Deconvolution
Course announcements

• Homework assignment 4 due November 2\textsuperscript{nd}.
  - Generally shorter to accommodate final project proposals.
  - Two bonus parts.

• Project logistics on Piazza and the course website.
  - Project ideas due on Piazza on October 23\textsuperscript{rd} (optional).
  - Project proposals due on Gradescope on October 30\textsuperscript{th}.

• Office hour logistics for this week:
  - Yannis will have extra office hours on Friday (time TBD).

• Late submissions:
  - We are making an exception for homework assignment 3 and we won’t count late
days for submissions that are a few minutes late due to uploading delays.
  - We will resume enforcing late days strictly for subsequent homeworks. One second
late is one late day.
Computational photography talks this week @ CMU

• Ce Liu, Google Research (October 20th, 11 am – noon).
  - Title: Advancing the State of the Art of Computer Vision for Billions of Users
  At Google, advancing the state of the art of computer vision is very impactful as there are billions of users of Google products, many of which require high-quality, artifact-free images. I will share what we learned from successfully launching core computer vision techniques for various Google products, including PhotoScan (Photos), seamless Google Street View panorama stitching (Geo), Super Res Zoom (Pixel 4), Auto Pop-out & Uncrop (Display Ads), and Rendering4AI (Cloud AI). We also conduct academic research and publish at top-tier conferences. I will give an overview of several representative works, including seeing through obstructions (Siggraph’15), learning the depth of moving people by watching frozen people (CVPR’19), GAN-based image uncrop (ICCV’19), and supervised contrastive learning (NeurIPS’20).

• Tali Dekel, Google Research (October 20th, noon – 1 pm).
  - Title: Learning to Retime People in Videos
  By changing the speed of frames, or the speed of objects, we can enhance the way we perceive events or actions in videos. In this talk, I will present two of my recent works on retiming videos, and more specifically, manipulating the timings of people's actions. 1) “SpeedNet” (CVPR 2020 oral): a method for adaptively speeding up videos based on their content, allowing us to gracefully watch videos faster while avoiding jerky and unnatural motions. 2) “Layered Neural Rendering for Retiming People” (SIGGRAPH Asia): a method for speeding up, slowing down, or entirely freezing certain people in videos, while automatically re-rendering properly all the scene elements that are related to those people, like shadows, reflections, and loose clothing. Both methods are based on novel deep neural networks that learn concepts of natural motion and scene decomposition just by observing ordinary videos, without requiring any manual labels. I'll show adaptively sped-up videos of sports, of boring family events (that all of us want to watch faster), and I'll demonstrate various retiming effects of people dancing, groups running, and kids jumping on trampolines.
Overview of today’s lecture

• Sources of blur.
• Deconvolution.
• Blind deconvolution.
Most of these slides were adapted from:

- Fredo Durand (MIT).
- Gordon Wetzstein (Stanford).
Why are our images blurry?
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.
Lens imperfections

- Ideal lens: An point maps to a point at a certain plane.
Lens imperfections

• Ideal lens: A point maps to a point at a certain plane.
• Real lens: A point maps to a circle that has non-zero minimum radius among all planes.

What is the effect of this on the images we capture?
Lens imperfections

- Ideal lens: An point maps to a point at a certain plane.
- Real lens: A point maps to a circle that has non-zero minimum radius among all planes.

Shift-invariant blur.
Lens imperfections

What causes lens imperfections?
Lens imperfections

What causes lens imperfections?

• Aberrations.

(Important note: Oblique aberrations like coma and distortion are not shift-invariant blur and we do not consider them here!)

• Diffraction.
Lens as an optical low-pass filter

Point spread function (PSF): The blur kernel of a lens.

- “Diffraction-limited” PSF: No aberrations, only diffraction. Determined by aperture shape.

![Diagram of lens as an optical low-pass filter](image)

**Object distance D**

**Focus distance D’**

**Blur kernel**

**Diffraction-limited PSF of a circular aperture (Airy disk)**
Lens as an optical low-pass filter

Point spread function (PSF): The blur kernel of a lens.
- “Diffraction-limited” PSF: No aberrations, only diffraction. Determined by aperture shape.

Optical transfer function (OTF): The Fourier transform of the PSF. Equal to aperture shape.
Lens as an optical low-pass filter

image from a perfect lens  \*  imperfect lens PSF  =  image from imperfect lens

X  \*  C  =  b
Lens as an optical low-pass filter

If we know \( c \) and \( b \), can we recover \( x \)?

\[
\begin{align*}
\text{image from a perfect lens} & \quad \times \quad \text{imperfect lens PSF} & \quad = \quad \text{image from imperfect lens} \\
X & \quad \ast \quad C & \quad = \quad b
\end{align*}
\]
Quick aside: optical anti-aliasing

Lenses act as (optical) low-pass filters.

- **Lenslets also filter the image to avoid resolution artifacts.**
- Lenslets are problematic when working with coherent light.
- Many modern cameras do not have lenslet arrays.

We will discuss these issues in more detail at a later lecture.
Quick aside: optical anti-aliasing

Lenses act as (optical) smoothing filters.

- Sensors often have a lenslet array in front of them as an anti-aliasing (AA) filter.
- However, the AA filter means you also lose resolution.
- Nowadays, due the large number of sensor pixels, AA filters are becoming unnecessary.

Photographers often hack their cameras to remove the AA filter, in order to avoid the loss of resolution.

a.k.a. “hot rodding”
Quick aside: optical anti-aliasing

Example where AA filter is needed

without AA filter

with AA filter
Quick aside: optical anti-aliasing

Example where AA filter is unnecessary

without AA filter

with AA filter
Lens as an optical low-pass filter

If we know $c$ and $b$, can we recover $x$?

image from a perfect lens  

imperfect lens PSF

image from imperfect lens

$X \ast c = b$
Deconvolution

\[ x \ast c = b \]

If we know \( c \) and \( b \), can we recover \( x \)?
Deconvolution

\[ x * c = b \]

Reminder: convolution is multiplication in Fourier domain:

\[ F(x) \cdot F(c) = F(b) \]

If we know c and b, can we recover x?
Deconvolution

\[ x \ast c = b \]

Reminder: convolution is multiplication in Fourier domain:

\[ F(x) \cdot F(c) = F(b) \]

Deconvolution is division in Fourier domain:

\[ F(x_{est}) = \frac{F(b)}{F(c)} \]

After division, just do inverse Fourier transform:

\[ x_{est} = F^{-1} \left( \frac{F(b)}{F(c)} \right) \]
Deconvolution

Any problems with this approach?
Deconvolution

- The OTF (Fourier of PSF) is a low-pass filter
- The measured signal includes noise

\[ b = c \ast x + n \]
Deconvolution

- The OTF (Fourier of PSF) is a low-pass filter
- The measured signal includes noise
- When we divide by zero, we amplify the high frequency noise
Naïve deconvolution

Even tiny noise can make the results awful.

- Example for Gaussian of $\sigma = 0.05$

\[
b \ast c^{-1} = x_{\text{est}}
\]
Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

\[ x_{\text{est}} = F^{-1} \left( \frac{|F(c)|^2}{|F(c)|^2 + 1/\text{SNR}(\omega)} \cdot \frac{F(b)}{F(c)} \right) \]

- Derived as solution to maximum-likelihood problem under Gaussian noise assumption
- Requires noise of signal-to-noise ratio at each frequency

\[ \text{SNR}(\omega) = \frac{\text{signal variance at } \omega}{\text{noise variance at } \omega} \]
Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

\[
X_{\text{est}} = F^{-1}\left( \frac{|F(c)|^2}{|F(c)|^2 + 1/\text{SNR}(\omega)} \cdot \frac{F(b)}{F(c)} \right)
\]

noise-dependent damping factor

Intuitively:
- When SNR is high (low or no noise), just divide by kernel.
- When SNR is low (high noise), just set to zero.
Deconvolution comparisons

naïve deconvolution  Wiener deconvolution
Deconvolution comparisons

\( \sigma = 0.01 \)  \hspace{1cm} \sigma = 0.05  \hspace{1cm} \sigma = 0.01
Derivation

Sensing model:

\[ \tilde{x} = c \times x + n \]

Noise \( n \) is assumed to be zero-mean and independent of signal \( x \).
Derivation

Sensing model:

\[ b = c \ast x + n \]

Noise \( n \) is assumed to be zero-mean and independent of signal \( x \).

Fourier transform:

\[ B = C \cdot X + N \]

Why multiplication?
Derivation

Sensing model:

\[ b = c \ast x + n \]

Noise \( n \) is assumed to be zero-mean and independent of signal \( x \).

Fourier transform:

\[ B = C \cdot X + N \]

Convolution becomes multiplication.

Problem statement: Find function \( H(\omega) \) that minimizes expected error in Fourier domain.

\[ \min_H E[\|X - HB\|^2] \]
Derivation

Replace B and re-arrange loss:

\[
\min_H E \left[ \| (1 + HC)X - HN \|^2 \right]
\]

Expand the squares:

\[
\min_H \| 1 - HC \|^2 E [\|X\|^2] - 2(1 - HC)E [XN] + \| H \|^2 E [\|N\|^2]
\]
Derivation

When handling the cross terms:
• Can I write the following?

\[ E[XN] = E[X]E[N] \]
Derivation

When handling the cross terms:

• Can I write the following?

\[ E[XN] = E[X]E[N] \]

Yes, because X and N are assumed independent.

• What is this expectation product equal to?
Derivation

When handling the cross terms:

• Can I write the following?

\[ E[XY] = E[X]E[Y] \]

Yes, because \(X\) and \(N\) are assumed independent.

• What is this expectation product equal to?

Zero, because \(N\) has zero mean.
Derivation

Replace B and re-arrange loss:

$$\min_H E[\| (1 + HC)X - HN \|^2]$$

Expand the squares:

$$\min_H \|1 - HC\|^2 E[\|X\|^2] - 2(1 - HC)E[YN] + \|H\|^2 E[\|N\|^2]$$

Simplify:

$$\min_H \|1 - HC\|^2 E[\|X\|^2] + \|H\|^2 E[\|N\|^2]$$

How do we solve this optimization problem?
Derivation

Differentiate loss with respect to $H$, set to zero, and solve for $H$:

$$\frac{\partial \text{loss}}{\partial H} = 0$$

$$\Rightarrow -2(1 - HC)E[||X||^2] + 2HE[||N||^2] = 0$$

$$\Rightarrow H = \frac{CE[||X||^2]}{C^2E[||X||^2] + E[||N||^2]}$$

Divide both numerator and denominator with $E[||X||^2]$, extract factor $1/C$, and done!
Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

\[ X_{\text{est}} = F^{-1} \left( \frac{|F(c)|^2}{|F(c)|^2 + 1/\text{SNR}(\omega)} \cdot \frac{F(b)}{F(c)} \right) \]

- Derived as solution to maximum-likelihood problem under Gaussian noise assumption
- Requires estimate of signal-to-noise ratio at each frequency

\[ \text{SNR}(\omega) = \frac{\text{signal variance at } \omega}{\text{noise variance at } \omega} \]
Natural image and noise spectra

Natural images *tend* to have spectrum that scales as $1 / \omega^2$

- This is a *natural image statistic*

http://www.cnbc.cmu.edu/cns/papers/
Natural image and noise spectra

Natural images *tend* to have spectrum that scales as $1 / \omega^2$

- This is a *natural image statistic*

Noise tends to have flat spectrum, $\sigma(\omega) = \text{constant}$

- We call this white noise

What is the SNR?
Natural image and noise spectra

Natural images *tend* to have spectrum that scales as $1 / \omega^2$
- This is a *natural image statistic*

Noise tends to have flat spectrum, $\sigma(\omega) = \text{constant}$
- We call this white noise

Therefore, we have that: $\text{SNR}(\omega) = 1 / \omega^2$
Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

\[ X_{est} = F^{-1} \left( \frac{|F(c)|^2}{|F(c)|^2 + \omega^2} \cdot \frac{F(b)}{F(c)} \right) \]

amplitude-dependent damping factor

- Derived as solution to maximum-likelihood problem under Gaussian noise assumption
- Requires noise of signal-to-noise ratio at each frequency

\[ SNR(\omega) = \frac{1}{\omega^2} \]
Wiener Deconvolution

For natural images and white noise, equivalent to the minimization problem:

$$\min_x \|b - c \ast x\|^2 + \|\nabla x\|^2$$

gradient regularization

How can you prove this equivalence?
Wiener Deconvolution

For natural images and white noise, it can be re-written as the minimization problem

$$\min_x \|b - c \ast x\|^2 + \|\nabla x\|^2$$

How can you prove this equivalence?

• Convert to Fourier domain and repeat the proof for Wiener deconvolution.
• Intuitively: The $\omega^2$ term in the denominator of the special Wiener filter is the square of the Fourier transform of $\nabla x$, which is $i \cdot \omega$. 
Deconvolution comparisons

blury input  naive deconvolution  gradient regularization  original
Deconvolution comparisons

blurry input  naive deconvolution  gradient regularization  original
... and a proof-of-concept demonstration

noisy input  naive deconvolution  gradient regularization
Question

Can we undo lens blur by deconvolving a PNG or JPEG image without any preprocessing?
Question

Can we undo lens blur by deconvolving a PNG or JPEG image without any preprocessing?
• All the blur processes we discuss today happen *optically* (before capture by the sensor).
• Blur model is accurate only if our images are *linear*.

Are PNG or JPEG images linear?
Question

Can we undo lens blur by deconvolving a PNG or JPEG image without any preprocessing?
• All the blur processes we discuss today happen *optically* (before capture by the sensor).
• Blur model is accurate only if our images are *linear*.

Are PNG or JPEG images linear?
• No, because of gamma encoding.
• Before deblurring, you must linearize your images.

How do we linearize PNG or JPEG images?
The importance of linearity

blurry input  deconvolution without linearization  deconvolution after linearization  original
Can we do better than that?
Can we do better than that?

Use different gradient regularizations:

- $L_2$ gradient regularization (Tikhonov regularization, same as Wiener deconvolution)
  \[ \min_x \|b - c \ast x\|^2 + \|\nabla x\|^2 \]

- $L_1$ gradient regularization (sparsity regularization, isotropic total variation)
  \[ \min_x \|b - c \ast x\|^2 + \|\nabla x\|^1 \]

- Anisotropic total variation
  \[ \min_x \|b - c \ast x\|^2 + \|\nabla x\|^2 \]

All of these are motivated by natural image statistics. Active research area.

How are these two different?
Total Variation

\[ x \]

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

better: isotropic

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

easier: anisotropic
Total Variation

\[
\min_{x} \|Cx - b\|_2^2 + \lambda TV(x) = \min_{x} \|Cx - b\|_2^2 + \lambda \|\nabla x\|_1
\]

\[
\|x\|_1 = \sum_i |x_i|
\]

- idea: promote sparse gradients (edges)

- $\nabla$ is finite differences operator, i.e. matrix

\[
\begin{bmatrix}
-1 & 1 \\
-1 & 1 \\
\vdotswithin{-1} & \\
-1
\end{bmatrix}
\]

Rudin et al. 1992
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: $l_1$-norm is not differentiable, can't use inverse filtering

- however: simple solution for data fitting along and simple solution for TV alone $\Rightarrow$ split problem!
Deconvolution with ADMM

- split deconvolution with TV prior:

\[
\text{minimize} \quad \|Cx - b\|_2^2 + \lambda \|z\|_1 \\
\text{subject to} \quad \nabla x = z
\]

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

\[
\begin{align*}
\text{minimize} & \quad f(x) + g(z) \\
\text{subject to} & \quad Ax + Bz = c
\end{align*}
\]

\[
\begin{align*}
f(x) &= \|Cx - b\|_2^2 \\
g(z) &= \lambda \|z\|_1 \\
A &= \nabla, \quad B = -I, \quad c = 0
\end{align*}
\]
minimize \( f(x) + g(z) \) \hspace{2cm} \text{ADMM} \\
subject to \( Ax + Bz = c \)

- 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
minimize \( f(x) + g(z) \) \quad \text{ADMM}

subject to \( Ax + Bz = c \)

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

\[
L_\rho(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \left(\frac{\rho}{2}\right)\|Ax + Bz - c\|_2^2
\]
minimize \quad f(x) + g(z) \quad \text{ADMM}

subject to \quad Ax + Bz = c

- ADMM consists of 3 steps per iteration $k$:

\[ x^{k+1} := \arg \min_x L_\rho(x, z^k, y^k) \]
\[ z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, y^k) \]
\[ y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \]
minimize \( f(x) + g(z) \)
subject to \( Ax + Bz = c \)

- ADMM consists of 3 steps per iteration \( k \):

\[
\begin{align*}
    x^{k+1} & := \arg \min_x \left( f(x) + \left( \frac{\rho}{2} \right) \| Ax + Bz^k - c + u^k \| \right) \\
z^{k+1} & := \arg \min_z \left( g(z) + \left( \frac{\rho}{2} \right) \| Ax^{k+1} + Bz - c + u^k \| \right) \\
u^{k+1} & := u^{k} + Ax^{k+1} + Bz^{k+1} - c
\end{align*}
\]

scaled dual variable: \( u = (1/\rho)y \)
minimize $f(x) + g(z)$  
subject to $Ax + Bz = c$

- ADMM consists of 3 steps per iteration $k$:
  - split $f(x)$ and $g(x)$ into independent problems!
  - $x^{k+1} := \arg\min_x \left( f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + u^k\|_2^2 \right)$
  - $z^{k+1} := \arg\min_z \left( g(z) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + u^k\|_2^2 \right)$
  - $u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c$

  scaled dual variable: $u = (1/\rho)y$
minimize \( \frac{1}{2} \| Cx - b \|_2^2 + \lambda \| z \|_1 \) \quad \text{Deconvolution with ADMM}

subject to \( \nabla x - z = 0 \)

- ADMM consists of 3 steps per iteration \( k \):

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

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

\[
u^{k+1} := u^k + \nabla x^{k+1} - z^{k+1}
\]
minimize \( \frac{1}{2} \|Cx - b\|_2^2 + \lambda \|z\|_1 \)

subject to \( \nabla x - z = 0 \)

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

solve normal equations \( (C^TC + \rho \nabla^T \nabla)x = (C^Tb + \rho \nabla^T v) \)

\( \nabla^T v = \begin{bmatrix} \nabla_x^T \\ \nabla_y^T \end{bmatrix} \)

\( v = \nabla_x^T v_1 + \nabla_y^T v_2 \)
minimize \[ \frac{1}{2} \|Cx - b\|_2^2 + \lambda \|z\|_1 \]

subject to \[ \nabla x - z = 0 \]

constant, say \( v = z^k - u^k \)

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

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

- inverse filtering: \( x^{k+1} = F^{-1} \left[ \begin{array}{c} F\{c\} \ast F\{b\} + \rho \left( F\{\nabla x\} \ast F\{v_1\} + F\{\nabla y\} \ast F\{v_2\} \right) \\
F\{c\} \ast F\{c\} + \rho \left( F\{\nabla x\} \ast F\{\nabla x\} + F\{\nabla y\} \ast F\{\nabla y\} \right) \end{array} \right] \)
minimize \( \frac{1}{2} \|Cx - b\|_2^2 + \lambda \|z\|_1 \)

subject to \( \nabla x - z = 0 \)

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

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

subject to $\nabla x - z = 0$

for $k=1:\text{max}_\text{iters}$

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

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

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

Wiener deconvolution

ADMM + TV, $\lambda = 0.01$
- image becomes too flat as we increase weight of TV prior

ADMM + TV, $\lambda = 0.1$
- Image becomes too noisy as we decrease weight of TV prior
Deconvolution comparisons

- Wiener deconvolution
- ADMM + TV, $\lambda = 0.01$
  - Image becomes too flat as we increase weight of TV prior
- ADMM + TV, $\lambda = 0.1$
  - Image becomes too noisy as we decrease weight of TV prior
Outlook ADMM

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

\[
\text{minimize } \frac{1}{2} ||Ax - b||_2^2 + \Gamma(x) \quad \rightarrow \quad \text{minimize } \begin{cases} f(x) + g(z) \\ \text{subject to } Ax = z \end{cases}
\]

- example priors: noise statistics, sparse gradient, smoothness, ...
- weighted sum of different priors also possible
- anisotropic TV is one of the easiest priors
Can we do better than that?

Use different gradient regularizations:

- $L_2$ gradient regularization (Tikhonov regularization, same as Wiener deconvolution)
  \[
  \min_x \| b - c \ast x \|^2 + \| \nabla x \|^2
  \]

- $L_1$ gradient regularization (sparsity regularization, same as total variation)
  \[
  \min_x \| b - c \ast x \|^2 + \| \nabla x \|^1
  \]

- $L_{n<1}$ gradient regularization (fractional regularization)
  \[
  \min_x \| b - c \ast x \|^2 + \| \nabla x \|^{0.8}
  \]

All of these are motivated by natural image statistics. Active research area.
Comparison of gradient regularizations

input  squared gradient regularization  fractional gradient regularization
Derivation

Sensing model:

\[ \tilde{x} = c \times x + n \]

Noise \( n \) is assumed to be zero-mean and independent of signal \( x \).

Is this a reasonable noise model?
Richardson-Lucy Algorithm + TV

- log-likelihood function:

$$\log (L_{TV}(x)) = \log (p(b|x)) + \log (p(x)) = \log (A^T b - (Ax)^T 1 - \sum_{i=1}^{M} \log (b_i!)) - \lambda \|Dx\|_1$$

- gradient:

$$\nabla \log (L_{TV}(x)) = A^T \text{diag}(Ax)^{-1} b - A^T 1 + \nabla \lambda \|\nabla x\|_1 = A^T \left( \frac{b}{Ax} \right) - A^T 1 - \nabla \lambda \|Dx\|_1$$

- recover signal by setting gradient to zero
- generally challenging
High quality images using cheap lenses

[Heide et al., “High-Quality Computational Imaging Through Simple Lenses,” TOG 2013]
Deconvolution

If we know b and c, can we recover x?

How do we measure this?

\[ X \ast c = b \]
PSF calibration

Take a photo of a point source

Image of PSF

Image with sharp lens  Image with cheap lens
Deconvolution

If we know \( b \) and \( c \), can we recover \( x \)?

\[
X \ast c = b
\]
Blind deconvolution

If we know $b$, can we recover $x$ and $c$?

$$X * C = b$$
Camera shake

Removing Camera Shake from a Single Photograph

Rob Fergus\(^1\)  Barun Singh\(^1\)  Aaron Hertzmann\(^2\)  Sam T. Roweis\(^2\)  William T. Freeman\(^1\)

\(^1\)MIT CSAIL \quad \(^2\)University of Toronto

Figure 1: Left: An image spoiled by camera shake. Middle: result from Photoshop “unsharp mask”. Right: result from our algorithm.
Camera shake as a filter

If we know $b$, can we recover $x$ and $c$?

$X * c = b$

image from static camera
PSF from camera motion
image from shaky camera
Multiple possible solutions

How do we detect this one?
Use prior information

Among all the possible pairs of images and blur kernels, select the ones where:

• The image “looks like” a natural image.

• The kernel “looks like” a motion PSF.
Use prior information

Among all the possible pairs of images and blur kernels, select the ones where:

- The image “looks like” a natural image.
- The kernel “looks like” a motion PSF.
Shake kernel statistics

Gradients in natural images follow a characteristic “heavy-tail” distribution.
Shake kernel statistics

Gradients in natural images follow a characteristic “heavy-tail” distribution.

Can be approximated by $\| \nabla x \|^{0.8}$
Use prior information

Among all the possible pairs of images and blur kernels, select the ones where:

• The image “looks like” a natural image.

  Gradients in natural images follow a characteristic “heavy-tail” distribution.

• The kernel “looks like” a motion PSF.

  Shake kernels are very sparse, have continuous contours, and are always positive.

How do we use this information for blind deconvolution?
Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x,b} \|b - c * x\|^2 + \|\nabla x\|^{0.8} + \|c\|_1$$

What does each term in this summation correspond to?
Regularized blind deconvolution

Solve regularized least-squares optimization

\[
\min_{x, b} \| b - c * x \|^2 + \| \nabla x \|^{0.8} + \| c \|_1
\]

data term  \hspace{1cm} natural image prior  \hspace{1cm} shake kernel prior

Note: Solving such optimization problems is complicated (no longer linear least squares).
A demonstration

input  deconvolved image and kernel
A demonstration

This image looks worse than the original...

This doesn’t look like a plausible shake kernel...
Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x, b} \|b - c \ast x\|^2 + \|\nabla x\|^{0.8} + \|c\|_1$$

loss function
Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x, b} \|b - c \ast x\|^2 + \|\nabla x\|^{0.8} + \|c\|_1$$

Where in this graph is the solution we find?
Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x, b} \|b - c \ast x\|^2 + \|\nabla x\|^{0.8} + \|c\|_1$$

inverse loss

loss function

many plausible solutions here

Rather than keep just maximum, do a weighted average of all solutions
A demonstration

input          maximum-only          average
More examples
Results on real shaky images
Results on real shaky images
Results on real shaky images
Results on real shaky images
More advanced motion deblurring

[Shah et al., High-quality Motion Deblurring from a Single Image, SIGGRAPH 2008]
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.

Can we solve all of these problems using (blind) deconvolution?
Why are our images blurry?

- Lens imperfections.
- Camera shake.
- Scene motion.
- Depth defocus.

Can we solve all of these problems using (blind) deconvolution?

- We can deal with (some) lens imperfections and camera shake, because their blur is shift invariant.
- We cannot deal with scene motion and depth defocus, because their blur is not shift invariant.
- See coded photography lecture.
References

Basic reading:
- Szeliski textbook, Sections 3.4.3, 3.4.4, 10.1.4, 10.3.
  the main motion deblurring and blind deconvolution paper we covered in this lecture.

Additional reading:
  the paper on high-quality imaging using cheap lenses, which also has a great discussion of all matters relating to
  blurring from lens aberrations and modern deconvolution algorithms.
  a sequence of papers developing the state of the art in blind deconvolution of natural images, including the use
  Laplacian (sparsity) and hyper-Laplacian priors on gradients, analysis of different loss functions and maximum a-
  posteriori versus Bayesian estimates, the use of variational inference, and efficient optimization algorithms.
  the paper explaining the mathematics of how to compute Bayesian estimators using variational inference.
  a more recent paper on motion deblurring.