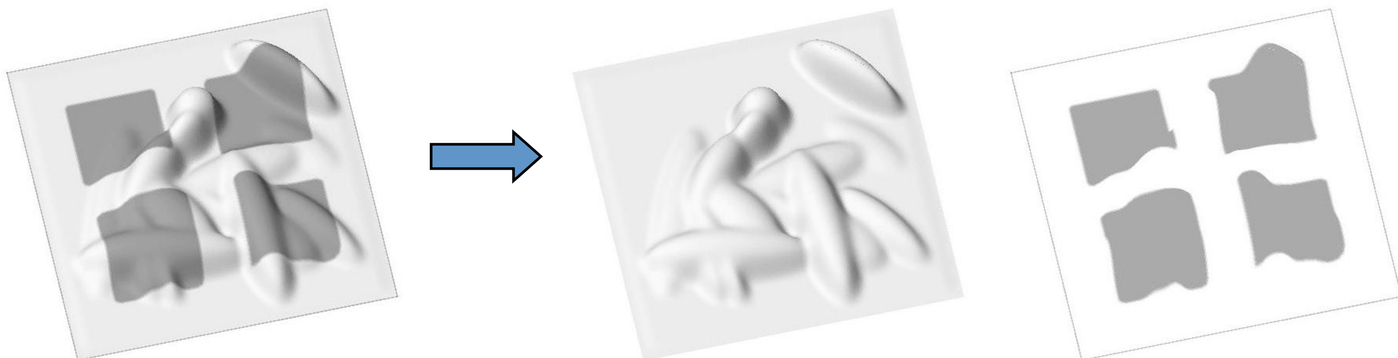
Lecture 17
Bayes

Solving "impossible" problems

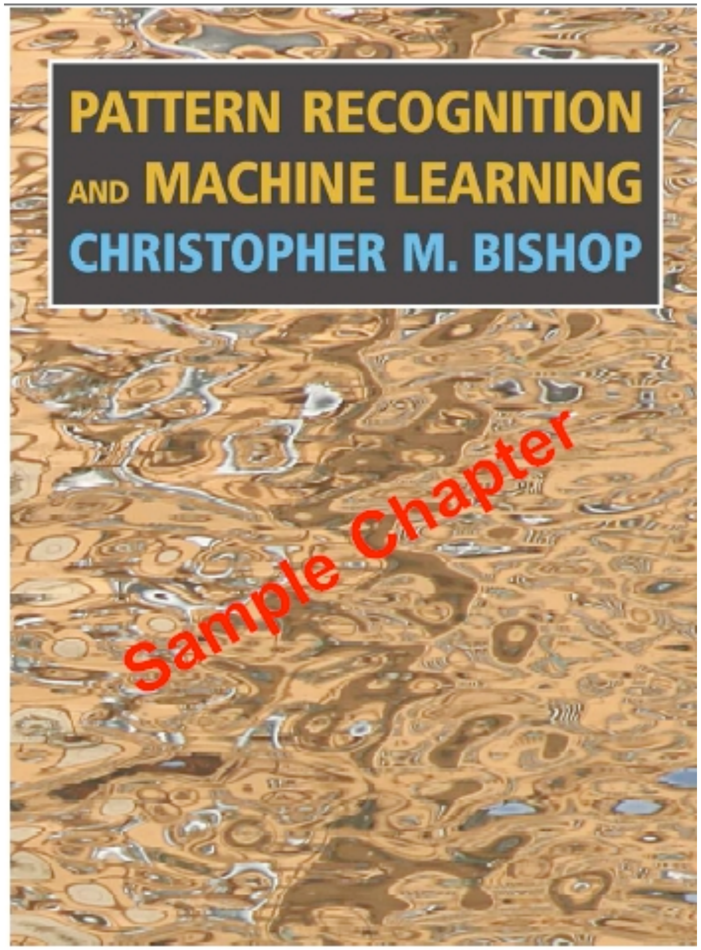


Image

Shading Image

Reflectance Image

Bayesian methods

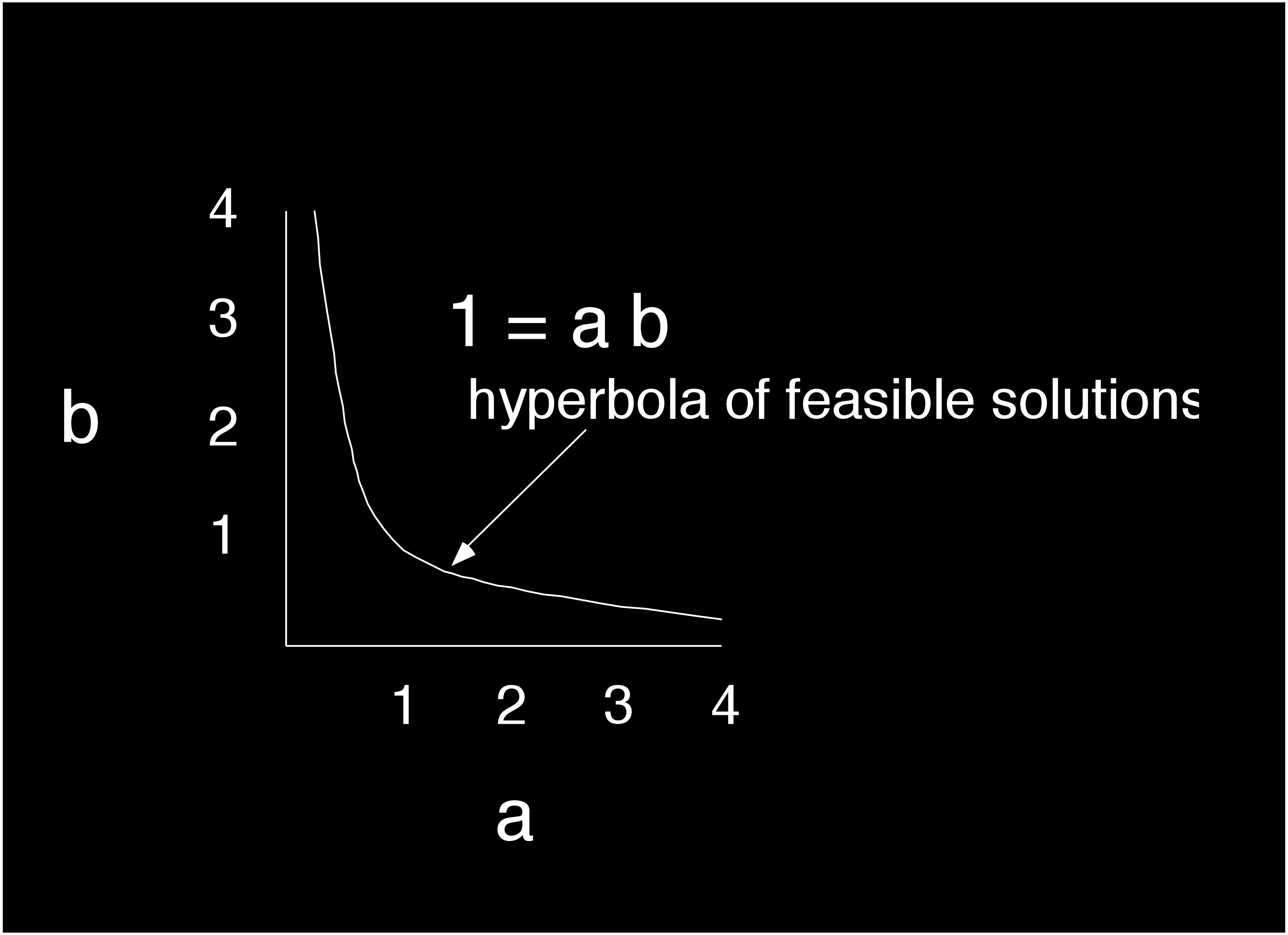


Chapter 8: Graphical models. (see also chapter 1 for nice introduction)
http://research.microsoft.com/~cmbishop/PRML/Bishop-PRML-sample.pdf

Simple, prototypical vision problem

- Observe some product of two numbers, say 1.0.
- What were those two numbers?
- Ie, $1 = ab$. Find a and b.

- Compare this with the prototypical graphics problem: here are two numbers; what is their product?


4
3
2
1
b
1 = a b
hyperbola of feasible solutions
1
2
3
4
a

Bayes rule

$$P(x|y) = P(y|x)\, P(x) \,/\, P(y)$$

Bayesian approach

- Want to calculate $P(a, b \mid y = 1)$.
- Use $P(a, b \mid y = 1) = k\, P(y=1 \mid a, b)\, P(a, b)$.

Likelihood function

Prior probability

Posterior probability

Likelihood function, P(obs|params)

- The forward model, or rendering model, taking into account observation noise.
- Example: assume Gaussian observation noise. Then for this problem:

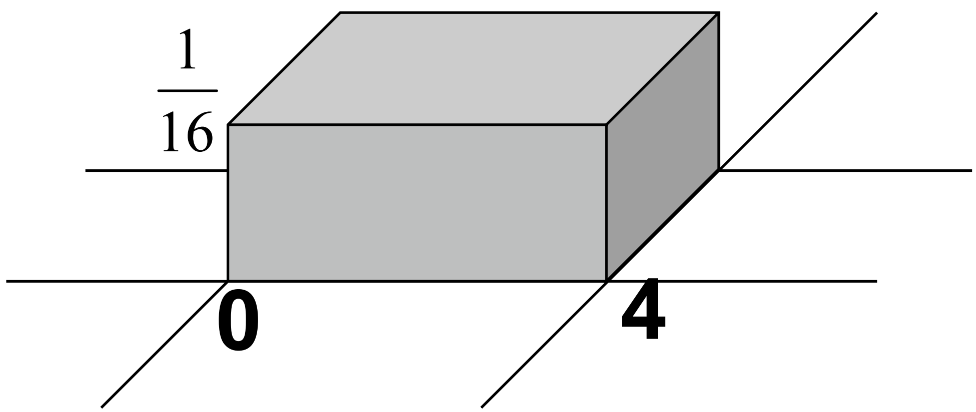
$$P(y=1 \mid a,b) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(1-ab)^2}{2\sigma^2}}$$

A common criticism of Bayesian methods

- "You need to make all those assumptions about prior probabilities".
- Response…?
- "Everyone makes assumptions. Bayesians put their assumptions out in the open, clearly stated, where they belong."

Prior probability

- In this case, we'll assume P(a,b)=P(a)P(b), and P(a) = P(b) = const., 0<a, b<4.



Posterior probability

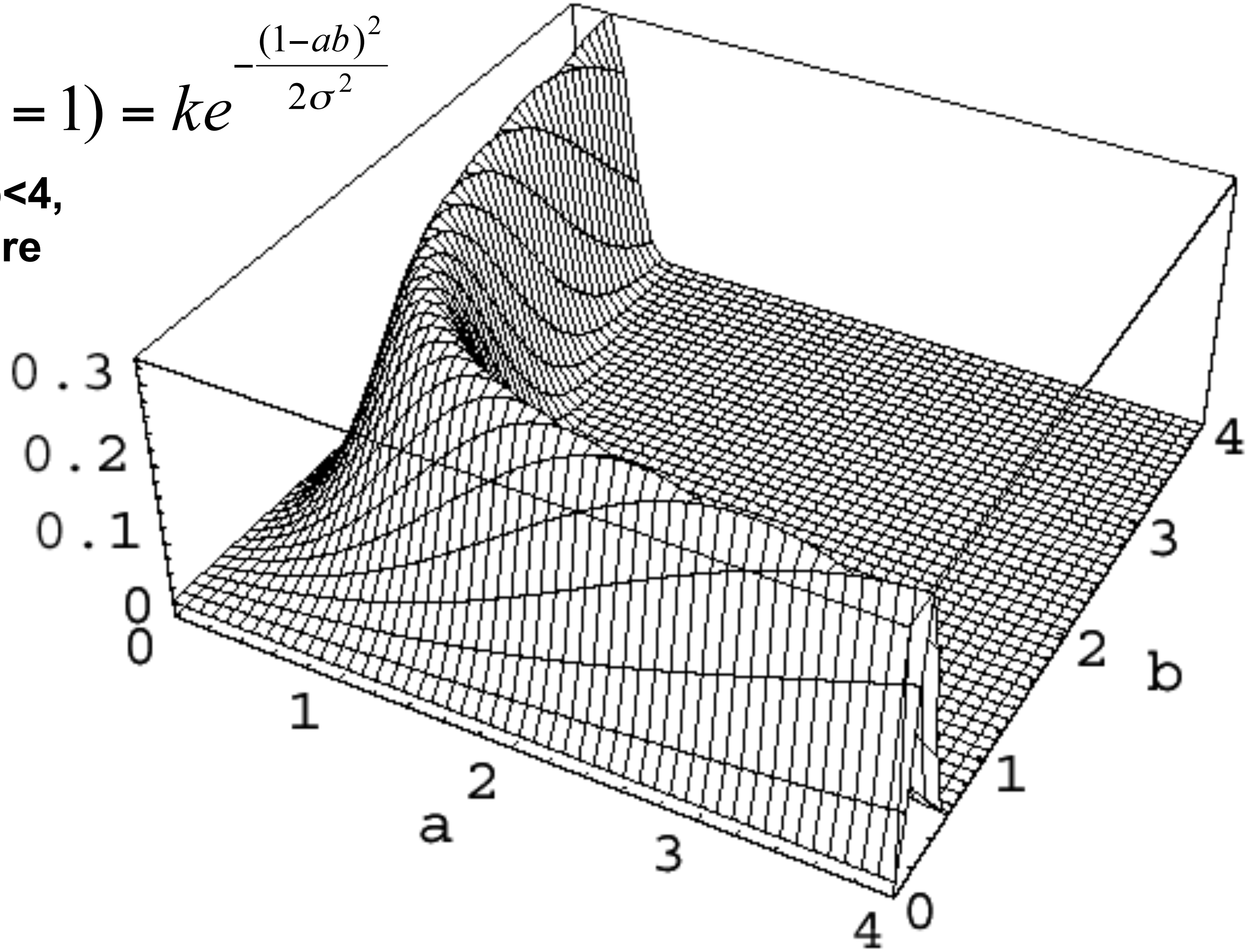
- Posterior = k likelihood prior

$$P(a,b \mid y=1) = ke^{-\frac{(1-ab)^2}{2\sigma^2}}$$

for 0 < a,b<4,
0 elsewhere

$$P(a,b \mid y=1) = ke^{-\frac{(1-ab)^2}{2\sigma^2}}$$

for 0 < a,b<4,
0 elsewhere

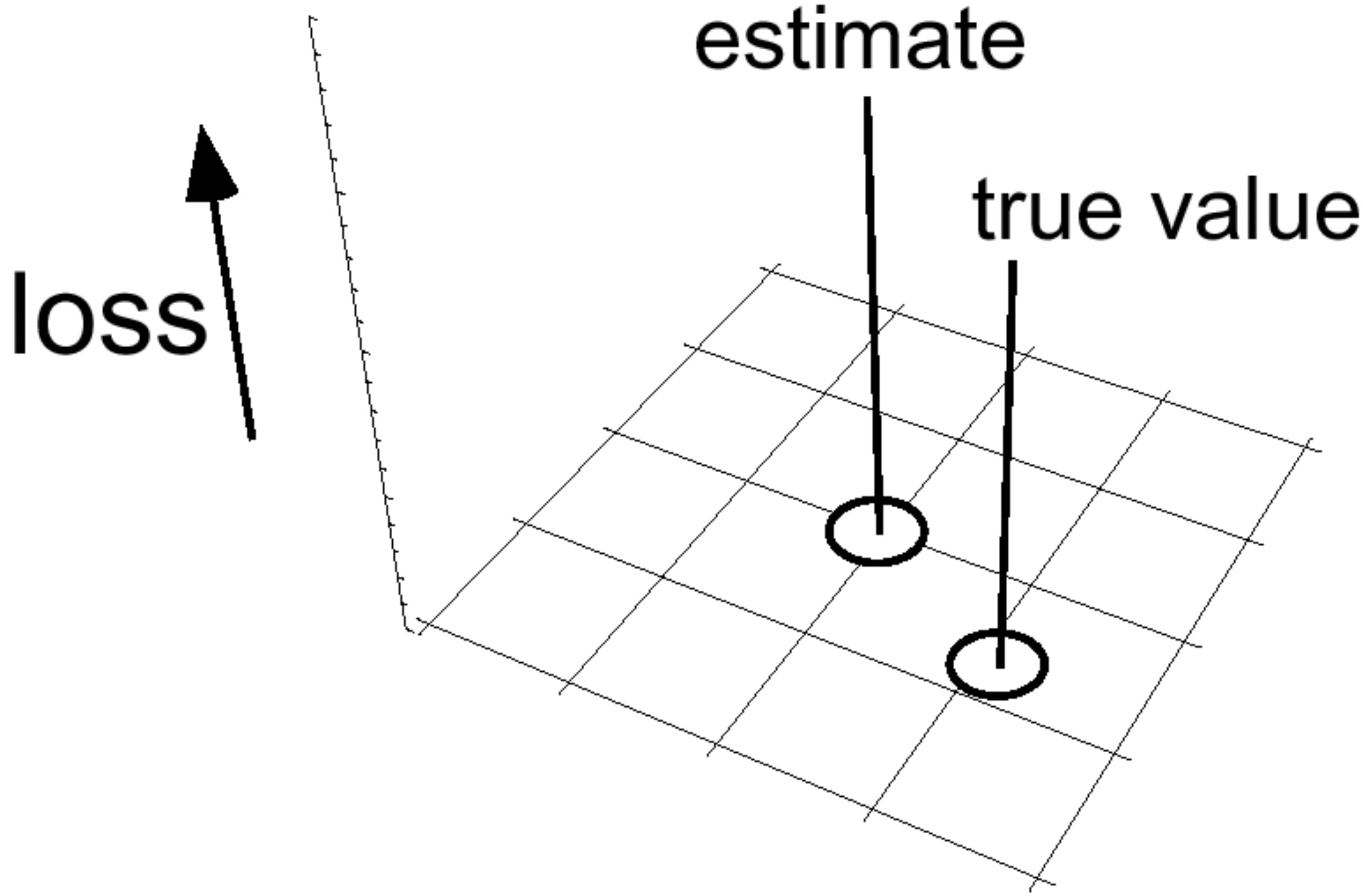


(a) Posterior Probability

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997

Loss functions

How important is each error?


estimate
true value
loss

Bayesian decision theory

parameter variable, $\mathbf{z}$. A loss function $L(\mathbf{z}, \tilde{\mathbf{z}})$ specifies the penalty for estimating $\tilde{\mathbf{z}}$ when the true value is $\mathbf{z}$. Knowing the posterior probability, one can select the parameter values which loss for a particular loss function:

$$\begin{aligned} \text{[expected loss]} &= \int \text{[posterior] [loss function] d [parameters]} \\ R(\tilde{\mathbf{z}}|\mathbf{y}) &= -C \int [\exp[-\frac{\mathcal{T}}{2\sigma^2}\|\mathbf{y} - \mathbf{f}(\mathbf{z})\|^2]\, \mathbf{P_z}(\mathbf{z})] \quad L(\mathbf{z}, \tilde{\mathbf{z}}) \quad d\mathbf{z}, \end{aligned} \tag{21}$$

where we have substituted from Bayes' rule, Eq. (4), and the noise model, Eq. (3). The optimal estimate is the parameter $\tilde{\mathbf{z}}$ of minimum risk.

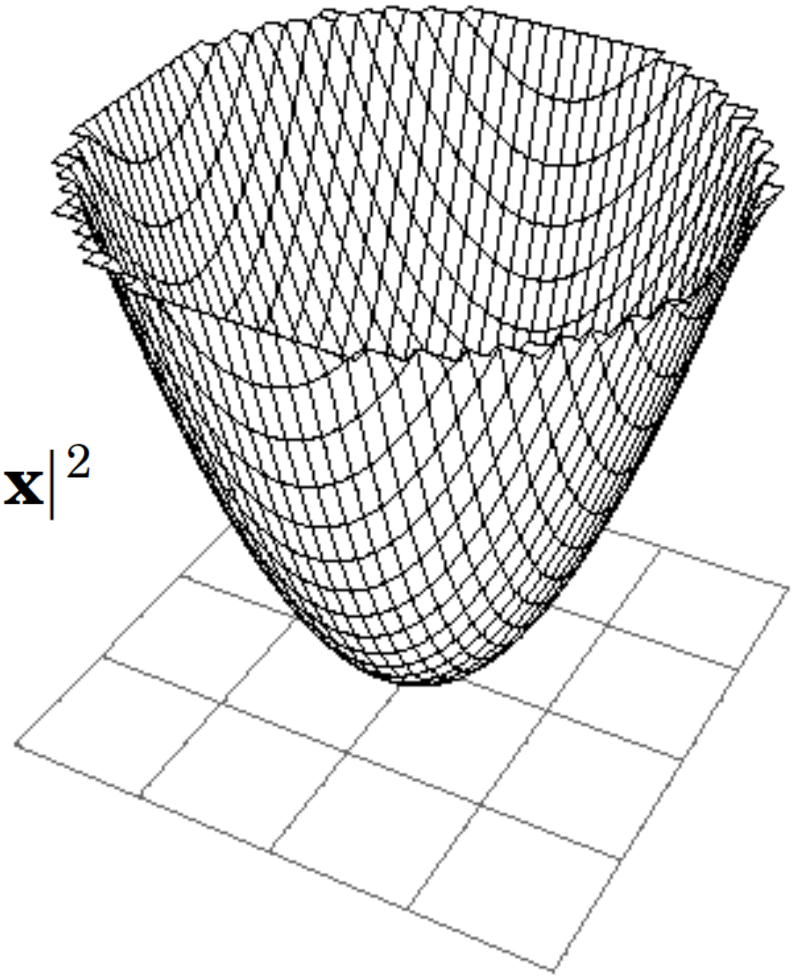
Note: if L(z,z') has the form L(z-z') then the expected loss is a convolution

$$R(z \mid y) = P(z \mid y) \otimes L(z)$$

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997

Minimum mean-squared-error (MMSE)

$$L(\tilde{\mathbf{x}},\ \mathbf{x}) = |\tilde{\mathbf{x}} - \mathbf{x}|^2$$

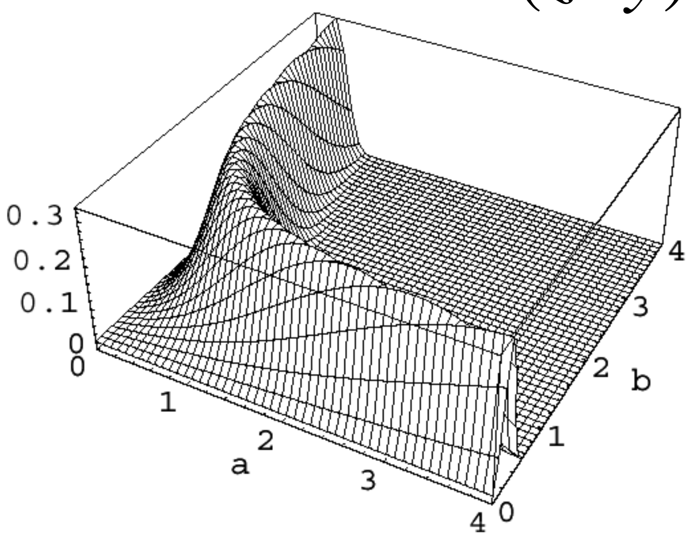


(b) MMSE loss fn.

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997

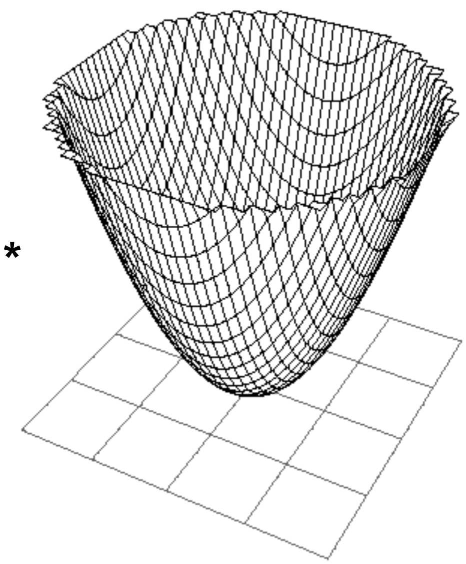
Ab = 1 problem

$P(z \mid y) \otimes L(z)$ $\qquad$ $R(z \mid y)$



(a) Posterior Probability

*

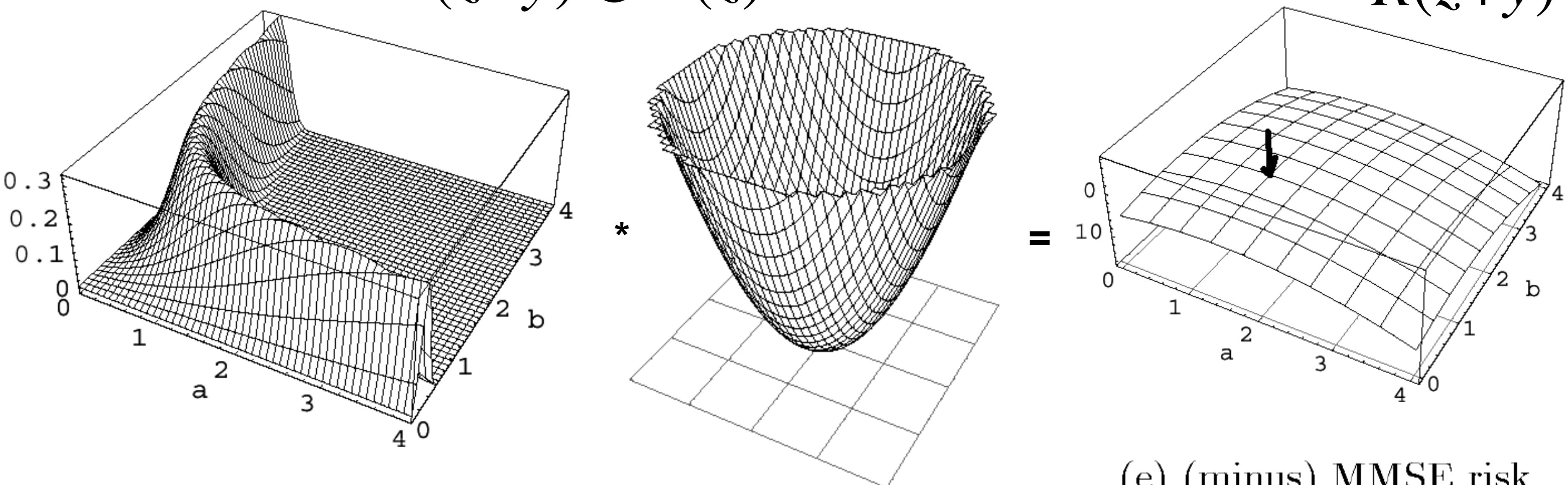


(b) MMSE loss fn.

=

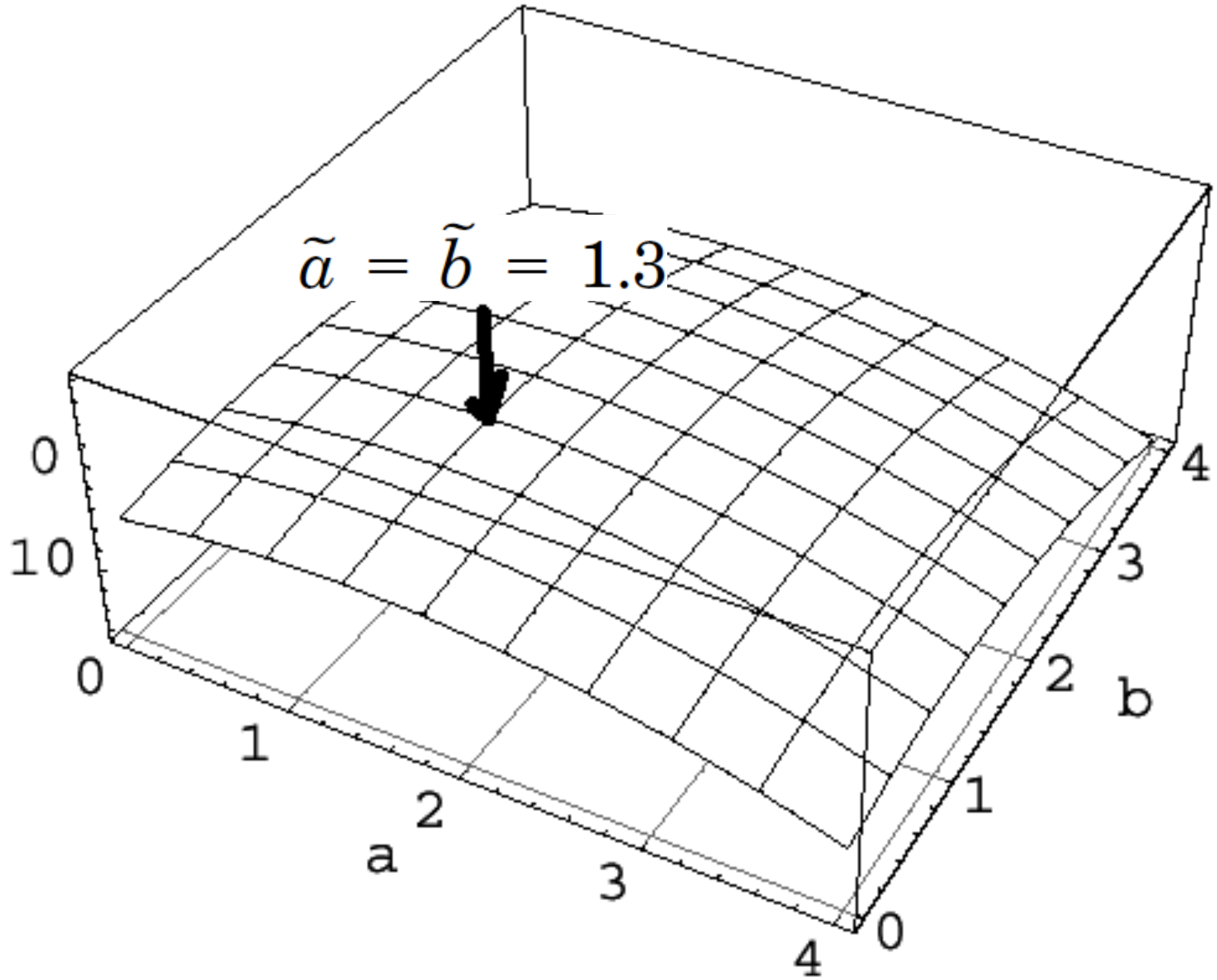
Ab = 1 problem

$P(z \mid y) \otimes L(z)$ $R(z \mid y)$



(a) Posterior Probability (b) MMSE loss fn. (e) (minus) MMSE risk

And the solution to ab = 1 is ….



(e) (minus) MMSE risk

The minimum of the squared error loss is the center of probability mass

Let $P_z(z)$ be the posterior probability.

The minimum of the expected loss will satisfy

$$\frac{\partial}{\partial \hat{z}} \int P_z(z)\ (z-\hat{z})^2\, dz = 0$$

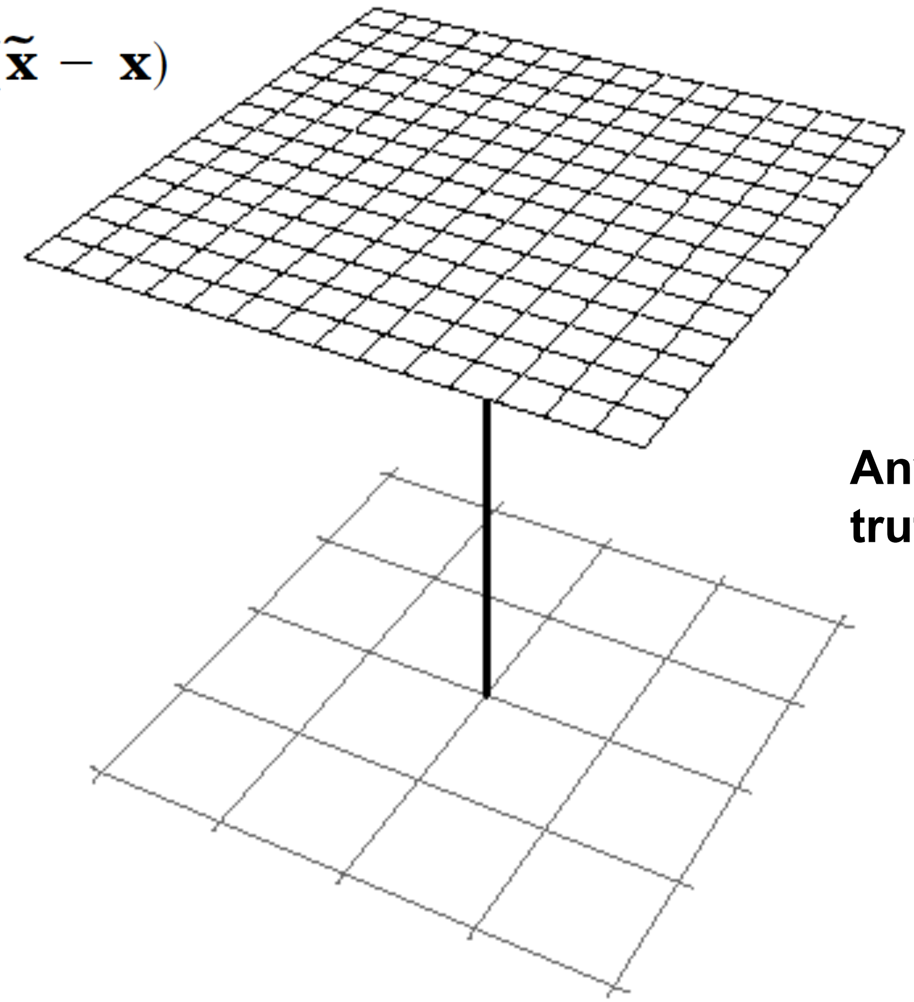
Differentiating, we have

$$\int P_z(z)\ (z-\hat{z})\ dz = 0$$

$$\int P_z(z)\ z\ dz = \hat{z}$$

This is E(z)

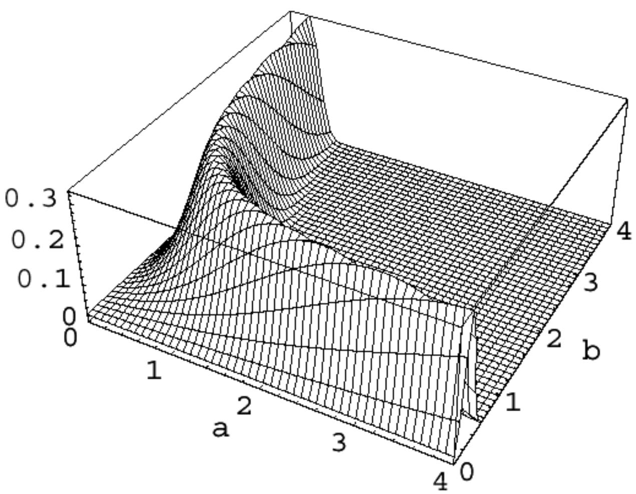
$$L(\tilde{\mathbf{x}}, \mathbf{x}) = -\delta(\tilde{\mathbf{x}} - \mathbf{x})$$



Any deviation from the truth is equally bad

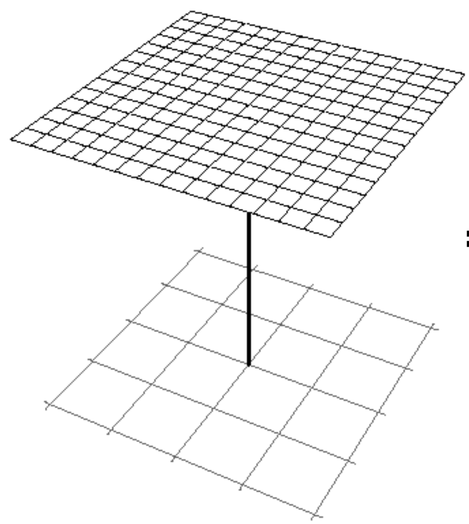
(a) MAP loss fn.

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997



(a) Posterior Probability

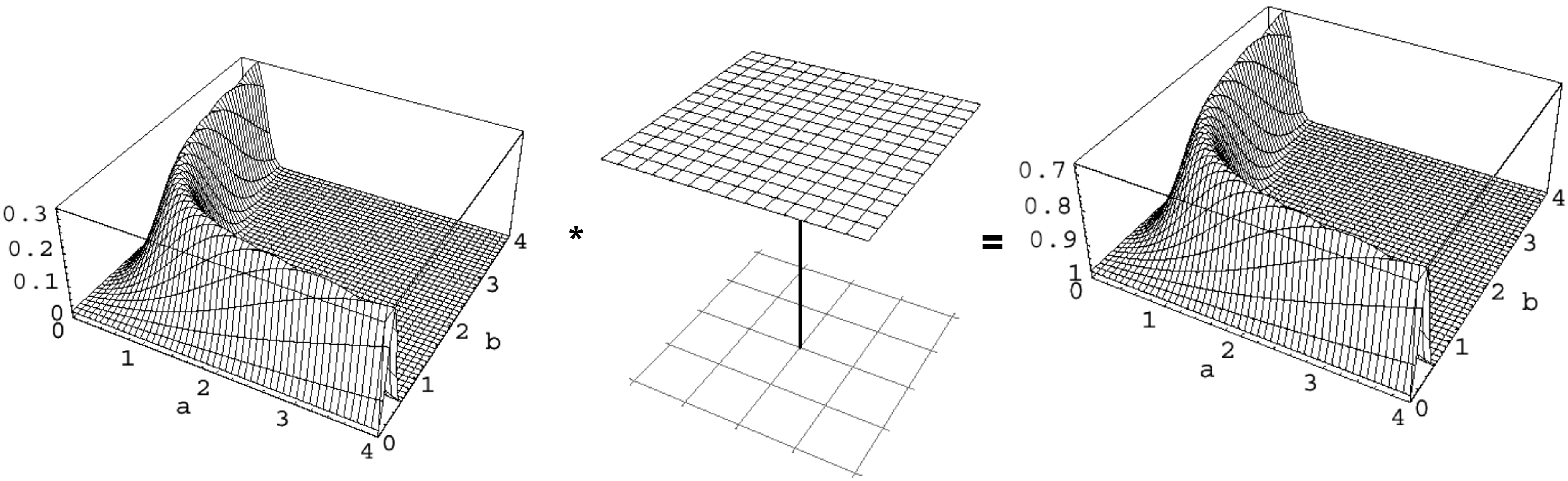
*



(a) MAP loss fn.

=

D. H. Brainard and W. T. Freeman, ***Bayesian Color Constancy***, **Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997**



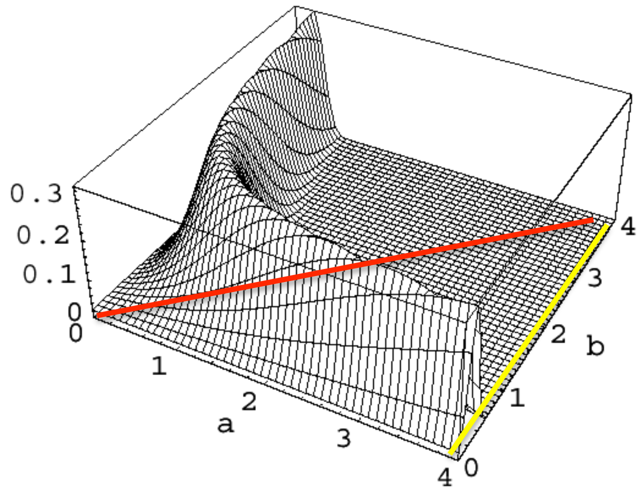
(a) Posterior Probability

(a) MAP loss fn.

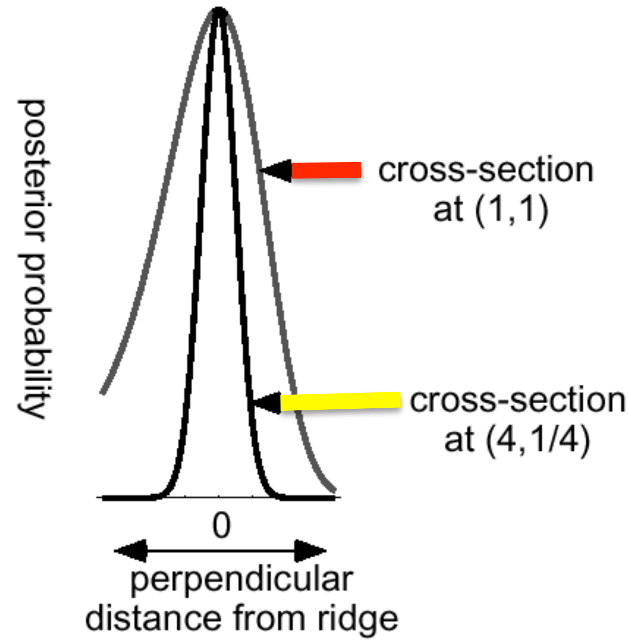
(d) (minus) MAP risk

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997

And the solution is?



(a) Posterior Probability

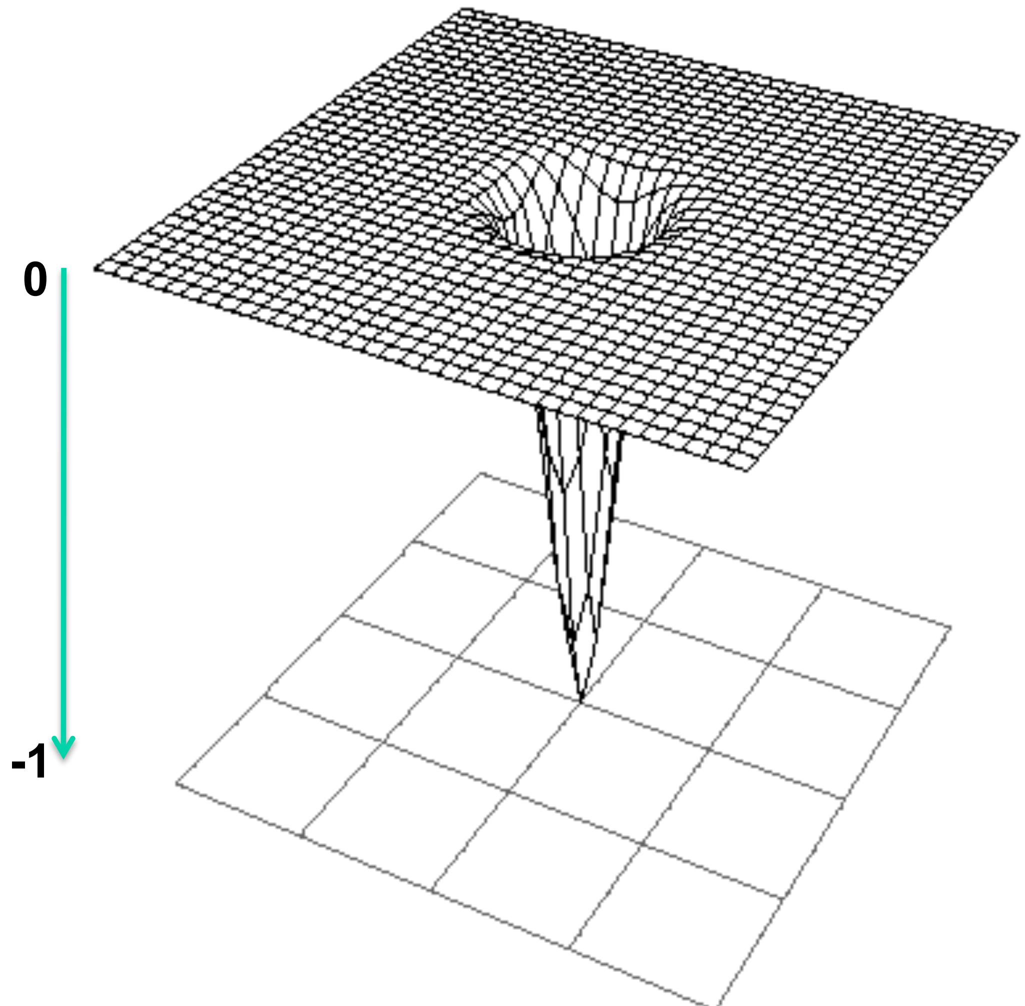


(b) Ridge Thickness Variations

Local mass loss function may be useful model for perceptual tasks

http://sportsillustrated.cnn.com/baseball/college/2000/college_world_series/news/2000/06/15/cws_notebook_ap/t1_borchard_ap_

$$L(\tilde{\mathbf{x}}, \mathbf{x}) = -\exp[-|\mathbf{K}_L{}^{-1/2}(\tilde{\mathbf{x}} - \mathbf{x})|^2]$$



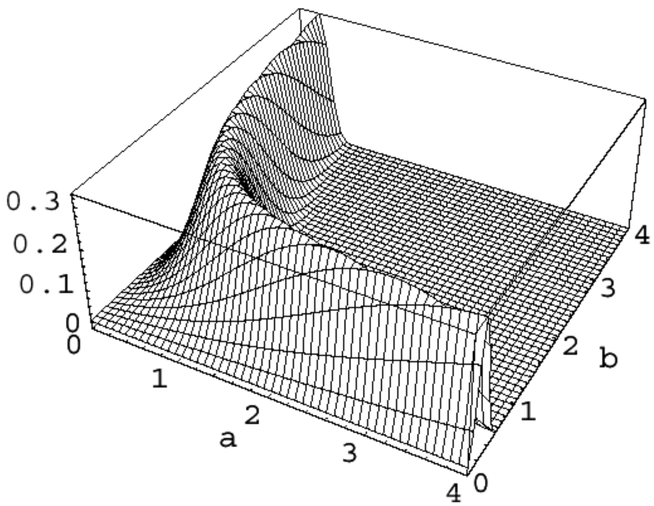
Maximum local mass (MLM)

(c) MLM loss fn.

D. H. Brainard and W. T. Freeman, *Bayesian Color Constancy*, Journal of the Optical Society of America, A, 14(7), pp. 1393-1411, July, 1997

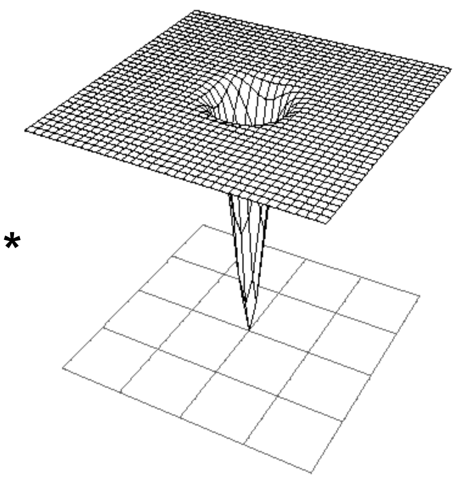
Maximum local mass (MLM) estimator

Ab = 1 problem



(a) Posterior Probability

*

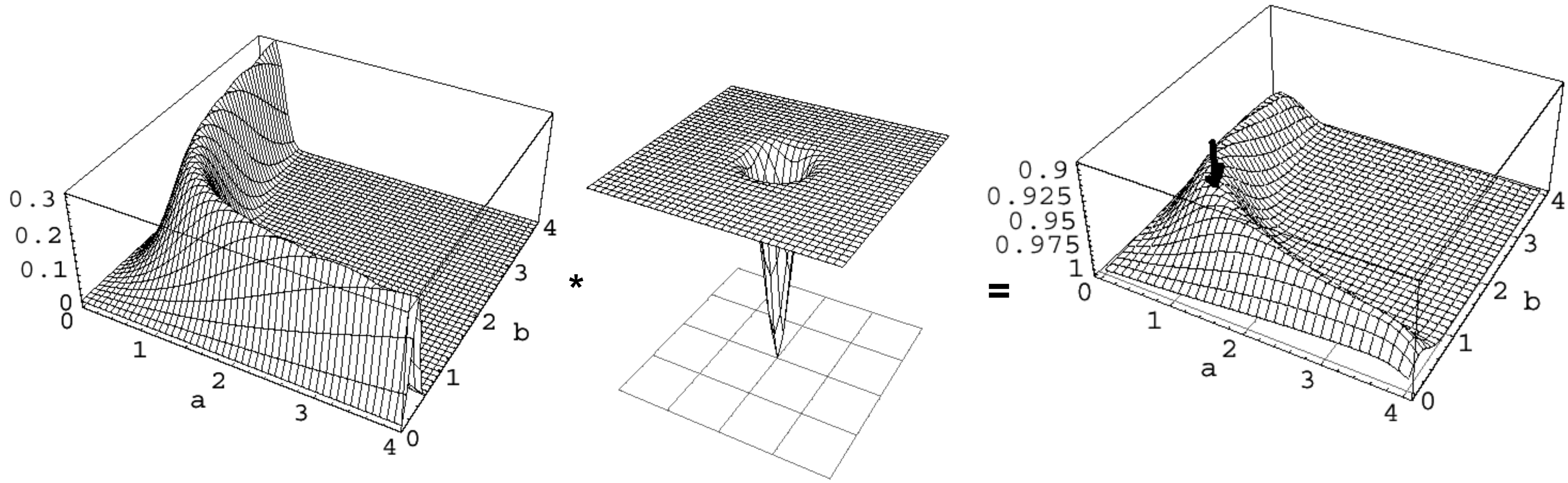


(c) MLM loss fn.

=

Maximum local mass (MLM) estimator

Ab = 1 problem



(a) Posterior Probability

(c) MLM loss fn.

(f) (minus) MLM risk

And the solution is

a=b=1

Regularization vs Bayesian interpretations

Regularization: minimize

$$(1-ab)^2 + \lambda(a^2 + b^2)$$

Bayes: maximize

$$e^{-\frac{(1-ab)^2}{2\sigma^2}} \, e^{-\gamma(a^2+b^2)}$$

likelihood **prior**

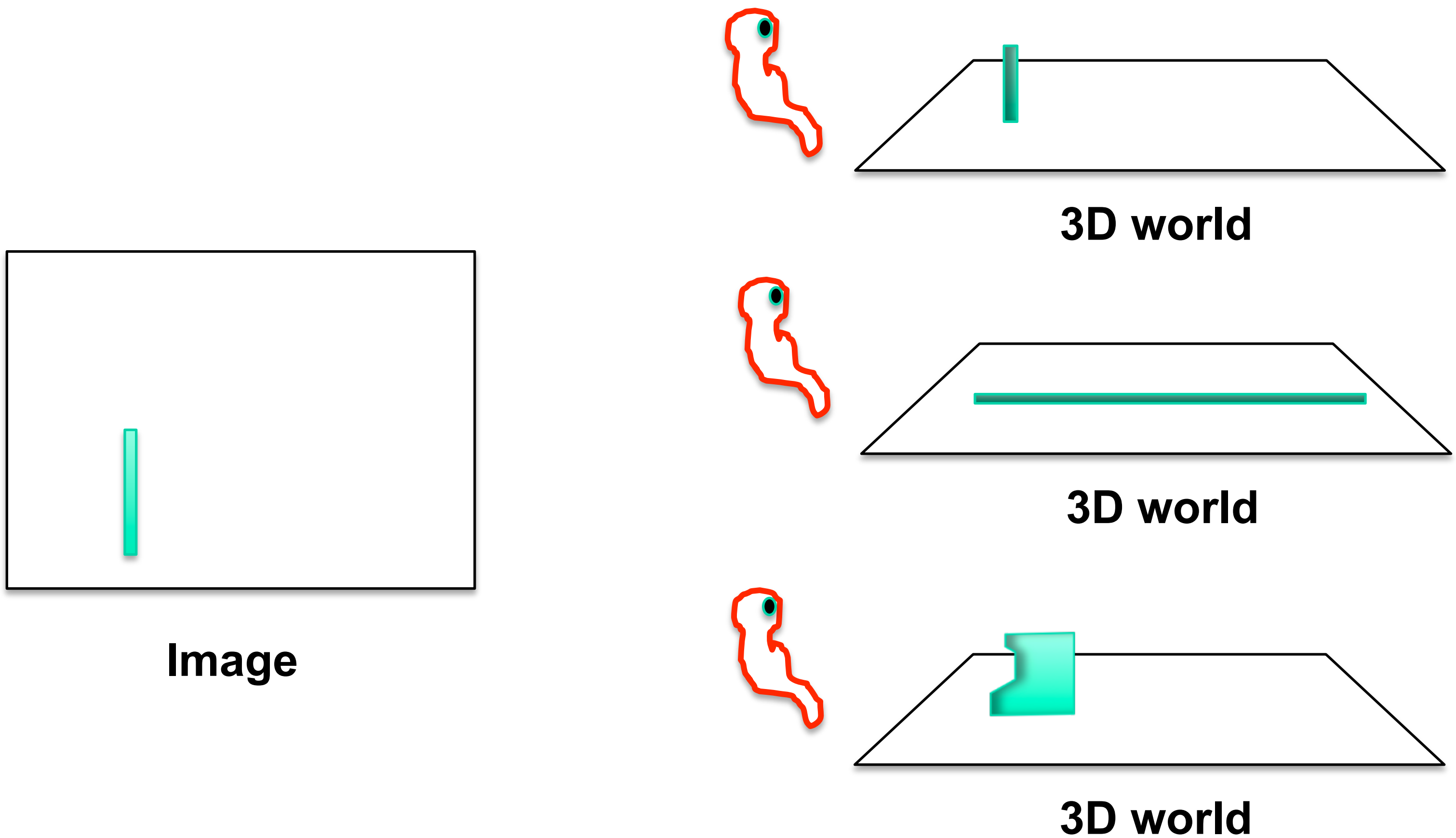
Bayesian interpretation of regularization approach

- For this example:
  - Assumes Gaussian random noise added before observation
  - Assumes a particular prior probability on a, b.
  - Uses MAP estimator (assumes delta fn loss).

Why the difference matters

- Know what the things mean
- Speak with other modalities in language of probability
- Loss function
- Bayes also offers principled ways to choose between different models.

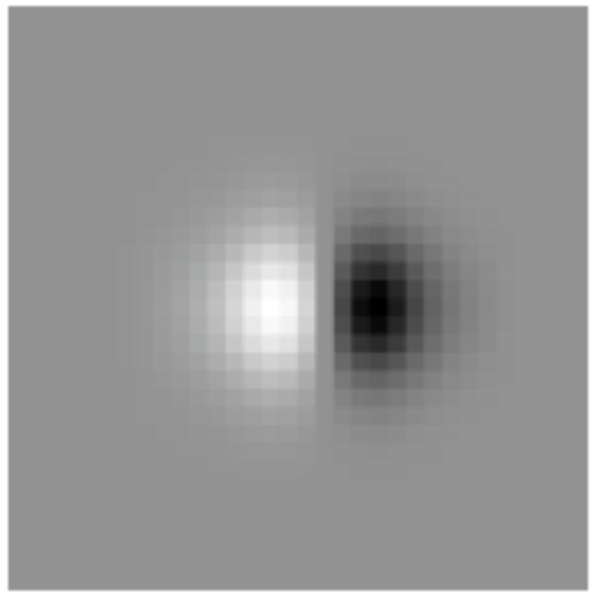
Generic view assumption



Generic view assumption: the observer should not assume that he has a special position in the world… The most generic interpretation is to see a vertical line as a vertical line in 3D.

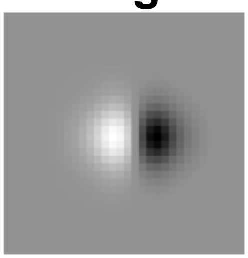
Freeman, 93

Example image

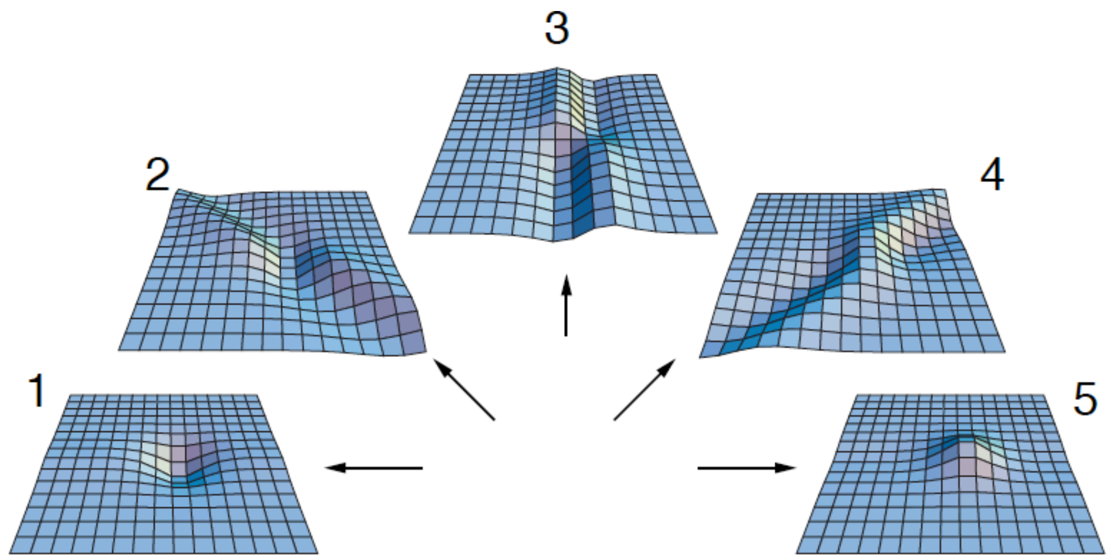


Multiple shape+illumination explanations

Image



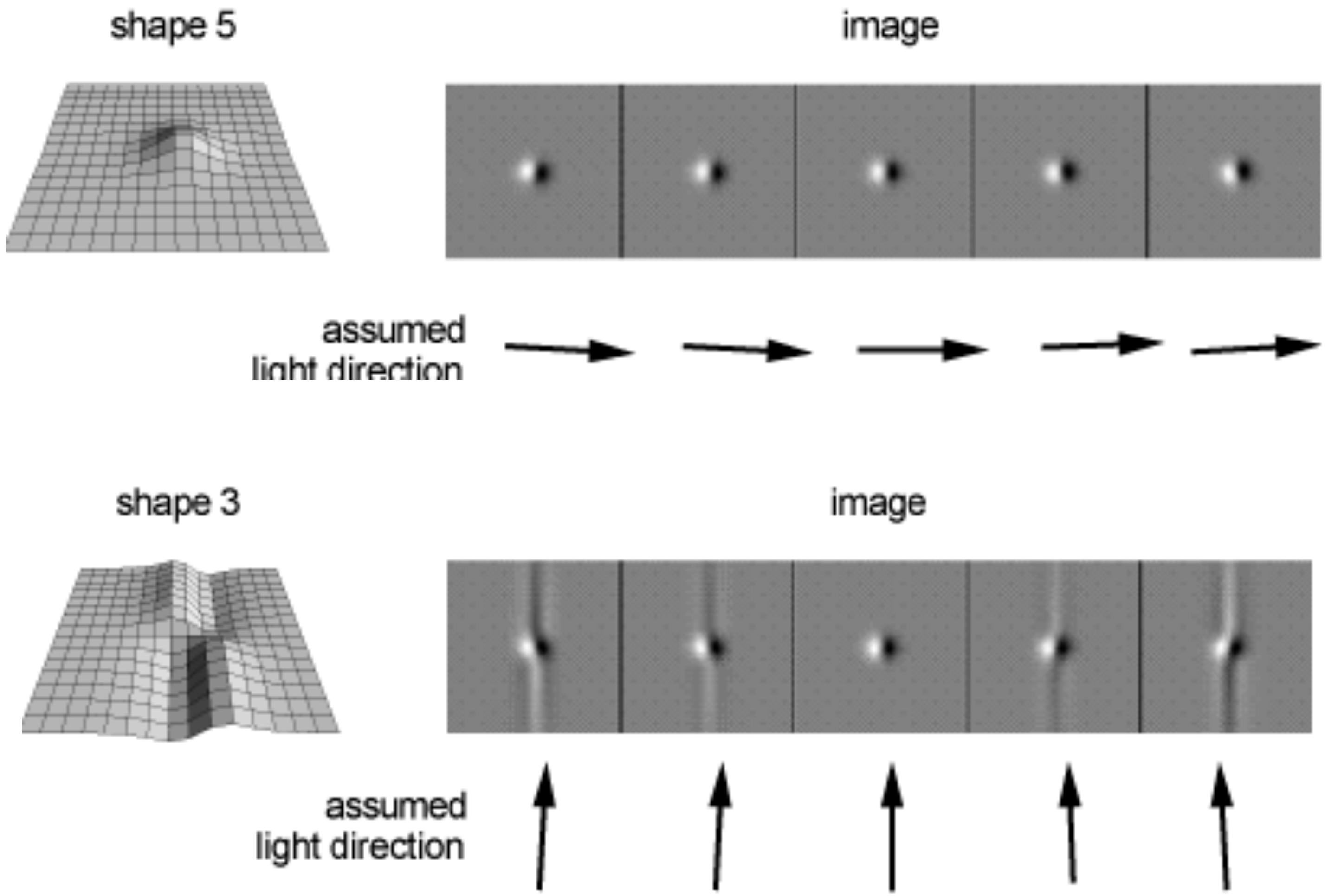
?



shapes for different assumed light directions

W. T. Freeman, *The generic viewpoint assumption in a framework for visual perception*, Nature, vol. 368, p. 542 - 545, April 7, 1994.

Generic shape interpretations render to the image over a range of light directions



W. T. Freeman, *The generic viewpoint assumption in a framework for visual perception*, Nature, vol. 368, p. 542 - 545, April 7, 1994.

Loss function

$$L(s,\theta \mid y) = \int P(s',\theta' \mid y)\, l(s,\theta,s',\theta')\, ds' s\theta'$$

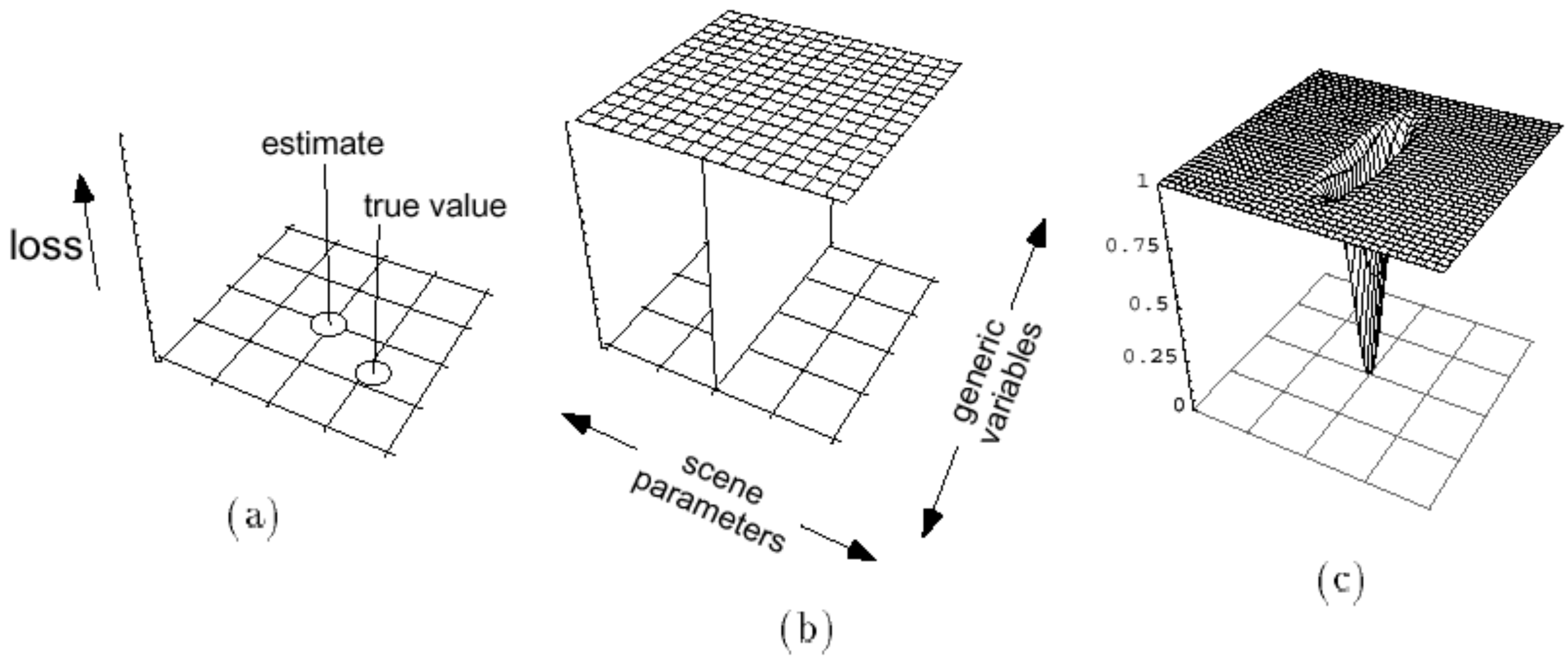
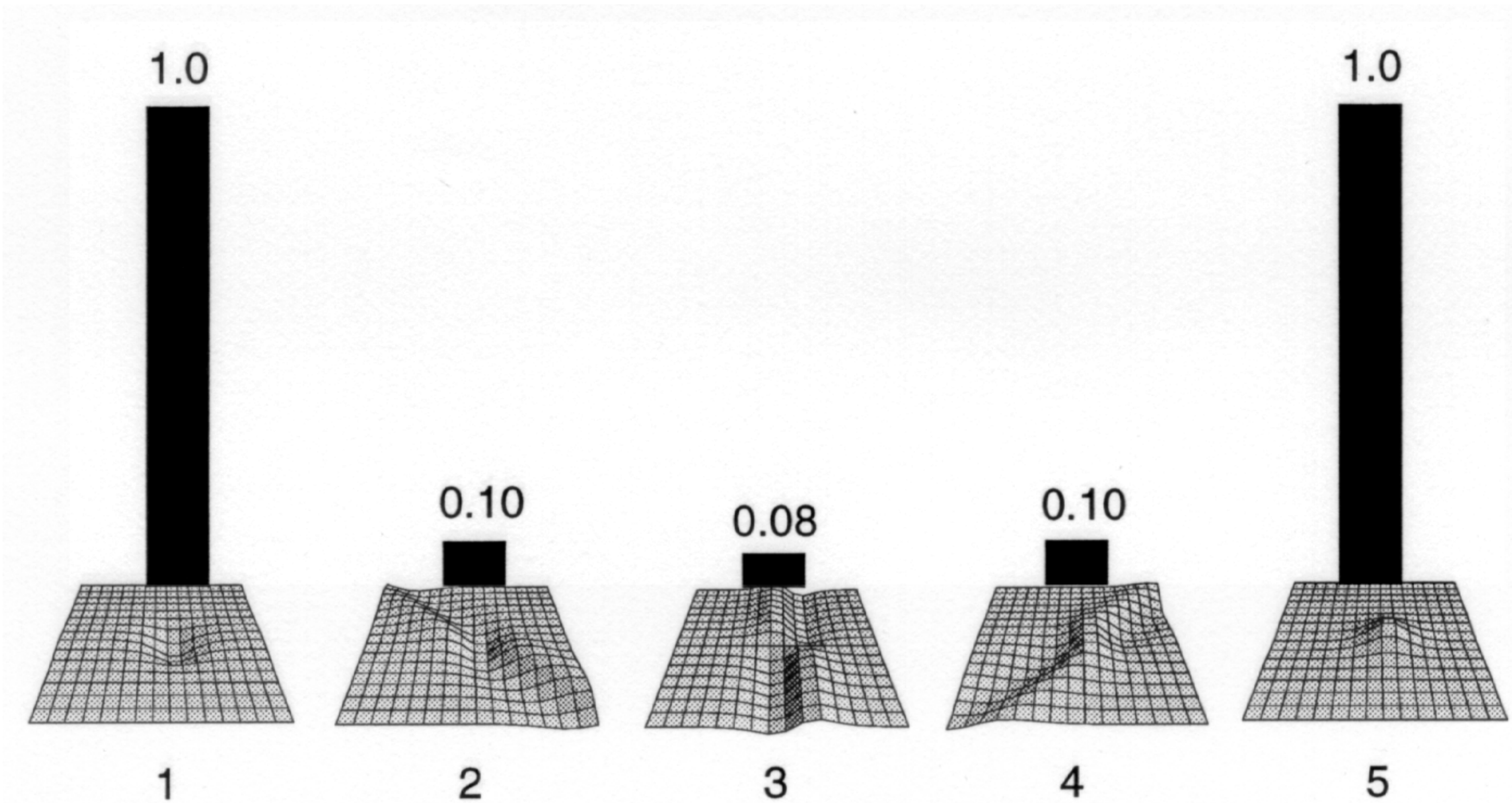


Figure 10: Loss function interpretation of generic viewpoint assumption. (a) shows the general form for a shift invariant loss function. The function $L(\mathbf{z}, \tilde{\mathbf{z}})$ describes the penalty for guessing the parameter $\tilde{\mathbf{z}}$ when the acutal value was $\mathbf{z}$. The marginalization over generic variables of Eq. (5) followed by MAP estimation is equivalent to using the loss function of (b). (c) Shows another possible form for the loss function, discussed in [11, 23, 24, 65].

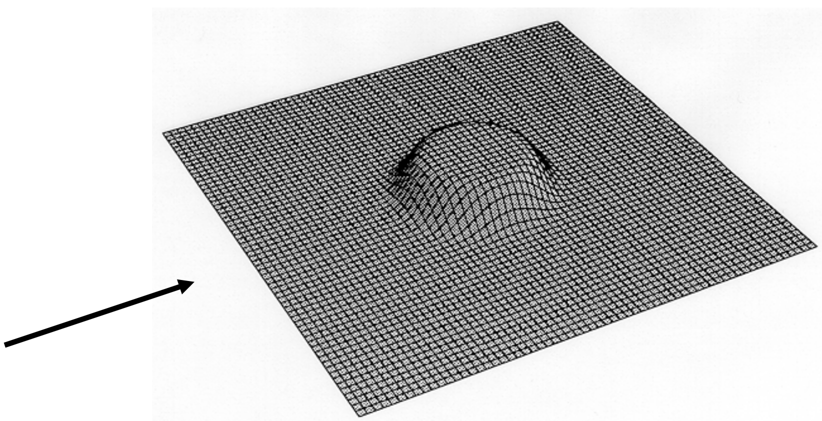
W. T. Freeman, ***The generic viewpoint assumption in a framework for visual perception***, **Nature, vol. 368, p. 542 - 545, April 7, 1994.**

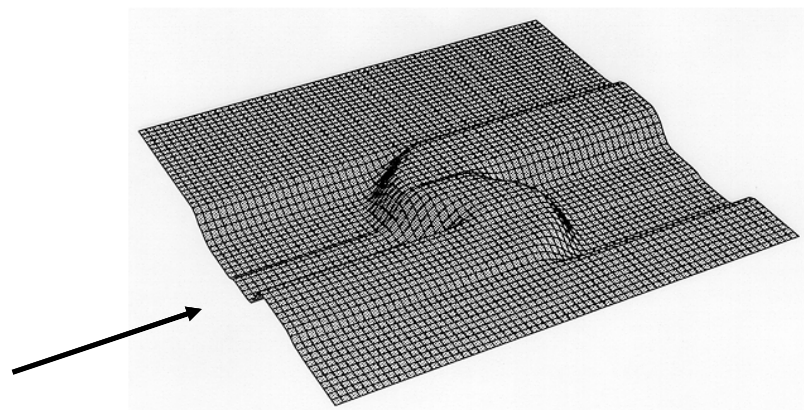
Shape probabilities



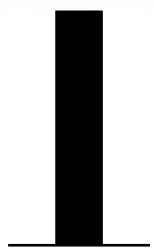
W. T. Freeman, *The generic viewpoint assumption in a framework for visual perception*, Nature, vol. 368, p. 542 - 545, April 7, 1994.

Comparison of shape explanations





- Lighting "genericity" of the shape explanation:



3.8

0.48

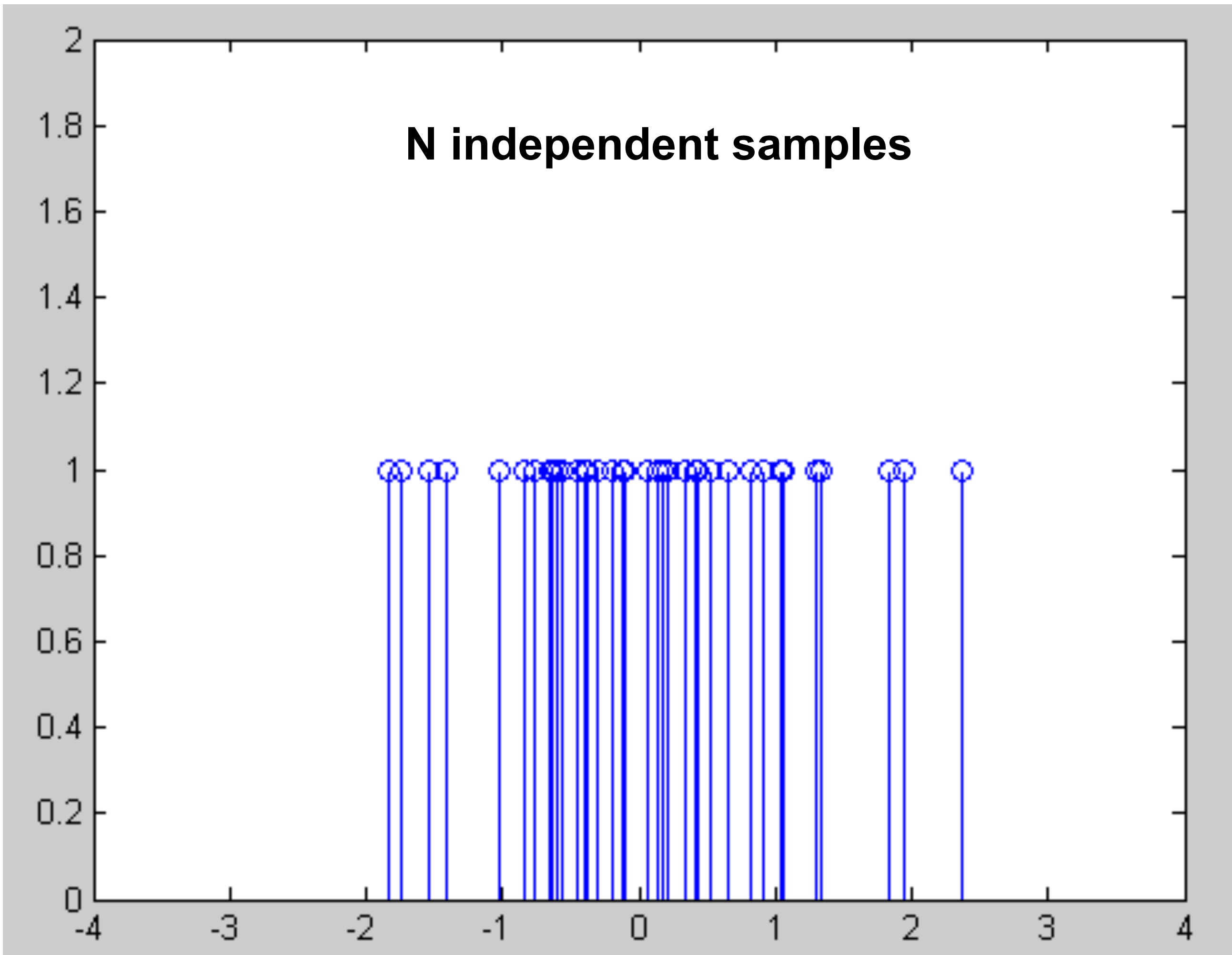
When is Bayesian decision theory most useful?

- Priors and loss functions are most useful in vision (and other fields) in cases where the observations don’t completely specify the answer.
- For example:
  – Human motion priors useful for human body tracking
  – Image priors useful for noise removal and super-resolution.
  – Object category priors useful for object recognition.

Some notes on fitting models

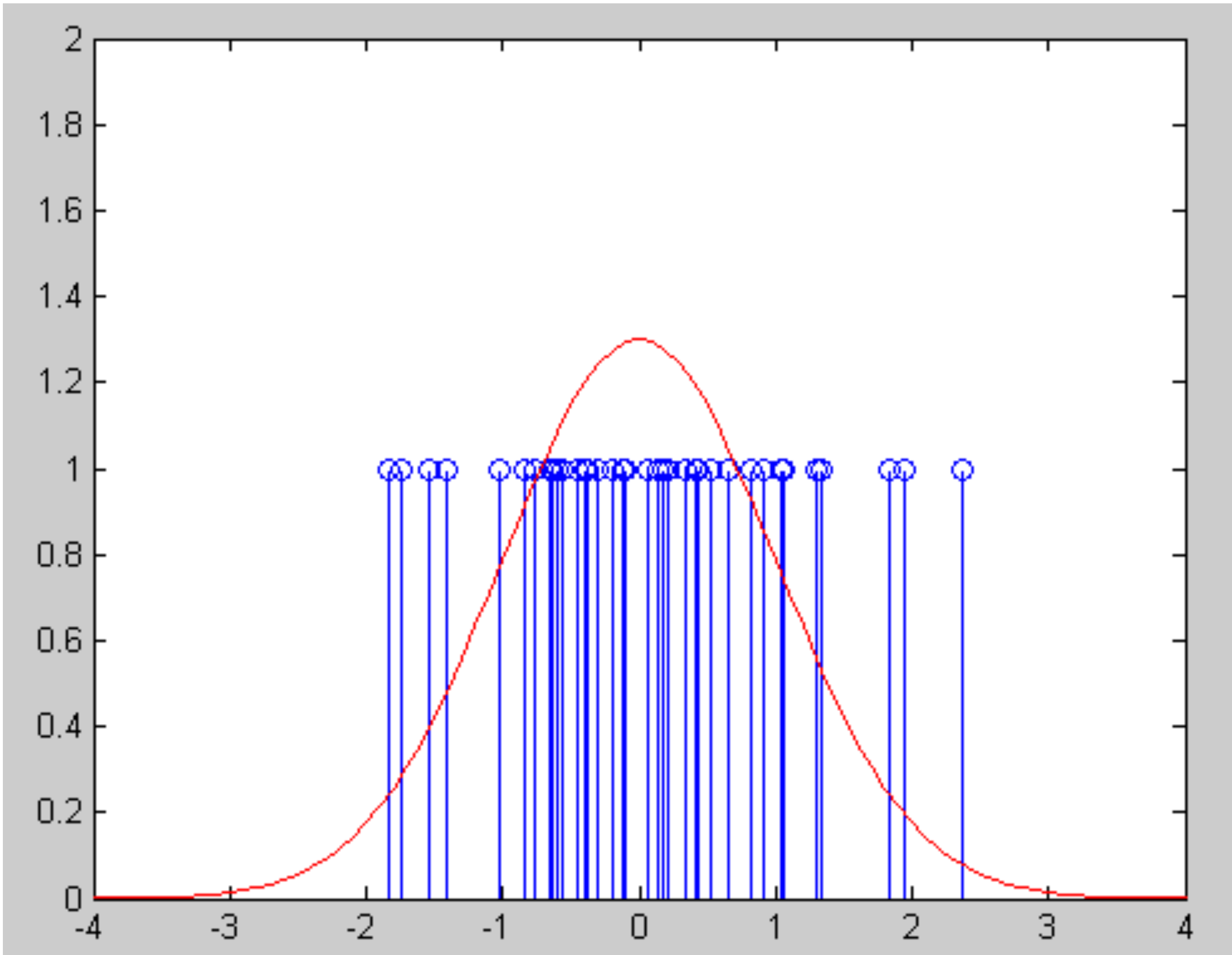
Slides by B. Freeman and A. Blake

The simplest data to model: a set of 1–d samples



Fit this distribution with a Gaussian

$$P(z_n|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(z_n-\mu)^2}{2\sigma^2}$$



How find the parameters of the best-fitting Gaussian?

Posterior probability **Likelihood function** **Prior probability**

$$P(\mu, \sigma | z) = \frac{P(z|\mu,\sigma)P(\mu,\sigma)}{P(z)}$$

mean

data points

std. dev.

Evidence

By Bayes rule

How find the parameters of the best-fitting Gaussian?

Posterior probability **Likelihood function** **Prior probability**

$$P(\mu, \sigma | z) = \frac{P(z|\mu,\sigma)P(\mu,\sigma)}{P(z)}$$

mean **data points**

std. dev. **Evidence**

Maximum likelihood parameter estimation:

$$\hat{\mu}, \hat{\sigma} = \mathrm{argmax}_{\mu,\sigma} \; P(z|\mu, \sigma)$$

Derivation of MLE for Gaussians

Observation density

$$p(z_n|\mu,\sigma^2) \propto \frac{1}{\sigma}\exp - \frac{1}{2\sigma^2}(z_n-\mu)^2$$

Log likelihood

$$L_n = \text{const} - \log\sigma - \frac{1}{2\sigma^2}(z_n-\mu)^2$$

$$L = \sum_n^N L_n.$$

Assuming all the samples are independent

Maximisation

$$0 = \frac{\partial L}{\partial \mu} = \frac{1}{2\sigma^2}\sum_n^N (z_n-\mu)$$

$$0 = \frac{\partial L}{\partial \sigma} = \sum_n^N \left(-\frac{1}{\sigma} + \frac{1}{\sigma^3}(z_n-\mu)^2\right)$$

Basic Maximum Likelihood Estimate (MLE) of a Gaussian distribution

$$0 = \frac{\partial L}{\partial \mu} = \frac{1}{2\sigma^2} \sum_n (z_n - \mu)$$

Mean

$$\hat{\mu} = m \equiv \frac{1}{N} \sum_{n=1}^{N} z_n$$

$$0 = \frac{\partial L}{\partial \sigma} = \sum_n \left(-\frac{1}{\sigma} + \frac{1}{\sigma^3} (z_n - \mu)^2 \right)$$

Variance

$$\hat{\sigma^2} = S \equiv \frac{1}{N} \sum_{n=1}^{N} (z_n - \mu)^2$$

Basic Maximum Likelihood Estimate (MLE) of a Gaussian distribution

Mean

$$\hat{\mu} = m \equiv \frac{1}{N} \sum_{n=1}^{N} z_n$$

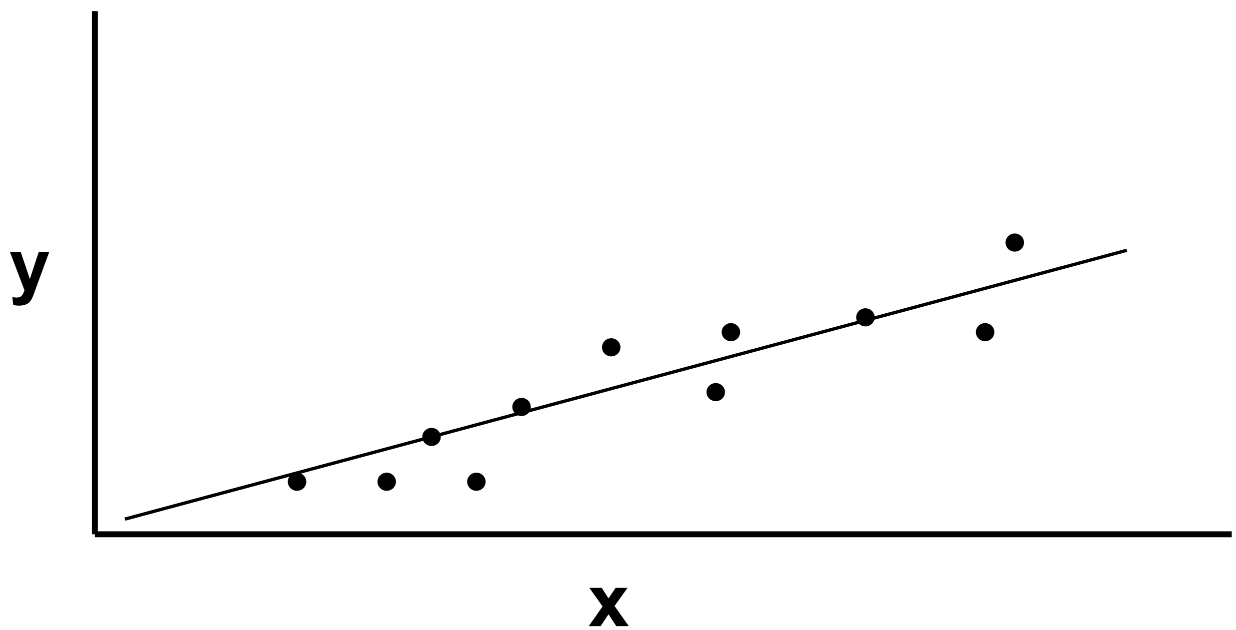
Variance

$$\hat{\sigma^2} = S \equiv \frac{1}{N} \sum_{n=1}^{N} (z_n - \mu)^2$$

For vector-valued data,
we have the Covariance Matrix

$$\hat{P} = S \equiv \frac{1}{N} \sum_{n=1}^{N} (z_n - \mu)(z_n - \mu)^\top$$

Model fitting example 2: Fit a line to observed data



Maximum likelihood estimation for the slope of a single line

data: $(X_n, Y_n), n = 1 \ldots, N$

model: $Y = aX + w$

where $w \sim N(\mu = 0, \sigma = 1)$.

Data likelihood for point n:

$$P(X_n, Y_n | a) = c \exp[-(Y_n - aX_n)^2/2]$$

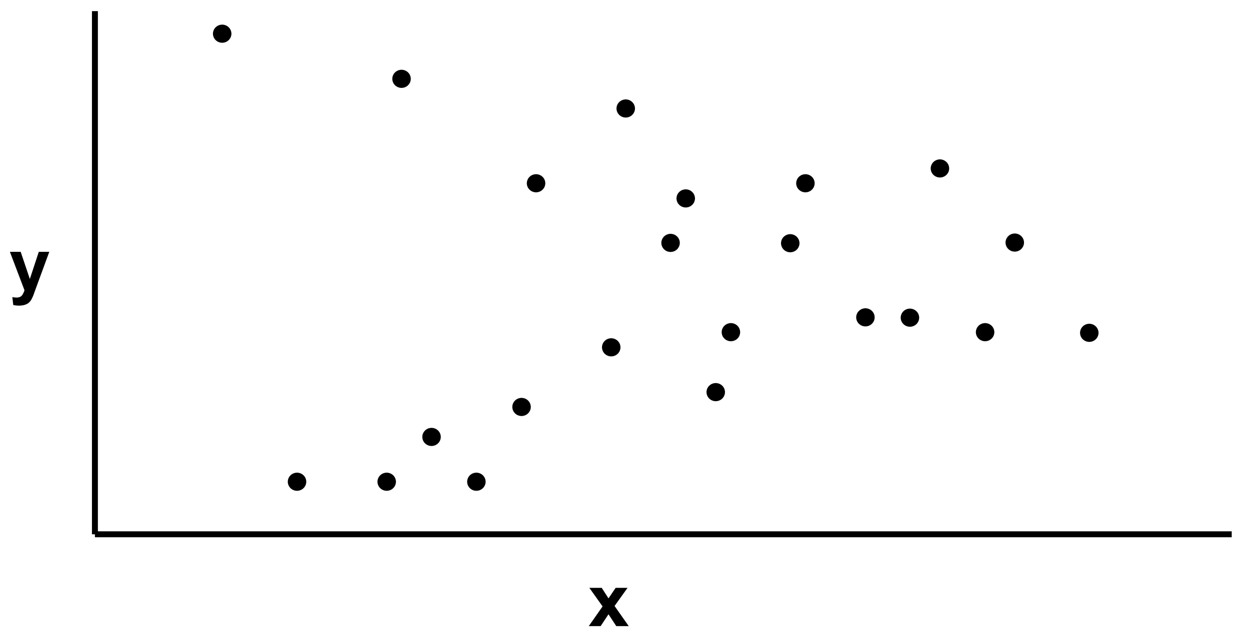
Maximum likelihood estimate:

$$\hat{a} = \arg\max_a p(Y_1, \ldots, Y_n | a) = \arg\max_a \sum_n -d(Y_n; a)^2/2$$

where $d(Y_n; a) = |Y_n - aX_n|$

gives regression formula $\hat{a} = \frac{\sum_n Y_n X_n}{\sum_n X_n^2}$.

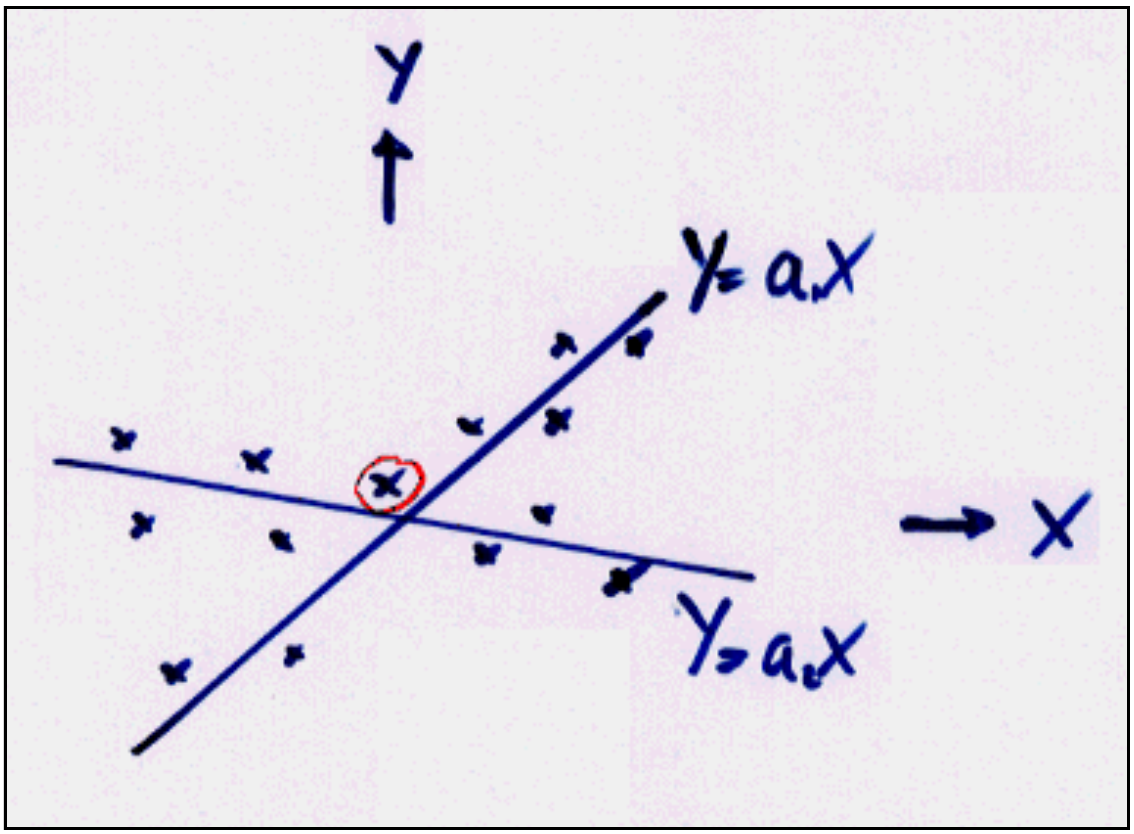
Model fitting example 3: Fitting two lines to observed data



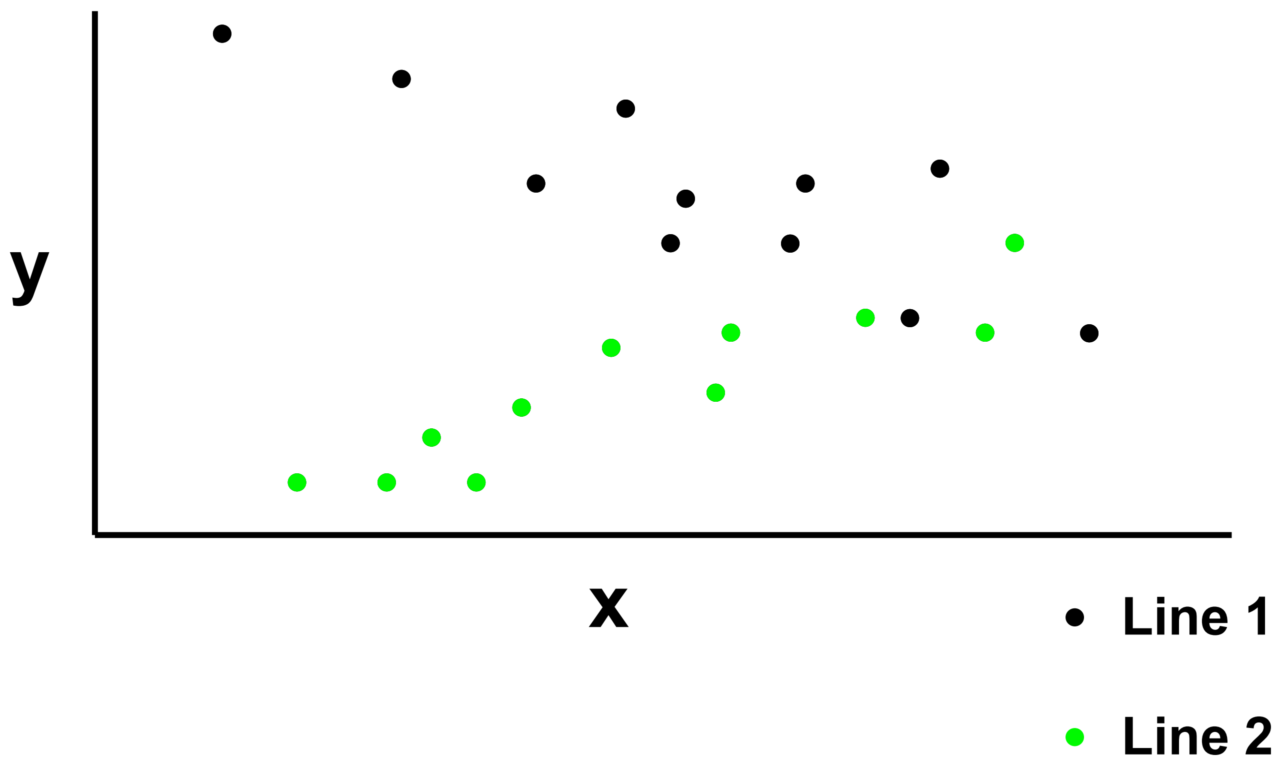
MLE for fitting a line pair

Lines $Y = a_1 X + w$ or $Y = a_2 X + w$, with $w \sim \mathcal{N}(0,1)$.

(a form of mixture dist. Y·)

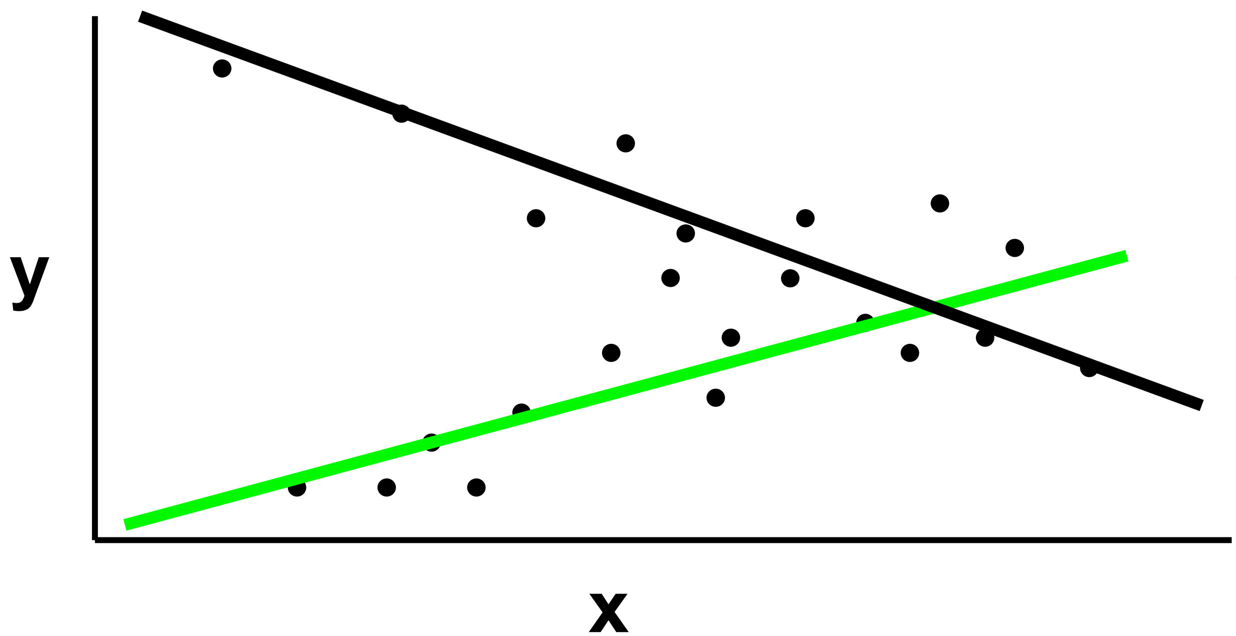


Fitting two lines: on the one hand…



If we knew which points went with which lines, we'd be back at the single line-fitting problem, twice.

Fitting two lines, on the other hand…



We could figure out the probability that any point came from either line if we just knew the two equations for the two lines.

Expectation Maximization (EM): a solution to chicken-and-egg problems

MLE with hidden/latent variables: Expectation Maximisation

General problem:

$$y = (Y_1, \ldots, Y_N); \;\; \theta = (a_1, a_2); \;\; z = (z_1, \ldots, z_N)$$

data **parameters** **hidden variables**

For MLE, want to maximise the log likelihood

The sum over z inside the log gives a complicated expression for the ML solution.

$$\begin{aligned}\widehat{\theta} &= \text{arg max}_\theta \log p(y|\theta) \\ &= \text{arg max}_\theta \log \textstyle\sum_z p(y, z|\theta)\end{aligned}$$

Maximizing the log likelihood of the data

if you knew the z_n labels for each sample n:

$$\hat{\theta} = \mathrm{argmax}_{\theta} \sum_{n} \delta(z_n = 1) \log p(y_n \mid z_n = 1, \theta) + \delta(z_n = 2) \log p(y_n \mid z_n = 2, \theta)$$

Maximizing the log likelihood of the data

if you knew the z_n labels for each sample n:

$$\hat{\theta} = \text{argmax}_{\theta} \sum_n \delta(z_n = 1) \log p(y_n \mid z_n = 1, \theta) + \delta(z_n = 2) \log p(y_n \mid z_n = 2, \theta)$$

In the EM algorithm, we replace those known labels with their expectation under the current algorithm parameters. So

$$E[\delta(z_n = i)] = p(z_n = i \mid y, \theta_{old})$$

Call that quantity $= \alpha_i(n)$

$$\propto p(y \mid z_n = i, \theta_{old}) \propto e^{-(y_n - a_i x_n)^2 / 2}$$

Maximizing gives

And then for the estimate of the line parameters, we have

$$\hat{\theta} = \text{argmin}_{\theta} \sum_{n} \alpha_1(n)(y_n - a_1 x_n)^2 + \alpha_2(n)(y_n - a_2 x_n)^2$$

and maximising that gives

$$\hat{a}_i = \frac{\sum_n \alpha_i(n) y_n x_n}{\sum_n \alpha_i(n) x_n^2}$$

EM fitting to two lines

with $$\alpha_i(n) \propto e^{-(y_n - a_i x_n)^2/2}$$

and $$\alpha_1(n) + \alpha_2(n) = 1$$

"E-step"

repeat

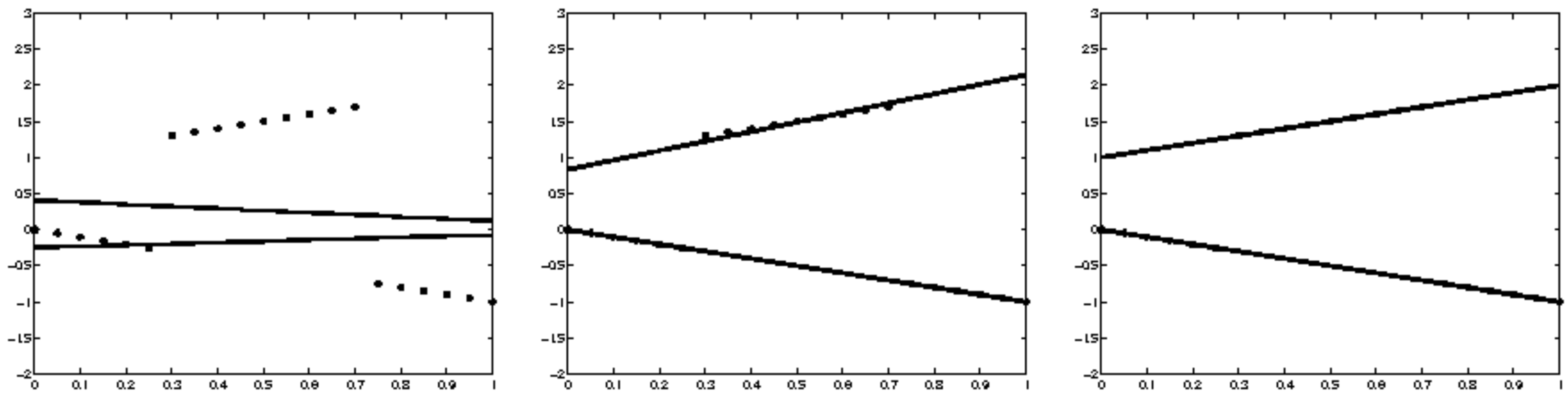
Regression becomes:

$$\hat{a}_i = \frac{\sum_n \alpha_i(n) y_n x_n}{\sum_n \alpha_i(n) x_n^2}$$

"M-step"

Experiments: EM fitting to two lines

(from a tutorial by Yair Weiss, http://www.cs.huji.ac.il/~yweiss/tutorials.html)



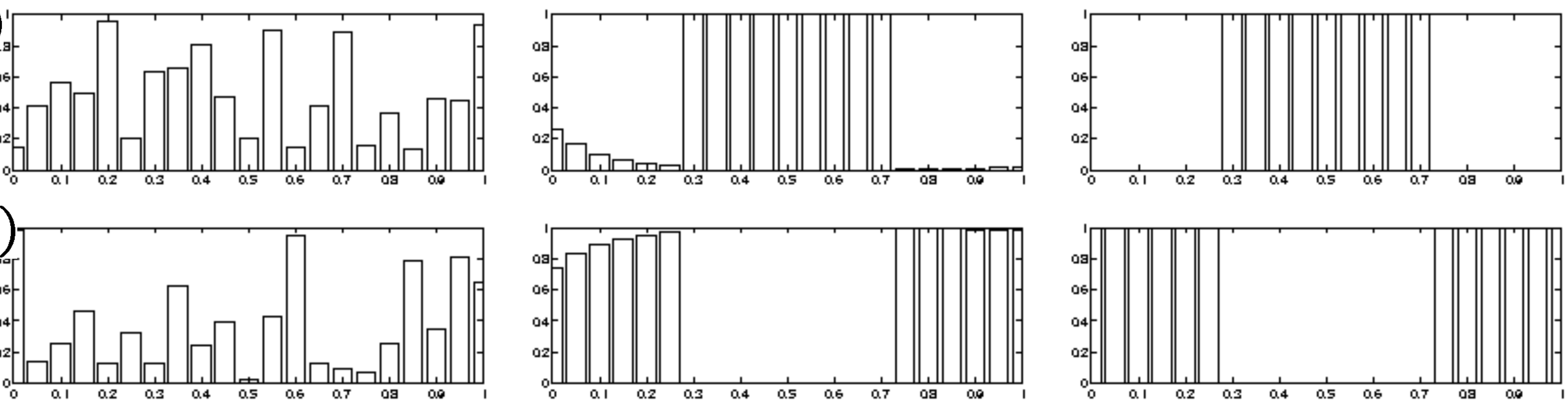
Line weights

$w_1(n)$

line 1

$w_2(n)$

line 2



Iteration **1** **2** **3**

Taking a picture…

What the camera give us…

How do we correct this?

Slides R. Fergus

Close-up

Original

Naïve Sharpening

Our algorithm

Slides R. Fergus

Why does picture appear blurry?

Let's take a photo

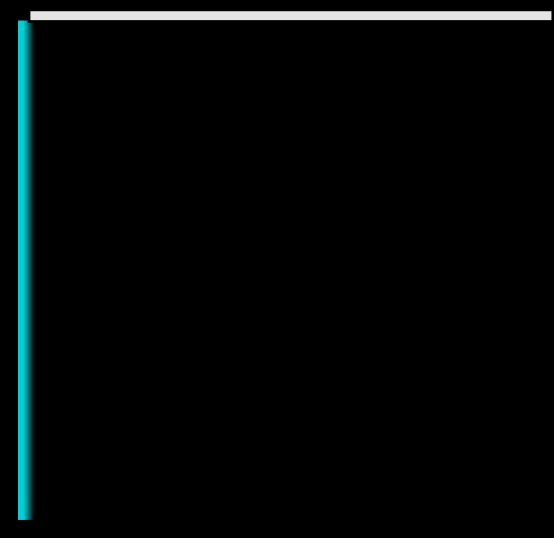
Blurry result

Slides R. Fergus

Slow-motion replay

Slides R. Fergus

Slow-motion replay



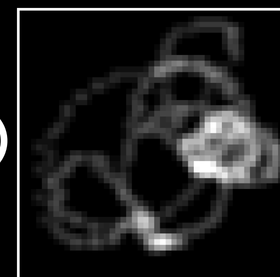
Motion of camera

Slides R. Fergus

Image formation process

=

$\otimes$



Blur kernel

Blurry image

Input to algorithm

Sharp image

Desired output

Convolution operator

Model is approximation

Why is this hard?

Simple analogy:

11 is the product of two numbers.

What are they?

No unique solution:

11 = 1 x 11

11 = 2 x 5.5

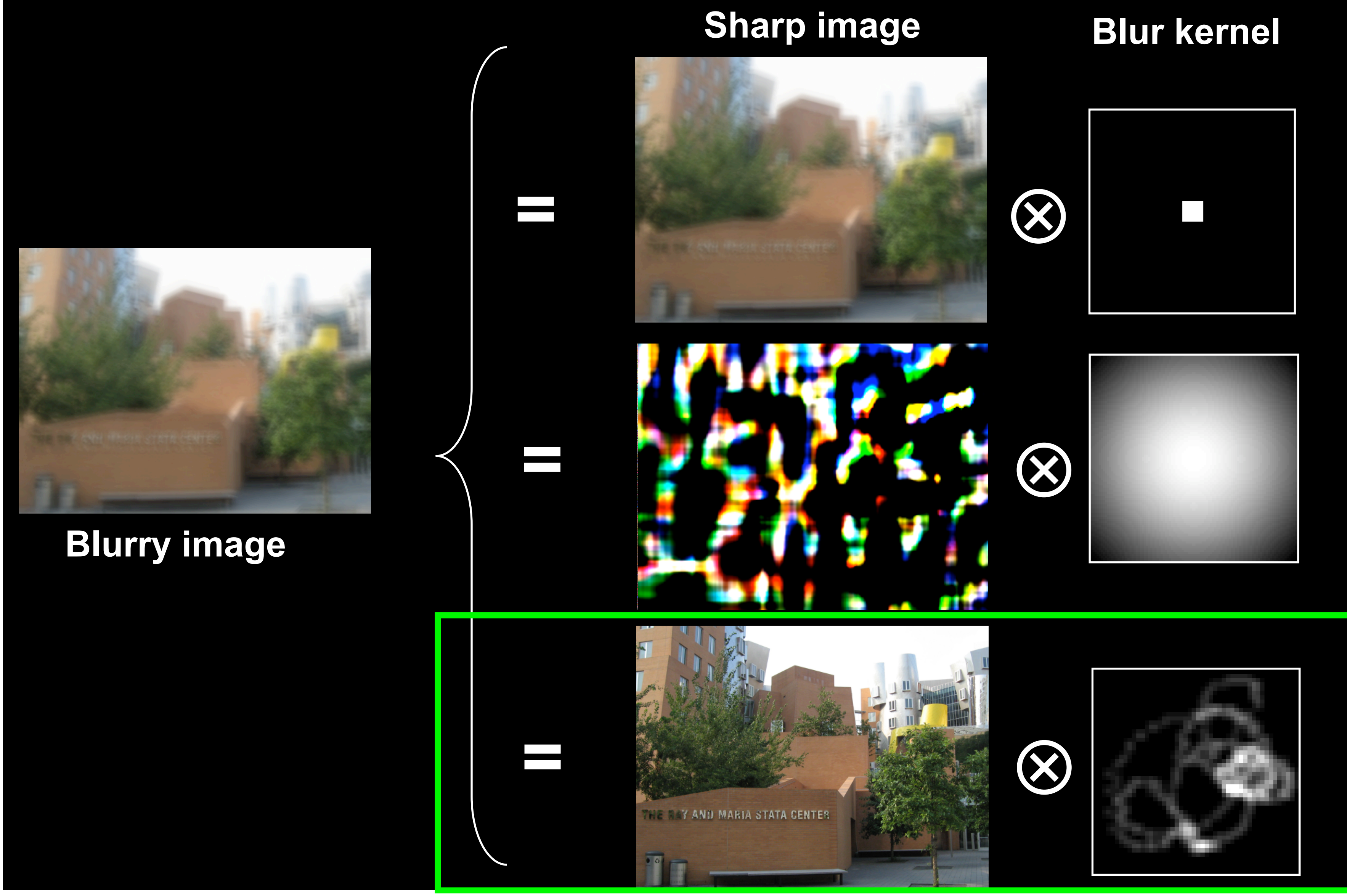
11 = 3 x 3.667

etc…..

Need more information !!!!

Slides R. Fergus

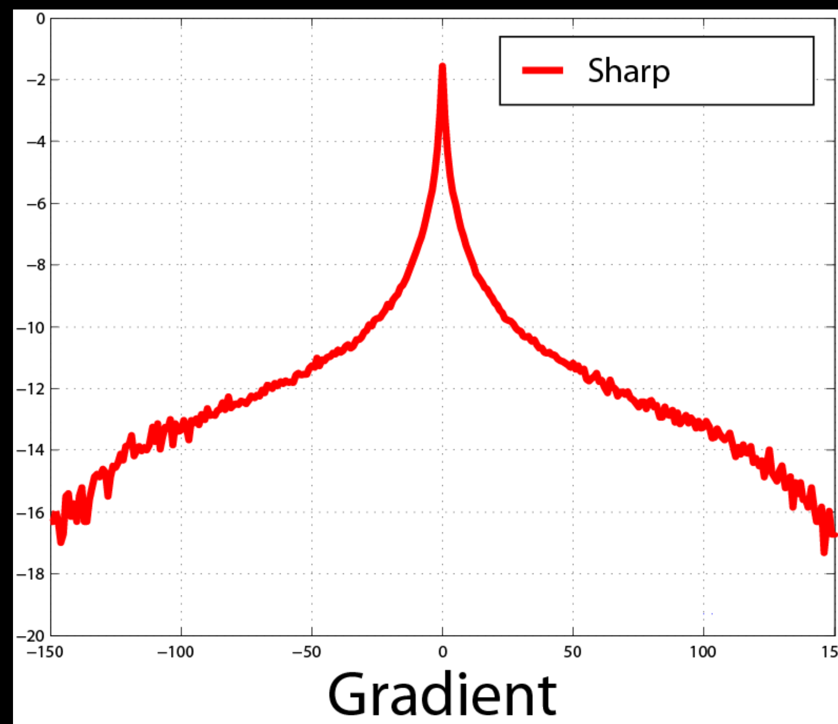
Multiple possible solutions



Natural image statistics

Characteristic distribution with heavy tails

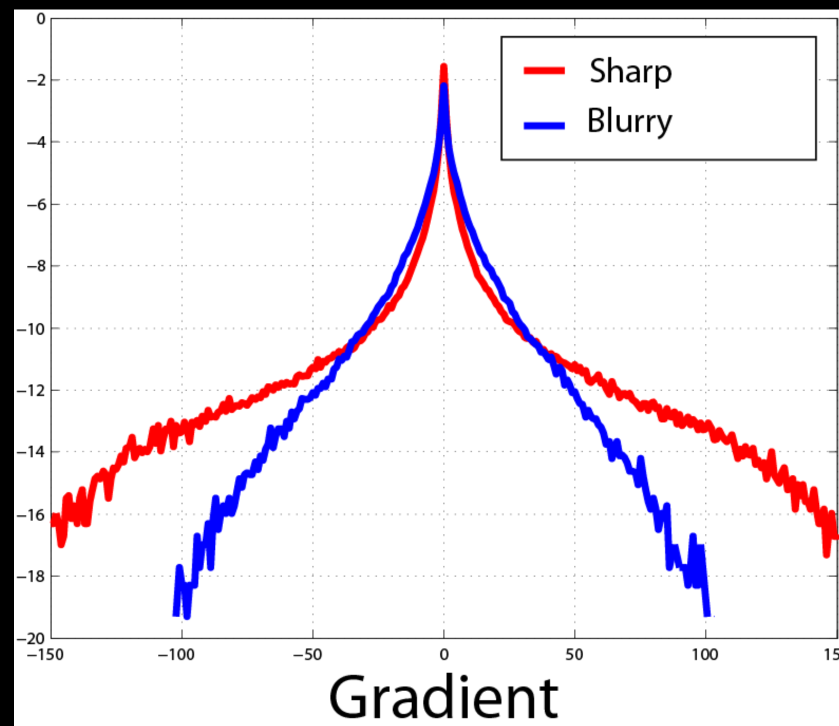
Histogram of image gradients



Slides R. Fergus

Blury images have different statistics

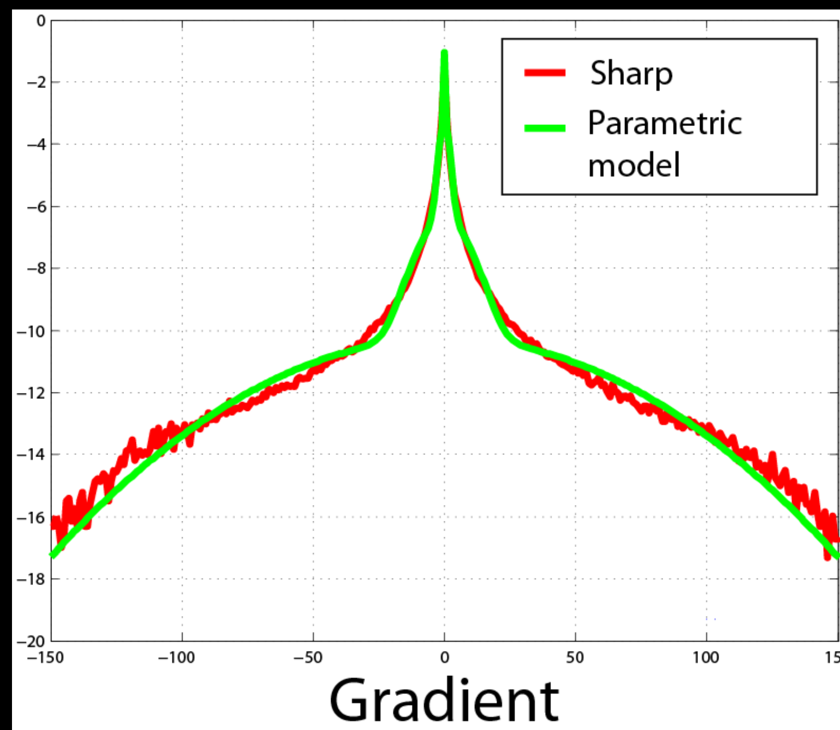
Histogram of image gradients



Slides R. Fergus

Parametric distribution

Histogram of image gradients



Use parametric model of sharp image statistics

Slides R. Fergus

Uses of natural image statistics

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

Corruption process assumed known

Slides R. Fergus

Existing work on image deblurring

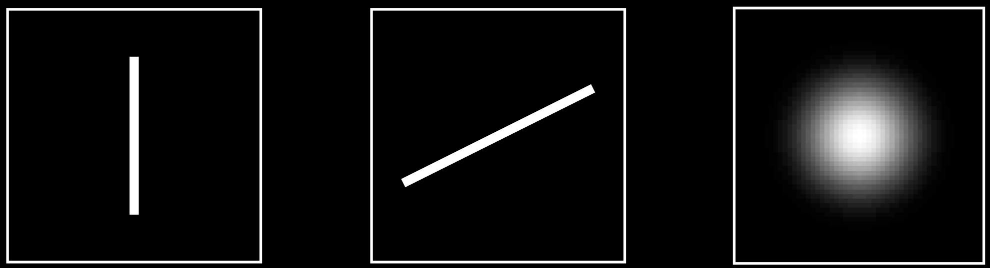
Software algorithms:

– Extensive literature in signal processing community

– Mainly Fourier and/or Wavelet based

– Strong assumptions about blur

→ not true for camera shake



Assumed forms of blur kernels

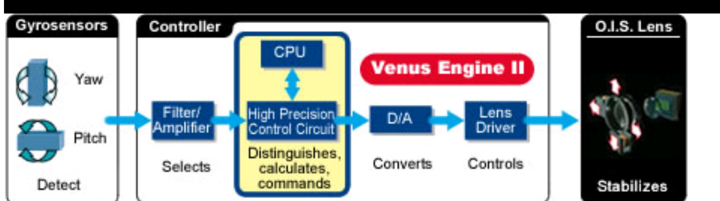
– Image constraints are frequency-domain power-laws

Slides R. Fergus

Existing work on image deblurring

Hardware approaches

Image stabilizers



Dual cameras

Ben-Ezra and Nayar 2004

Coded shutter

Raskar et al. SIGGRAPH 2006

Our approach can be combined with these hardware methods

Three sources of information

1. Reconstruction constraint:

Estimated sharp image

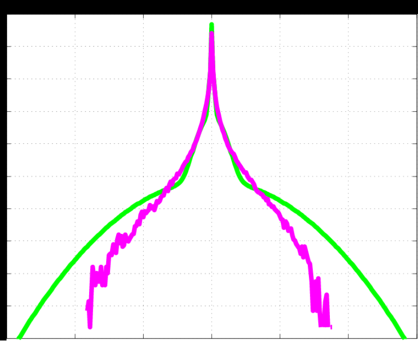
$\otimes$

Estimated blur kernel

=

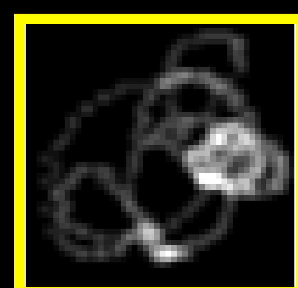
Input blurry image

2. Image prior:



Distribution of gradients

3. Blur prior:



Positive & Sparse

Three sources of information

y = observed image   b = blur kernel   x = sharp image

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

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:

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

$$\propto \prod_i e^{-\frac{(x_i \otimes b - y_i)^2}{2\sigma^2}}$$

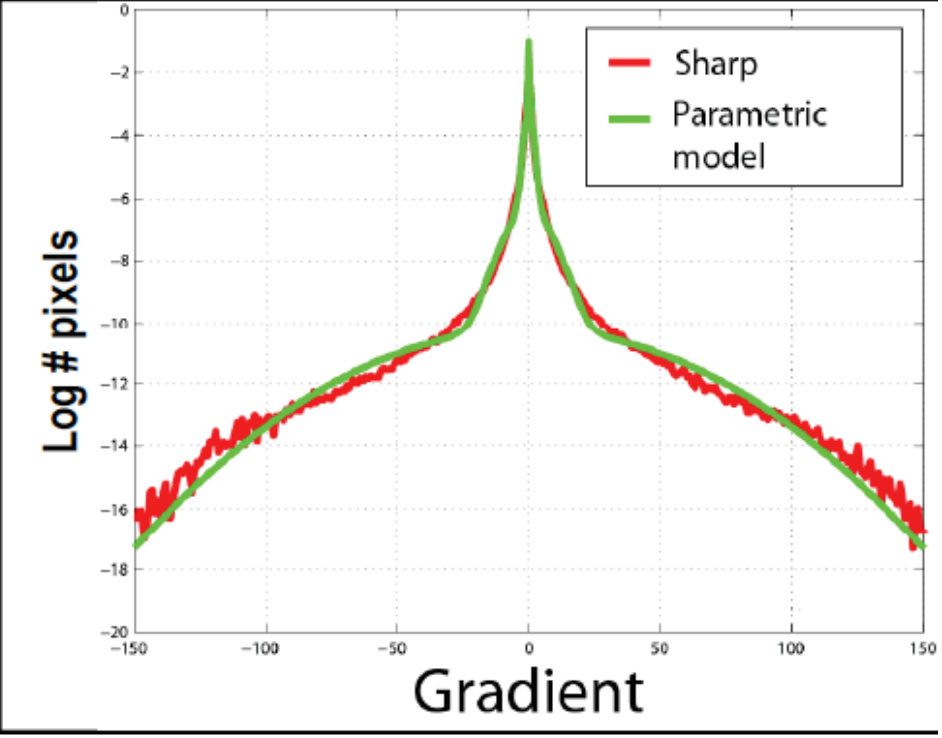
i - pixel index

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

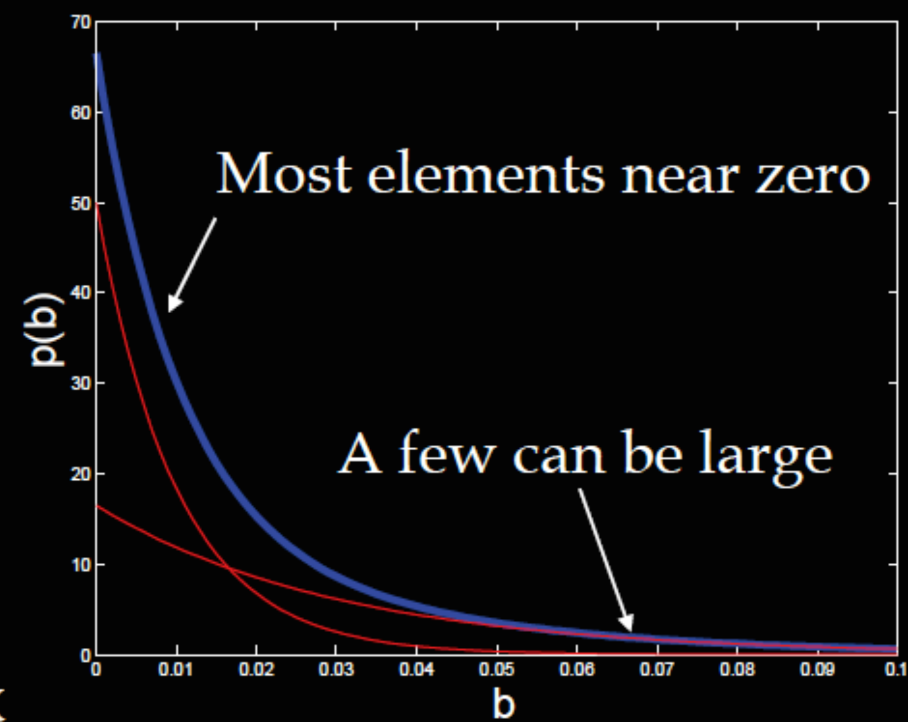
3. Blur prior $p(b)$

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

How do we use this information?

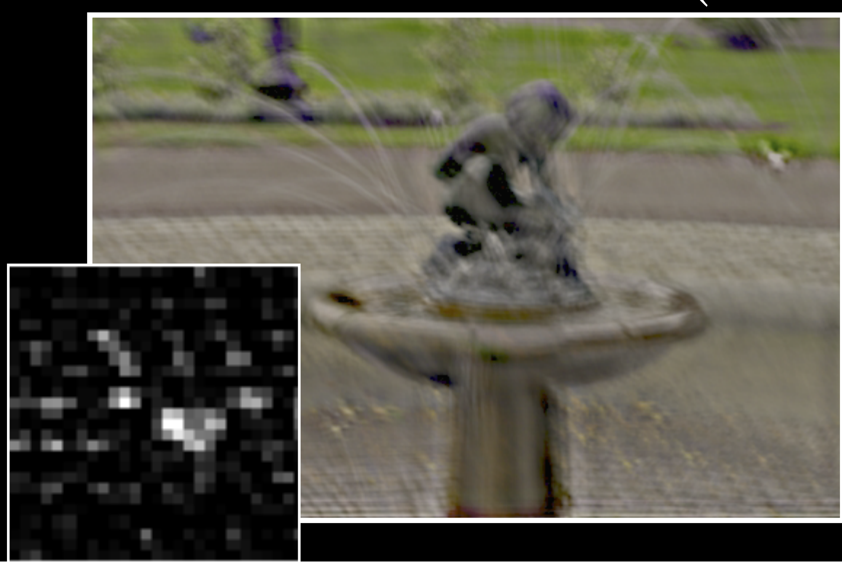
Obvious thing to do:

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

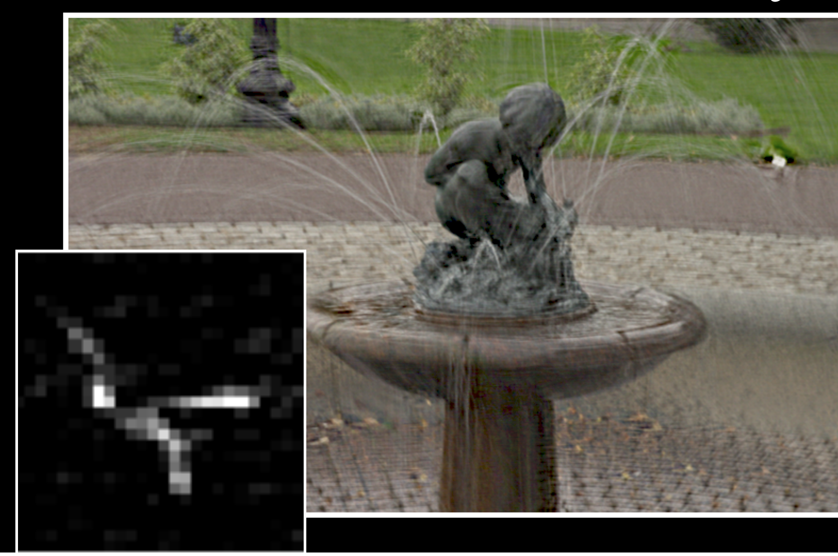
Results from MAP estimation

Input blurry image

Maximum a-Posteriori (MAP)

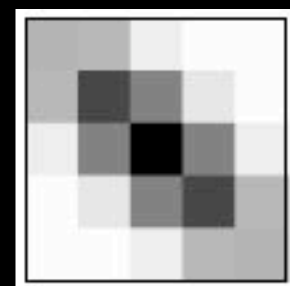


Our method: Variational Bayes



Variational Bayesian method

Based on work of Miskin & Mackay 2000



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

Helps avoid local maxima and over-fitting

Overview of algorithm

1. Pre-processing

2. Kernel estimation
   - Multi-scale approach

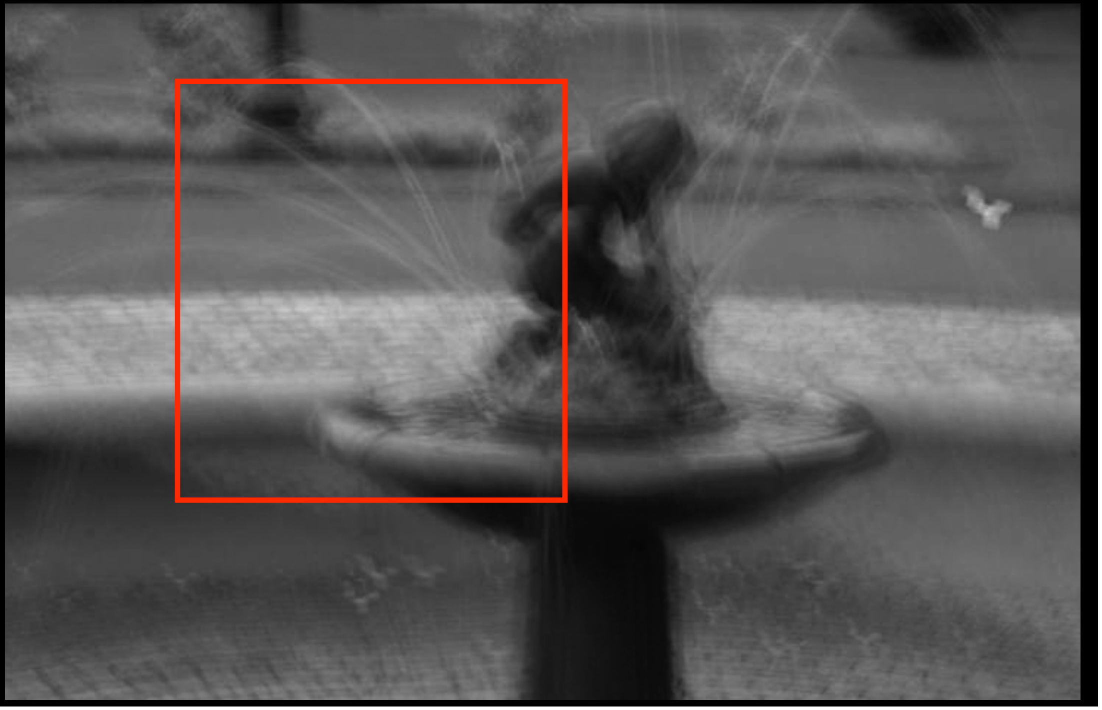
3. Image reconstruction
   - Standard non-blind deconvolution routine

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

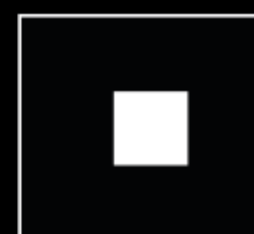
Initialization

Input image

Convert to grayscale → Remove gamma correction → User selects patch from image → **Initialize 3x3 blur kernel**

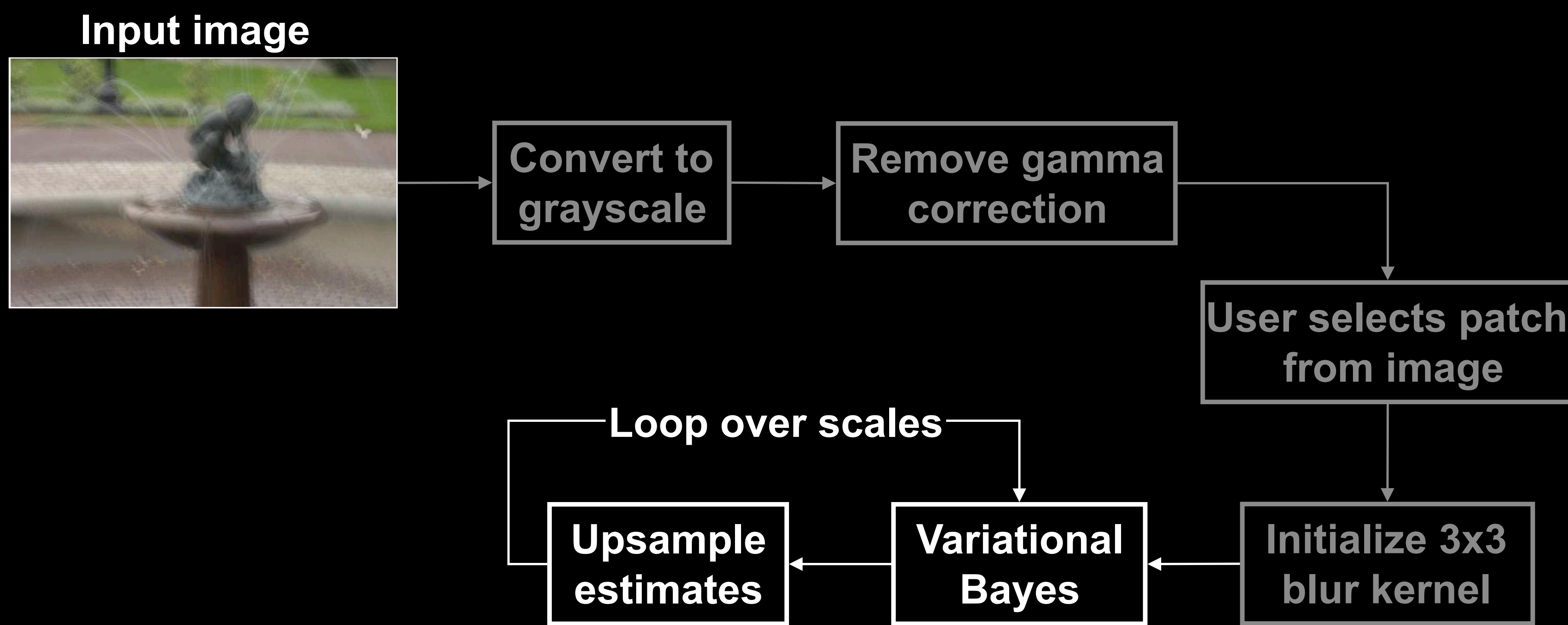
Blurry patch

Initial image estimate



Initial blur kernel

Inferring the kernel: multiscale method



Use multi-scale approach to avoid local minima:

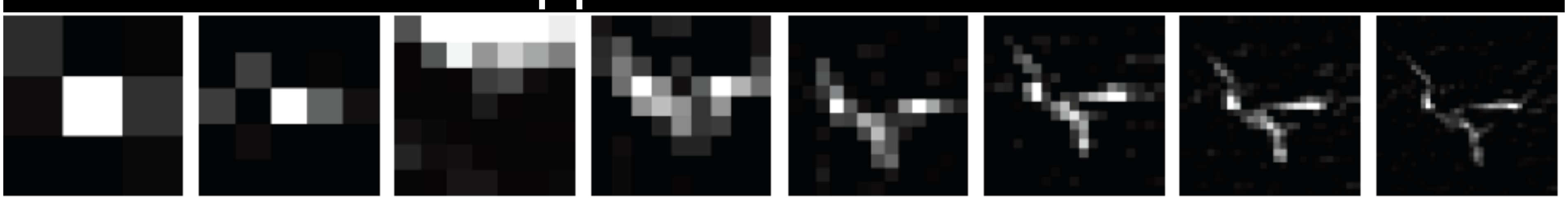
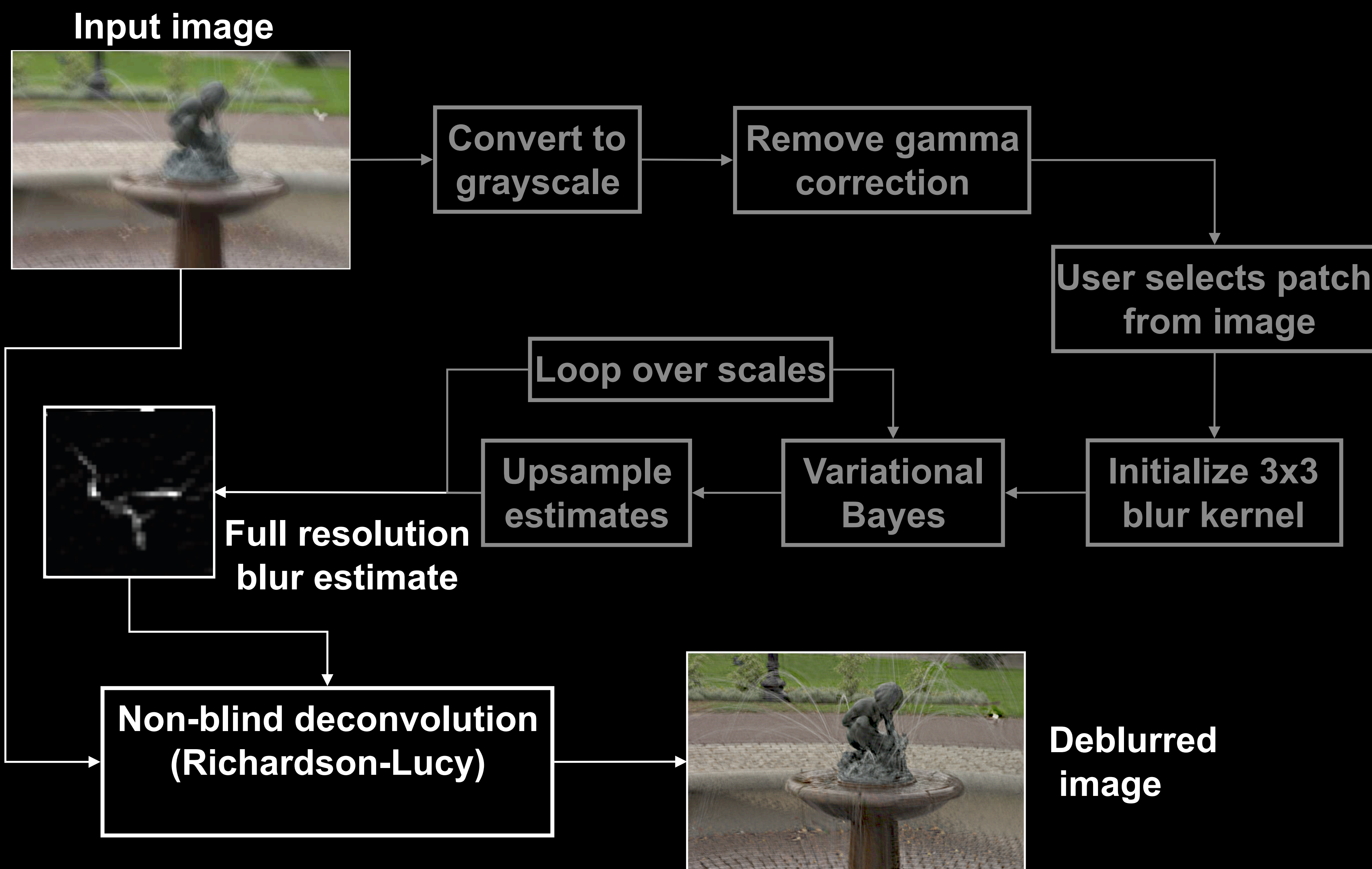


Image Reconstruction



Results on real images

Submitted by people from their own photo collections

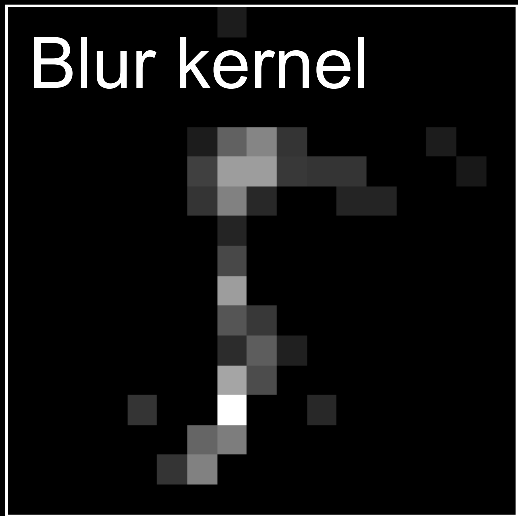
Type of camera unknown

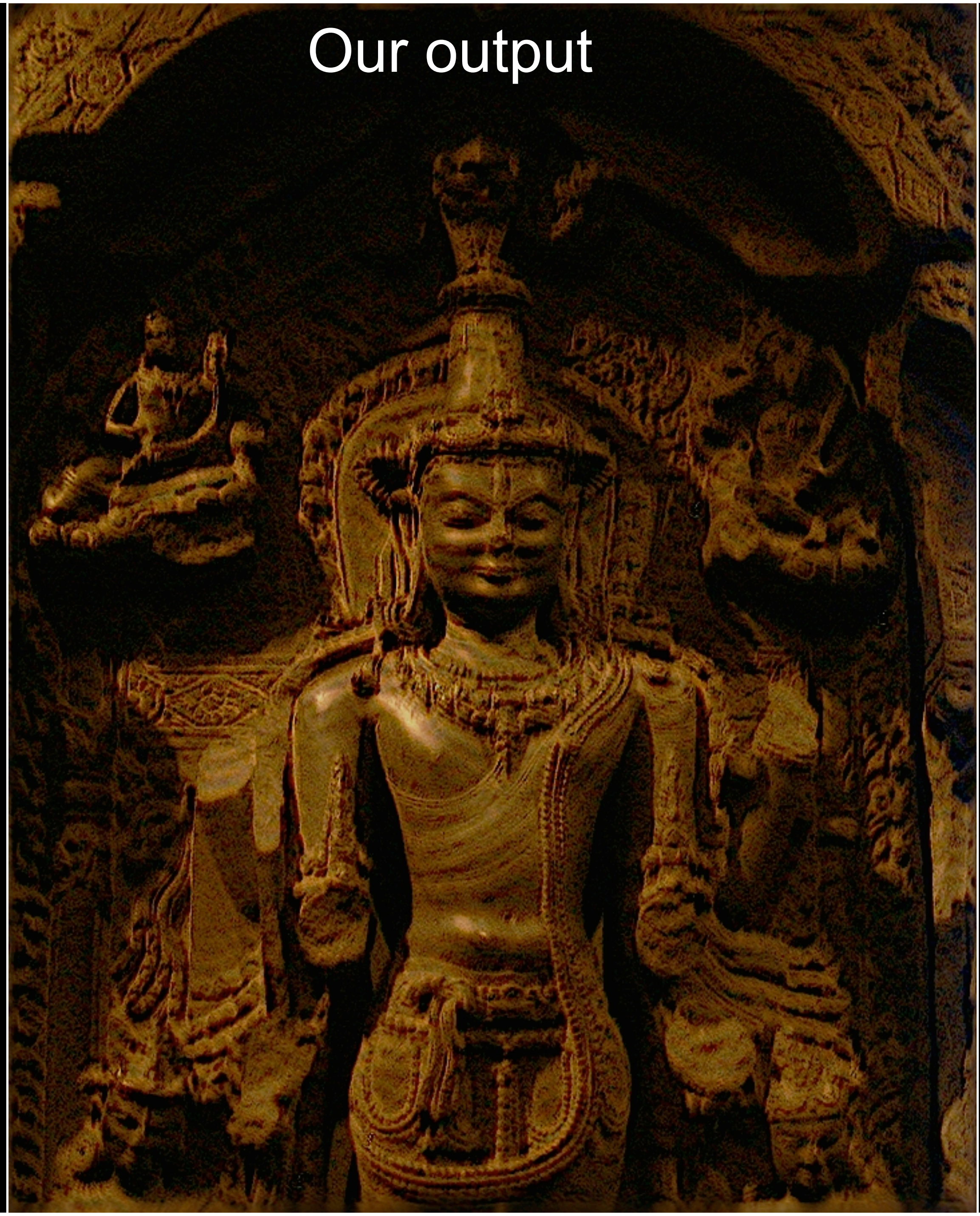
Output does contain artifacts

- Increased noise
- Ringing

Compares well to existing methods


Original photograph

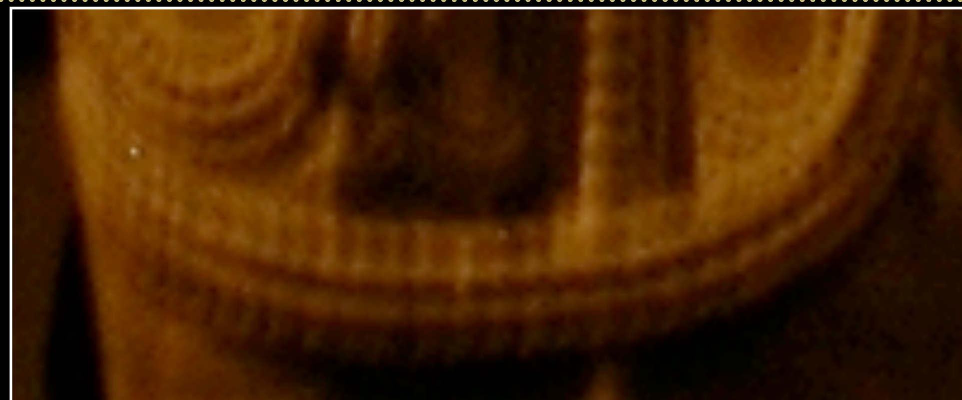

Blur kernel


Our output

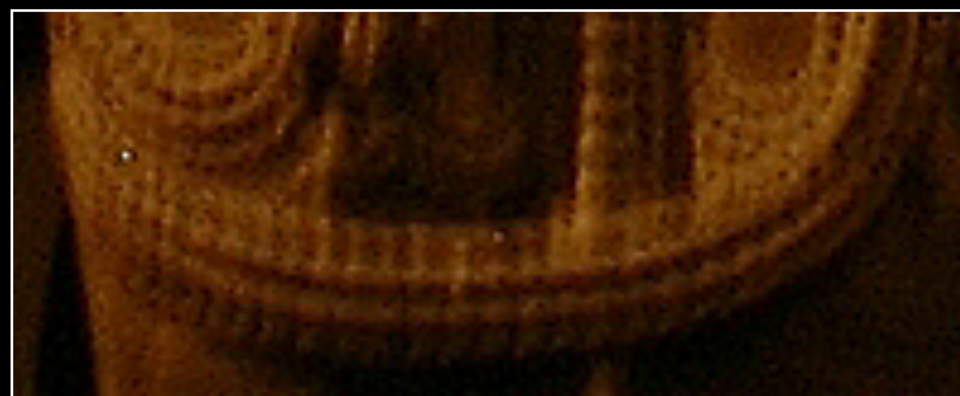
Matlab's `deconvblind`

Close-up of garland

Original



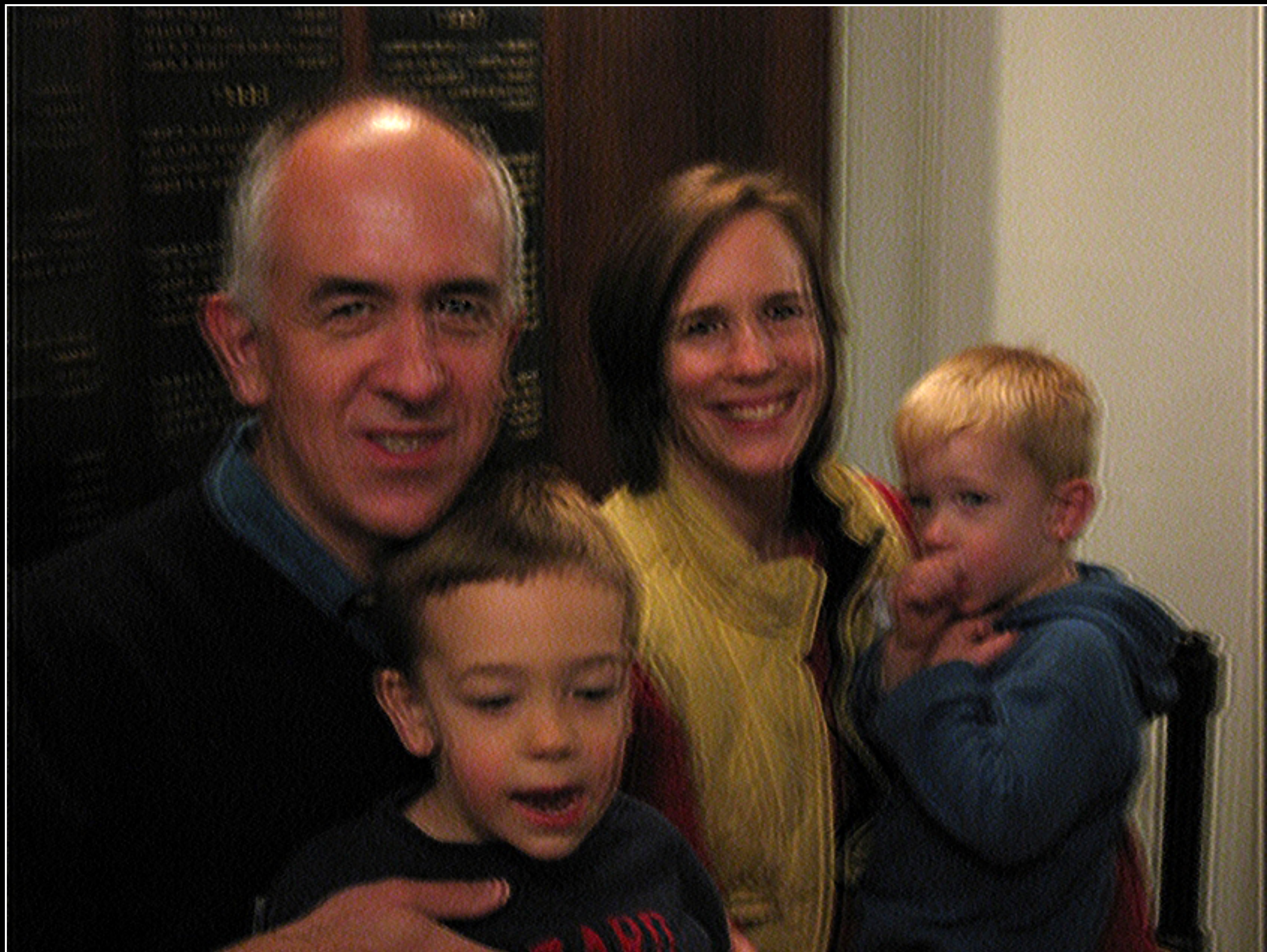
Matlab's
`deconvblind`



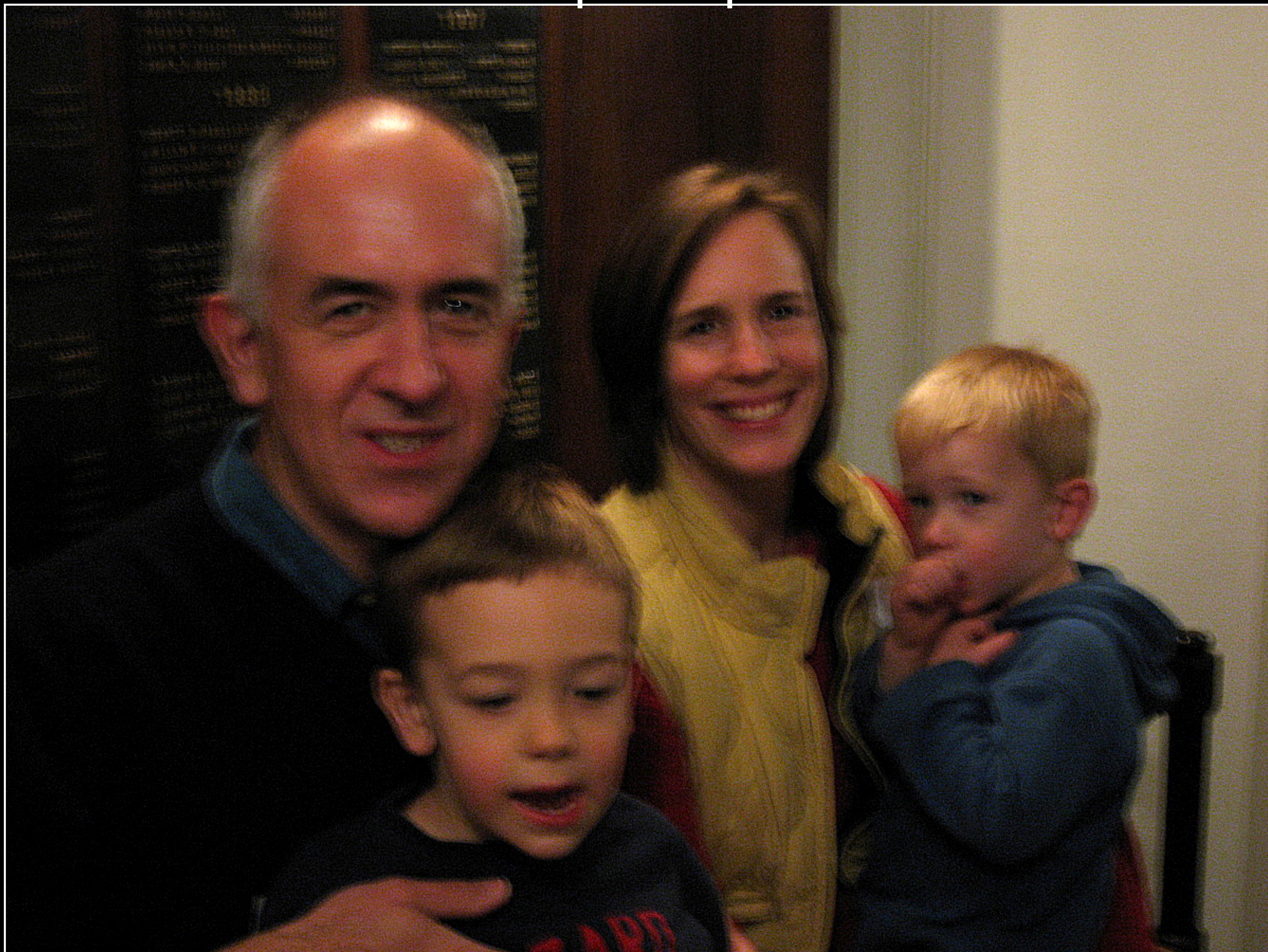
Our output

Original photograph

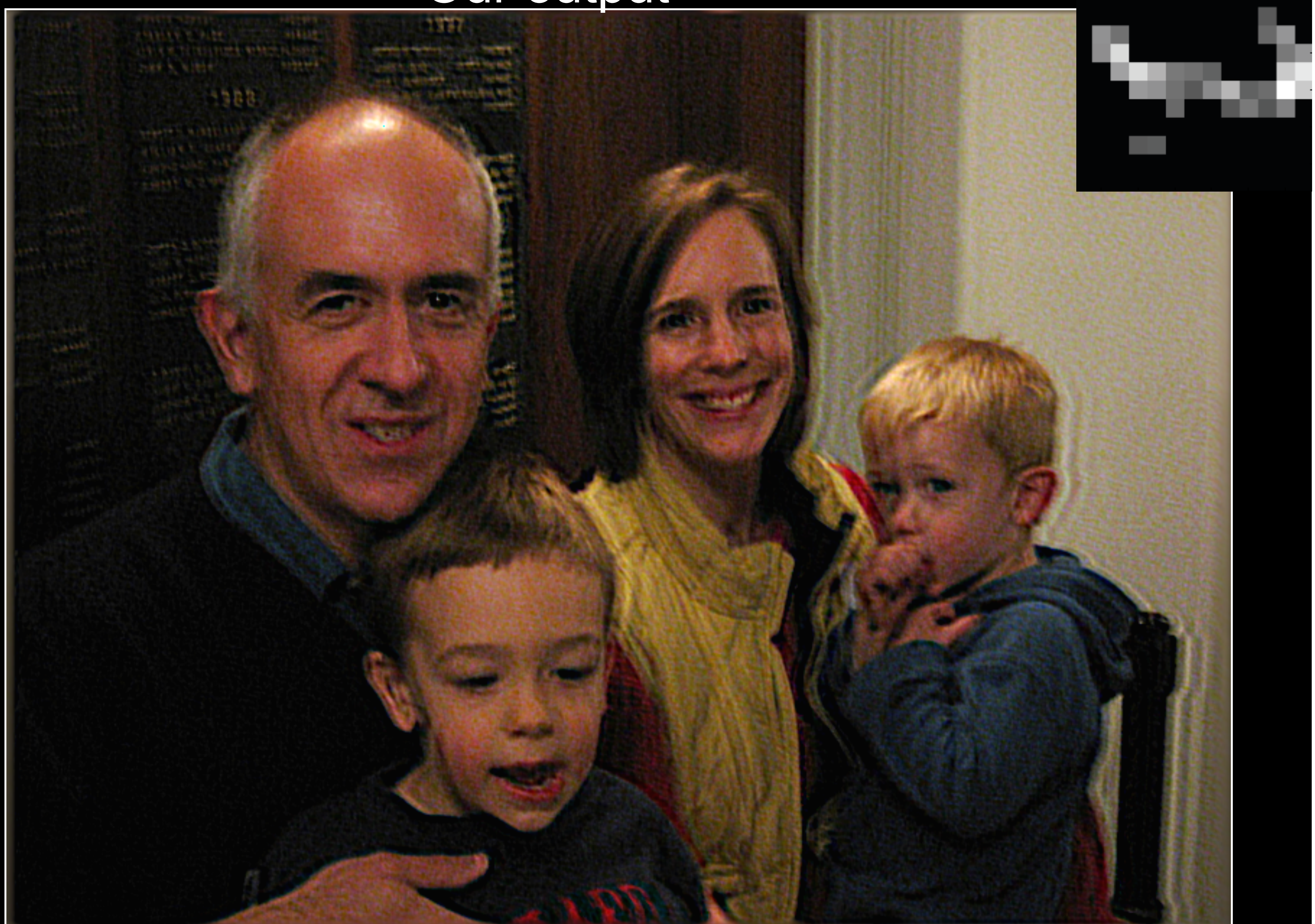
Matlab's `deconvblind`

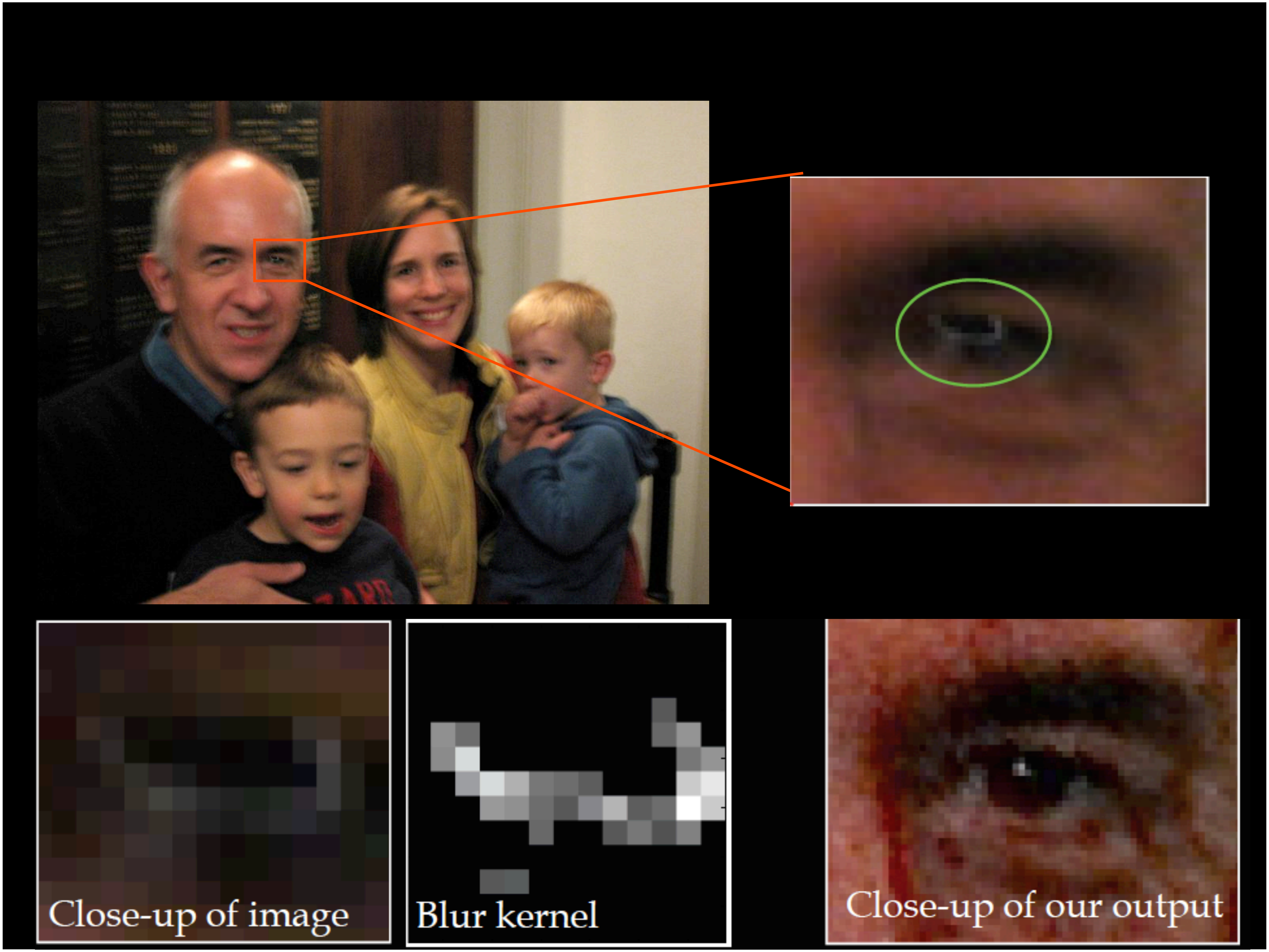


Photoshop sharpen more



Our output




Close-up of image
Blur kernel
Close-up of our output

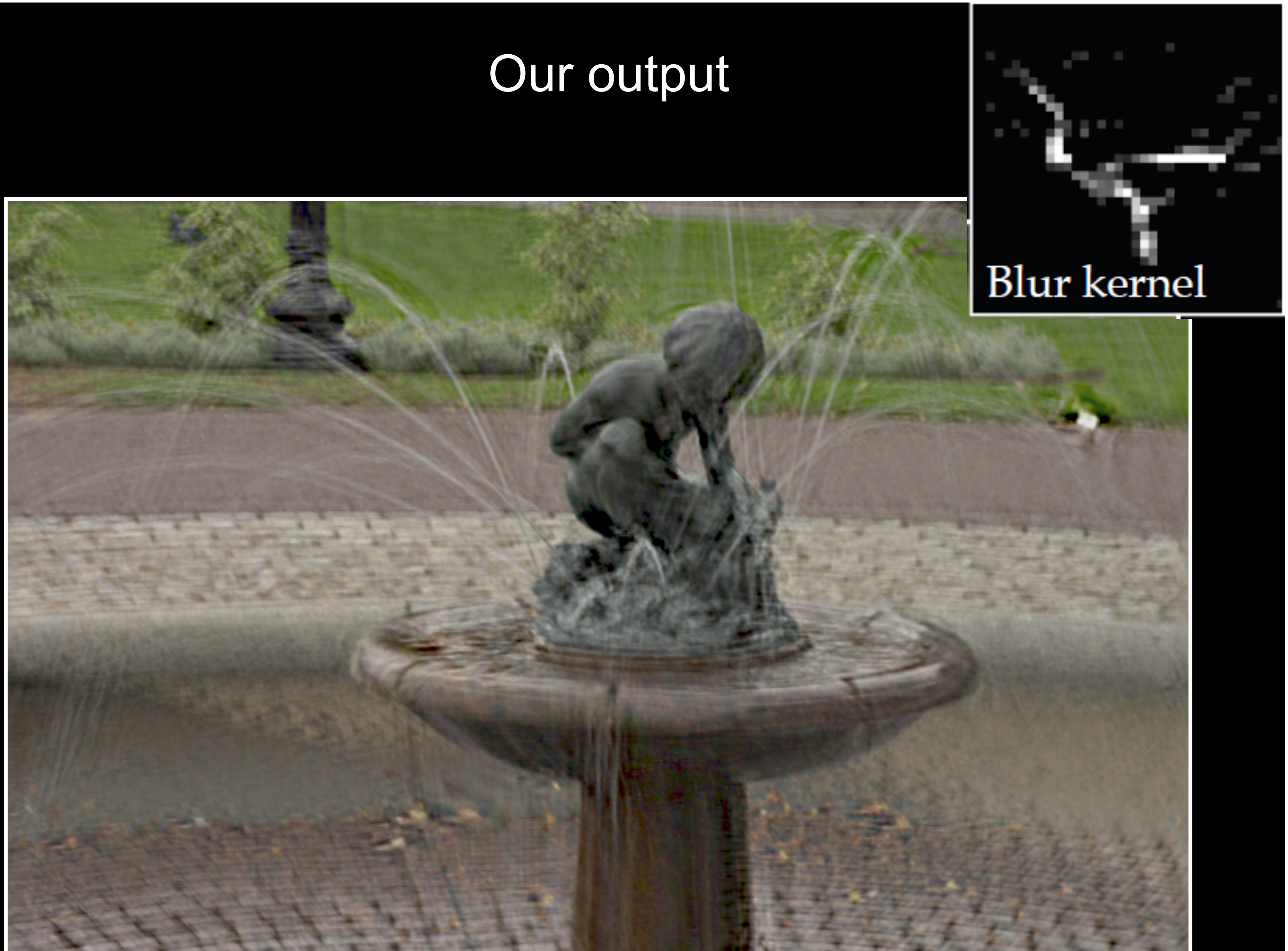
Original photograph

Our output

Blur kernel

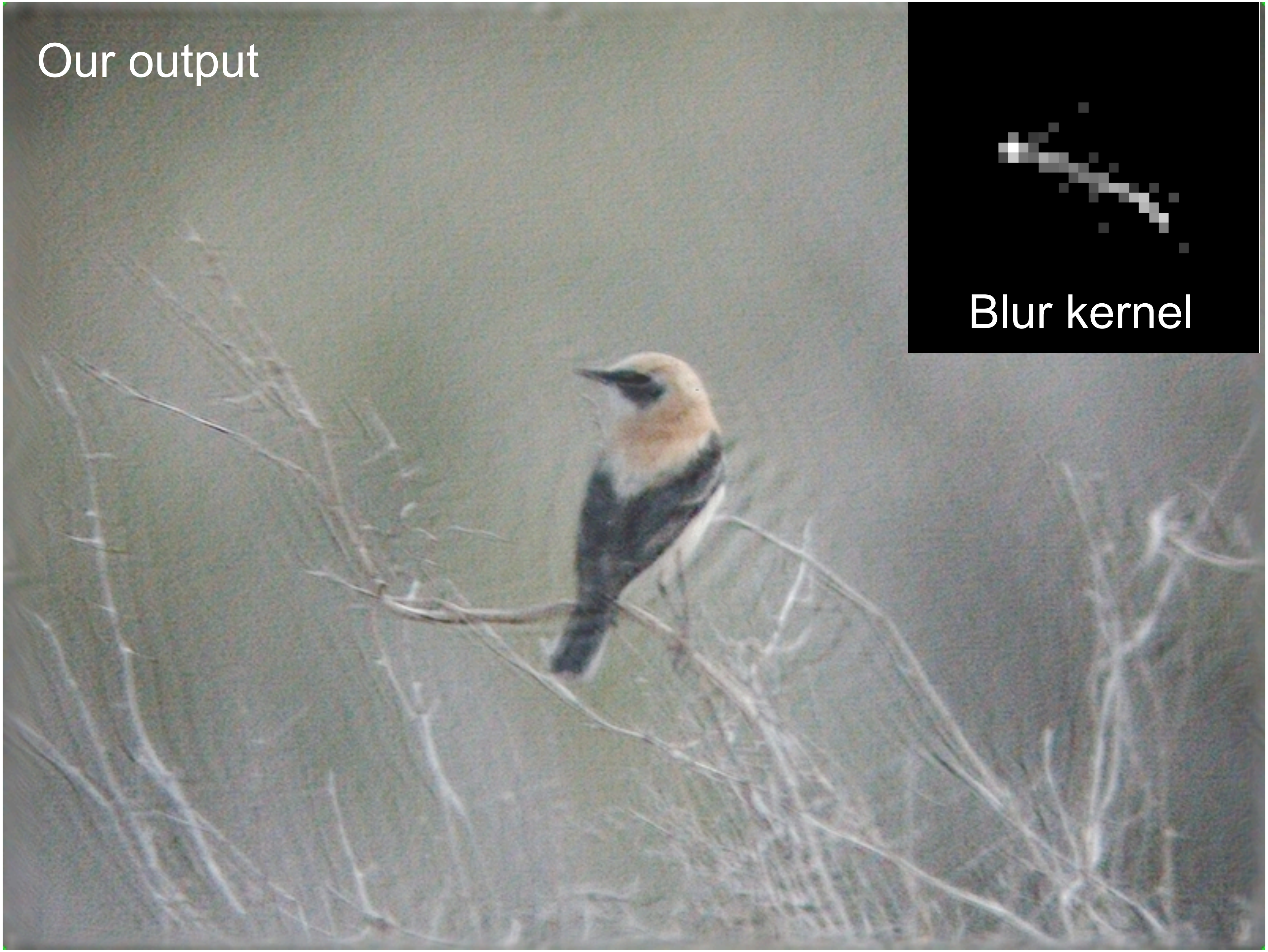
Original photograph

Our output



Matlab’s `deconvblind`

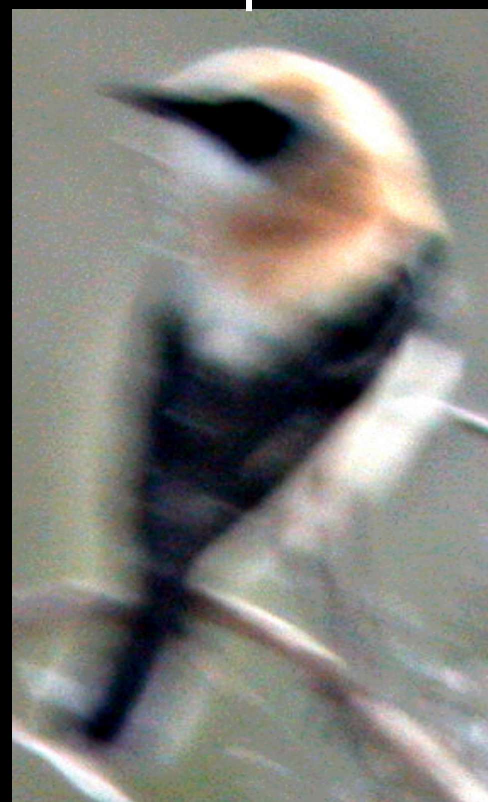

Original photograph


Our output
Blur kernel

Close-up of bird

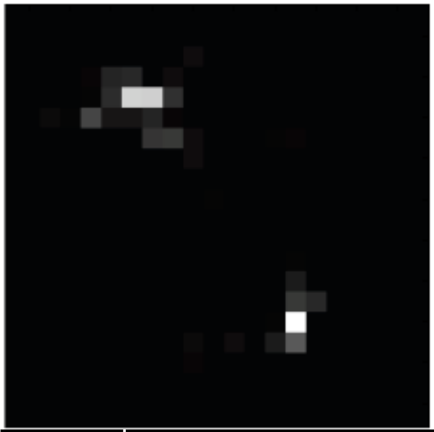
Original

Unsharp mask



Our output


Original photograph



Blur kernel

output

Image artifacts & estimated kernels

Blur kernels

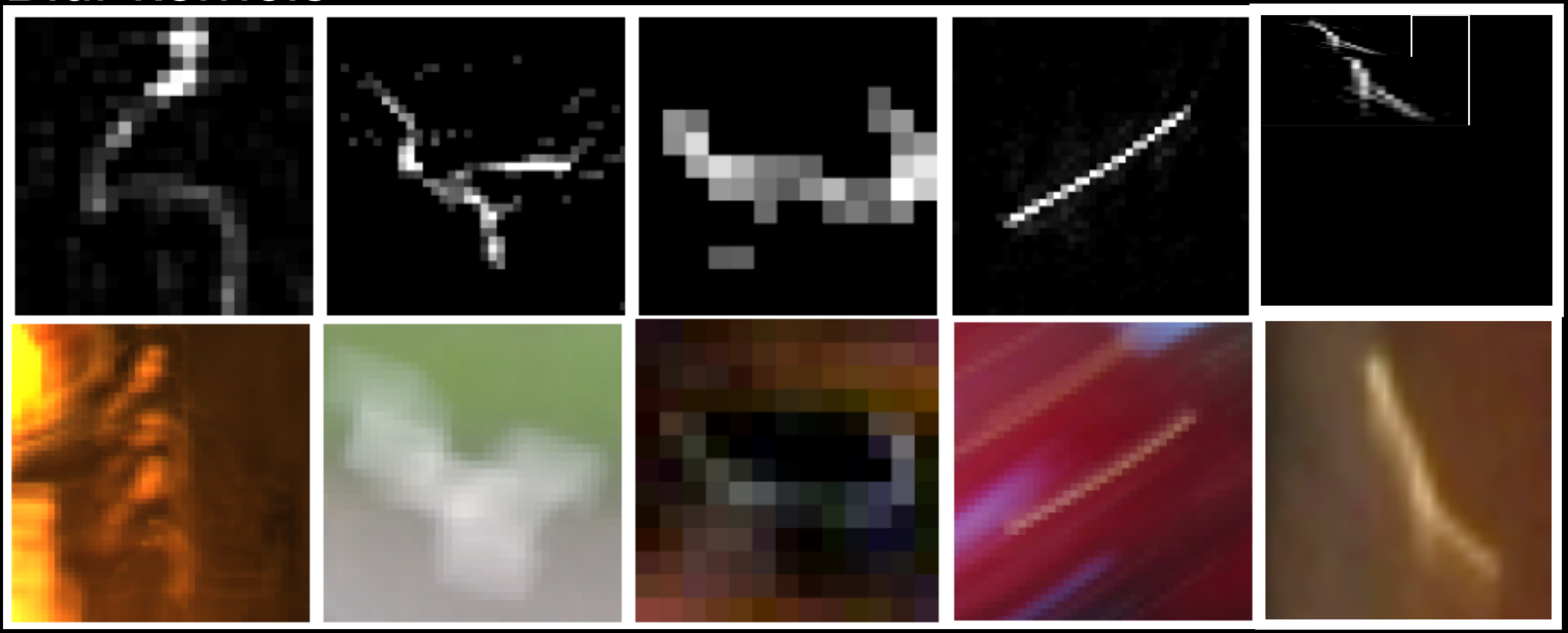


Image patterns

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