Deconvolution

15-463, 15-663, 15-862
Computational Photography
Fall 2017, Lecture 17

http://graphics.cs.cmu.edu/courses/15-463

Course announcements

- Homework 4 is out.
  - Due October 26th.
  - There was another typo in HW4, download new version.
  - Drop by Yannis’ office to pick up cameras any time.

- Homework 5 will be out on Thursday.
  - You will need cameras for that one as well, so keep the ones you picked up for HW4.

- Project ideas were due on Piazza on Friday 20th.
  - Responded to most of you.
  - Some still need to post their ideas.

- Project *proposals* are due on Monday 30th.

Overview of today’s lecture

- Telecentric lenses.
- Sources of blur.
- Deconvolution.
- Blind deconvolution.

Slide credits

Most of these slides were adapted from:

- Fredo Durand (MIT).
- Gordon Wetzstein (Stanford).

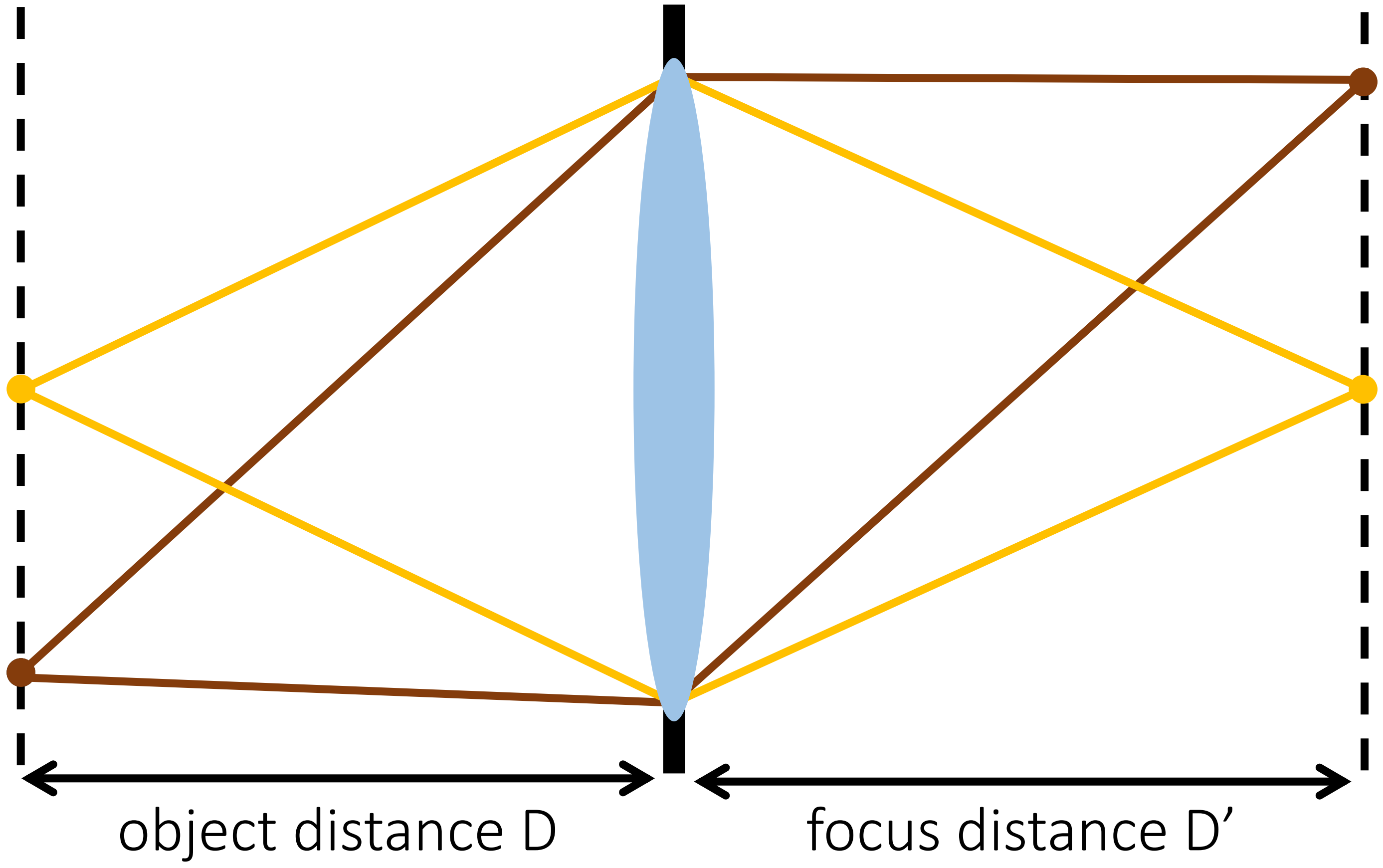
Why are our images blurry?

Why are our images blurry?

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

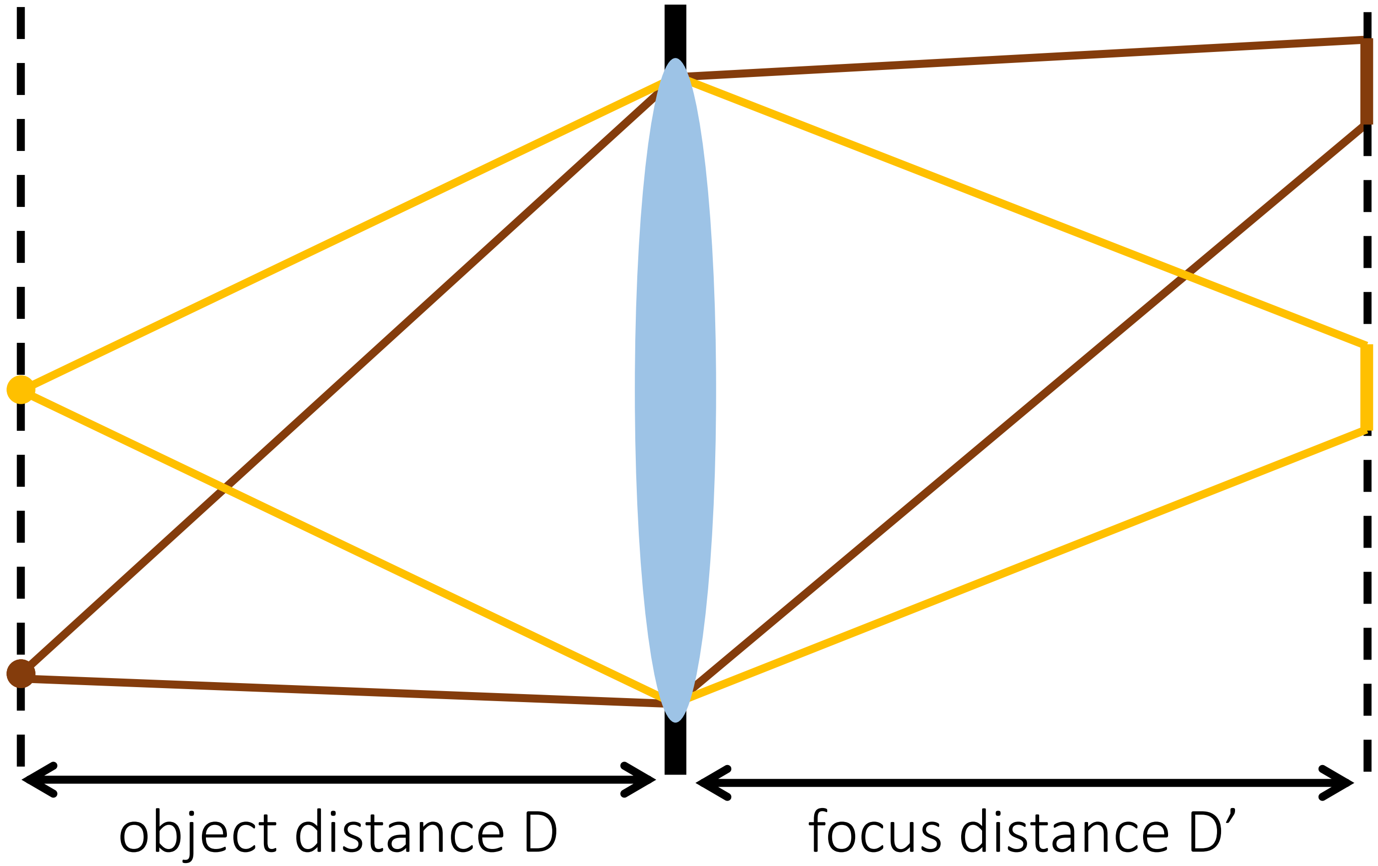
Lens imperfections

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



Lens imperfections

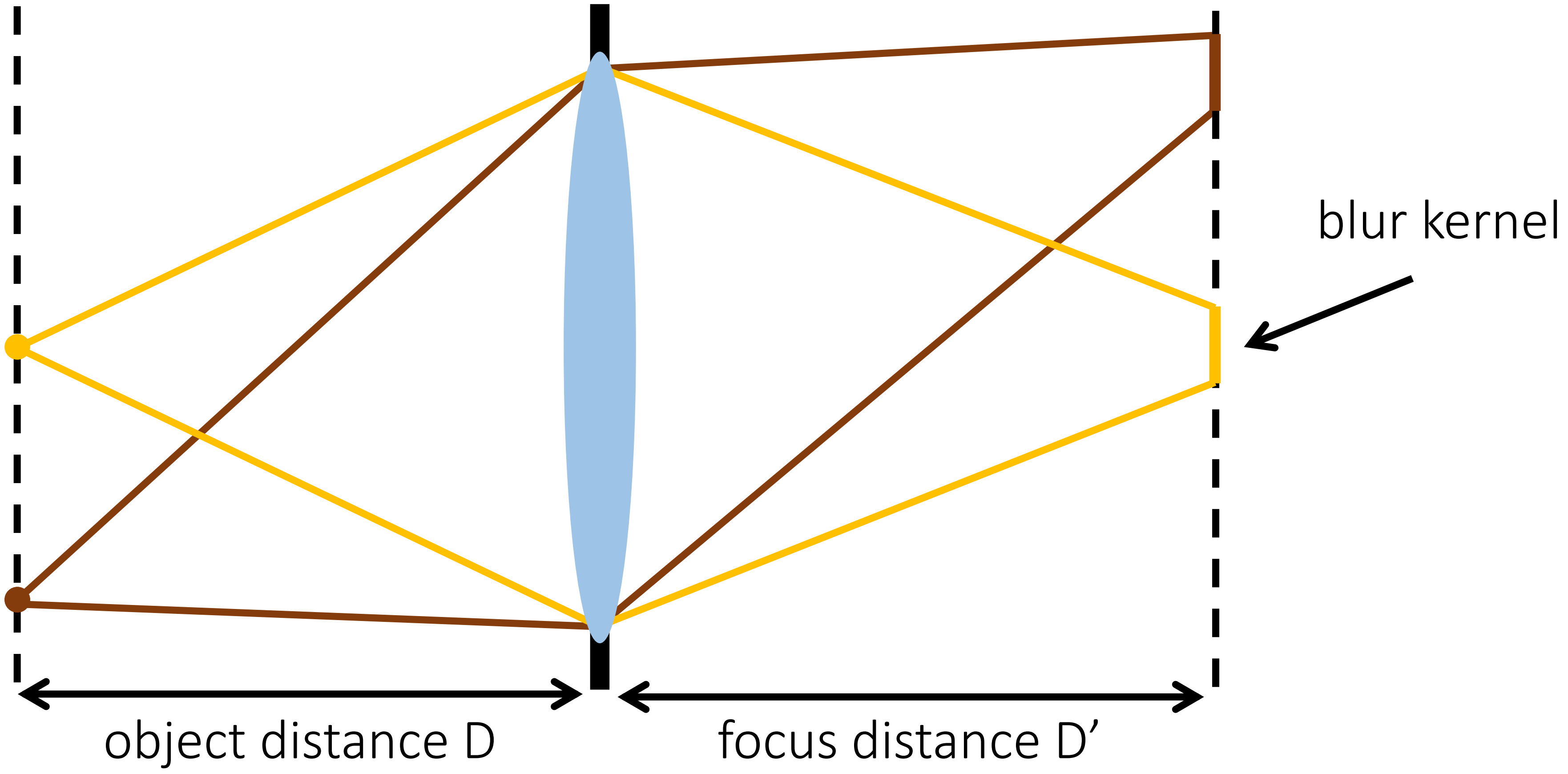
- Ideal lens: An point maps to a point at a certain plane.
- Real lens: A point maps to a circle that has non-zero minimum radius among all planes.



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.

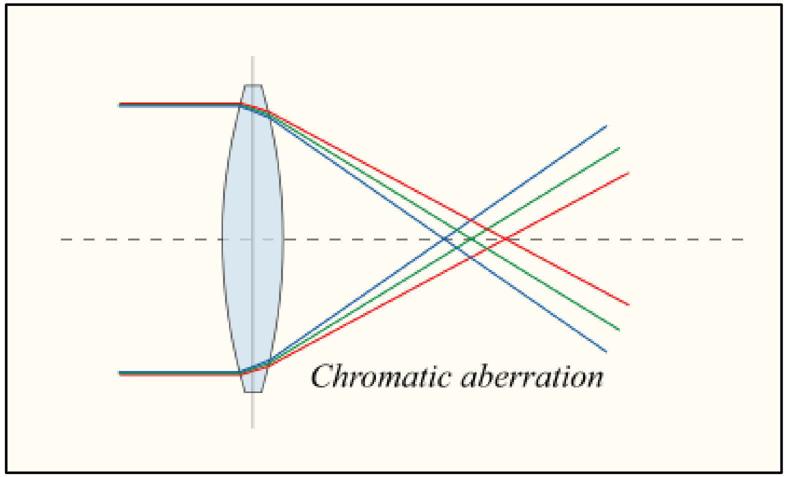
Lens imperfections

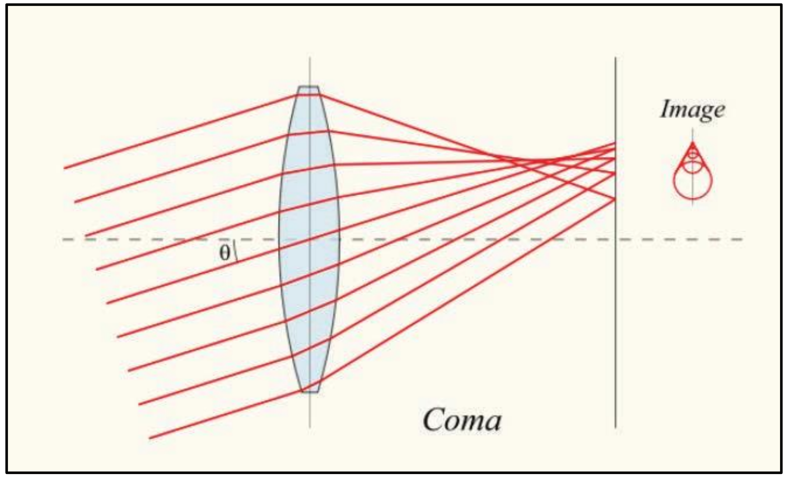
What causes lens imperfections?

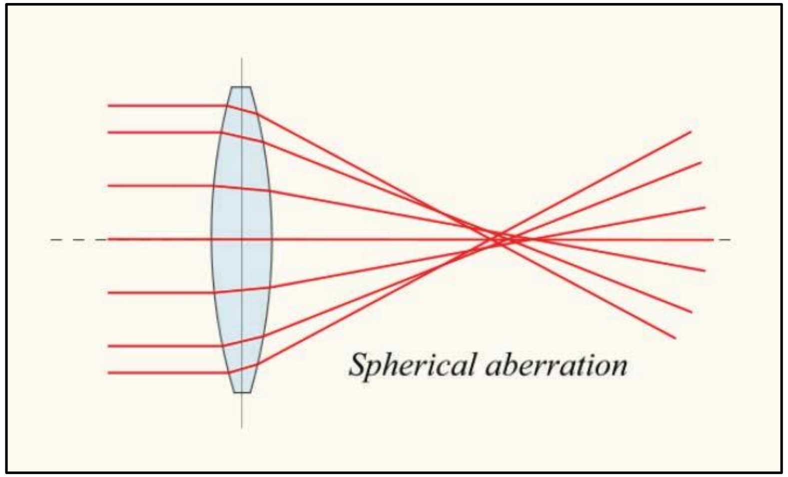
Lens imperfections

What causes lens imperfections?

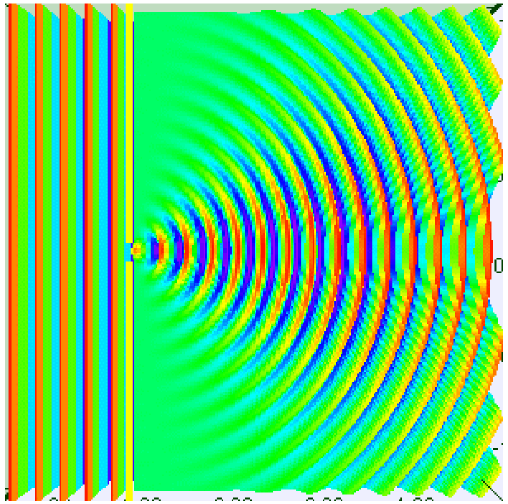
- Aberrations.



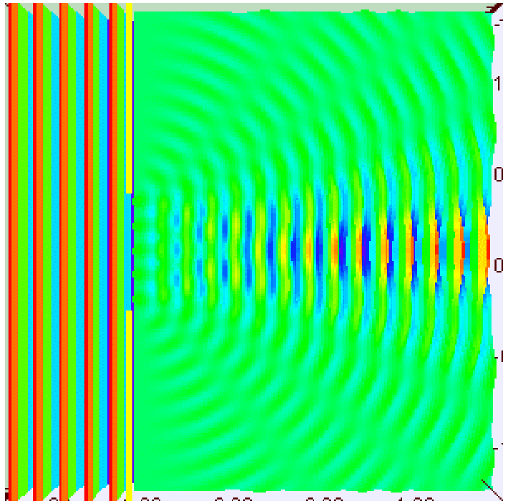




- Diffraction.



small aperture

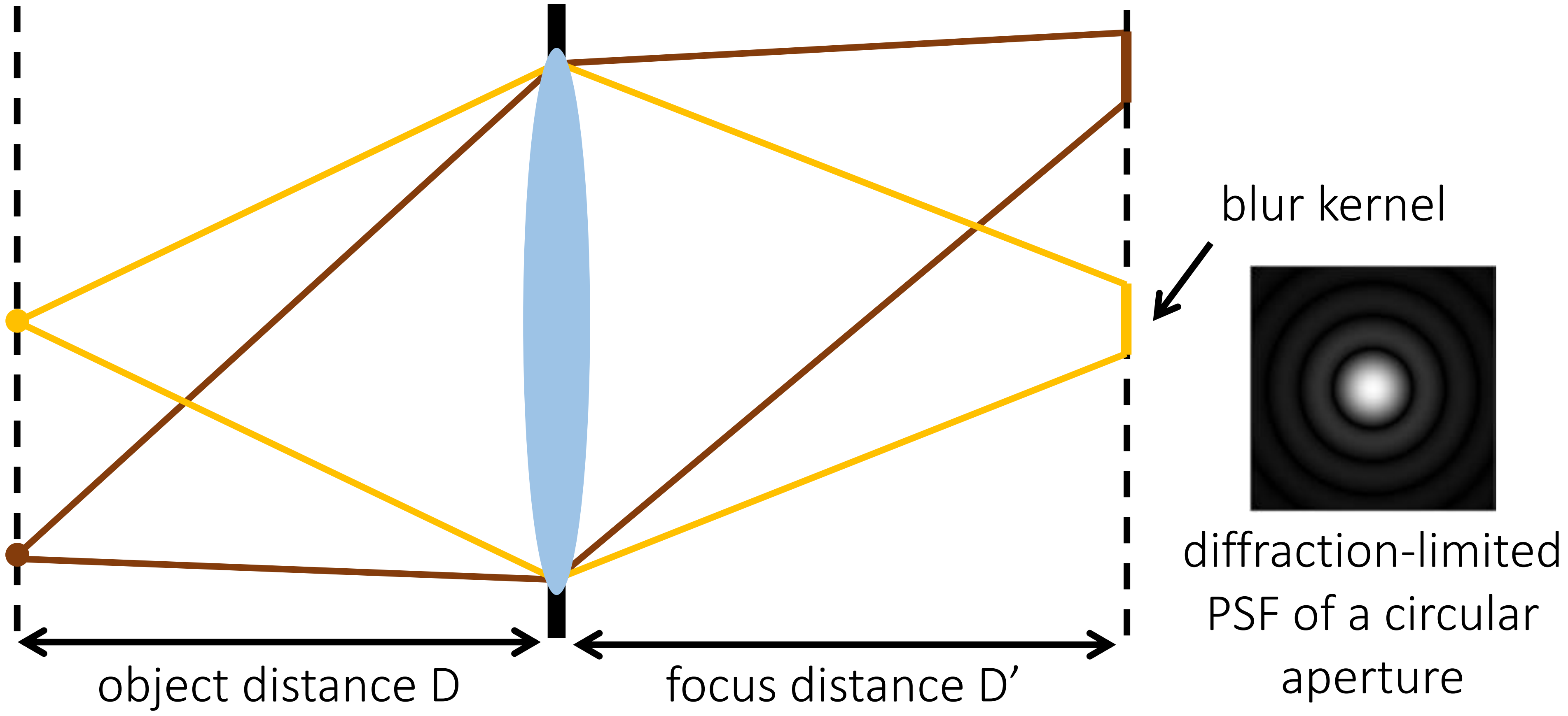


large aperture

Lens as an optical low-pass filter

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

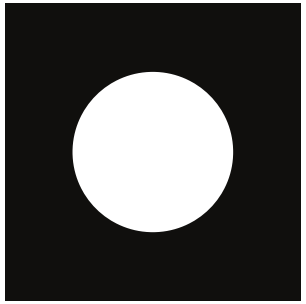
- "Diffraction-limited" PSF: No aberrations, only diffraction. Determined by aperture shape.



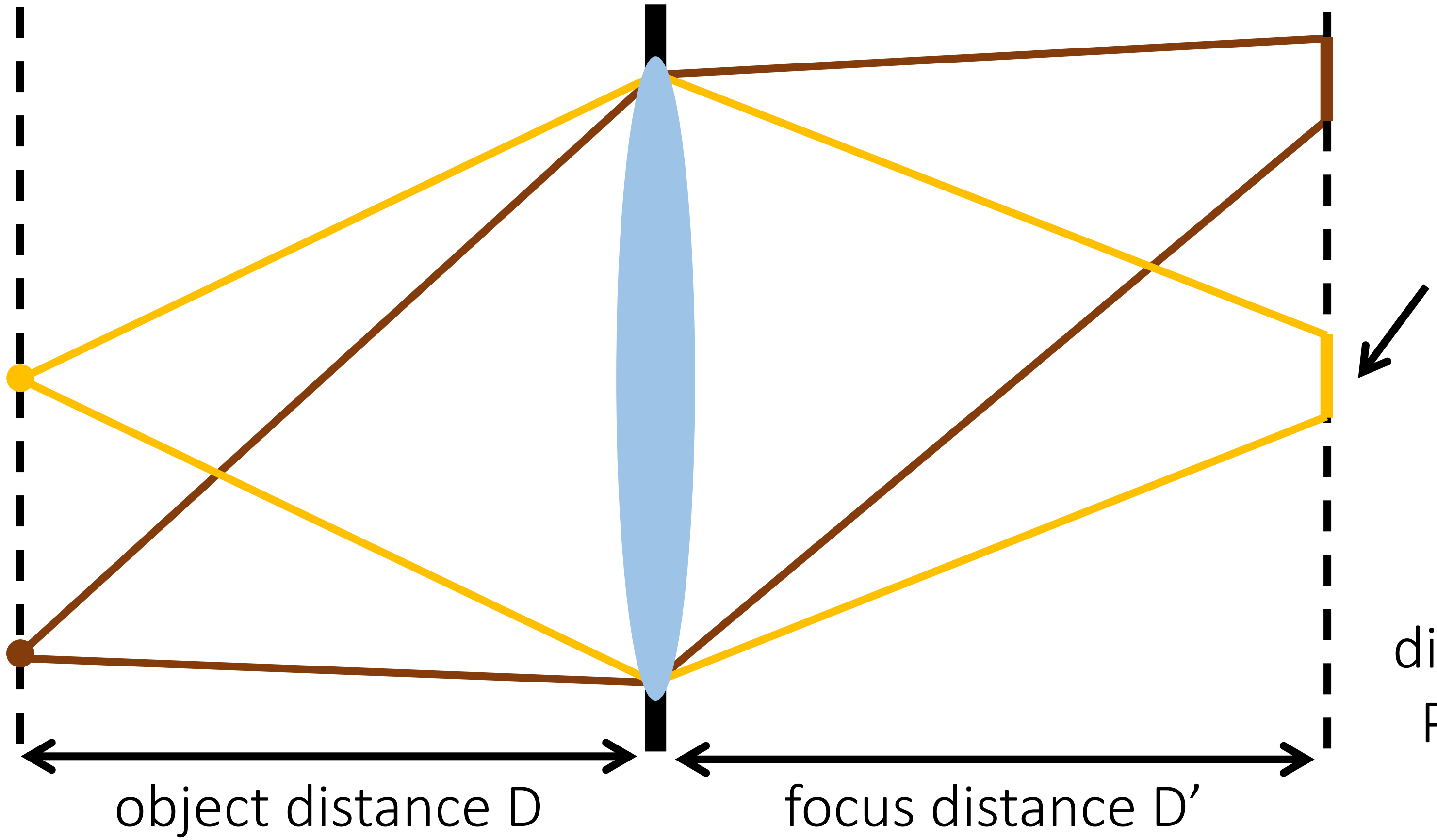
Lens as an optical low-pass filter

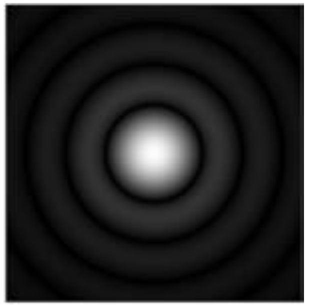
Point spread function (PSF): The blur kernel of a lens.

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



diffraction-limited OTF of a circular aperture





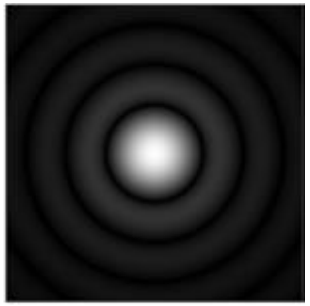
diffraction-limited PSF of a circular aperture

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

=

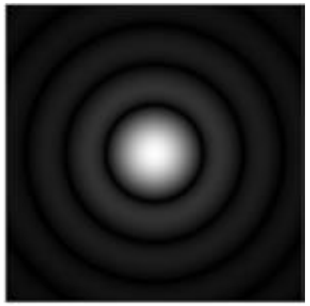
image from imperfect lens

$$x \quad * \quad c \quad = \quad b$$

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

$$x \quad * \quad c \quad = \quad b$$

Deconvolution

$$x \quad * \quad c \quad = \quad b$$

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

Deconvolution

$$x \quad * \quad c \quad = \quad b$$

Reminder: convolution is multiplication in Fourier domain:

$$F(x) \quad \cdot \quad F(c) \quad = \quad F(b)$$

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

Deconvolution

$$x \quad * \quad c \quad = \quad b$$

Reminder: convolution is multiplication in Fourier domain:

$$F(x) \cdot F(c) = F(b)$$

Deconvolution is division in Fourier domain:

$$F(x_{est}) = F(c) \setminus F(b)$$

After division, just do inverse Fourier transform:

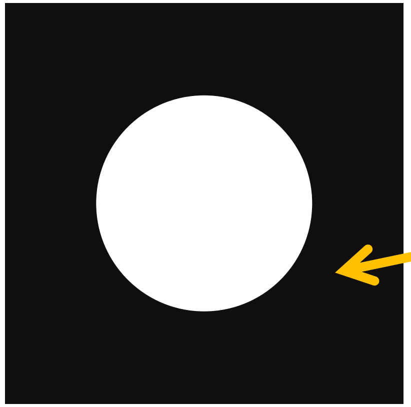
$$x_{est} = F^{-1}\left(F(c) \setminus F(b) \right)$$

Deconvolution

Any problems with this approach?

Deconvolution

- The OTF (Fourier of PSF) is a low-pass filter



zeros at high frequencies

- The measured signal includes noise

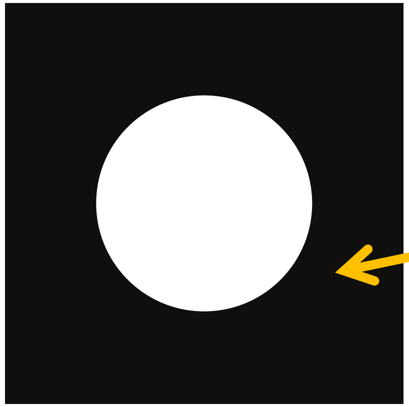
$$b = c * x + n$$

noise term

Any problems with this approach?

Deconvolution

- The OTF (Fourier of PSF) is a low-pass filter



zeros at high frequencies

- The measured signal includes noise

$$b = c * x + n$$

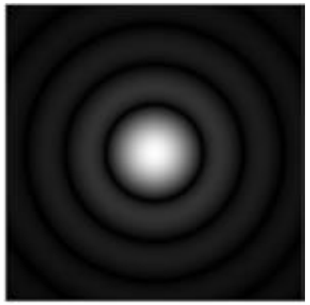
noise term

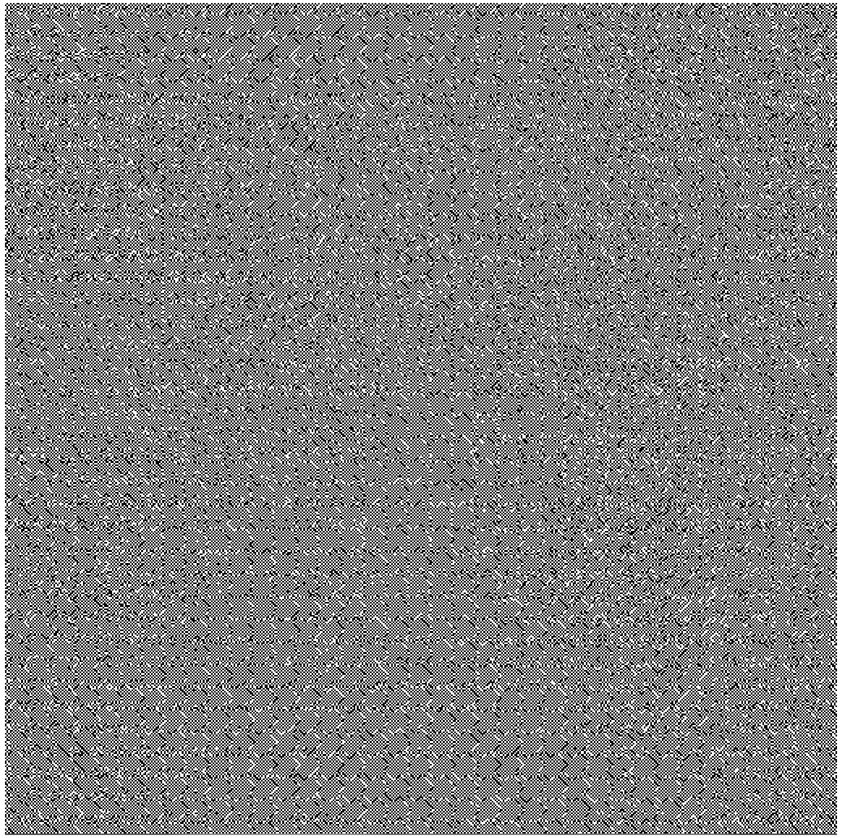
- When we divide by zero, we amplify the high frequency noise

Naïve deconvolution

Even tiny noise can make the results awful.

- Example for Gaussian of σ = 0.05

$*$ ()$^{-1}$ $=$



$$b \quad * \quad c^{-1} \quad = \quad x_{est}$$

Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

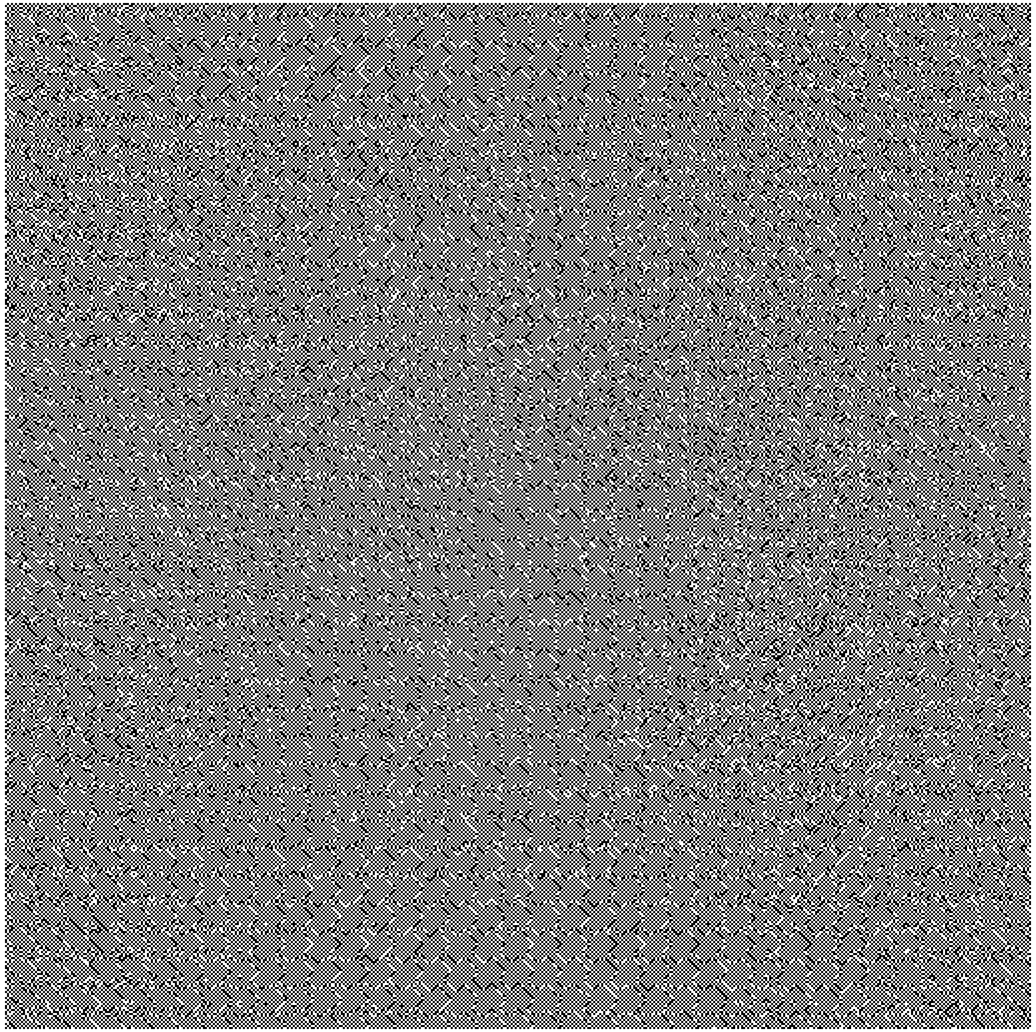
$$x_{est} = F^{-1}\left(\boxed{\frac{|F(c)|^2}{|F(c)|^2 + 1/SNR(\omega)}} \cdot \frac{F(b)}{F(c)}\right)$$

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

$$SNR(\omega) = \frac{\text{mean signal at } \omega}{\text{noise std at } \omega}$$

Deconvolution comparisons



naïve deconvolution

Wiener deconvolution

Deconvolution comparisons

σ = 0.01

σ = 0.05

σ = 0.01

Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

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

amplitude-dependent damping factor

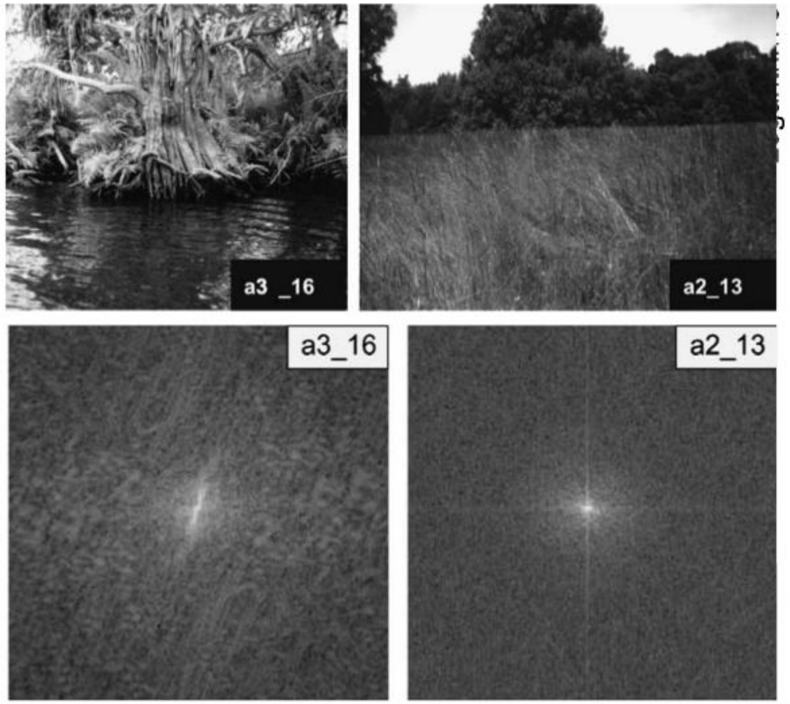
- Derived as solution to maximum-likelihood problem under Gaussian noise assumption
- $\boxed{\text{Requires noise of signal-to-noise ratio at each frequency}}$

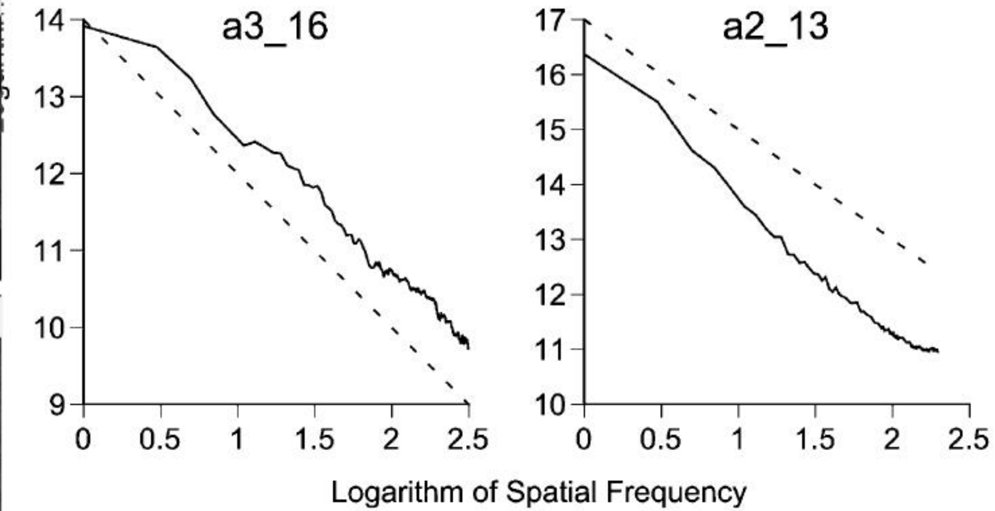
$$SNR(\omega) = \frac{\text{mean signal at } \omega}{\text{noise std at } \omega}$$

Natural image and noise spectra

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

- This is a *natural image statistic*



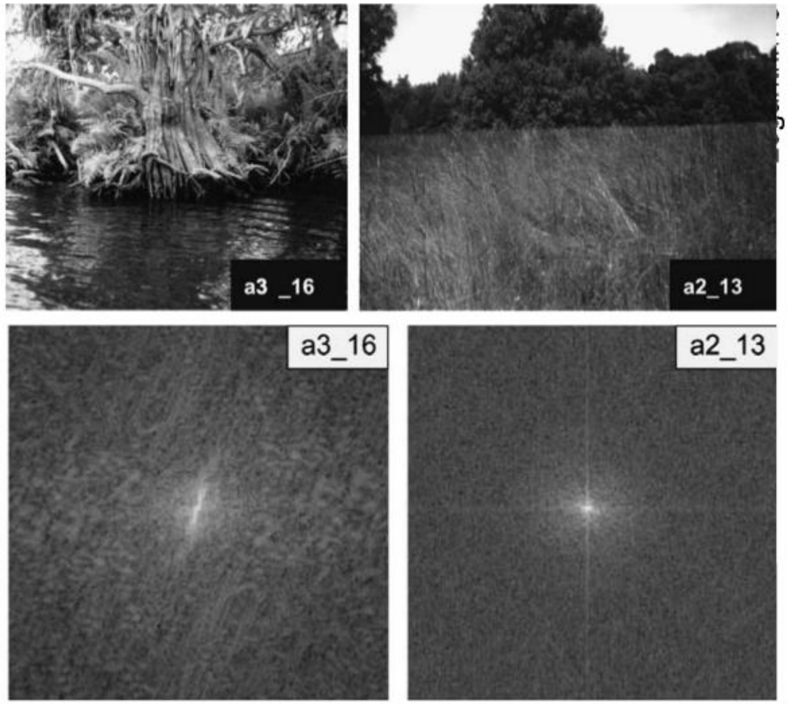


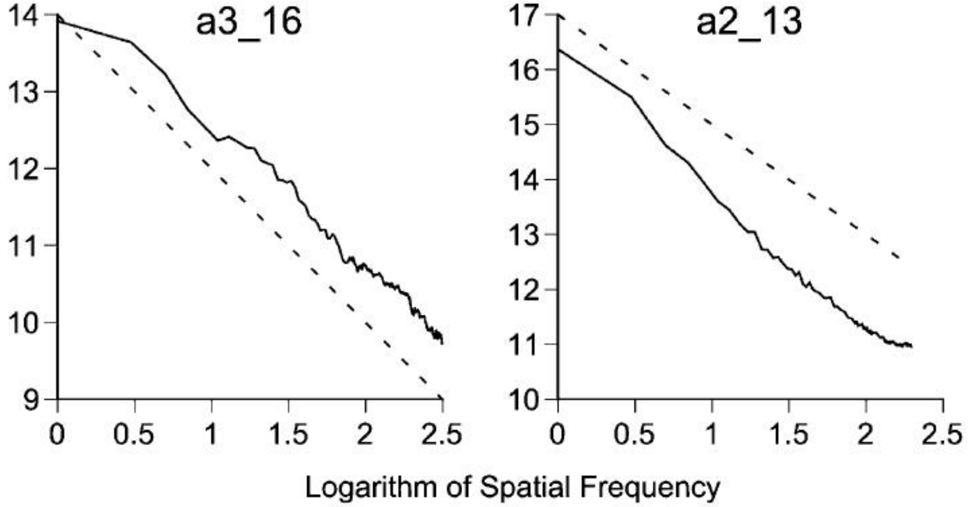
http://www.cnbc.cmu.edu/cns/papers/Balboa_PowerSpectra2003.pdf

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/Balboa_PowerSpectra2003.pdf

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

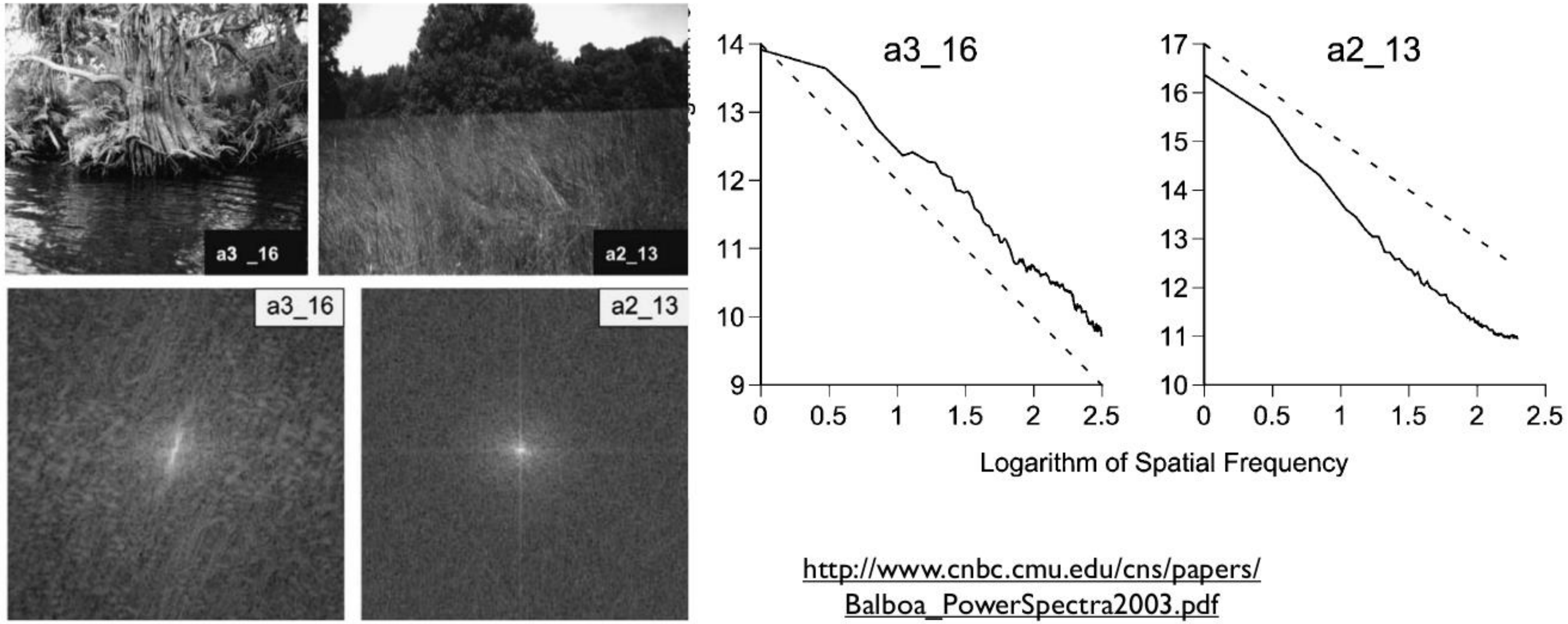
- We call this white noise

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*



http://www.cnbc.cmu.edu/cns/papers/Balboa_PowerSpectra2003.pdf

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

- We call this white noise

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

Wiener Deconvolution

Apply inverse kernel and do not divide by zero:

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

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

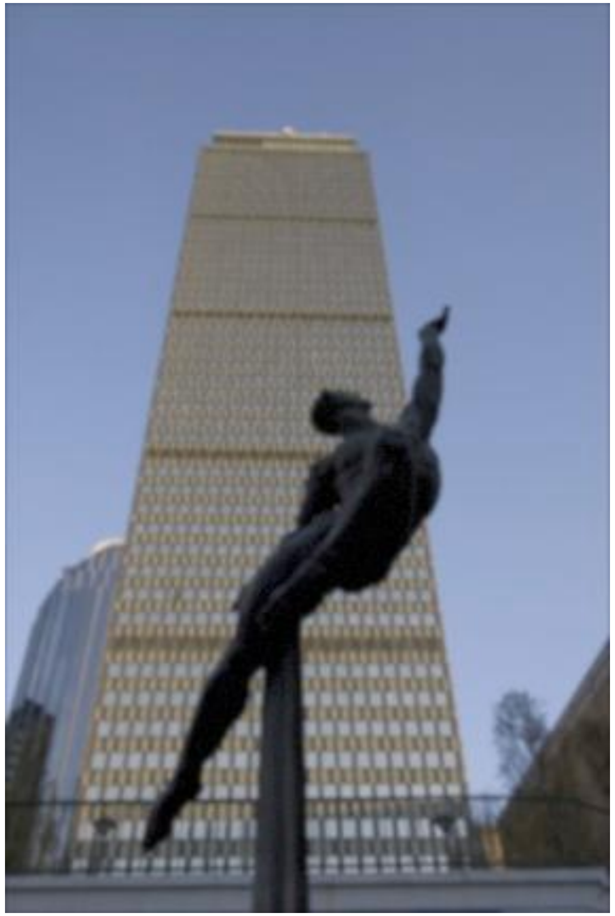
For natural images and white noise, it can be re-written as the minimization problem

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

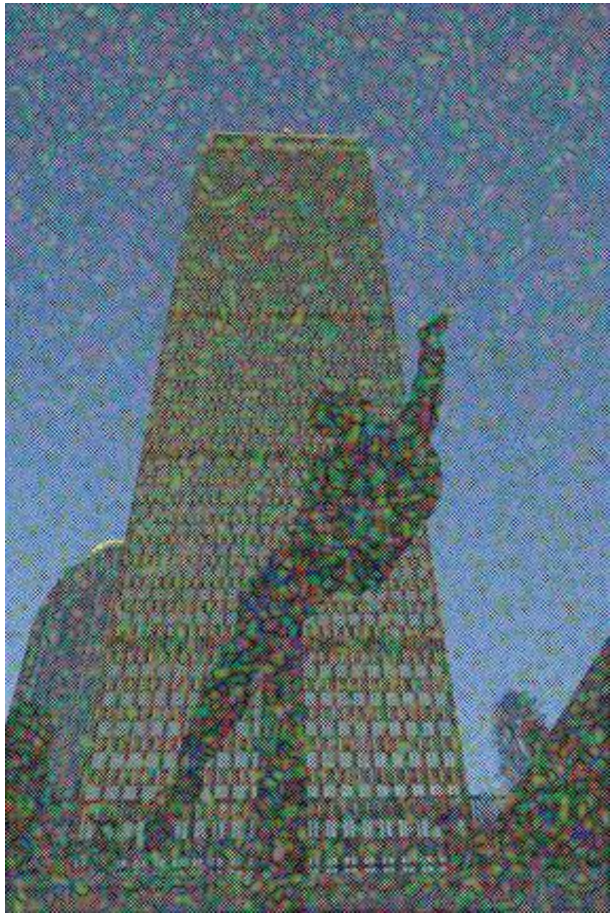
gradient *regularization*

- What does this look like?
- How can it be solved?

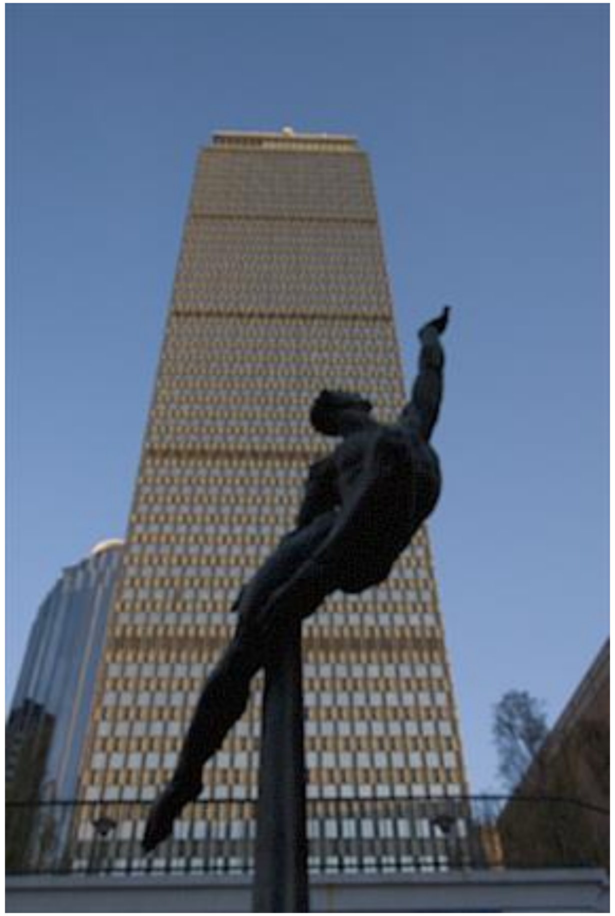
Deconvolution comparisons



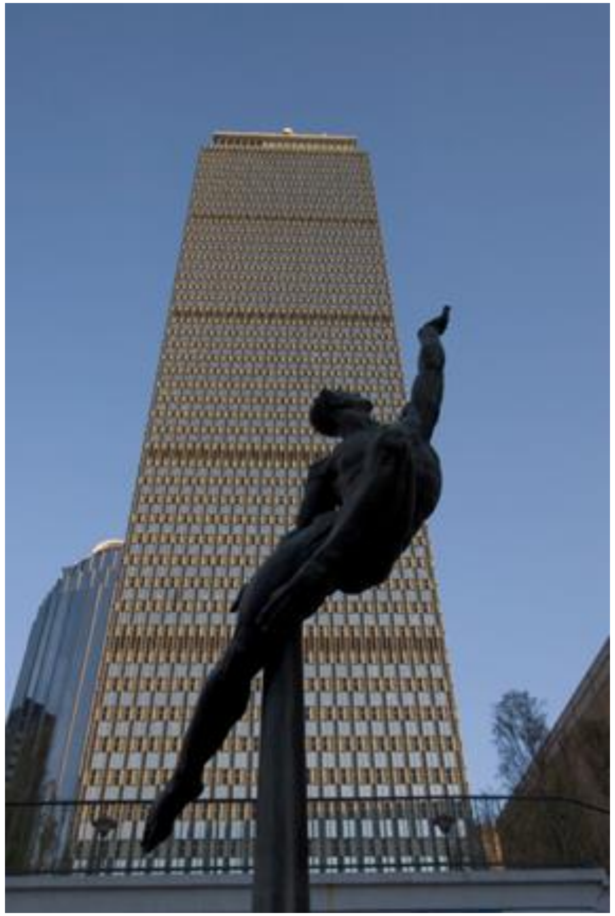
blurry input



naive deconvolution

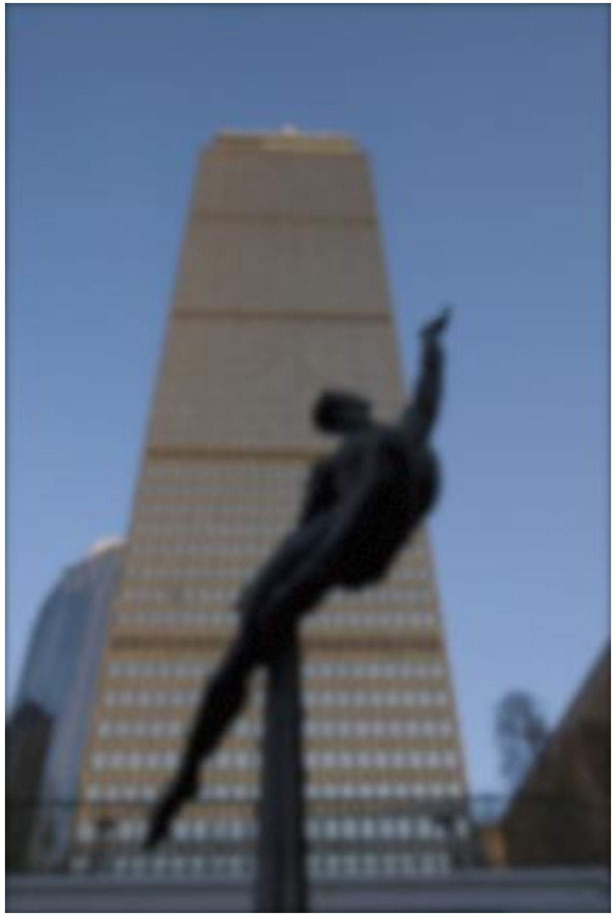


gradient regularization

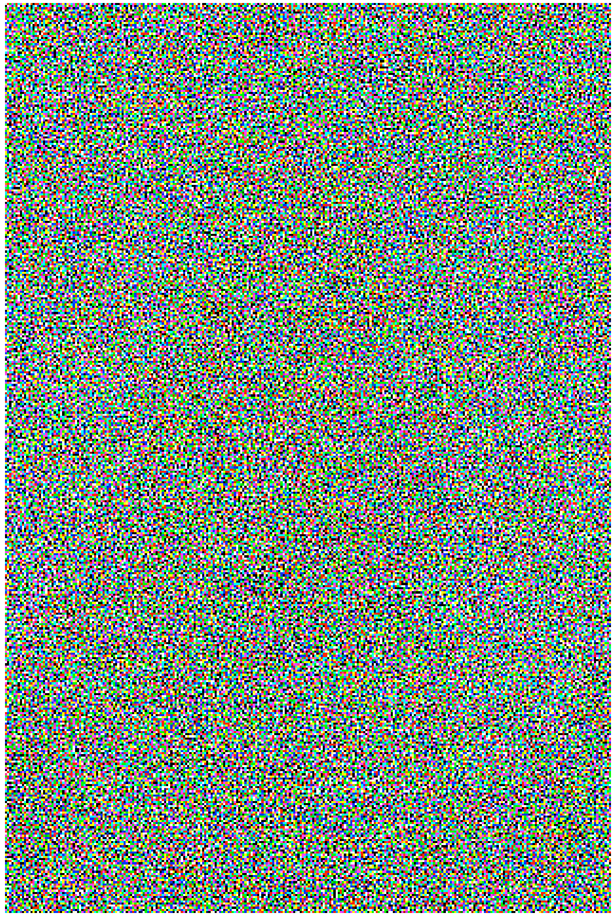


original

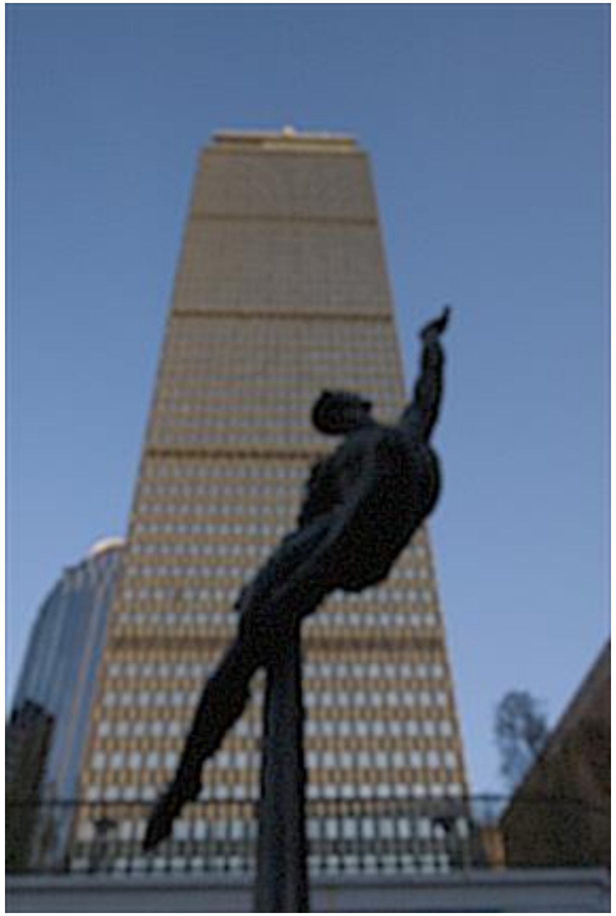
Deconvolution comparisons



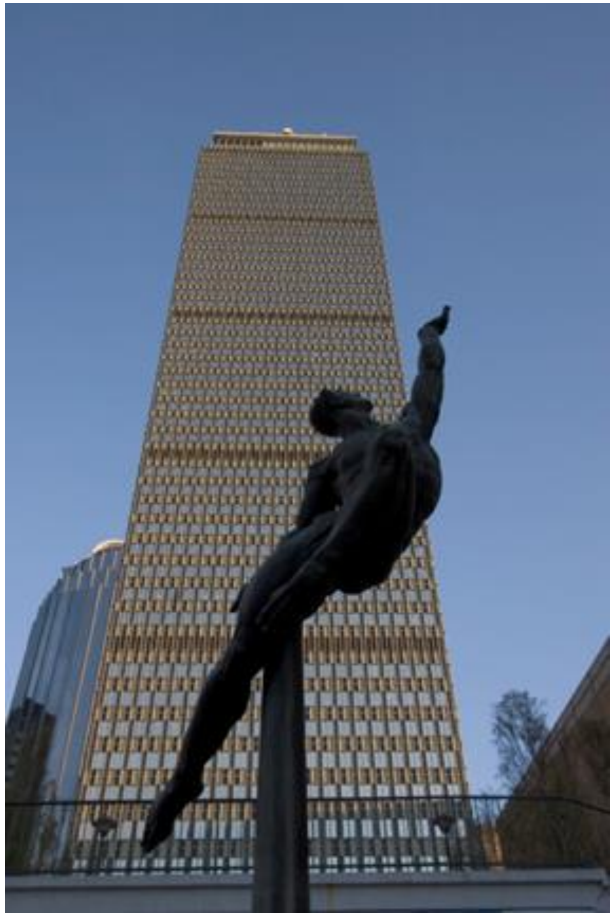
blurry input



naive deconvolution

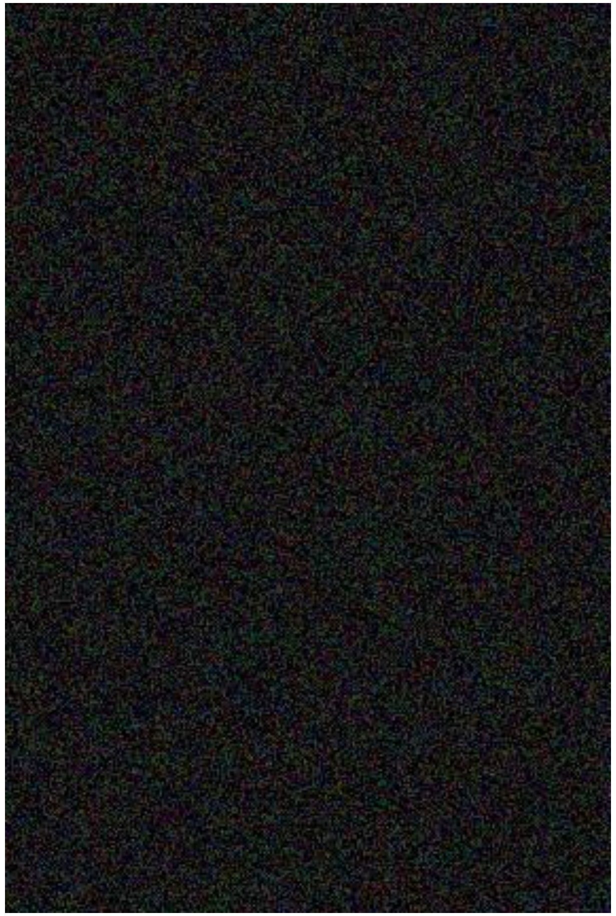


gradient regularization

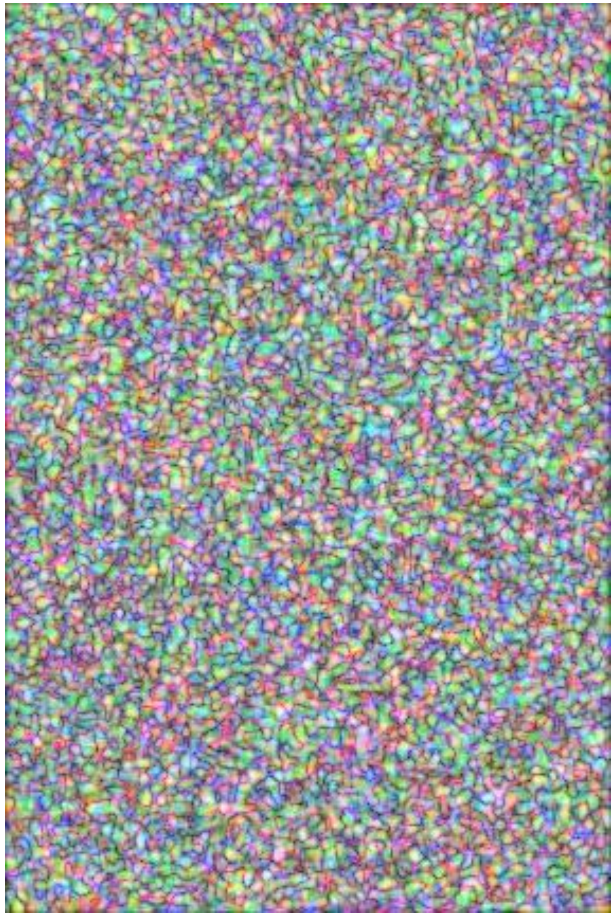


original

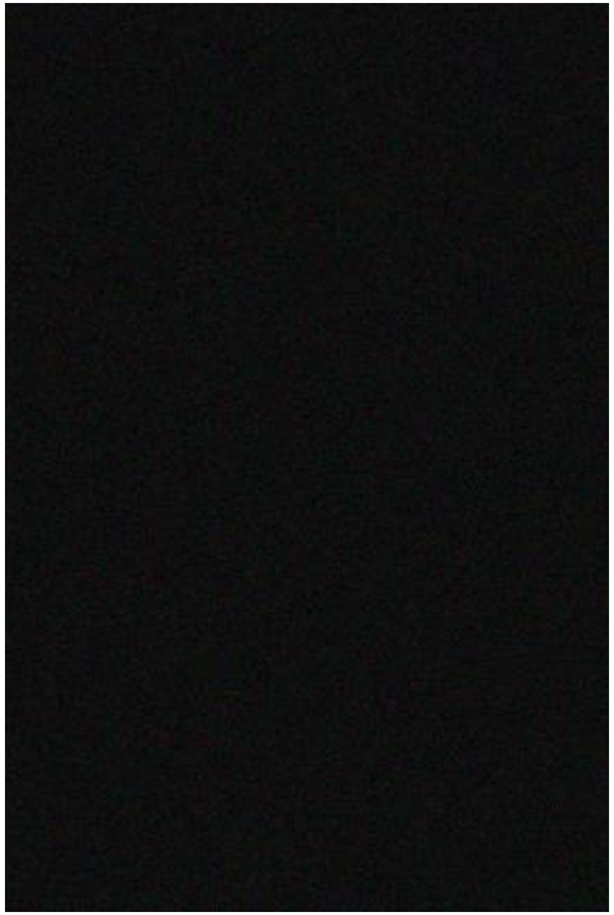
… and a proof-of-concept demonstration



noisy input



naive deconvolution



gradient regularization

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)

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

- L_1 gradient regularization (sparsity regularization, same as total variation)

$$\min_x ||b - c * x||^2 + ||\nabla x||^1$$

How do we solve for these?

- $L_{n<1}$ gradient regularization (fractional regularization)

$$\min_x ||b - c * x||^2 + ||\nabla x||^{0.8}$$

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

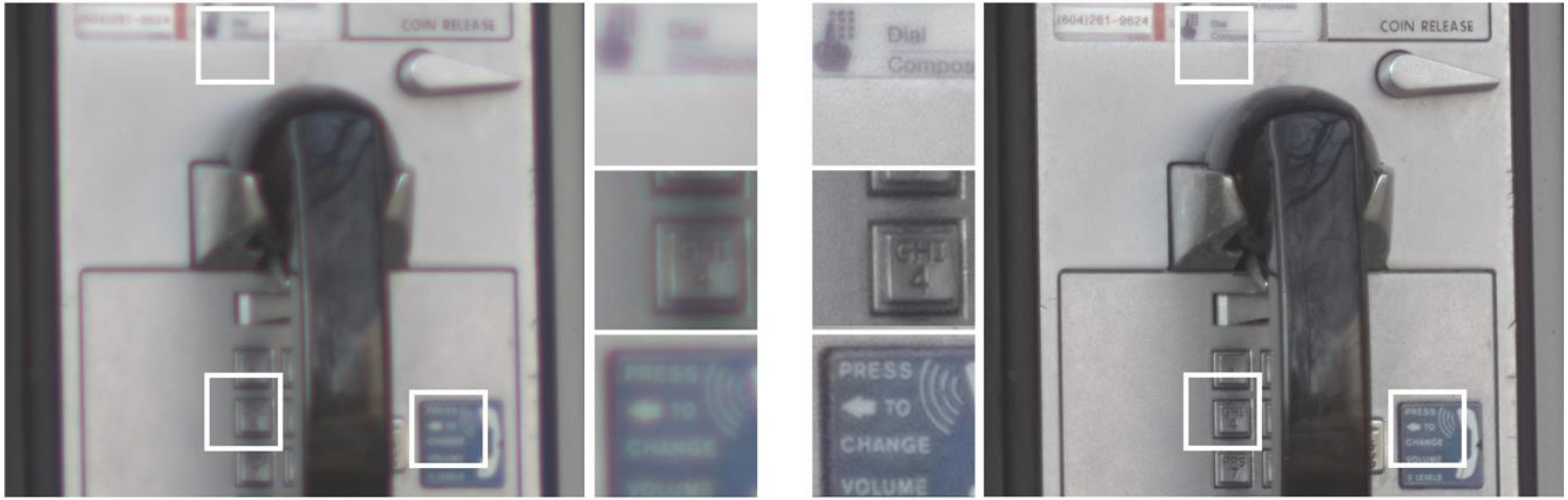
Comparison of gradient regularizations

input

squared gradient regularization

fractional gradient regularization

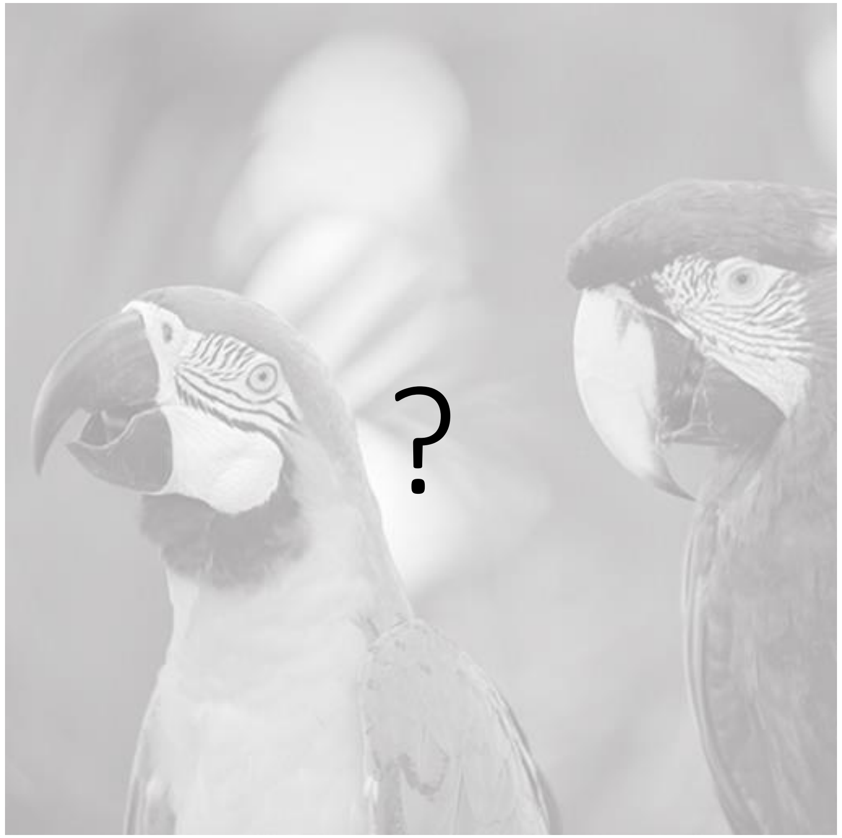
High quality images using cheap lenses



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

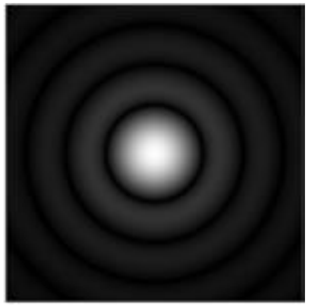
Deconvolution

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



*

How do we measure this?



=

$$x \quad * \quad c \quad = \quad b$$

PSF calibration

Take a photo of a point source

Image of PSF

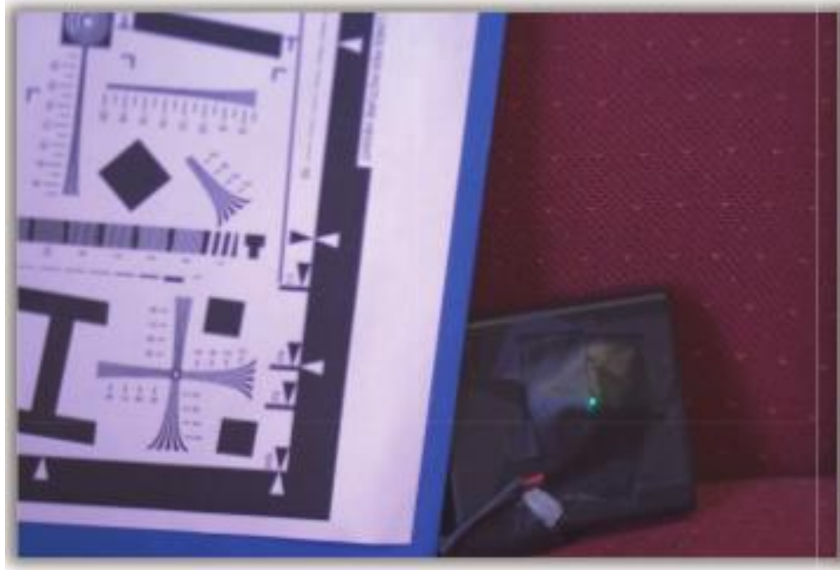
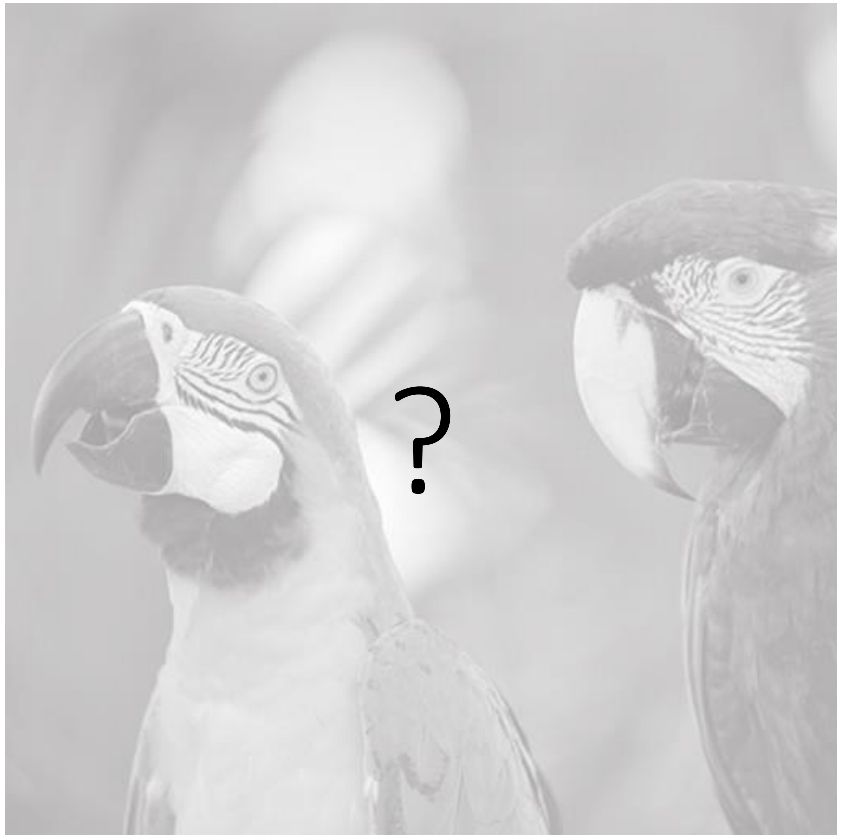


Image with sharp lens

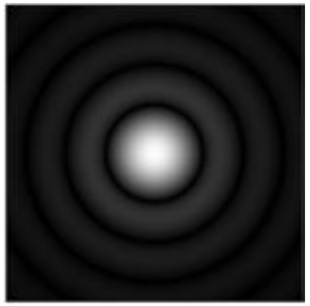
Image with cheap lens

Deconvolution

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



*

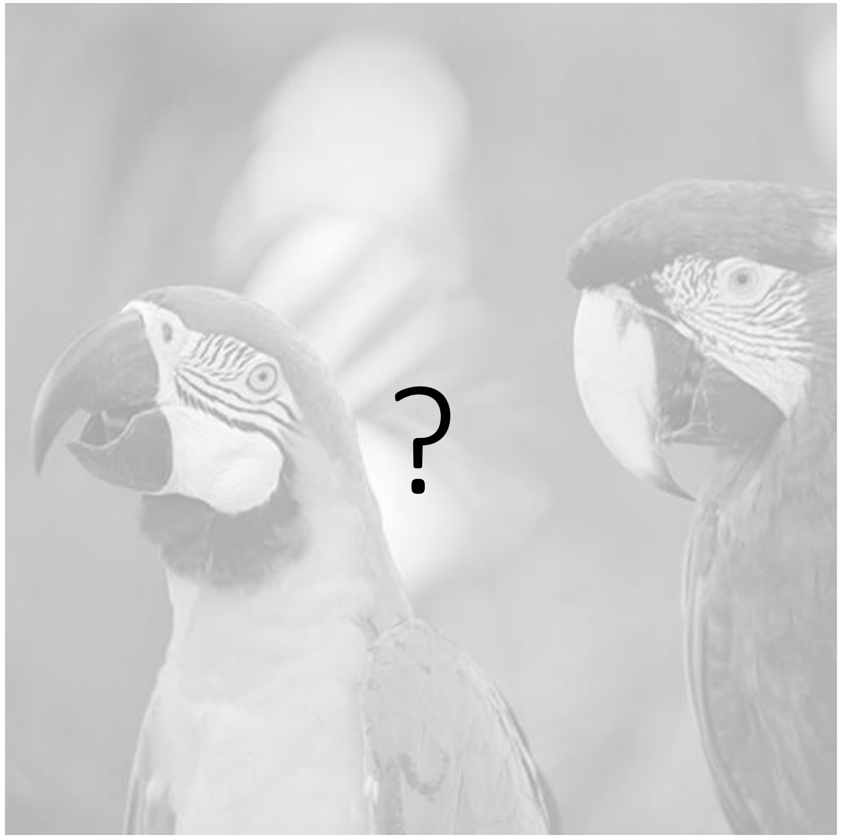


=

$$x * c = b$$

Blind deconvolution

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



$*$

$=$

$$x \quad * \quad c \quad = \quad b$$

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 [2]University of Toronto

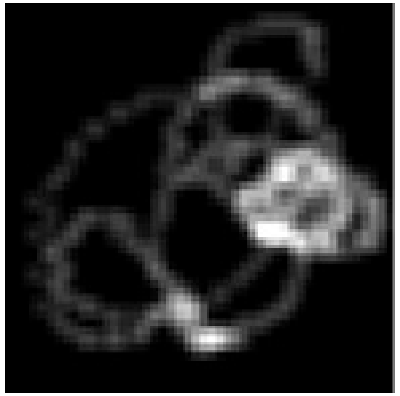
Figure 1: *Left*: An image spoiled by camera shake. *Middle*: result from Photoshop "unsharp mask". *Right*: result from our algorithm.

Camera shake as a filter

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

image from static camera

$*$



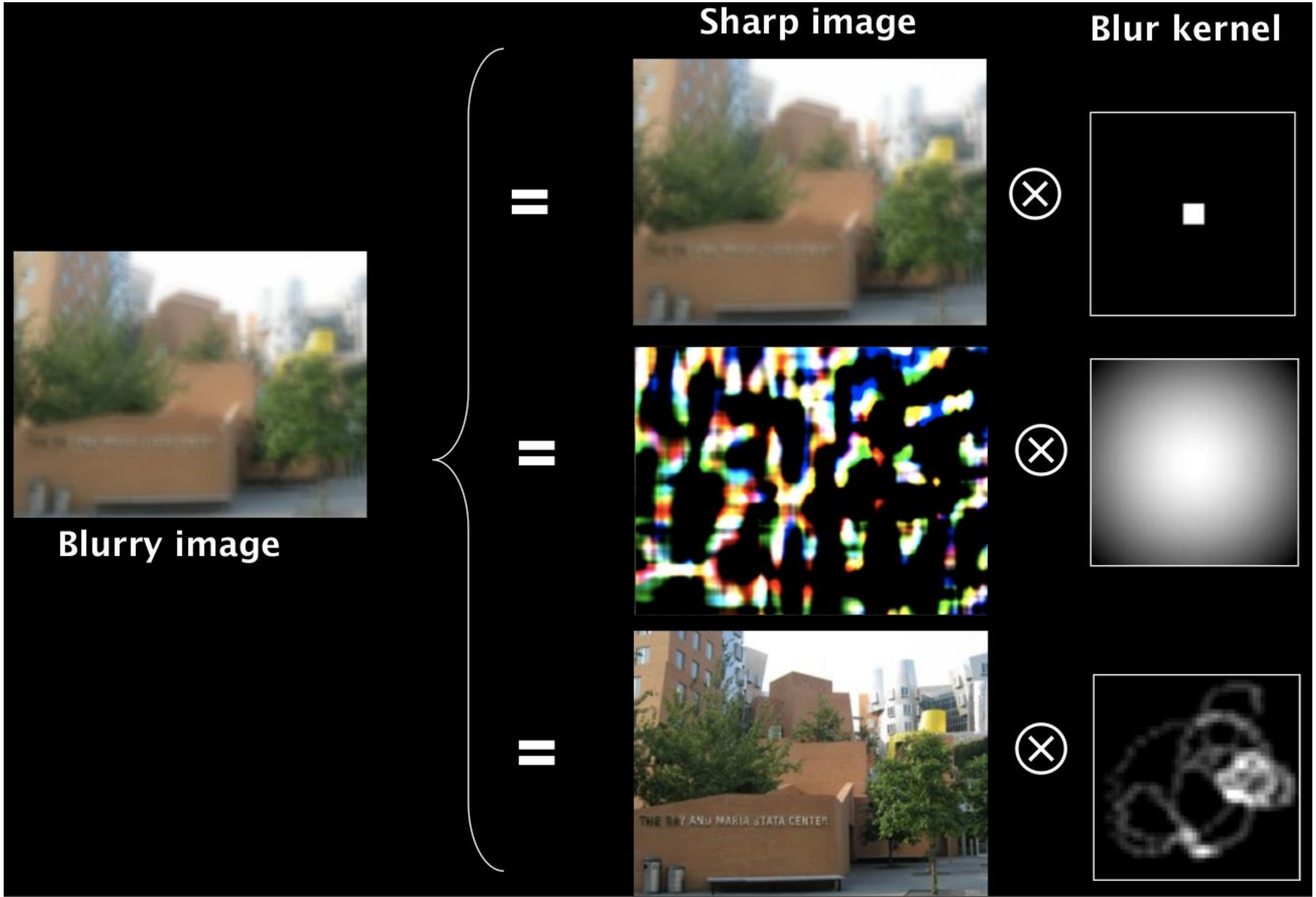
PSF from camera motion

$=$

image from shaky camera

$$x \quad * \quad c \quad = \quad b$$

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.

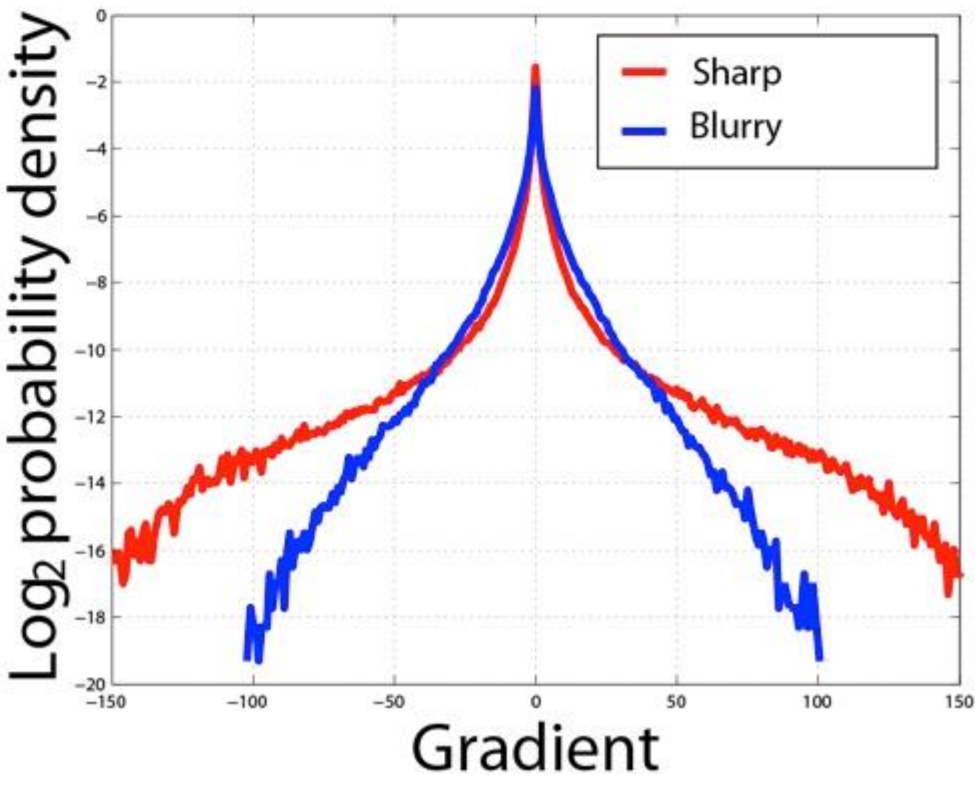
Use prior information

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.

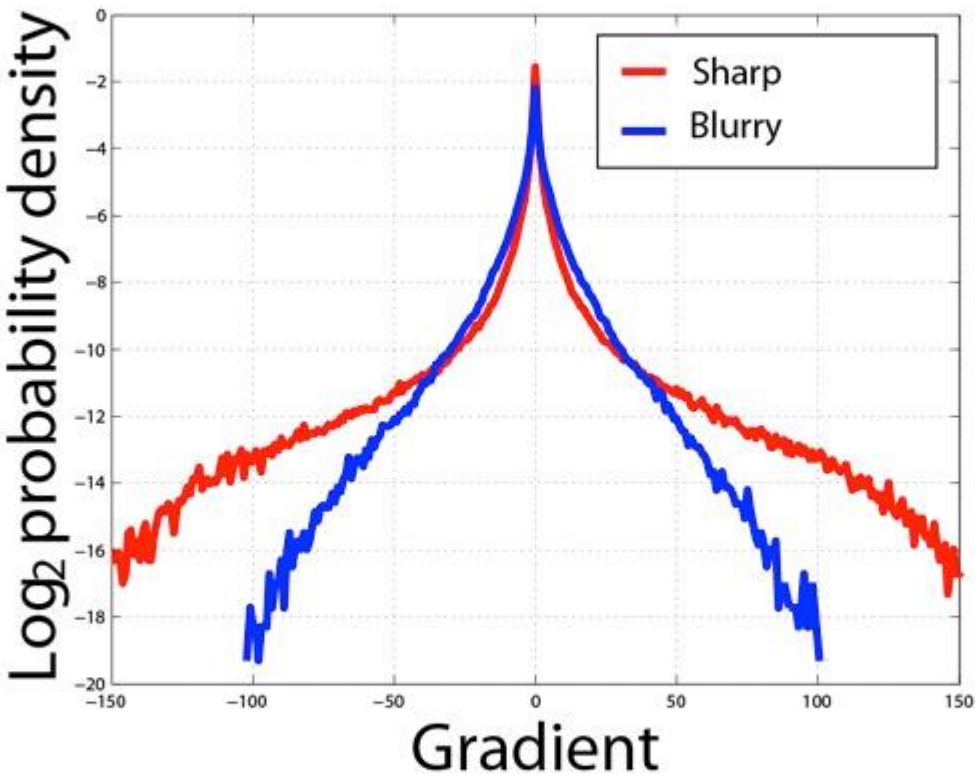


sharp natural image

blurry natural image

Shake kernel statistics

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



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

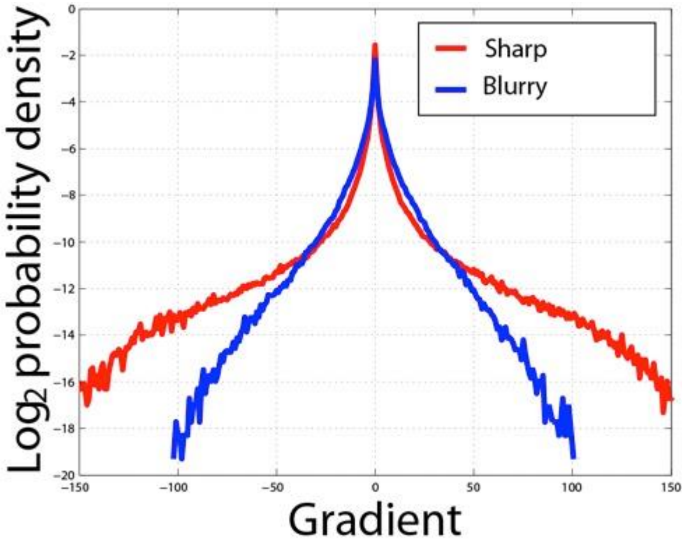
sharp natural image

blurry natural image

Use prior information

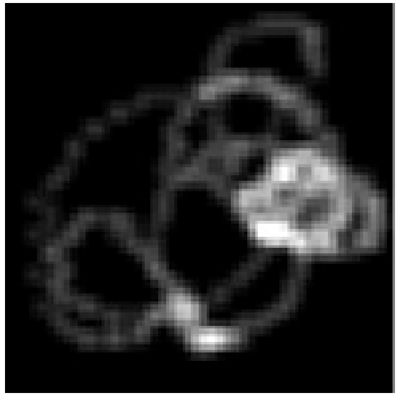
Among all the possible pairs of images and blur kernels, select the ones where:

- 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

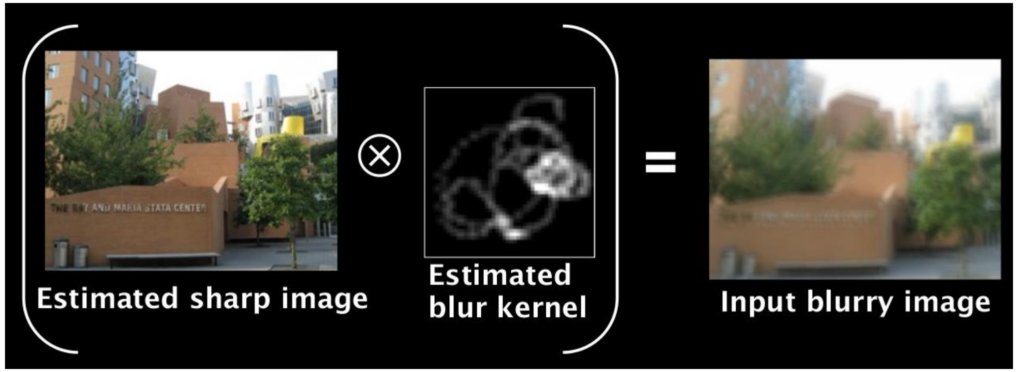
Solve regularized least-squares optimization

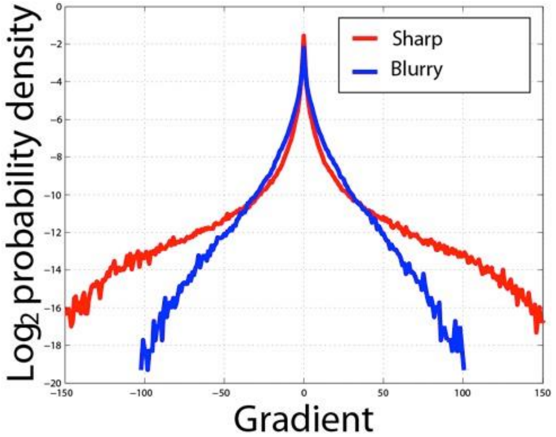
$$\min_{x,b} ||b - c * x||^2 + ||\nabla x||^{0.8} + ||c||_1$$

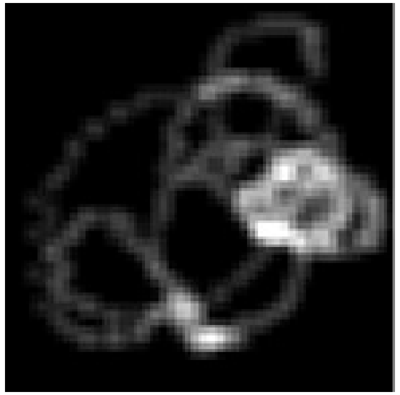
data term

natural image prior

shake kernel prior





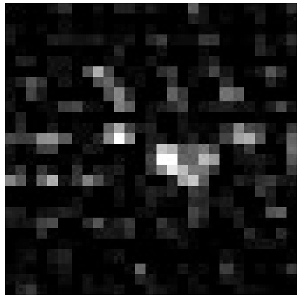


Note: Solving such optimization problems is complicated (no longer *linear* least squares).

A demonstration

input

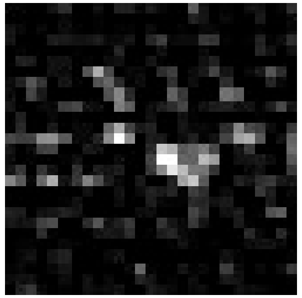
deconvolved image and kernel



A demonstration

input

deconvolved image and kernel



This image looks worse than the original…

This doesn't look like a plausible shake kernel…

Regularized blind deconvolution

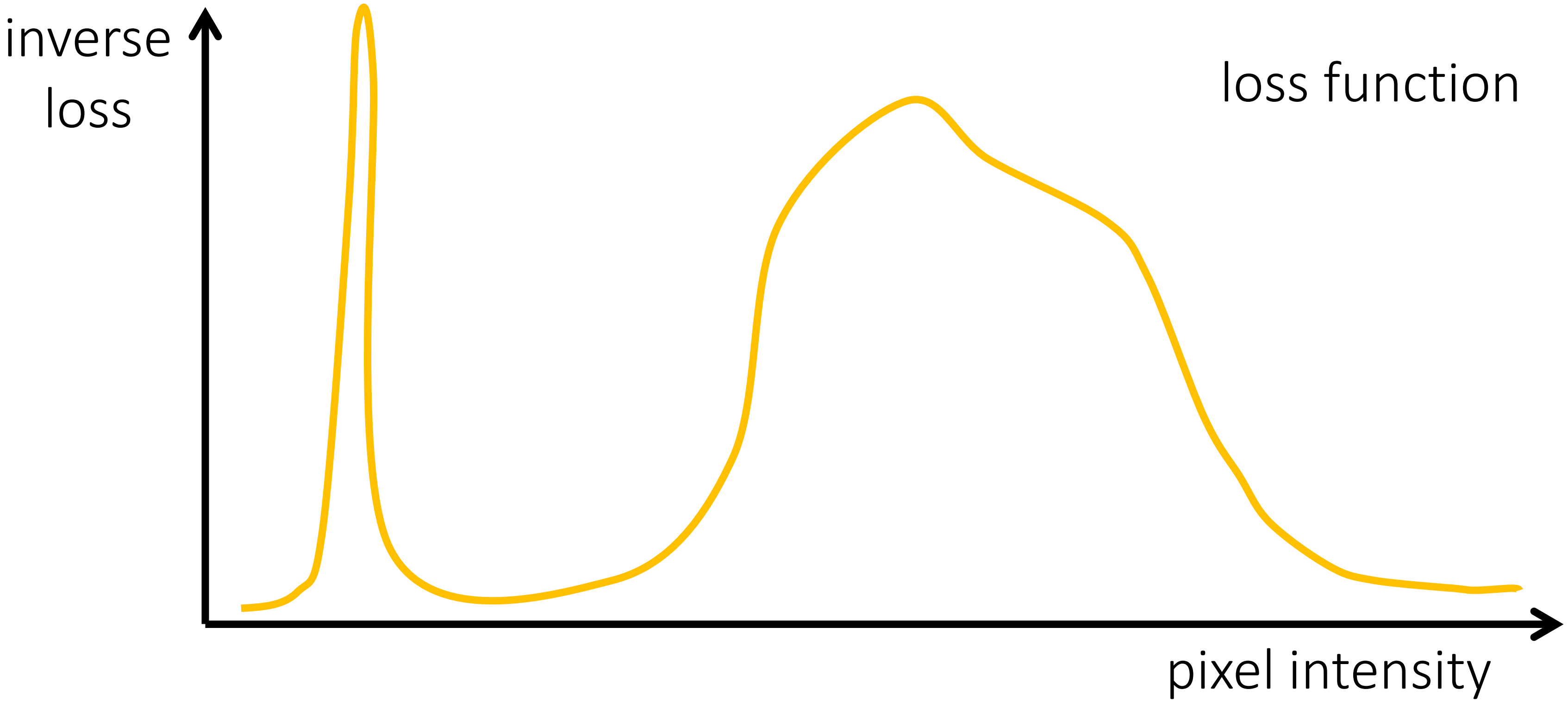
Solve regularized least-squares optimization

$$\min_{x,b} \underbrace{||b - c * x||^2 + ||\nabla x||^{0.8} + ||c||_1}_{\text{loss function}}$$

Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x,b} \underbrace{\|b - c * x\|^2 + \|\nabla x\|^{0.8} + \|c\|_1}_{\text{loss function}}$$

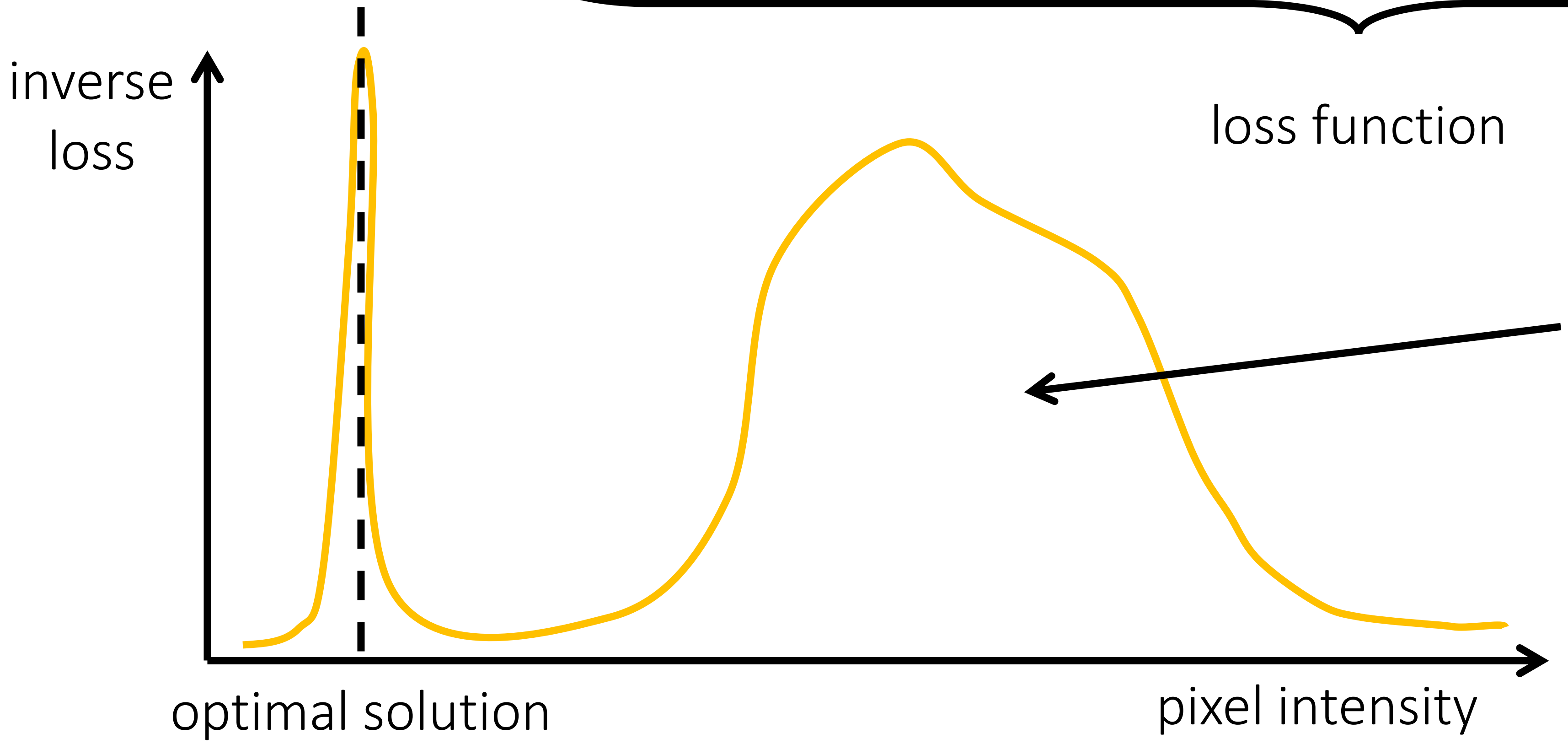


Where in this graph is the solution we find?

Regularized blind deconvolution

Solve regularized least-squares optimization

$$\min_{x,b} \underbrace{||b - c * x||^2 + ||\nabla x||^{0.8} + ||c||_1}_{\text{loss function}}$$



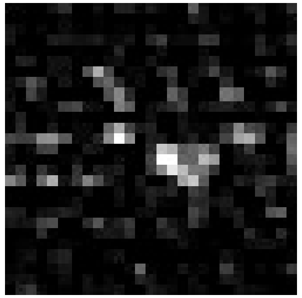
many plausible solutions here

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

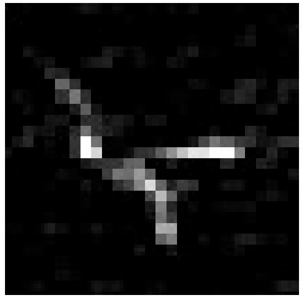
A demonstration

input

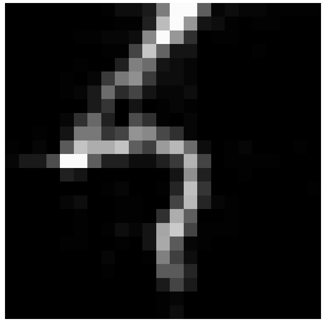
maximum-only

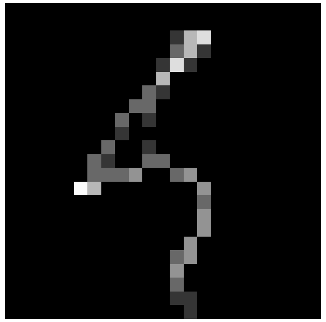


average

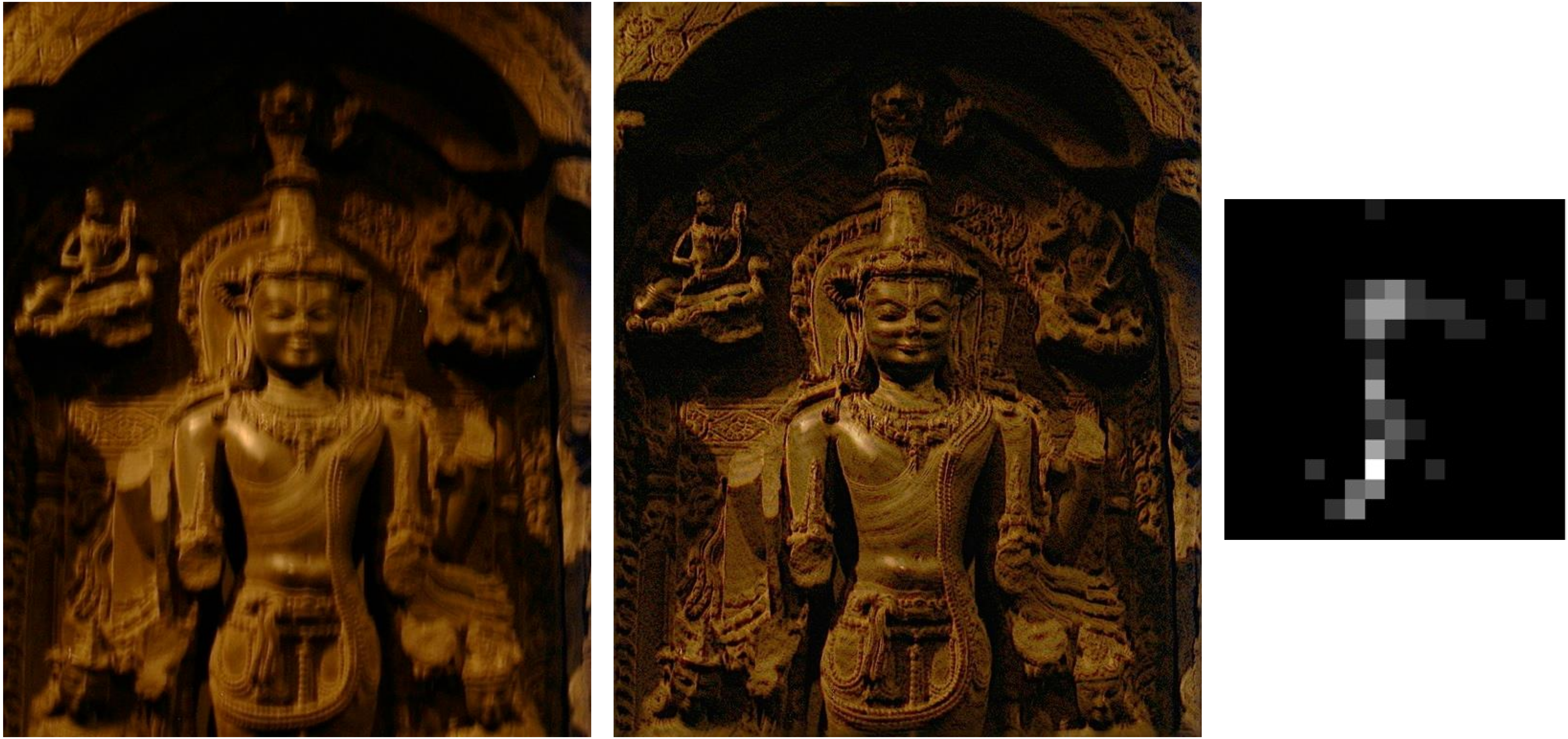


More examples



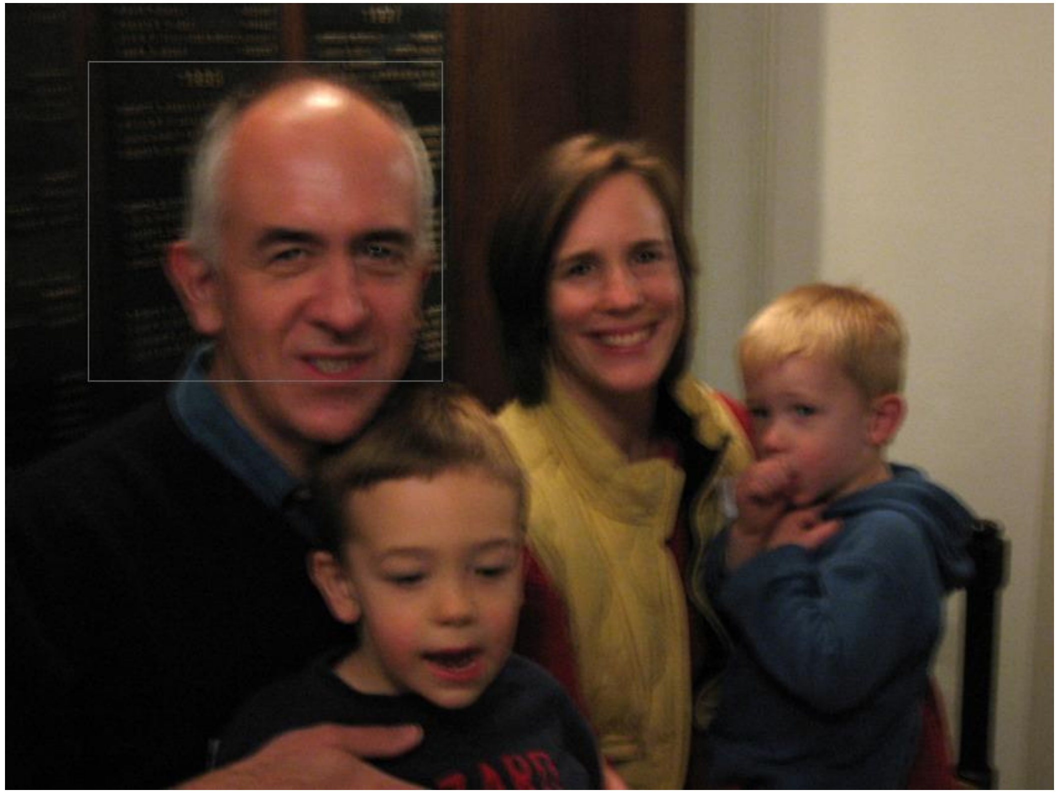


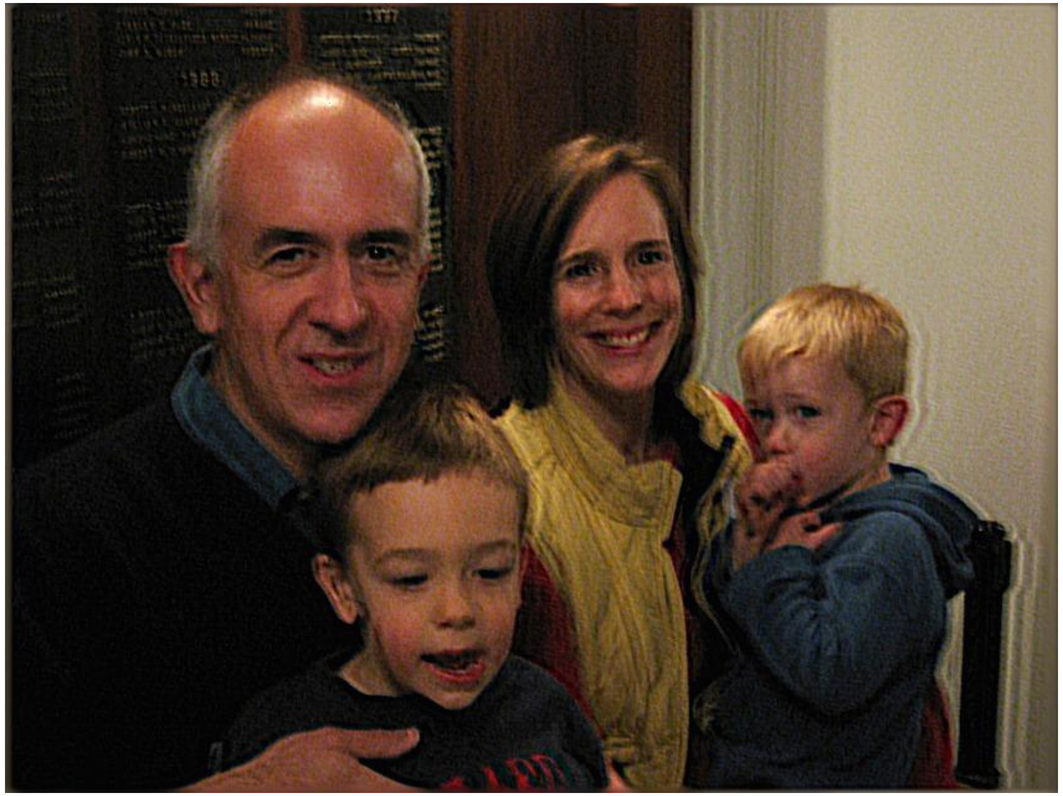
Results on real shaky images

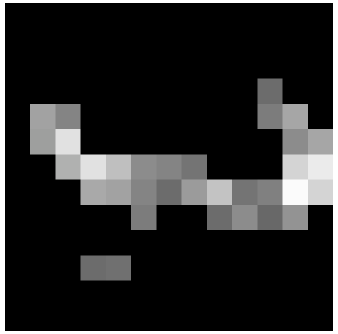


Results on real shaky images

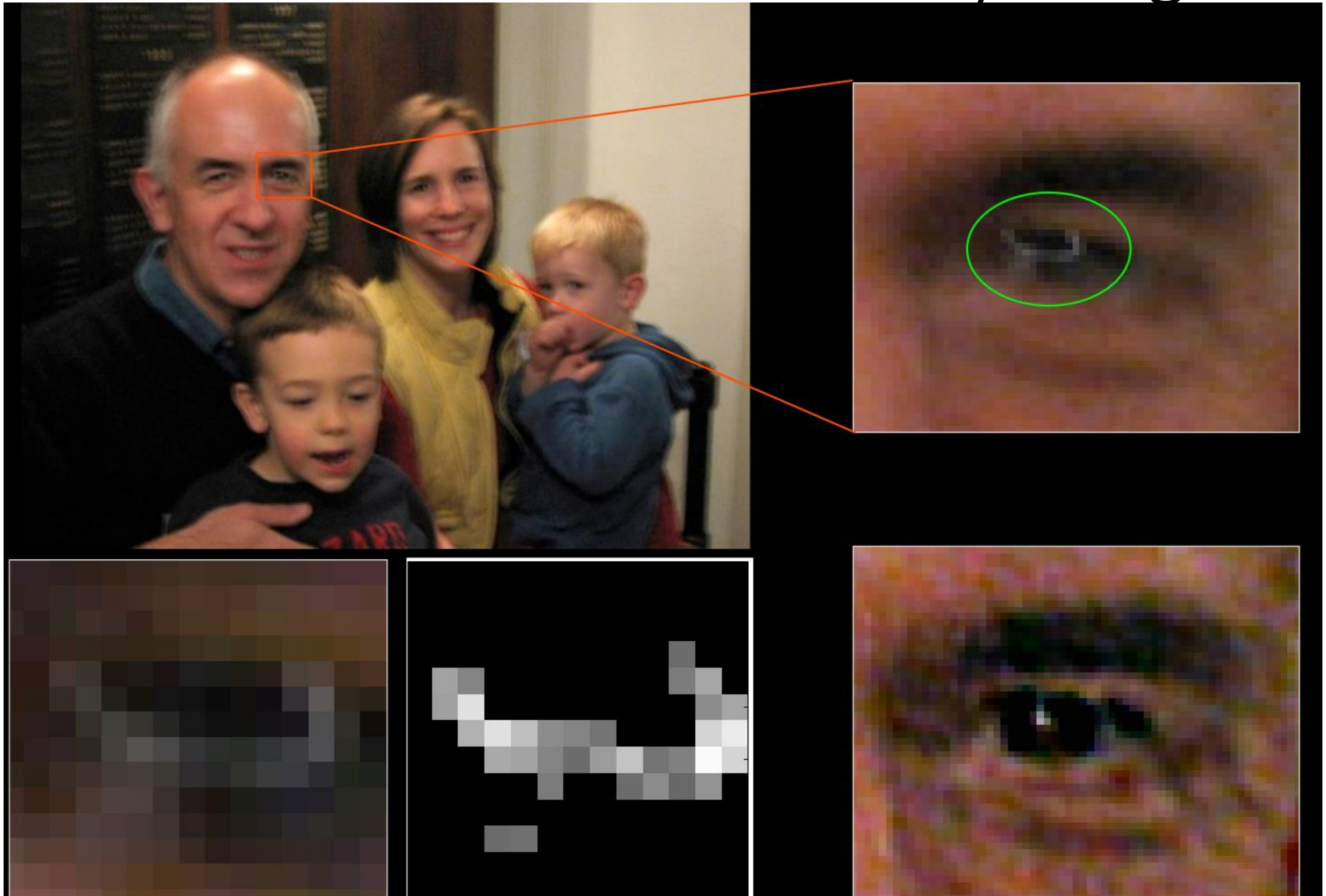
Results on real shaky images

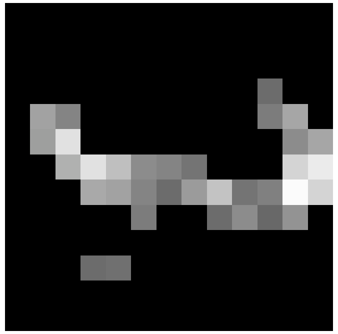






Results on real shaky images





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 in the same way?

Why are our images blurry?

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

Can we solve all of these problems in the same way?

- No, because blur is not always shift invariant.
- See next lecture.

References

Basic reading:

- Szeliski textbook, Sections 3.4.3, 3.4.4, 10.1.4, 10.3.
- Fergus et al., “Removing camera shake from a single image,” SIGGRAPH 2006.
  the main motion deblurring and blind deconvolution paper we covered in this lecture.

Additional reading:

- Heide et al., “High-Quality Computational Imaging Through Simple Lenses,” TOG 2013.
  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.
- Levin, “Blind Motion Deblurring Using Image Statistics,” NIPS 2006.
- Levin et al., “Image and depth from a conventional camera with a coded aperture,” SIGGRAPH 2007.
- Levin et al., “Understanding and evaluating blind deconvolution algorithms,” CVPR 2009 and PAMI 2011.
- Krishnan and Fergus, “Fast Image Deconvolution using Hyper-Laplacian Priors,” NIPS 2009.
- Levin et al., “Efficient Marginal Likelihood Optimization in Blind Deconvolution,” CVPR 2011.
  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.
- Minskin and MacKay, “Ensemble Learning for Blind Image Separation and Deconvolution,” AICA 2000.
  the paper explaining the mathematics of how to compute Bayesian estimators using variational inference.
- Shah et al., “High-quality Motion Deblurring from a Single Image,” SIGGRAPH 2008.
  a more recent paper on motion deblurring.