Lecture 3: Image Restoration

B14 Image Analysis Michaelmas 2014 A. Zisserman

- Image degradations
  - motion blur, focus blur, resolution
- The inverse filter
- The Wiener filter
- MAP formulation

In contrast to image enhancement, in image restoration the degradation is modelled. This enables the effects of the degradation to be (largely) removed

Degradations

- original

- optical blur

- motion blur

- spatial quantization (discrete pixels)

- additive intensity noise

Overview – Deconvolution

The objective is to restore a degraded image to its original form.

An observed image can often be modelled as:

$$g(x,y) = \int\int h(x-x', y-y') f(x', y')\, dx'\, dy' + n(x,y)$$

where the integral is a convolution, h is the point spread function of the imaging system, and n is additive noise.

The objective of image restoration in this case is to estimate the original image f from the observed degraded image g.

Degradation model

Model degradation as a convolution with a linear, shift invariant, filter $h(x,y)$

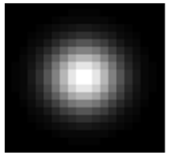
- Example: for out of focus blurring, model $h(x,y)$ as a Gaussian

original

$f(x,y)$

$*$

$$\frac{1}{2\pi\sigma^2}e^{-r^2/2\sigma^2}$$



$h(x,y)$

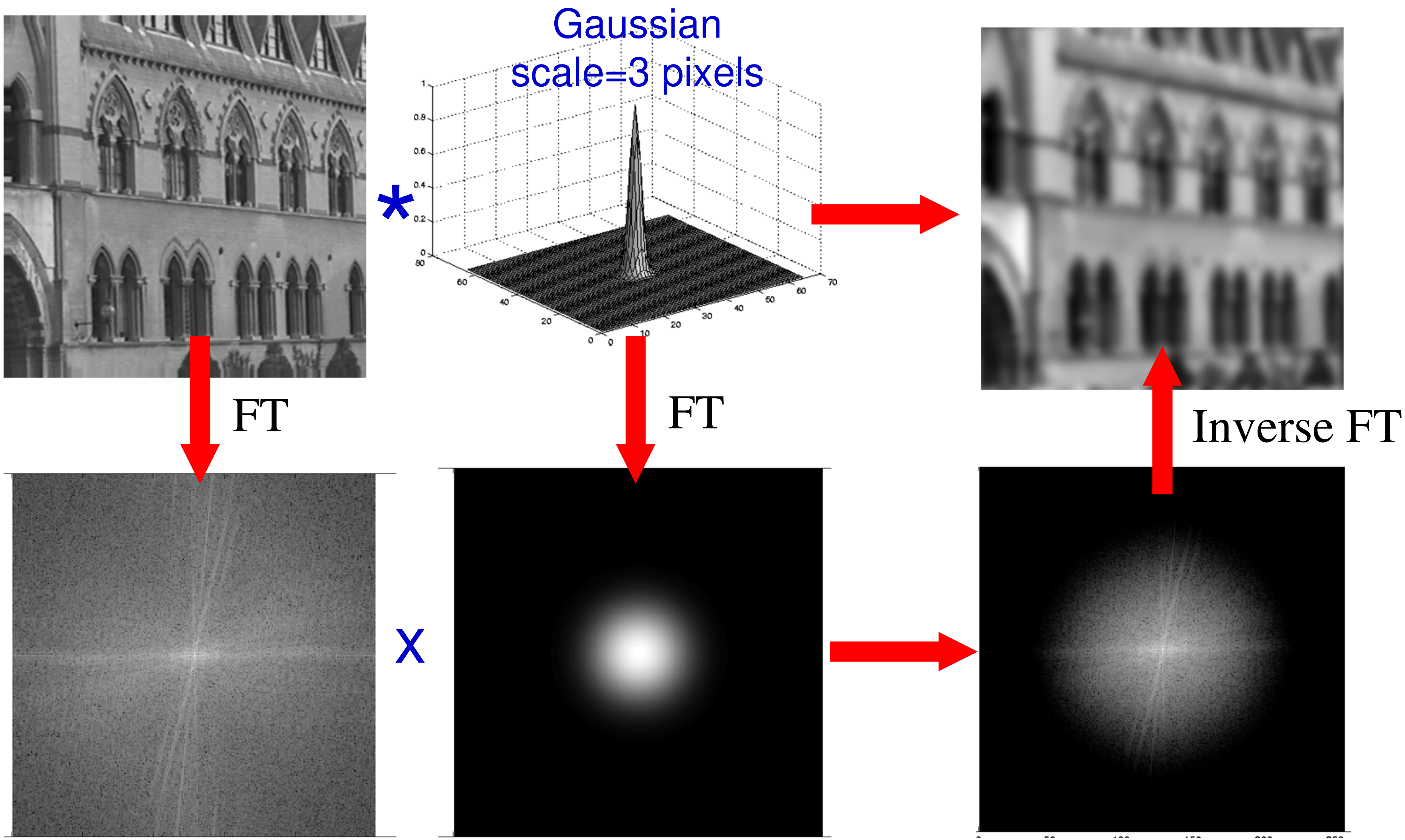
blurred

$g(x,y)$

i.e. : $g(x,y) = h(x,y) * f(x,y)$

$h(x,y)$ is the impulse response or point spread function of the imaging system

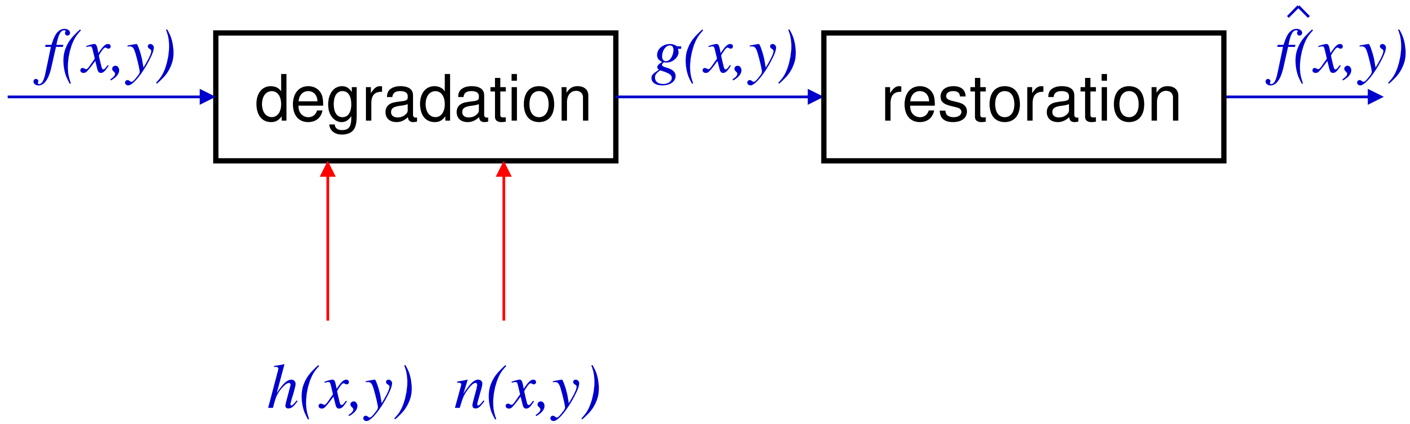
The challenge: loss of information and noise



Blurring acts as a low pass filter and attenuates higher spatial frequencies

Definitions

- $f(x,y)$ – image before degradation, 'true image'
- $g(x,y)$ – image after degradation, 'observed image'
- $h(x,y)$ – degradation filter
- $\hat{f}(x,y)$ – estimate of $f(x,y)$ computed from $g(x,y)$
- $n(x,y)$ – additive noise



$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \Leftrightarrow \quad G(u,v) = H(u,v)\, F(u,v) + N(u,v)$$

The inverse filter

Start from the generative model

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \Leftrightarrow \quad G(u,v) = H(u,v)\, F(u,v) + N(u,v)$$

and for the moment ignore $n(x,y)$, then an estimate of $f(x,y)$ is obtained from

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

Restoration with an inverse filter

1D vector explanation

$$g = h * f$$

$$g_n = \frac{1}{4} f_{n-1} + \frac{1}{2} f_n + \frac{1}{4} f_{n+1}$$

$$g_1 = \frac{1}{4} f_0 + \frac{1}{2} f_1 + \frac{1}{4} f_2$$

$$g_2 = \frac{1}{4} f_1 + \frac{1}{2} f_2 + \frac{1}{4} f_3$$

For n points (ignoring boundaries)

$$\mathbf{g} = \mathtt{A}\mathbf{f}$$

where $\mathbf{g}$ and $\mathbf{f}$ are n-vectors, and $\mathtt{A}$ is an $n \times n$ matrix. Hence, ignoring possible problems if $\mathtt{A}$ is singular,

$$\mathbf{f} = \mathtt{A}^{-1}\mathbf{g}$$

Fourier trick

$$g = h * f$$

$$G = HF$$

$$F = G/H$$

$$f = \mathrm{IFT}(G/H)$$

Example : Deblurring (deconvolution)

Image blurred with Gaussian point spread function

$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2}$ $n(x,y)$ = Normal distribution, mean zero

$f(x,y)$

$g(x,y)$

blur σ = 1.0 pixels

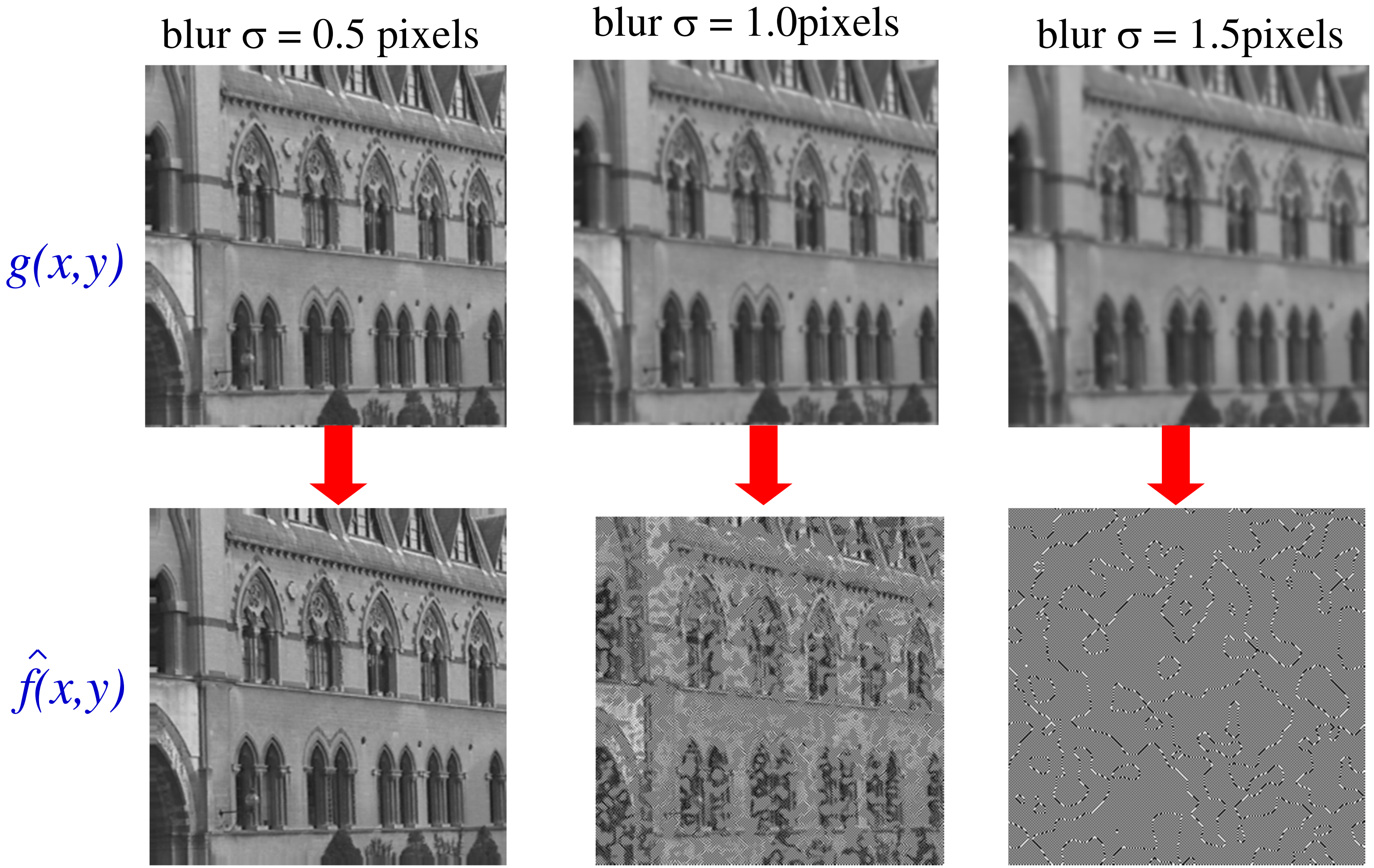
noise σ = 0.3 grey levels

Restoration with an inverse filter

$\hat{F}(u,v) = G(u,v) / H(u,v)$ where $H(u,v)$, is the FT of the Gaussian

Deblurring with an inverse filter

noise σ = 0.3 grey levels $\hat{F}(u,v) = G(u,v) \,/\, H(u,v)$

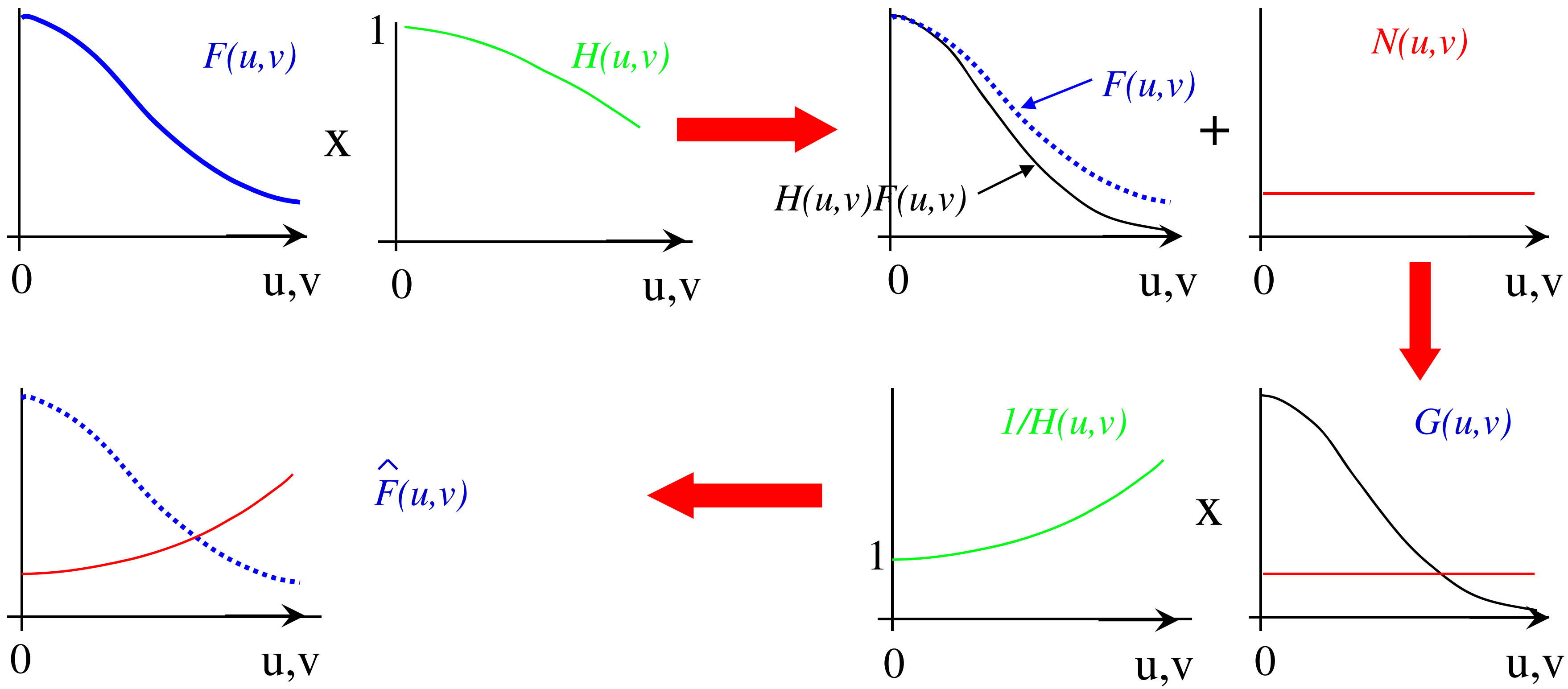


The problem of noise amplification

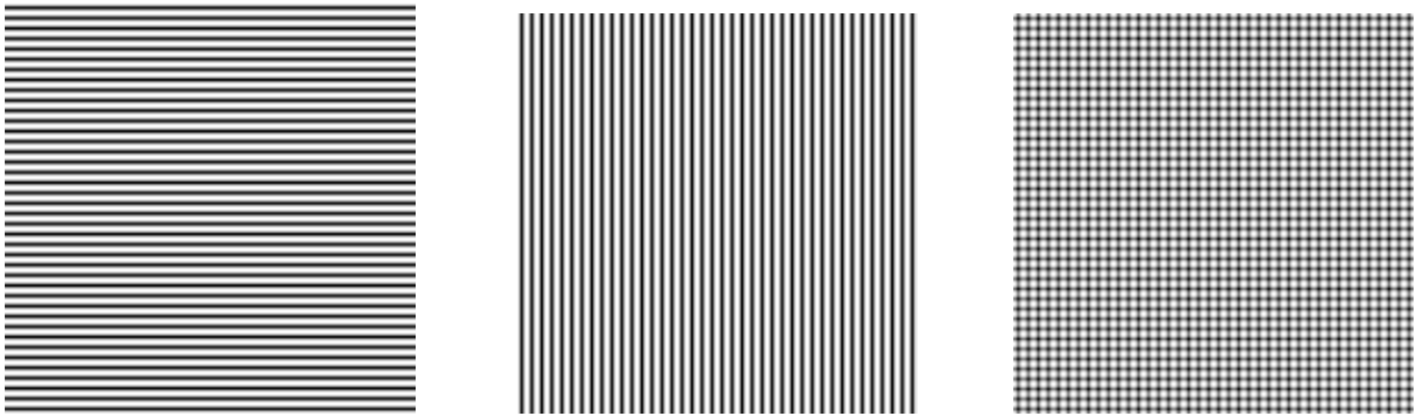
$$G(u,v) = H(u,v)\,F(u,v) + N(u,v)$$

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

Schematically …

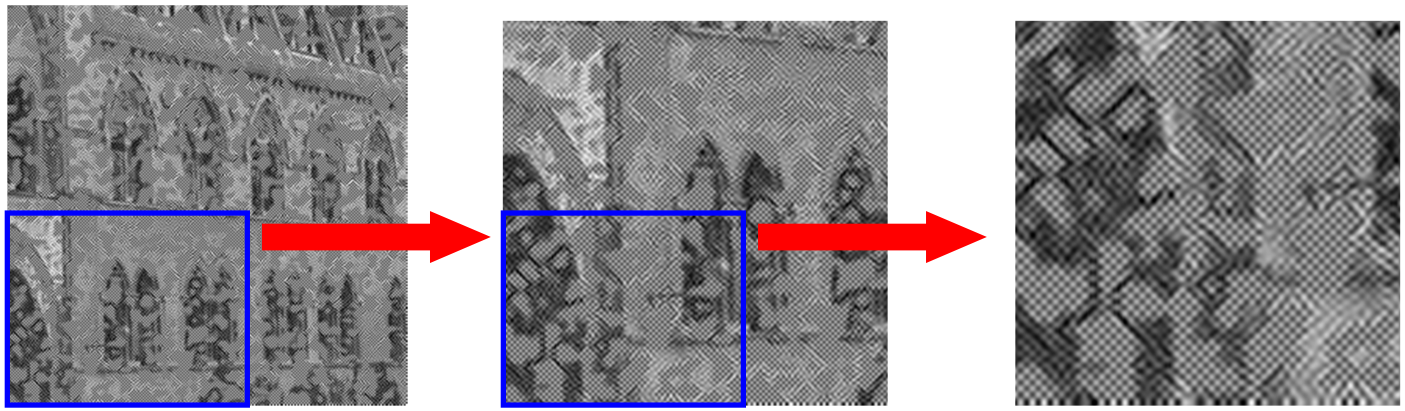


high spatial frequency sinusoids



$\hat{f}(x,y)$

blur σ = 1.0 pixels



The Wiener filter

The Wiener filter

$\widehat{F}(u,v) = W(u,v)\, G(u,v)$

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

where

$K(u,v) = S_\eta(u,v)/S_f(u,v)$

$S_f(u,v) = |F(u,v)|^2$ power spectral density of $f(x,y)$

$S_\eta(u,v) = |N(u,v)|^2$ power spectral density of $\eta(x,y)$

Frequency behaviour

$\widehat{F}(u,v) = W(u,v)\ G(u,v)$

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

- If $K = 0$ then $W(u,v) = 1 / H(u,v)$, i.e. an inverse filter
- If K >> $|H(u,v)|$ *for large* u,v, then high frequencies are attenuated

- $|F(u,v)|$ and $|N(u,v)|$ are often known approximately, or
- K is set to a constant scalar which is determined empirically

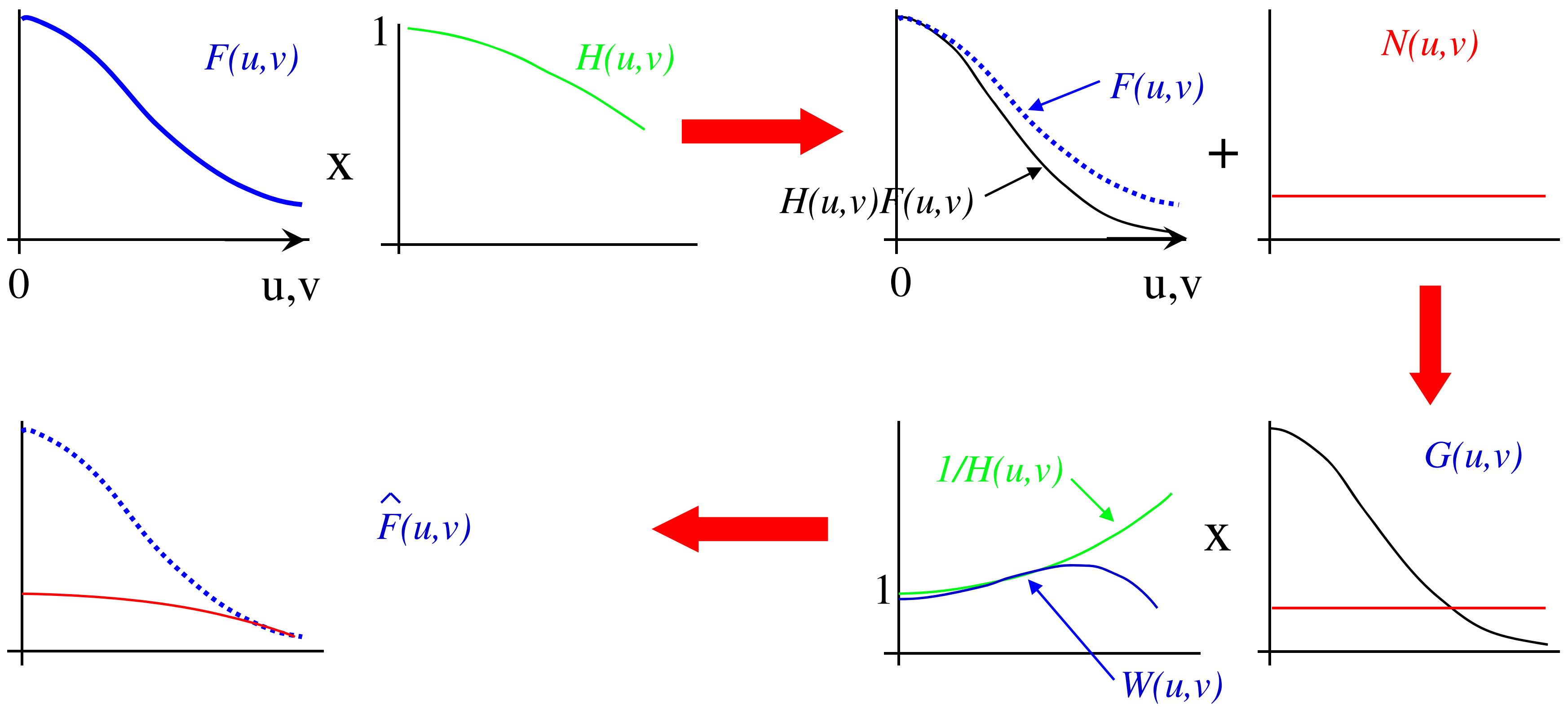
- A Wiener filter minimizes the least square error $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(f(x,y) - \hat{f}(x,y)\right)^2 dxdy$

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

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

Schematically …

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



Restoration with a Wiener filter

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

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

Example 1: Focus deblurring with a Wiener filter

blur σ = 1.5 pixels

noise σ = 0.3 grey levels

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

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

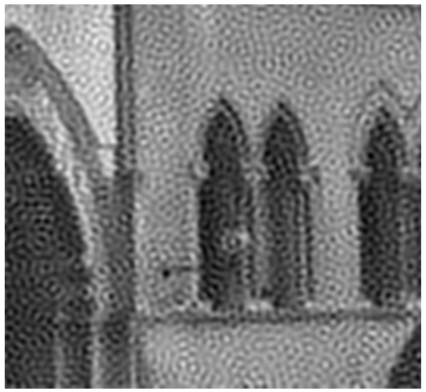
$g(x,y)$

$\hat{f}(x,y)$

K = 1.0 e -5

K = 1.0 e -3

K = 1.0 e -1



blur $\sigma = 3.0$ pixels

noise $\sigma = 0.3$ grey levels

$f(x,y)$

$g(x,y)$

$\hat{f}(x,y)$

$K = 5.0$ e -4

Wiener filter – sketch derivation

Aim is to find filter which minimizes

$$\mathcal{E} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(f(x,y) - \hat{f}(x,y)\right)^2 dxdy$$

$$\begin{aligned}\mathcal{E} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|f(x,y) - \hat{f}(x,y)\right|^2 dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|F(u,v) - \hat{F}(u,v)\right|^2 dudv\end{aligned}$$

Parseval's Theorem

$$\hat{F} = WG = WHF + WN$$

$$F - \hat{F} = (1 - WH)F - WN$$

$$\begin{aligned}\mathcal{E} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|(1 - WH)F - WN|^2\, dudv \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left\{|(1 - WH)F|^2 + |WN|^2\right\} dudv\end{aligned}$$

since $f(x,y)$ and $\eta(x,y)$ uncorrelated

- Note, integrand is sum of two squares

Minimize integral if integrand minimum for all *(u,v)*

NB $\frac{\partial}{\partial z}(zz^*) = 2z^*$

$$\frac{\partial}{\partial z} \rightarrow 2\left(-(1 - W^*H^*)H|F|^2 + W^*|N|^2\right) = 0$$

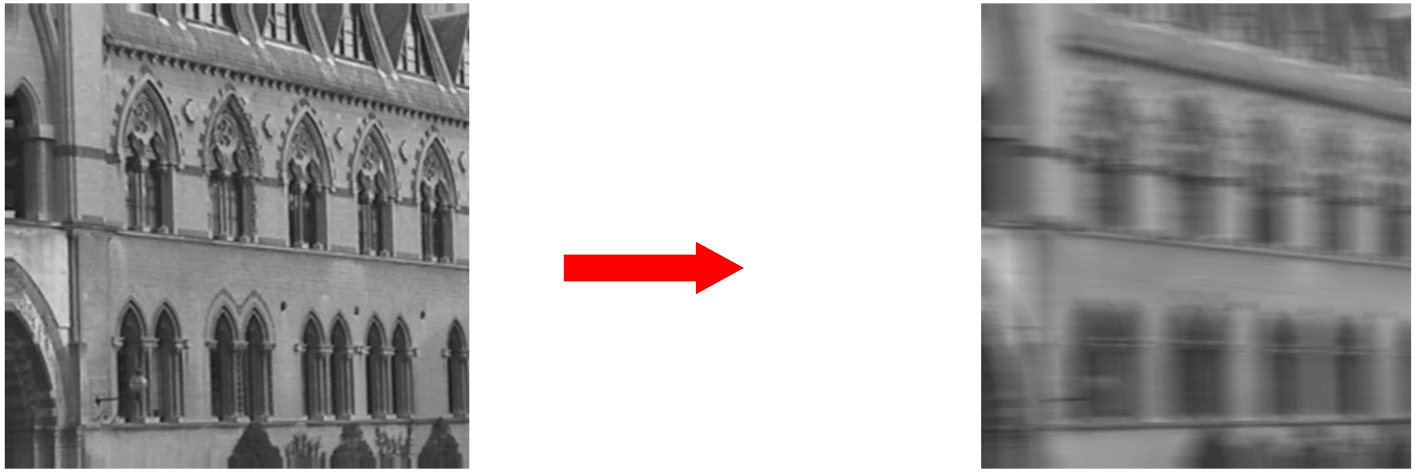
$$W^* = \frac{H|F|^2}{|H|^2|F|^2 + |N|^2}$$

$$W = \frac{H^*}{|H|^2 + |N|^2/|F|^2}$$

Note: filter is defined in the Fourier domain

Example 2: Motion deblurring

Suppose there is blur only in the horizontal direction
e.g. camera pans as image is acquired



Degradation model

$$g(x,y) = \frac{1}{T}\int_{-T/2}^{T/2} f(x - x_0(t), y)\, dt$$

Require *H(u,v)* for Wiener filter

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy,$$

$$= \frac{1}{T}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left\{\int_{-T/2}^{T/2} f(x-x_0(t),y)dt\right\}e^{-j2\pi(ux+vy)}\,dxdy$$

interchange order of spatial and temporal integration

$$G(u,v) = \frac{1}{T}\int_{-T/2}^{T/2}\left\{\underline{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-x_0(t),y)e^{-j2\pi(ux+vy)}\,dxdy}\right\}dt$$

Fourier transform of $f(x-x_0(t),y)$.

$$G(u,v) = \frac{1}{T}\int_{-T/2}^{T/2} F(u,v)e^{-j2\pi ux_0(t)}\,dt$$

$$= F(u,v)\frac{1}{T}\int_{-T/2}^{T/2} e^{-j2\pi ux_0(t)}\,dt$$

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

where

$$H(u,v) = \frac{1}{T}\int_{-T/2}^{T/2} e^{-j2\pi ux_0(t)}\,dt$$

suppose $x_0(t) = st$, and $sT = d$ pixels

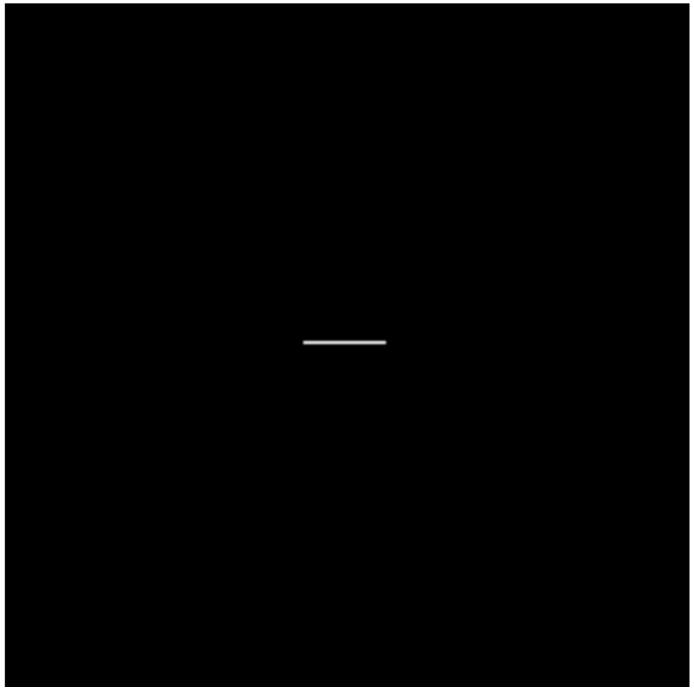
FT of … $h(x,y) = \text{hat}_d(x)\delta(y)$

$$H(u,v) = \frac{1}{T}\int_{-T/2}^{T/2} e^{-j2\pi ust}\,dt$$

$$= \frac{1}{T}\left[\frac{e^{-j2\pi ust}}{-j2\pi us}\right]_{-T/2}^{T/2}$$

$$= \frac{1}{j2\pi ud}\left(e^{j\pi ud} - e^{-j\pi ud}\right)$$

$$= \text{sinc}\pi ud$$

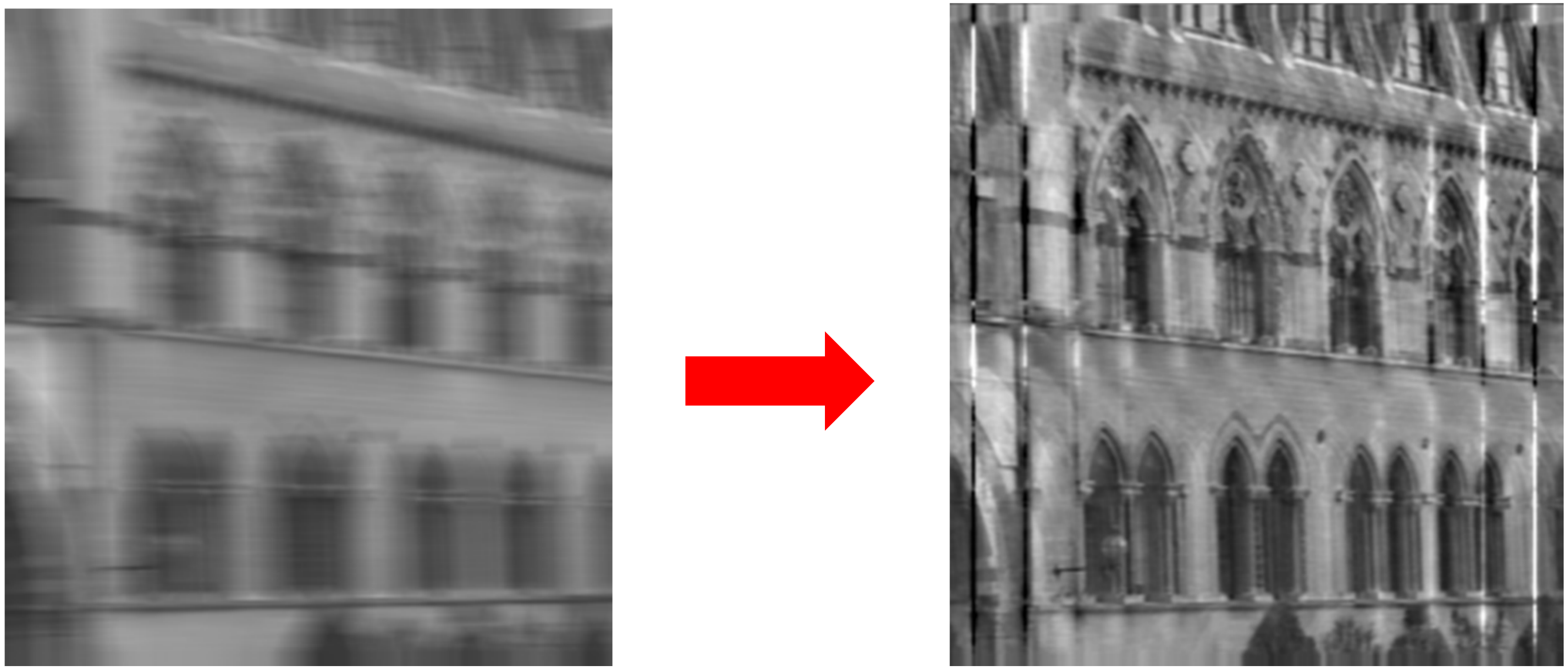


Note, *H(u,v)* has zeros – a problem for an inverse filter

Motion deblurring with a Wiener filter

blur = 20 pixels

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



1. Compute the FT of the blurred image
2. Multiply the FT by the Wiener filter
3. Compute the inverse FT

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

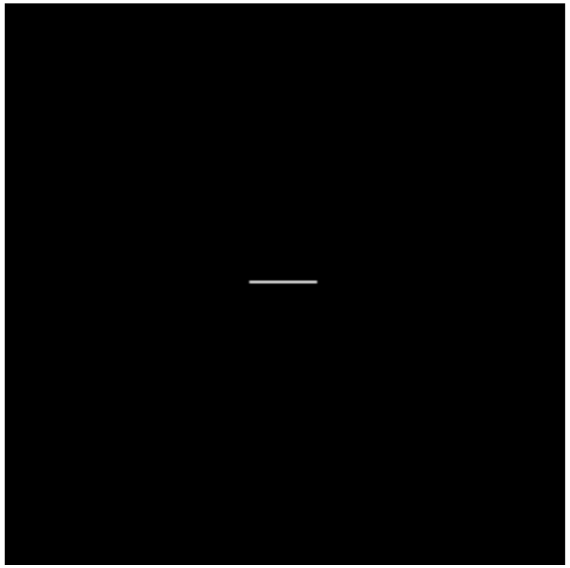
Application: Reading number plates

Algorithm

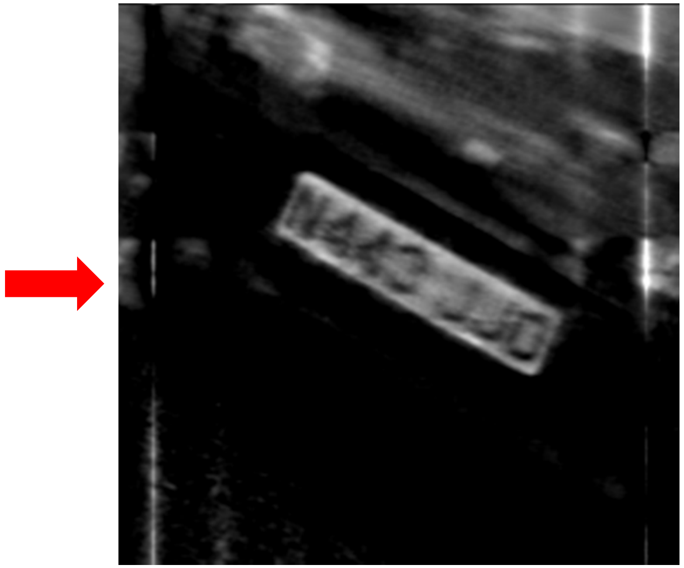
1. Rotate image so that blur is horizontal
2. Estimate length of blur
3. Construct a bar modelling the convolution
4. Compute and apply a Wiener filter
5. Optimize over values of K

$f(x,y)$

$h(x,y)$



$\hat{f}(x,y)$



blur = 30 pixels

Maximum a posteriori (MAP) Estimation

Generative model (forward process)

- original $f(x,y)$
- motion blur
- additive intensity noise

For an image with n pixels, write this process as

$$\hat{\mathbf{g}} = \mathtt{A}\mathbf{f} + \mathbf{n}$$

where $\hat{\mathbf{g}}$ and $\mathbf{f}$ are n-vectors, and $\mathtt{A}$ is an $n \times n$ matrix.

Inverse problem

- Estimate *f(x,y)* by optimizing a cost function:

observed image → $\mathbf{g}$; generated image → $\mathtt{A}\mathbf{f}$

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \underbrace{(\mathbf{g} - \mathtt{A}\mathbf{f})^2}_{\text{Likelihood/ loss function}} + \lambda \underbrace{p(\mathbf{f})}_{\text{prior/ regularization}}$$

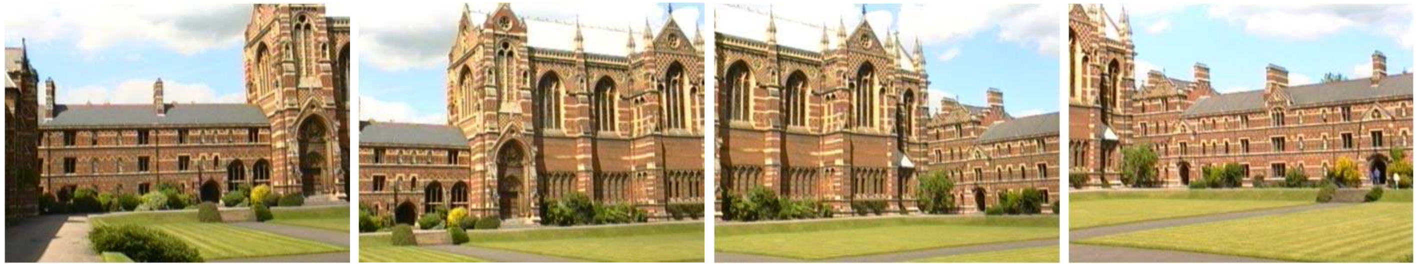
Example

$$p(f) = (\nabla \mathbf{f})^2$$

to suppress high frequency noise

Example 3: Super resolution

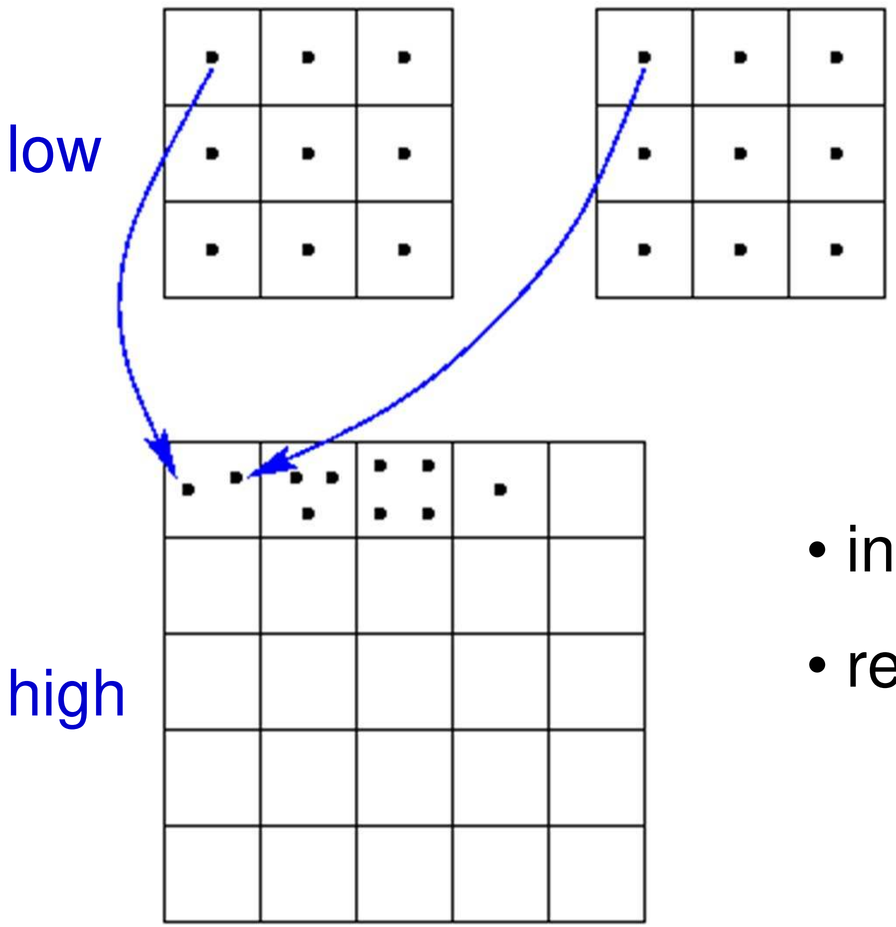
Suppose there are multiple images of the same scene
each displaced spatially …



After registration the samples are not coincident
and this may be used to defeat the Nyquist limit.

Intuitive model

Treat images as point samples



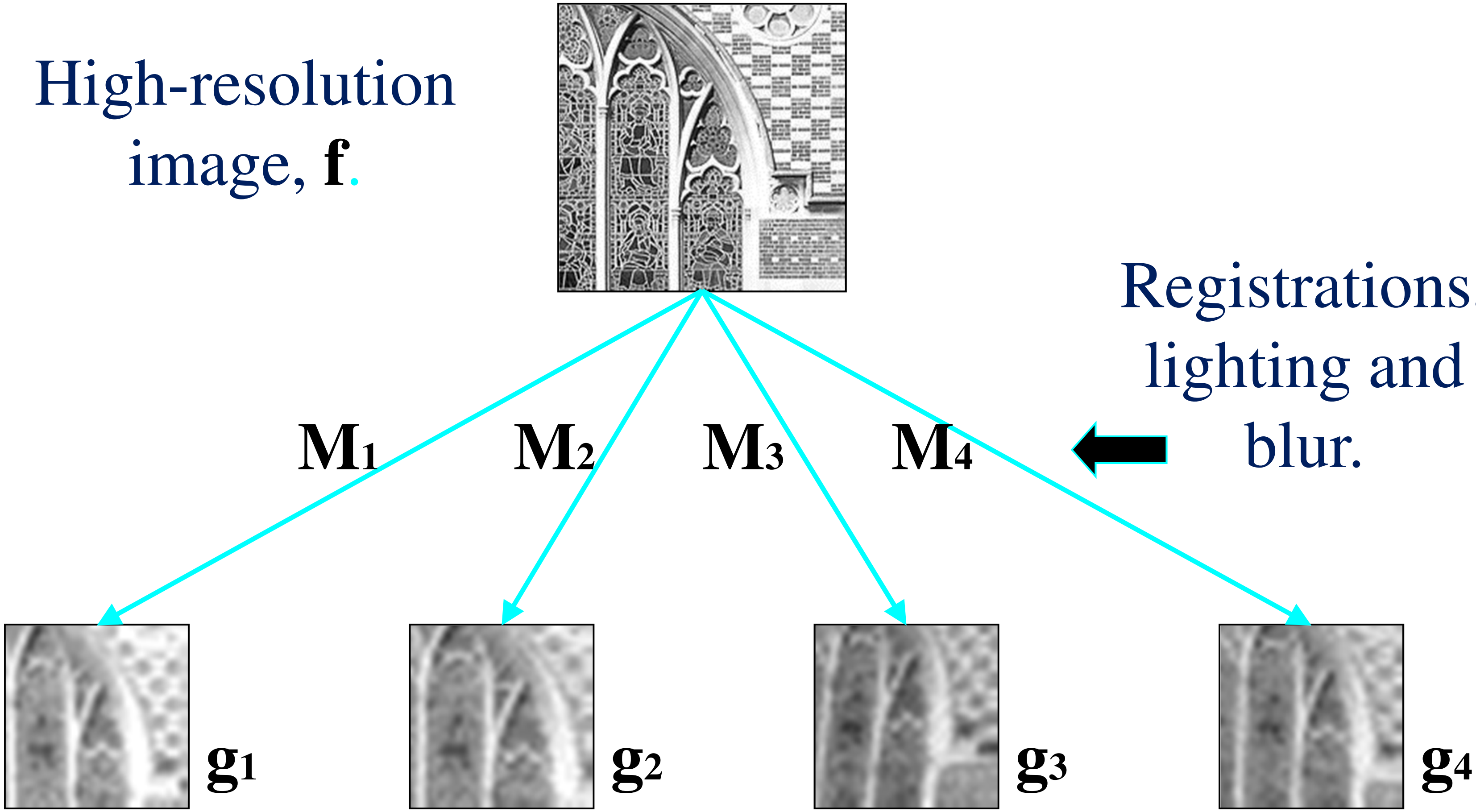
• • • more images

• increase resolution

• reduce noise

Generative Model

High-resolution image, **f**.



Registrations, lighting and blur.

Low-resolution images

Sketch solution

Non-examinable

- Estimate the super resolution image which minimizes the error between predicted and observed images.

Write the generative model for one image i as

$$\mathbf{g}_i = \mathtt{M}_i \mathbf{f} + \boldsymbol{\eta}_i$$

where $\mathtt{M}_i$ combines registration, lighting and down-sampling.

Finally the generative models of all N images are stacked vertically to form an over-determined linear system

$$\begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{N-1} \end{bmatrix} = \begin{bmatrix} \mathtt{M}_0 \\ \mathtt{M}_1 \\ \vdots \\ \mathtt{M}_{N-1} \end{bmatrix} \mathbf{f} + \begin{bmatrix} \boldsymbol{\eta}_0 \\ \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_{N-1} \end{bmatrix}$$

$$\mathbf{g} = \mathtt{M}\mathbf{f} + \boldsymbol{\eta}$$

Maximum a posterior estimation

The MAP estimator has the form:

$$\mathbf{f}_{\mathrm{MAP}} = \underset{\mathbf{f}}{\operatorname{argmin}} \left(\underbrace{(\mathbf{g} - \mathtt{M}\mathbf{f})^2}_{\text{likelihood}} + \lambda^2 \underbrace{p(\mathbf{f})}_{\text{prior}} \right)$$

Example

$$\mathbf{f}_{\mathrm{MAP}} = \underset{\mathbf{f}}{\operatorname{argmin}} \left((\mathbf{g} - \mathtt{M}\mathbf{f})^2 + \lambda^2 \sum_{\forall x,y} p(\nabla \mathrm{f}(x,y)) \right)$$

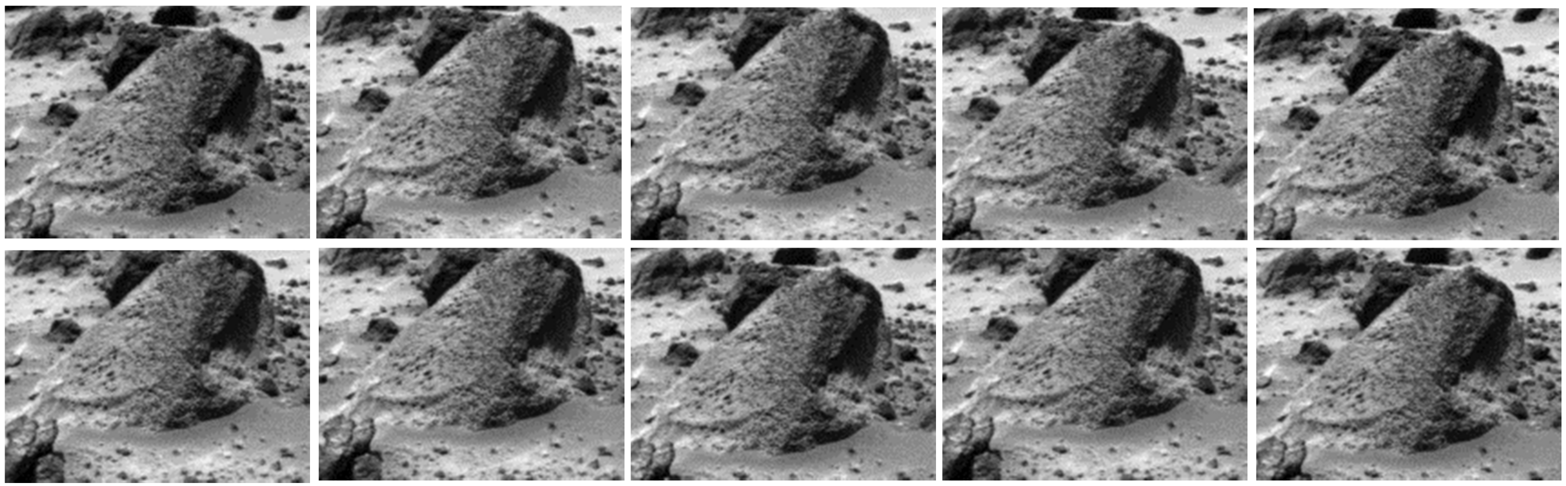
where the prior $p(x)$ is defined by the Huber function,

$$\begin{aligned} p(x) &= x^2 &&, \text{if } x \leq \alpha \\ &= 2\alpha\,|x| - \alpha^2 &&, \text{otherwise} \end{aligned}$$

Super resolution example I: Mars

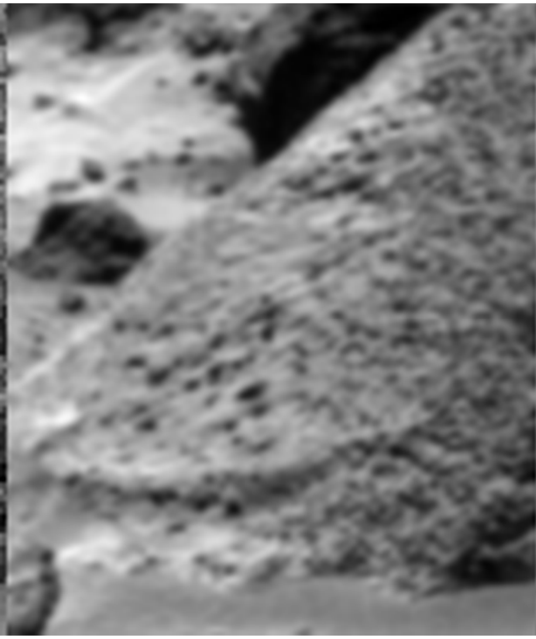
25 JPEG images courtesy of the Mars lander

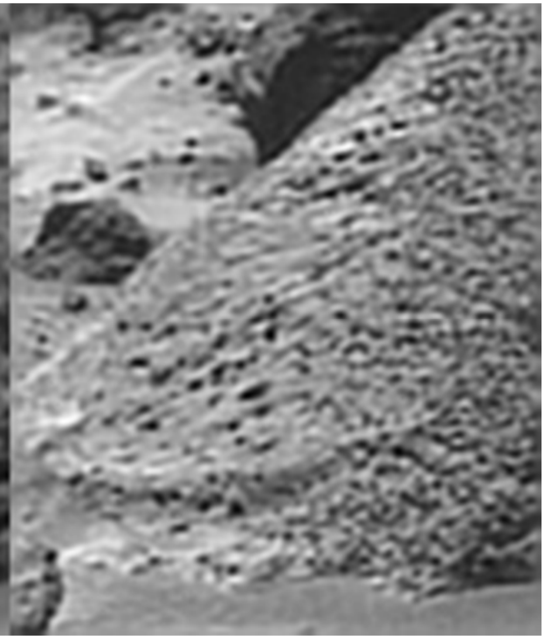
images are from different sweeps of a rotating camera



Super resolution result


Original frame


Average image


Super-resolution

2x zoom from 25 JPEG images.

Super resolution example II: car sequence

rotating DV camera

Mosaic

Super-resolution result for ROI

85 JPEG images

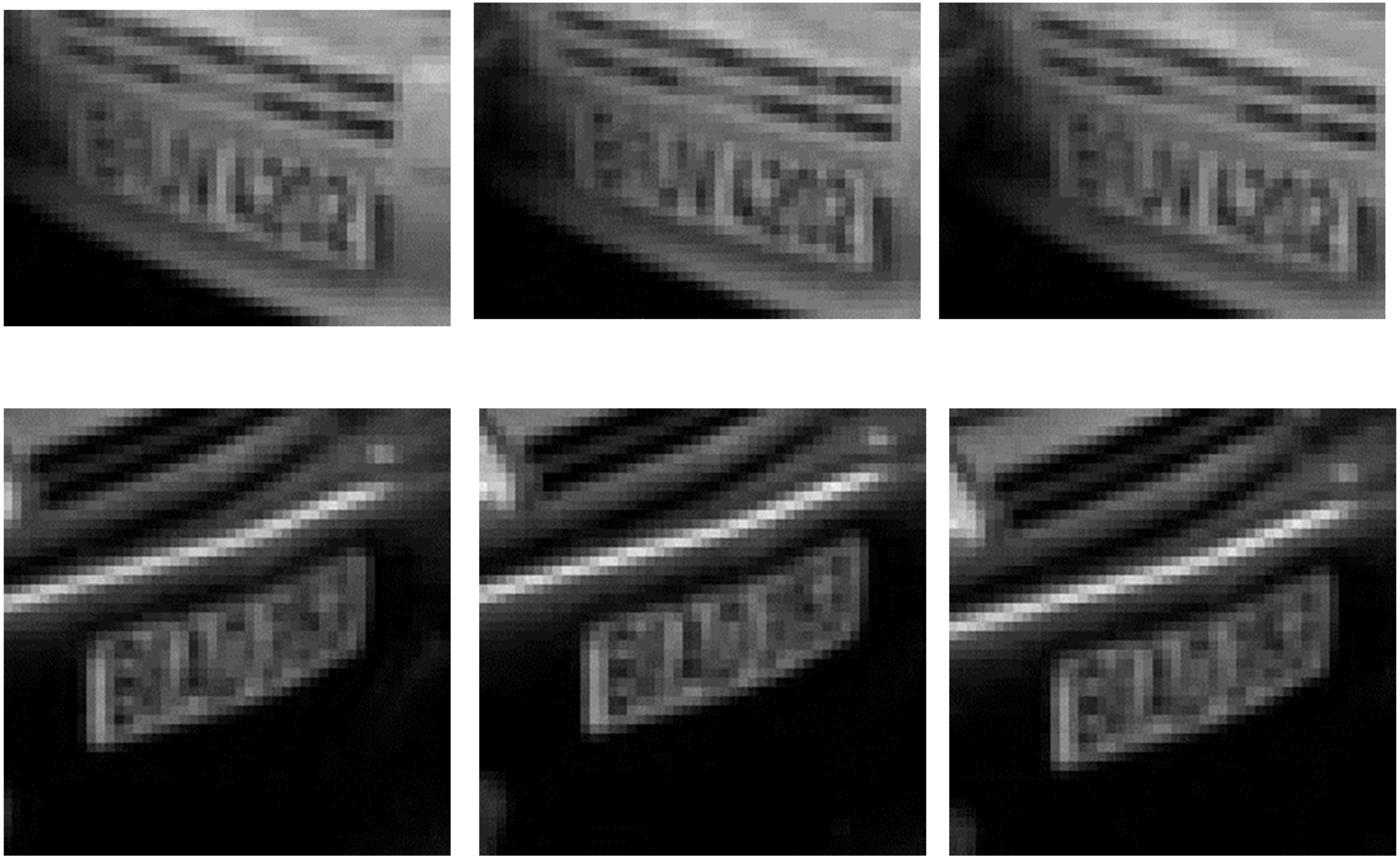
original ROI

35 x 20 pixels

four times resolution

Super resolution example III: Run Lola Run

Input – low resolution



Super-resolution output

Blind deblurring   Non-examinable

So far we have assumed that we know the generative model, e.g.

$$\mathbf{g} = A(\mathbf{h})\,\mathbf{f}$$

$$G = H\,F$$

i.e. that $h(x,y)$ is known, so that given the observed image $g(x,y)$, then the original image $f(x,y)$ can be estimated (restored)

Consider if only the observed image $g(x,y)$ is known. This is the problem of blind estimation.

Blind deblurring continued

- Estimate *f(x,y)* and *h(x,y)* by optimizing a cost function:

observed image → $\mathbf{g}$; generated image → $\mathtt{A}(\mathbf{h})\,\mathbf{f}$

$$\min_{\mathbf{f},\mathbf{h}} \underbrace{(\mathbf{g} - \overbrace{\mathtt{A}(\mathbf{h})\,\mathbf{f}}^{\text{generated image}})^2}_{\text{Likelihood/ loss function}} + \lambda \underbrace{p_f(\mathbf{f})}_{\text{image prior}} + \mu \underbrace{p_h(\mathbf{h})}_{\text{blur prior}}$$

Likelihood/ loss function

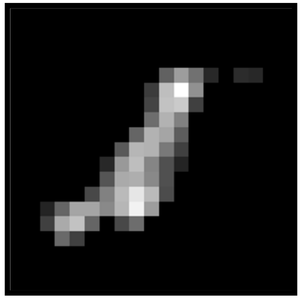
image prior

blur prior

Example I: Blind deblurring

blurred image

estimated blur filter



restored image

More examples of blind deblurring