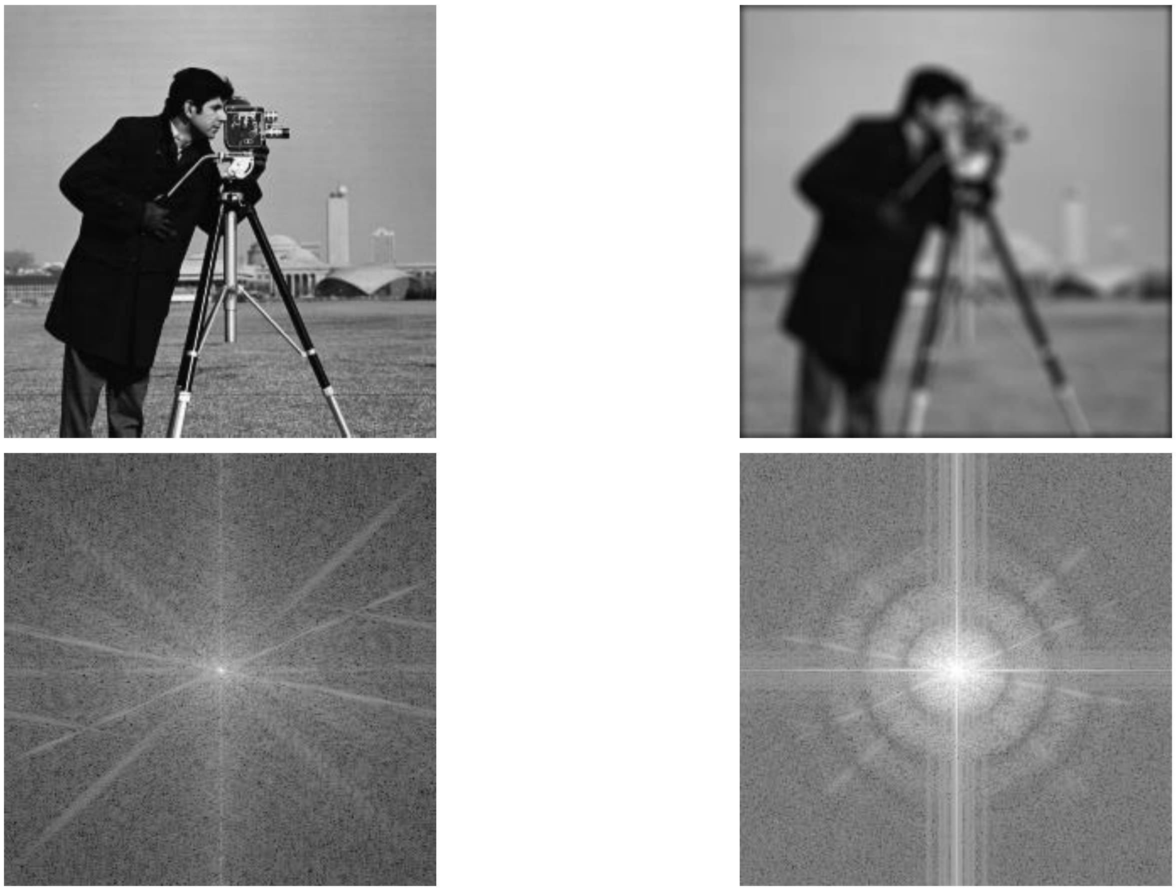
Chapter5 Image Restoration

- Preview
- 5.1 Introduction
- 5.2 Diagonalization
- 5.3 Unconstrained Restoration(inverse filtering)
- 5.4 Constrained Restoration(wiener filtering)
- 5.5 Estimating the Degradation Function
- 5.6 Geometric Distortion Correction
- 5.7 Image Inpaiting

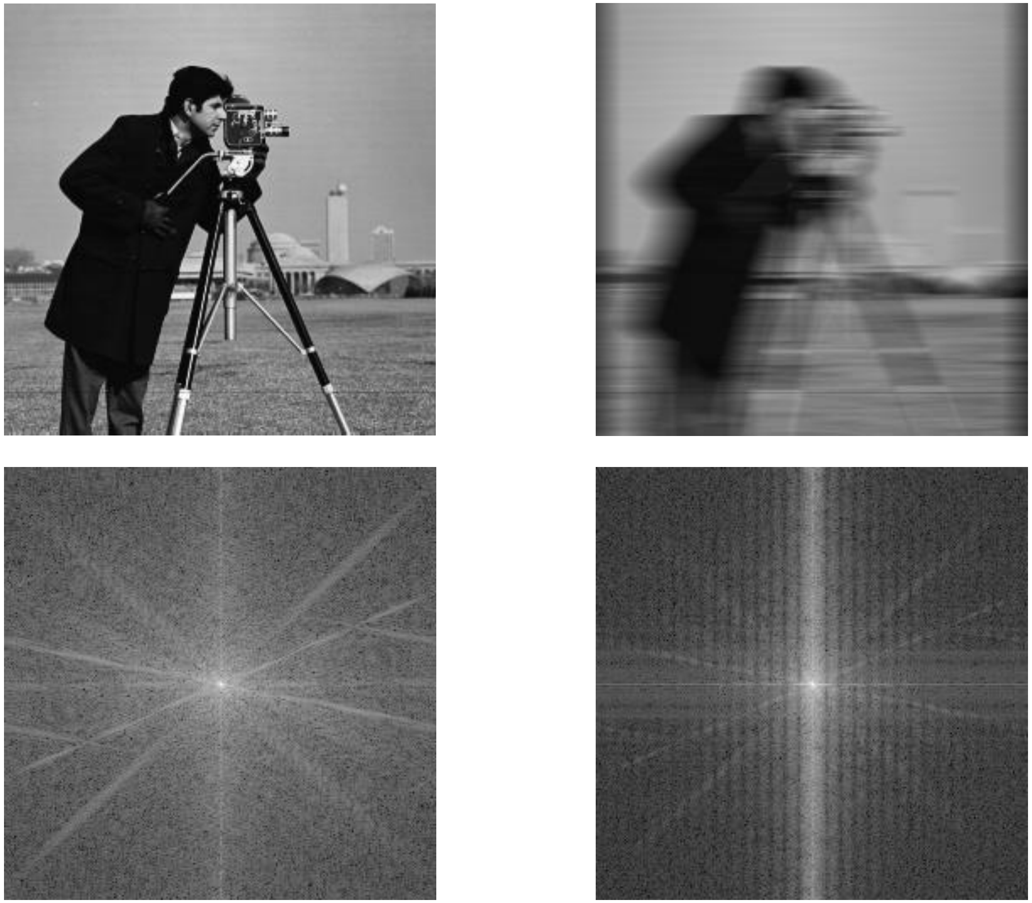
Preview

Defocused image and its DFT



Preview

Moved image and its DFT



Preview

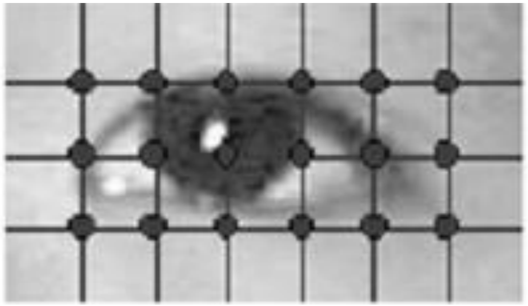
Atmospheric turbulence

Preview

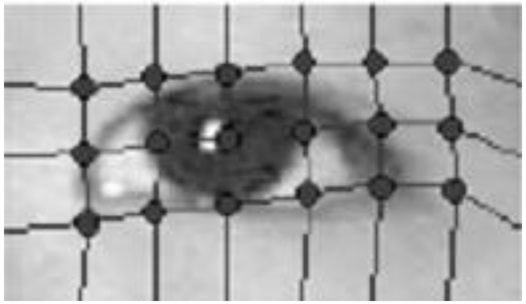
Image inpainting

Preview

Geometric transformation



grid



Changed grid

spatial transform result

5.1 Introduction

5.1.1 Purpose

"compensate for" or "undo" defects which degrade an image.

5.1.2 Degrade Causes

(1) atmospheric turbulence

(2) sampling, quantization

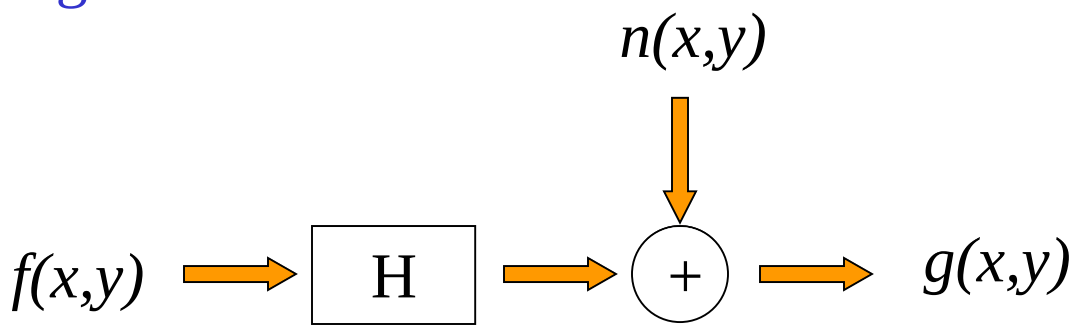
(3) motion blur

(4) camera misfocus

(5) noise

5.1 Introduction

5.1.3 degradation model



Assume it is a linear, position- invariant system, We can model a blurred image by

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

Where *h(x,y)* is called as Point Spread Function (PSF)

$$g(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n) + n(x,y)$$

5.1 Introduction

5.1.4 Methods

Unconstrained Restoration: inverse filtering

Constrained Restoration: wiener filtering

5.1.5 problem expression

Estimate a true image $f(x,y)$ from a degraded image $g(x,y)$ based on prior knowledge of PSF $h(x,y)$ and the statistical properties of noise $n(x,y)$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

$$g(x) = f(x) * h(x)$$

$$f_e(x) = \begin{cases} f(x) & 0 \le x \le A-1 \\ 0 & \text{else} \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \le x \le B-1 \\ 0 & \text{else} \end{cases}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

$$g_e(x) = \sum_{m=0}^{M-1} f_e(m) h_e(x-m) + n_e(x)$$

$M=A+B-1$

$x=0,1,\ldots,M-1$

$$g = Hf + n = \begin{bmatrix} g_e(0) \\ g_e(1) \\ \vdots \\ g_e(M\text{-}1) \end{bmatrix} = \begin{bmatrix} h_e(0) & h_e(-1) & \cdots & h_e(-M+1) \\ h_e(1) & h_e(0) & \cdots & h_e(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \cdots & h_e(0) \end{bmatrix} \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(M\text{-}1) \end{bmatrix} + \begin{bmatrix} n_e(0) \\ n_e(1) \\ \vdots \\ n_e(M-1) \end{bmatrix}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

$$h_e(x) = h_e(x+M)$$

$$H = \begin{bmatrix} h_e(0) & h_e(M-1) & \cdots & h_e(1) \\ h_e(1) & h_e(0) & \cdots & h_e(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \cdots & h_e(0) \end{bmatrix}$$

H is circulant

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

$$f_e(x) = \begin{cases} f(x,y) & 0 \le x \le A-1 \text{ and } 0 \le y \le B-1 \\ 0 & \mathrm{A} \le x \le M-1 \text{ or } \mathrm{B} \le y \le N-1 \end{cases}$$

$$h_e(x) = \begin{cases} h(x,y) & 0 \le x \le C-1 \text{ and } 0 \le y \le D-1 \\ 0 & \mathrm{A} \le x \le M-1 \text{ or } \mathrm{B} \le y \le N-1 \end{cases}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

$$g_e(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)h_e(x-m,y-n)$$

$x=0,1,\ldots,M-1$

$y=0,1,\ldots,N-1$

$$g=Hf+n=\begin{bmatrix} H_0 & H_{M-1} & \cdots & H_1 \\ H_1 & H_0 & \cdots & H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & \cdots & H_0 \end{bmatrix}\begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(MN\text{-}1) \end{bmatrix}+\begin{bmatrix} n_e(0) \\ n_e(1) \\ \vdots \\ n_e(MN-1) \end{bmatrix}$$

H is block-circulant

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

where

$$H_i = \begin{bmatrix} h_e(i,0) & h_e(i,N-1) & \cdots & h_e(i,1) \\ h_e(i,1) & h_e(i,0) & \cdots & h_e(i,2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(i,N-1) & h_e(i,N-2) & \cdots & h_e(i,0) \end{bmatrix}$$

5.2 Diagonalization

5.2.2 Diagonliztion: 1-D

The eigenvector and eigenvalue of a circulant matrix H are

$$w(k)=\left[1 \quad \exp\left(j\frac{2\pi}{M}k\right) \quad \cdots \quad \exp\left(j\frac{2\pi}{M}(M-1)k\right)\right]^T$$

$$\lambda(k)=h_e(0)+h_e(1)\exp\left(-j\frac{2\pi}{M}k\right)+\cdots+h_e(\mathrm{M-1})\exp\left(-j\frac{2\pi}{M}(M-1)k\right)$$

Combine the M eigenvectors to a matrix

$$W=[w(0) \quad w(1) \quad \cdots \quad w(M-1)]$$

then the H can be expressed as

$$H=WDW^{-1} \qquad \text{where} \qquad D(k,k)=\lambda(k)$$

5.2 Diagonalization

5.2.2 Diagonliztion: 1-D

for $$g = Hf + n$$

$$W^{-1}g = W^{-1}Hf + W^{-1}n$$

$$= W^{-1}WDW^{-1}f + W^{-1}n$$

$$= DW^{-1}f + W^{-1}n$$

$$\Downarrow$$

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

5.2 Diagonalization

5.2.2 Diagonliztion: 2-D

$$W(i,m) = \exp\left(j\frac{2\pi}{M} im \right) W_N$$

$$W_N(k,n) = \exp\left(j\frac{2\pi}{N} kn \right)$$

$$H = WDW^{-1} \Rightarrow G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$g(x,y) = f(x,y)*h(x,y) + n(x,y)$ in Spatial Coordinates

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

$g = Hf + n$ in vector form

5.3 Inverse Filtering

5.3.1 assumption

H is given, and the noise is negligible

5.3.2 degradation model

$$F(u,v) \longrightarrow \boxed{H} \longrightarrow G(u,v)$$

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

5.3 Inverse Filtering

5.3.3 restoration

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

$$H_I(u,v)=\frac{1}{H(u,v)}$$

deconvolution

5.3.4 properties

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

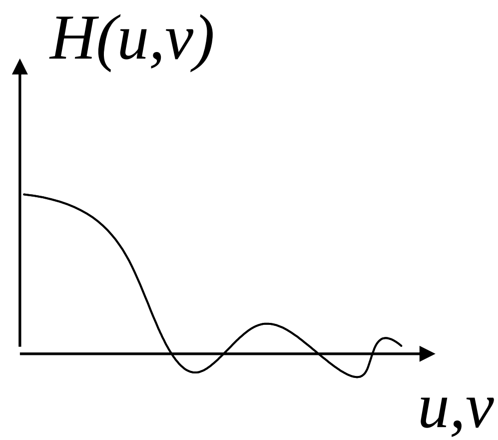
$$=\frac{H(u,v)F(u,v)+N(u,v)}{H(u,v)}$$

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

5.3 Inverse Filtering

5.3.4 properties

sensitive to additive noise: if *H(u,v)* has zero or very small value, the *N(u,v)/H(u,v)* could easily dominate the estimate



5.3.5 improvements

$$M(u,v) = \begin{cases} 1/H(u,v) & \text{if } u^2 + v^2 < w_0^2 \\ 1 & \text{else} \end{cases}$$

or

$$M(u,v) = \begin{cases} k & \text{if } H(u,v) < d \\ 1/H(u,v) & \text{else} \end{cases}$$

5.3 Inverse Filtering

5.3.4 examples: without noise

Original image

Blurred image

Restored image

5.3 Inverse Filtering

5.3.4 examples: with noise


Blurred and noised image


Restored image

5.3 Inverse Filtering

5.3.4 examples: deferent cutoff

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with *H* cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

5.4 Wiener Filtering

5.4.1 assumption

H is given, and consider the image and noise as random processes

5.4.2 request

The mean square error between uncorrupted image and estimated image is minimized. This error measure is given by

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

5.4 Wiener Filtering

5.4.3 restoration

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

$$H_W(u,v) = \frac{1}{H(u,v)} \times \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

$$= \frac{H(u,v)^*}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

Wiener filter, 1942

where

$S_n(u,v) = |N(u,v)|^2$ Power spectrum of the noise

$S_f(u,v) = |F(u,v)|^2$ Power spectrum of the undegraded image

5.4 Wiener Filtering

5.4.5 estimate the power spectrum

$$H_w(u,v) = \frac{H(u,v)^*}{\left|H(u,v)\right|^2 + K}$$

5.4.4 properties

- optimal in terms of the mean square error
- When $H(u,v)=0$, $H_w(u,v)=0$

5.4 Wiener Filtering

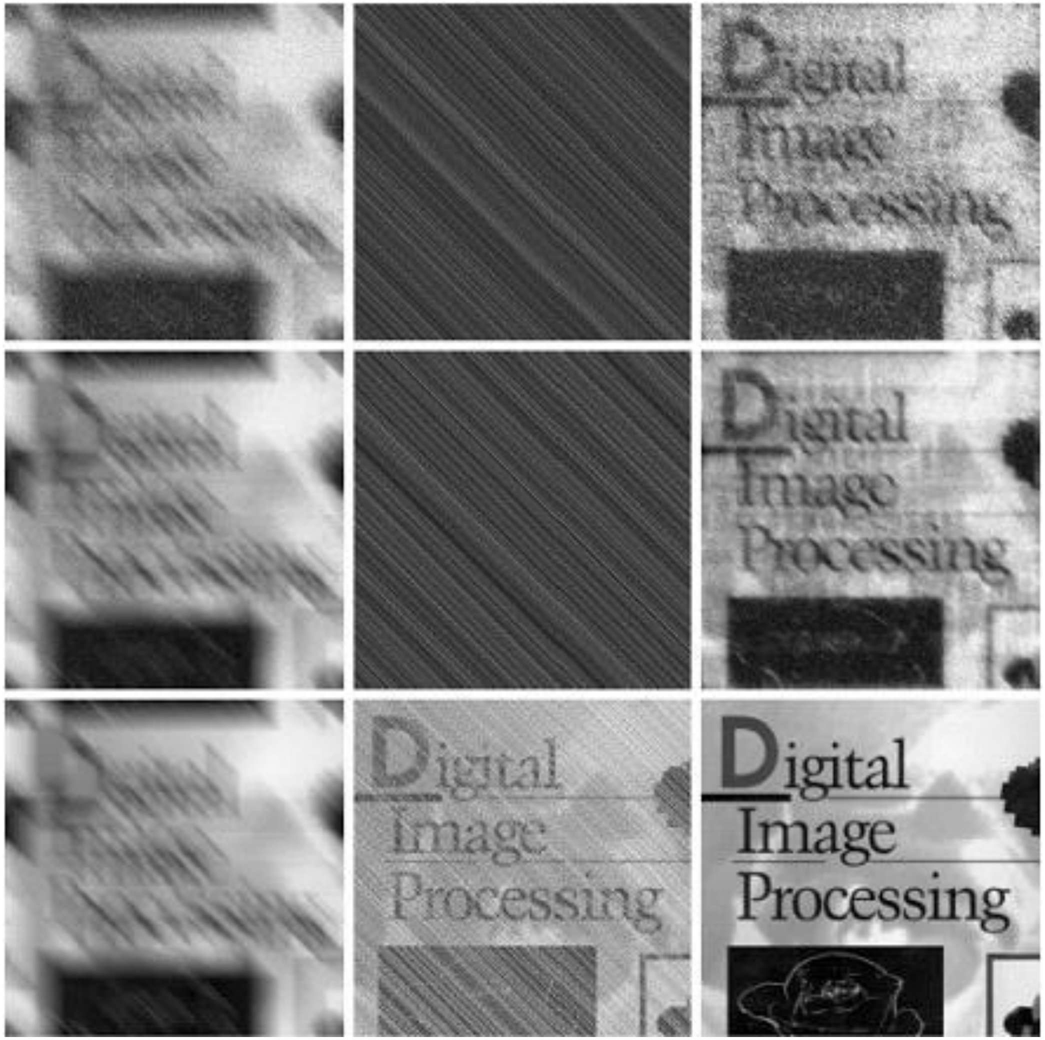
5.4.5 experimental results:

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

5.4 Wiener Filtering

5.4.5 experimental results:



The first row:
Image corrupted by motion blur and additive noise

The second row:
Results of inverse filtering

The third row:
Results of Wiener filtering

5.5Estimating the Degradation Function

(blind deconvolution)

5.5.1 Estimation by image observation

(1) Choose observed sub-image $g_s(x, y)$

(2) Denote the constructed sub-image as $\hat{f}_s(x, y)$

(3) Assume noise is negligible

then

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

5.5Estimating the Degradation Function

5.5.2 Estimation by experimentation

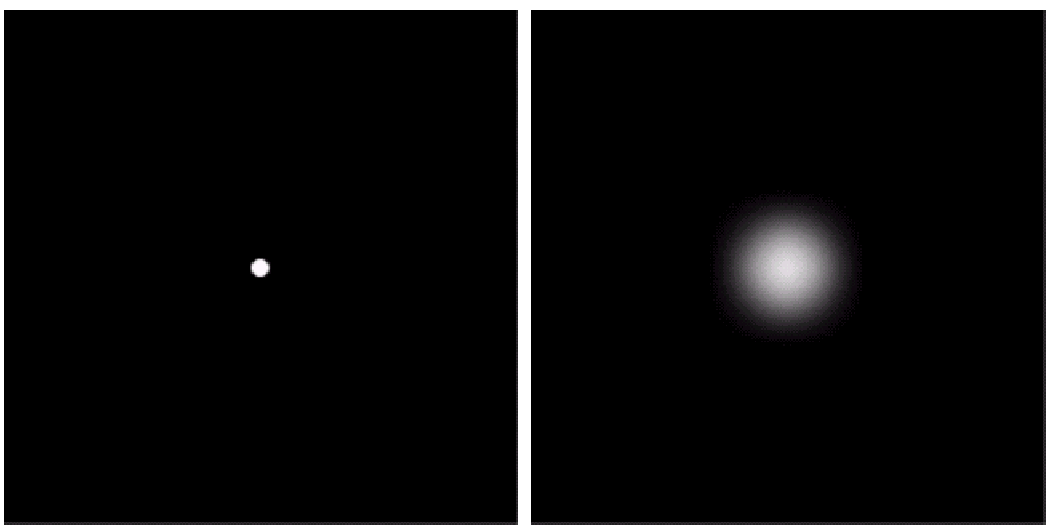
If $f(x,y)=\delta(x,y)$

$$\delta(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

Impulse response

then

$$g(x,y)=\delta(x,y)*h(x,y)=h(x,y)$$



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling

An atmospheric turbulence model based on the physical characteristics

Hufnagel and Stanley, 1964

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

where k is a constant

it has the same form as the Gaussian lowpass filter

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

If *T* is the duration of the exposure, the effect of image motion follows that

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

where $x_0(t)$ and $y_0(t)$ are the time varying components of motion in the *x*-direction and *y*-direction. Its Fourier transform is

$$G(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-2\pi j(ux+vy)}dxdy$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x-x_0(t),y-y_0(t)]dt\right]e^{-2\pi j(ux+vy)}dxdy$$

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

Reversing the order of integration:

$$G(u,v)=\int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f[x-x_0(t),y-y_0(t)]e^{-2\pi j(ux+vy)}dxdy\right]dt$$

Using the translation Properties of Fourier transformation, then

$$G(u,v)=\int_0^T F(u,v)e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

$$=F(u,v)\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

By defining $$H(u,v)=\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

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

Suppose that the image in question undergoes uniform linear motion in the *x*-direction only, at a rate given by $x_0(t)=at/T$

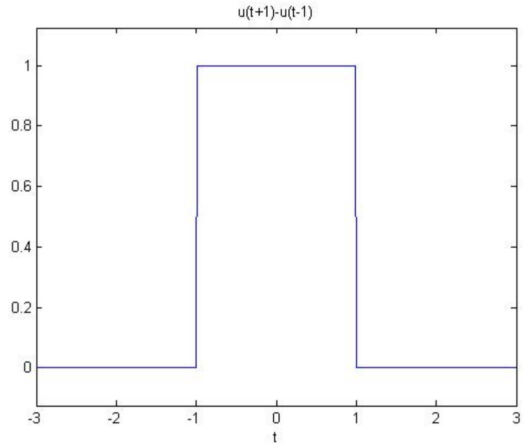
$$H(u,v)=\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$
$$=\int_0^T e^{-2\pi juat/T}dt$$
$$=\frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

5.5Estimating the Degradation Function

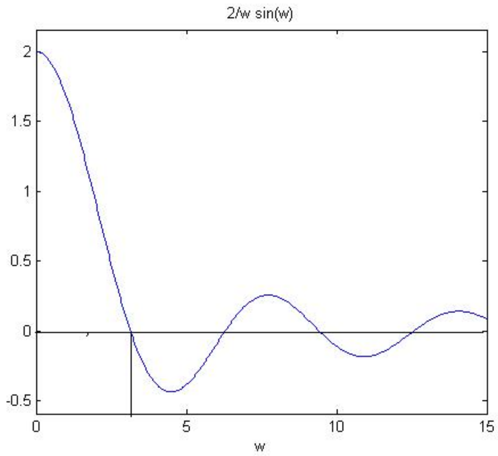
5.5.3 Estimation by modeling :restoration of uniform linear motion

If we allow the y-component to wary as well with the motion given by $y_0(t) = bt/T$ then the degradation function becomes

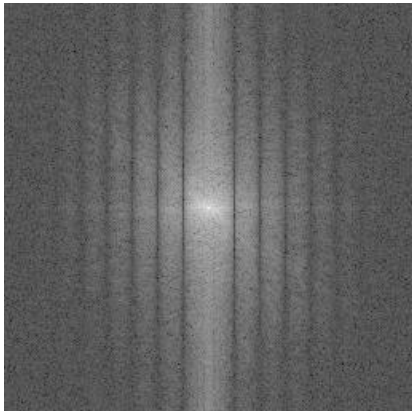
$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$

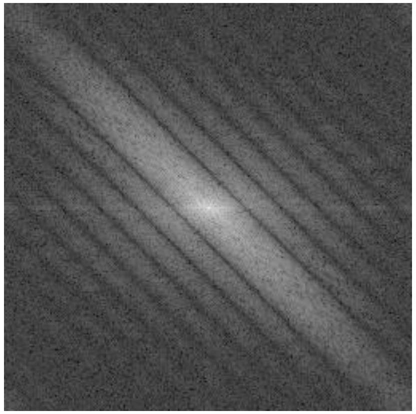


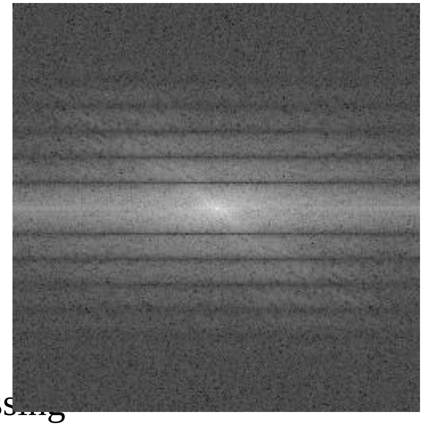
(a)



(b)







5.6 Geometric Transformation

5.6.introduction

$$f(x,y) \longrightarrow \boxed{T} \longrightarrow g(x,y)$$

$$g(x,y) = T[(f(x,y)] = f(x',y')$$

where

$$x' = r(x,y) \quad y' = s(x,y)$$

for example: zoom out $x' = x/2 \quad y' = y/2$

zoom in $x' = 2x \quad y' = 2y$

a b c
d e f

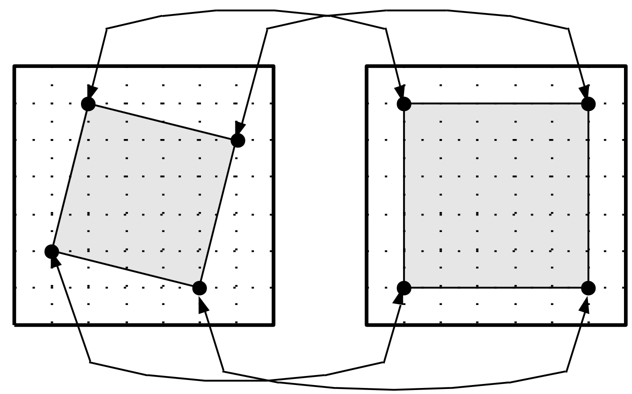
FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

5.6 Geometric Distortion Correction

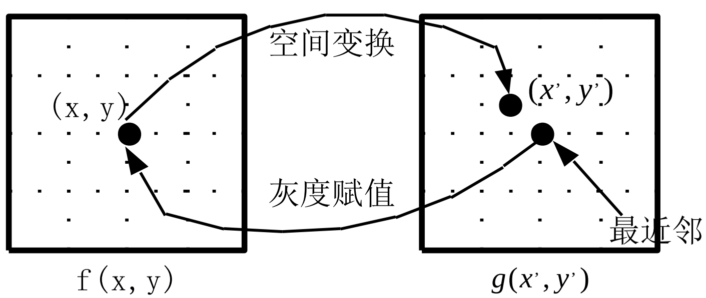
5.6.1 introduction

A geometric transformation consists of two basic operations:

(1) Spatial transformation



(2) Gray-level interpolation



Direct transform

Inverse transform

5.6 Geometric Distortion Correction

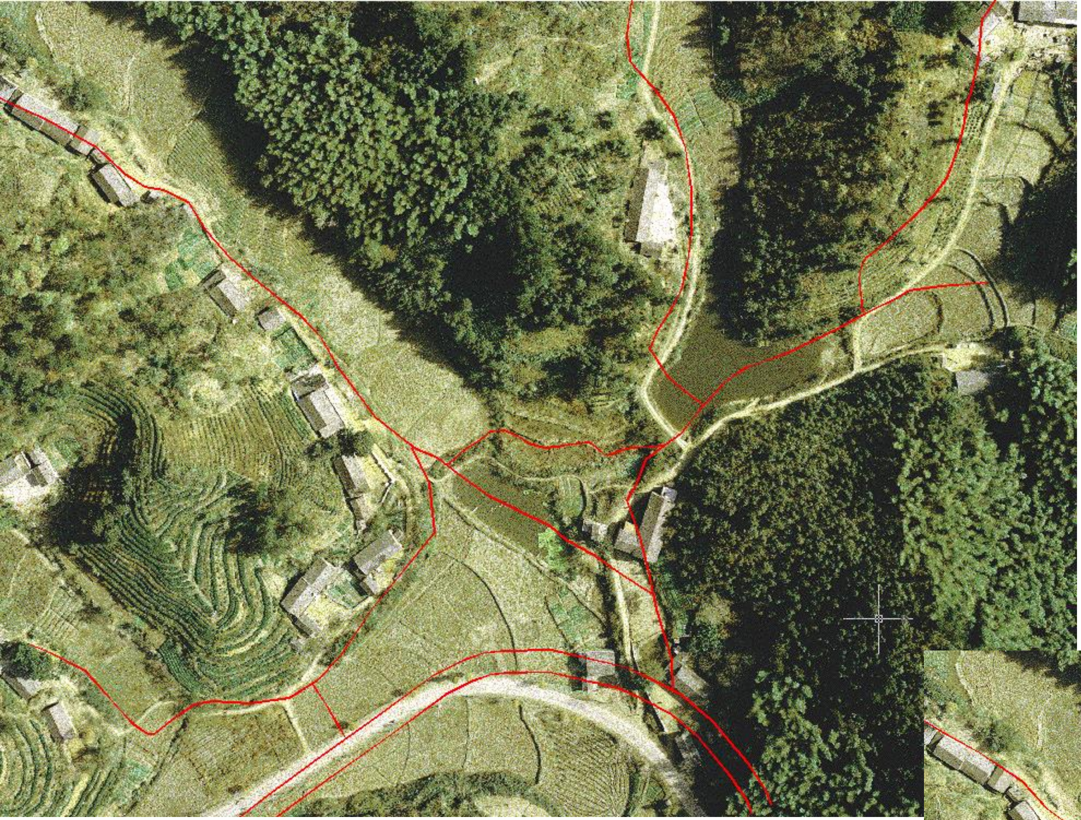
5.6.2 spatial transformations

Linear correction

$$r(x,y) = a_1 x + a_2 y + a_3$$
$$s(x,y) = b_1 x + b_2 y + b_3$$

Quadratic correction

$$r(x,y) = a_1 x^2 + a_2 y^2 + a_3 xy + a_4 x + a_5 y + a_6$$
$$s(x,y) = b_1 x^2 + b_2 y^2 + b_3 xy + b_4 x + b_5 y + b_6$$



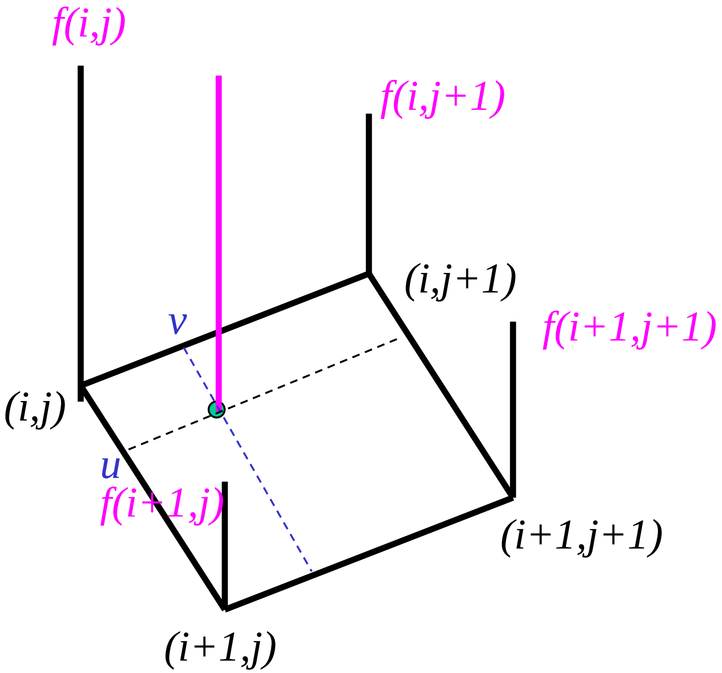
5.6 Geometric Distortion Correction

5.6.3 gray-level interpolation

- Nearest neighbor interpolation

$$g(x, y) = f(x', y') = f(i+u, j+v)$$

$$u, v \in (0,1) \qquad i, j \in Z$$



$$x' = round(x') \qquad y' = round(y')$$

5.6 Geometric Distortion Correction

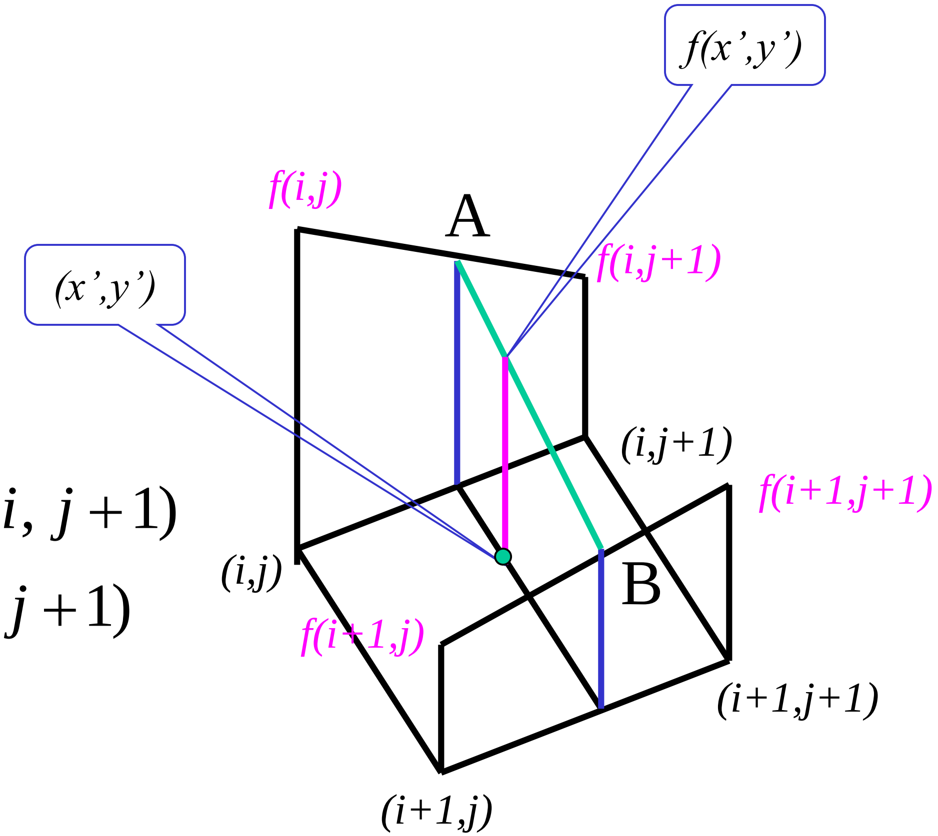
5.6.3 gray-level interpolation

•Bi-linear interpolation

$$f(x', y') = f(i+u, j+v)$$

$$u, v \in (0,1) \qquad i, j \in Z$$

$$= (1-u)(1-v)f(i,j) + (1-u)vf(i,j+1)$$
$$+ u(1-v)f(i+1,j) + uvf(i+1,j+1)$$

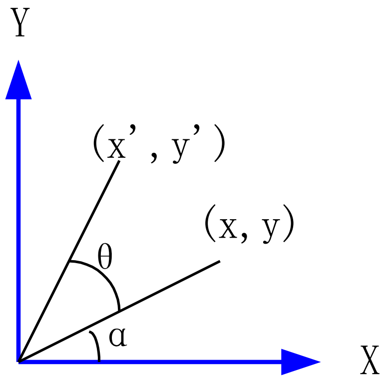


It can be operated by a mask

$$\begin{vmatrix} (1-u)(1-v) & (1-u)v \\ u(1-v) & uv \end{vmatrix}$$

5.6 Geometric Distortion Correction

5.6.3 gray-level interpolation



Rotation in 2D plane

$$x' = r\cos(\alpha+\theta) = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta$$

$$y' = r\sin(\alpha+\theta) = r\cos\alpha\sin\theta + r\sin\alpha\cos\theta$$

The original coordinated of the point in polar coordinates are:

$$x = r\cos\alpha \qquad y = r\sin\alpha$$

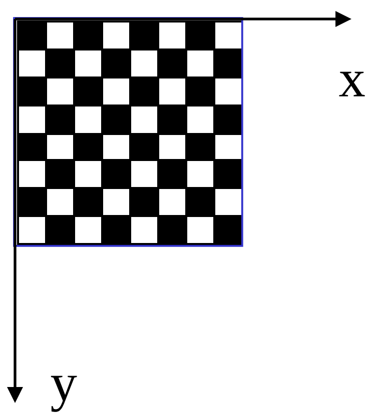
5.6 Geometric Distortion Correction

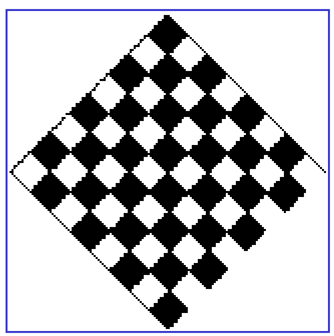
5.6.3 gray-level interpolation

Rotation in 2D plane

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$





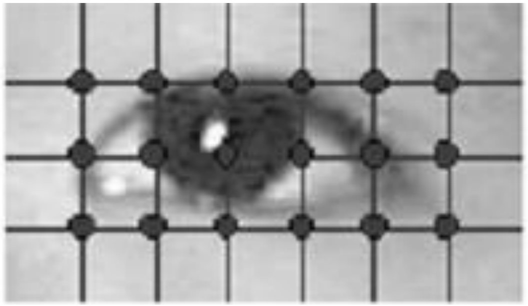
$$\theta = 45^\circ$$

Notes:

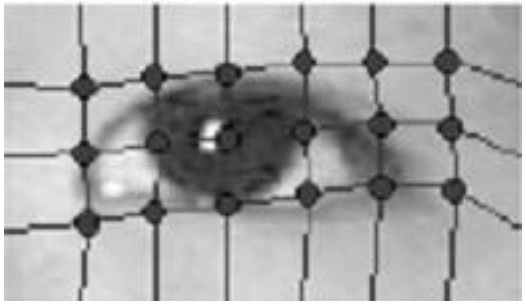
1. **The origin is in the center of image and the direction of x-axis is opposite**
2. **The result image's size is depended on the rotate angle**
3. **Interpolation is needed**

5.6 Geometric Distortion Correction

5.6.4 experimental results



grid

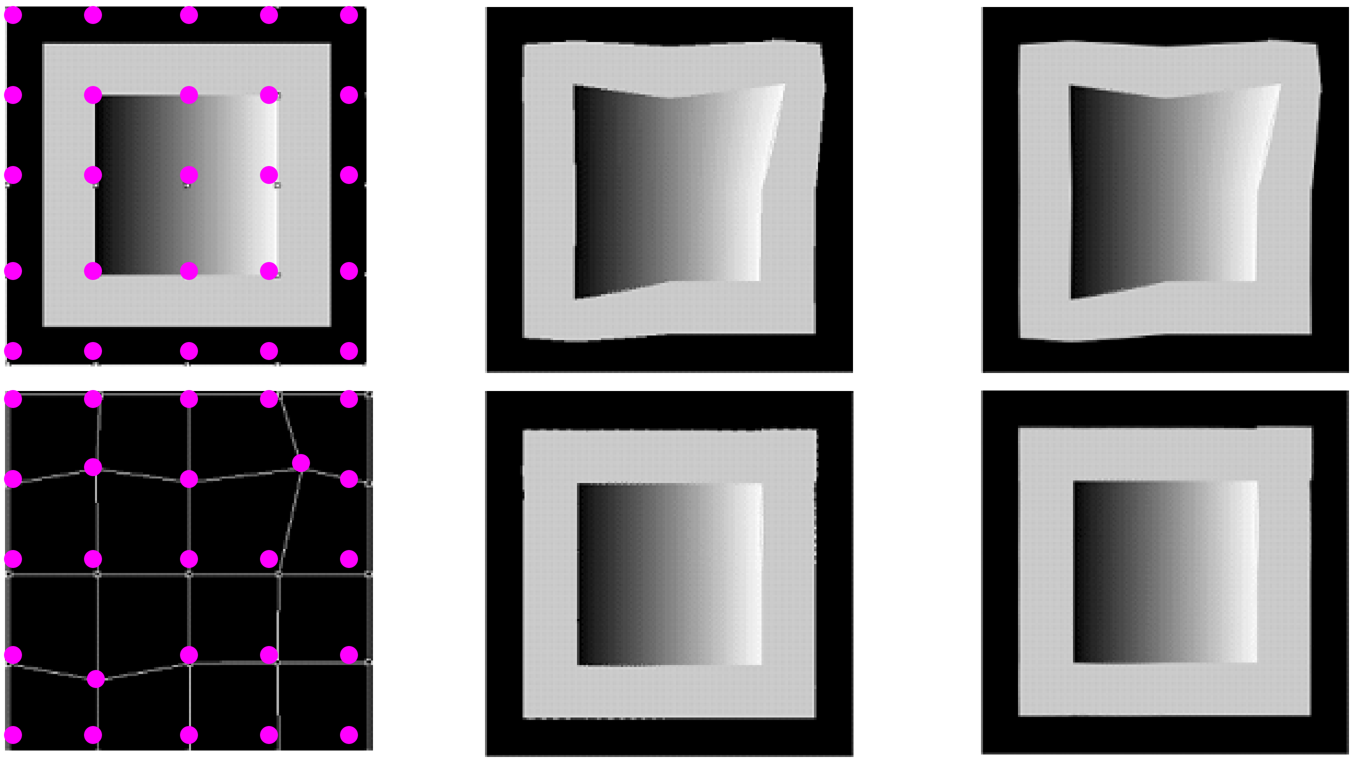


Changed grid

spatial transform result

5.6 Geometric Distortion Correction

5.6.4 experimental results



Spatial transform

Nearest neighborhood interpolation

Bi- linear interpolation

Summary

• Inverse filtering is a very easy and accurate way to restore an image provided that we know what the blurring filter is and that we have no noise

•Wiener filtering is the optimal tradeoff between the inverse filtering and noise smoothing

• It is possible to restore an image without having specific knowledge of degradation filter and additive noise. However, not knowing the degradation filter **h** imposes the strictest limitations on our restoration capabilities.

homework

- 写出逆滤波和维纳滤波图象恢复的具体步骤。
- 推导水平匀速直线运动模糊的点扩展函数的数学公式并画出曲线。
- 编程实现lema.bmp的任意角旋转。

The End