Deconvolution

EE367/CS448I: Computational Imaging and Display

stanford.edu/class/ee367

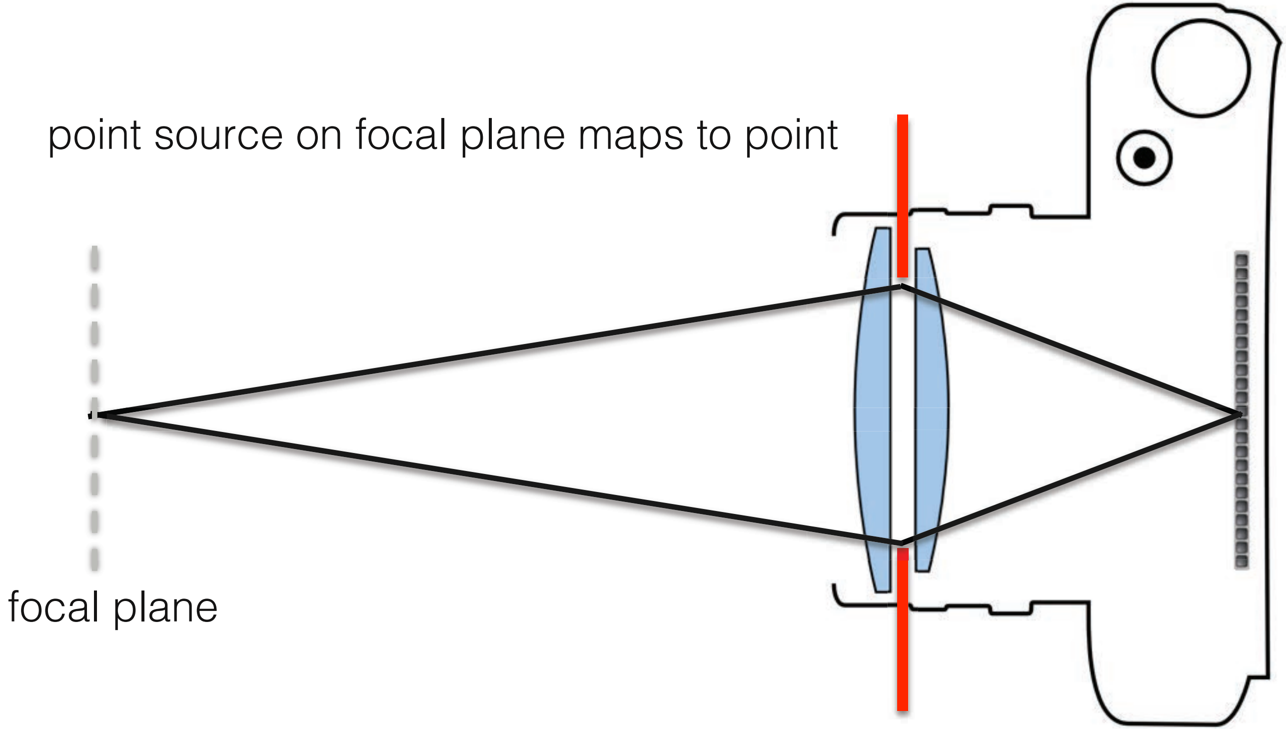
Lecture 6

Gordon Wetzstein

Stanford University

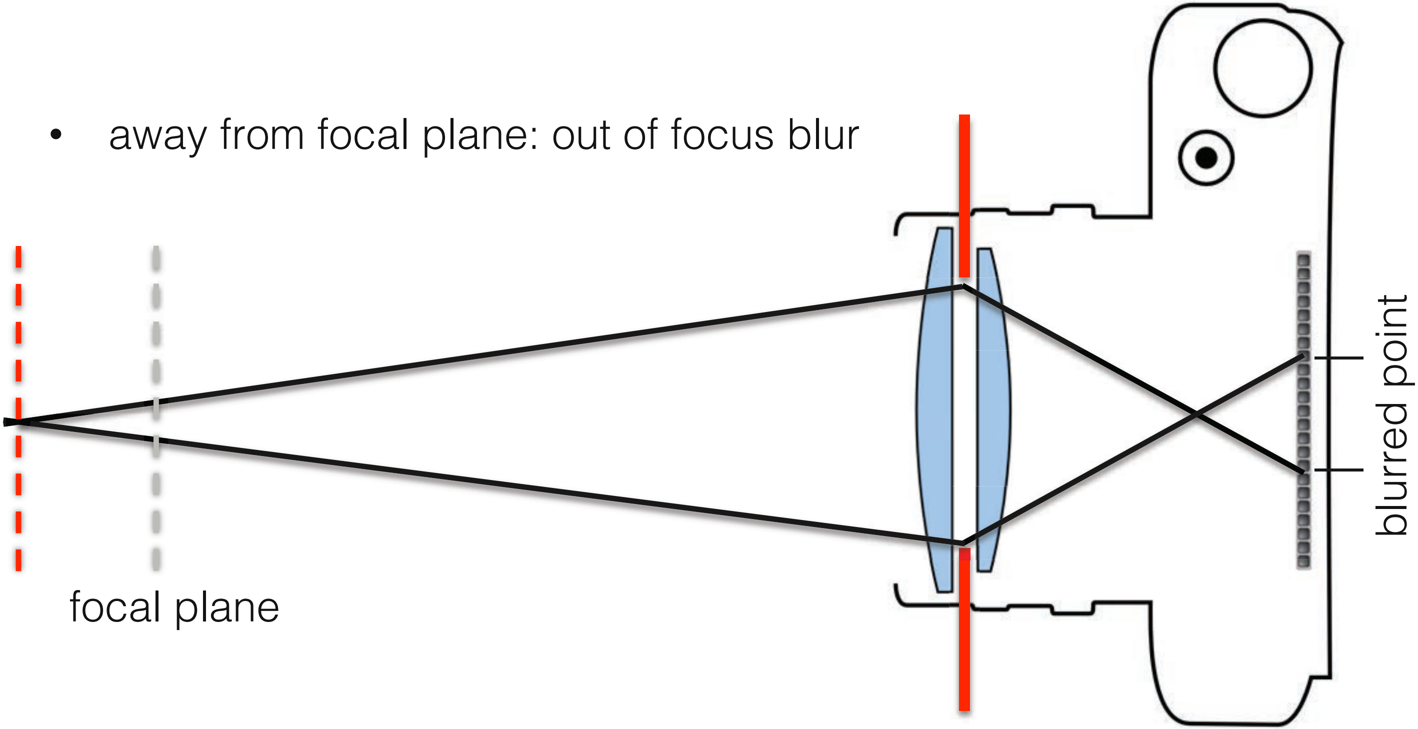
Lens as Optical Low-pass Filter

- point source on focal plane maps to point



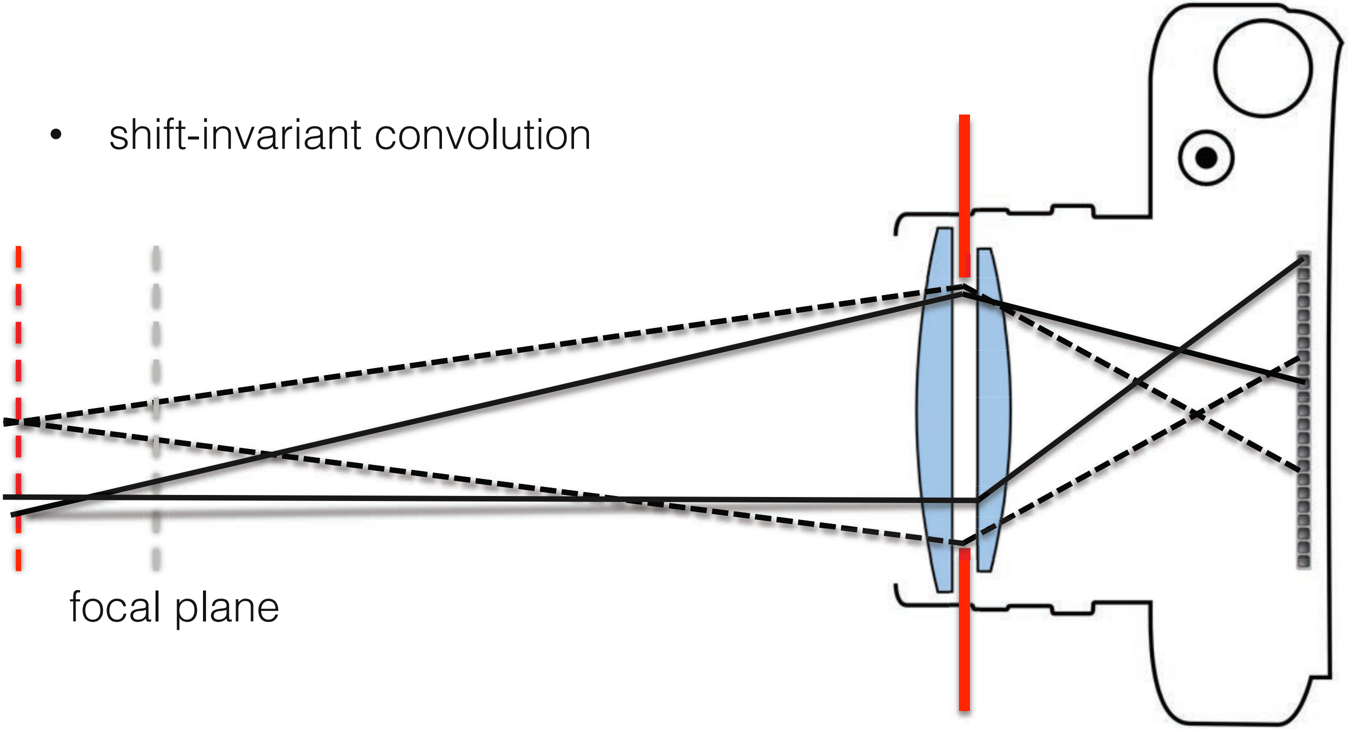
Lens as Optical Low-pass Filter

- away from focal plane: out of focus blur



Lens as Optical Low-pass Filter

- shift-invariant convolution

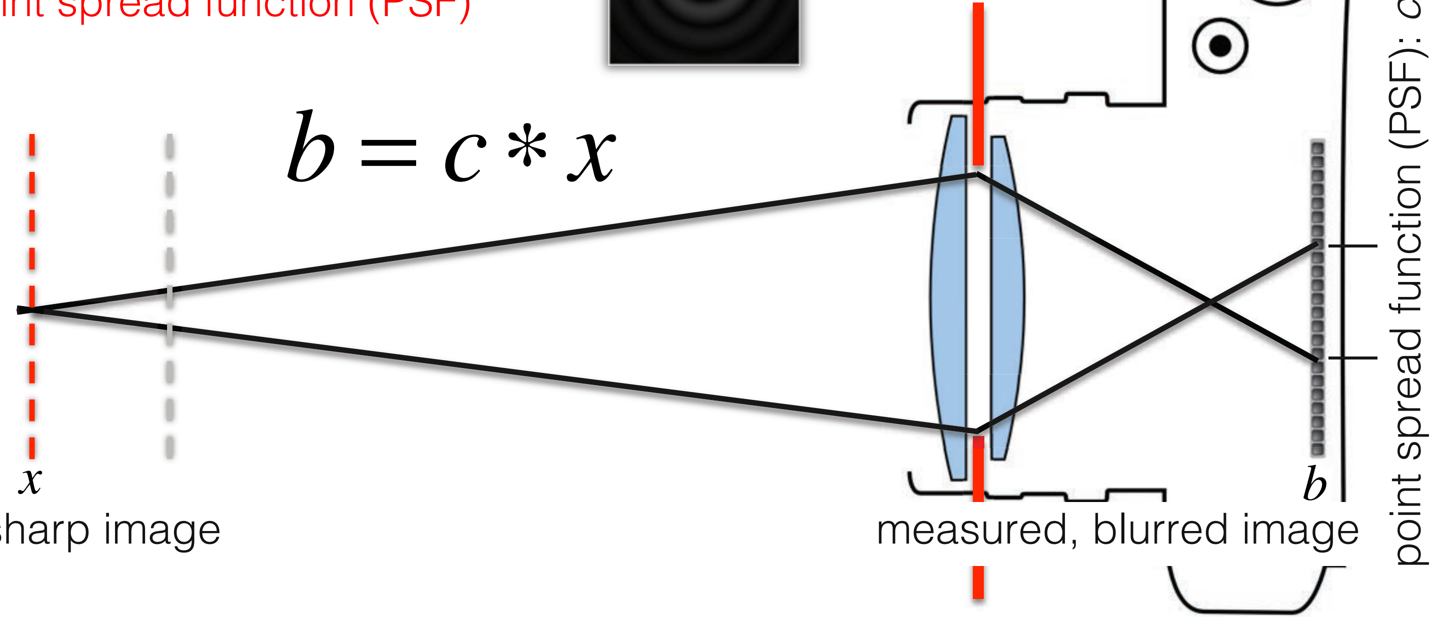


Lens as Optical Low-pass Filter

convolution kernel is called

point spread function (PSF)

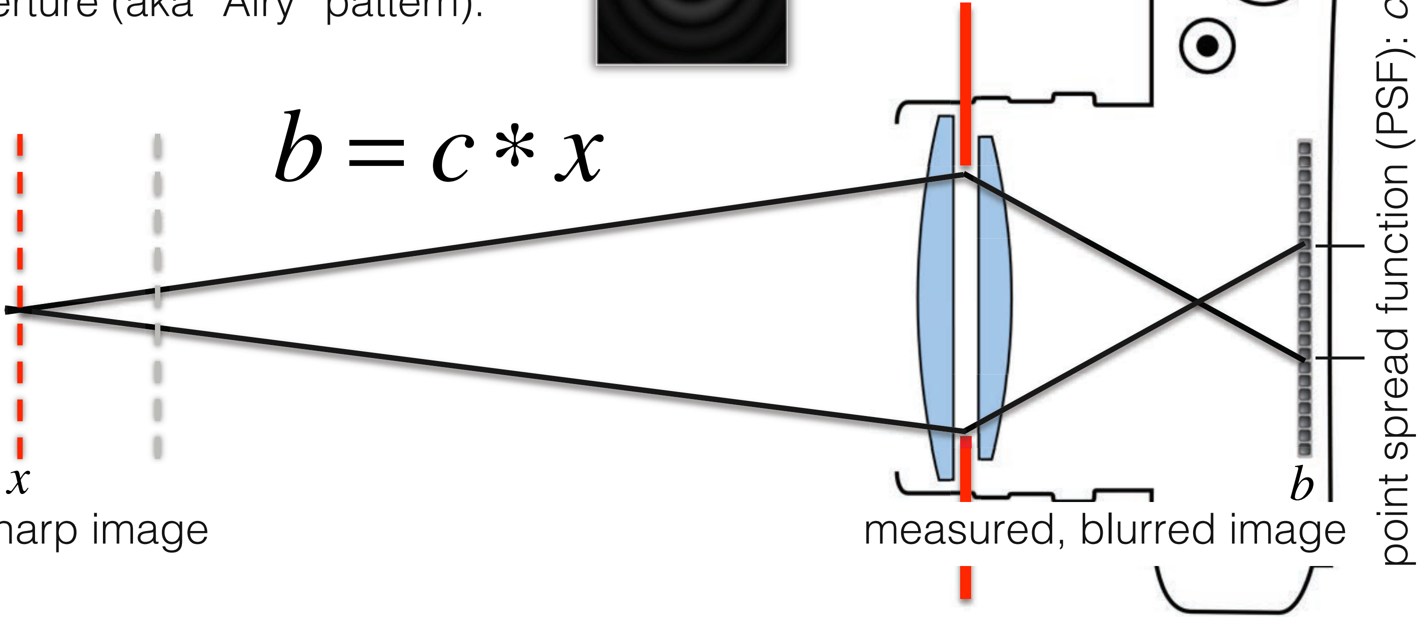
$$b = c * x$$



Lens as Optical Low-pass Filter

diffraction-limited PSF of circular aperture (aka “Airy” pattern):

$$b = c * x$$

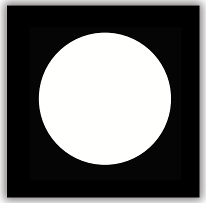


PSF, OTF, MTF

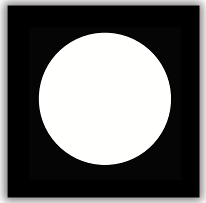
- point spread function (PSF) is fundamental concept in optics
- optical transfer function (OTF) is (complex) Fourier transform of PSF
- modulation transfer function (MTF) is magnitude of OTF

- example:

MTF=|OTF|

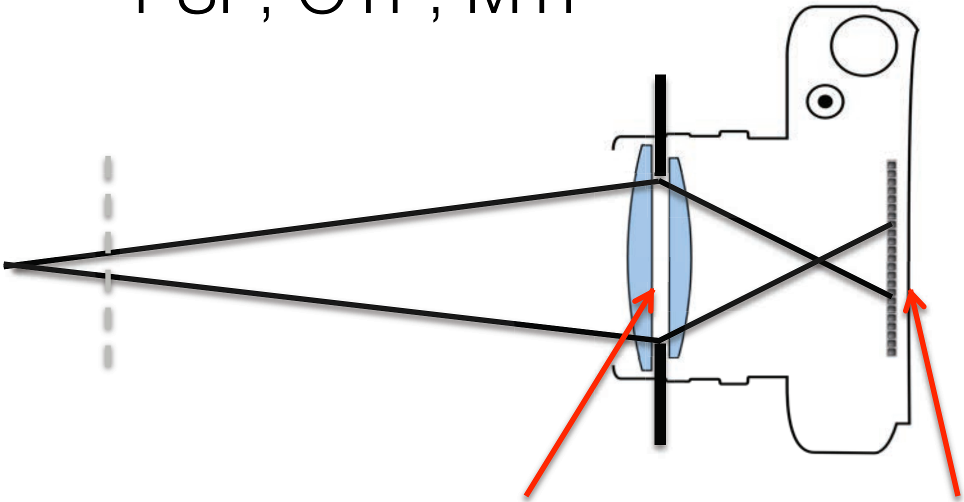


OTF=F{PSF}



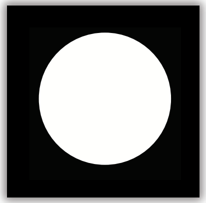
PSF

PSF, OTF, MTF

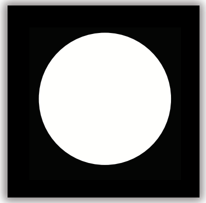


- example:

$\mathrm{MTF}=|\mathrm{OTF}|$



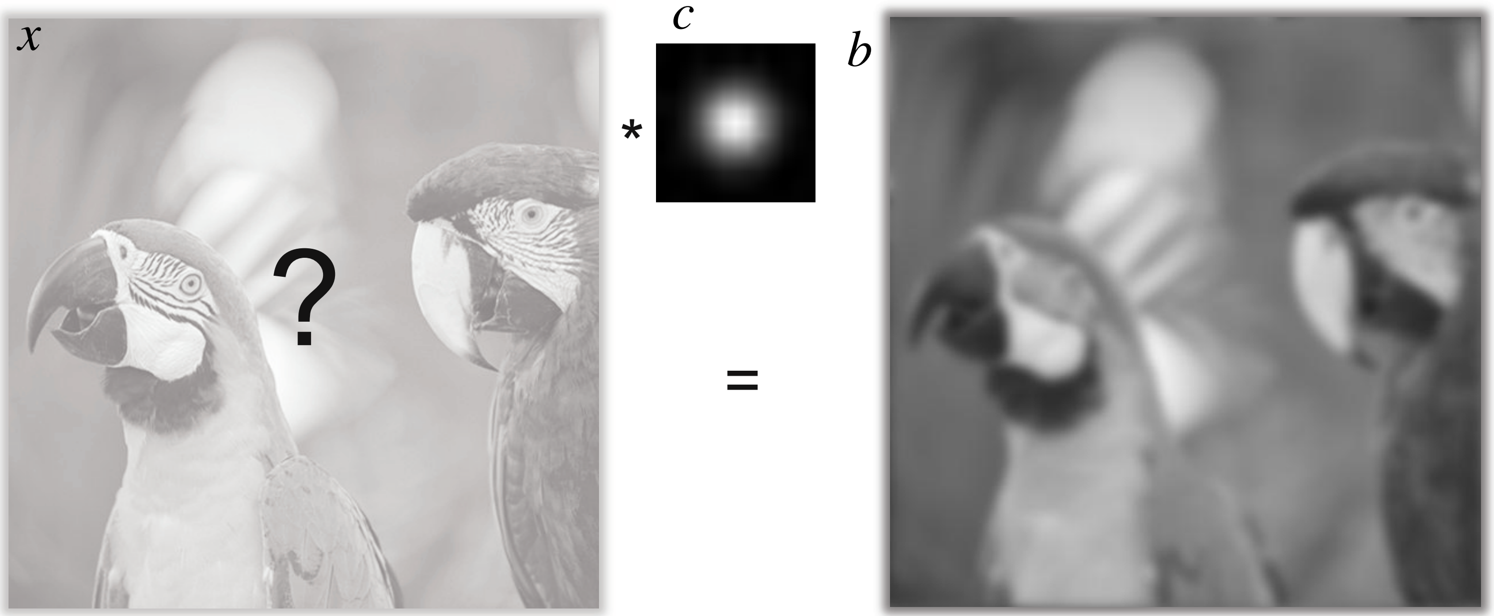
$\mathrm{OTF}=\mathrm{F}\{\mathrm{PSF}\}$



PSF

Deconvolution

- given measurements *b* and convolution kernel *c*, what is *x*?



Deconvolution with Inverse Filtering

- naive solution: apply inverse kernel $\tilde{x} = c^{-1} * b = F^{-1}\left\{\frac{F\{b\}}{F\{c\}}\right\}$

x

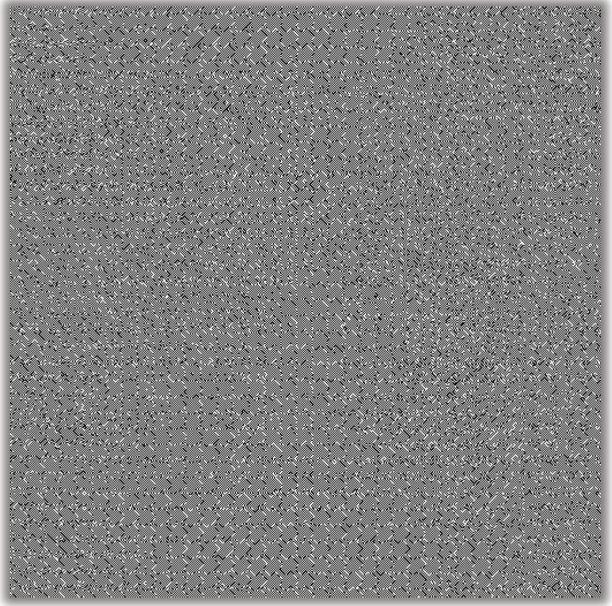
$\tilde{x}$

Deconvolution with Inverse Filtering & Noise

- naive solution: apply inverse kernel $\tilde{x} = c^{-1} * b = F^{-1}\left\{\frac{F\{b\}}{F\{c\}}\right\}$

- Gaussian noise, $\sigma = 0.05$

$\tilde{x}$



Deconvolution with Inverse Filtering & Noise

- results: terrible!

- why? this is an ill-posed problem (division by (close to) zero in frequency domain) → noise is drastically amplified!

- need to include prior(s) on images to make up for lost data
  - for example: noise statistics (signal to noise ratio)

Deconvolution with Wiener Filtering

- apply inverse kernel and don't divide by 0

$$\tilde{x} = F^{-1}\left\{ \boxed{\frac{\left|F\{c\}\right|^2}{\left|F\{c\}\right|^2 + 1/SNR}} \cdot \frac{F\{b\}}{F\{c\}} \right\}$$

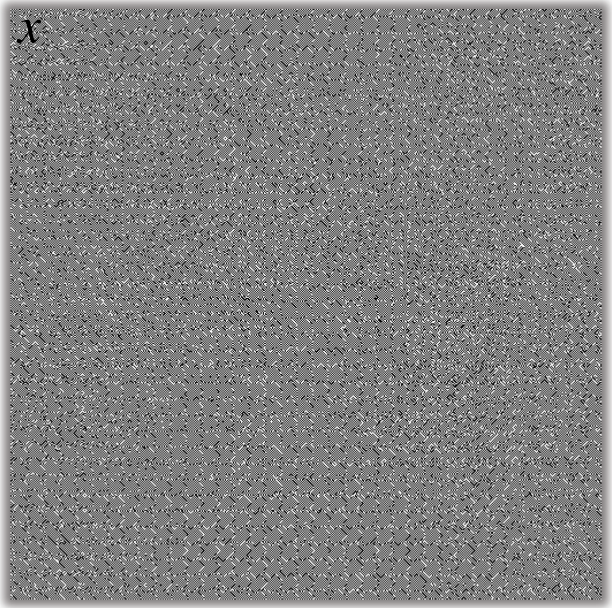
amplitude-dependent

damping factor!

$$SNR = \frac{\text{mean signal} = 0.5}{\text{noise std} = \sigma}$$

Deconvolution with Wiener Filtering

naive



Wiener

$\tilde{x}$

Deconvolution with Wiener Filtering

$\sigma = 0.01$ $\sigma = 0.05$ $\sigma = 0.1$

Deconvolution with Wiener Filtering

- results: not too bad, but noisy

- this is a heuristic → dampen noise amplification

Total Variation

$$\underset{x}{\text{minimize}} \|Cx - b\|_2^2 + \lambda TV(x) = \underset{x}{\text{minimize}} \|Cx - b\|_2^2 + \lambda \|\nabla x\|_1$$

$$\|x\|_1 = \sum_i |x_i|$$

- idea: promote sparse gradients (edges)
- ∇ is finite differences operator, i.e. matrix $\begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & \ddots & \\ & & & -1 \end{bmatrix}$

Rudin et al. 1992

Total Variation

express (forward finite difference) gradient as convolution! →

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

x

$\nabla_x x$

$\nabla_y x$

Total Variation

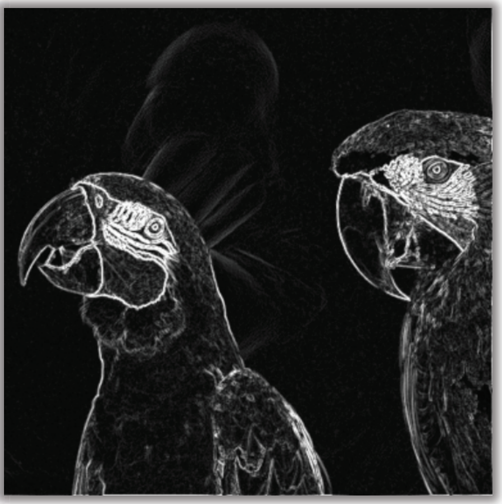
x

better: isotropic

$$\sqrt{(\nabla_x x)^2 + (\nabla_y x)^2}$$

easier: anisotropic

$$\sqrt{(\nabla_x x)^2} + \sqrt{(\nabla_y x)^2}$$



Total Variation

- for simplicity, this lecture only discusses anisotropic TV:

$$TV(x) = \|\nabla_x x\|_1 + \|\nabla_y x\|_1 = \left\| \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix} x \right\|_1$$

- problem: l1-norm is not differentiable, can't use inverse filtering

- however: simple solution for data fitting along and simple solution for TV alone → split problem!

Deconvolution with ADMM

- split deconvolution with TV prior:

$$\begin{aligned} &\text{minimize} \quad \left\|Cx - b\right\|_2^2 + \lambda \left\|z\right\|_1 \\ &\text{subject to} \quad \nabla x = z \end{aligned}$$

- general form of ADMM (alternating direction method of multiplies):

$$\begin{aligned} &\text{minimize} \quad f(x) + g(z) \\ &\text{subject to} \quad Ax + Bz = c \end{aligned}$$

$$f(x) = \left\|Cx - b\right\|_2^2$$

$$g(z) = \lambda \left\|z\right\|_1$$

$$A = \nabla, \; B = -I, c = 0$$

Deconvolution with ADMM

- split deconvolution with TV prior:

$$\begin{array}{ll}\text{minimize} & \boxed{\|Cx-b\|_2^2} + \lambda\|z\|_1 \\ \text{subject to} & \nabla x = z\end{array}$$

- general form of ADMM (alternating direction method of multiplies):

$$\begin{array}{ll}\text{minimize} & \boxed{f(x)} + g(z) \\ \text{subject to} & Ax + Bz = c\end{array}$$

$$\boxed{f(x) = \|Cx-b\|_2^2}$$

$$g(z) = \lambda\|z\|_1$$

$$A = \nabla,\ B = -I,\ c = 0$$

Deconvolution with ADMM

- split deconvolution with TV prior:

$$\begin{array}{ll} \text{minimize} & \|Cx-b\|_2^2 + \boxed{\lambda\|z\|_1} \\ \text{subject to} & \nabla x = z \end{array}$$

- general form of ADMM (alternating direction method of multiplies):

$$\begin{array}{ll} \text{minimize} & f(x) + \boxed{g(z)} \\ \text{subject to} & Ax + Bz = c \end{array}$$

$$f(x) = \|Cx-b\|_2^2$$

$$\boxed{g(z) = \lambda\|z\|_1}$$

$$A = \nabla,\ B = -I,\ c = 0$$

Deconvolution with ADMM

- split deconvolution with TV prior:

$$\begin{aligned} &\text{minimize} \quad \|Cx-b\|_2^2 + \lambda\|z\|_1 \\ &\text{subject to} \quad \nabla x = z \end{aligned}$$

- general form of ADMM (alternating direction method of multiplies):

$$\begin{aligned} &\text{minimize} \quad f(x)+g(z) \\ &\text{subject to} \quad Ax+Bz=c \end{aligned}$$

$$f(x) = \|Cx-b\|_2^2$$

$$g(z) = \lambda\|z\|_1$$

$$\boxed{A=\nabla,\ B=-I,\ c=0}$$

ADMM

$$\text{minimize} \quad f(x) + g(z)$$
$$\text{subject to} \quad Ax + Bz = c$$

- Lagrangian (bring constraints into objective = penalty method):

$$L(x, y, z) = f(x) + g(z) + y^T (Ax + Bz - c)$$

dual variable or Lagrange multiplier

ADMM

$$\text{minimize} \quad f(x) + g(z)$$
$$\text{subject to} \quad Ax + Bz = c$$

- augmented Lagrangian is differentiable under mild conditions (usually better convergence etc.)

$$L_\rho(x,y,z) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)\|Ax + Bz - c\|_2^2$$

ADMM

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- ADMM consists of 3 steps per iteration k:

$$\begin{aligned} x^{k+1} &:= \arg\min_x L_\rho(x, z^k, y^k) \\ z^{k+1} &:= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

ADMM

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax + Bz = c \end{aligned}$$

- ADMM consists of 3 steps per iteration k:

constant

$$\begin{aligned} x^{k+1} &:= \operatorname*{arg\,min}_x \left(f(x) + (\rho/2) \left\| Ax + \boxed{Bz^k - c + u^k} \right\| \right) \\ z^{k+1} &:= \operatorname*{arg\,min}_z \left(g(z) + (\rho/2) \left\| \boxed{Ax^{k+1}} + Bz \boxed{- c + u^k} \right\| \right) \\ u^{k+1} &:= u^k + Ax^{k+1} + Bz^{k+1} - c \end{aligned}$$

scaled dual variable: $u = (1/\rho)y$

ADMM

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- ADMM consists of 3 steps per iteration k:

split f(x) and g(x) into independent problems!

(u connects them)

$$x^{k+1} \quad := \quad \arg\min_x \left(f(x) + (\rho/2) \left\| Ax + Bz^k - c + u^k \right\|_2^2 \right)$$

$$z^{k+1} \quad := \quad \arg\min_z \left(g(z) + (\rho/2) \left\| Ax^{k+1} + Bz - c + u^k \right\|_2^2 \right)$$

$$u^{k+1} \quad := \quad u^k + Ax^{k+1} + Bz^{k+1} - c$$

scaled dual variable: $u = (1/\rho)y$

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx - b\|_2^2 + \lambda \|z\|_1$$

$$\text{subject to} \qquad \nabla x - z = 0$$

- ADMM consists of 3 steps per iteration k:

$$x^{k+1} \quad := \quad \arg\min_x \left(\frac{1}{2}\|Cx - b\|_2^2 + (\rho / 2)\left\|\nabla x - z^k + u^k\right\|_2^2 \right)$$

$$z^{k+1} \quad := \quad \arg\min_z \left(\lambda \|z\|_1 + (\rho / 2)\left\|\nabla x^{k+1} - z + u^k\right\|_2^2 \right)$$

$$u^{k+1} \quad := \quad u^k + \nabla x^{k+1} - z^{k+1}$$

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2 + \lambda\|z\|_1$$

$$\text{subject to} \qquad \nabla x - z = 0$$

constant, say $v = z^k - u^k$

1. x-update: $$x^{k+1} := \arg\min_x \left(\frac{1}{2}\|Cx-b\|_2^2 + (\rho/2)\left\|\nabla x - \boxed{z^k + u^k}\right\|_2^2 \right)$$

solve normal equations $$\left(C^T C + \rho\nabla^T\nabla\right)x = \left(C^T b + \rho\nabla^T v\right)$$

$$\nabla^T v = \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix}^T v = \nabla_x^T v_1 + \nabla_y^T v_2$$

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2+\lambda\|z\|_1$$

$$\text{subject to} \qquad \nabla x - z = 0$$

constant, say $v = z^k - u^k$

1. x-update: $$x^{k+1} := \arg\min_x \left(\frac{1}{2}\|Cx-b\|_2^2 + (\rho/2)\left\|\nabla x - \boxed{z^k + u^k}\right\|_2^2 \right)$$

$$x = \left(C^T C + \rho \nabla^T \nabla\right)^{-1}\left(C^T b + \rho \nabla^T v\right)$$

- inverse filtering: $$x^{k+1} = F^{-1}\left\{ \frac{\boxed{F\{c\}^* \cdot F\{b\}} + \rho\left(\boxed{F\{\nabla_x\}^*} \cdot F\{v_1\} + \boxed{F\{\nabla_y\}^*} \cdot F\{v_2\}\right)}{\boxed{F\{c\}^* \cdot F\{c\} + \rho\left(F\{\nabla_x\}^* \cdot F\{\nabla_x\} + F\{\nabla_y\}^* \cdot F\{\nabla_y\}\right)}} \right\}$$

precompute!

→ may blow up, but that's okay

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2+\lambda\|z\|_1$$

$$\text{subject to} \quad \nabla x - z = 0$$

constant, say $a=\nabla x^{k+1}+u^k$

2. z-update: $$z^{k+1} := \arg\min_z \left(\lambda\|z\|_1+(\rho/2)\left\|\boxed{\nabla x^{k+1}}-z+\boxed{u^k}\right\|_2^2\right)$$

- l1-norm is not differentiable! yet, closed-form solution via **element-wise soft thresholding**:

$$z^{k+1} := S_{\lambda/\rho}(a) \qquad S_\kappa(a)=\begin{cases} a-\kappa & a>\kappa \\ 0 & |a|\le\kappa \\ a+\kappa & a<-\kappa \end{cases} = (a-\kappa)_+ - (-a-\kappa)_+$$

$$\kappa=\lambda/\rho$$

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2+\lambda\|z\|_1$$

$$\text{subject to} \qquad \nabla x - z = 0$$

for k=1:max_iters

$$x^{k+1} \quad := \quad \arg\min_x \left(\frac{1}{2} \left\| \begin{bmatrix} C \\ \rho\nabla \end{bmatrix} x - \begin{bmatrix} b \\ \rho v \end{bmatrix} \right\|_2^2 \right)$$ inverse filtering

$$z^{k+1} \quad := \quad S_{\lambda/\rho}(\nabla x^{k+1} + u^k)$$ element-wise threshold

$$u^{k+1} \quad := \quad u^k + \nabla x^{k+1} - z^{k+1}$$ trivial

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2 + \lambda\|z\|_1$$

$$\text{subject to} \qquad \nabla x - z = 0$$

for k=1:max_iters

$$x^{k+1} \quad := \quad \arg\min_x \left(\frac{1}{2} \left\| \begin{bmatrix} C \\ \rho\nabla \end{bmatrix} x - \begin{bmatrix} b \\ \rho v \end{bmatrix} \right\|_2^2 \right)$$ inverse filtering

$$z^{k+1} \quad := \quad S_{\lambda/\rho}(\nabla x^{k+1} + u^k)$$ element-wise threshold

$$u^{k+1} \quad := \quad u^k + \nabla x^{k+1} - z^{k+1}$$ trivial

→ easy! ☺

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx - b\|_2^2 + \lambda \|z\|_1$$

$$\text{subject to} \quad \nabla x - z = 0$$

Wiener filtering

ADMM with anisotropic TV, $\lambda = 0.01, \rho = 10$

Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2 + \lambda\|z\|_1$$

$$\text{subject to} \quad \nabla x - z = 0$$

- too much TV: “patchy”, too little TV: noisy

$\lambda = 0.01, \rho = 10$

$\lambda = 0.05, \rho = 10$

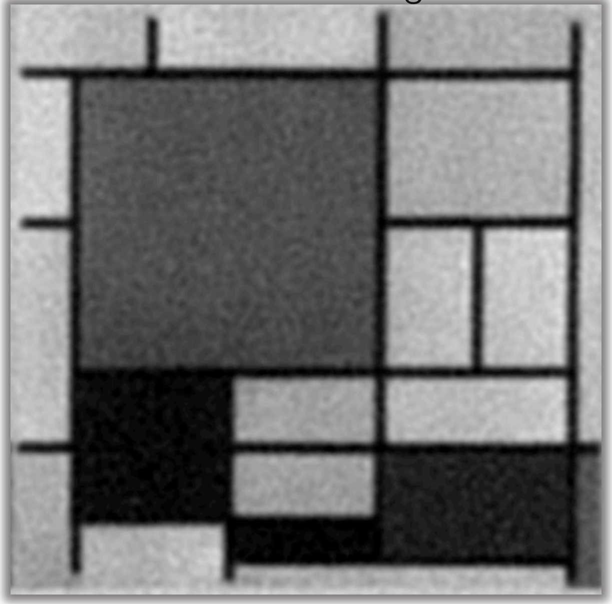
$\lambda = 0.1, \rho = 10$

Deconvolution with ADMM

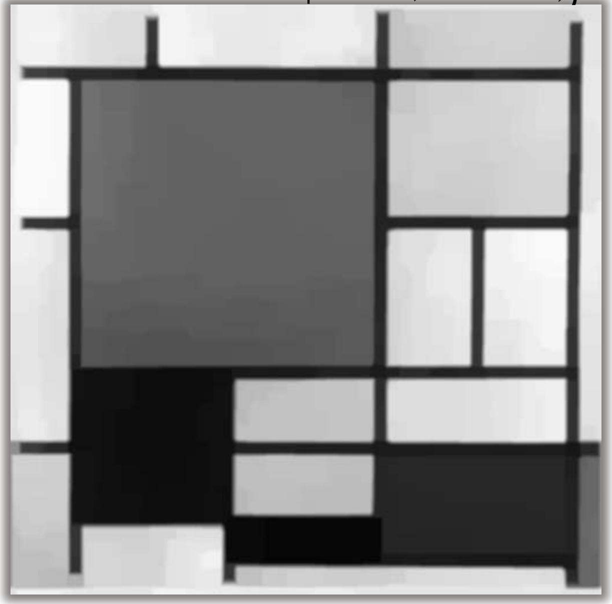
$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2 + \lambda\|z\|_1$$

$$\text{subject to} \quad \nabla x - z = 0$$

Wiener filtering



ADMM with anisotropic TV, $\lambda = 0.1, \rho = 10$



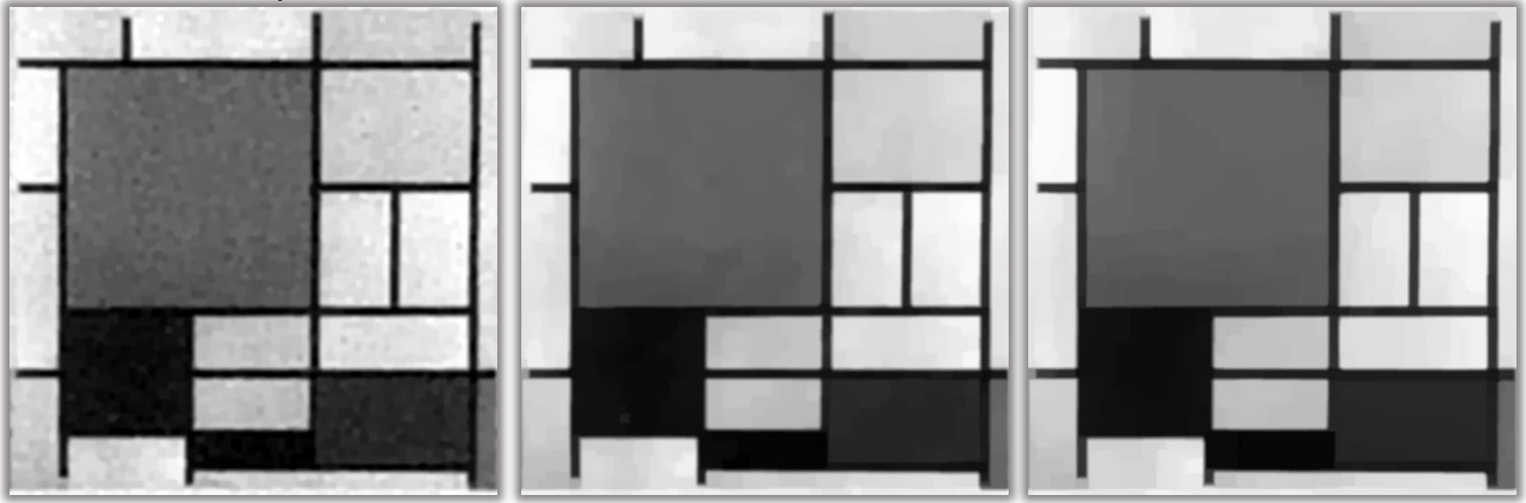
Deconvolution with ADMM

$$\text{minimize} \quad \frac{1}{2}\|Cx-b\|_2^2 + \lambda\|z\|_1$$

$$\text{subject to} \quad \nabla x - z = 0$$

- too much TV: okay because image actually has sparse gradients!

$\lambda = 0.01, \rho = 10$ $\lambda = 0.05, \rho = 10$ $\lambda = 0.1, \rho = 10$



Outlook ADMM

- powerful tool for many computational imaging problems
- include generic prior in g(z), just need to derive proximal operator

$$\underset{x}{\text{minimize}} \underbrace{\frac{1}{2}\|Ax - b\|_2^2}_{\text{data fidelity}} + \underbrace{\Gamma(x)}_{\text{regularization}} \quad \longrightarrow \quad \begin{array}{ll} \underset{\{x,z\}}{\text{minimize}} & f(x) + g(z) \\ \text{subject to} & Ax = z \end{array}$$

- example priors: noise statistics, sparse gradient, smoothness, …
- weighted sum of different priors also possible
- anisotropic TV is one of the easiest priors

Remember!

- implement matrix-free operations for Ax and A'x if efficient (e.g. multiplications and divisions in frequency space)

- split difficult problems (e.g., inverse problems with non-differentiable priors) into easier subproblems - ADMM

Homework 3

- implement:
  - filtering
  - inverse filtering and Wiener filtering
  - deconvolution with ADMM + (anisotropic) TV prior

Notes for Homework 3

$$I \in \Re^{M \times N}, X \in \Re^{MN \times 1}$$

$$U \in \Re^{2MN \times 1}, Z \in \Re^{2MN \times 1}$$

- notes for ADMM implementation:
  - initialize U, Z, X with 0
  - implement with matrix-free form: all FT multiplications / divisions
  - in 2D, finite differences matrix becomes (anisotropic form), use matrix free-operations as well!

$$\nabla = \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix}$$

  - see note notes in HW
  - check ADMM example scripts: http://web.stanford.edu/~boyd/papers/admm/

Notes for Homework 3

- signal-to-noise ratio (<u>SNR</u>): $SNR = \frac{P_{signal}}{P_{noise}}$ $\quad SNR_{dB} = 10 \cdot \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$

- peak signal-to-noise ratio (<u>PSNR</u>): $MSE = \frac{1}{mn}\sum_m \sum_n \left(x_{target} - x_{est}\right)^2$

  (always in dB)

  $$PSNR = 10 \cdot \log_{10}\left(\frac{\max(x_{target})^2}{MSE}\right) = 10 \cdot \log_{10}\left(\frac{1}{MSE}\right)$$

- <u>residual</u> is value of objective function:

  not regularized: $\frac{1}{2}\left\|Cx - b\right\|_2^2$ $\quad$ regularized: $\frac{1}{2}\left\|Cx - b\right\|_2^2 + \lambda \left\| \begin{bmatrix} \nabla_x \\ \nabla_y \end{bmatrix} x \right\|_1$

- <u>convergence</u>: residual for increasing iterations (should always decrease!)

References and Further Reading

- Boyd, Parikh, Chu, Peleato, Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”, Foundations and Trends in Machine Learning, 2011

- A. Chambolle, T. Pock “A first-order primal-dual algorithm for convex problems with applications in imaging”, Journal of Mathematical Imaging and Vision, 2011
- Boreman, “Modulation Transfer Function in Optical and ElectroOptical Systems”, SPIE Publications, 2001
- Rudin, Osher, Fatemi, “Nonlinear total variation based noise removal algorithms”, Physica D: Nonlinear Phenomena 60, 1
- http://www.imagemagick.org/Usage/fourier/