# Restoration

# Outline

## Restoration model

- What is *restoration*
- Typical image *degradations*
- A *model* for degradation/restoration
- The *point spread function* (PSF)

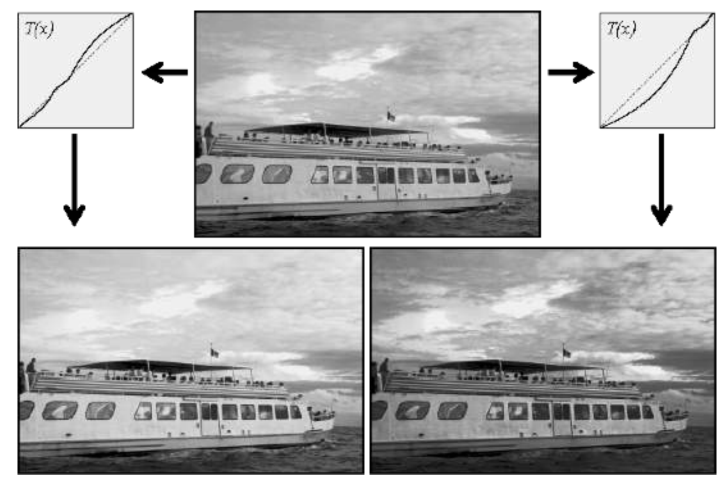
## Algorithms

- Special case: only noise
- *Direct inverse filtering*
- *Wiener* deconvolution (linear approach)
- *Lucy-Richardson* deconvolution (non-linear approach)

# What is restoration?

- **GOAL**: improve an image in some predefined sense


What's the difference with before?



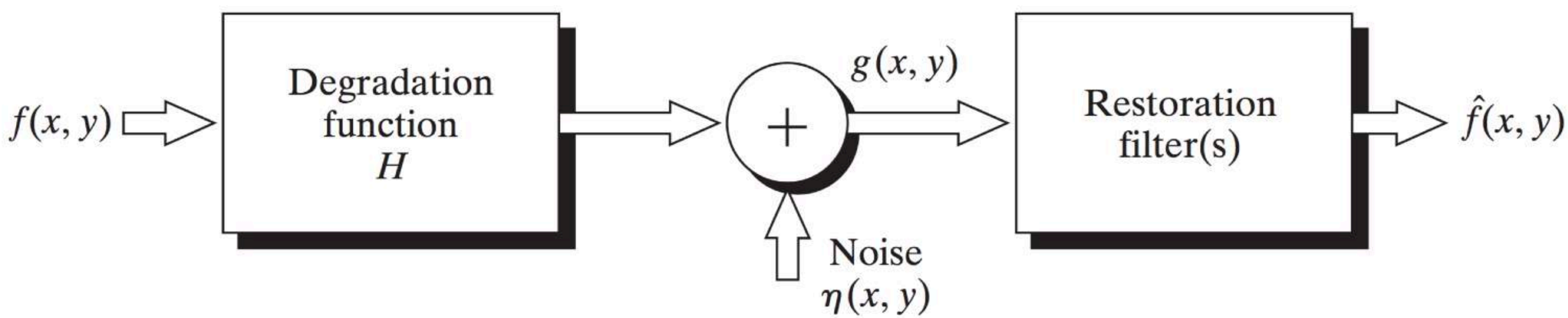

- **Image enhancement**: *subjective* process
  - Modify image to make it look better, e.g. contrast
  - **Heuristic/subjective** process: try until pleased
- **Image restoration**: *objective* process
  - An image is **degraded** by external factors acting on the scene/sensor system
  - Remove them by **modeling the degradation** and applying the **inverse process**

# A model of image degradation/restoration

- **Degradation** is modeled as a process $H$ that operates on the *input image* $f(x,y)$ to produce a *degraded image* $g(x,y)$

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


- **Restoration** attempts to obtain an estimate $\hat{f}(x,y)$ of the original image $f(x,y)$ **given some prior knowledge** on the *degradation process* $H$ and the *additive noise* $\eta(x,y)$



# A model of image degradation/restoration (2/4)

## Typical image degradations

- Noise
- Optical blur
  e.g. imperfect optics
- Motion blur
  e.g. camera motion


original


noise


optical blur


motion blur

## **Example**: restoration in *forensic science*

**identify a license plate**
from *low quality images*
e.g. bad camera, far objects, air pollution…

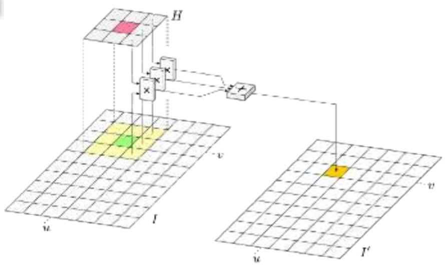
**identify a license plate**
from *moving cars*

# A model of image degradation/restoration (3/4)

- If $H$ is a **linear, position-invariant process**, it can be shown that the degraded image $g(x,y)$ is given in the *spatial domain* by

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

  - $h(x,y)$ is the *spatial representation* of the degradation process
  - The symbol “*” represents the **convolution**



- From the **convolution theorem** we know…

$$\mathrm{F}\{f * g\} = \mathrm{F}\{f\} \cdot \mathrm{F}\{g\} \qquad \mathrm{F}\{f \cdot g\} = \mathrm{F}\{f\} * \mathrm{F}\{g\}$$

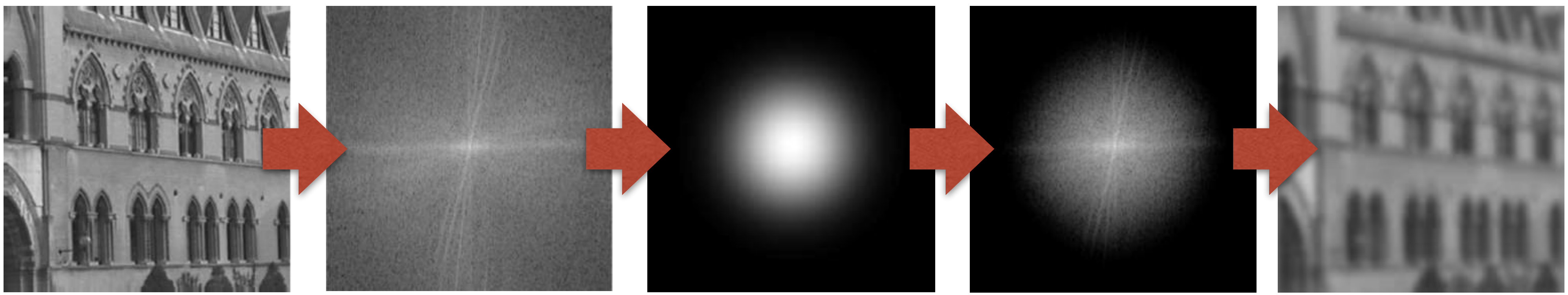
  - *“…convolution in one space = multiplication in the other one…”*

- The model can be express also in *frequency domain* as:

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

# A model of image degradation/restoration (4/4)

- <u>Note 1</u>: **challenging process**, loss of information and noise
  - Blurring is a *low-pass filter* ➜ removes high frequencies ➜ information is lost
  - *Noise* make the inversion of the process unstable



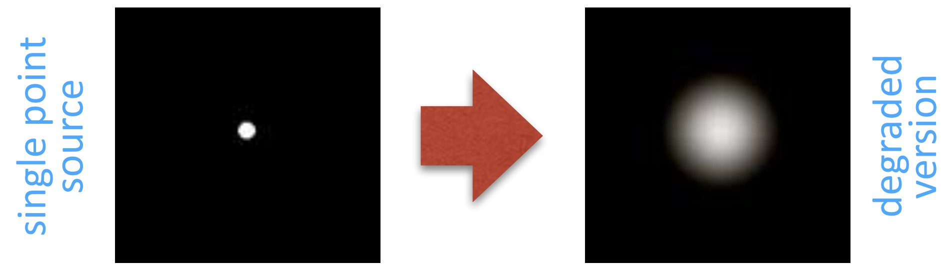
- <u>Note 2</u>: restoration relies on the assumption that $H$ **is the same everywhere** *(“…H is linear, position-invariant…”)*
  - Cannot be applied on images with *different out-of-focus*
  - Cannot be applied if different areas are subject to *different types of motion*

    e.g. *Running person*, arms and legs move differently

    Combination of *motion* and *rotation*

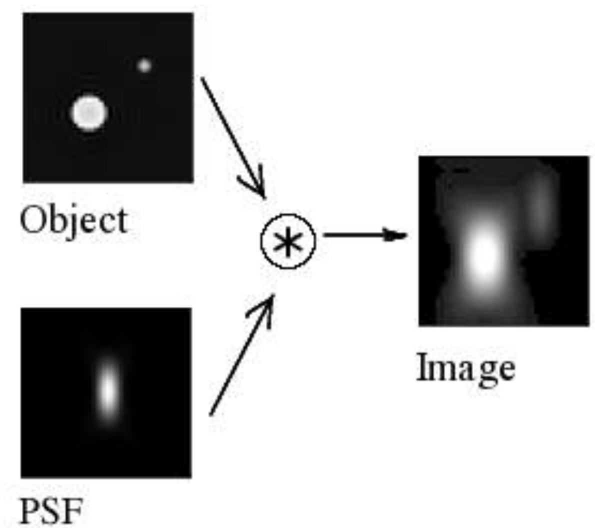
# The point spread function (PSF)

- In spatial domain, $h(x,y)$ is called **Point Spread Function** (PSF)
  - Represents the *spreading that a single bright point* in the ideal image would result in the degraded image
  - i.e. effect of degradation on a *single point*

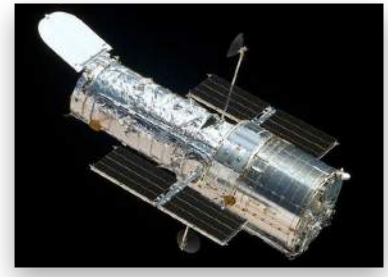



- This gives the **characteristics/effect of the degradation** for any type of input

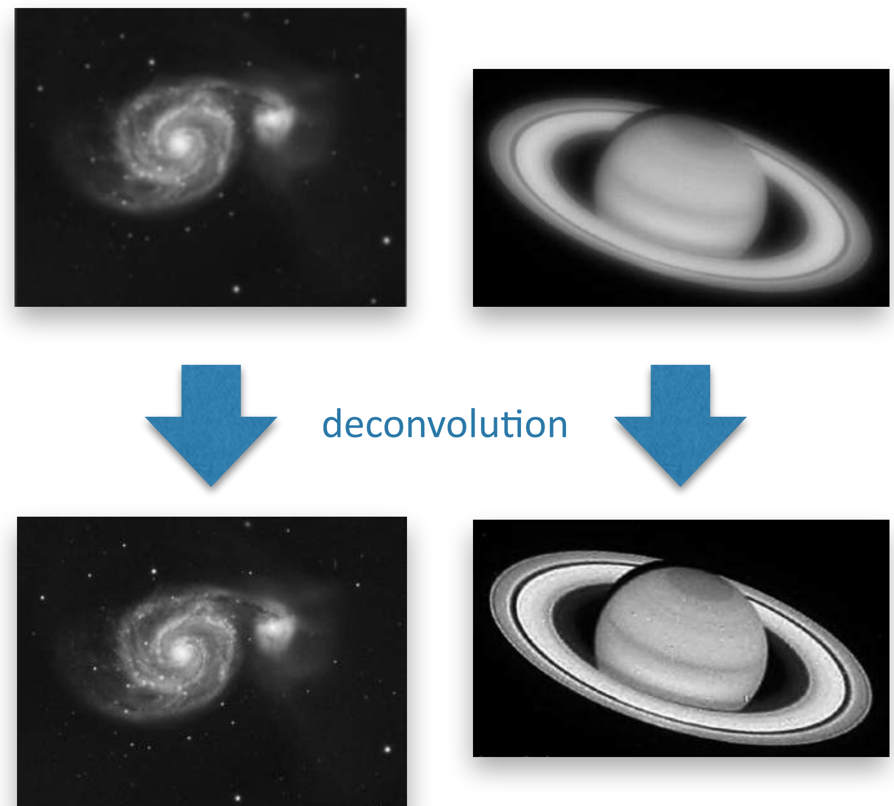
- **Convolution** vs **deconvolution**
  - *Degradation* = **convolution** with $h$
  - *Restoration* = **deconvolution** by $h$ (i.e. remove the effect of $h$)
  - Sometimes called *deblurring*



# The point spread function (PSF) (2/5)



- Problem of blurred images arose with **Hubble Space telescope**
  - Launched in 1990 with a *flawed mirror*
  - The resulting images were **out-of-focus**

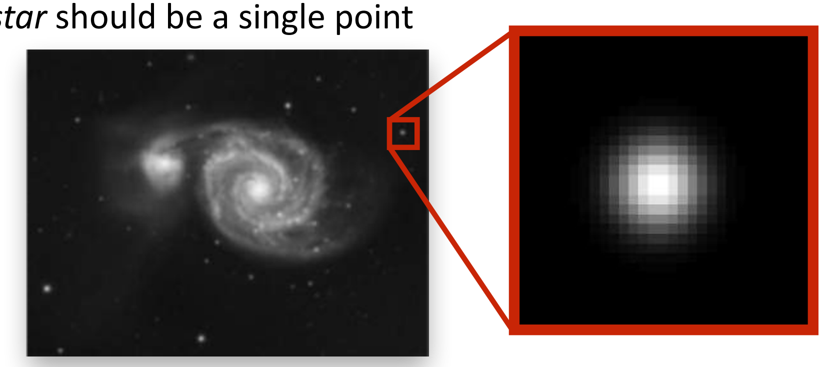


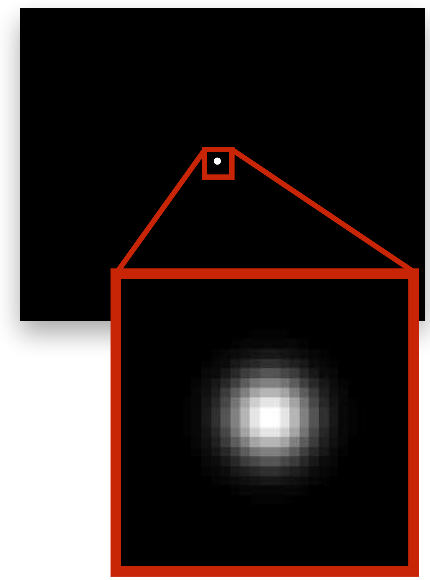

- **Deconvolution** was initially used to correct images before *corrective optics* installed in 1993

  - New lenses/mirror solved the problem

# The point spread function (PSF) (3/5)

- The **main problem** in most situations is to **determine the PSF**
  - Most of the times it is *unknown*
- **Practical approaches** to modeling the PSF
  - Analytical or **empirical estimation**
    - *Example*: for Hubble Space telescope, PSF computed from measurements of mirror curvature
  - Image **acquisition with same optics** of *single point of light*
    - Also an *object with known shape*
  - Analyzing **features of the image**
    - *Example*: an *isolated star* should be a single point

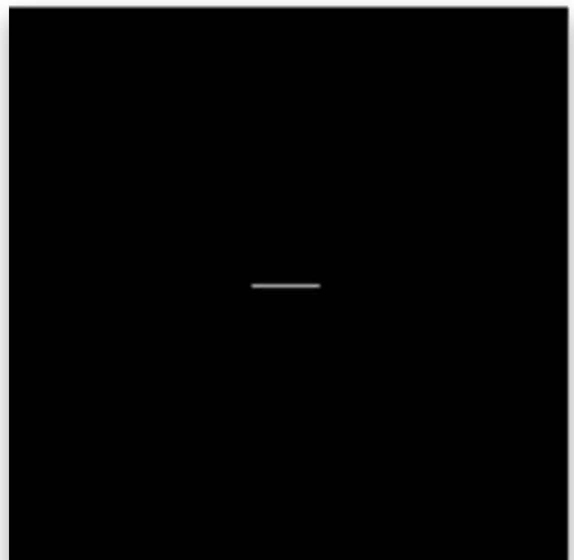




- Note: methods exist also in case when **PSF is not known**
  - e.g. *blind deconvolution* (not seen in this course)

## How does the **PSF of motion blur** looks like?

- For simplicity, assume there is only *horizontal motion*
  e.g. camera pans while image is acquired



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

## Valid also in case of **arbitrary direction of motion**

- **Orientation** matches the one of the motion
- Its **magnitude** (length of “bar”) proportional to amount of motion
  i.e. how many pixels each point was shifted

# Restoration when degradation is only noise

- When the **only degradation is noise**, model can be written as

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

  - **NB**: equivalent to say $H$ = *identity operator*

- Does this ring a bell?

- In this special case, **spatial filtering** is the method of choice!
  - e.g. low-pass filters, median filter etc

# Direct inverse filtering (1/3)

- **Convolution theorem** relates *convolution* and *deconvolution*

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

- Simplest approach is to **directly invert this equation**

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

direct filtering

- Notes
  - If we **consider noise**, i.e. $G(u,v) = F(u,v)H(u,v) + N(\mathrm{u,v})$, then

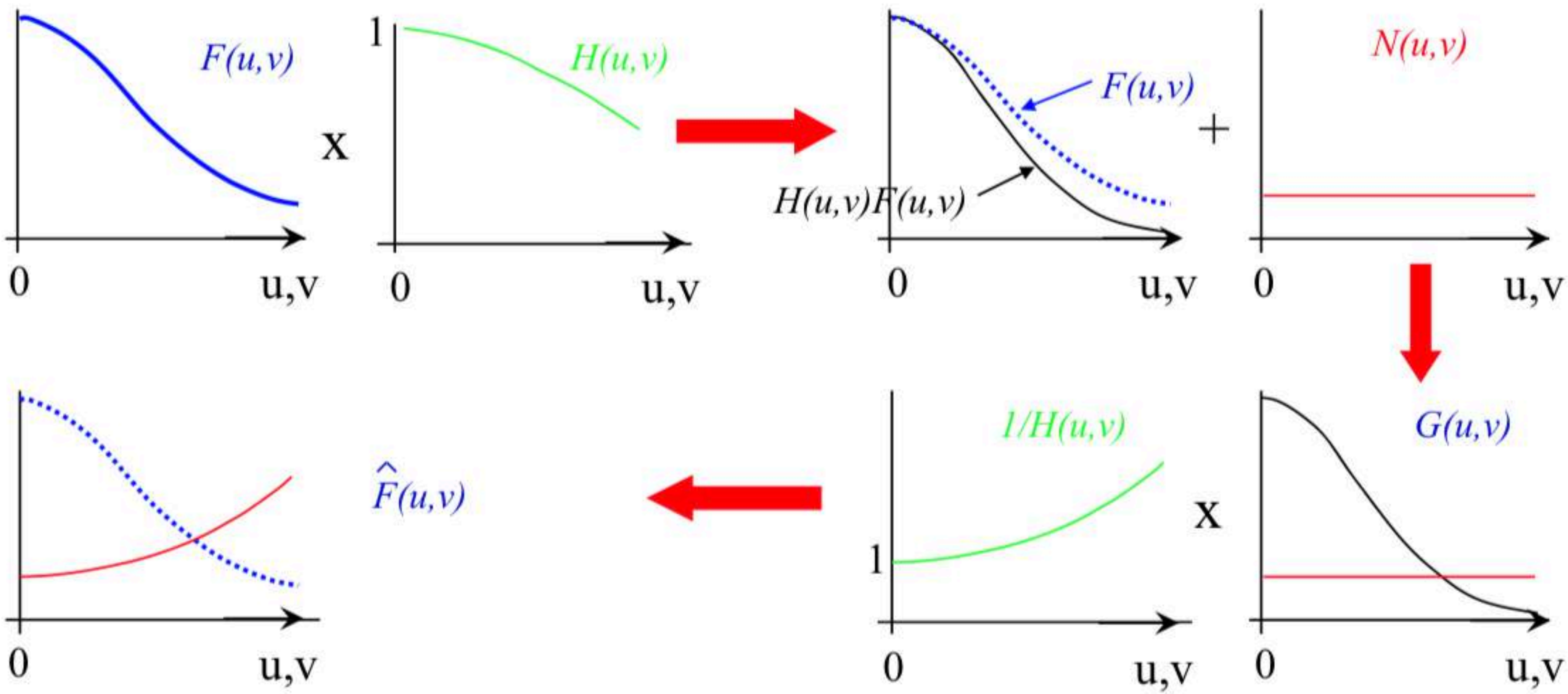
$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

  - Even if we knew $H(u,v)$ exactly, **we could never recover** $F(u,v)$ because *noise* $\eta(x,y)$ *is random* and its FFT $N(u,v)$ is *not known*
  - Practical issues in **division by** $H(u,v)$, as it usually contains many zeros (or small values which would create problems)

# Direct inverse filtering

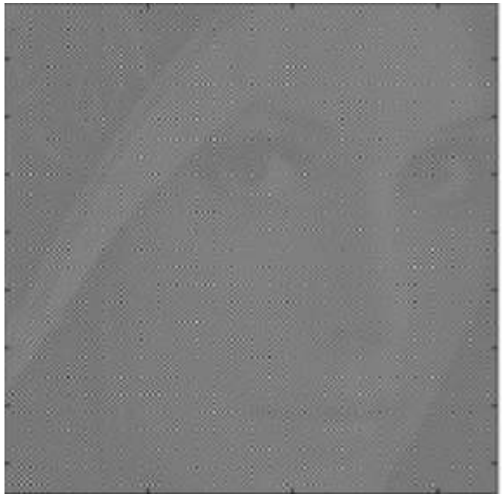
## The problem of **noise amplification**

- Recall that $\hat{F}(u,v) = \dfrac{G(u,v)}{H(u,v)} = F(u,v) + \dfrac{N(u,v)}{H(u,v)}$



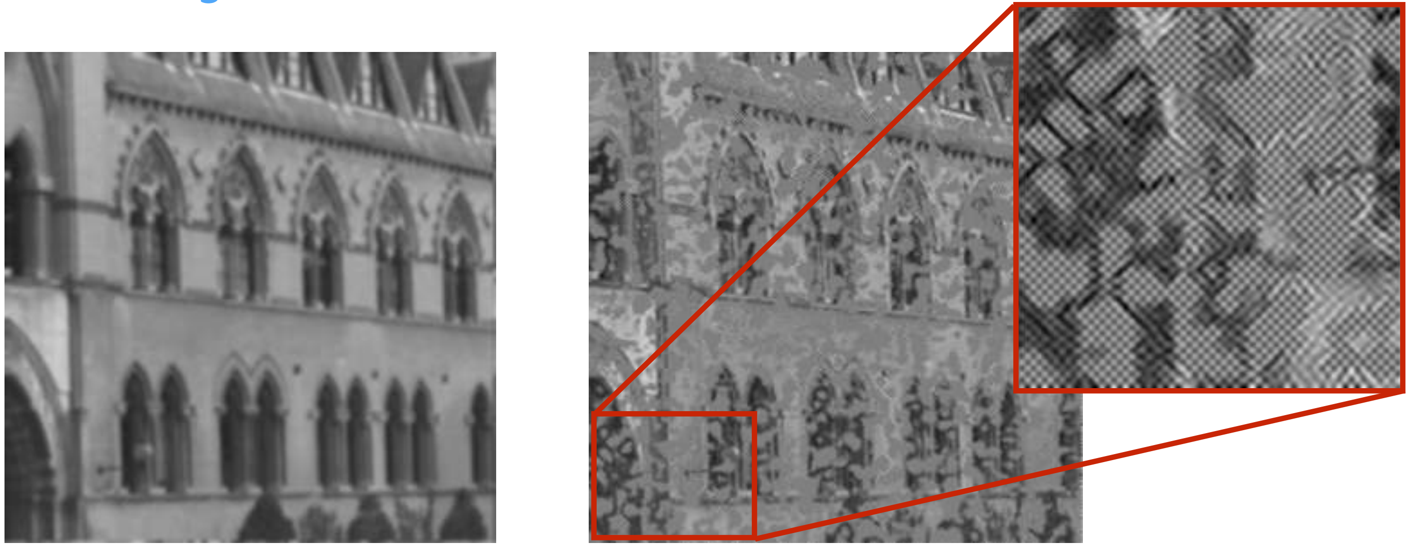

- **NB**: noise at high frequencies tends to be amplified

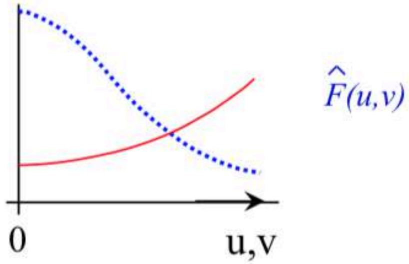
## Examples



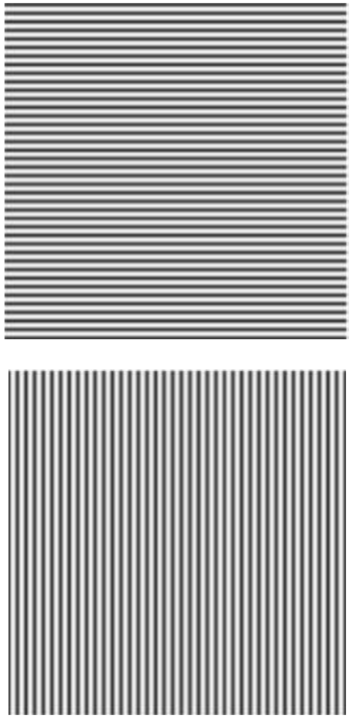
*degraded*

*restored*






Recall **high frequencies** in 2D images



# Linear restoration: Wiener deconvolution (1/3)

- Seeks for an estimated image $\hat{f}(x,y)$ that **minimizes the mean square error** to the *theoretical ideal image* $f(x,y)$

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

  - $\hat{f} \neq f$, but **it is the best approximation** that has minimum differences from *ideal image* $f(x,y)$ in the least-squares sense

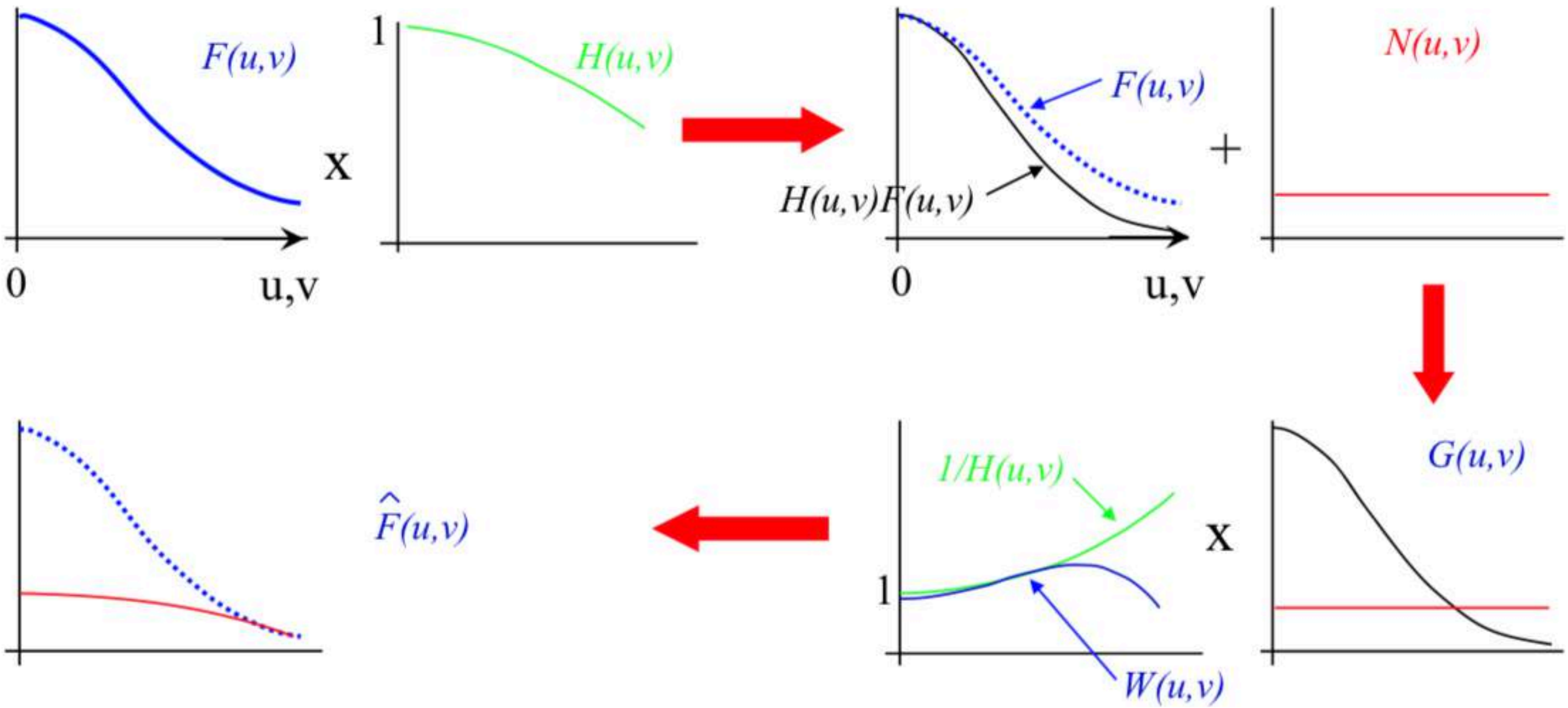
- The **solution** to this expression is (in the *frequency domain*):

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

  - $|H(u,v)|^2 = H^*(u,v)H(u,v)$, with $H^*(u,v)$ complex conjugate of $H(u,v)$
  - Scalar $K$ is added to **limit the effect of noise** and numerical *overflow*
  - The value of $K$ depends on the *amount of noise in the image*
    - This is almost *never known* ➜ $K$ is usually determined *empirically*
  - **NB**: if $K$=0 ➜ *direct inverse filtering*

# Linear restoration: Wiener deconvolution (2/3)

- Basic idea: **selective filtering of frequencies**
  - Recall that $\hat{F}(u,v) = \left[ \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$

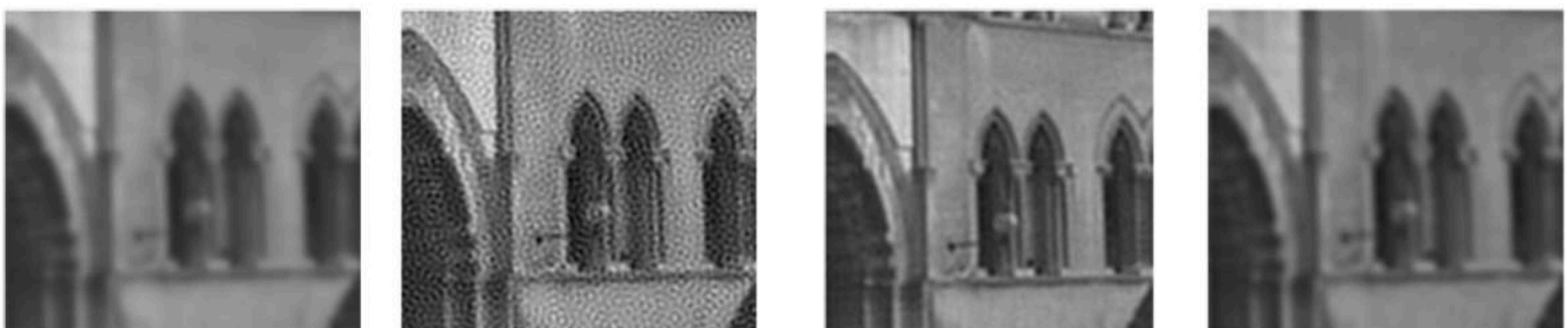



  - **NB**: when $K$ >> $|H(u,v)|$ for large $u, v$ ➜ <u>high frequencies</u> are attenuated

## Empirically

- If $K$ is **too small** ➜ *noise* will dominate
- If $K$ is **too large** ➜ *blur* mostly remains

degraded | $K$ too small | good $K$ | $K$ too big



# Non-linear restoration: LR deconvolution

- **Wiener deconvolution** is an important tool in image processing
  - It's **simple**: once $H(\mathrm{u}, v)$ is specified, $g(x, y)$ obtained via *one filter application*
  - This is very **efficient** and **easy to implement**

- **Lucy and Richardson** proposed a *non-linear iterative algorithm* that often yields **superior results** than linear techniques
  - Arises from a *maximum-likelihood* formulation
  - The solution is **updated iteratively** ($k$ is the iteration number):

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) \left[ h(-x,-y) * \frac{g(x,y)}{h(x,y) * \hat{f}_k(x,y)} \right]$$

- Notes
  - *Non-linear* because division by $\hat{f}_k(x,y)$
  - **Main problem**: when to stop the iterations? Usually, when results are acceptable

# Application: reading license plates

- The plate is out-of-focus because **car is moving**
  - The actual $h(x,y)$ is **unknow**

- **Try different orientations and shifts** until you can read it