Image Restoration

Computer Vision – Lecture 04

Practicals: weeks 3,4,6,7

Further Reading

- Slides from A Zisserman
- Slides from A Efros
- Slides from A Torralba and A Oliva
- Video from S Seitz: Fourier Transform in 5 minutes: The Case of the Splotched Van Gogh, Part 3

https://www.youtube.com/watch?v=Vxq9yj2pVWk

Image Restoration

- In contrast to image enhancement, in image restoration the degradation is modelled.
- This enables the effects of the degradation to be (largely) removed.
- The objective is to restore a degraded image to its original form.

Typical Degradations

Original

Optical Blur

Motion Blur

Spatial quantization

Additive noise

Modelling Degradation

We can model an observed image as:

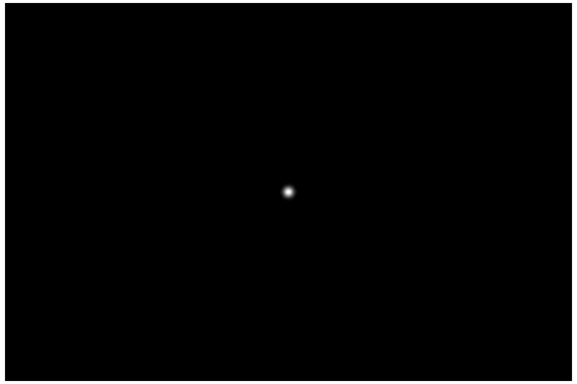
$$g(x,y) = \iint d(x-u, y-v) f(u,v)\, du\, dv + n(x,y)$$

This means the observed image g is created from the true image f through a convolution with d and added noise n.

$g(x,y)$

$=$

$d(x,y)$

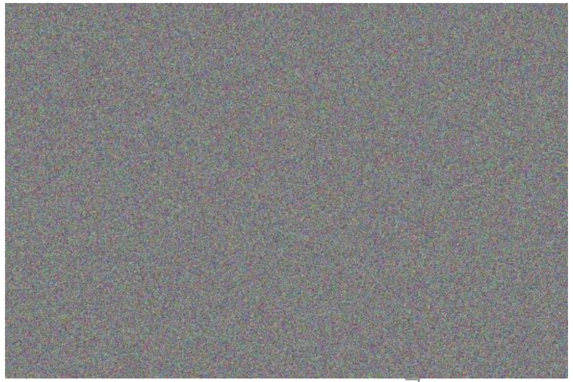


$*$

$f(x,y)$

$+$

$n(x,y)$



Degradation Model

$$g(x,y) = \iint d(x-u, y-v) f(u,v) du\, dv + n(x,y)$$

- This is only one way of modelling degradation. Others are possible too.

- $d(x,y)$ is called the *impulse response* or *point spread* function of the imaging system.

Fourier Transforms

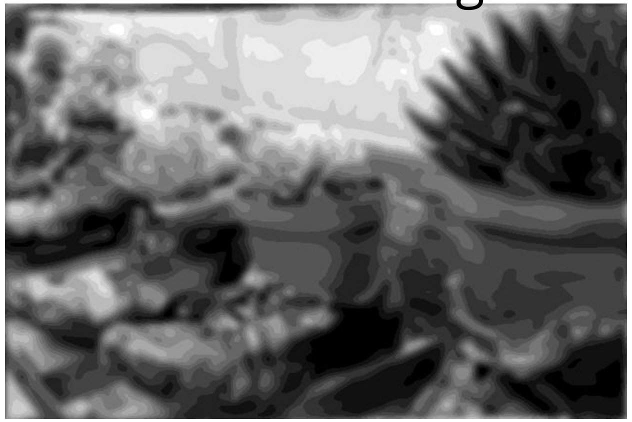
Image

$*$

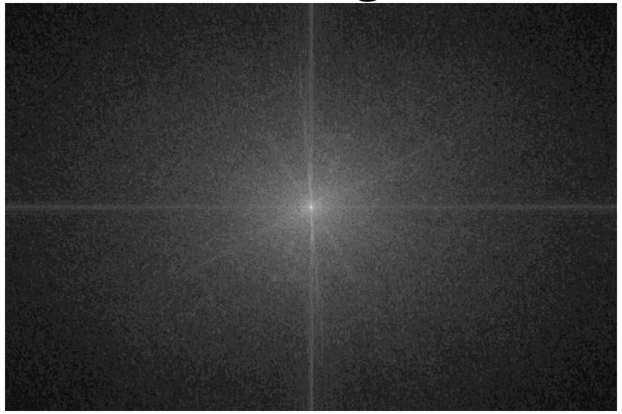
Filter

$=$

Filtered image

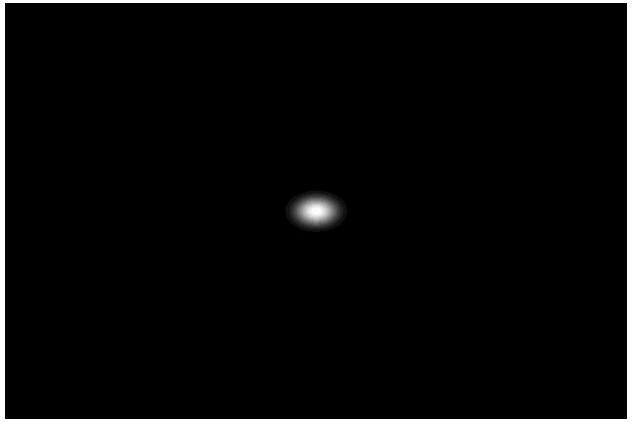


FT(Image)



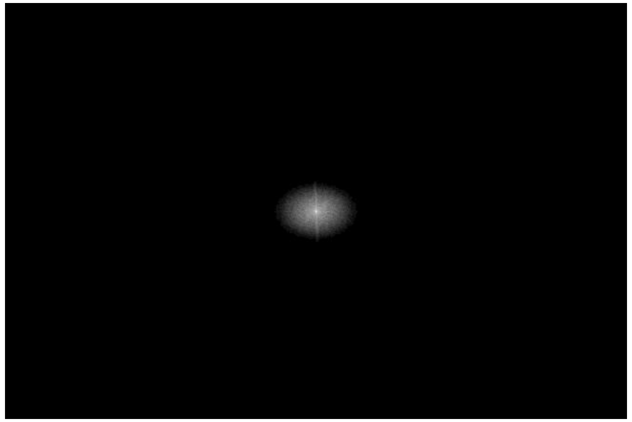
$\times$

FT(Filter)



$=$

FT(Filtered image)



Fourier Transforms

- Ignoring additive noise for now.
- Instead of a convolution in image space, we can use multiplication in Fourier space.

$$g = d * f \quad \rightarrow \quad G = D \times F$$
$$G = FT(g),\ D = FT(D),\ F = FT(F)$$

- Recover an estimate $\hat{f}$ of the true image:

$$\hat{f} = FT^{-1}\left(\frac{G}{D}\right) = FT^{-1}\left(\frac{FT(g)}{FT(d)}\right)$$

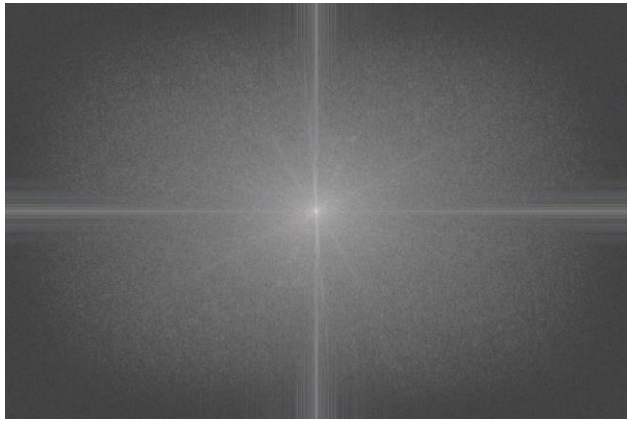
Filter Inversion

Blurred Image

Filter

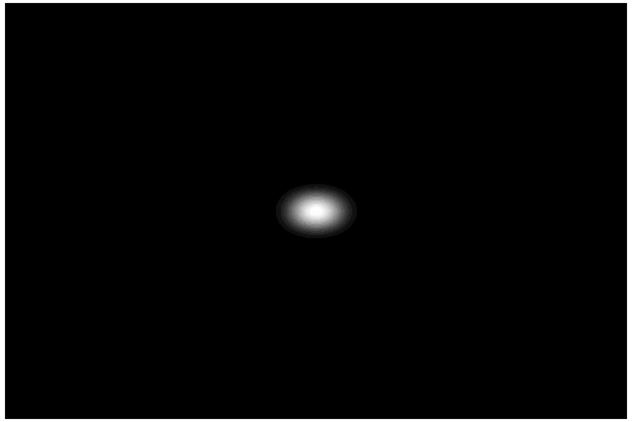
“deblurred” image

FT(Blurred Image)



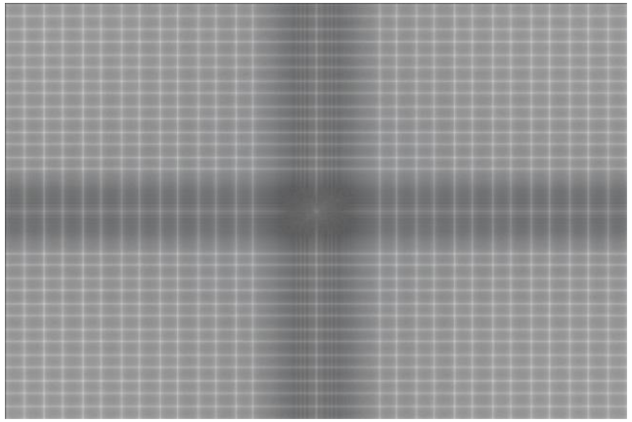
/

FT(Filter)



=

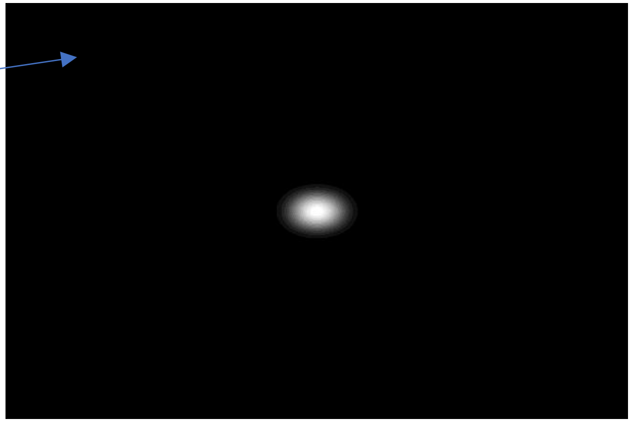
FT(deblurred image)



What went wrong?

- The deblurred image is almost purely noise
- This is because we are dividing by small numbers
- The filter is almost 0 in the high-frequency regions
- We are amplifying the noise!

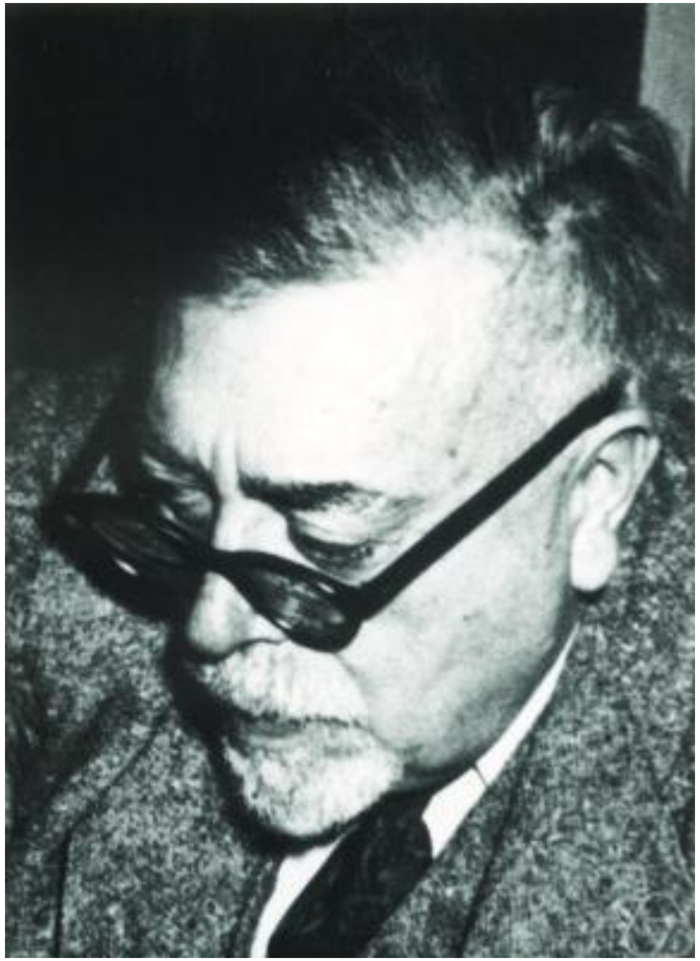
Zoom into “deblurred image”



FT(Filter)

The Wiener Filter

- To avoid up-scaling the noise, add a second constraint
- Minimise: $\mathbb{E}\left(\left|g - \hat{f}\right|^2\right)$
- The reconstruction should be close to the observation



Norbert Wiener 1894-1964

The Wiener Filter

- In Fourier space we had

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

- Bad solution (noise amplification):

$$\hat{F}(u,v) = \frac{1}{D(u,v)}\, G(u,v)$$

- Wiener Filter:

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

The Wiener Filter

complex conjugate

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

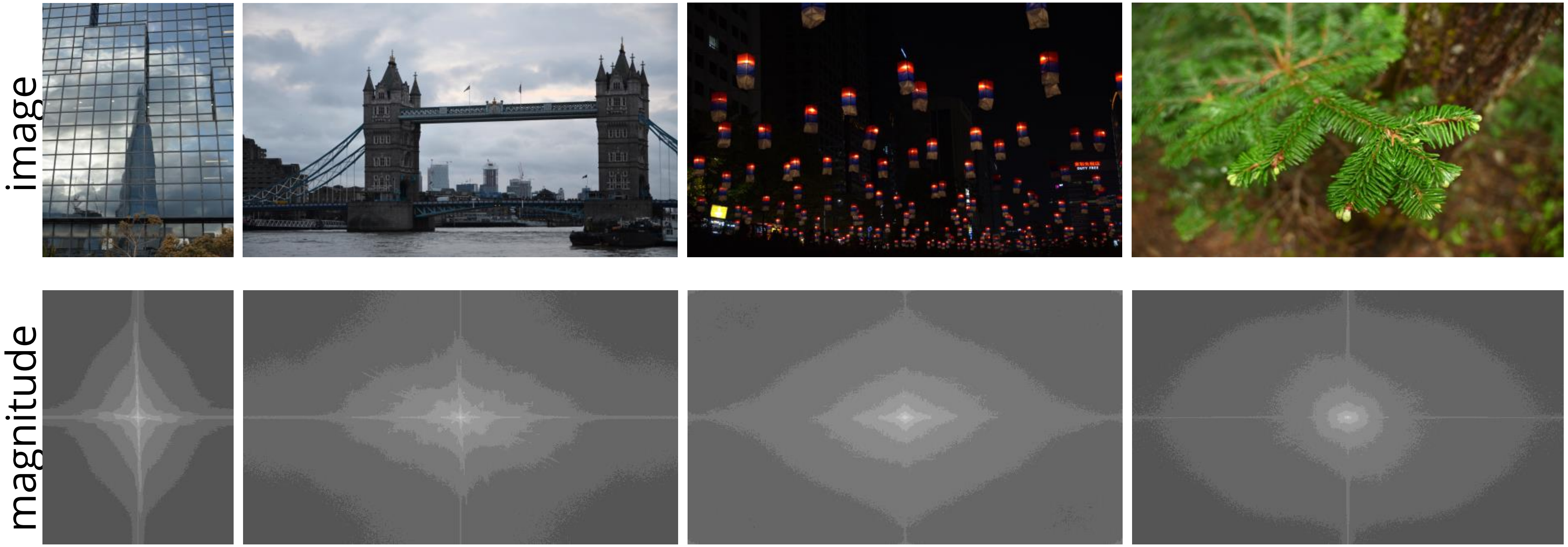
- $S(u,v) = \mathbb{E}(|F(u,v)|^2)$ is the mean power spectral density of the original signal
- $K(u,v) = \mathbb{E}(|N(u,v)|^2)$ is the mean power spectral density of the noise

- Often: $S(u,v) = 1$ and $K(u,v)$ is a small (real) constant.

Energy Spectral Density

- Power spectrum: distribution of total energy across frequencies.

- This is the magnitude diagram that we have been looking at.



Energy Spectral Density

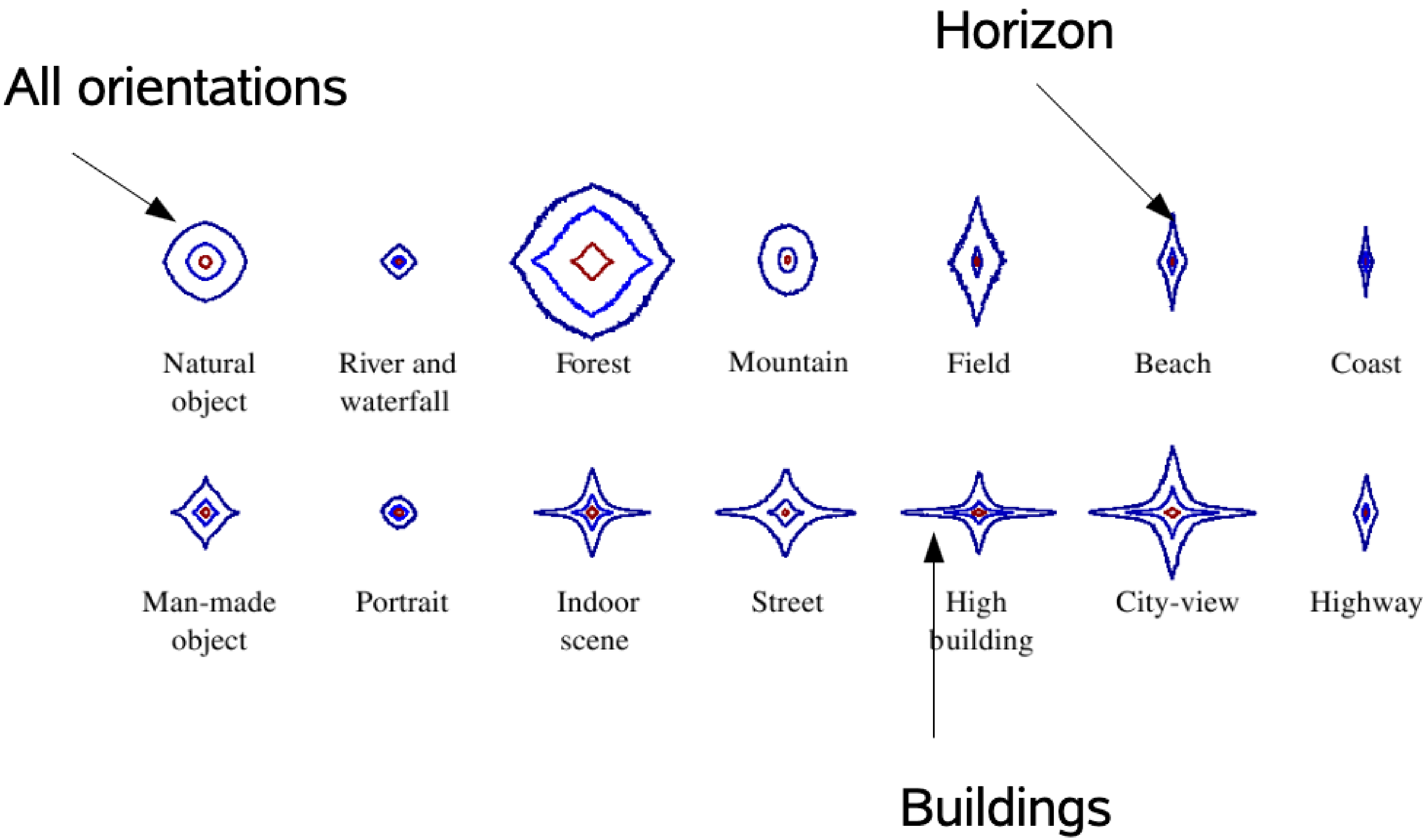


Image from [A Torralba and A Oliva]

Interpretation

We can rewrite the filter as:

$$W(u,v) = \frac{1}{D(u,v)} \left[\frac{1}{1 + \frac{1}{|D(u,v)|^2 SNR(f)}} \right]$$

Filter inversion

Scaling factor

Signal-to-noise ratio

Intuition: we invert the filter, but scale inversely with the expected noise, so that we do not end up amplifying it.

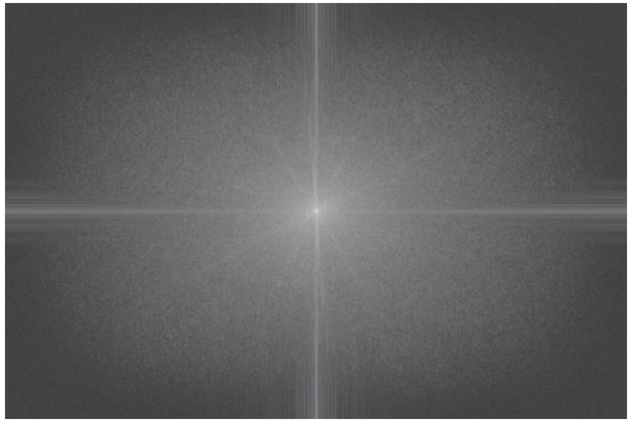
Filter Inversion

Blurred Image

Filter

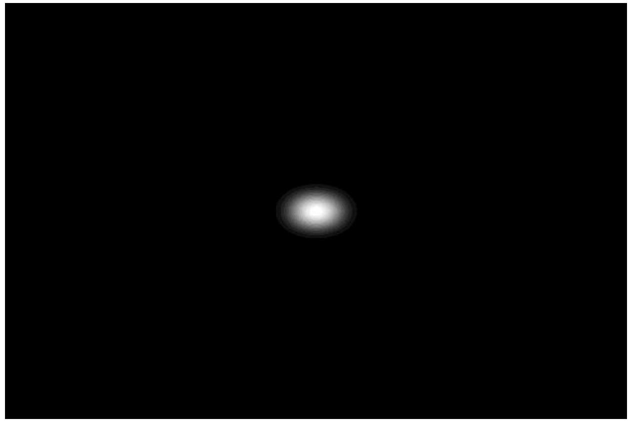
“deblurred” image

FT(Blurred Image)



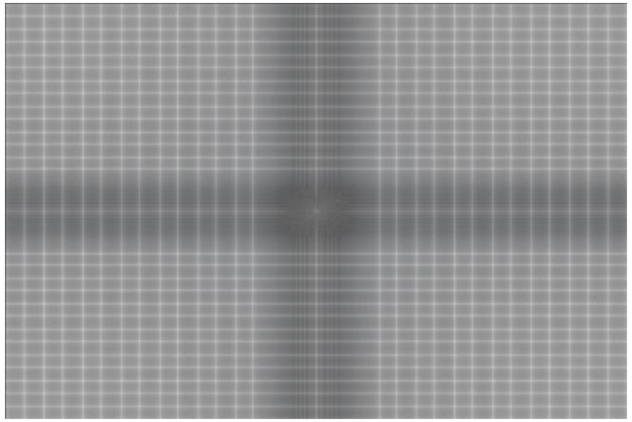
/

FT(Filter)

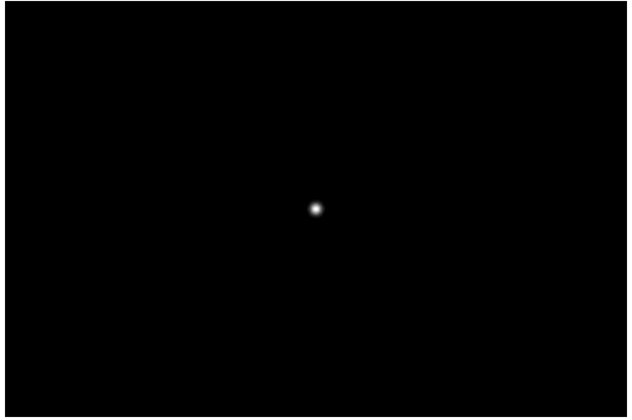


=

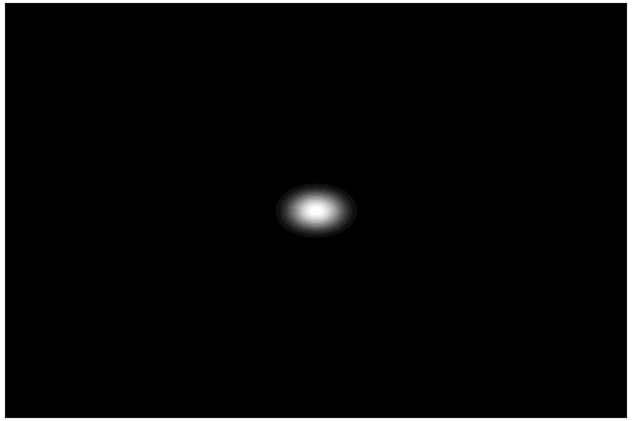
FT(deblurred image)



Deblurring



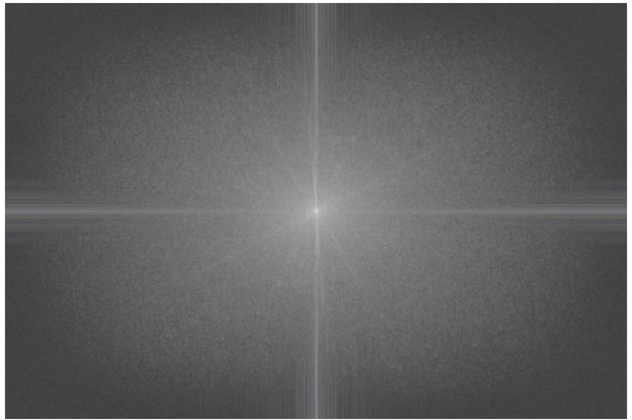
FT(Filter)



Blurred Image

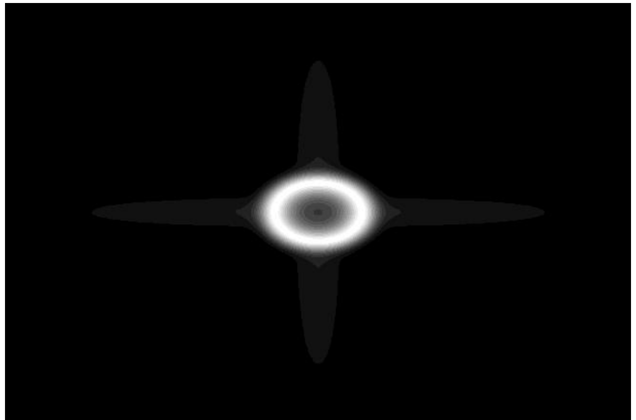
Deblurred Image

FT(Blurred Image)



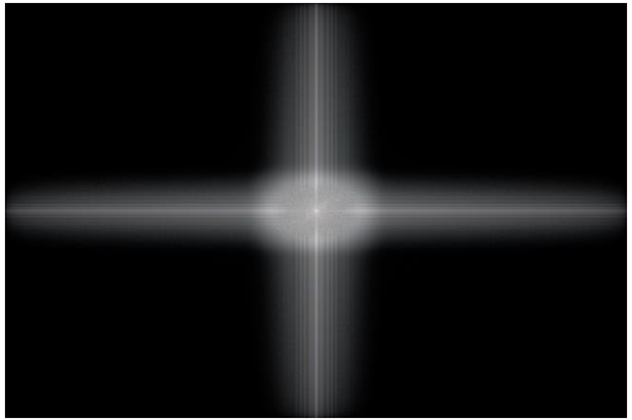
*

Wiener Filter



=

FT(deblurred image)



Deblurring with a WF

- Boundary artifacts because DFT assumes infinitely tiled image.

- In practice, we do not know the filter that degraded the image!

Deblurring with WF

Blurred Image

Original

WF with varying σ

Finding WF Parameters

- It is difficult to find good parameters automatically, but easy for humans to see.
- Photoshop has a “strength” slider.
- Bayesian optimization:

Bayesian estimation of regularization and point spread function parameters for Wiener-Hunt deconvolution

François Orieux, Jean-François Giovannelli, Thomas Rodet

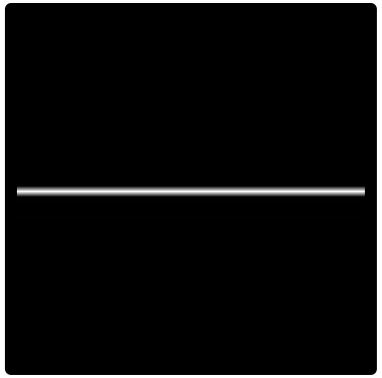
Motion Blur and WFs

Motion blur can be modelled as a convolution with a line segment filter.

Algorithm to remove motion blur:

1. Rotate image so that blur is horizontal.
2. Estimate length of blur.
3. Construct line segment filter.
4. Compute and apply Wiener filter.

line segment filter



Simulated motion blur

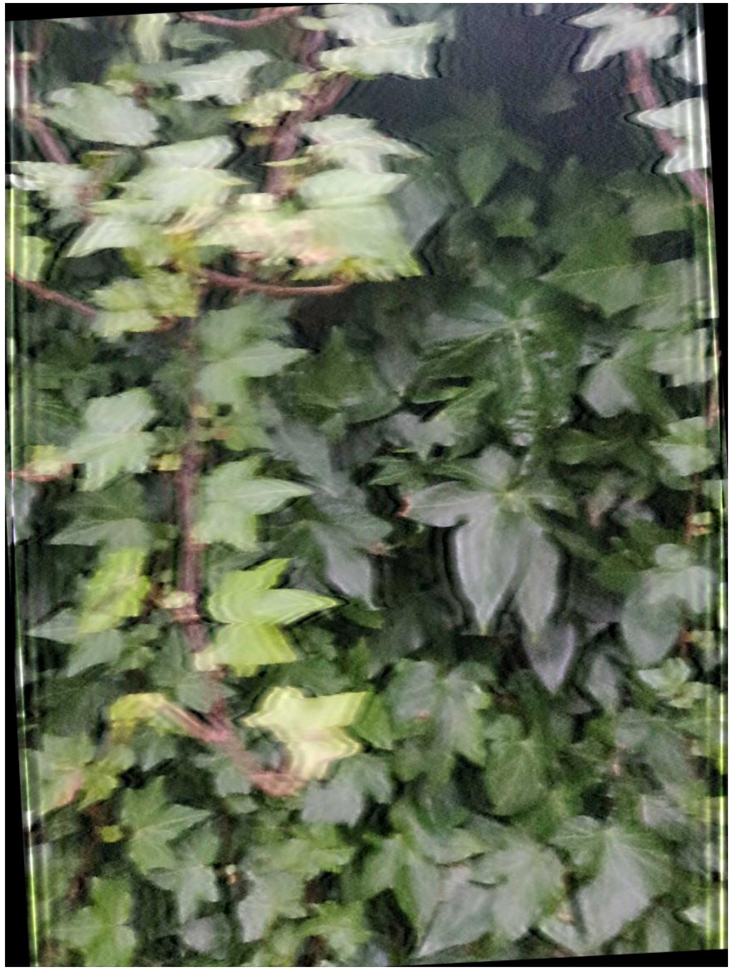
Deblur Example

Needs guessing the blur that was "applied":

My guess:

- Angle: 3 deg
- Blur: 18px
- Noise: 0.03

original



deblurred

Generative Models

Instead of convolutions we can formulate the degradation as a linear operation on pixels.

$$g = Af + n$$

- For an image with N pixels, the true image f and the observed image g can be written as N-vectors.
- A is an $N \times N$ matrix.
- n is an N-vector of noise.

Inverse Problem

We can estimate the true image by optimising a cost function:

$$\hat{f} = \mathrm{argmin}_f[(g - Af)^2 + \lambda p(f)]$$

- Af is a generated image. We minimise the difference to g.
- $p(f)$ is a prior or regulariser for the optimisation.
- λ is a weight, controlling the influence of regularisation.

- For example: $p(f) = (\nabla f)^2$ (∇f is the *gradient image*)

Inverse Problem

- A can affect each pixel individually and is thus more flexible than convolution.

- A needs to be manually defined and depends on the problem we are solving.

Inverse Problem: Super Resolution

If we register multiple images of the same scene (how: later), we get multiple samples per pixel!

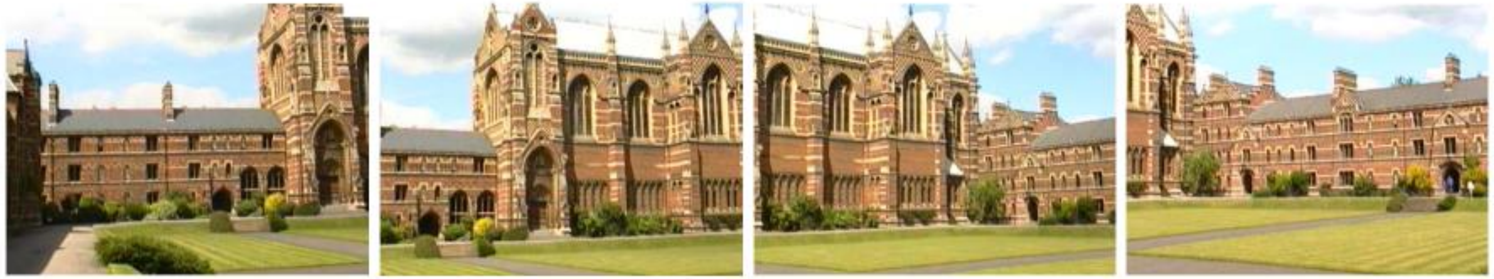
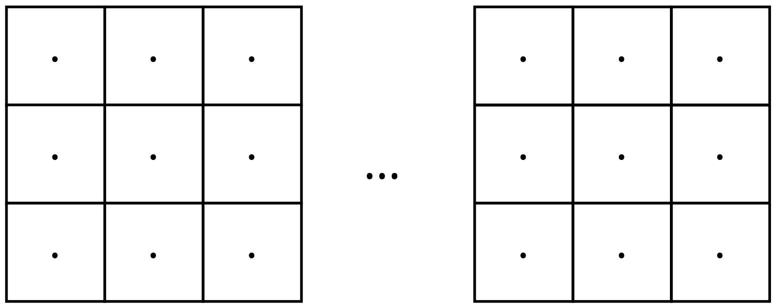


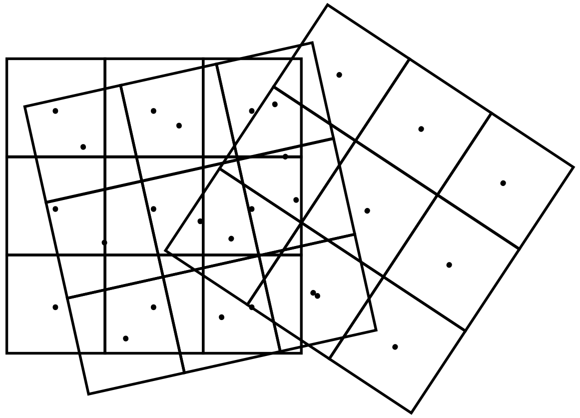
Image from A Zisserman

Super Resolution

- Pixels as samples.
- After registration, samples will not align perfectly.
- Can treat it as higher sampling rate (Shannon-Nyquist).
- Higher resolution estimate is possible.
- Construct A matrix based on bi-linear interpolation weights.



many images



multiple samples per pixel

Super Resolution

Images from Mars lander: rotating camera – capture the same frame multiple times

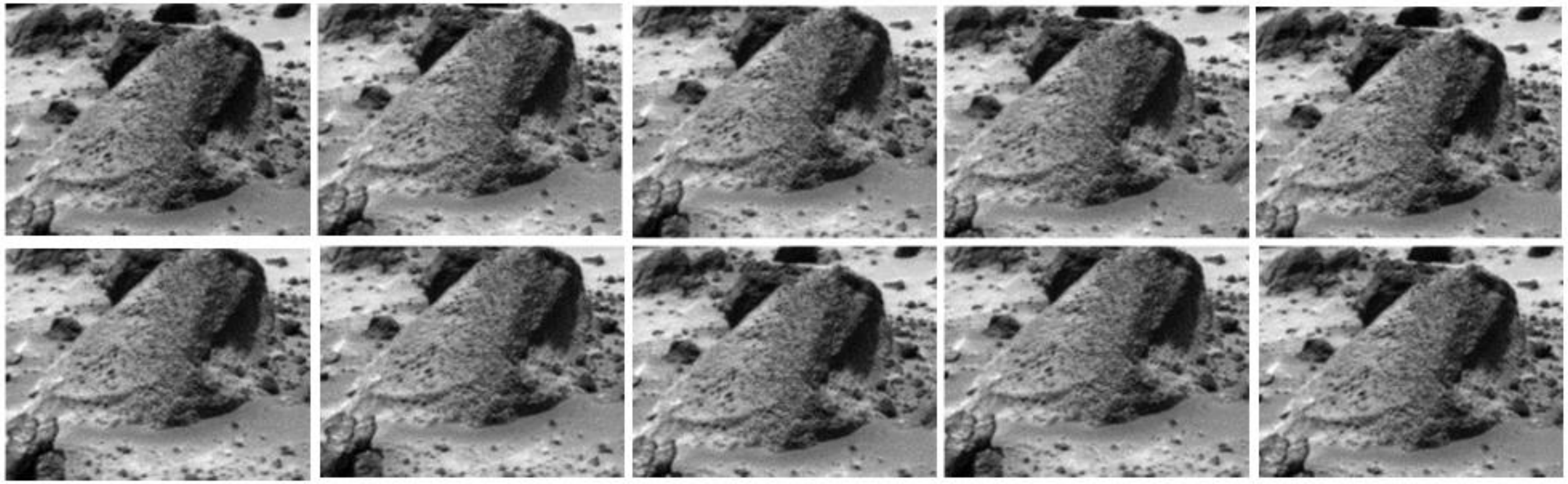
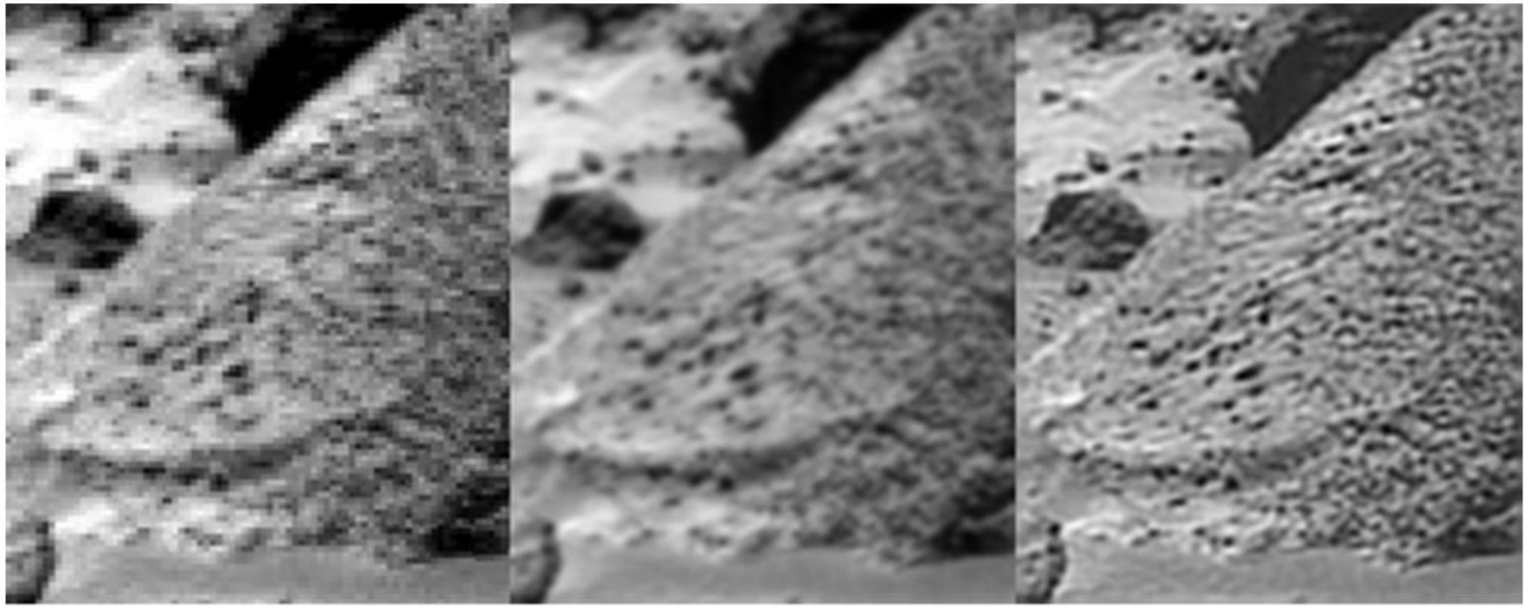


Image from A Zisserman

Super Resolution

- 2x super resolution from 25 JPEG images
- JPEG compression artefacts largely removed



Original frame Average image Super-resolution

Image from A Zisserman

Mirror Average Example

- A virus has corrupted our image! It was blended with a mirrored version (40% old vs. 60% mirrored) and noise was added.

- $\hat{f} = \mathrm{argmin}_f [(g - Af)^2 + \lambda p(f)]$

- How does A look like?

Mirror Matrix

- Construct a matrix M that mirrors an image of size $H \times W$.
- We represent the image as a vector containing its pixels.
- How do we flip a row of pixels horizontally?

$$K = \begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \iddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix}, \qquad K \in \mathbb{R}^{W \times W}$$

- Now we can build M as a block-diagonal matrix of Ks.

$$M = \mathrm{diag}_{i=1}^{\mathrm{H}}(K)$$

$$M = \begin{bmatrix} K & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & K \end{bmatrix}, \qquad M \in \mathbb{R}^{HW \times HW}$$

Image Priors

- We only have noisy observations.

- Assume a prior (models what a reasonable image is):
- For example smoothness: $p(f) = \sum_{x,y} |\Delta_x| + |\Delta_y|$

- $\nabla f(x, y) = \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$ is the image gradient. Can be computed with the Sobel Operator or finite differences.

Optimisation

- We assume that our image was made with $A = \frac{4}{10} I + \frac{6}{10} M$

- Find: $\hat{f} = \mathrm{argmin}_f E(f)$
- $E(f) = (g - Af)^2 + \lambda p(f)$

- Can be done with gradient descent on f:
    - Compute the gradient ∇E of $(g - Af)^2 + \lambda p(f)$ wrt. f
    - Update $\hat{f} \leftarrow \hat{f} + \mu \nabla E$

Optimisation

Blind Deblurring

- So far, we were guessing the deblurring parameters.
- The inverse formulation is quite flexible! We can also optimise a blur filter h.

$$\hat{f}, \hat{h} = \text{argmin}_{f,h}\left[(g - A(h)f)^2 + \lambda_f p_f(f) + \lambda_h p_h(h)\right]$$

observed image

image prior

generated image

blur prior

Blind Deblurring

- This is even more under-constrained.
- Needs to rely on good priors or multiple observations.

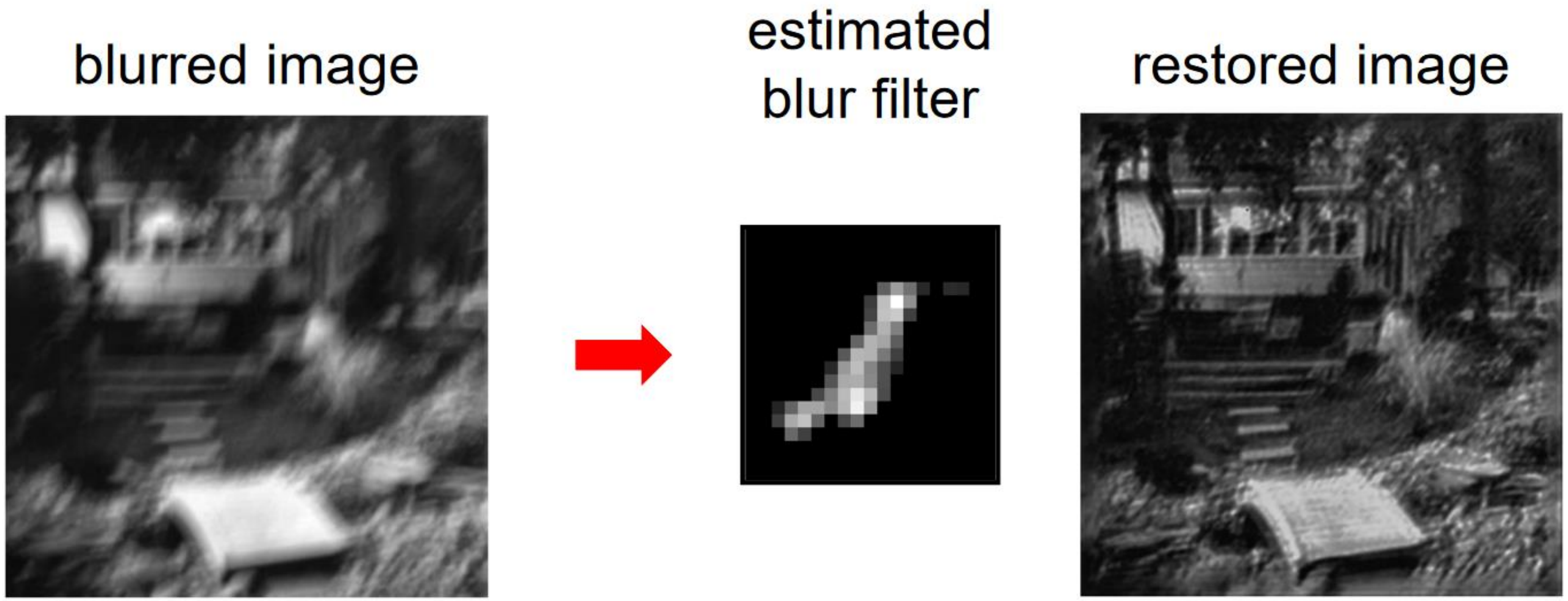


Image from A Zisserman