=fft2(y);                            %FFT of the degraded image
X=fft2(x);                            %FFT of the source image
Hg=fft2(hhg);                         %FFT of the low-pass impulse response
%----------spectra plots ----------------
figure(3),imshow(log(abs(fftshift(X)))+1,[ ]);
figure(4),imshow(log(abs(fftshift(Y)))+1,[ ]);
figure(5),imshow(log(abs(fftshift(Hg)))+1,[ ]);

Xp=Y./Hg;                             %inverse filtering
xp=abs(ifft2(Xp));                    %IFFT
figure,imshow(fftshift(xp),[ ]);
```

Example of interactive image restoration



Geometric image distortions

Assume image pixel coordinates (x,y) undergo geometric distortions. A new image $g(x', y')$ is obtained with coordinates defined as:



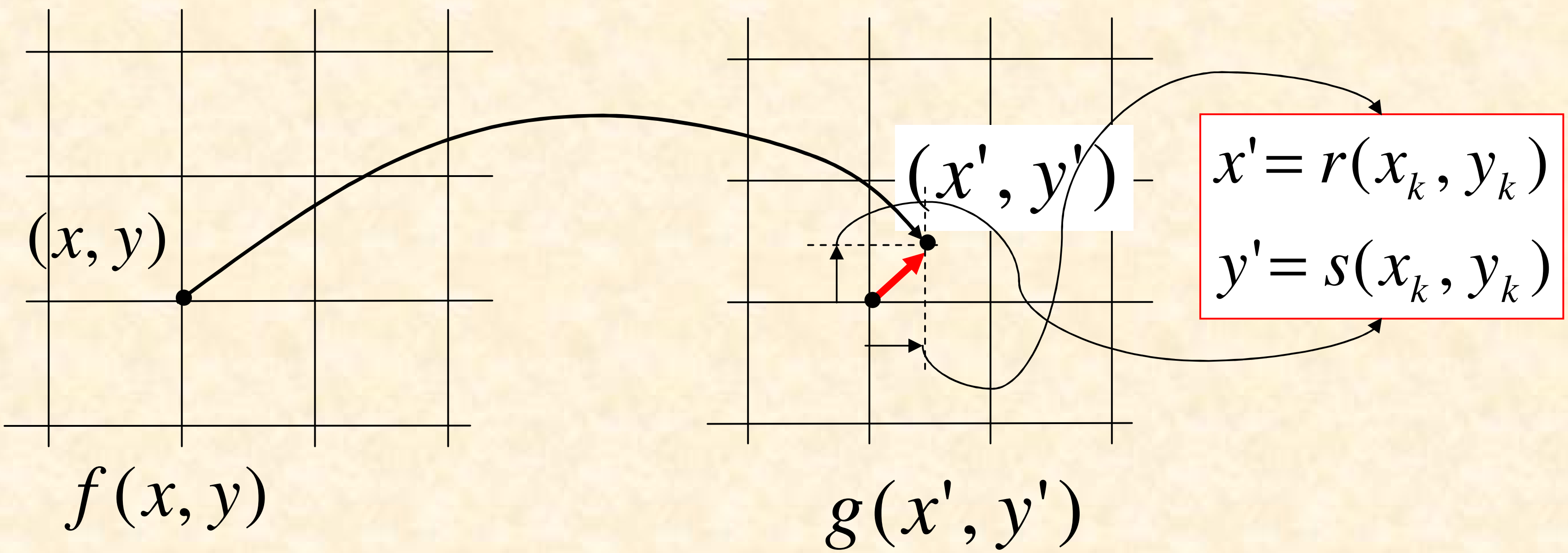
$$x' = r(x, y)$$

$$y' = s(x, y)$$

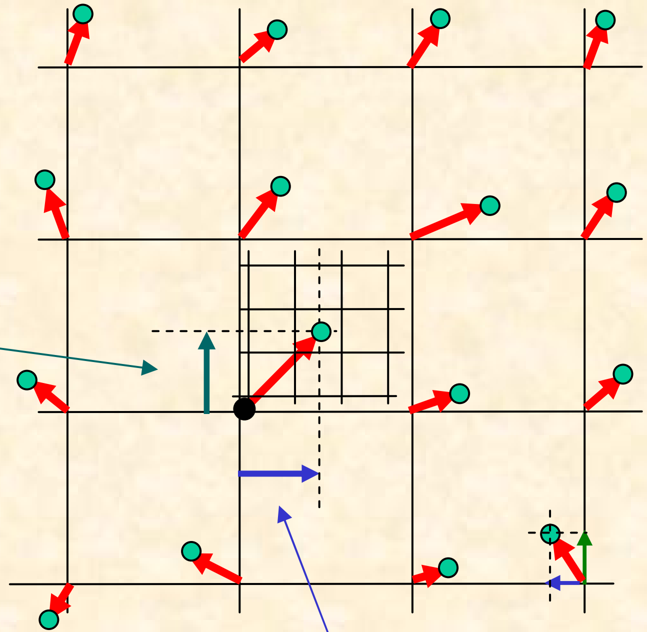
eg. for $r(x,y) = x/2$ and $s(x,y) = y/2$ image size is reduced by a factor of 2.

Correction of geometric image distortions

Spatial transformation functions $r(x,y)$ i $s(x,y)$ can be estimated from a limited number of **tie** points eg. 16×16 pixels distributed regularly in the image.



Correction of geometric image distortions



$s(x_k, y_k)$

$r(x_k, y_k)$

$k = 1, \ldots, 16$

Correction of geometric image distortions

Distortion functions $r(x,y)$ i $s(x,y)$ can be approximated e.g. by second order polynomials:

$$\begin{cases} r'(x,y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \\ s'(x,y) = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \end{cases}$$

Hence, 12 coeffcients *a*0, *a*1,…, *a*5 and *b*0, *b*1,…, *b*5 need to be determined.

Correction of geometric image distortions

Optimum polynomial coefficients can be calculated from:

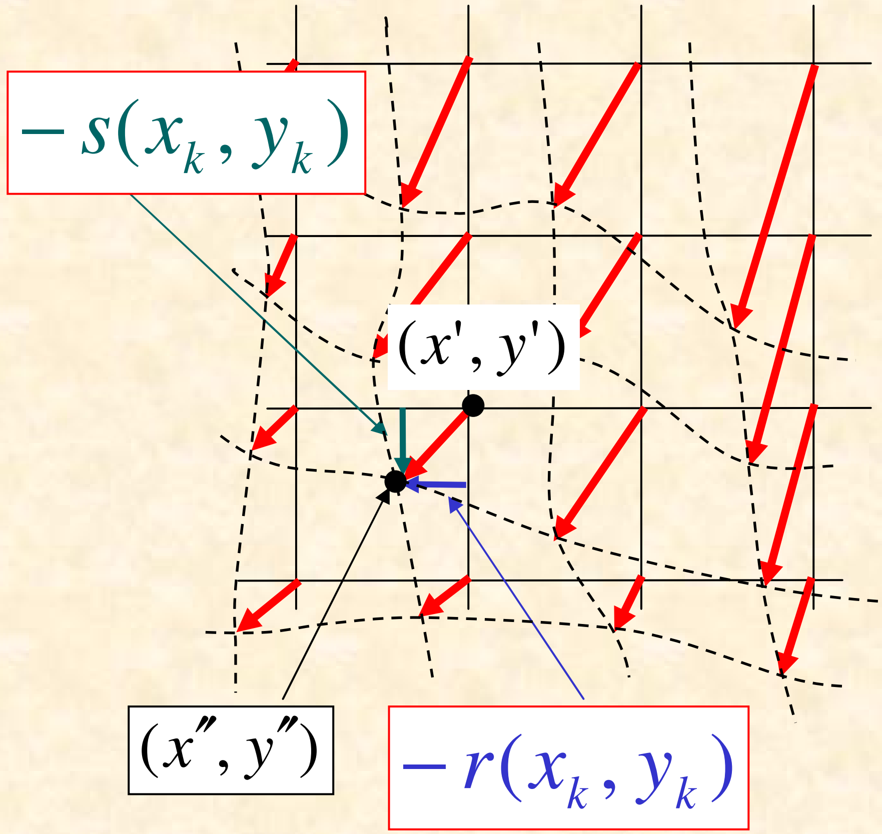
$$\begin{cases} \varepsilon_r = \sum_{k=1}^{K} \left(r'(x, y) - r(x_k, y_k) \right)^2 \\ \varepsilon_s = \sum_{k=1}^{K} \left(s'(x, y) - s(x_k, y_k) \right)^2 \end{cases}$$

Samples taken for the tie points K>=6

Correction of geometric image distortions

Coordinates after correction:

$$\begin{cases} x'' = x' - r'(x', y') \\ y'' = y' - s'(x', y') \end{cases}$$



Correction of geometric image distortions

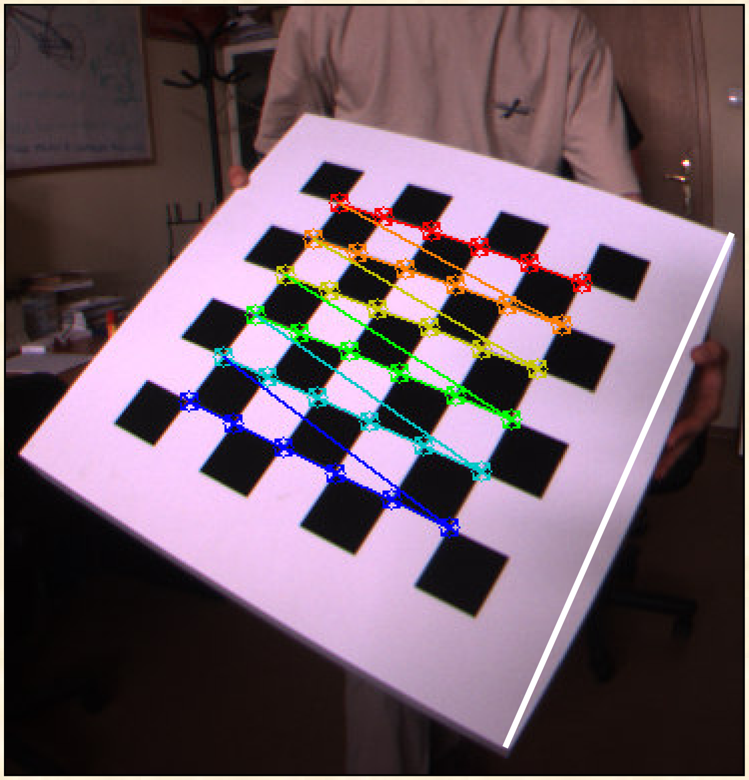
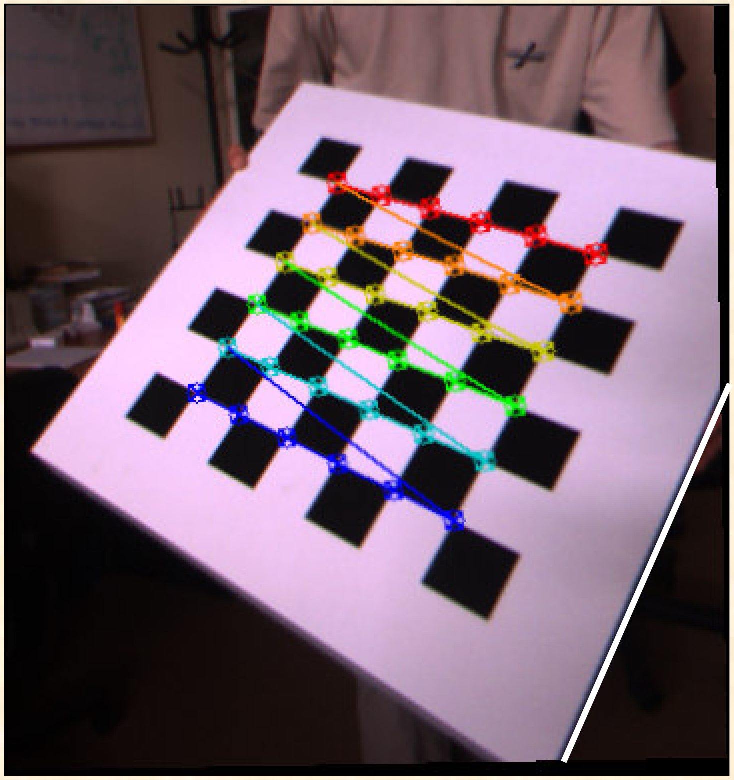
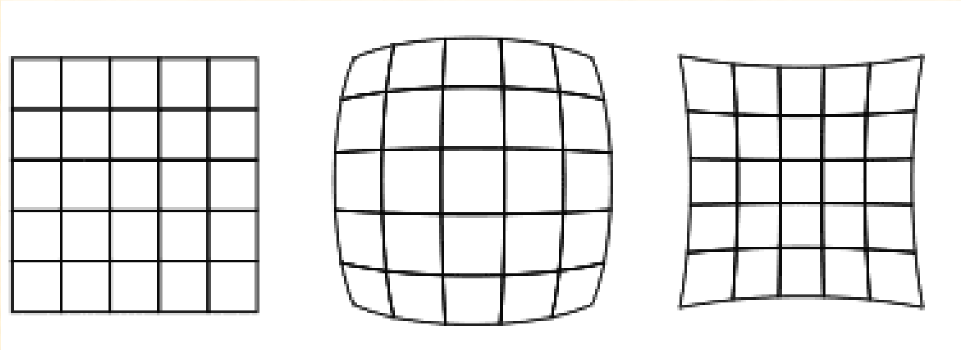


Image geometrically distorted

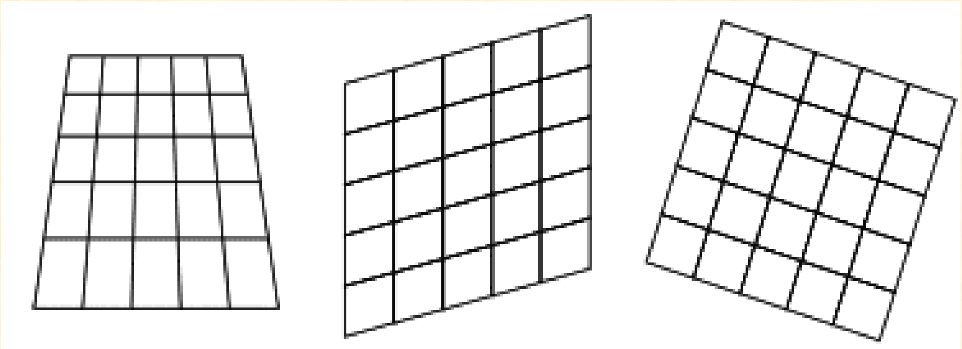


Corrected image

Examples of geometric distortions



source barrel pincushion



perspective shift skew rotation

Correction of geometric distortions - examples