

Removing Camera Shake from a Single Photograph

Rob Fergus, Barun Singh, Aaron Hertzmann,
Sam T. Roweis and William T. Freeman

Massachusetts Institute of Technology
and
University of Toronto

Overview

Joint work with B. Singh, A. Hertzmann, S.T. Roweis & W.T. Freeman

Original

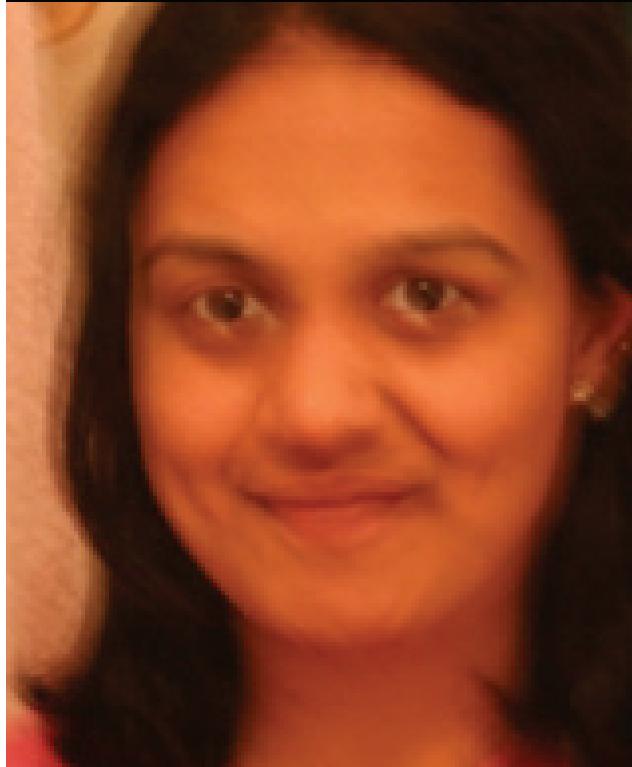


Our algorithm

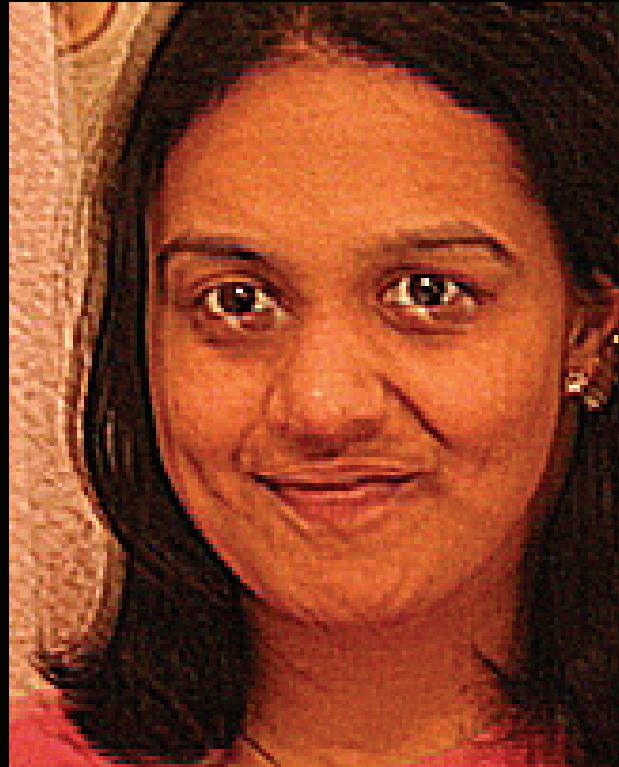


Close-up

Original



Naïve sharpening



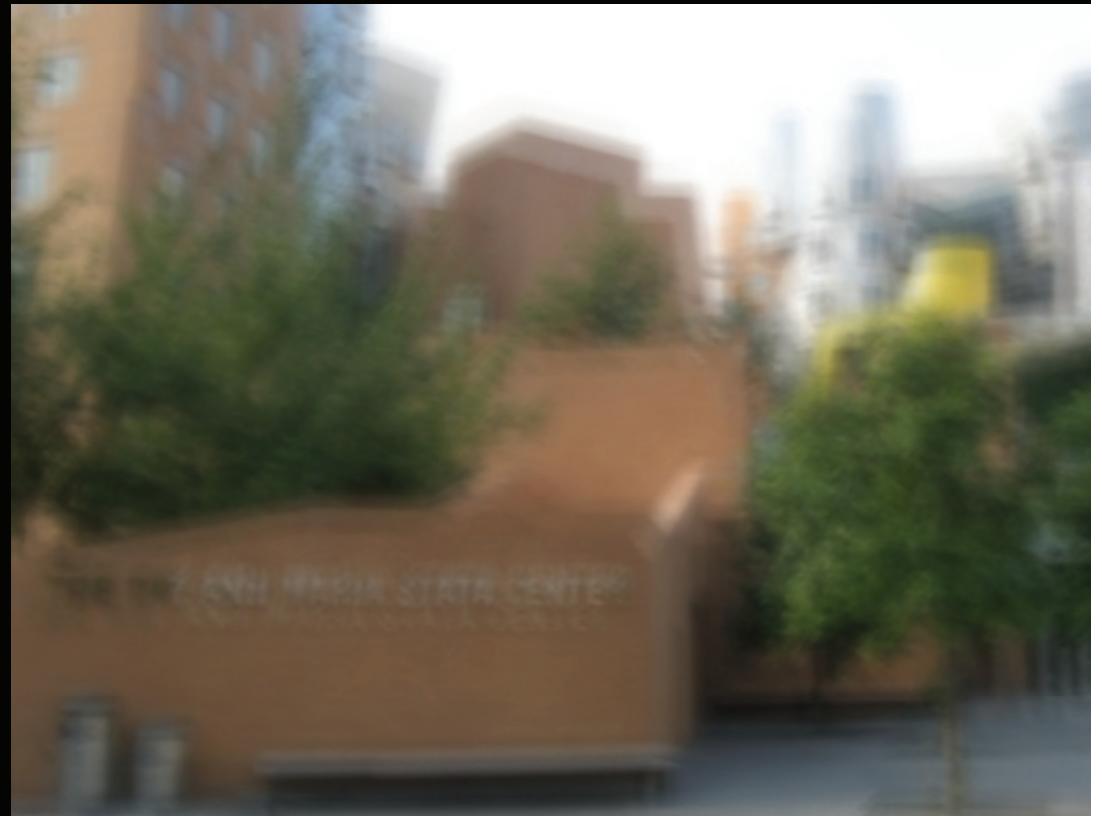
Our algorithm



Let's take a photo



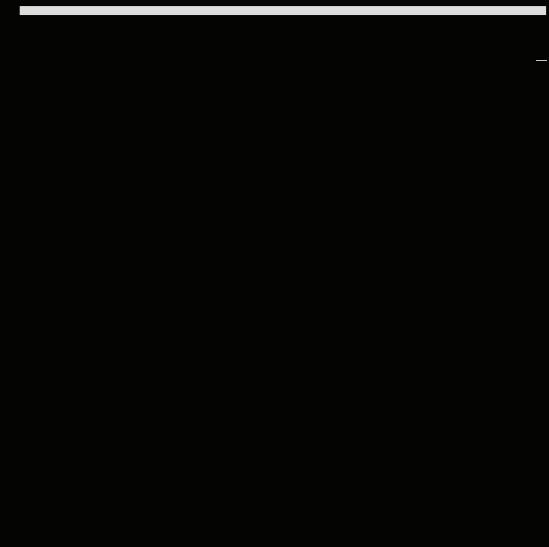
Blurry result



Slow-motion replay

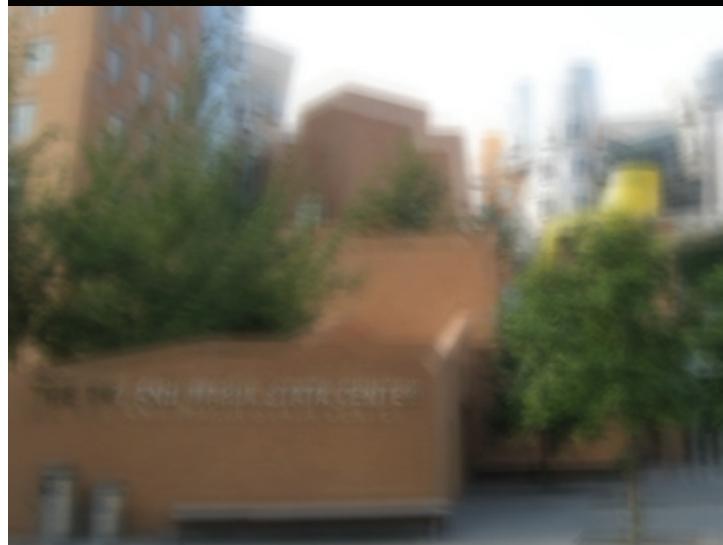


Slow-motion replay



Motion of camera

Image formation process



Blurry image

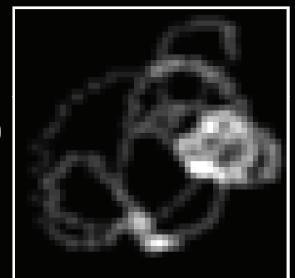
Input to algorithm

Model is approximation
Assume static scene



Sharp image

Desired output



Blur kernel

Convolution
operator

Existing work on image deblurring

Old problem:

- Trott, T., “The Effect of Motion of Resolution”, *Photogrammetric Engineering*, Vol. 26, pp. 819-827, 1960.
- Slepian, D., “Restoration of Photographs Blurred by Image Motion”, *Bell System Tech.*, Vol. 46, No. 10, pp. 2353-2362, 1967.

The screenshot shows a Google Scholar search results page. The search query "camera shake" is entered in the search bar. The results section displays 11,600 items, with the first few entries being patent documents related to camera shake prevention and correction. A red circle highlights the search count "Results 1 - 10 of about 11,600 for camera shake. (0.07 seconds)".

Google Scholar BETA

camera shake

Search Advanced Scholar Search Scholar Preferences Scholar Help

Scholar All articles Recent articles Results 1 - 10 of about 11,600 for camera shake. (0.07 seconds)

All Results

T Teramoto

S Enomoto

D Gray

M Hamada

A Katayama

[Camera capable of correcting camera-shake](#) - group of 2 »
H Ootsuka, T Okada, H Masumoto, M Hamada - US Patent 5,561,485, 1996 - patentstorm.us
Camera capable of correcting camera-shake - US Patent 5561485 from Patent Storm.
A camera comprises an angular velocity sensor for detecting camera-shake. ...
Cited by 26 - Related Articles - Cached - Web Search

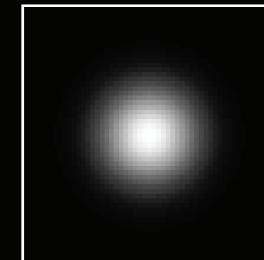
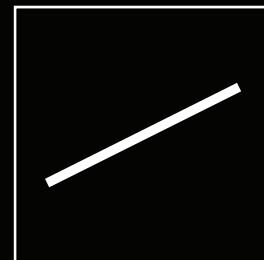
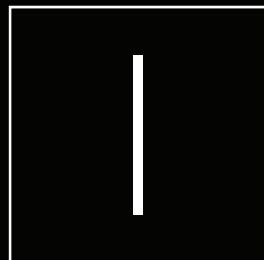
[Camera-shake preventing device](#) - group of 2 »
K Imafuji, N Terui - US Patent 5,337,098, 1994 - Google Patents
... when it is detected that said battery has been consumed beyond a predetermined amount, said control means starts compensation of the camera shake in response ...
Cited by 22 - Related Articles - Web Search

[Camera shake correction system](#) - group of 4 »
A Misawa, K Ikari, S Ueda... - US Patent 5,041,852, 1991 - Google Patents
... Misawa et al. [ii] Patent Number: [45] Date of Patent: [54] CAMERA SHAKE CORRECTION ...
FIG. 27 PRIOR ART 7B Page 23. 5,041,852 CAMERA SHAKE CORRECTION SYSTEM ...

Existing work on image deblurring

Software algorithms for natural images

- Many require multiple images
- Mainly Fourier and/or Wavelet based
- Strong assumptions about blur
 - not true for camera shake



Assumed forms of blur kernels

- Image constraints are frequency-domain power-laws

Existing work on image deblurring

Hardware approaches

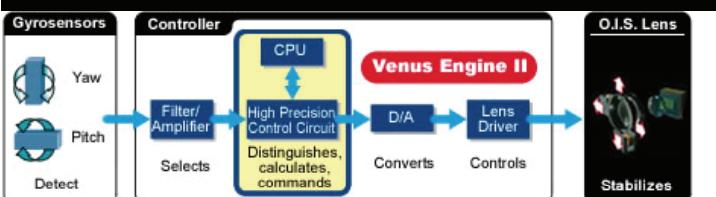
Image stabilizers



Dual cameras



Coded shutter



Ben-Ezra & Nayar
CVPR 2004

Raskar et al.
SIGGRAPH 2006

Our approach can be combined with these hardware methods

Why is this hard?

Simple analogy:

11 is the product of two numbers.

What are they?

No unique solution:

$$11 = 1 \times 11$$

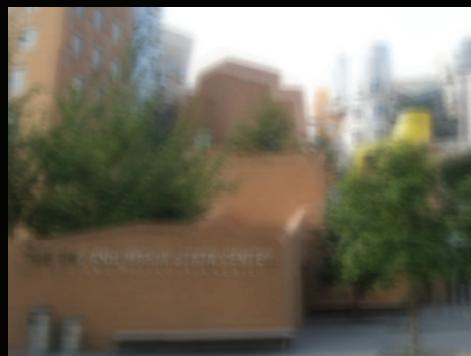
$$11 = 2 \times 5.5$$

$$11 = 3 \times 3.667$$

etc.....

Need more information !!!

Multiple possible solutions



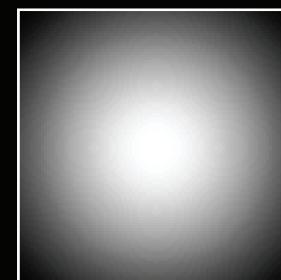
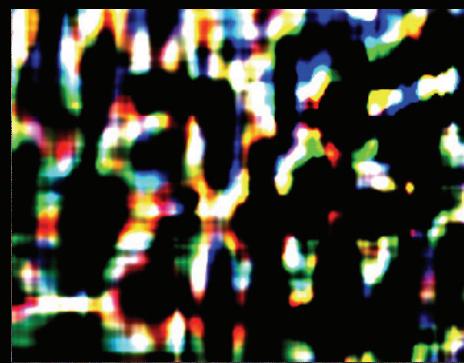
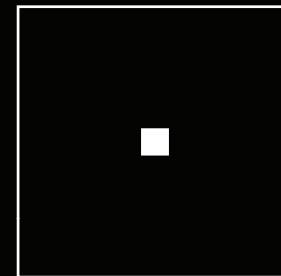
Blurry image

$$\left. \begin{array}{c} = \\ = \\ - \end{array} \right\}$$

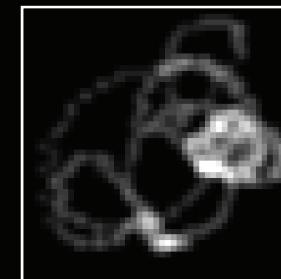
Sharp image



Blur kernel



$$-$$

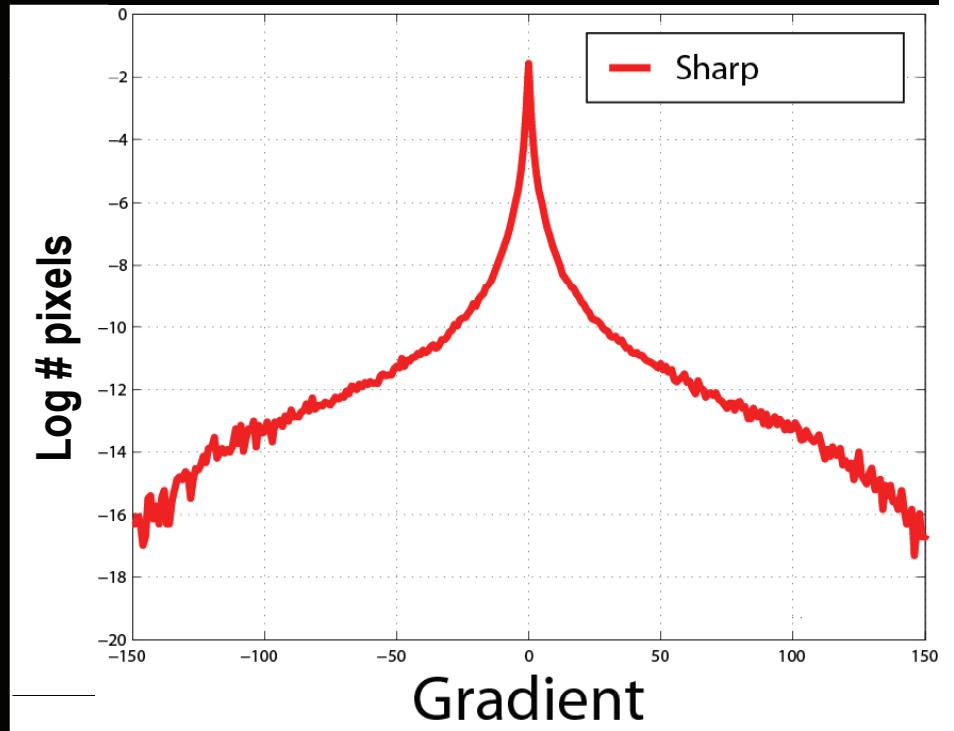


$$-$$

Natural image statistics

Characteristic distribution with heavy tails

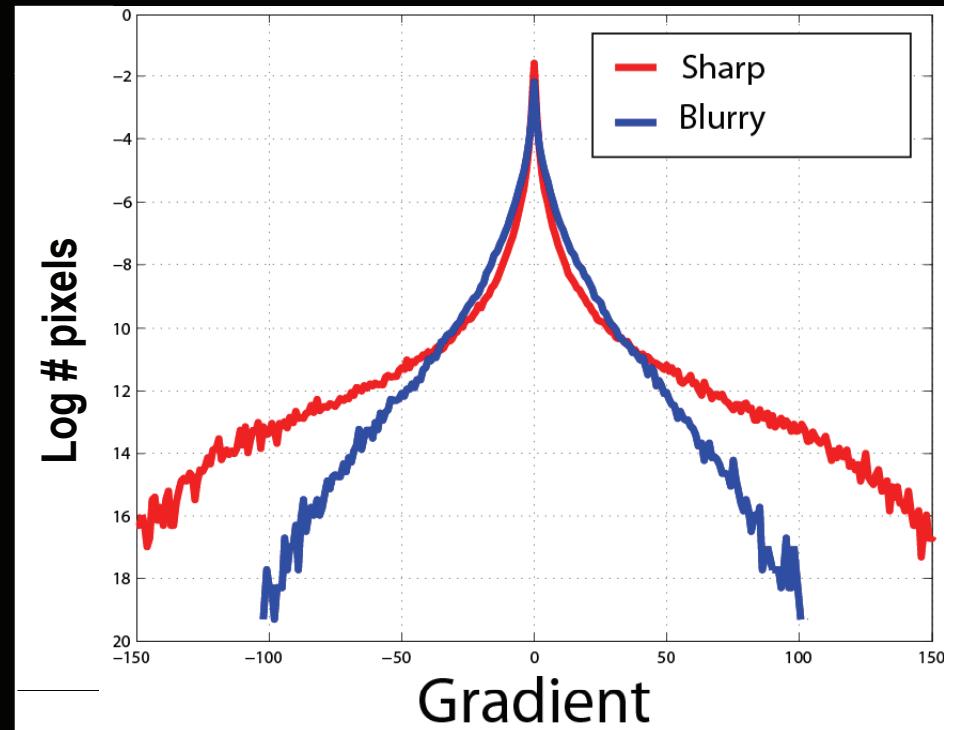
Histogram of image gradients



Blurry images have different statistics



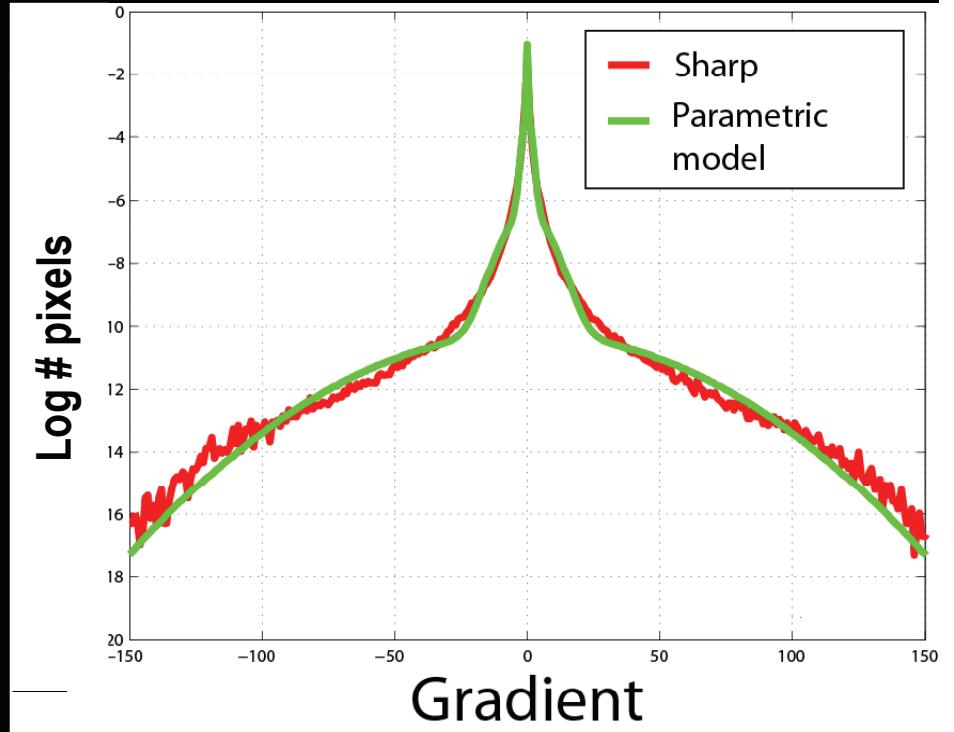
Histogram of image gradients



Parametric distribution



Histogram of image gradients



Use parametric model of sharp image statistics

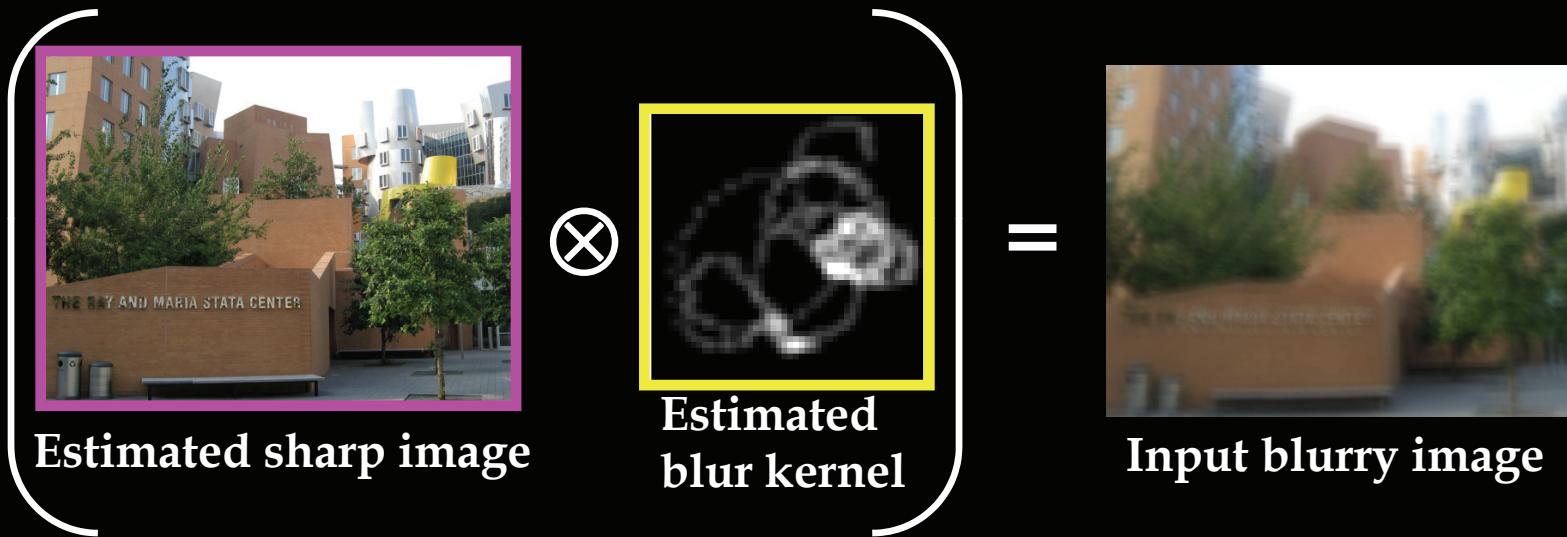
Uses of natural image statistics

- Denoising [Portilla et al. 2003, Roth and Black, CVPR 2005]
- Superresolution [Tappen et al., ICCV 2003]
- Intrinsic images [Weiss, ICCV 2001]
- Inpainting [Levin et al., ICCV 2003]
- Reflections [Levin and Weiss, ECCV 2004]
- Video matting [Apostoloff & Fitzgibbon, CVPR 2005]

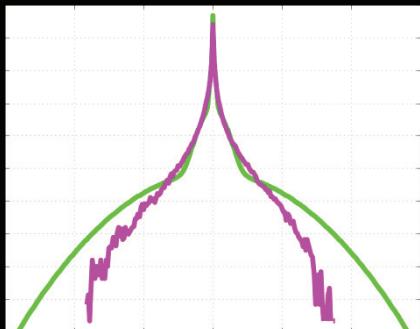
Corruption process assumed known

Three sources of information

1. Reconstruction constraint:



2. Image prior:



Distribution
of gradients

3. Blur prior:



Positive
&
Sparse

Three sources of information

y = observed image

b = blur kernel

x = sharp image

Three sources of information

y = observed image

b = blur kernel

x = sharp image

$$p(b, x|y)$$

Posterior

Three sources of information

y = observed image

b = blur kernel

x = sharp image

$$p(b, x|y) = k \quad p(y|b, x) \quad p(x) \quad p(b)$$

Posterior 1. Likelihood
 (Reconstruction
 constraint)
 2. Image prior 3. Blur prior

1. Likelihood $p(y|b, x)$

y = observed image

b = blur

x = sharp image

Reconstruction constraint:

$$p(y|b, x) = \prod_i \mathcal{N}(y_i | x_i \otimes b, \sigma^2)$$
$$\propto \prod_i e^{-\frac{(x_i \otimes b - y_i)^2}{2\sigma^2}}$$

i - pixel index

2. Image prior $p(x)$

y = observed image

b = blur

x = sharp image

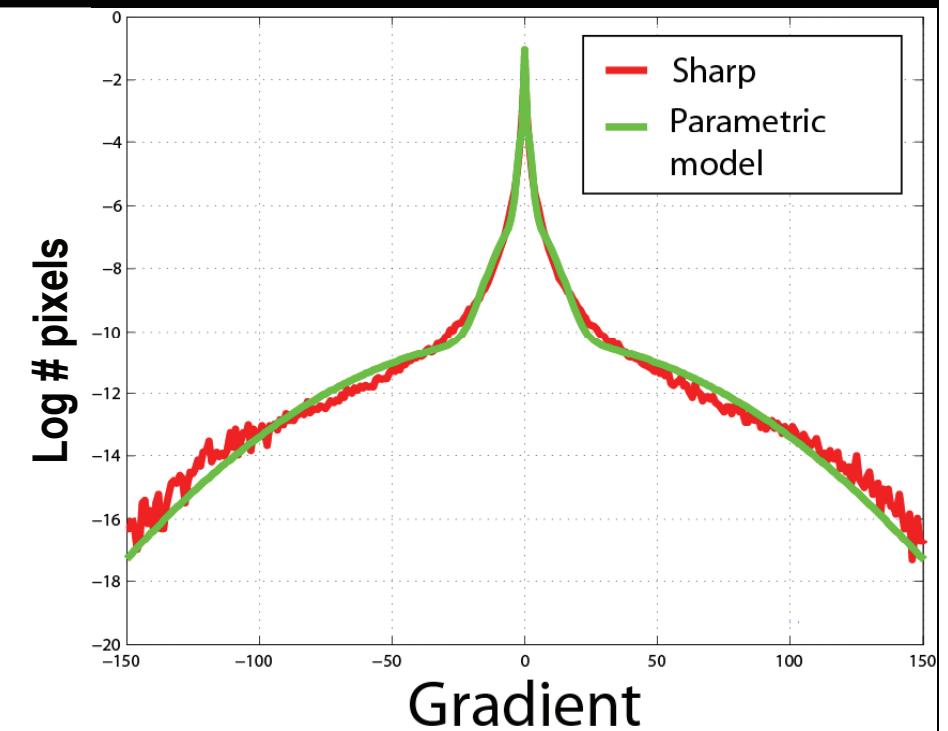
$$p(x) = \prod_i \sum_{c=1}^C \pi_c \mathcal{N}(f(x_i) | 0, s_c^2)$$

Mixture of Gaussians fit to empirical distribution of image gradients

i - pixel index

c - mixture component index

f - derivative filter



3. Blur prior $p(b)$

y = observed image

b = blur

x = sharp image

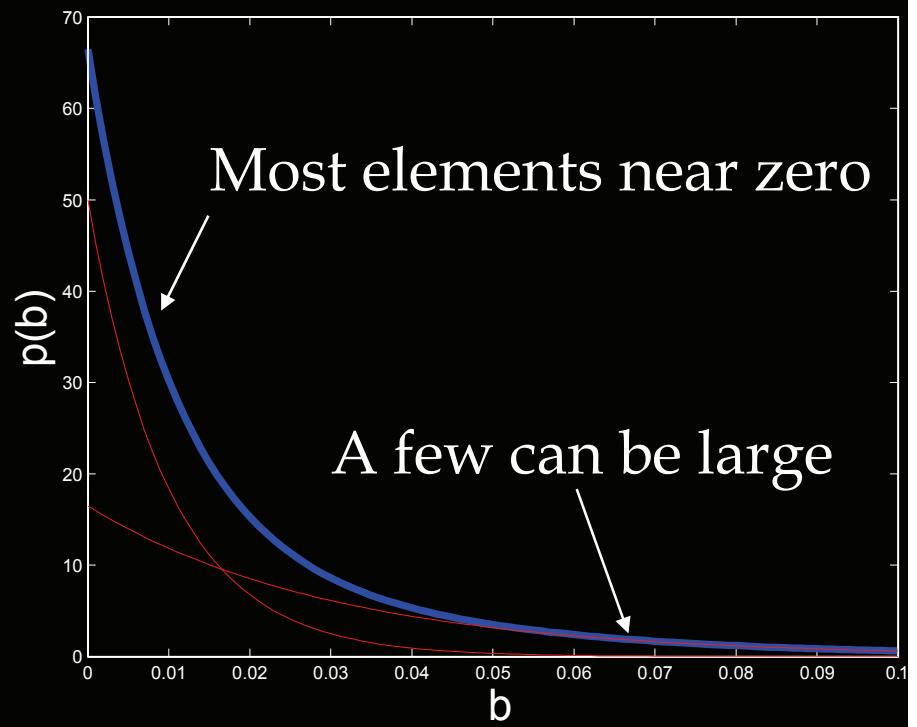
$$p(b) = \prod_j \sum_{d=1}^D \pi_d \mathcal{E}(b_j | \lambda_d)$$

Mixture of Exponentials

- Positive & sparse
- No connectivity constraint

j - blur kernel element

d - mixture component index



The obvious thing to do

$$p(b, x|y) = k \ p(y|b, x) \ p(x) \ p(b)$$

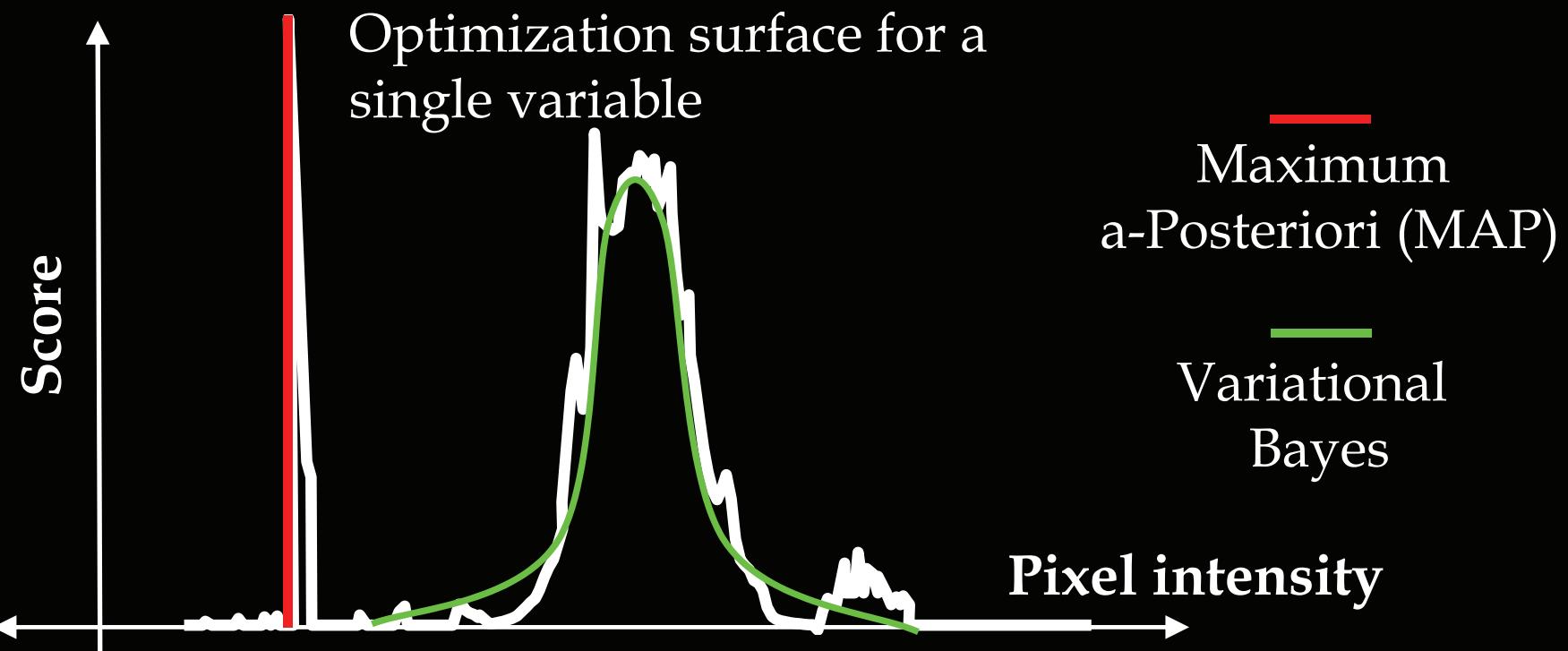
Posterior 1. Likelihood
 (Reconstruction
 constraint) 2. Image prior 3. Blur prior

- Combine 3 terms into an objective function
- Run conjugate gradient descent
- This is Maximum a-Posteriori (MAP)

No success!

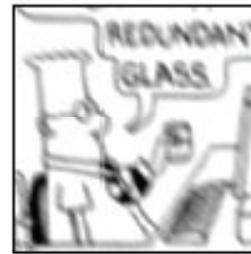
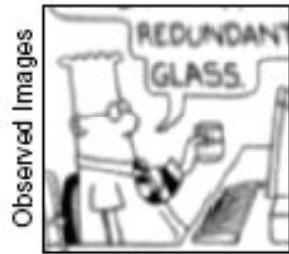
Variational Bayesian approach

Keeps track of uncertainty in estimates of image and blur by using a distribution instead of a single estimate

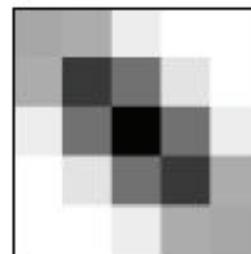
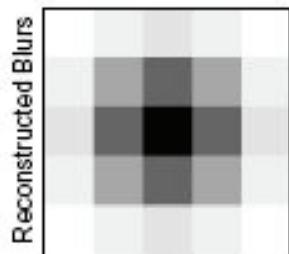


Variational Independent Component Analysis

Miskin and Mackay, 2000



- Binary images
- Priors on intensities
- Small, synthetic blurs



- Not applicable to natural images



Fig. 9. Demonstration of the deconvolution of two blurred images. In each test the same image was blurred by a different filter. The reconstructed filters match the true filters. The reconstructed images are close to the hidden images. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

Setup of Variational Approach

Work in gradient domain:

$$x \otimes b = y \rightarrow \nabla x \otimes b = \nabla y$$

Approximate posterior $p(\nabla x, b | \nabla y)$
with $q(\nabla x, b)$

Assume $q(\nabla x, b) = q(\nabla x)q(b)$

$q(\nabla x)$ is Gaussian on each pixel

$q(b)$ is rectified Gaussian on each blur kernel element

Cost function $KL(q(\nabla x)q(b) || p(\nabla x, b | \nabla y))$

Overview of algorithm

1. Pre-processing

2. Kernel estimation

- Multi-scale approach

3. Image reconstruction

- Standard non-blind deconvolution routine

Input image



Preprocessing

Input image



Convert to
grayscale

Remove gamma
correction

User selects patch
from image

Bayesian inference
too slow to run on
whole image

Infer kernel
from this patch



Initialization

Input image



Convert to
grayscale

Remove gamma
correction

User selects patch
from image

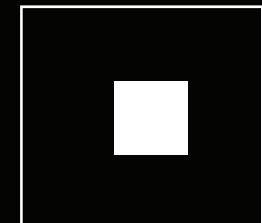
Initialize 3x3
blur kernel



Blurry patch

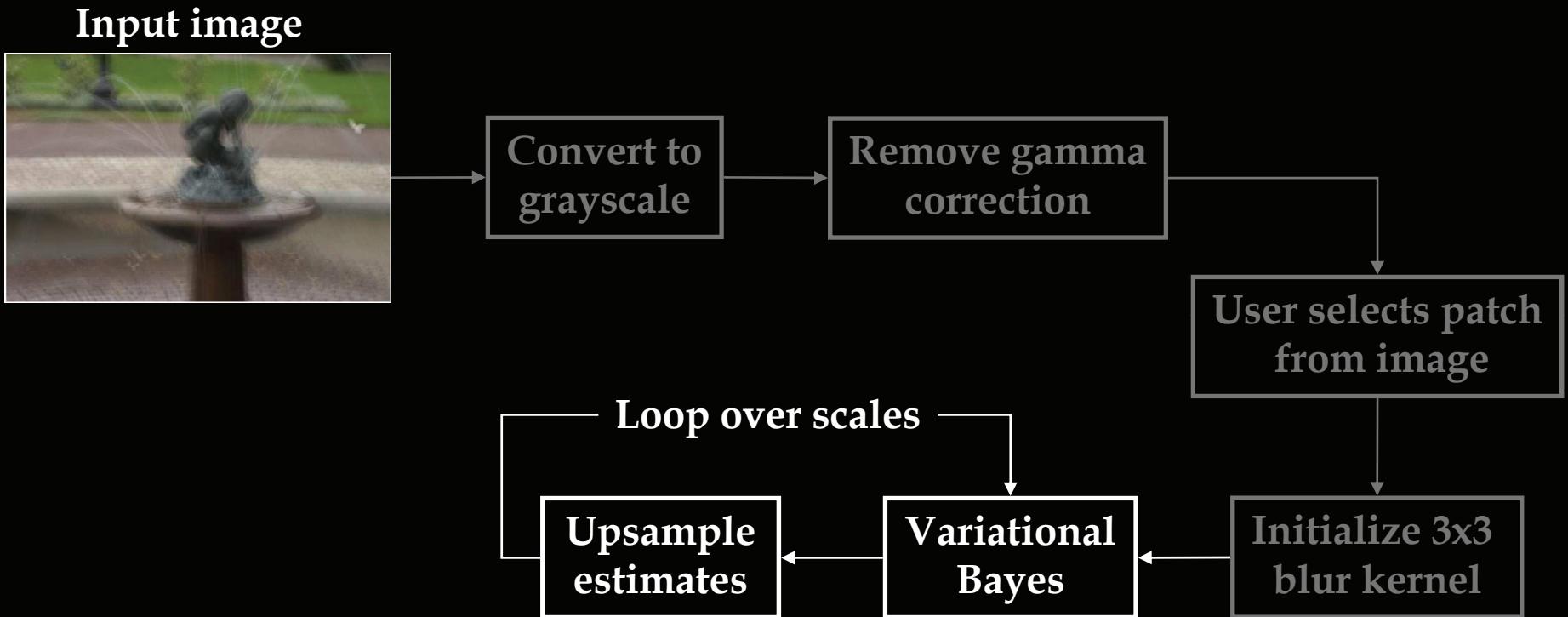


Initial image estimate



Initial blur kernel

Inferring the kernel: multiscale method



Use multi-scale approach to avoid local minima:

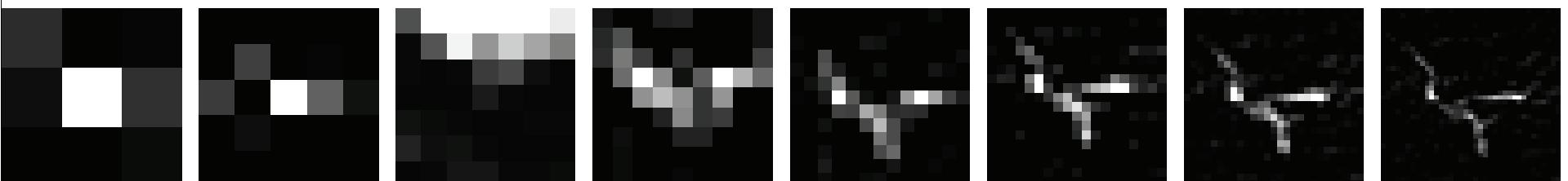
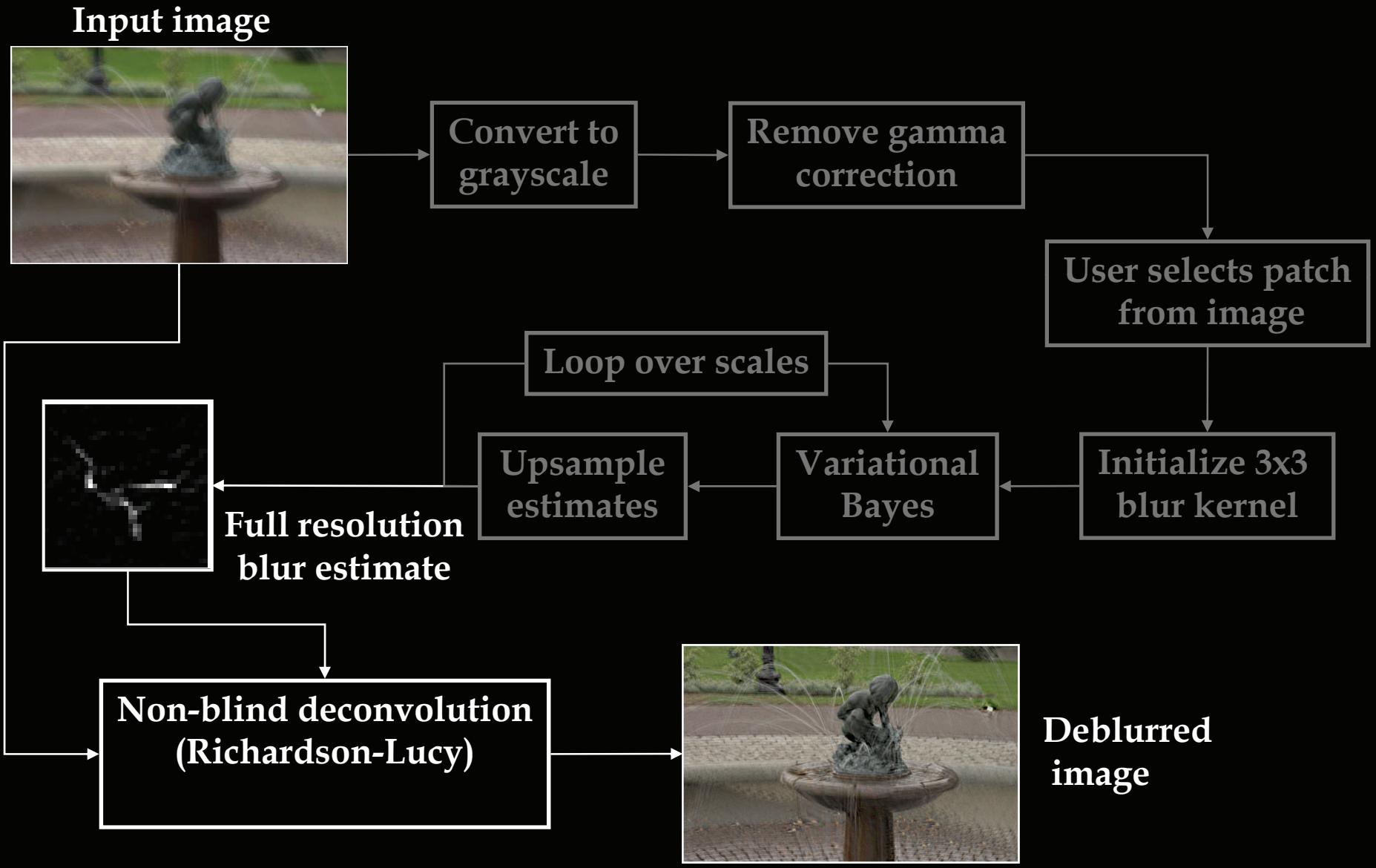


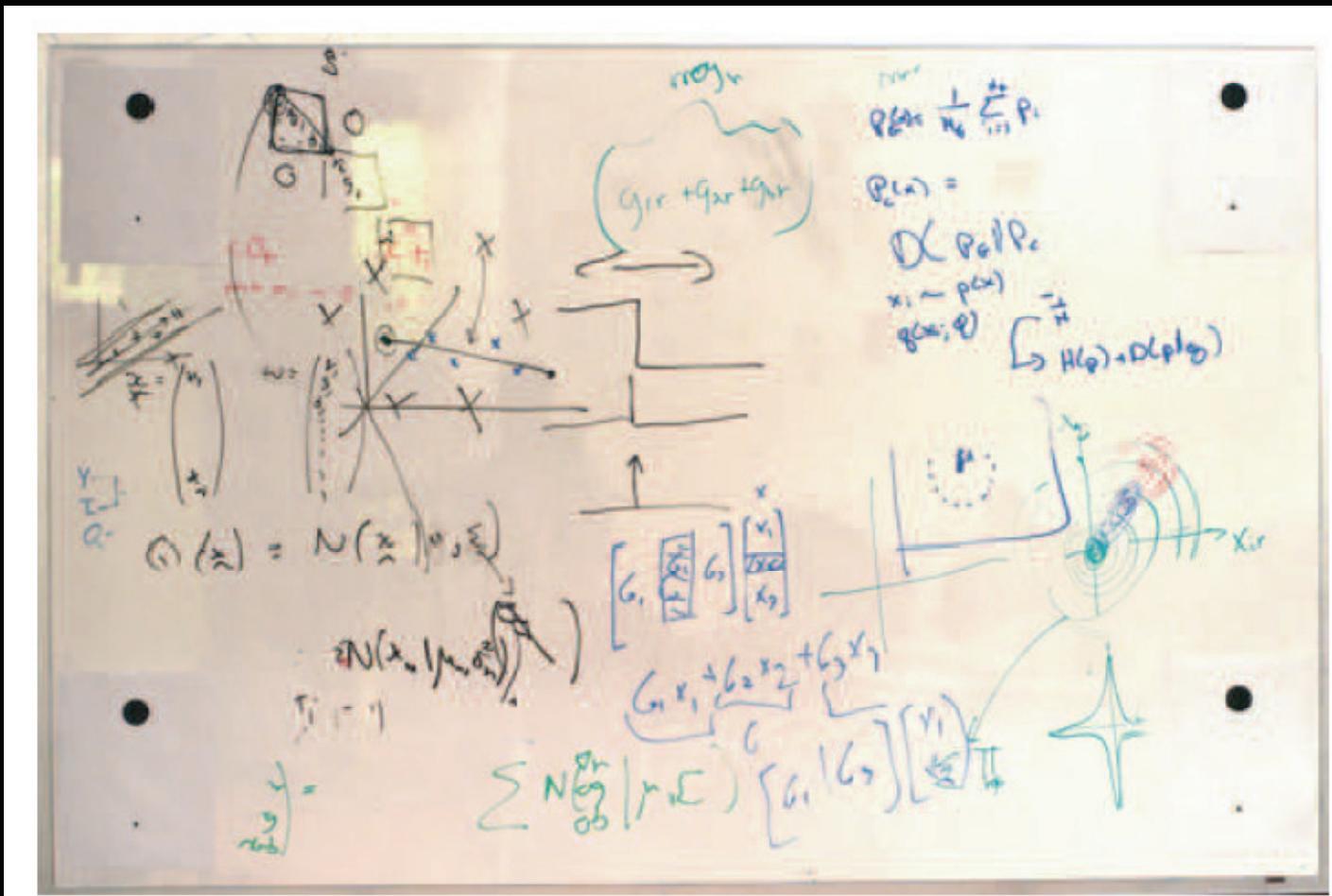
Image Reconstruction



Synthetic experiments

Is blur kernel really stationary?

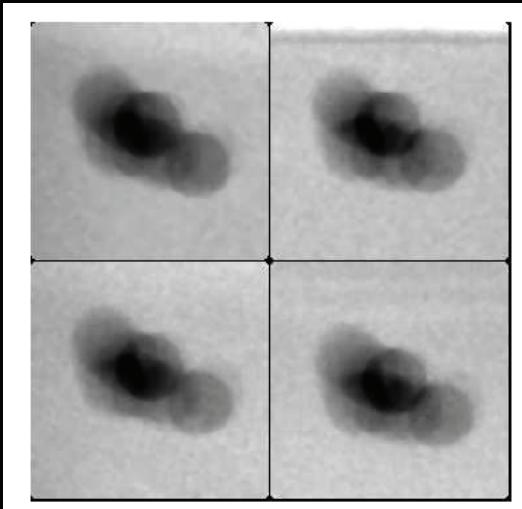
8 different people, handholding camera, using 1 second exposure



Dots from each corner

Person 1

Top
left

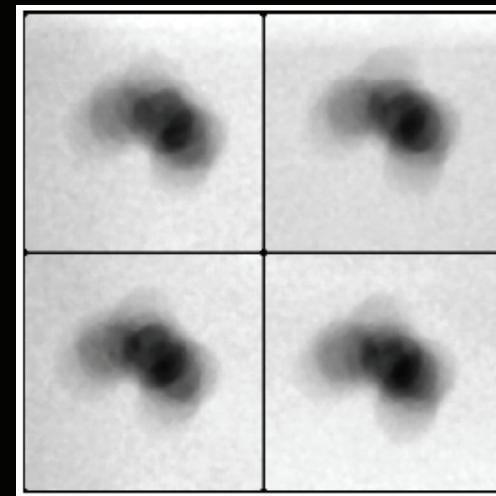


Top
right

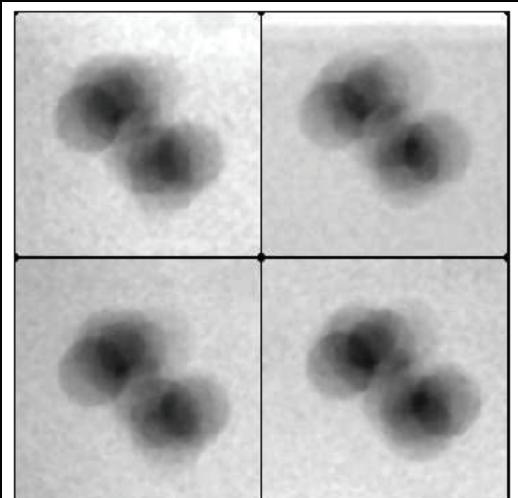
Bot.
left

Bot.
right

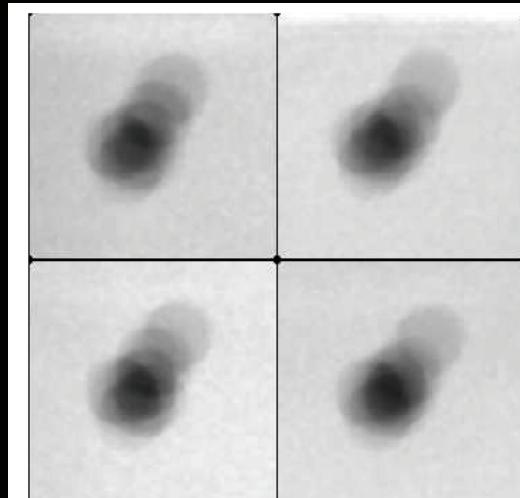
Person 2



Person 3



Person 4

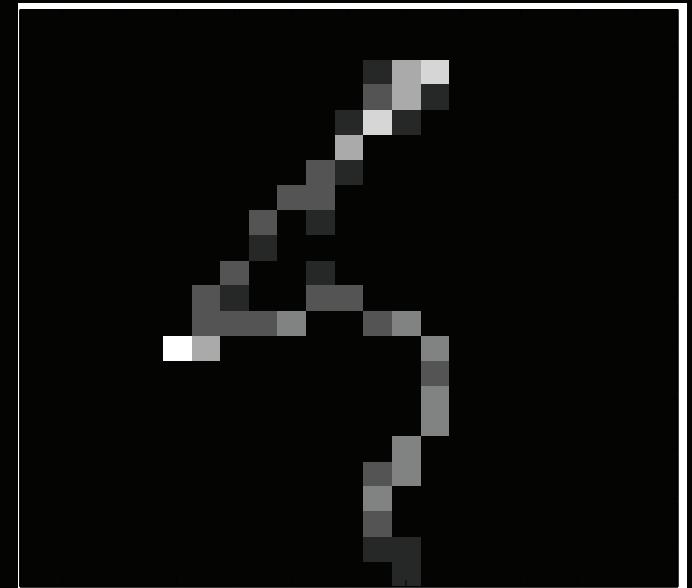


Synthetic example

Sharp image



Artificial
blur trajectory



Synthetic blurry image



Inference – initial scale

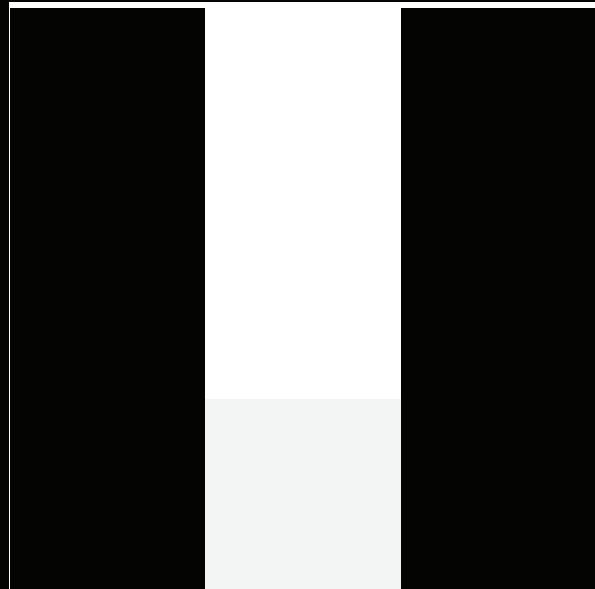
Image before



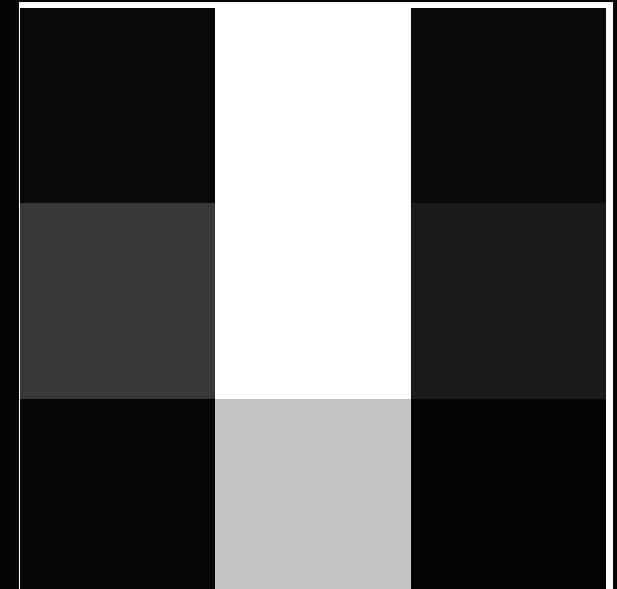
Image after



Kernel before



Kernel after



Inference – scale 2

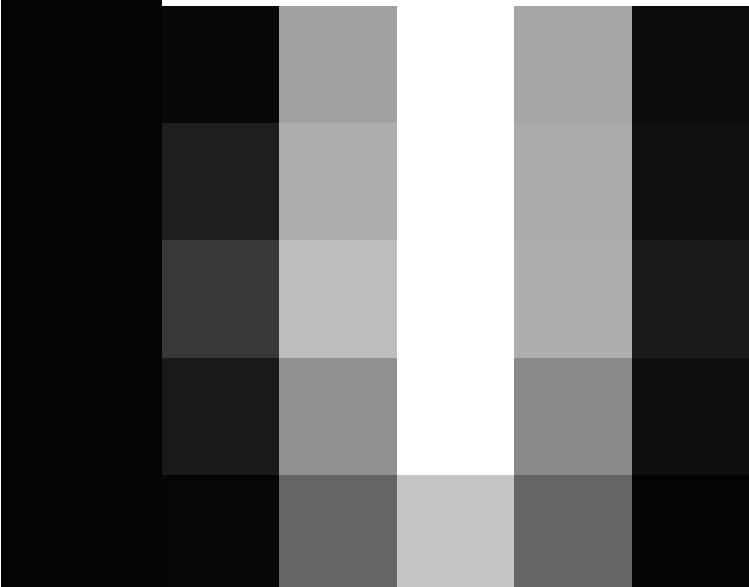
Image before



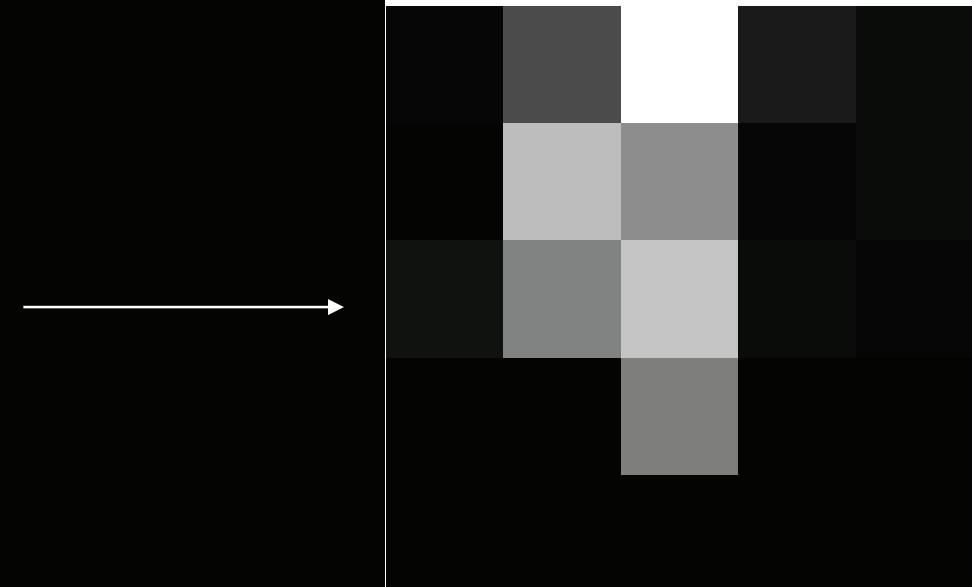
Image after



Kernel before



Kernel after



Inference – scale 3

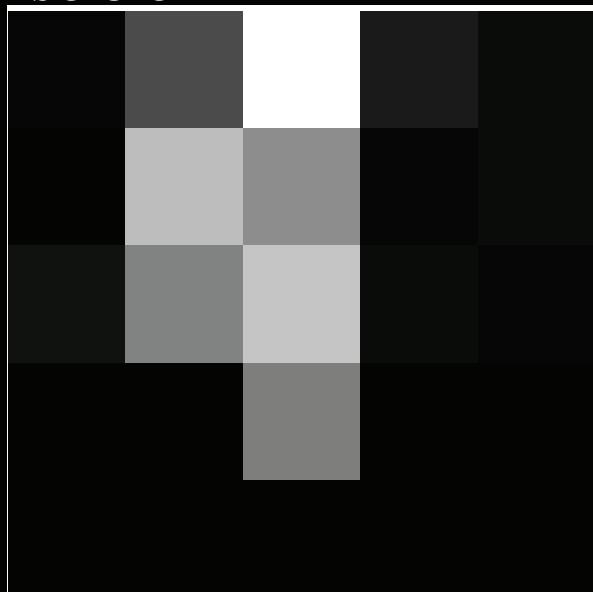
Image before



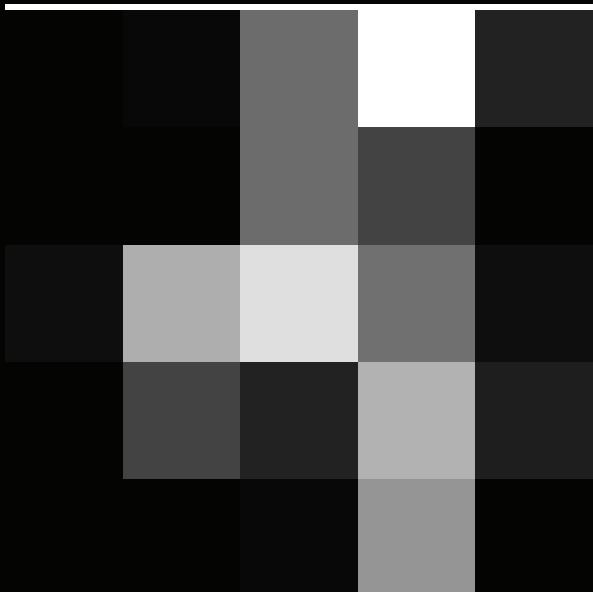
Image after



Kernel before



Kernel after



Inference – scale 4

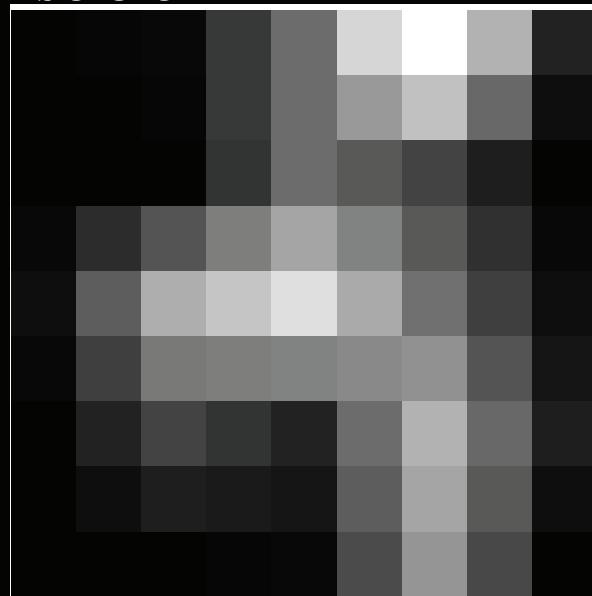
Image before



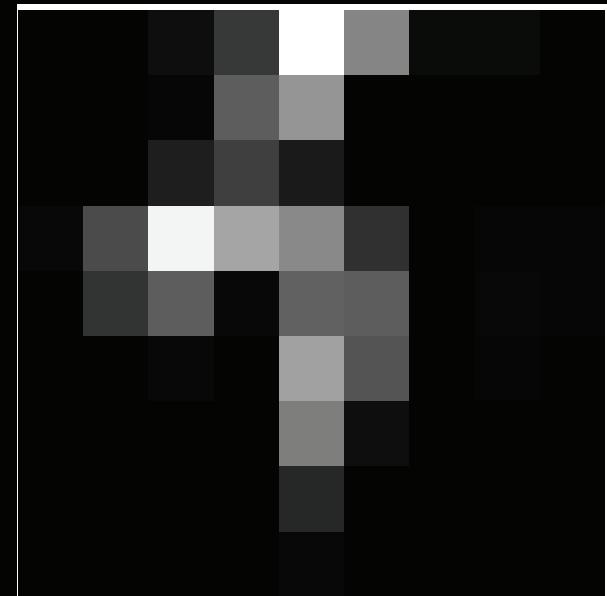
Image after



Kernel before



Kernel after



Inference – scale 5

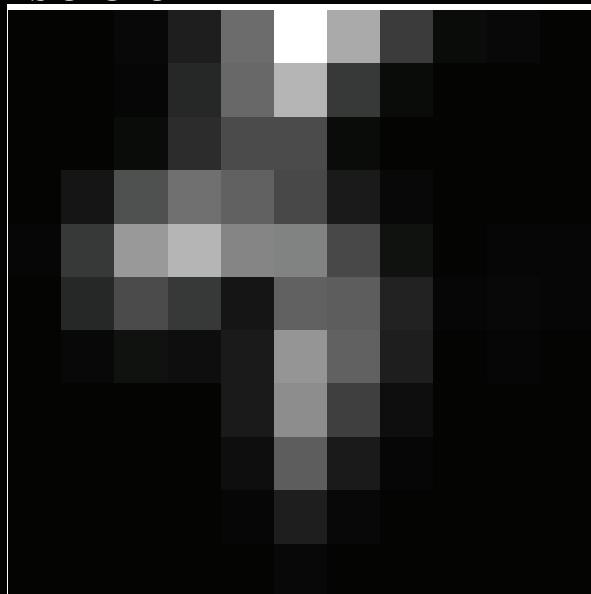
Image before



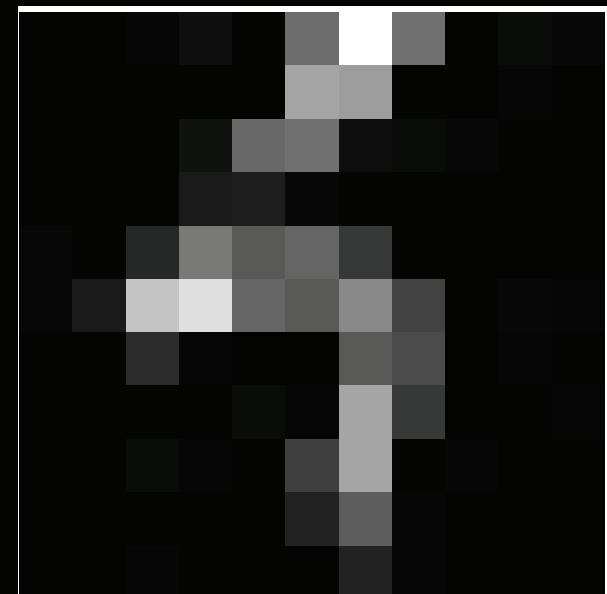
Image after



Kernel before



Kernel after



Inference – scale 6

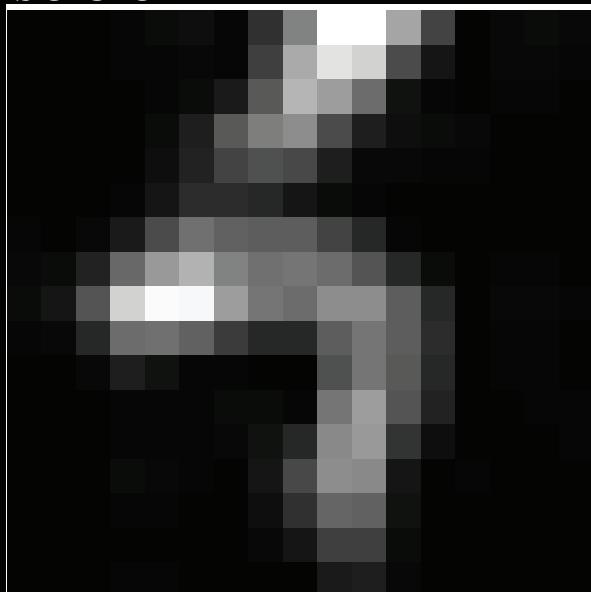
Image before



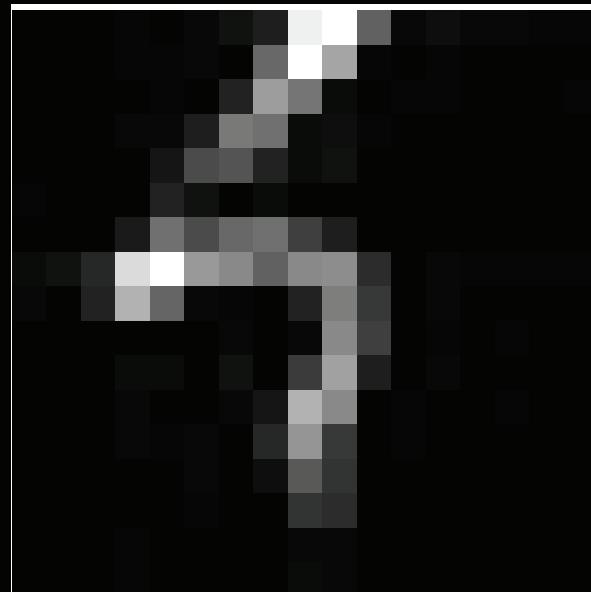
Image after



Kernel before



Kernel after



Inference – Final scale

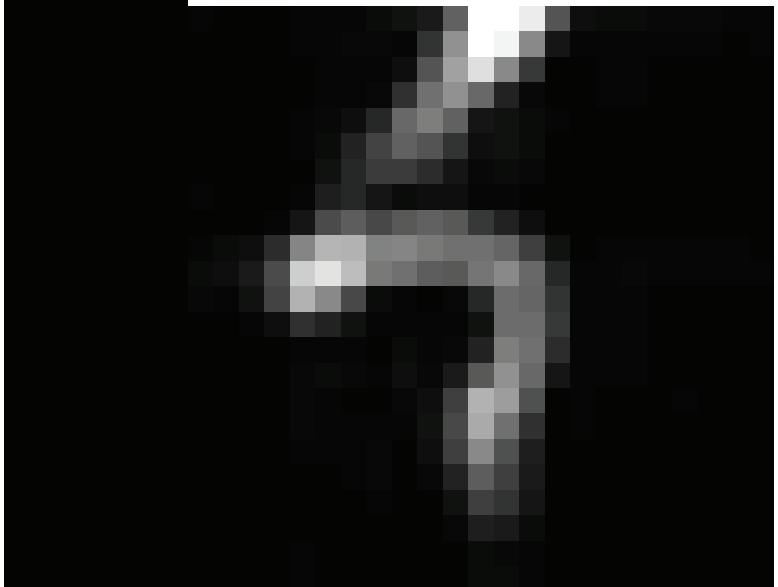
Image before



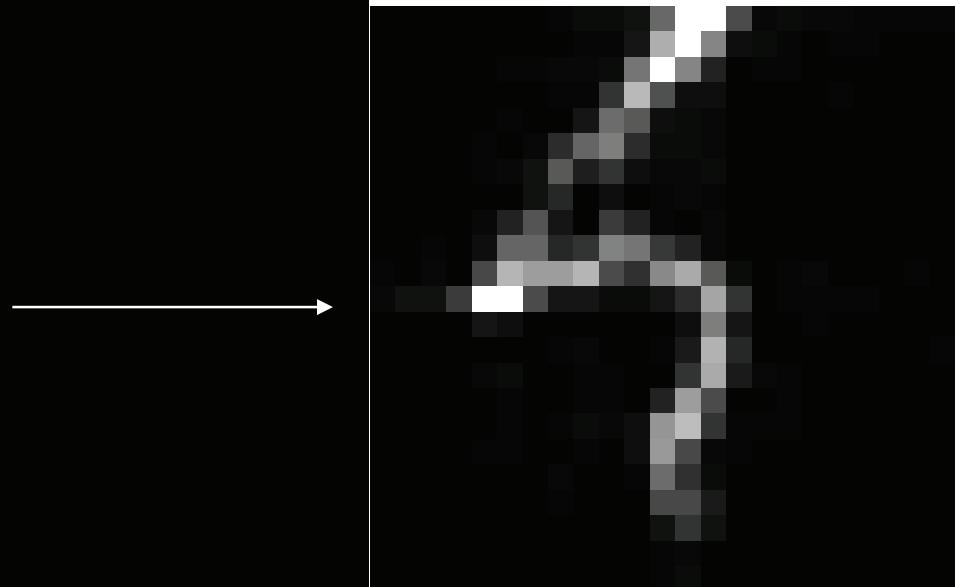
Image after



Kernel before

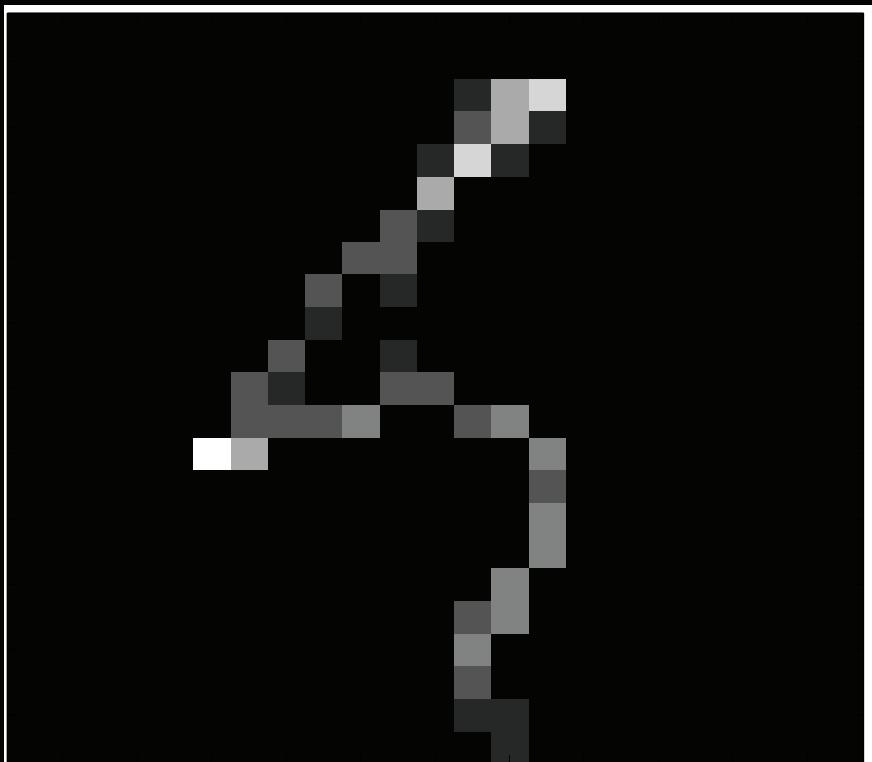


Kernel after

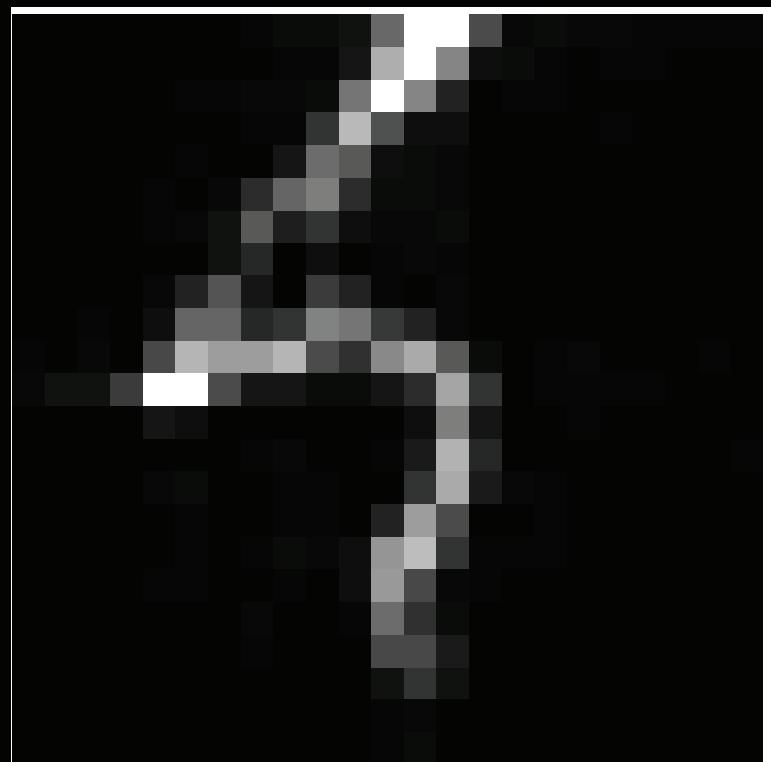


Comparison of kernels

True kernel



Estimated kernel



Blurry image



Matlab's deconvblind



Blurry image



Our output



True sharp image



What we do and don't model

DO

- Gamma correction
- Tone response curve (if known)

DON'T

- Saturation
- Jpeg artifacts
- Scene motion
- Color channel correlations

**Real
experiments**

Results on real images

Submitted by people from their own photo collections

Type of camera unknown

Output does contain artifacts

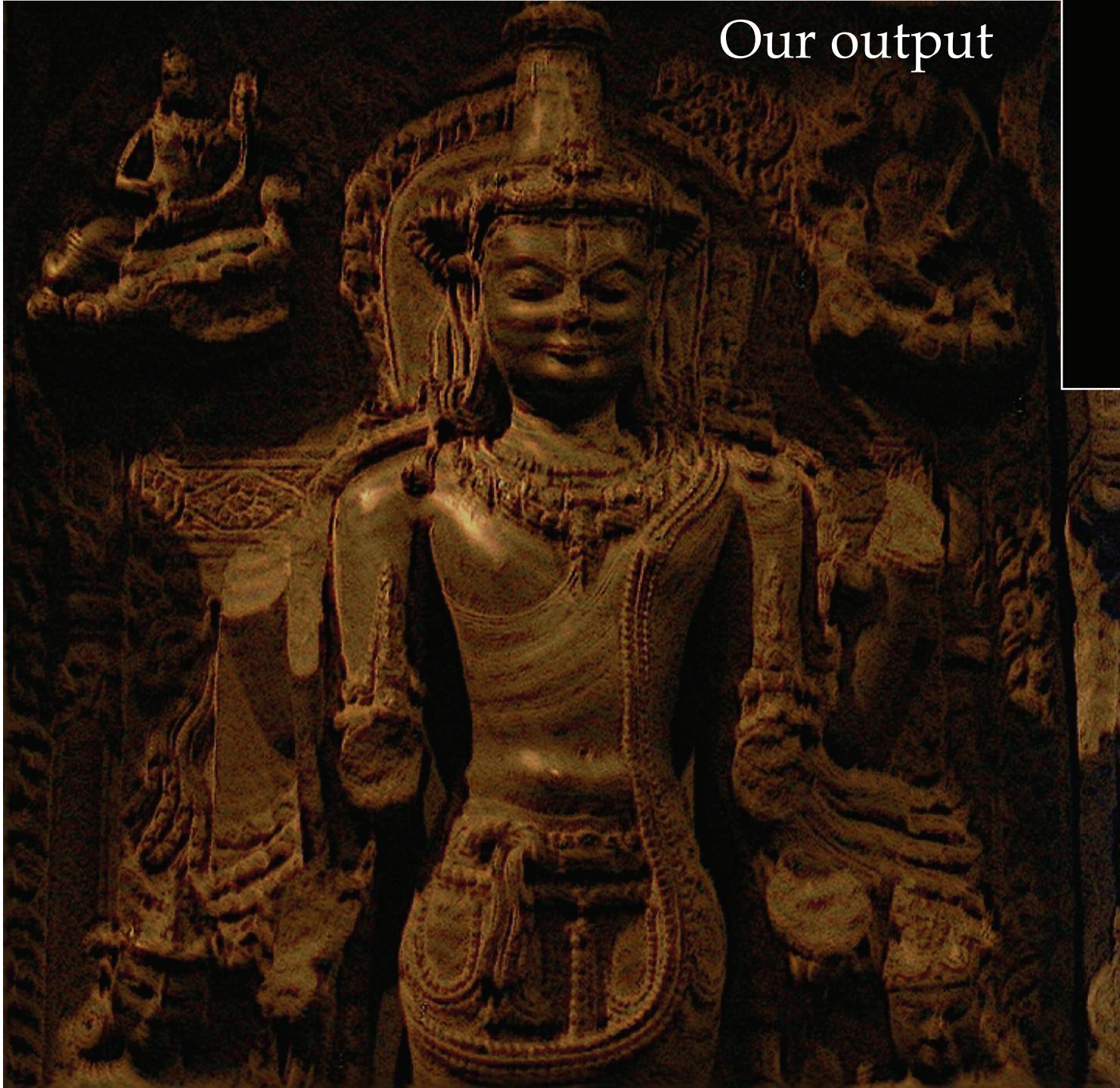
- Increased noise
- Ringing

Compare with existing methods

Original photograph



Our output



Blur kernel



Matlab's deconvblind



Close-up

Original



Our output



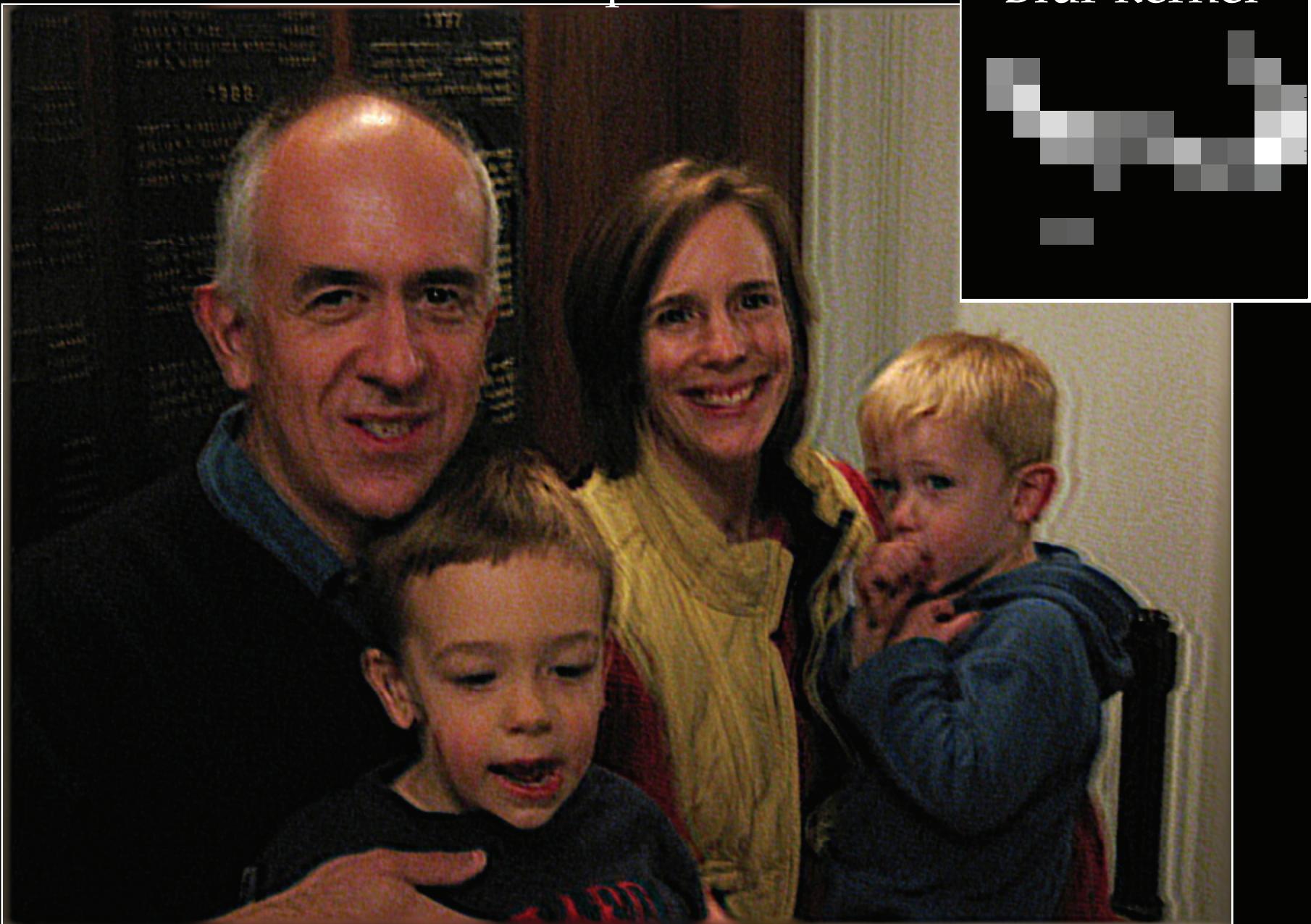
Matlab's
deconvblind



Original photograph



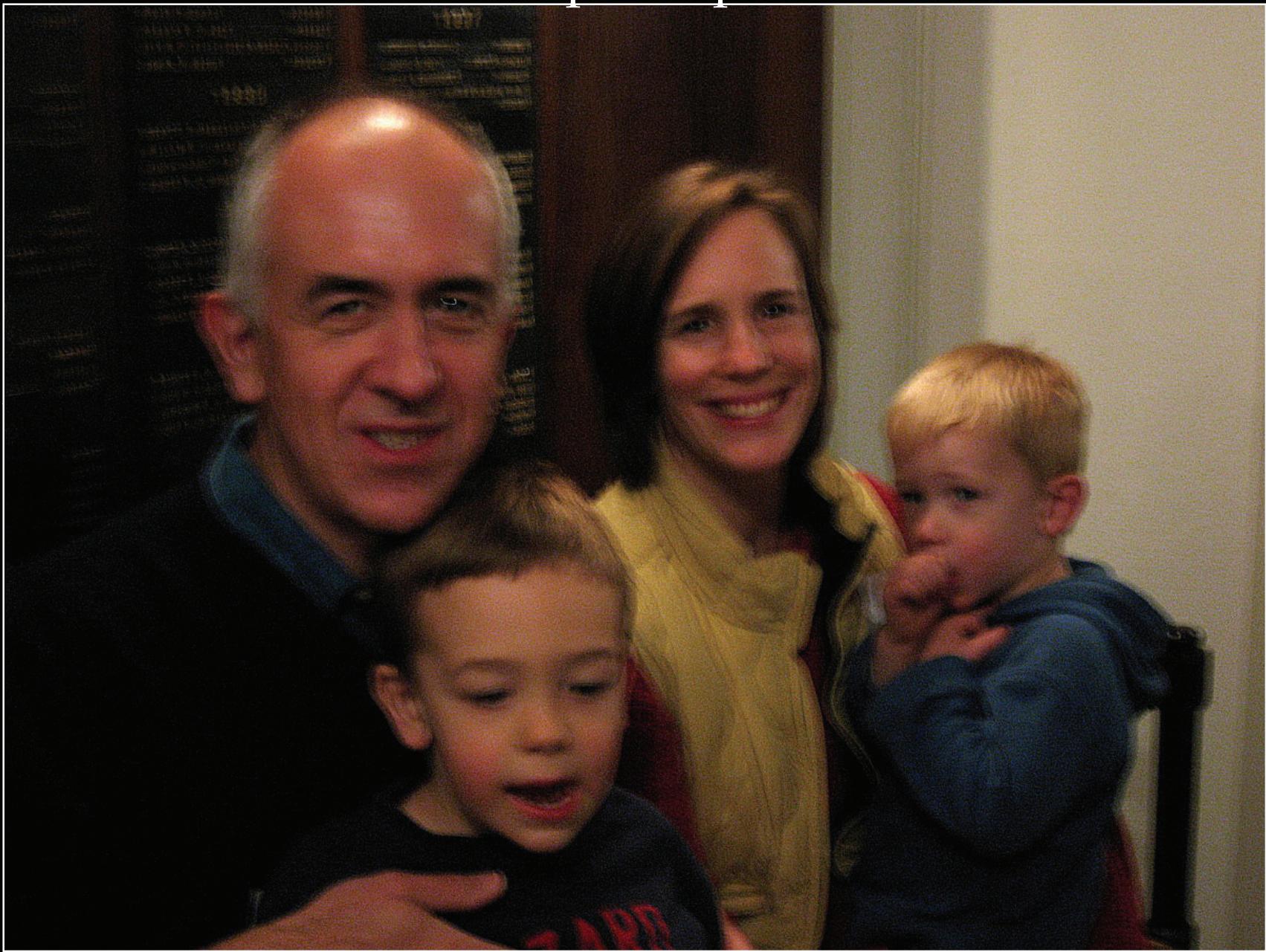
Our output



Blur kernel



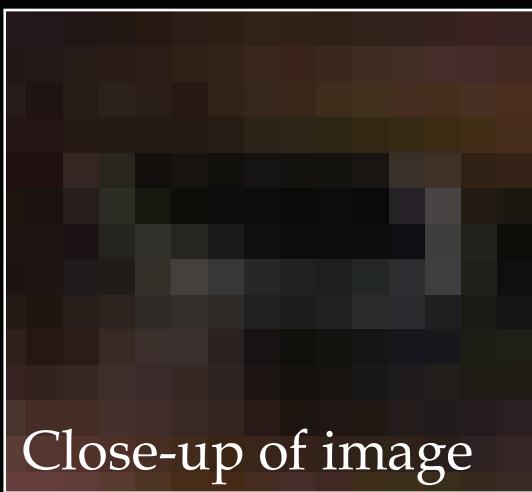
Photoshop sharpen more



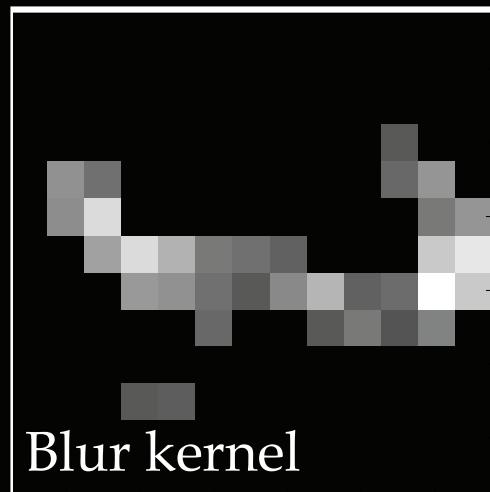
Original image



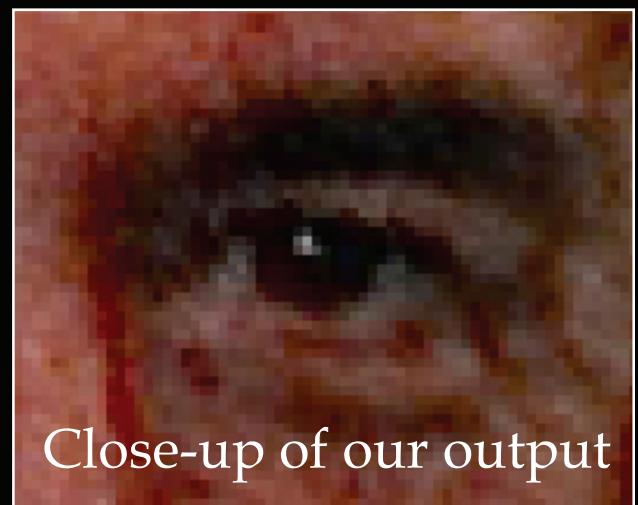
Close-up



Close-up of image



Blur kernel



Close-up of our output

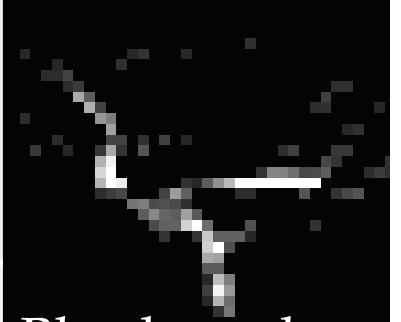
Original photograph



Our output



Blur kernel

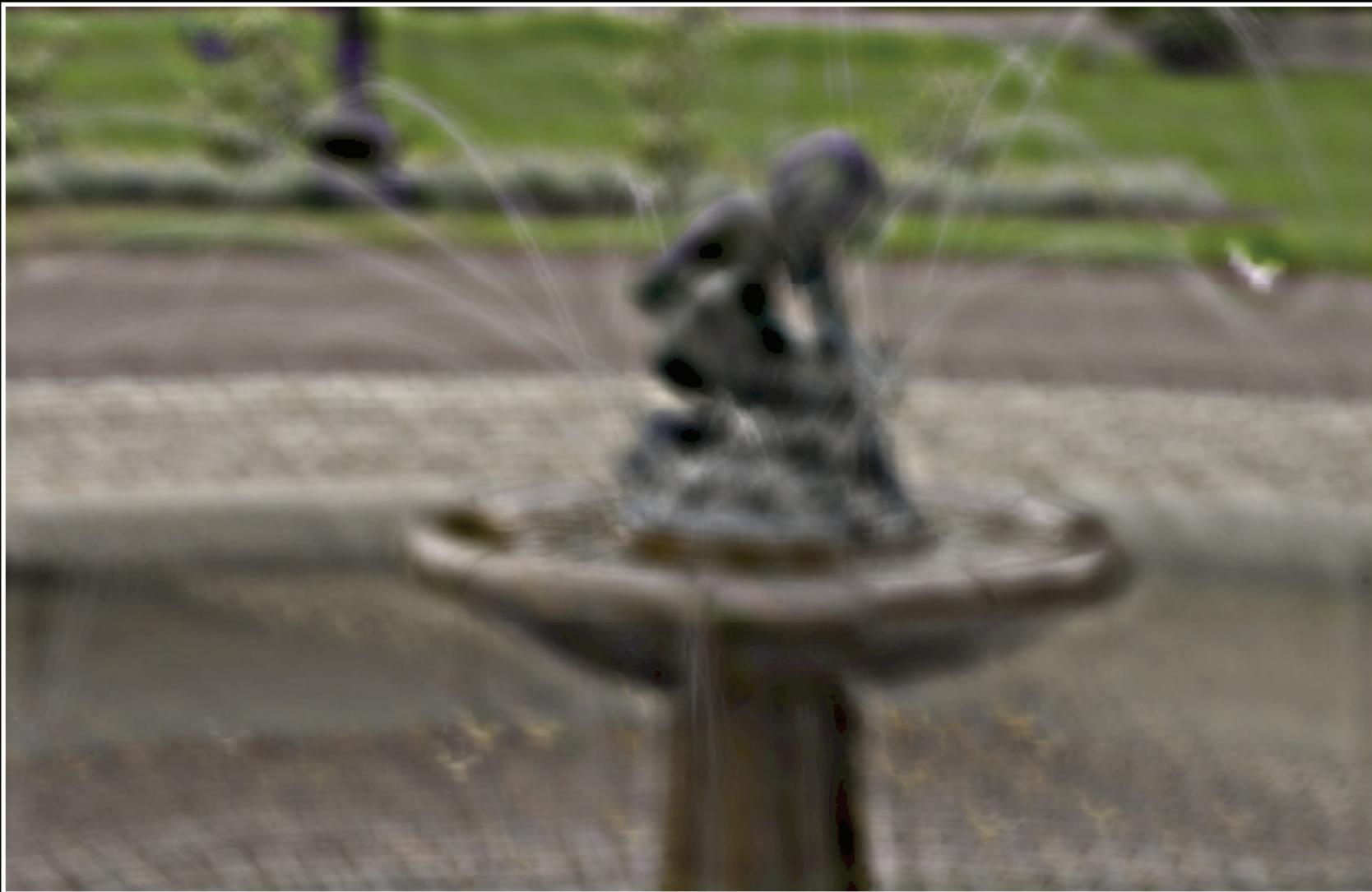


Photoshop “Smart Sharpen”



Blur kernel

Matlab's deconvblind



Original photograph



Our output



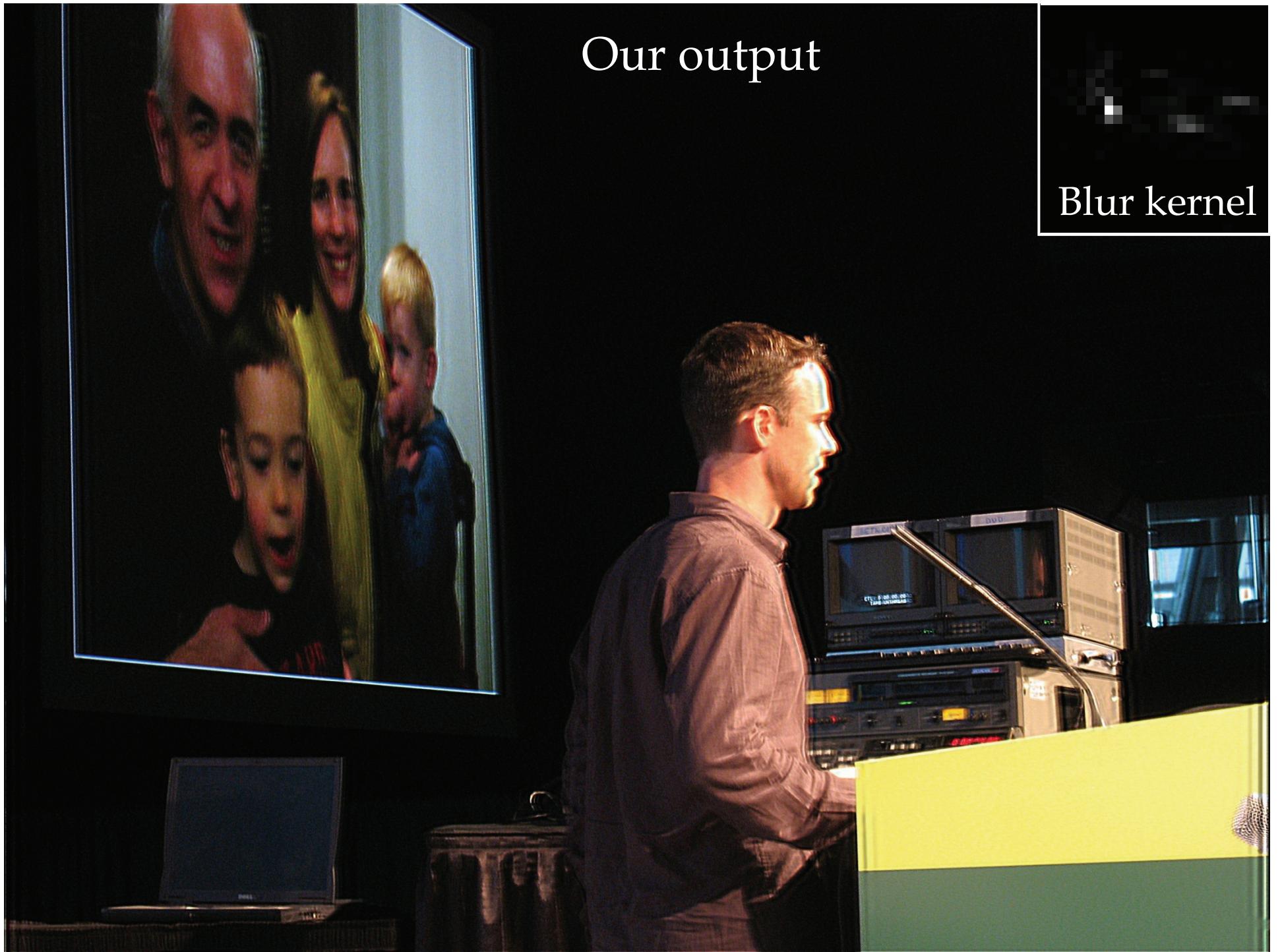
Blur kernel

Original photograph



Our output

Blur kernel

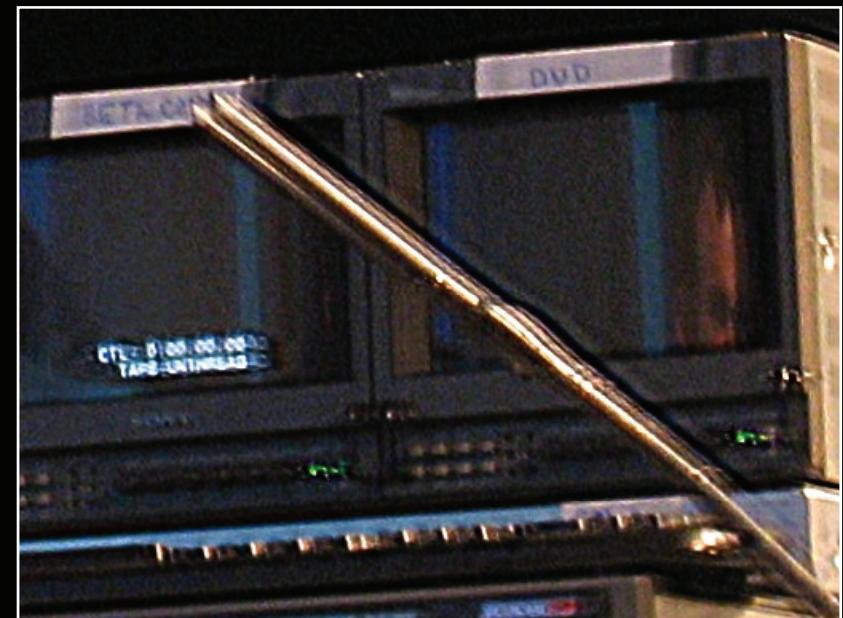


Close-up of AV equipment

Original photograph

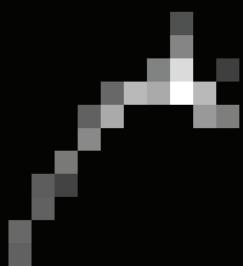


Our output



Original image





Our output

Blur kernel

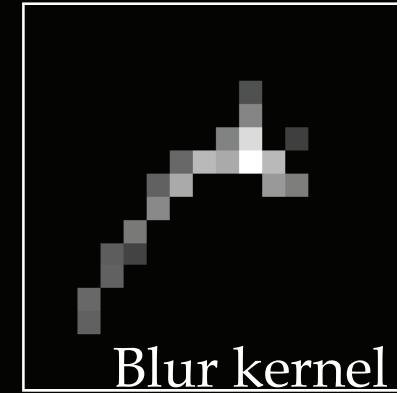


Close-up

Original image



Our output

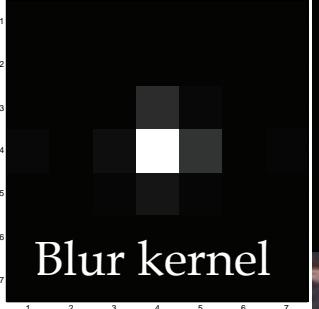


Blur kernel

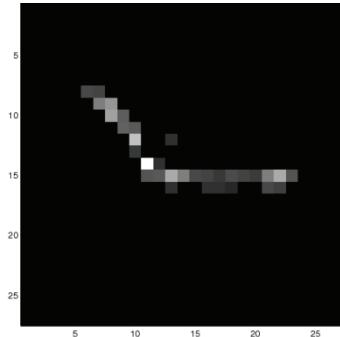
What about a sharp image?

Original photograph



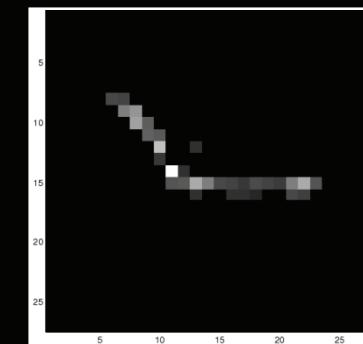






Close-up

- Original



- Output





Original photograph



Blur kernel

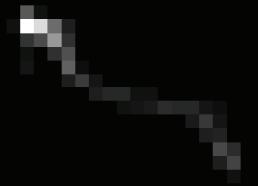
Our output

Close-up

Original image



Our output

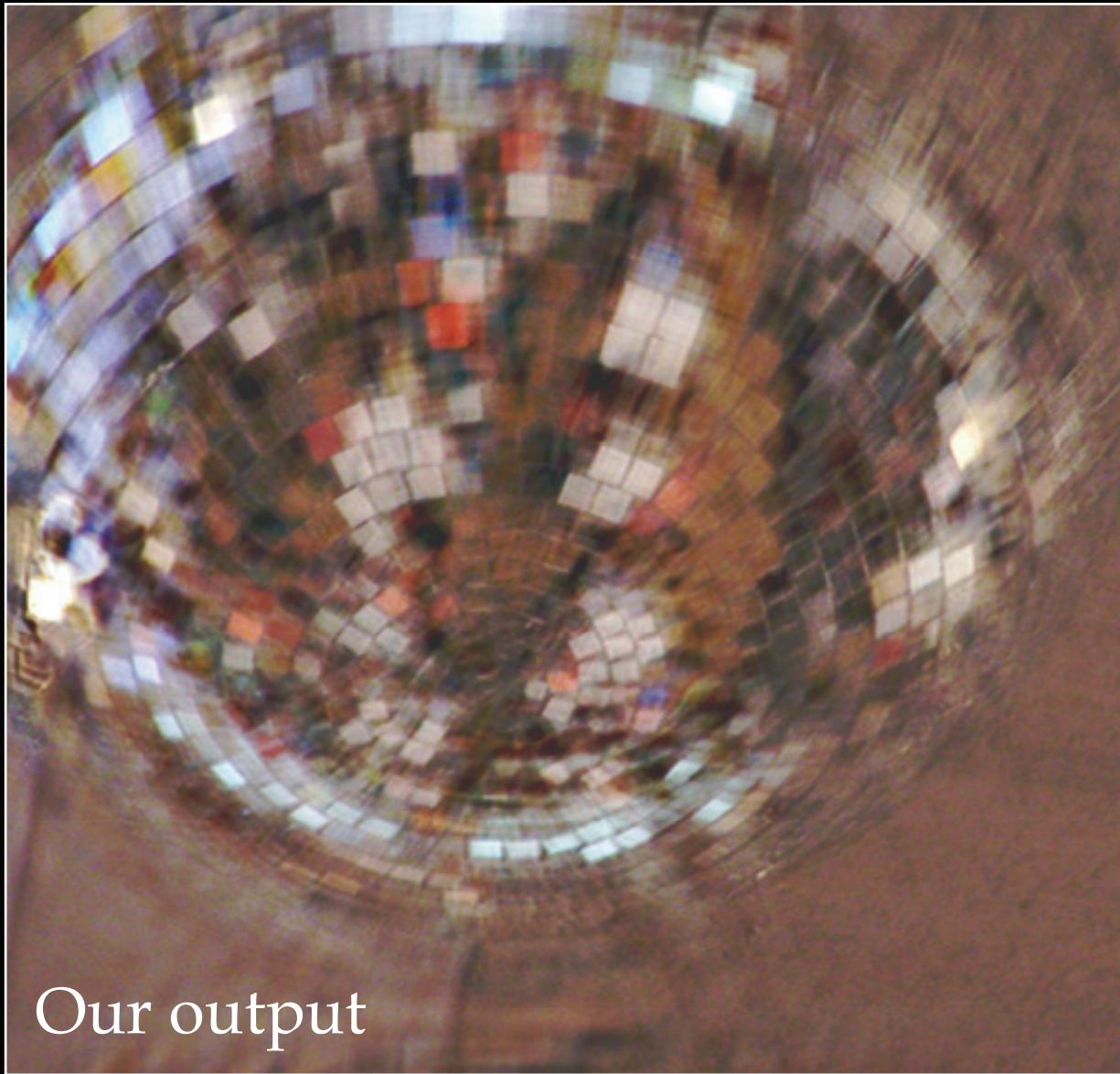


Blur kernel



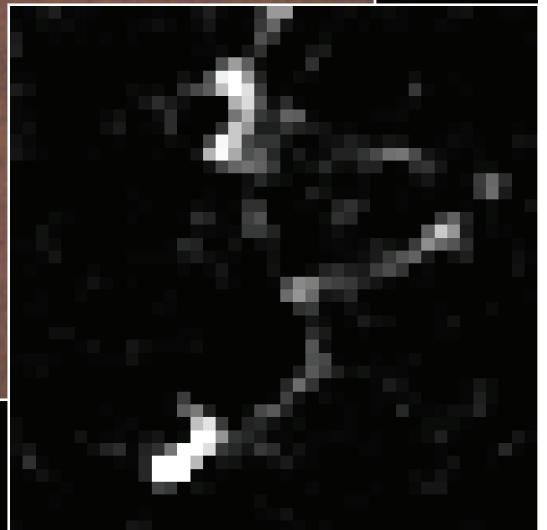
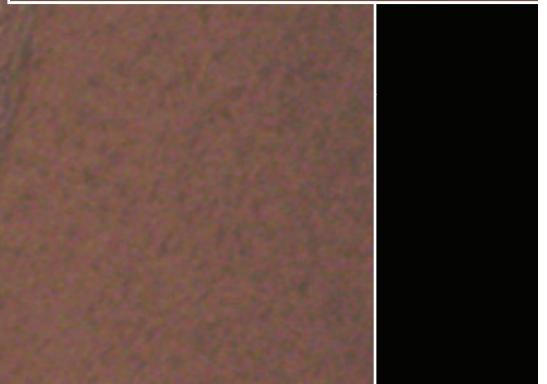
Original photograph

Blurry image patch



Our output

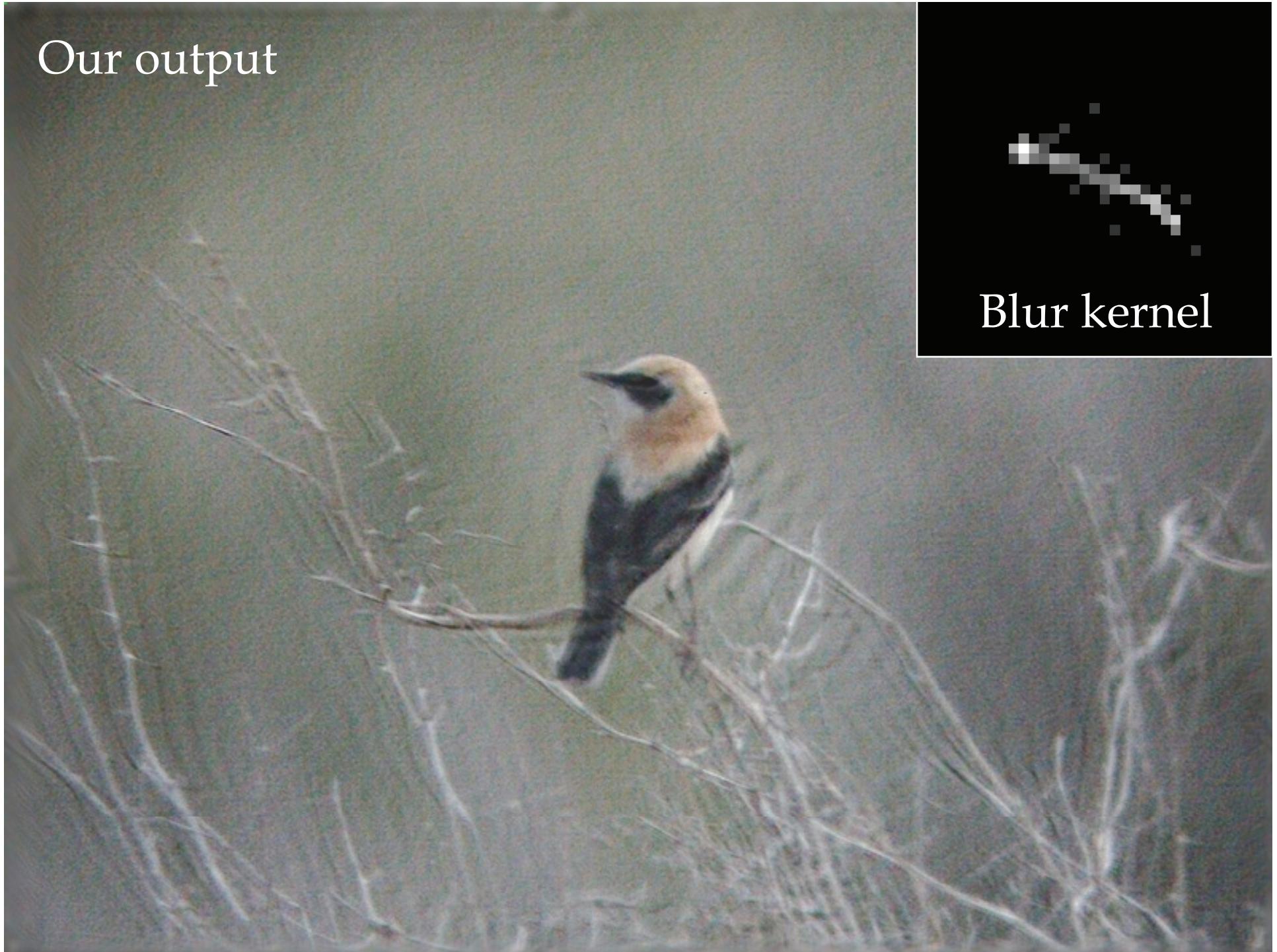
Blur kernel



Original photograph



Our output



Blur kernel

Close-up of bird

Original



Unsharp mask

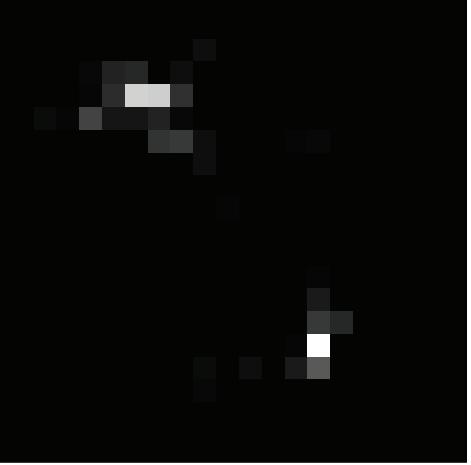


Our output



Original photograph





Blur kernel



Our output

Image artifacts & estimated kernels

Blur kernels

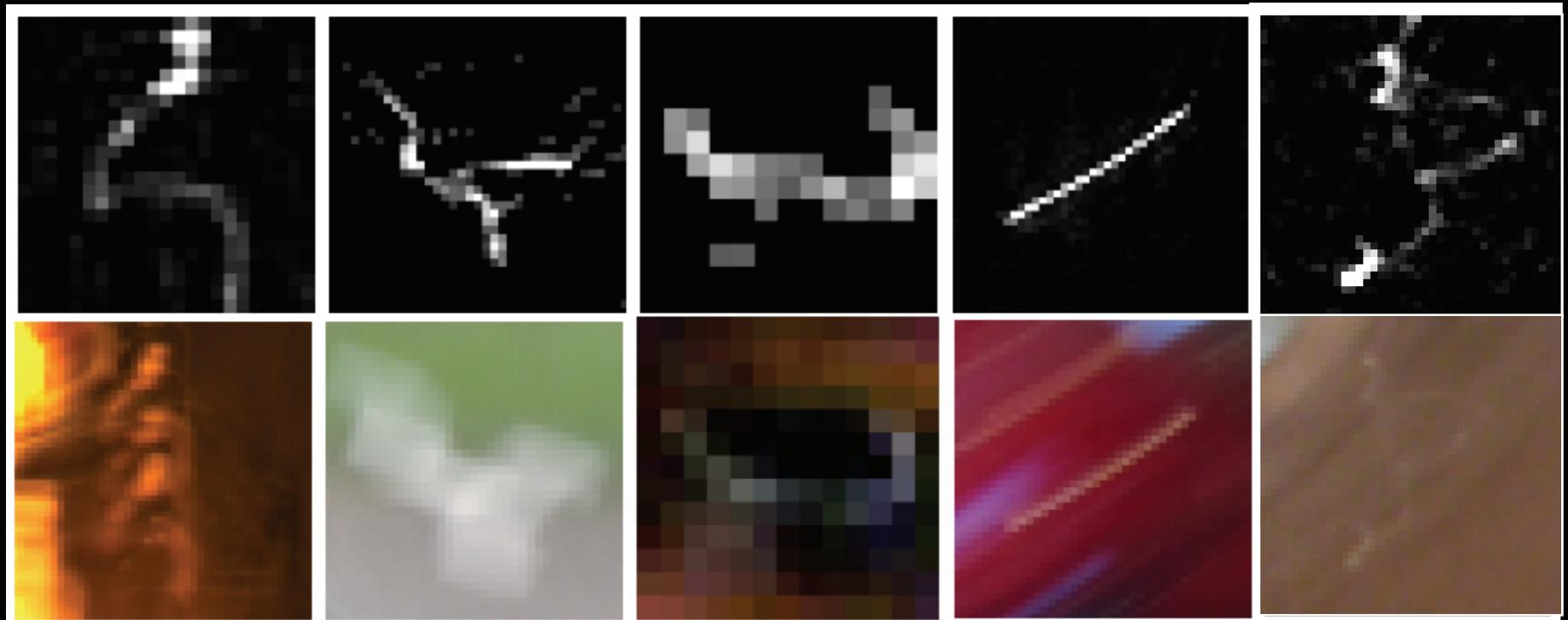


Image patterns

Note: blur kernels were inferred from large image patches,
NOT the image patterns shown

Code available online

<http://people.csail.mit.edu/fergus/research/deblur.html>

The screenshot shows a Mozilla Firefox browser window with the title bar "Removing camera shake from a single image - Mozilla Firefox". The address bar contains the URL "http://people.csail.mit.edu/fergus/research/deblur.html". The page content is as follows:

Removing Camera Shake from a Single Photograph

Rob Fergus, Barun Singh, Aaron Hertzmann,
Sam T. Roweis and William T. Freeman

Massachusetts Institute of Technology
and
University of Toronto

Powerpoint for SIGGRAPH presentation (59MB)

Videos from the presentation:
[Slow motion, pt.1](#) (wmv, 4.3MB), [Slow motion, pt.2](#) (wmv, 10.5MB)
[Camera Trajectory](#) (avi, 4.9MB), [Normal speed](#) (wmv, 4.6MB)
[** A zip file including ppt and all the videos \(78MB\)](#)

Some pictures from the paper (11.8MB)
Includes inferred blur kernels.
N.B. Please take care not to compress the images - compression artifacts look very much like blur.
I will add more images to the ZIP file as I get permission from their owners.
If you are from the media, please email me before using them.

Matlab source code

Please fill in the form [here](#) and I will email you the code.
Here is the accompanying [README](#) file.

Summary

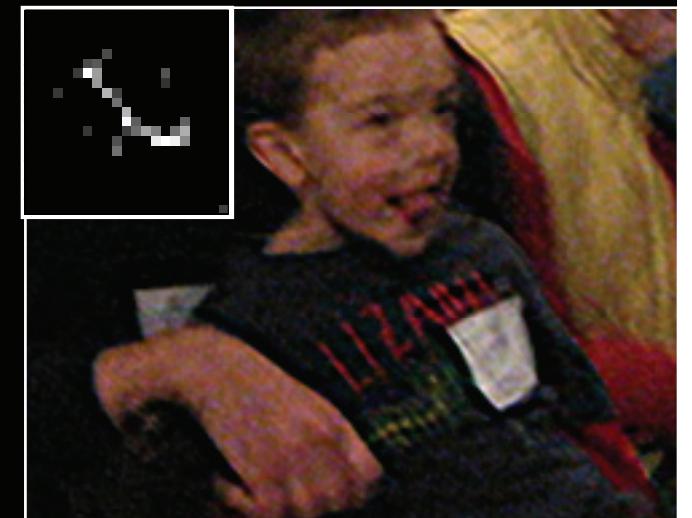
First method that can handle
complicated real-world blur kernels

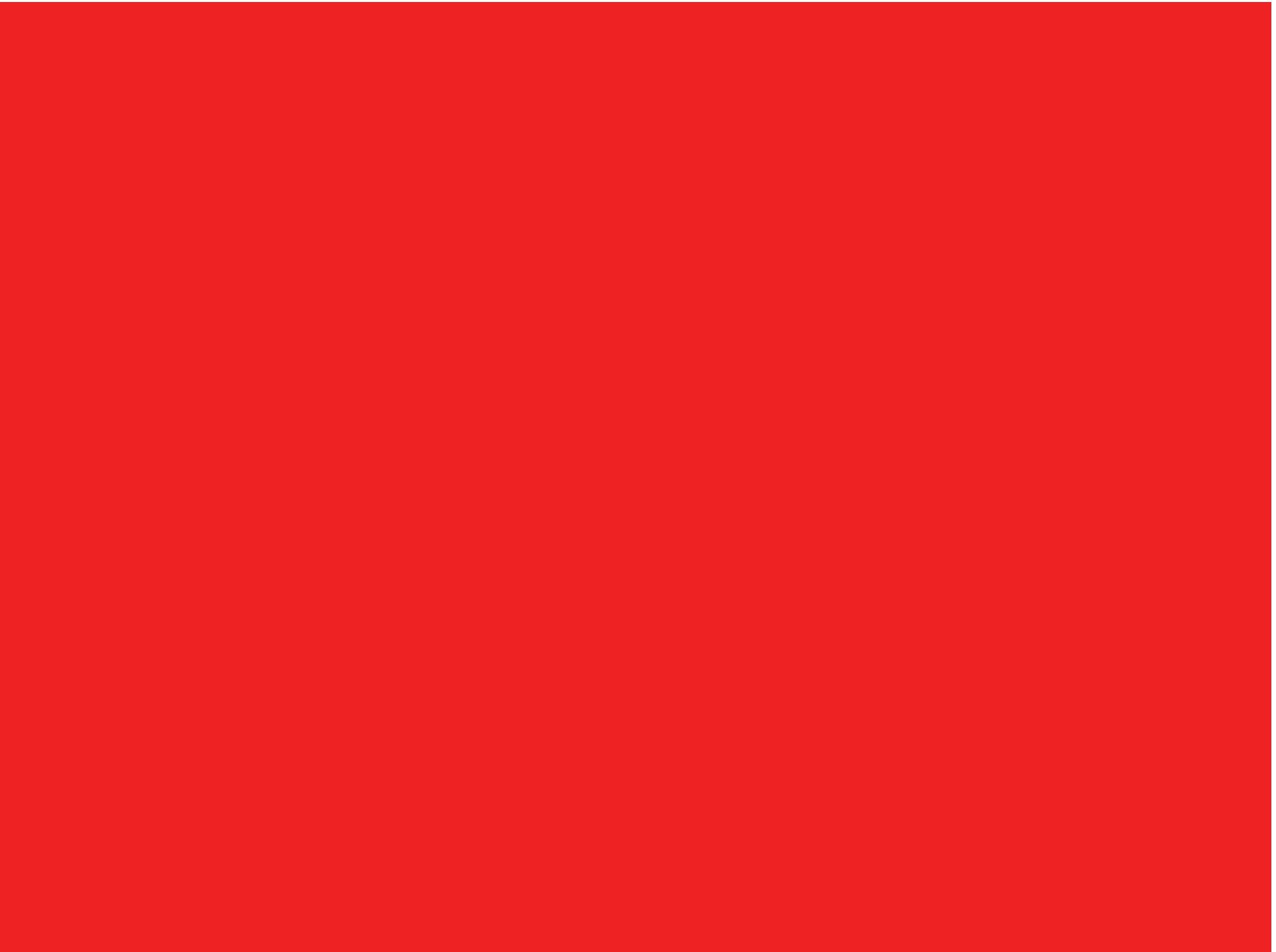
Results still contain artifacts

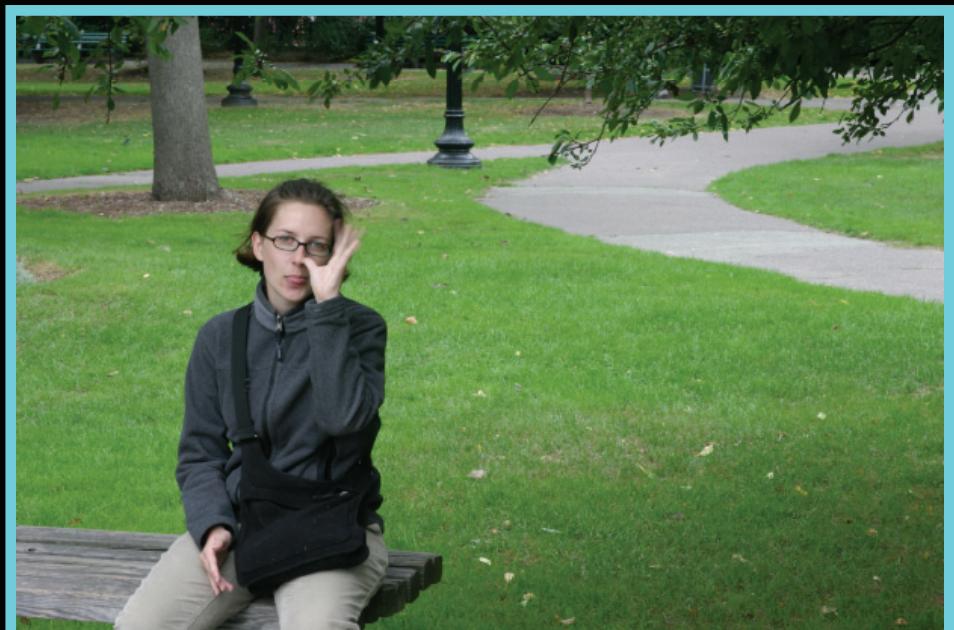
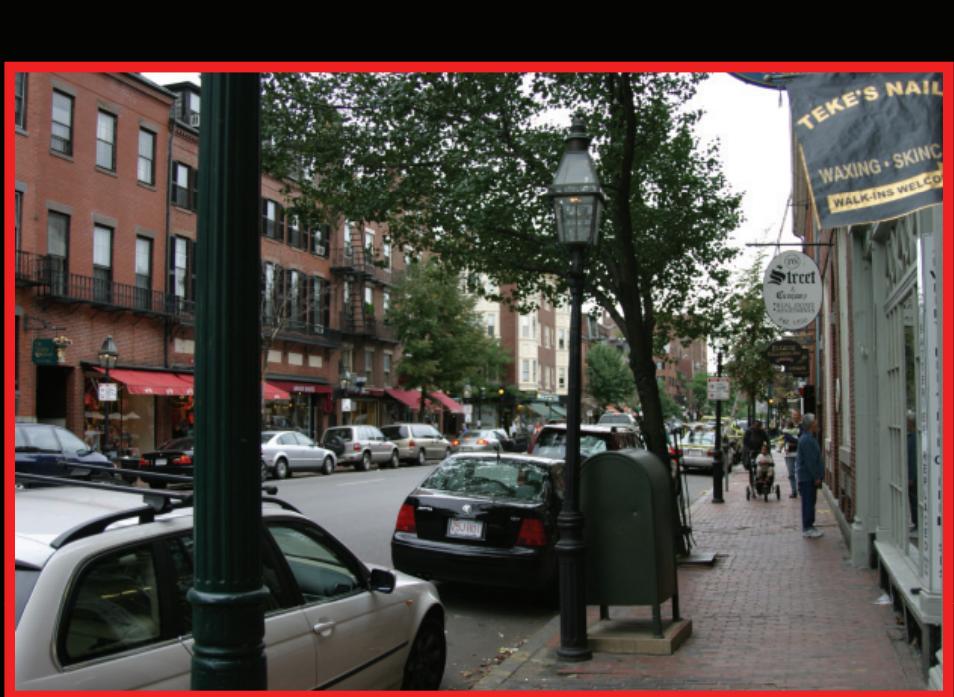
Big leap on an old, hard problem

Many things to improve:

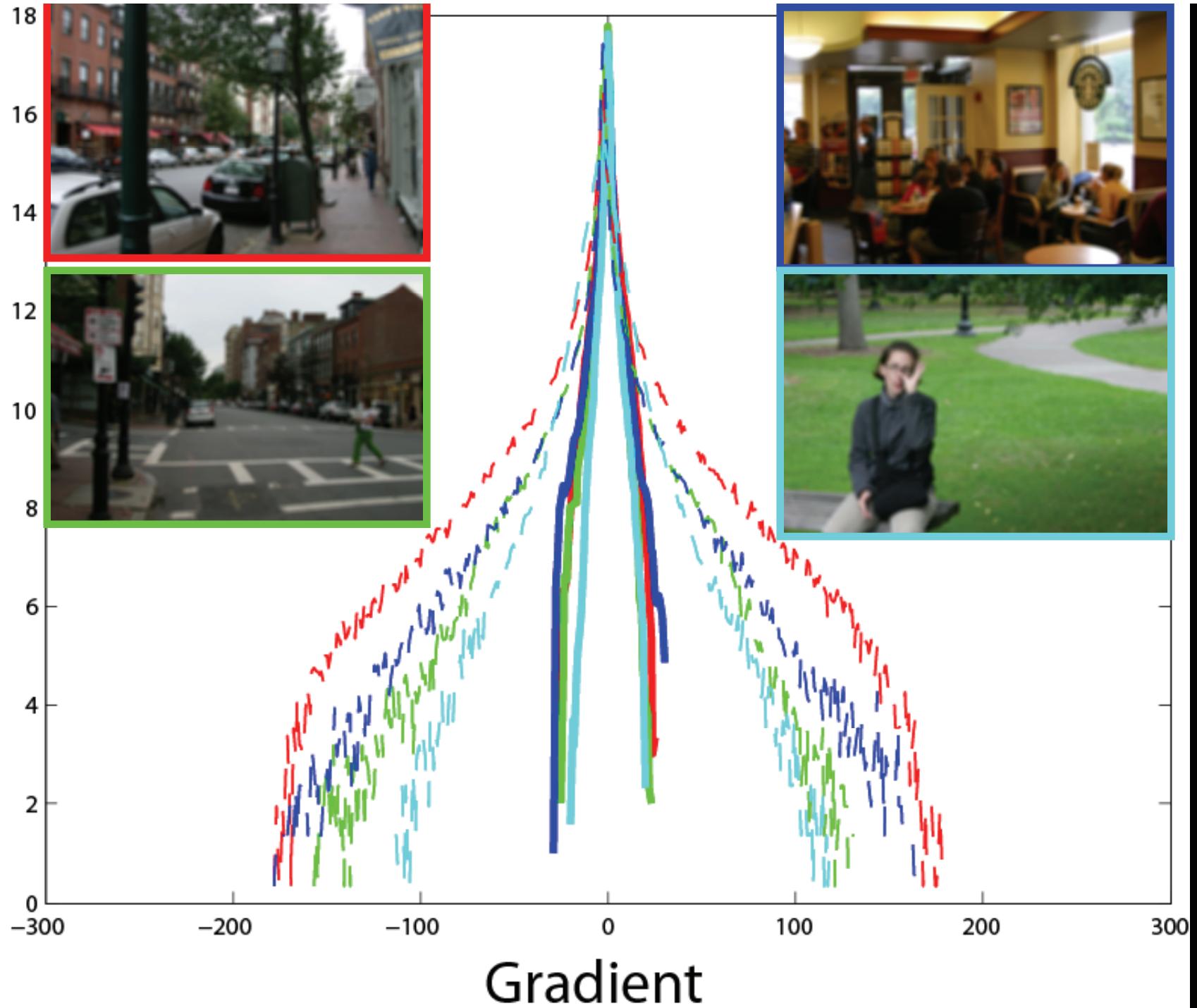
- Non-blind deconvolution, saturation etc.



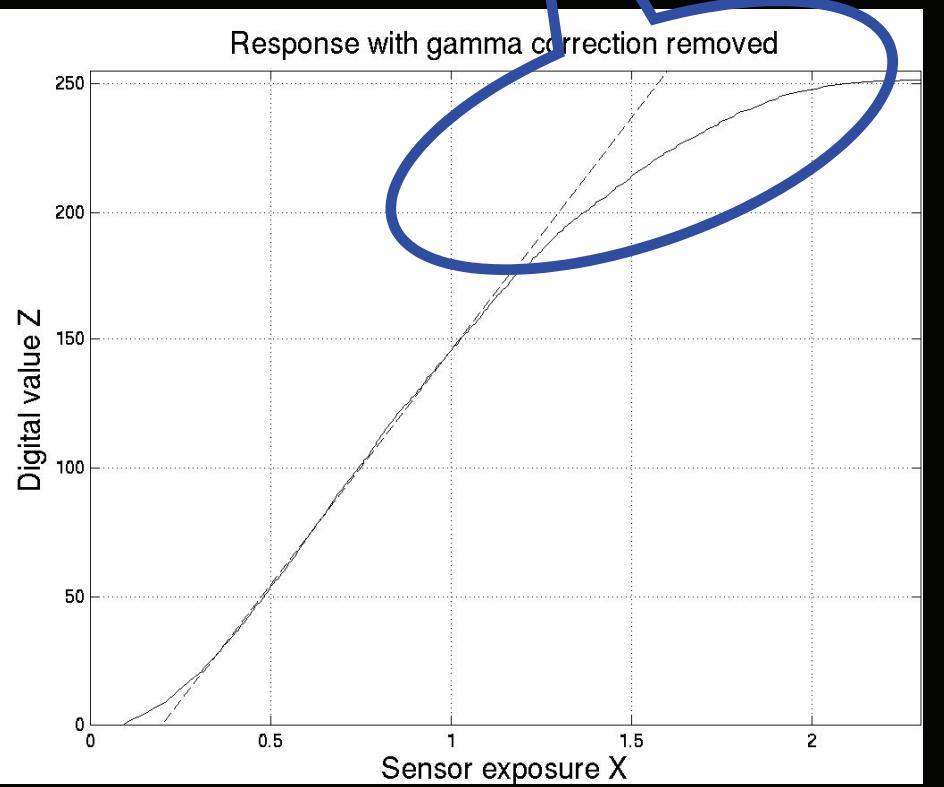
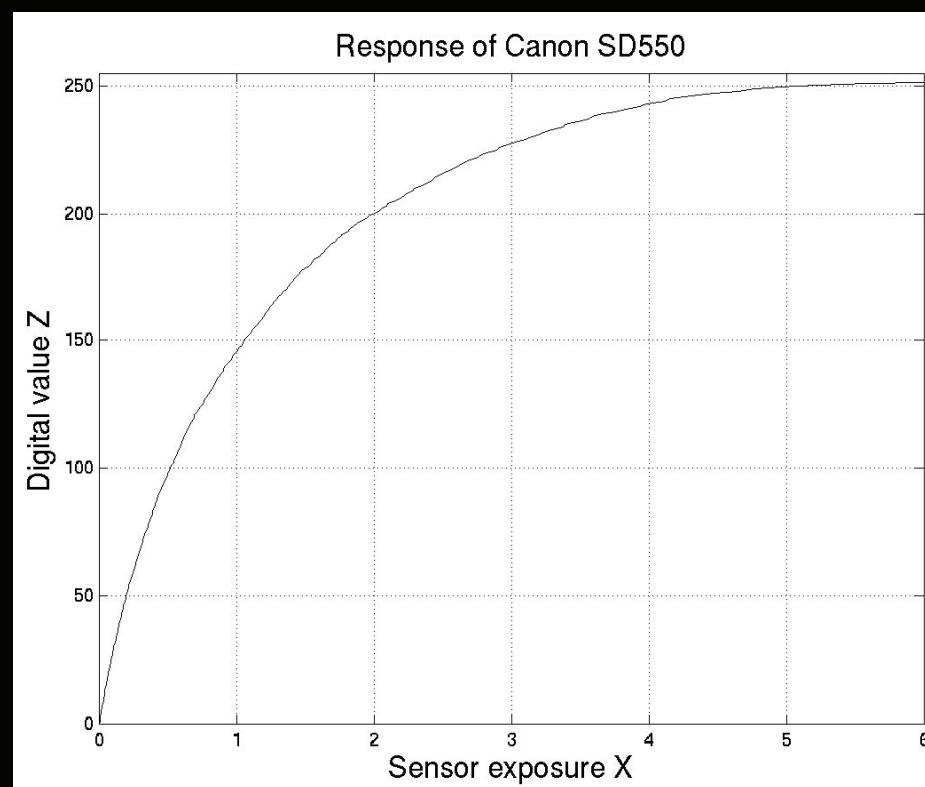
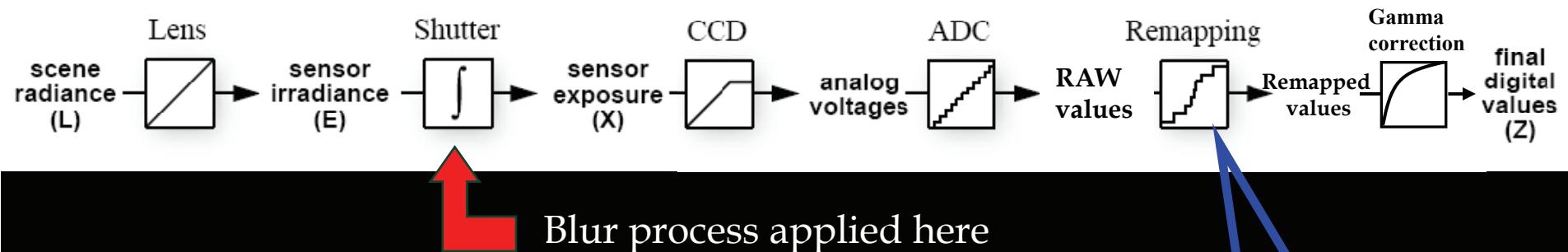




log₂ # of pixels



Digital image formation process



Simple 1-D example

$$y = bx + n$$

y = observed image

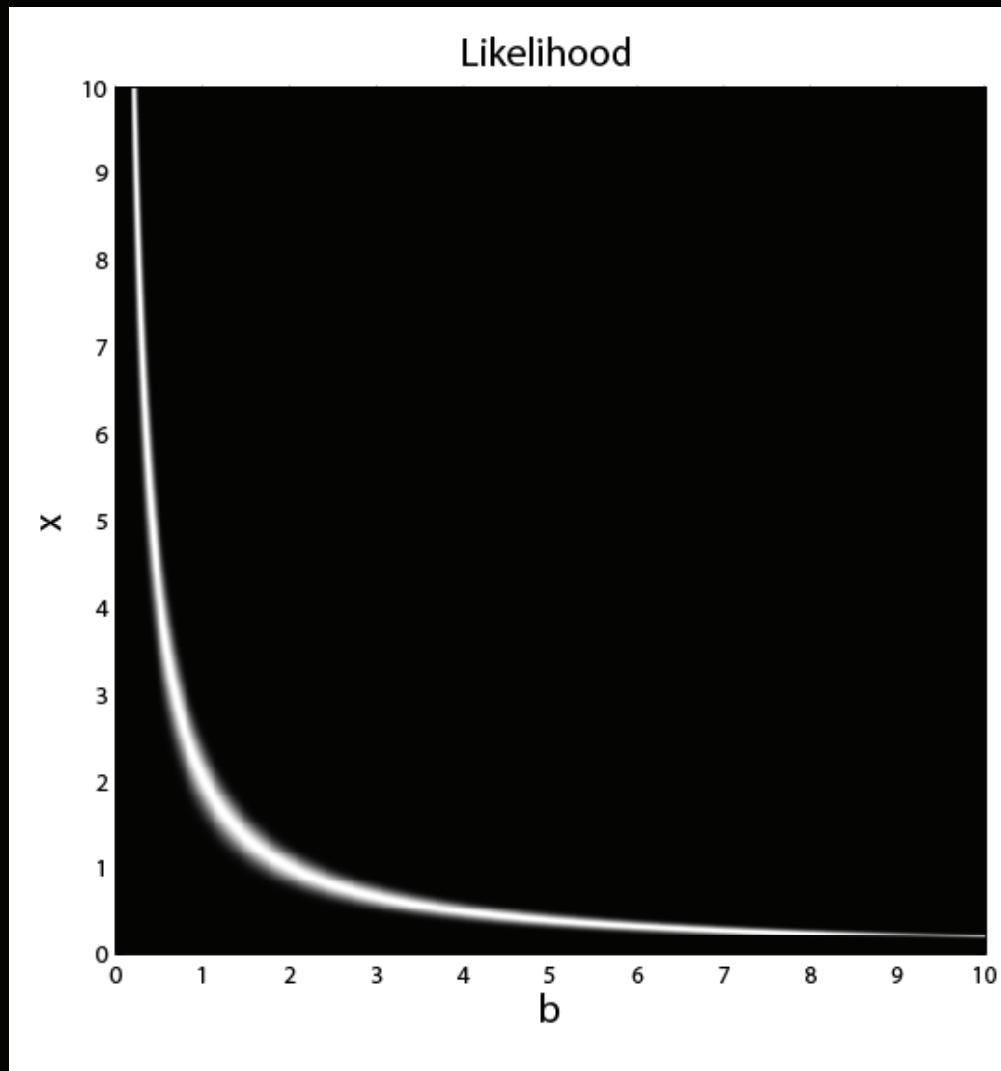
b = blur

x = sharp image

n = noise $\sim N(0, \sigma^2)$

Let $y = 2$

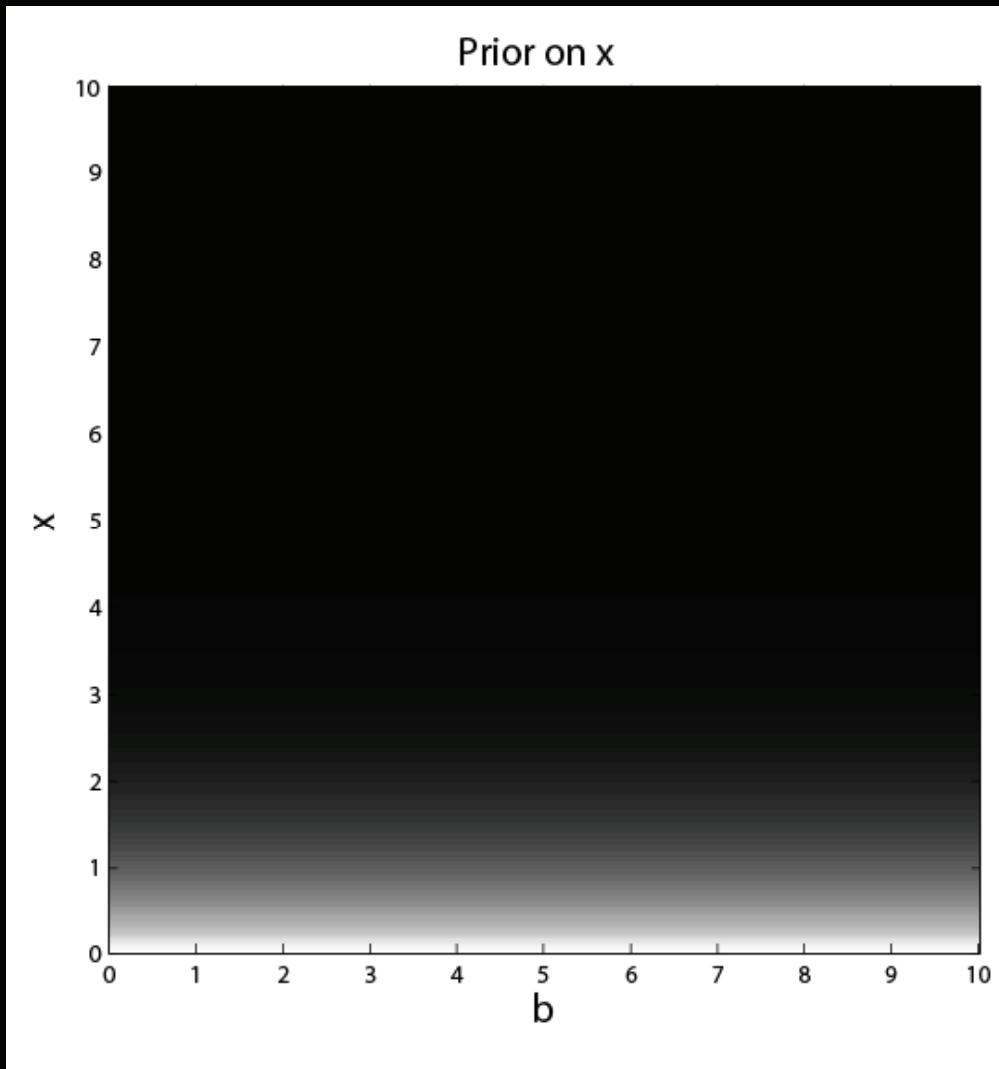
$$p(b, x|y) = k \boxed{p(y|b, x)} p(x) p(b)$$



Let $y = 2$
 $\sigma^2 = 0.1$

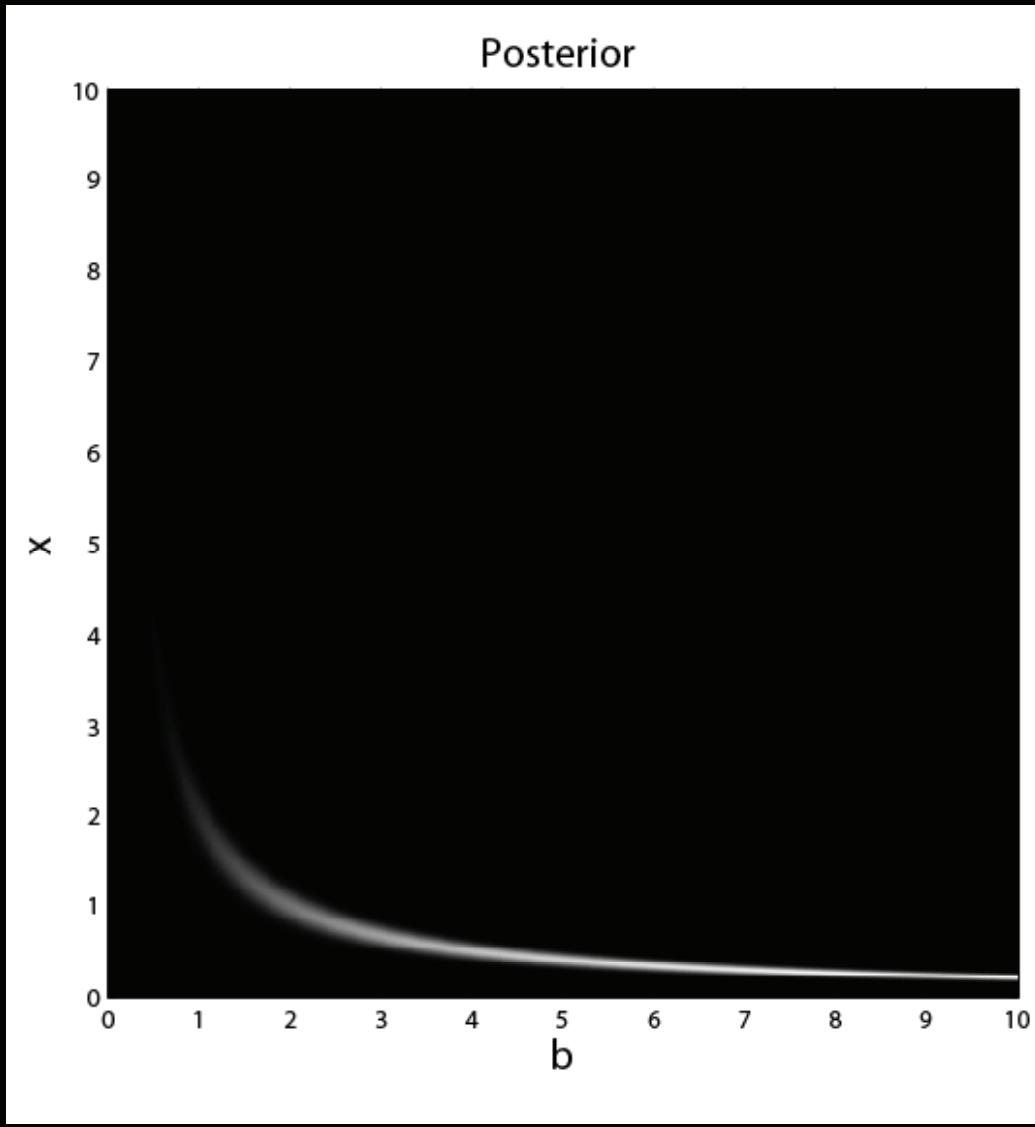
$$\mathcal{N}(y|bx, \sigma^2)$$

$$p(b, x|y) = k \quad p(y|b, x) \boxed{p(x)} \quad p(b)$$



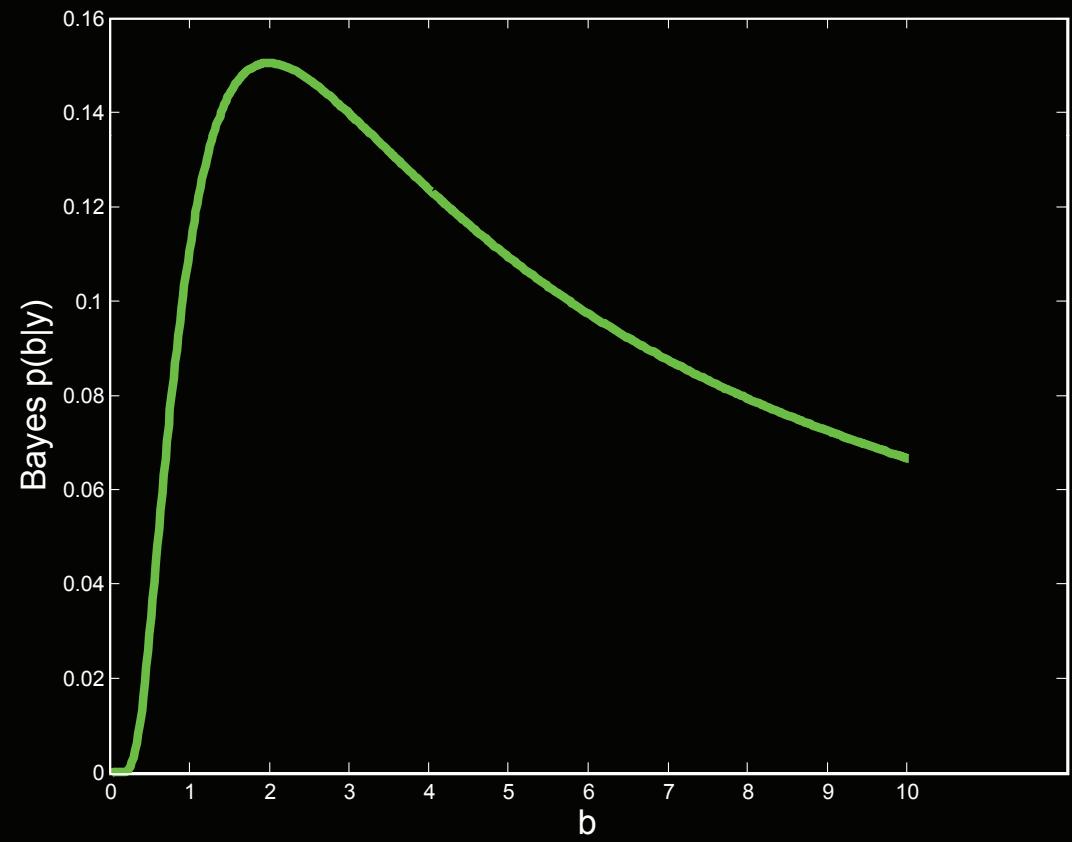
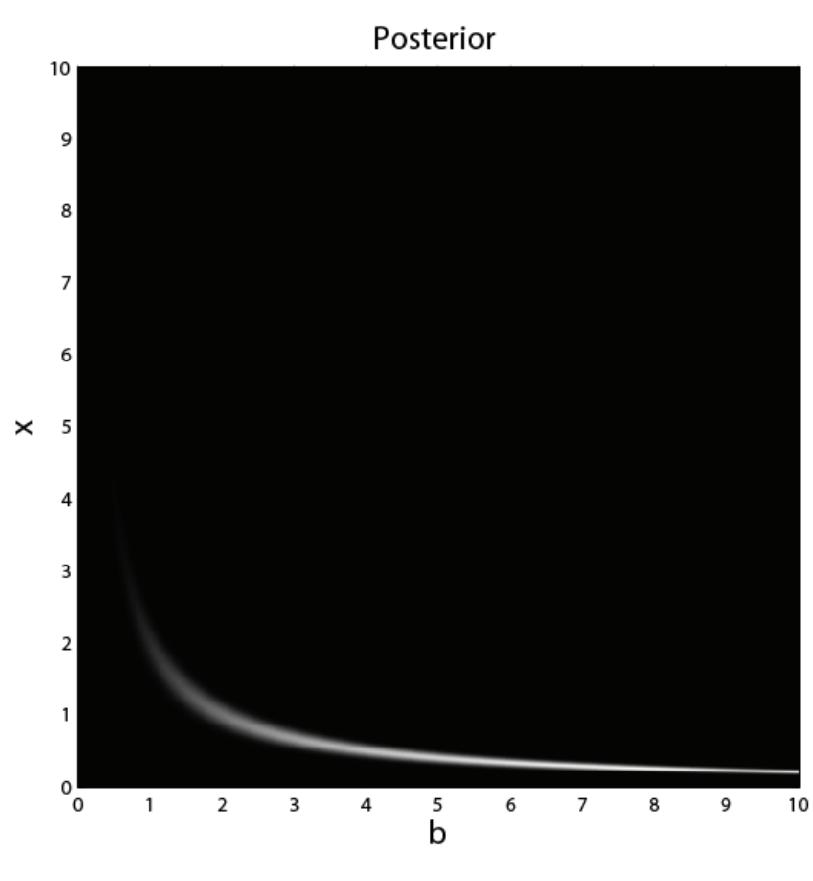
Gaussian distribution:
 $\mathcal{N}(x|0, 2)$

$$p(b, x|y) = k \ p(y|b, x) \ p(x) \ p(b)$$



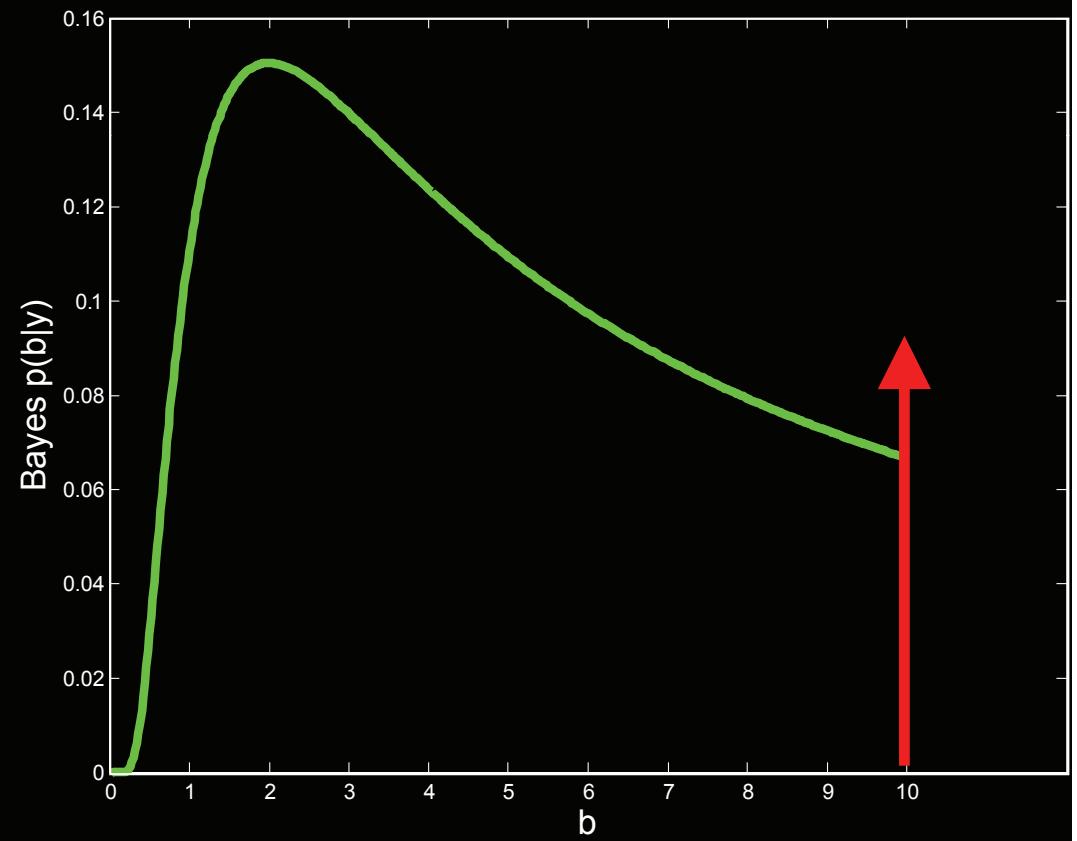
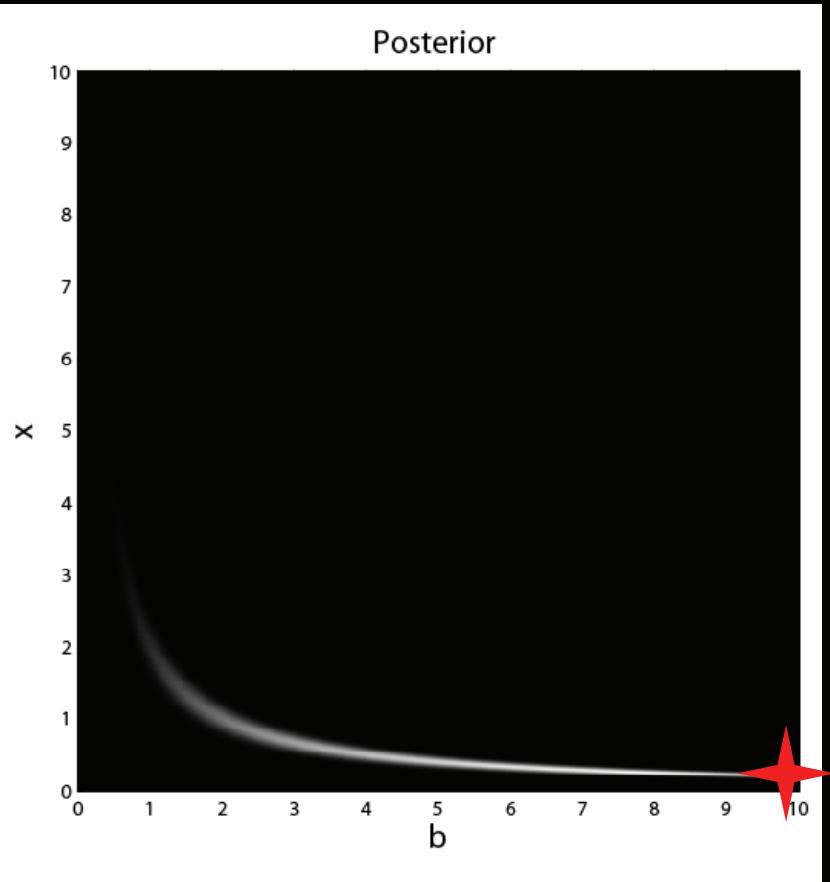
Marginal distribution $p(b|y)$

$$p(b|y) = \int p(b, x|y) dx = k \int p(y|b, x) p(x) dx$$



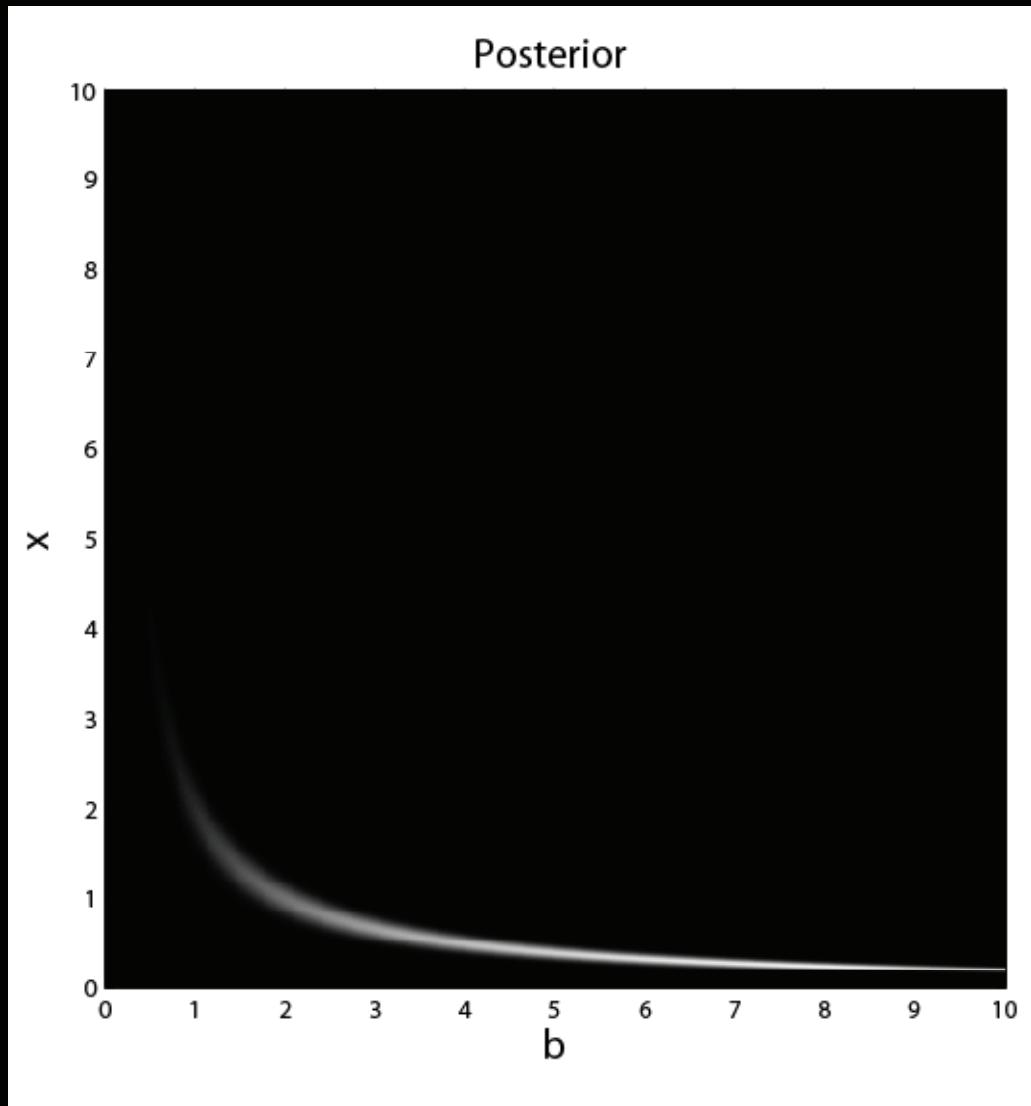
MAP solution

Highest point on surface: $\operatorname{argmax}_{b,x} p(x, b|y)$



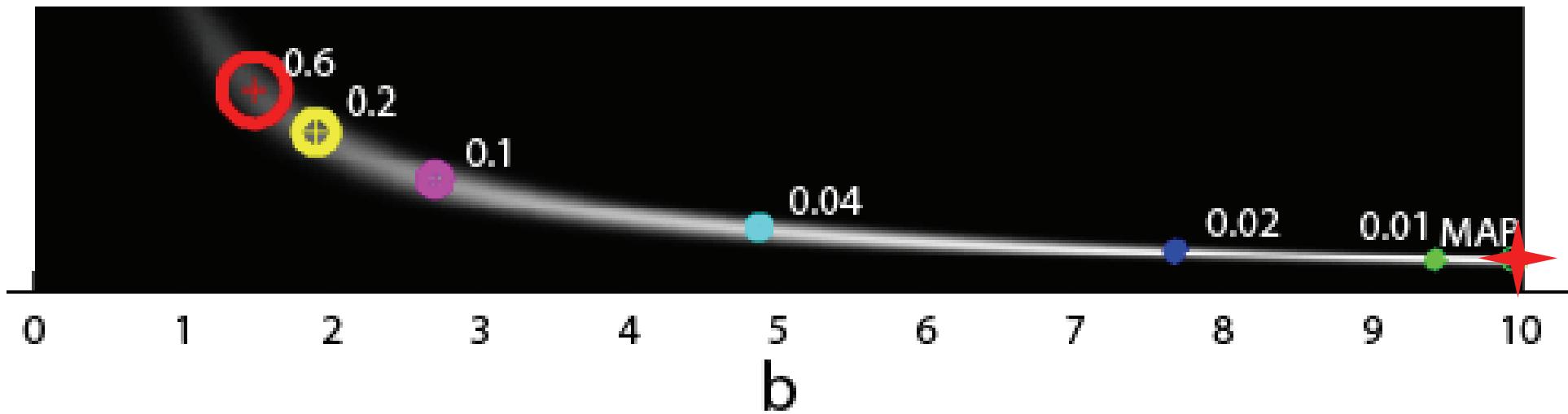
Variational Bayes

- True Bayesian approach not tractable
- Approximate posterior with simple distribution

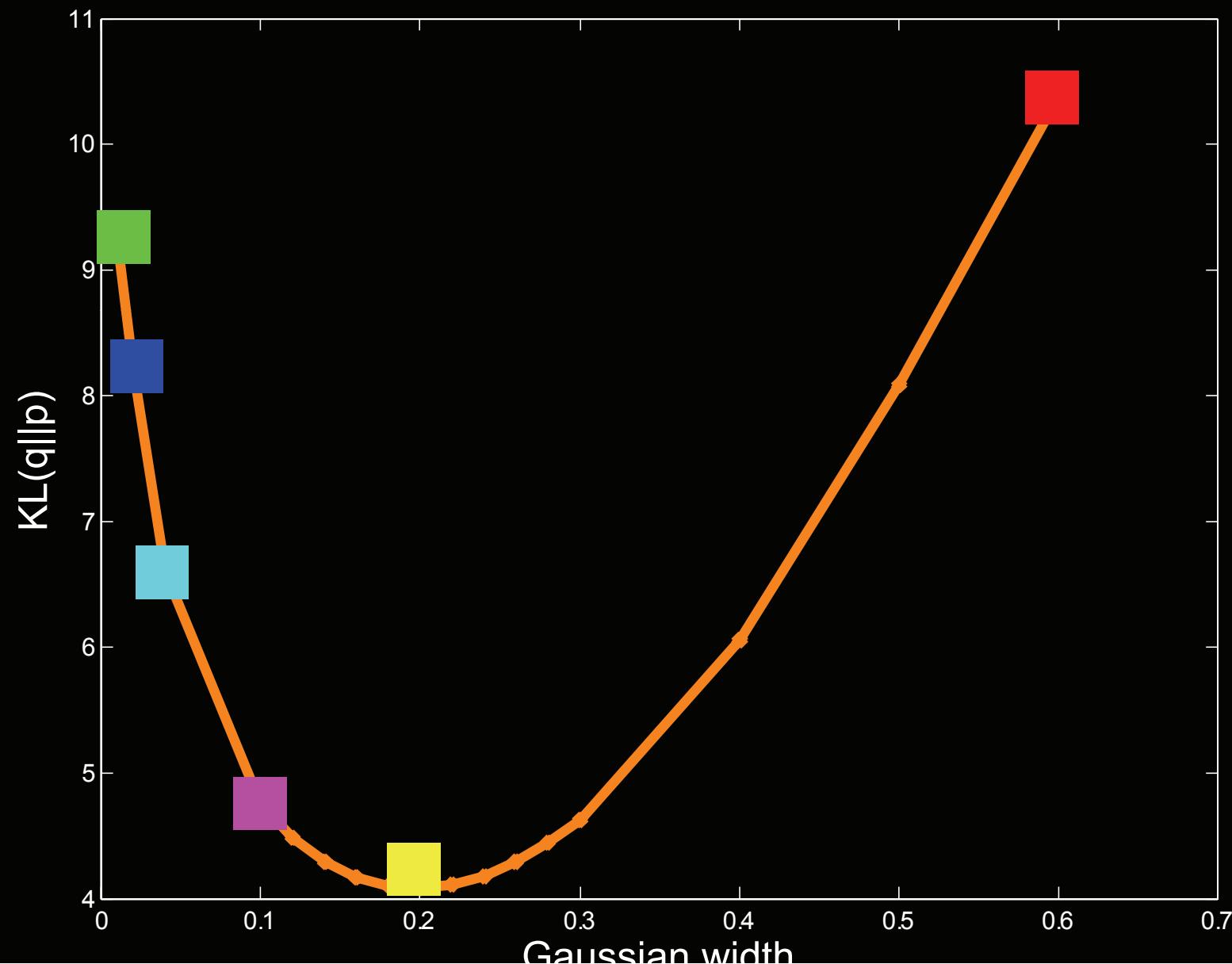


Fitting posterior with a Gaussian

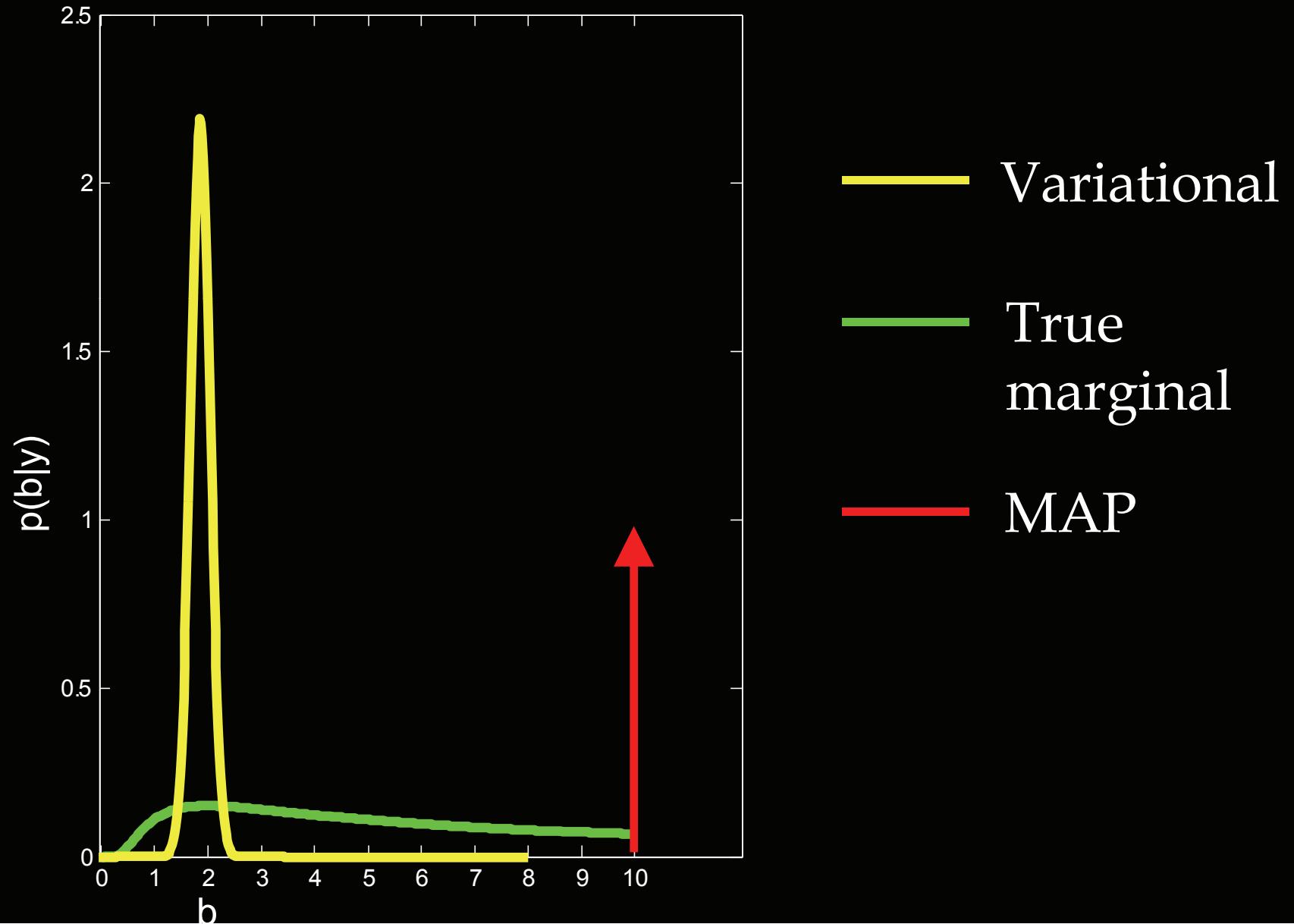
- Approximating distribution $q(x, b)$ is Gaussian
- Minimize $KL(q(x, b) \parallel p(x, b|y))$



KL-Distance vs Gaussian width



Variational Approximation of Marginal

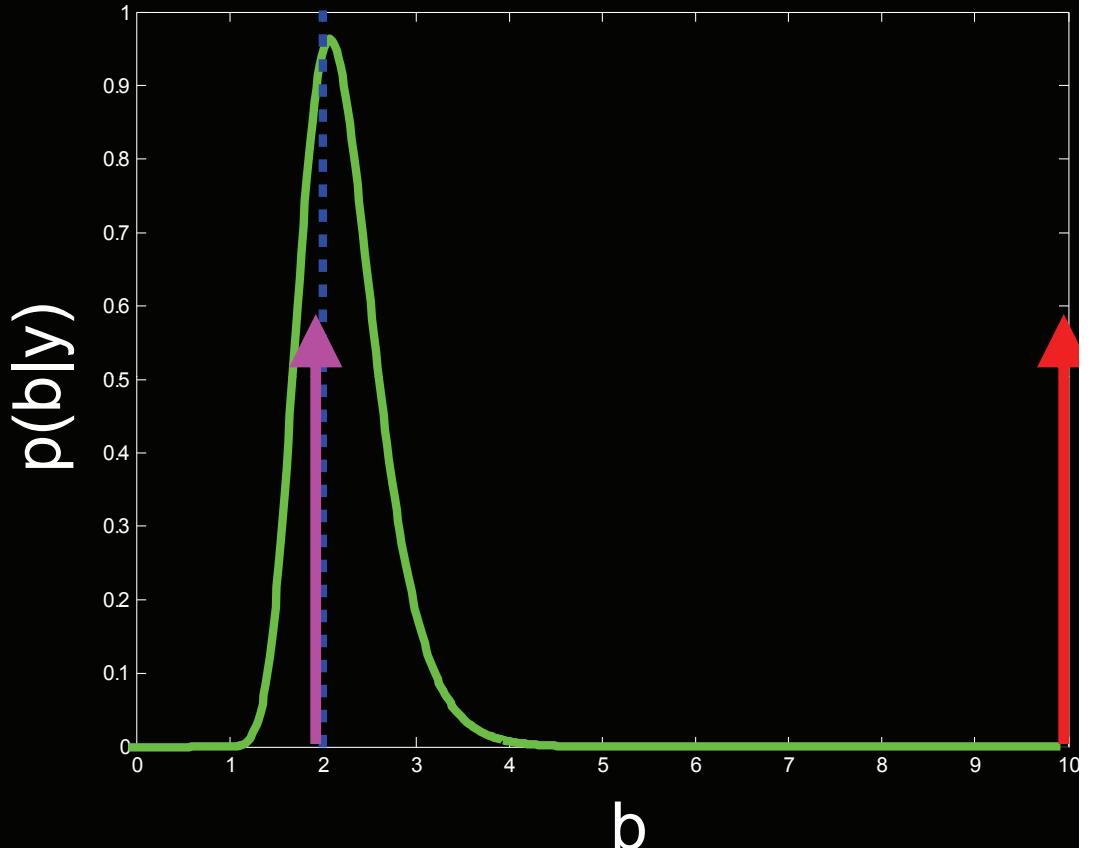


Try sampling from the model

Let true $b = 2$

Repeat:

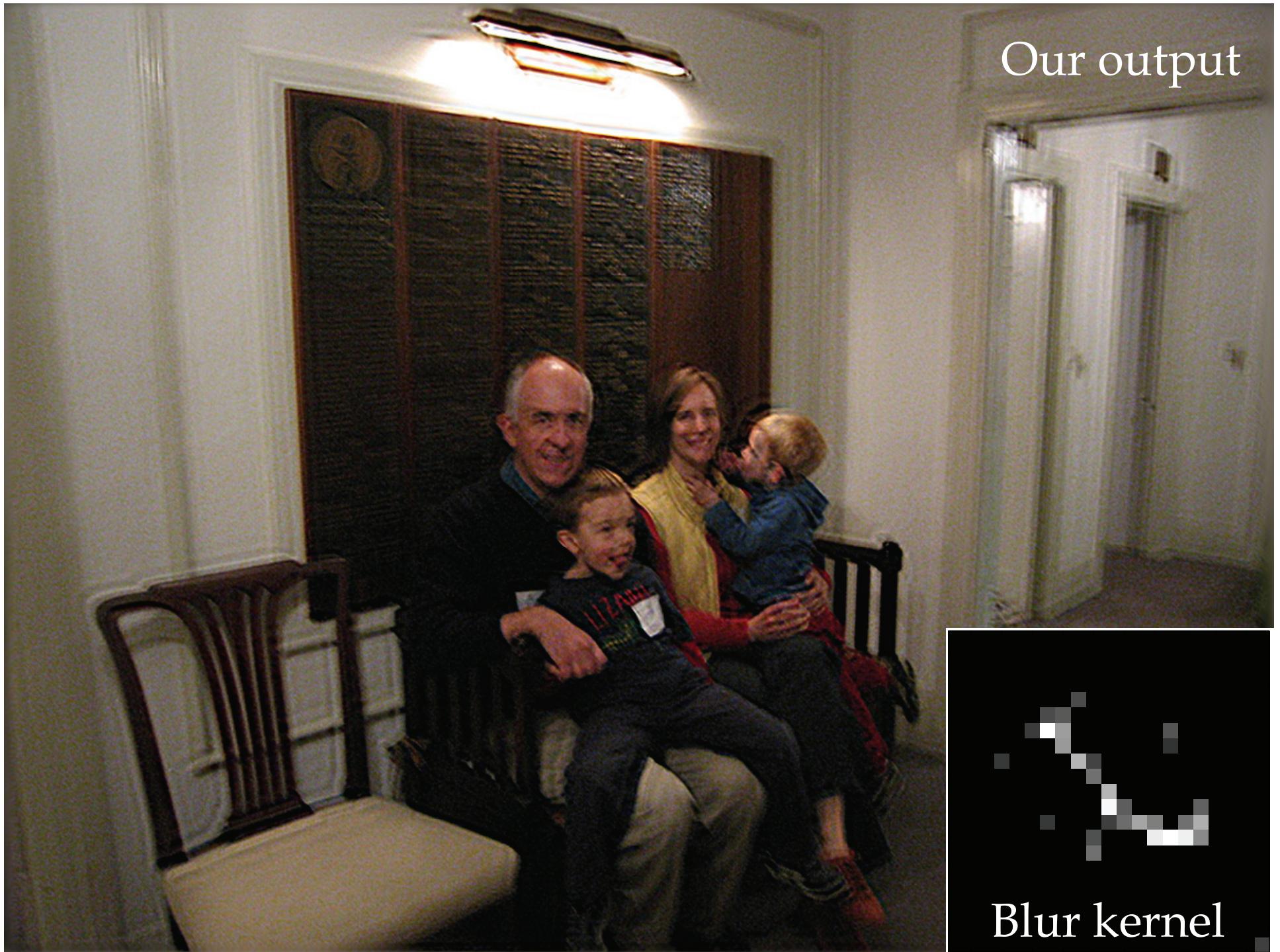
- Sample $x \sim N(0, 2)$
- Sample $n \sim N(0, \sigma^2)$
- $y = xb + n$
- Compute $p_{\text{MAP}}(b | y)$, $p_{\text{Bayes}}(b | y)$ & $p_{\text{Variational}}(b | y)$
- Multiply with existing density estimates (assume iid)



Original photograph



Our output



Blur kernel

Close-up of child

Original photograph



Our output

