



MIT CSAIL

**6.869: Advances in Computer Vision**

MIT  
COMPUTER  
VISION

## Lecture 5

### Statistical Image Models

# Bayesian approach

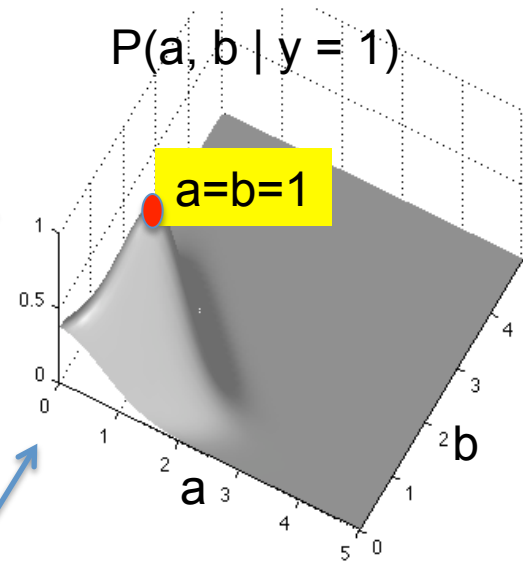
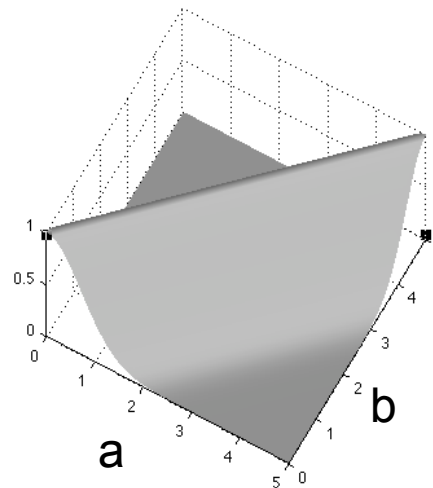
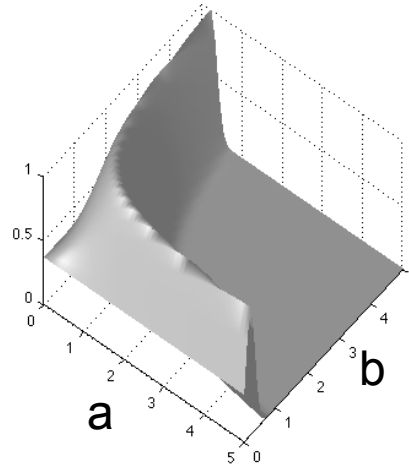
Use  $P(a, b \mid y = 1) = k P(y=1 \mid a, b) P(a, b)$

Likelihood function

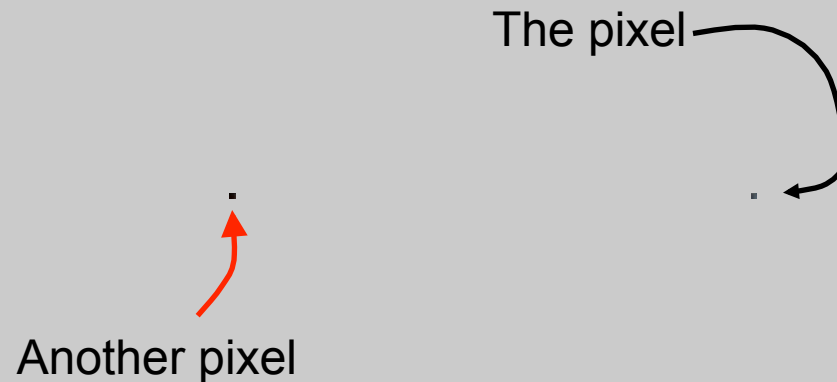
$$P(y = 1 \mid a, b) = ke^{-\frac{(1-ab)^2}{2\sigma^2}}$$

Prior probability

$$P(a, b) = ke^{-\frac{(a-b)^2}{2\sigma^2}} \text{ if } a > 0, b > 0 \\ = 0 \text{ otherwise}$$



# Statistical modeling of images



$$C(\Delta x, \Delta y) = \rho [\mathbf{I}(x + \Delta x, y + \Delta y), \mathbf{I}(x, y)]$$

# Gaussian model

We want a distribution that captures the correlation structure typical of natural images.

Let  $\mathbf{C}$  be the covariance matrix of the image

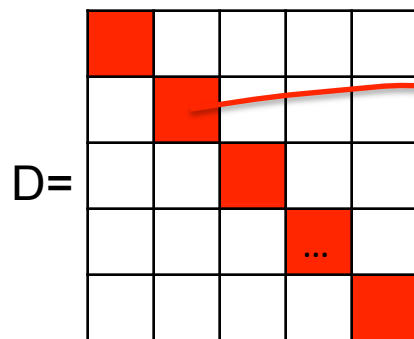
$$p(\mathbf{I}) = \exp\left(-\frac{1}{2}\mathbf{I}^T \mathbf{C}^{-1} \mathbf{I}\right) \quad \mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & c_2 & \vdots \\ & c_{n-1} & c_0 & c_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & c_2 \\ c_1 & \cdots & & c_{n-1} & c_0 \end{bmatrix}$$

Stationarity assumption: Symmetrical circulant matrix

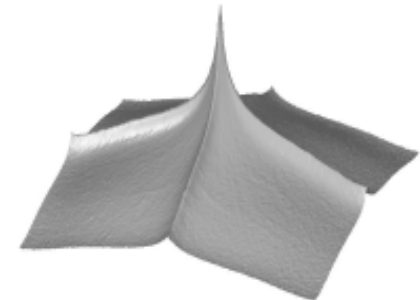
Diagonalization of circulant matrices:  $\mathbf{C} = \mathbf{E} \mathbf{D} \mathbf{E}^T$

The eigenvectors are the Fourier basis

The eigenvalues are the squared magnitude of the Fourier coefficients



$$|\hat{\mathbf{I}}(v)|^2 \approx \frac{1}{|v|^{2\alpha}}$$

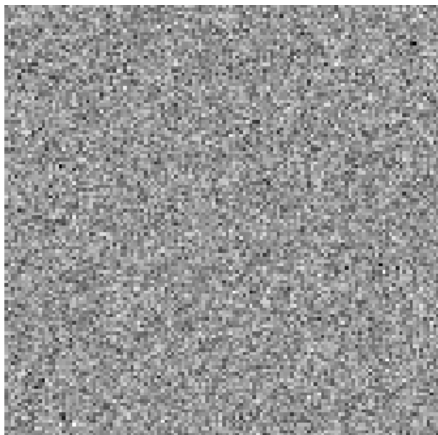


# Statistical modeling of images

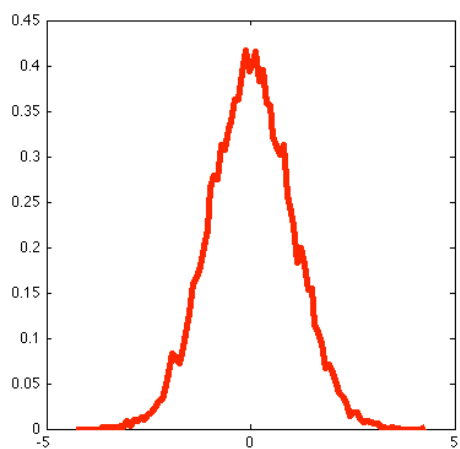
A small neighborhood



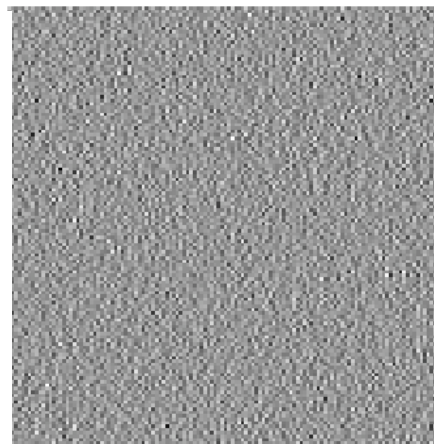
Image



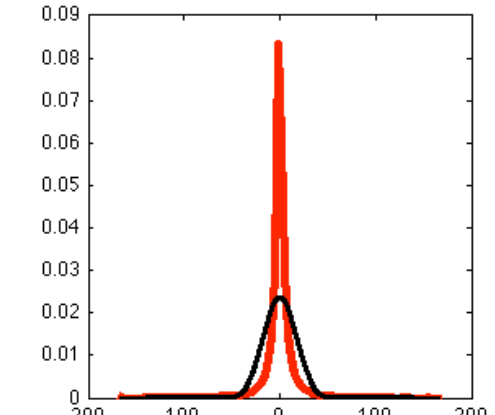
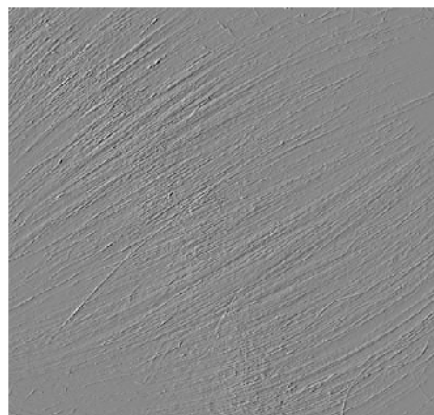
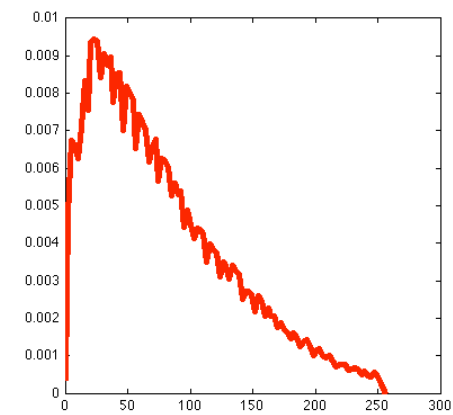
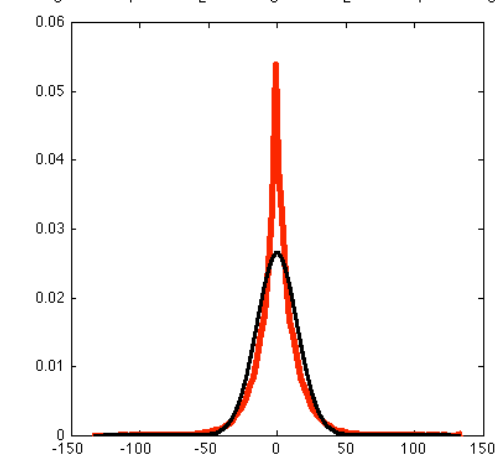
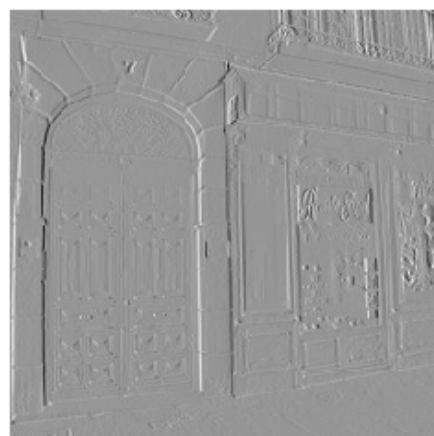
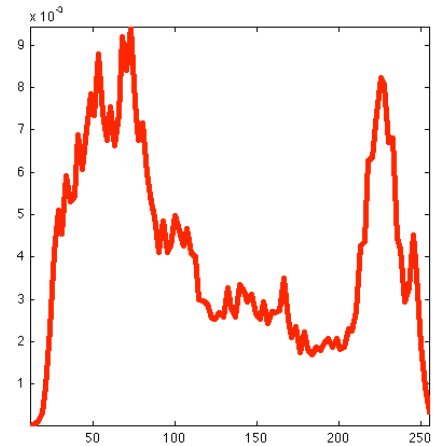
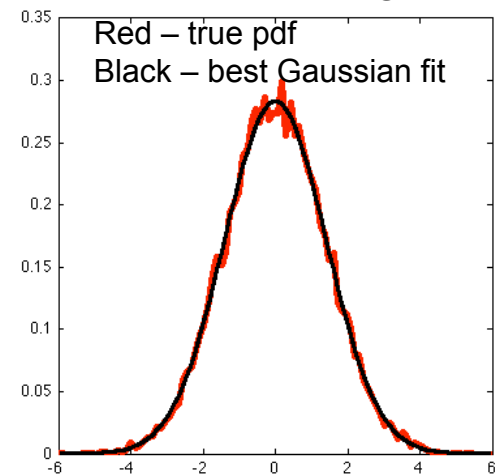
Intensity histogram



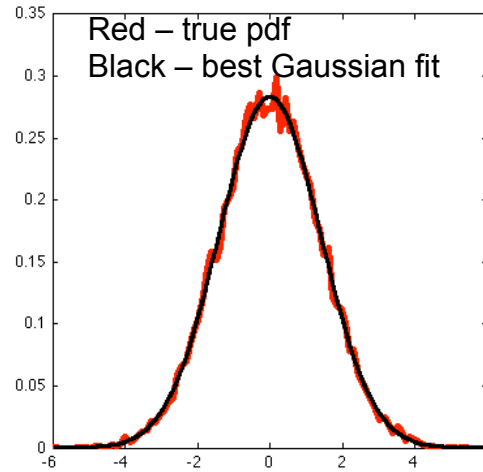
[1 -1] filter output



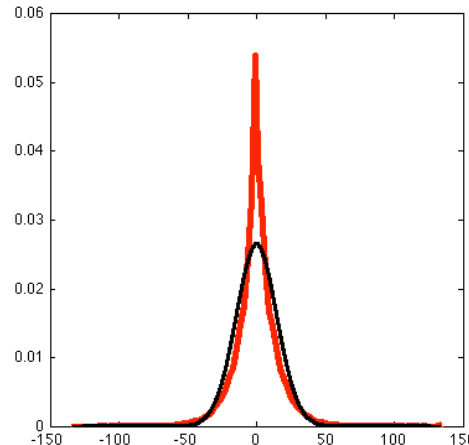
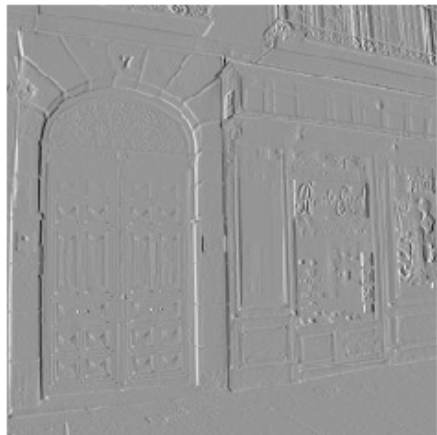
[1 -1] output histogram



# A model for the distribution of filter outputs



$$p(x) = \frac{\exp(-x^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$$



$$p(x) = \frac{\exp(-|x/s|^r)}{2s/r\Gamma(1/r)}$$

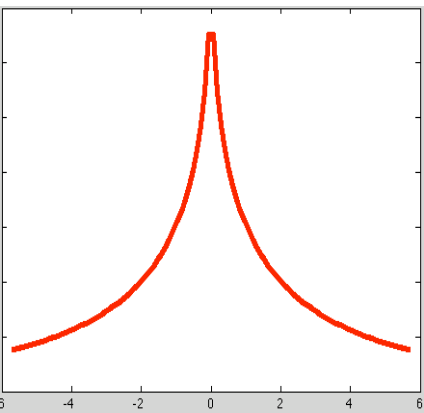
$$r \sim 0.8 (< 2)$$

**Note:** this is not a good model for ALL filter outputs

# Generalized Gaussian

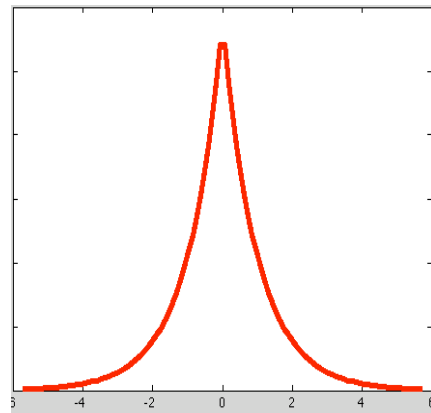
$$p(x) = \frac{\exp(-|x/s|^r)}{2s/r\Gamma(1/r)}$$

$r = 0.5$



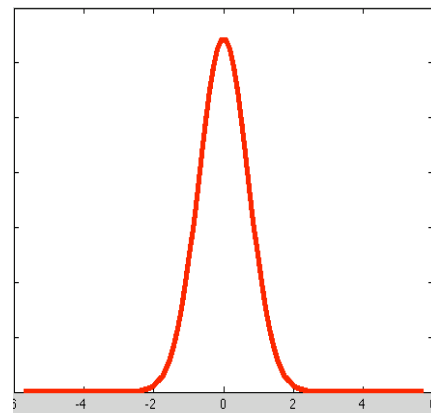
$r = 1$

Laplacian distribution

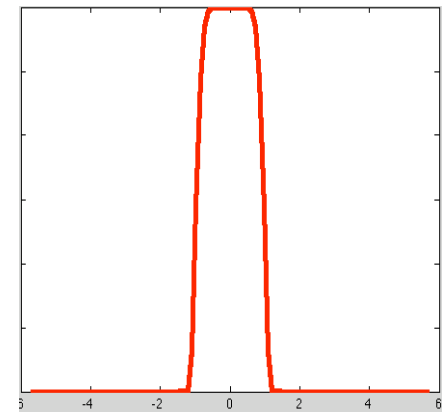


$r = 2$

Gaussian distribution



$r = 10$

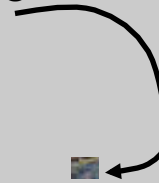


Uniform distribution  
 $r \rightarrow \infty$



# The wavelet marginal model

A small neighborhood

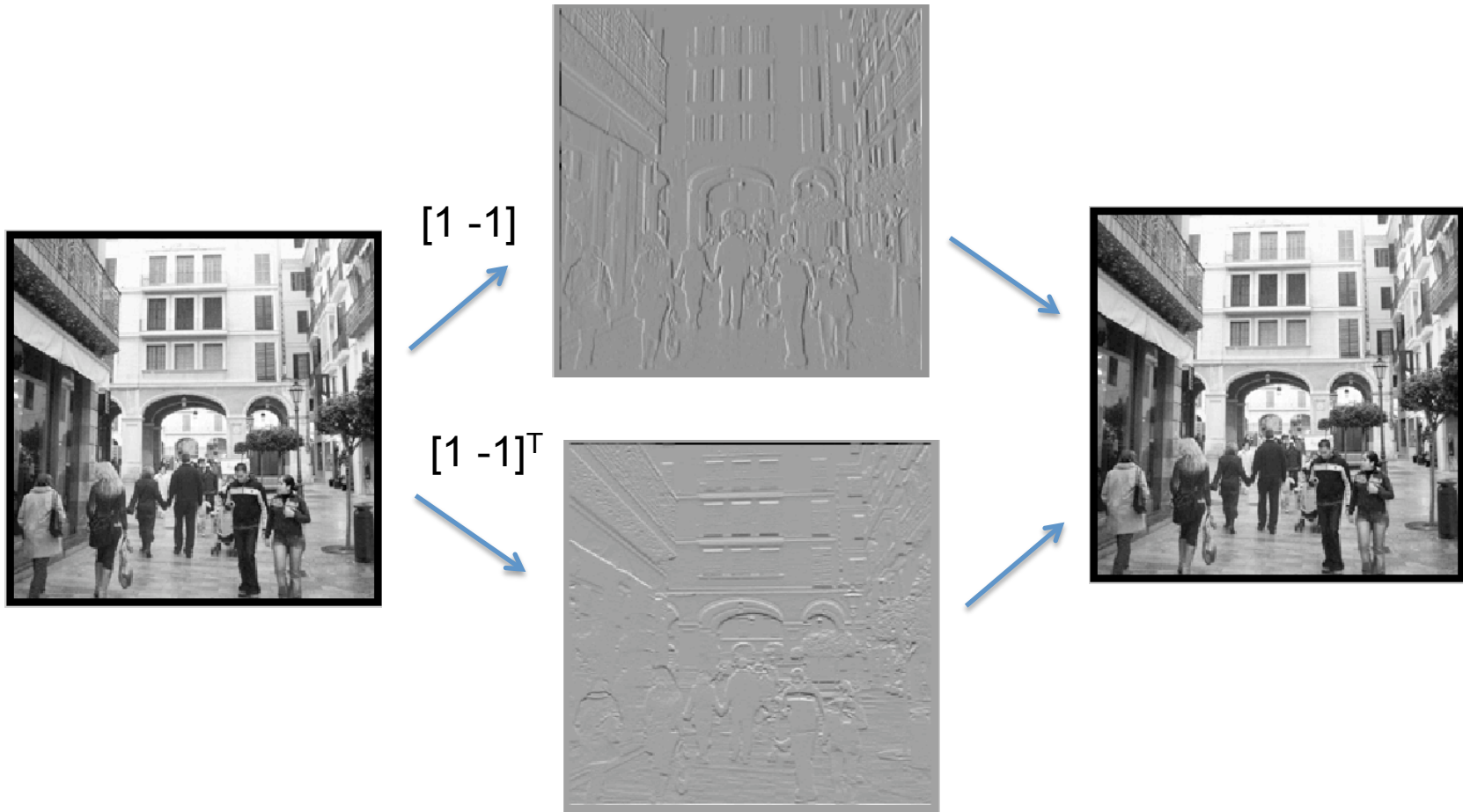


$$p(\mathbf{I}) = \prod_k \prod_{x,y} p(h_k(x,y))$$

All pixels and all outputs are independent

Filter outputs

# The wavelet marginal model



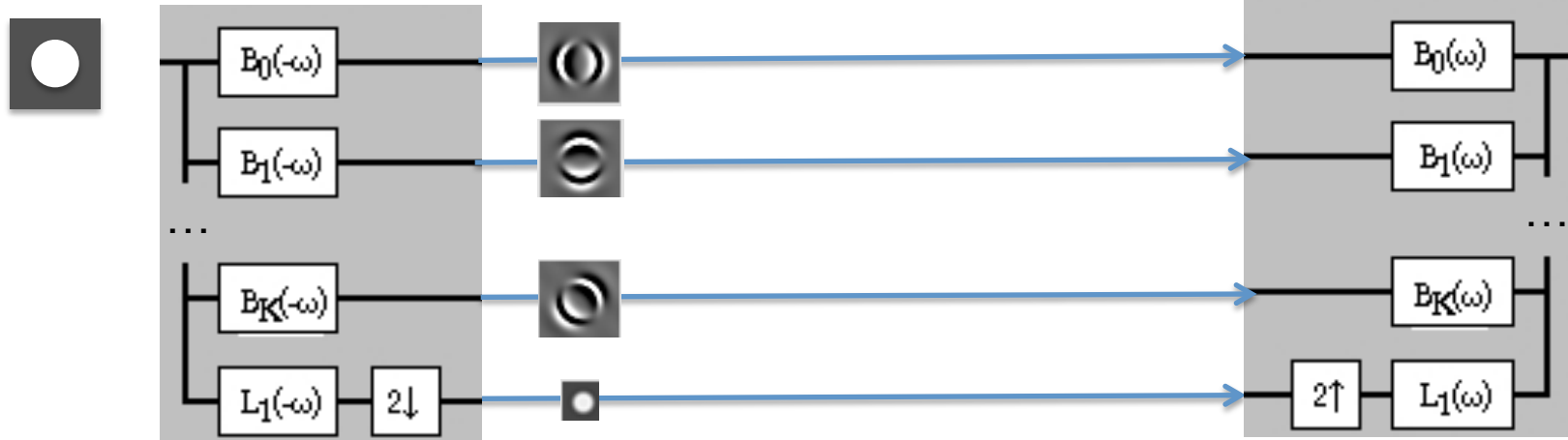
$$p(\mathbf{I}) = \prod_k \prod_{x,y} p(h_k(x,y))$$

# Steerable Pyramid

We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below

Decomposition

Reconstruction

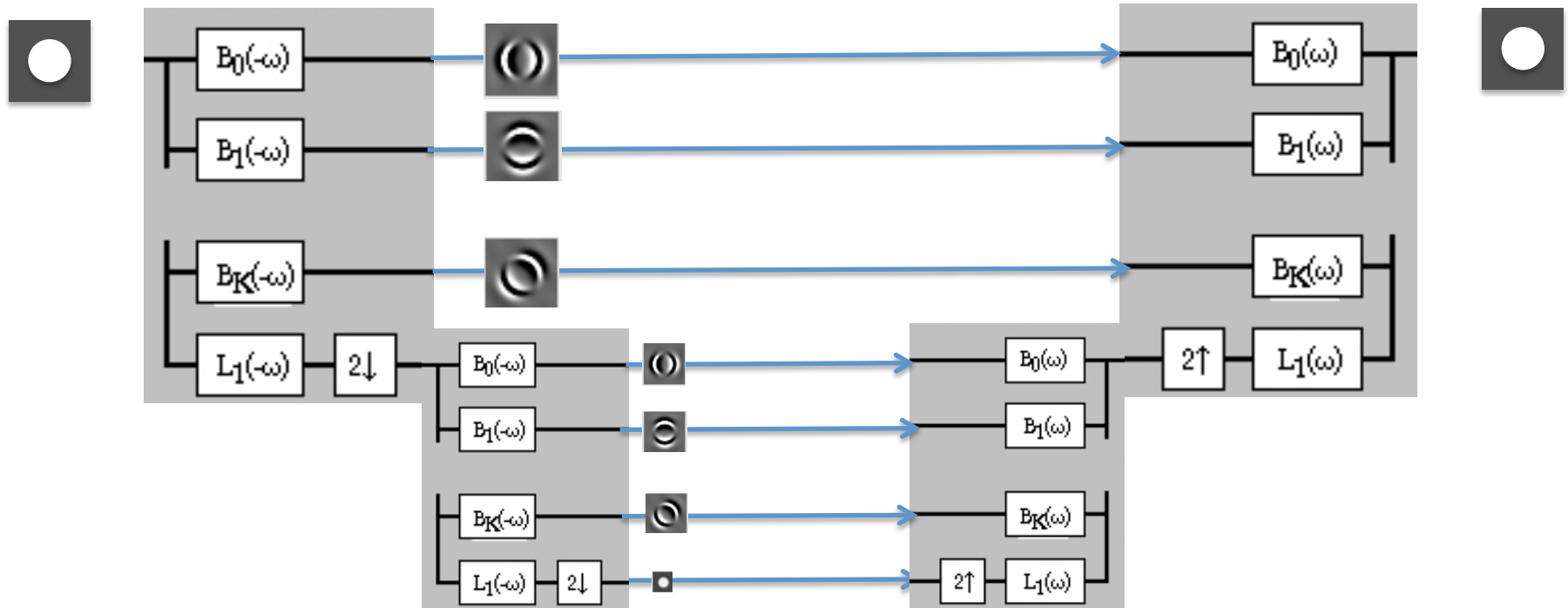


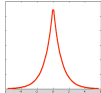
# Steerable Pyramid

We may combine Steerability with Pyramids to get a Steerable Laplacian Pyramid as shown below

Decomposition

Reconstruction



$$p(\mathbf{I}) = \prod_k \prod_{x,y} p(h_k(x,y))$$


# Sampling images

Gaussian model

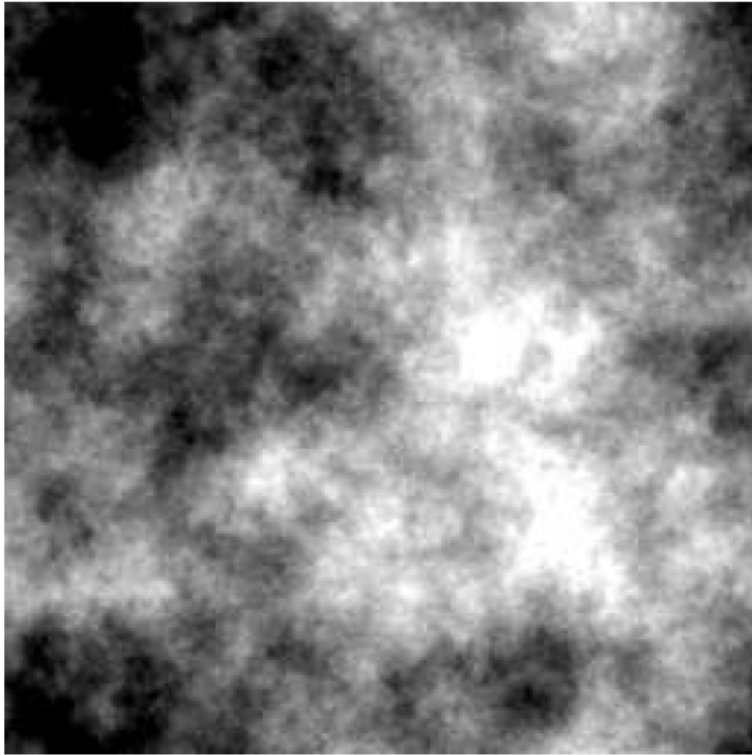


Fig. 3. Example image randomly drawn from the Gaussian spectral model, with  $\gamma = 2.0$ .

Wavelet marginal model

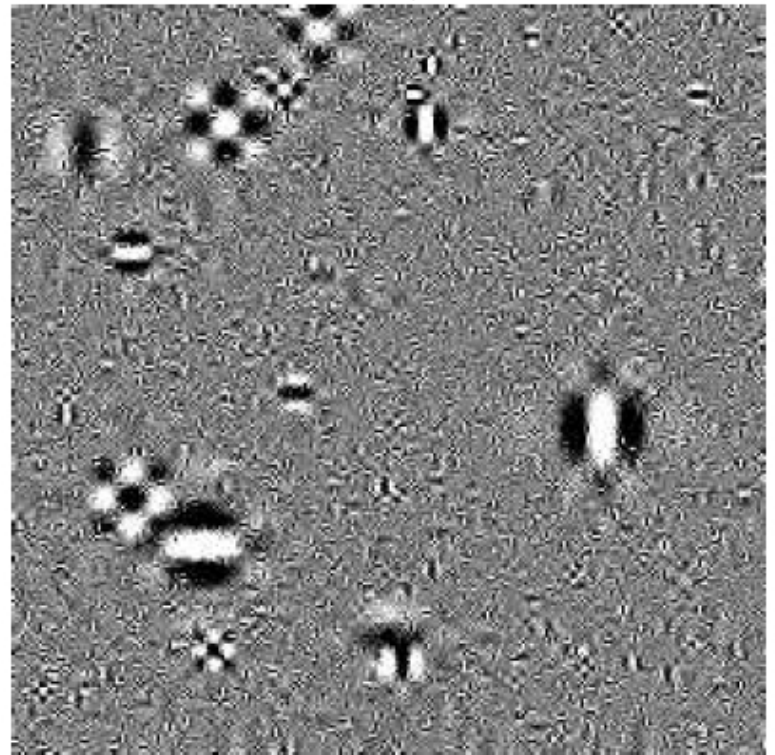


Fig. 6. A sample image drawn from the wavelet marginal model, with subband density parameters chosen to fit the image of Fig. 7.

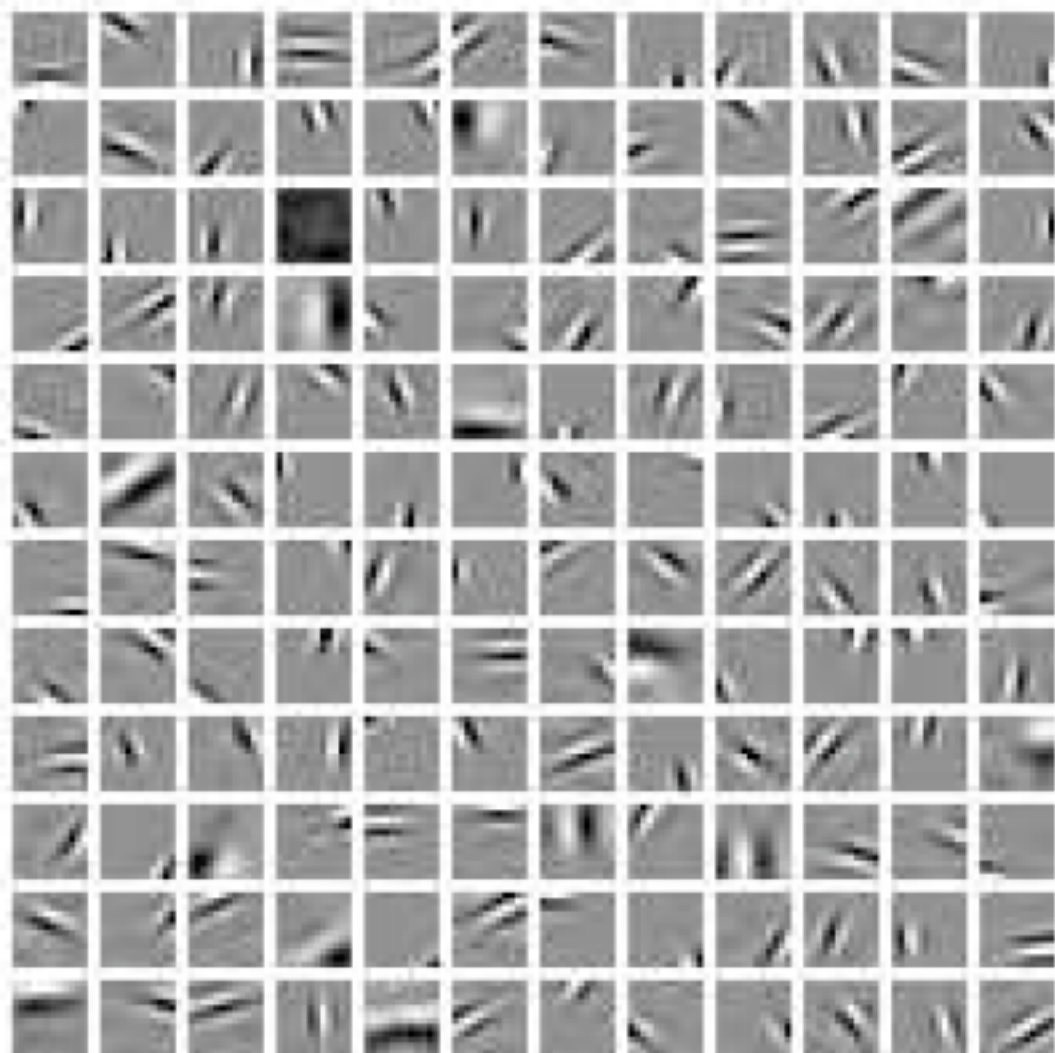
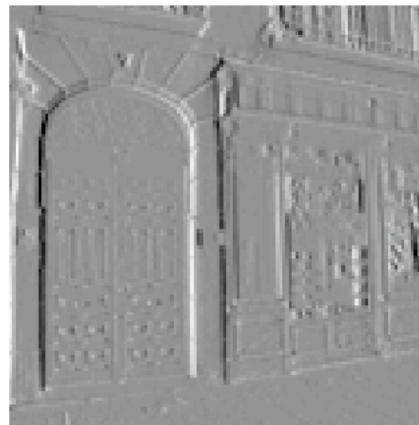


Fig. 5. Example basis functions derived by optimizing a marginal kurtosis criterion [see 35].

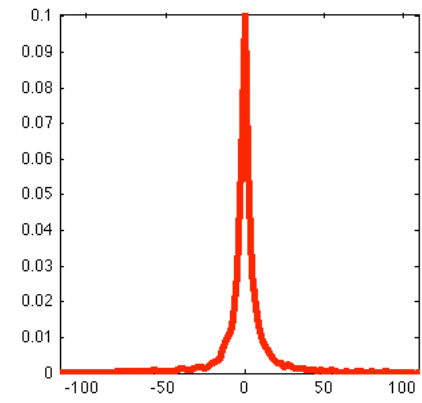
# Denoising



+

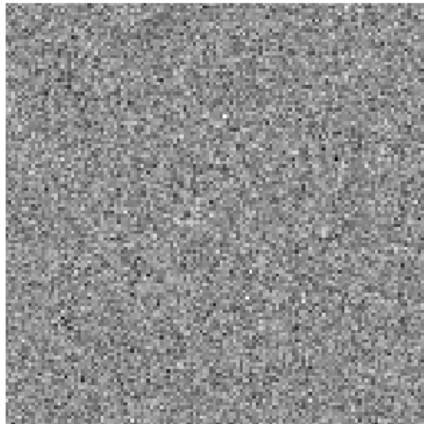


+

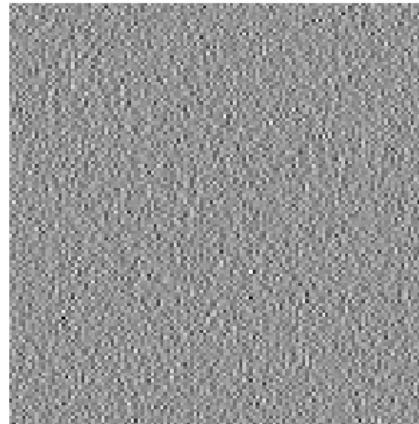


\*

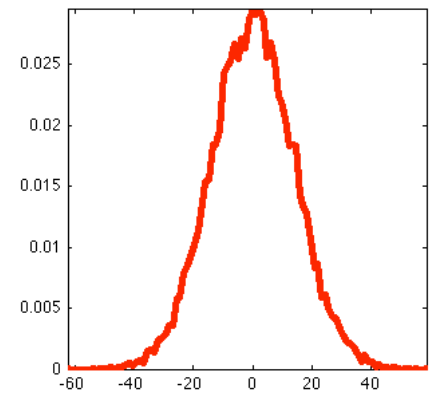
White  
Gaussian  
noise



||

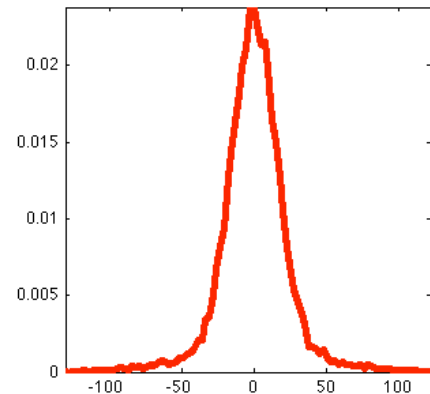
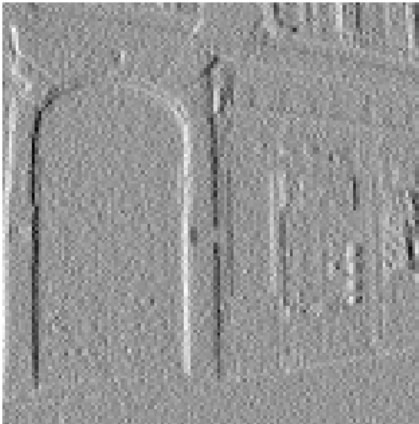


||



||

Noisy  
image



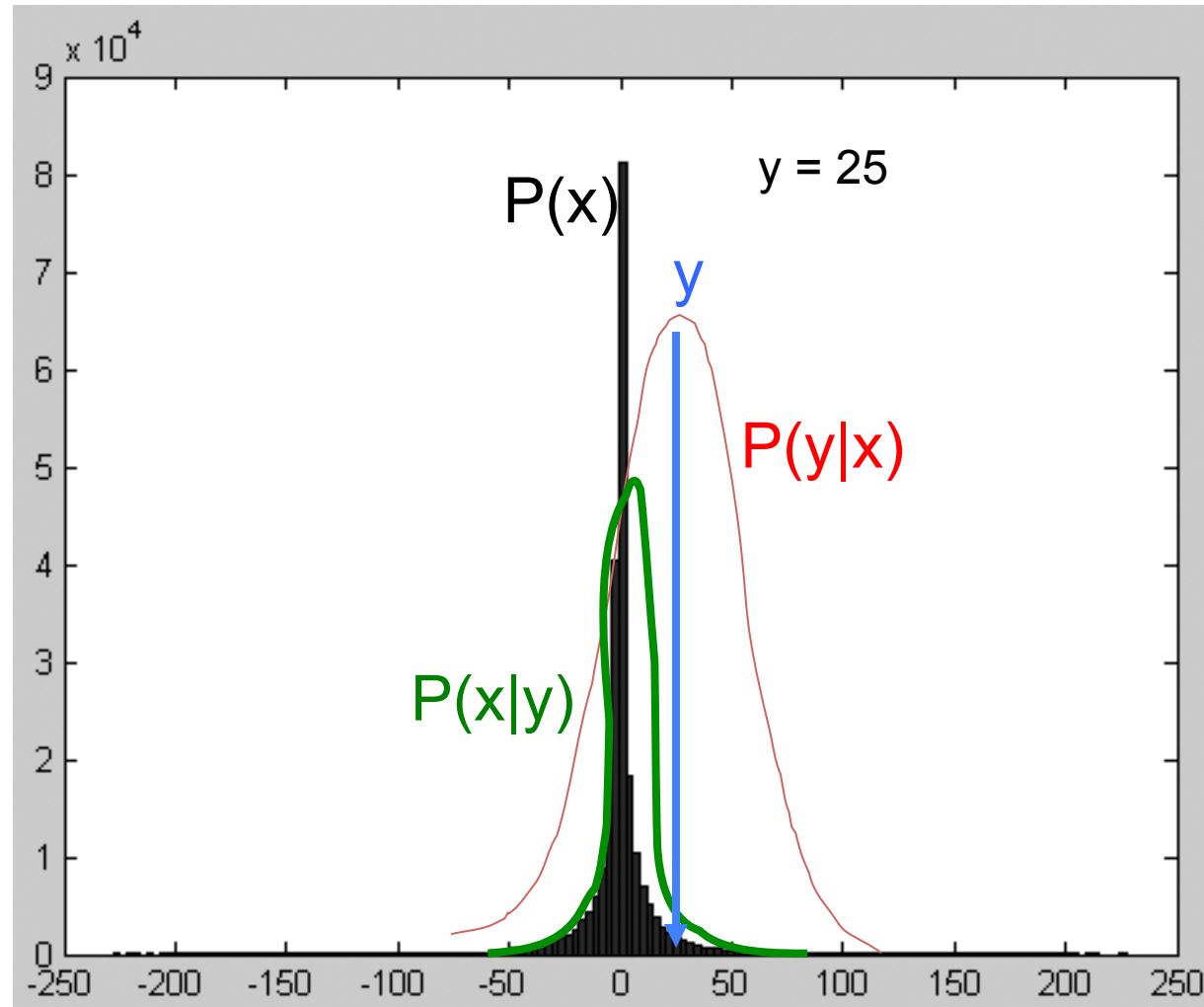
# Denoising with the marginal wavelet model

Let  $x$  = bandpassed image value before adding noise.

Let  $y$  = noise-corrupted observation.

By Bayes theorem

$$P(x|y) \sim P(y|x) P(x)$$





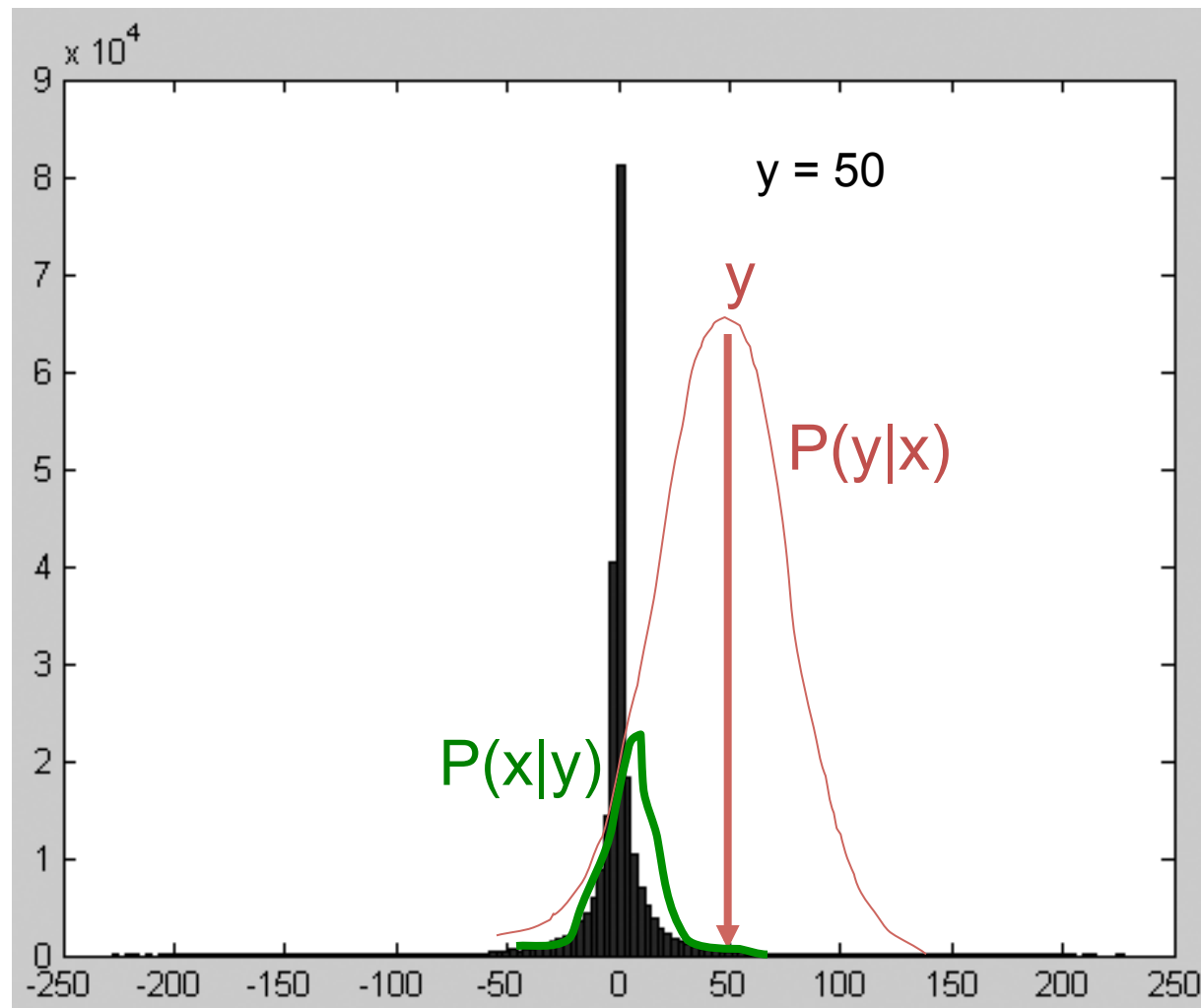
# Denoising with the marginal wavelet model

Let  $x$  = bandpassed image value before adding noise.

Let  $y$  = noise-corrupted observation.

By Bayes theorem

$$P(x|y) \sim P(y|x) P(x)$$



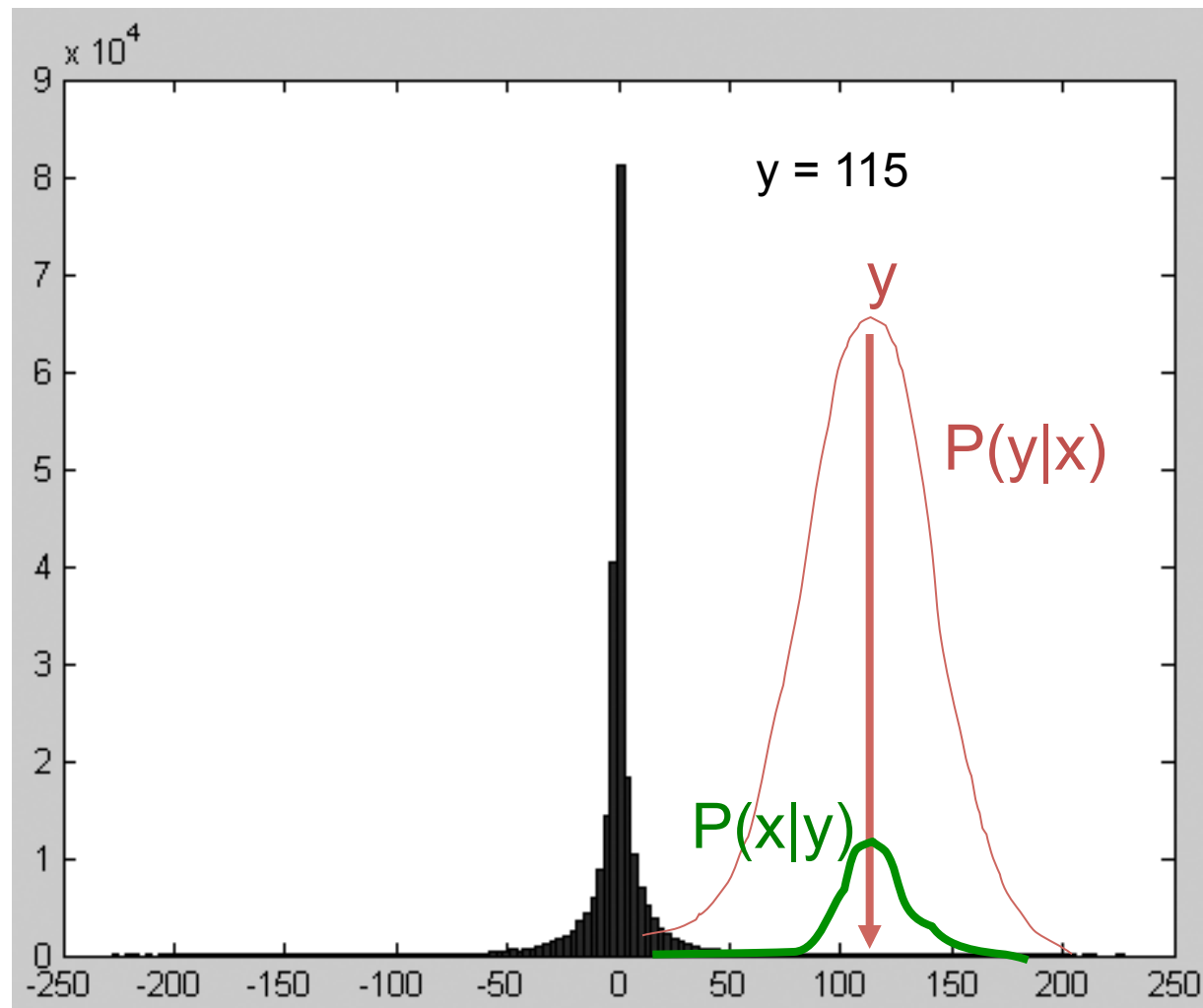
# Denoising with the marginal wavelet model

Let  $x$  = bandpassed image value before adding noise.

Let  $y$  = noise-corrupted observation.

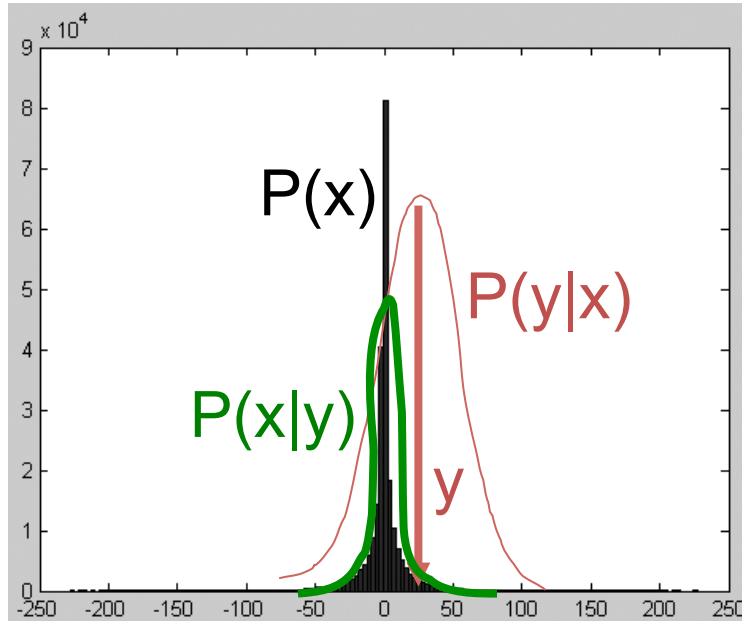
By Bayes theorem

$$P(x|y) \sim P(y|x) P(x)$$

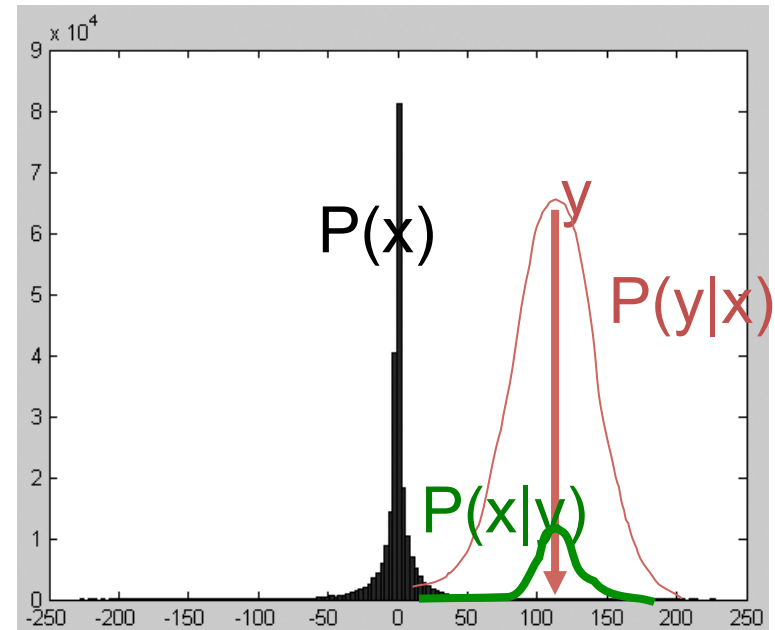


# Denoising with the marginal wavelet model

$y = 25$



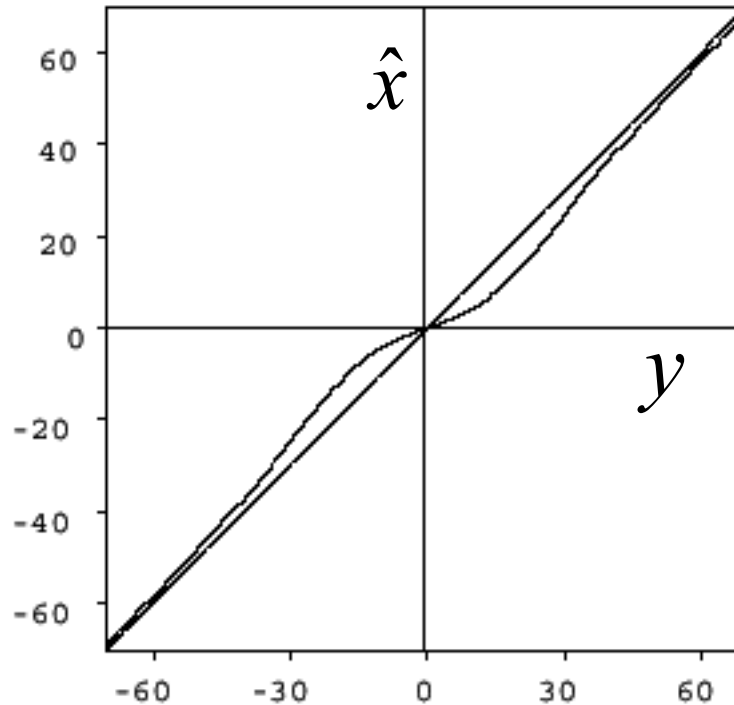
$y = 115$



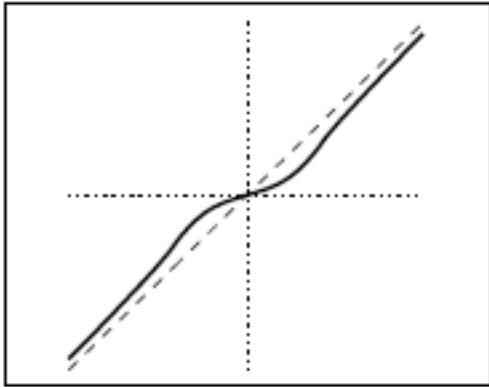
For small  $y$ : probably it is due to noise and  $y$  should be set to 0

For large  $y$ : probably it is due to an image edge and it should be kept untouched

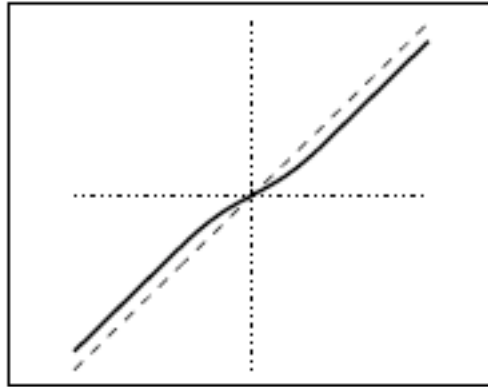
MAP estimate,  $\hat{x}$ , as function of  
observed coefficient value,  $y$



**Figure 2:** Bayesian estimator (symmetrized) for the signal and noise histograms shown in figure 1. Superimposed on the plot is a straight line indicating the identity function.

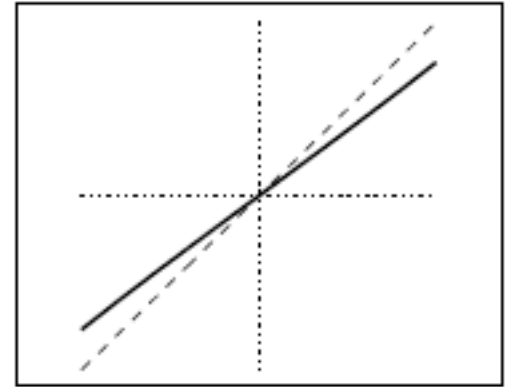


$r = 0.5$



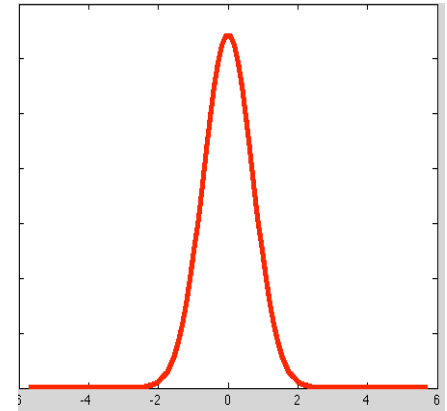
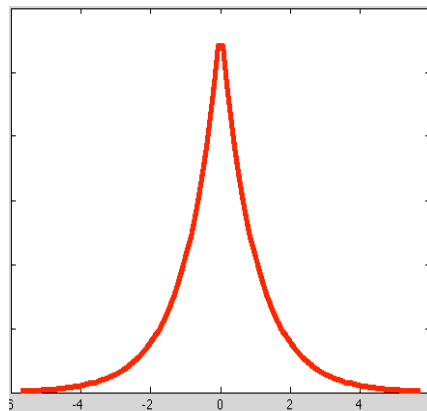
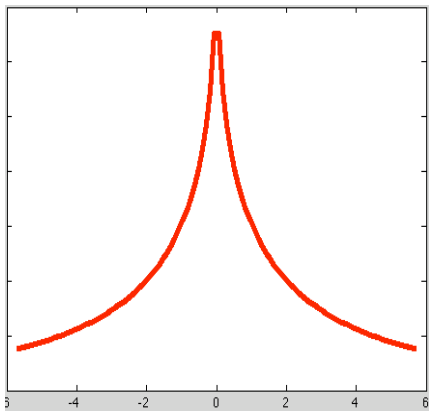
$r = 1$

Laplacian distribution



$r = 2$

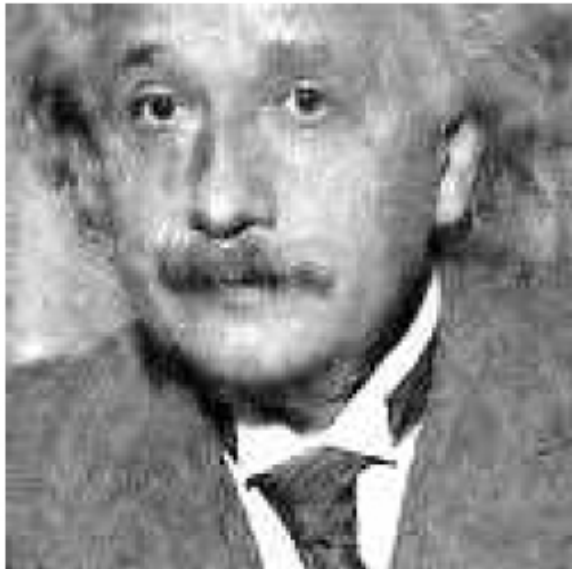
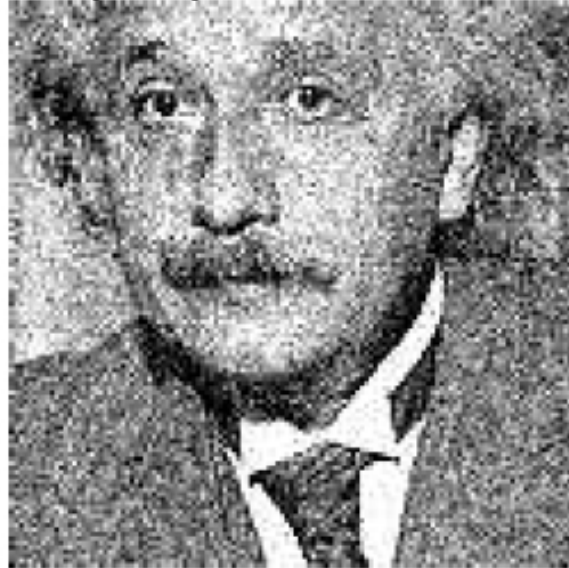
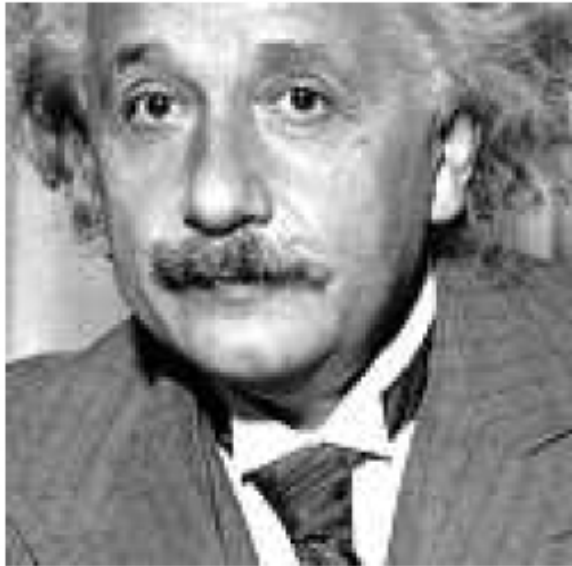
Gaussian distribution



original

With Gaussian noise of  
std. dev. 21.4 added,  
giving PSNR=22.06

(1) Denoised with  
Gaussian model,  
PSNR=27.87



(2) Denoised  
with wavelet  
marginal model,  
PSNR=29.24

# Gaussian scale mixtures

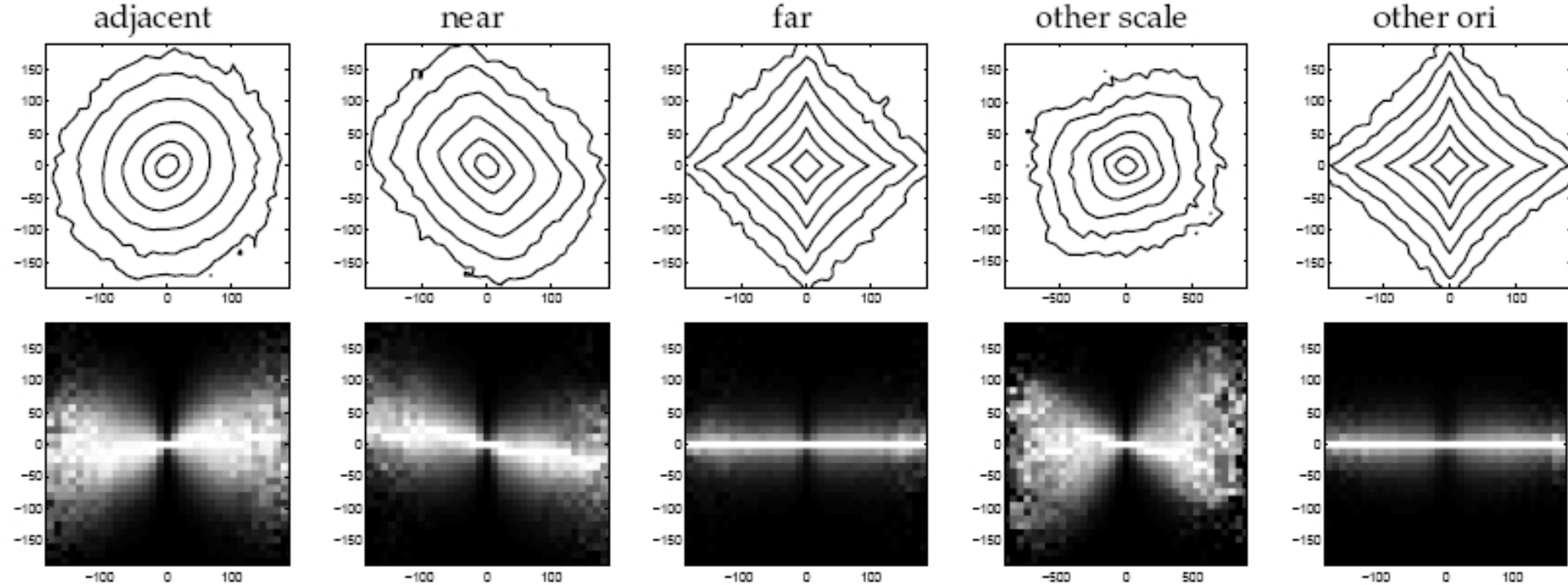


Note correlations between the amplitudes of each wavelet subband.

Fig. 7. Amplitudes of multi-scale wavelet coefficients for the "Einstein" image. Each subimage shows coefficient amplitudes of a subband obtained by convolution with a filter of a different scale and orientation, and subsampled by an appropriate factor. Coefficients that are spatially near each other within a band tend to have similar amplitudes. In addition, coefficients at different orientations or scales but in nearby (relative) spatial positions tend to have similar amplitudes.

# Statistics of pairs of wavelet coefficients

## Contour plots of the joint histogram of various wavelet coefficient pairs



## Conditional distributions of the corresponding wavelet pairs

Fig. 8. Empirical joint distributions of wavelet coefficients associated with different pairs of basis functions, for a single image of a New York City street scene (see Fig. 1 for image description). The top row shows joint distributions as contour plots, with lines drawn at equal intervals of log probability. The three leftmost examples correspond to pairs of basis functions at the same scale and orientation, but separated by different spatial offsets. The next corresponds to a pair at adjacent scales (but the same orientation, and nearly the same position), and the rightmost corresponds to a pair at orthogonal orientations (but the same scale and nearly the same position). The bottom row shows corresponding conditional distributions: brightness corresponds to frequency of occurrence, except that each column has been independently rescaled to fill the full range of intensities.



# Gaussian scale mixtures

$$P(\vec{x}) = \int \frac{\exp(-\frac{1}{2} \vec{x}^T (z\Lambda)^{-1} \vec{x})}{(2\pi)^{N/2} |z\Lambda|^{1/2}} P_z(z) dz$$

Wavelet coefficient probability

A mixture of Gaussians of scaled covariances

$z$  is a spatially varying hidden variable that can be used to  
 (a) Create the non-gaussian histograms from a mixture of Gaussian densities, and  
 (b) model correlations between the neighboring wavelet coefficients.

Gaussian scale mixture model simulation

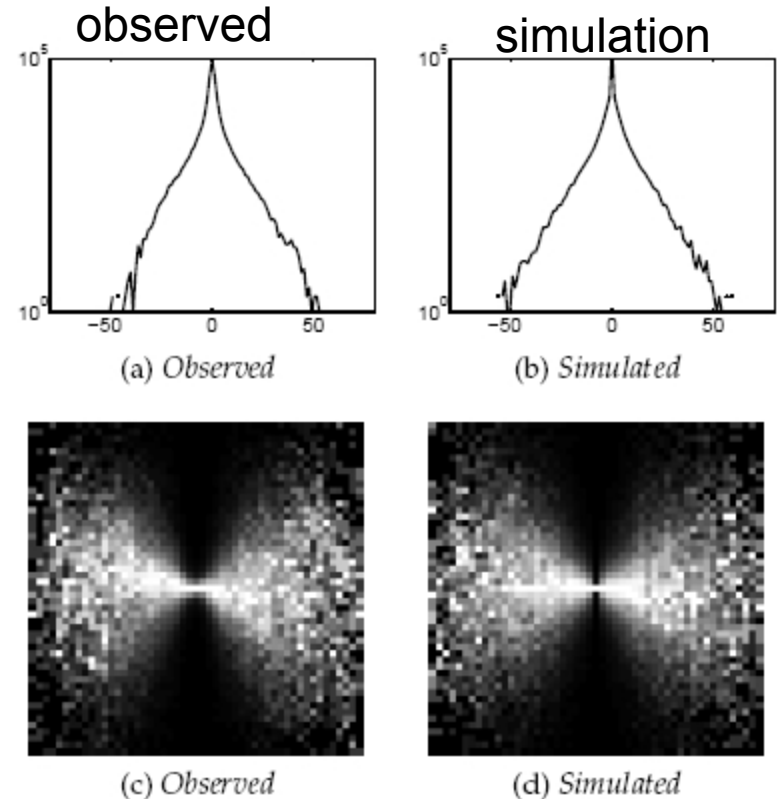
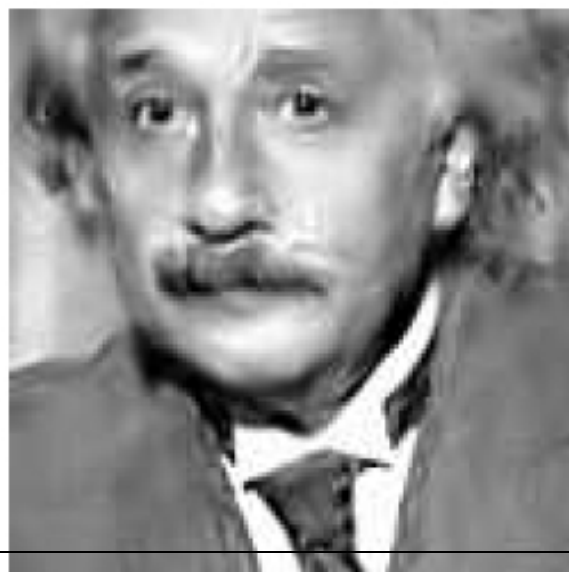
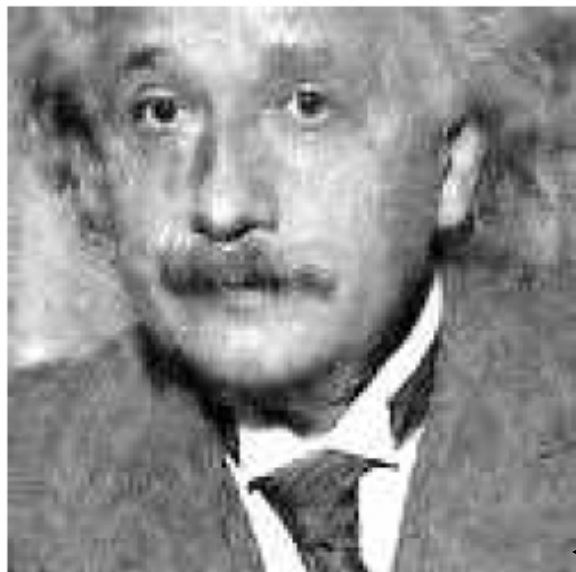
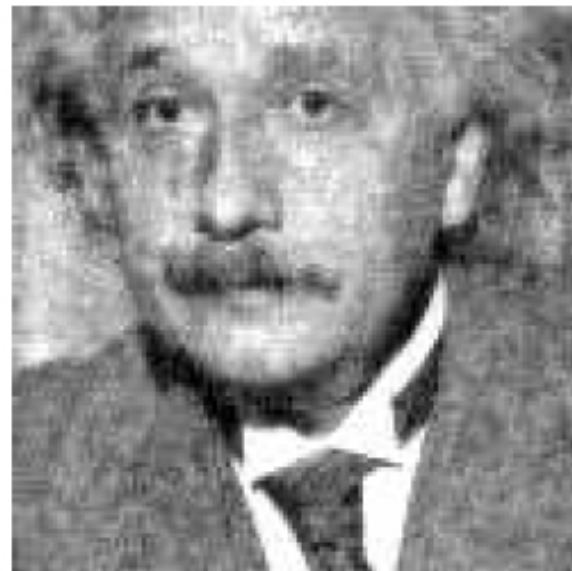
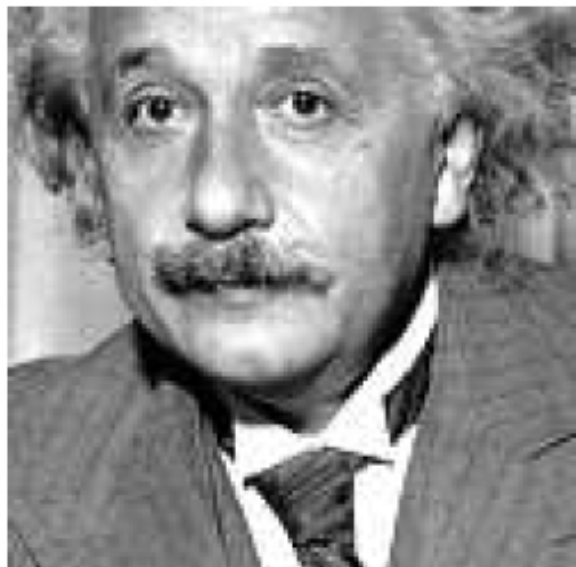


Fig. 9. Comparison of statistics of coefficients from an example image subband (left panels) with those generated by simulation of a local GSM model (right panels).

original

With Gaussian noise of  
std. dev. 21.4 added,  
giving PSNR=22.06

(1) Denoised with  
Gaussian model,  
PSNR=27.87



(3) Denoised with  
Gaussian scale  
mixture model,  
PSNR=30.86

(2) Denoised with wavelet  
marginal model,  
PSNR=29.24

# Separating reflections from a single image using local features

Anat Levin

Assaf Zomet

Yair Weiss



(a)



(b)



(c)



(d)



(e)



(f)



(g)

very simple cost function: it favors decompositions which have a small number of edges and corners. Surprisingly, this simple cost function gives the “right” decompositions for challenging real images.<sup>1</sup>



(a)



(b)



(c)



(d)



(e)



Figure 2: An input image and some decompositions

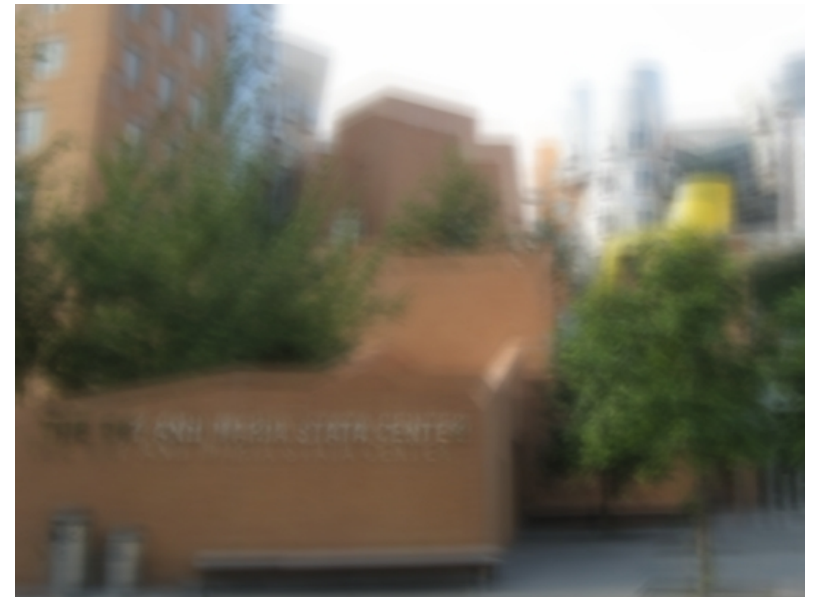
Figure 1: (a) Original input image (constructed by summing the two images in b). (b) the correct decomposition. (c)-(g) alternative possible decompositions. Why should the decomposition in (b) be favored?

# Applications

- Detecting fake images



- Camera shake removal



# Visual Worlds



Prof. Hany Farid,  
Dartmouth University

# How do you tell if an image is fake?

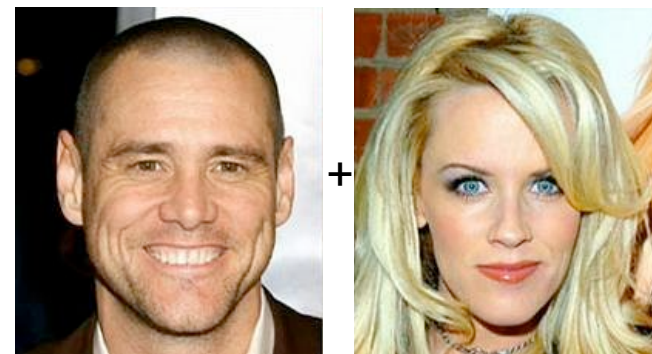
**Real or Fake?**  
What do you think? Is the photo fake? Or could it possibly be real?



EMAIL  
LINK TO  
DIGG  
3 SHARE  
TWEET  
SUBMIT  
FARK IT

Real or Fake

Back to Archive



=



<http://www.life.com/archive/realfake>

Image circulated on internet

## Fonda Speaks To Vietnam Veterans At Anti-War Rally



Actress And Anti-War Activist Jane Fonda Speaks to a crowd of Vietnam Veterans as Activist and former Vietnam Vet John Kerry (LEFT) listens and prepares to speak next concerning the war in Vietnam (AP Photo)

<http://www.cs.dartmouth.edu/farid/publications/deception09.pdf>

<http://www.cs.dartmouth.edu/farid/publications/significance06.pdf>

# The source images



**Update: [Fonda, Kerry and Photo Fakery](#)** (free reg. required) - Photographer Ken Light describes the experience of discovering his 1970 photograph of John Kerry circulating in altered form on the Internet. "As far as I know, John Kerry never shared a demonstration podium with Jane Fonda, and the fact that a widely circulated photo showed him doing so — until it was exposed in recent weeks as a hoax — tells us more about the troublesome combination of Photoshop and the Internet than it does about the prospective Democratic candidate for president." (*Washington Post*)



IEEE Transactions on Signal Processing, 53(2):845-850, 2005

# How Realistic is Photorealistic?

Siwei Lyu and Hany Farid  
Department of Computer Science  
Dartmouth College  
Hanover, NH 03755  
Email: {lyu, farid}@cs.dartmouth.edu

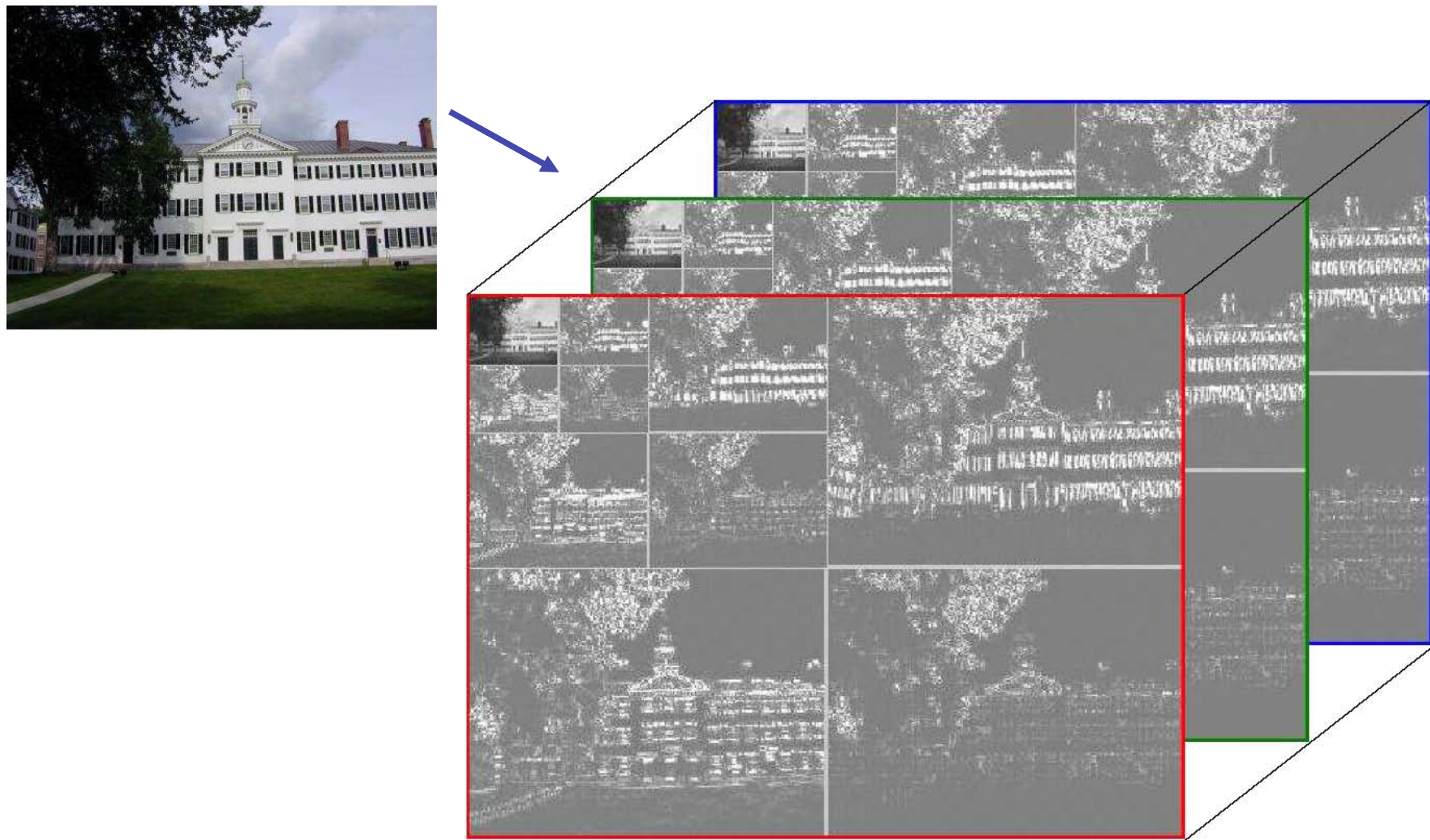
*Abstract*—Computer graphics rendering software is capable of generating highly photorealistic images that can be impossible to differentiate from photographic images. As a result, the unique stature of photographs as a definitive recording of events is being diminished (the ease with which digital images can be manipulated is, of course,

There has been some work in evaluating the photorealism of computer graphics rendered images from a human perception point of view (e.g., [10], [9], [11]). To our knowledge, however, no computational techniques exist to differentiate between photographic and photorealistic images (a method for differentiating between photo

# Input image

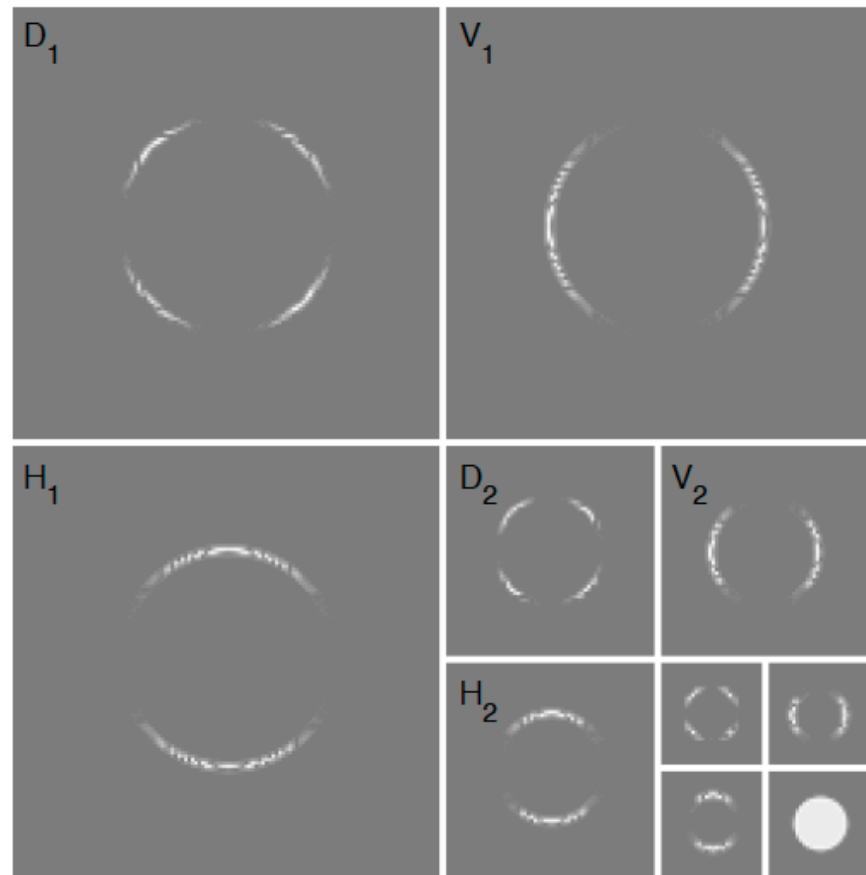
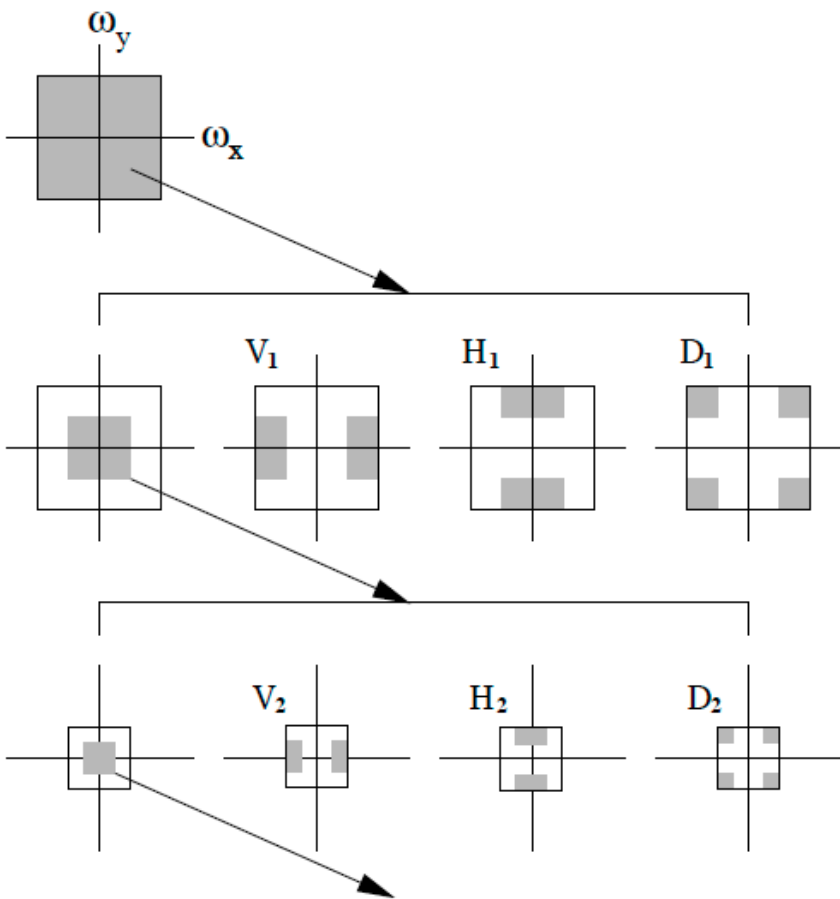


# Representation of color input image in wavelet subbands



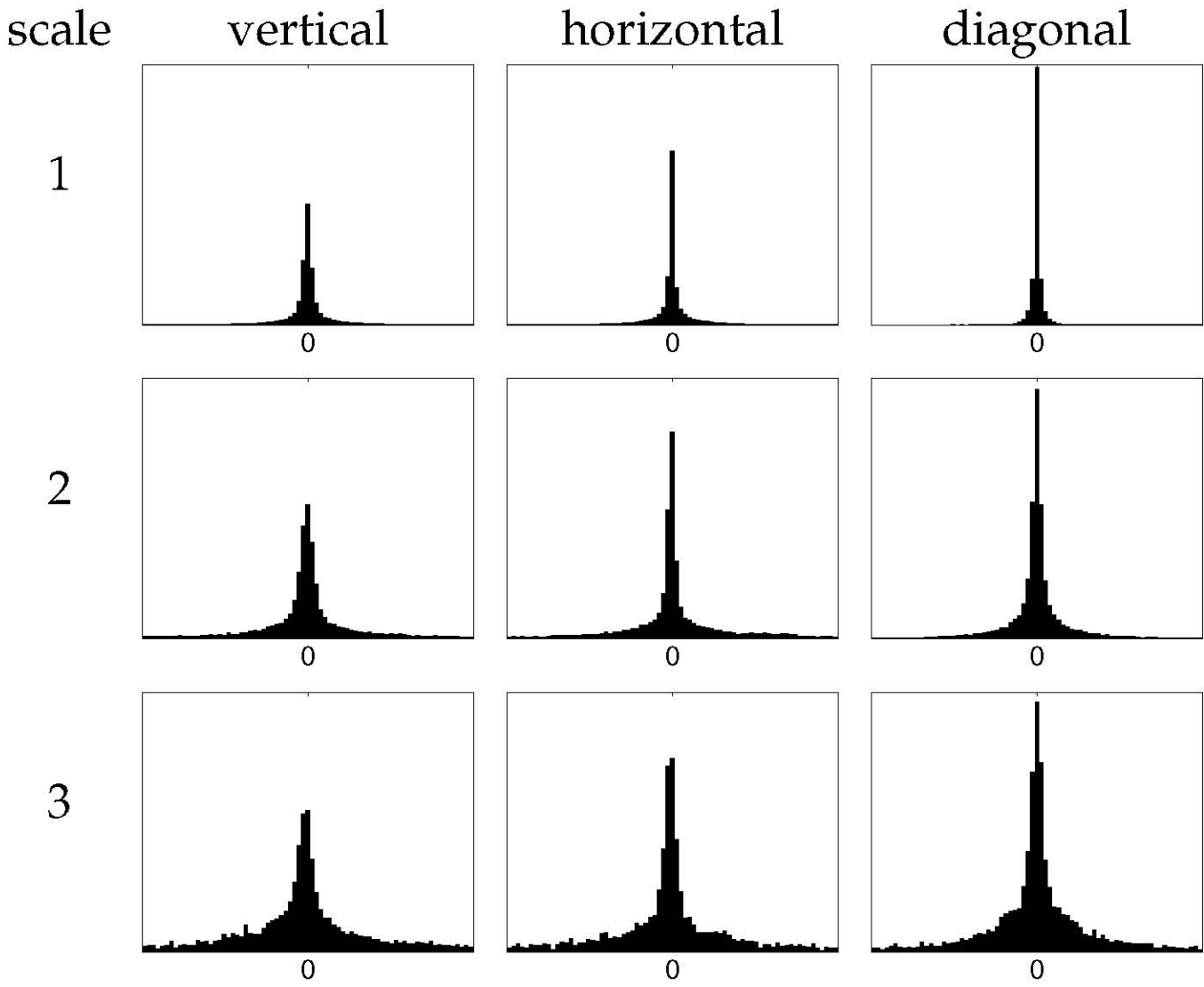
# Filter bank

## Separable Quadrature Mirror Filters



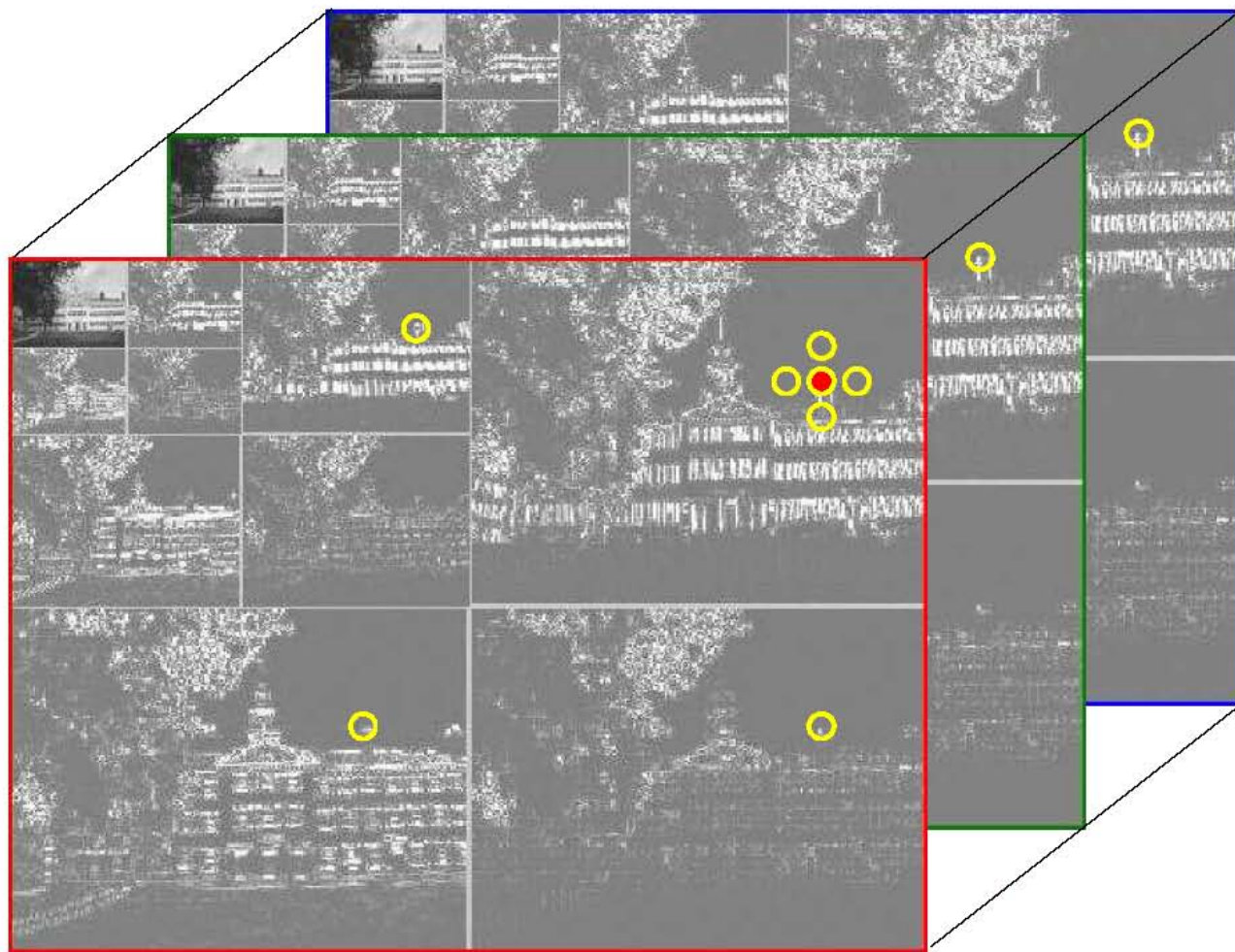
Each output is called *subband*

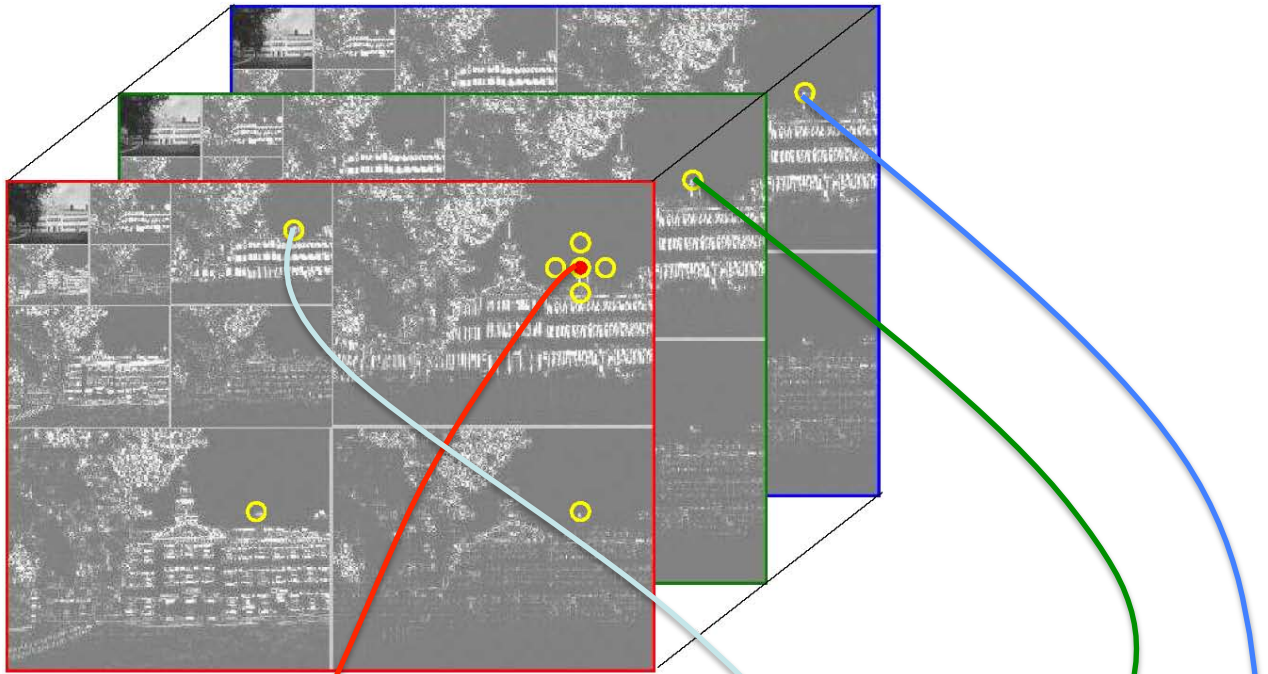
# Histograms of wavelet subband coefficients



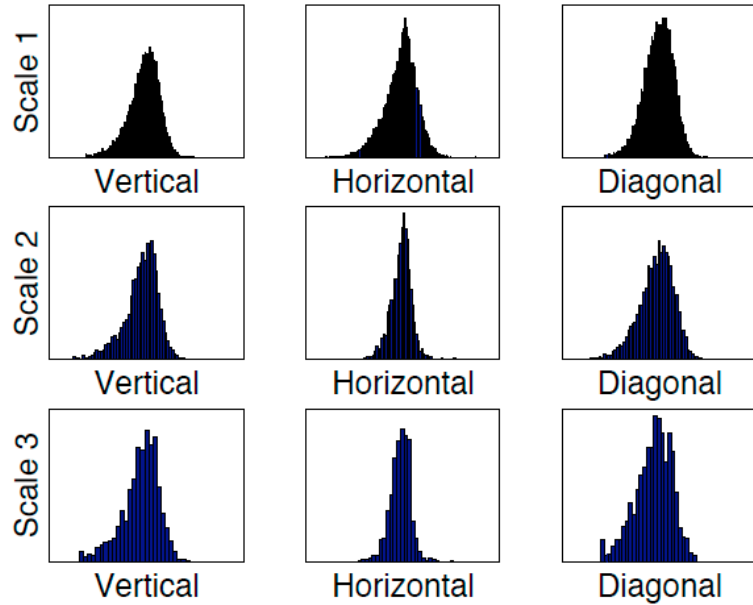
mean ( $\mu$ ), variance ( $\mu_2$ ), skewness ( $\mu_3/\sigma^3$ ), kurtosis ( $\mu_4/\sigma^4$ )

There are correlations between subband coefficients





$$\begin{aligned}
 V_i(x, y) = & w_1 V_i(x - 1, y) + w_2 V_i(x + 1, y) + w_3 V_i(x, y - 1) \\
 & + w_4 V_i(x, y + 1) + w_5 V_{i+1}(x/2, y/2) + w_6 H_i(x, y) \\
 & + w_7 D_i(x, y) + w_8 V_i(x, y) + w_9 V_i(x, y)
 \end{aligned}$$

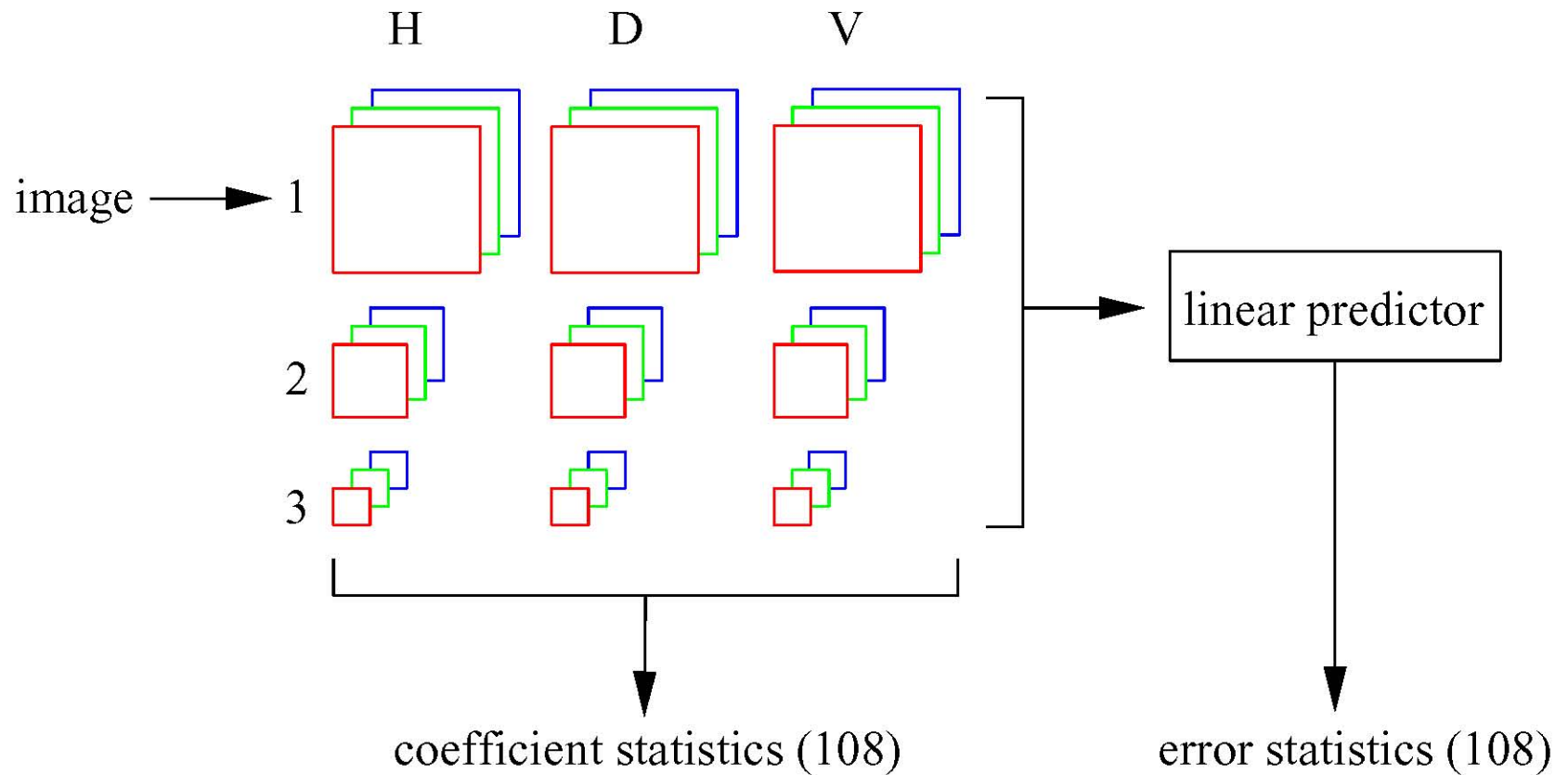


**Figure 2.10:** A natural image (left) and the histograms of the linear prediction errors of coefficient magnitudes for all subbands in a three-scale QMF pyramid decomposition of the image on the left.

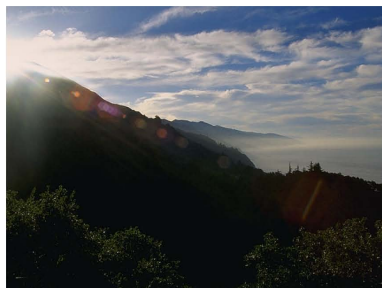
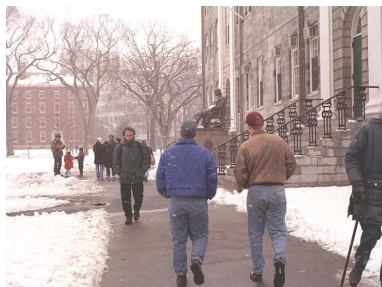
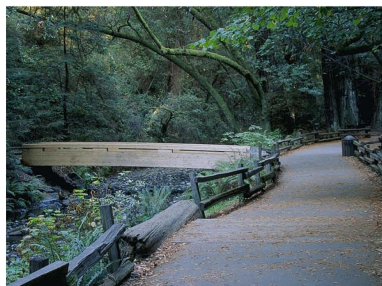
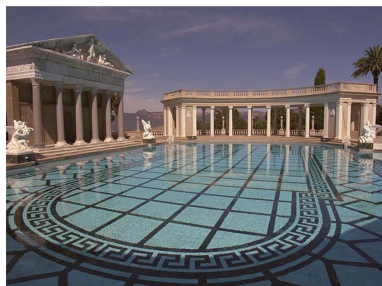
Hypothesis: there is something different in the correlation between wavelet coefficients between real images and computer generated images.



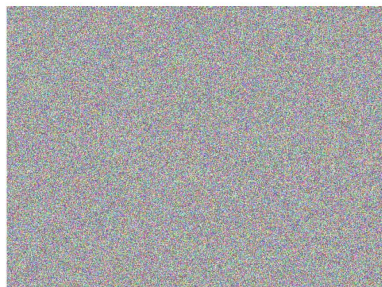
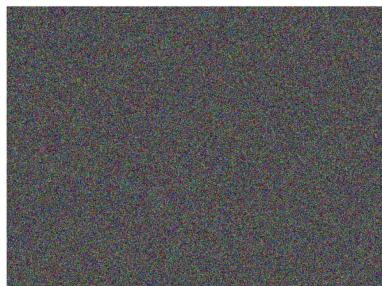
# Summary of features used for image classification



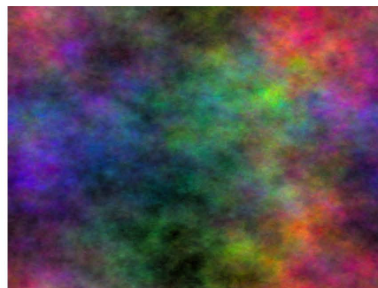
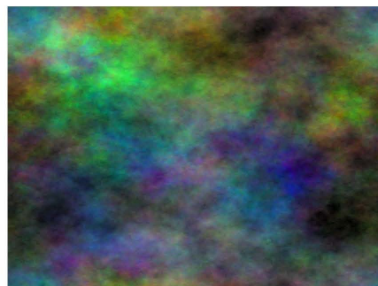
natural



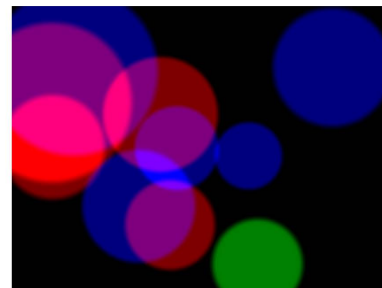
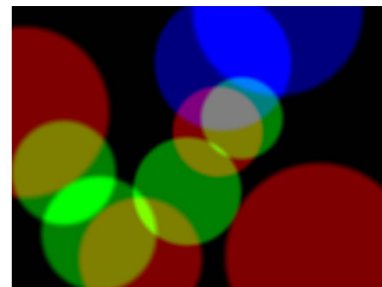
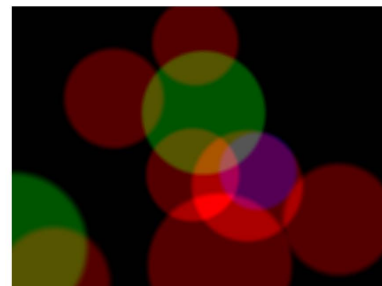
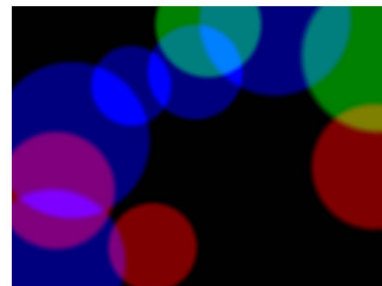
noise



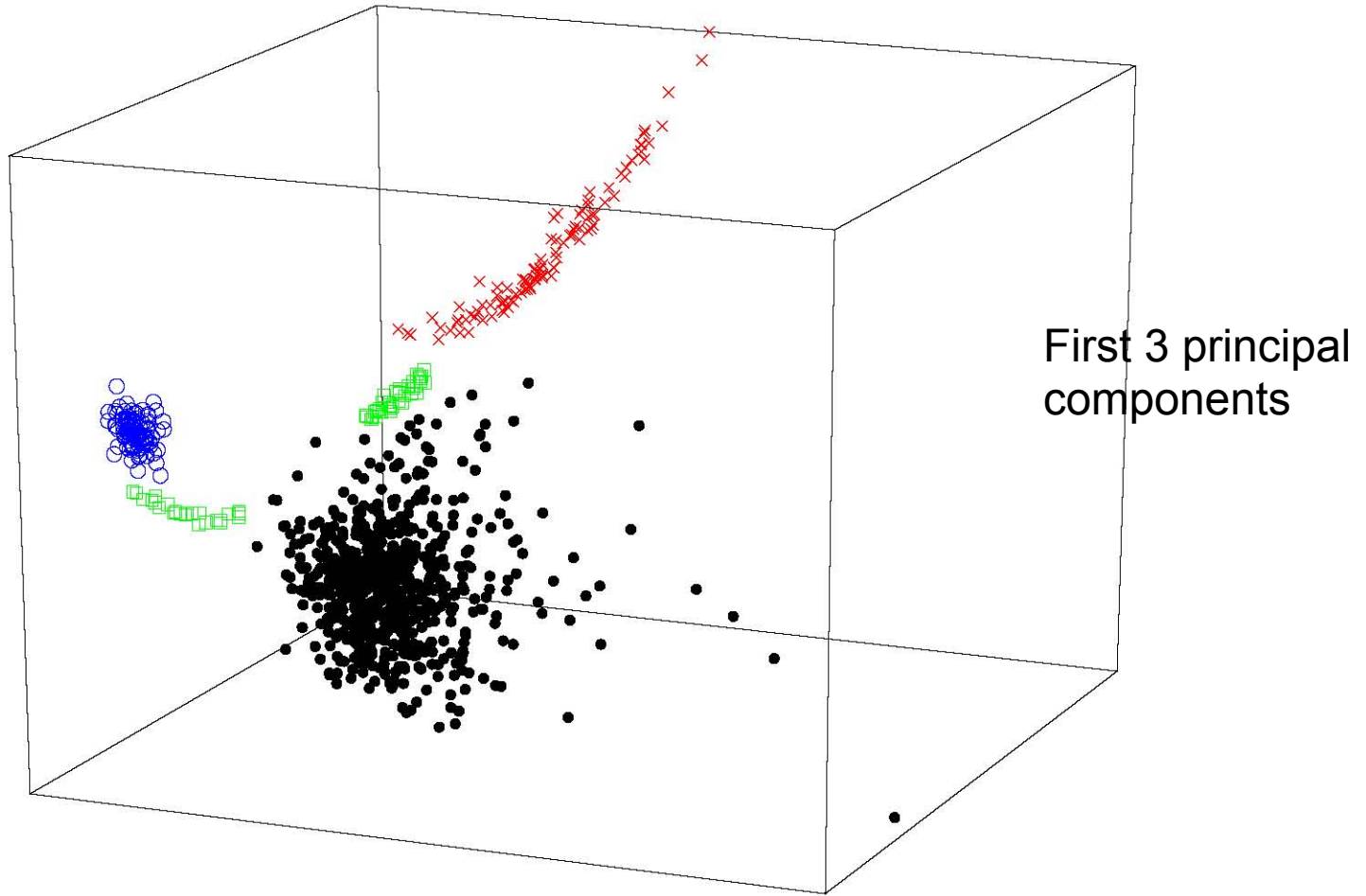
fractal



discs



Projection of measured features into a 3-d space: well separated even in that low-dimensional space



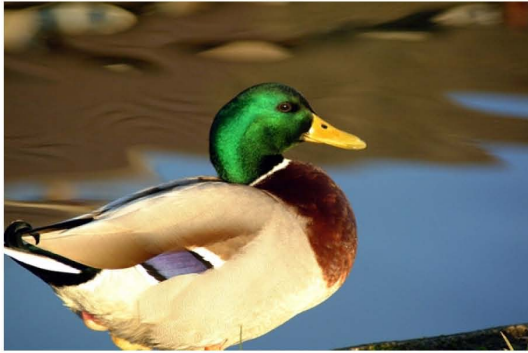
noise

fractal

discs

natural

# Photographic training set: downloaded from [www.freefoto.com](http://www.freefoto.com)



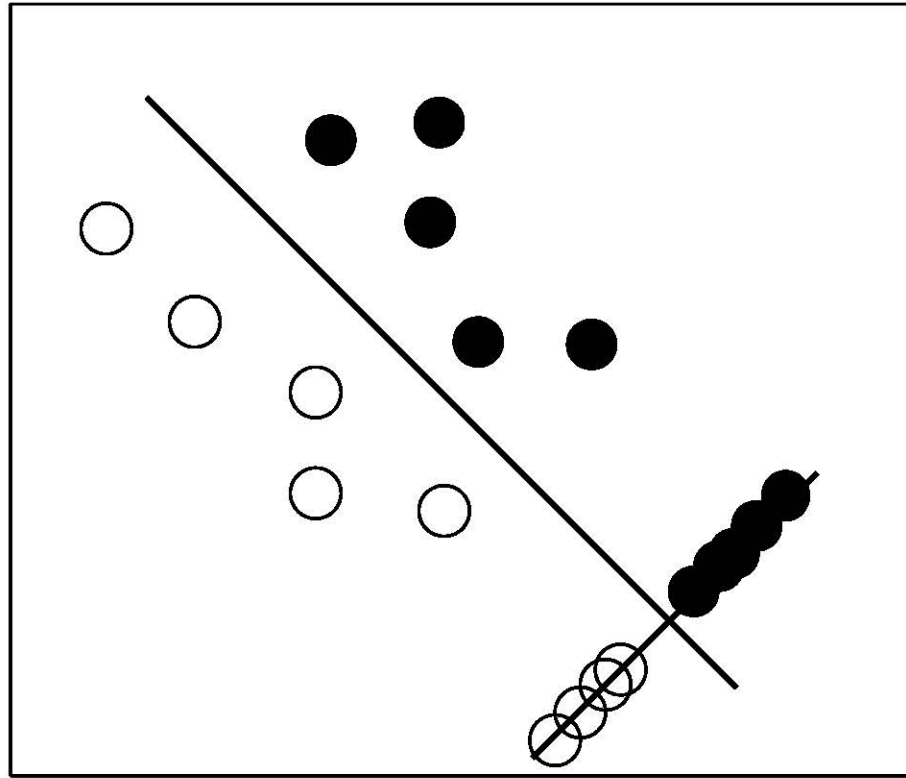
photographic (40,000)

# Photorealistic training set: downloaded from [www.raph.com](http://www.raph.com) and [www.irtc.org](http://www.irtc.org)



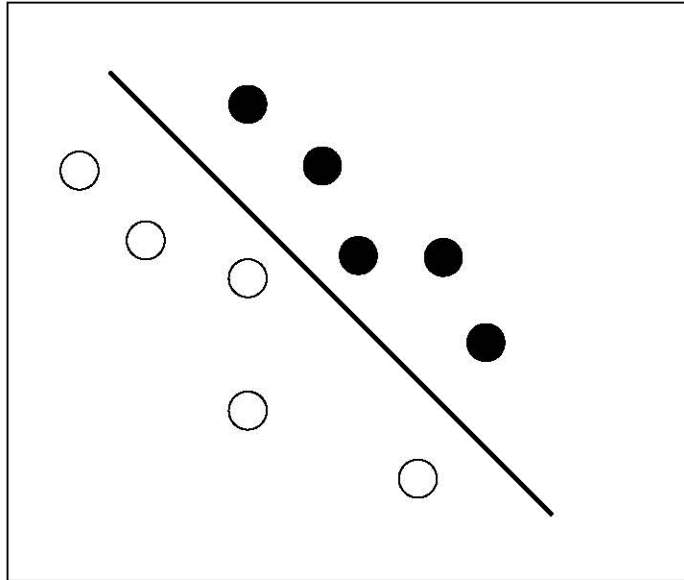
photorealistic (6, 000)

# Classifier 1: LDA. Simple, amenable to analysis

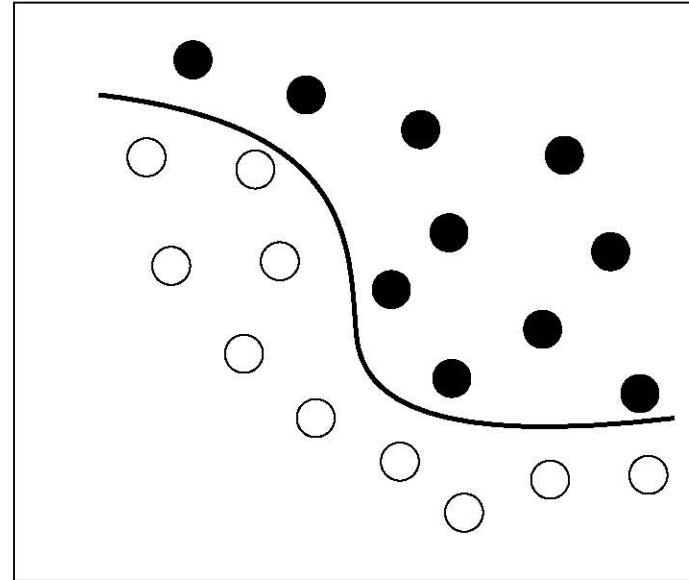


linear discriminant analysis (LDA)

## Classifier 2: SVM. State of the art.

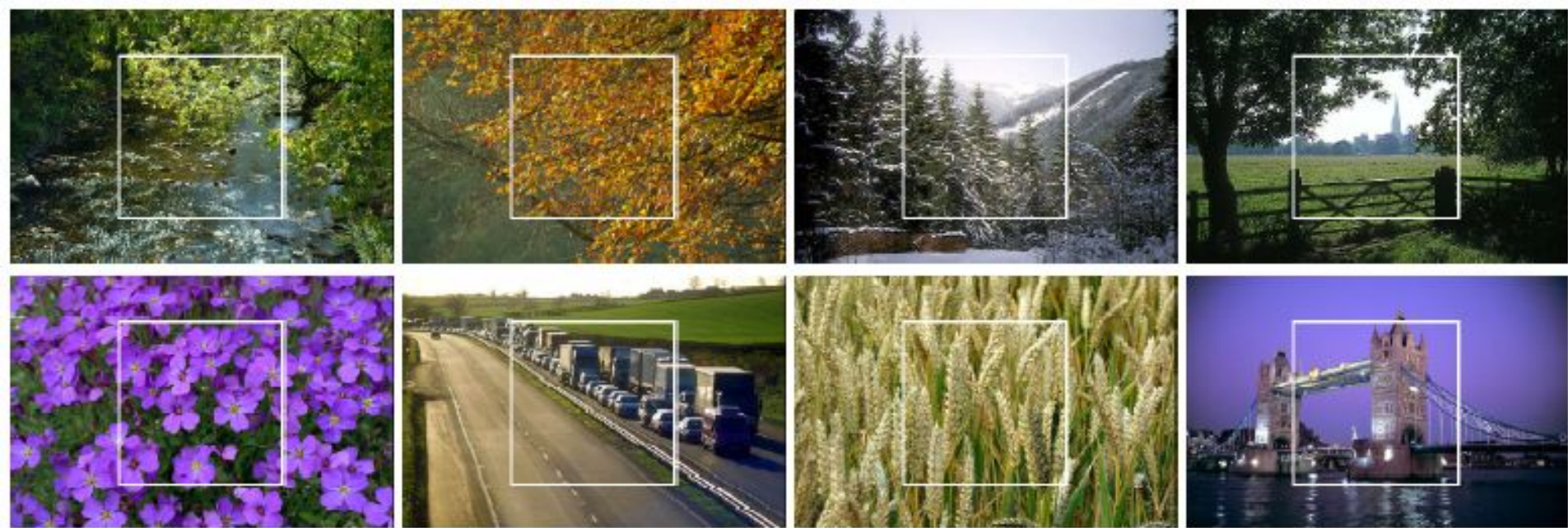


linear SVM



non-linear SVM

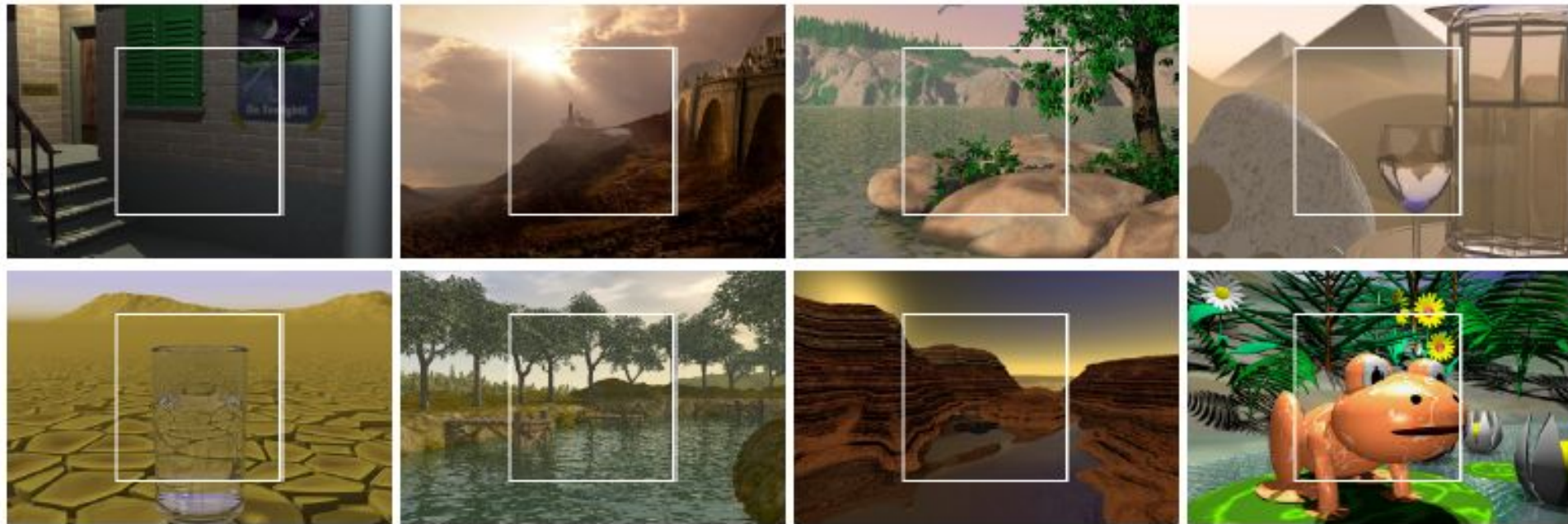
# Easily classified photographic images



**Fig. 4:** Easily classified photographic images.



# Easily classified photorealistic images



**Fig. 5:** Easily classified photorealistic images.

# Incorrectly classified photographic images



Fig. 6: Incorrectly classified photographic images.

# Incorrectly classified photorealistic images

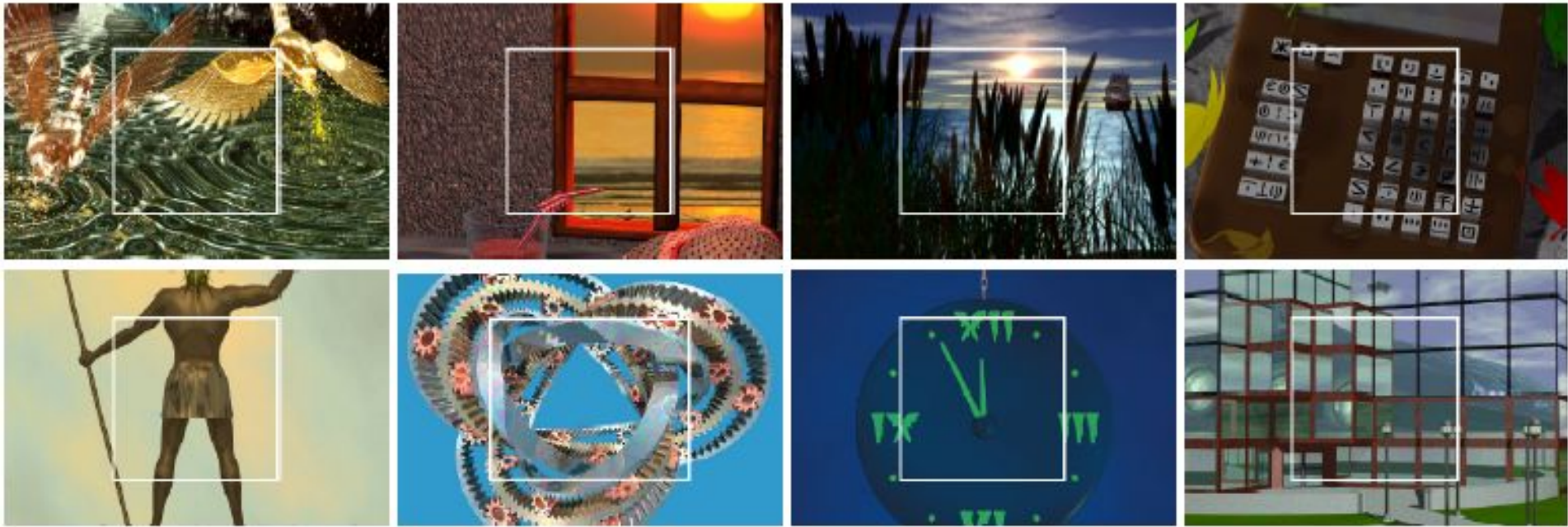
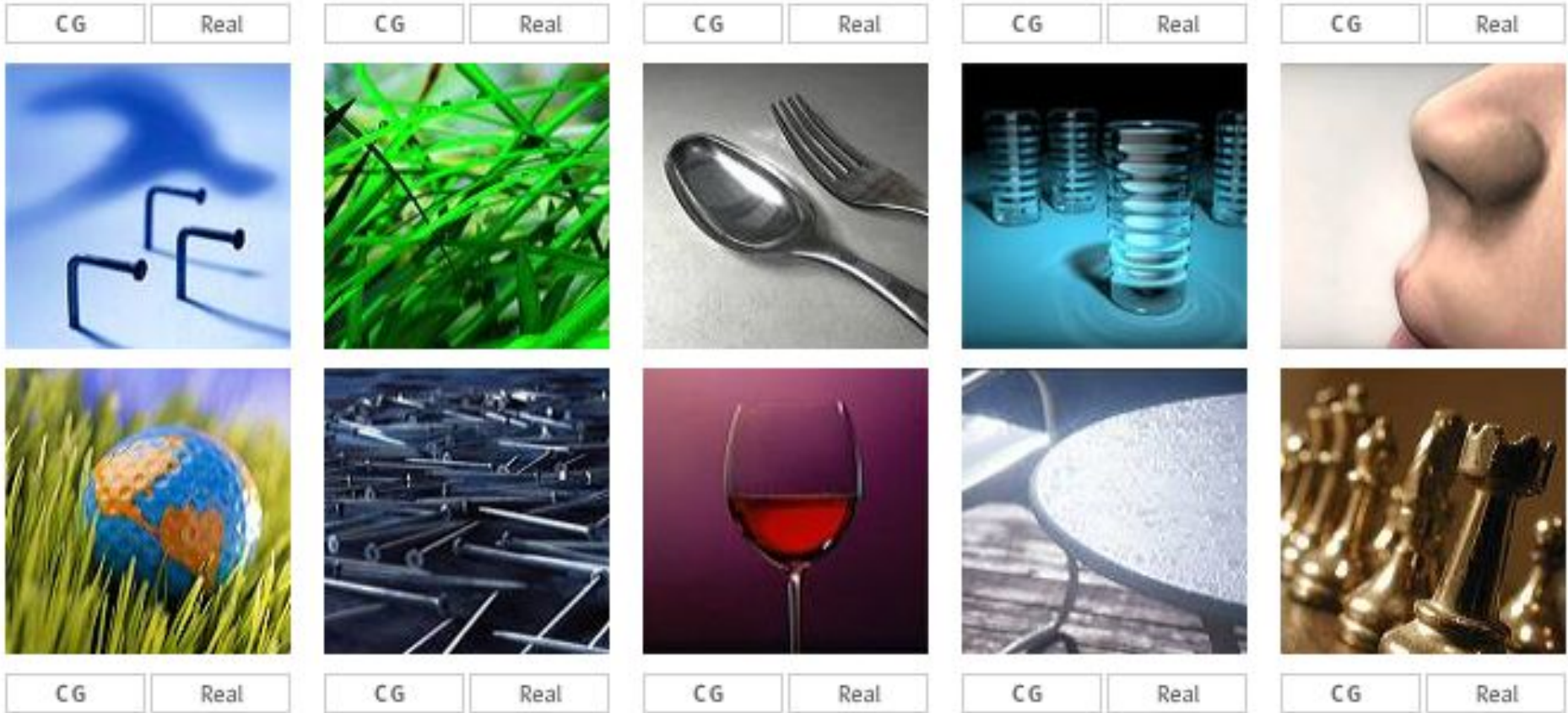


Fig. 7: Incorrectly classified photorealistic images.

# www.fakeorfoto.com

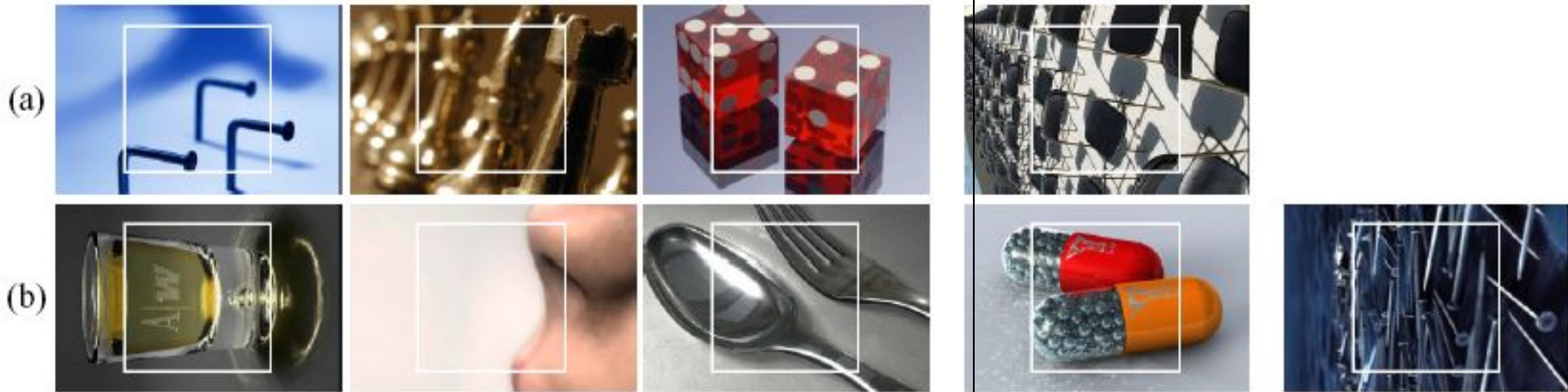


# Results of algorithm

Photographic images

Photorealistic images

Correct



Incorrect



**Fig. 9:** Images from [www.fakeorfoto.com](http://www.fakeorfoto.com). Shown in (a) and (c) are correctly and incorrectly classified photographic images, respectively. Shown in (b) and (d) are correctly and incorrectly classified photorealistic images, respectively.

# Taking a picture...

What the camera give us...

How do we correct this?



# Close-up

---

Original



Naïve Sharpening



Our algorithm



Why does picture appear blurry?

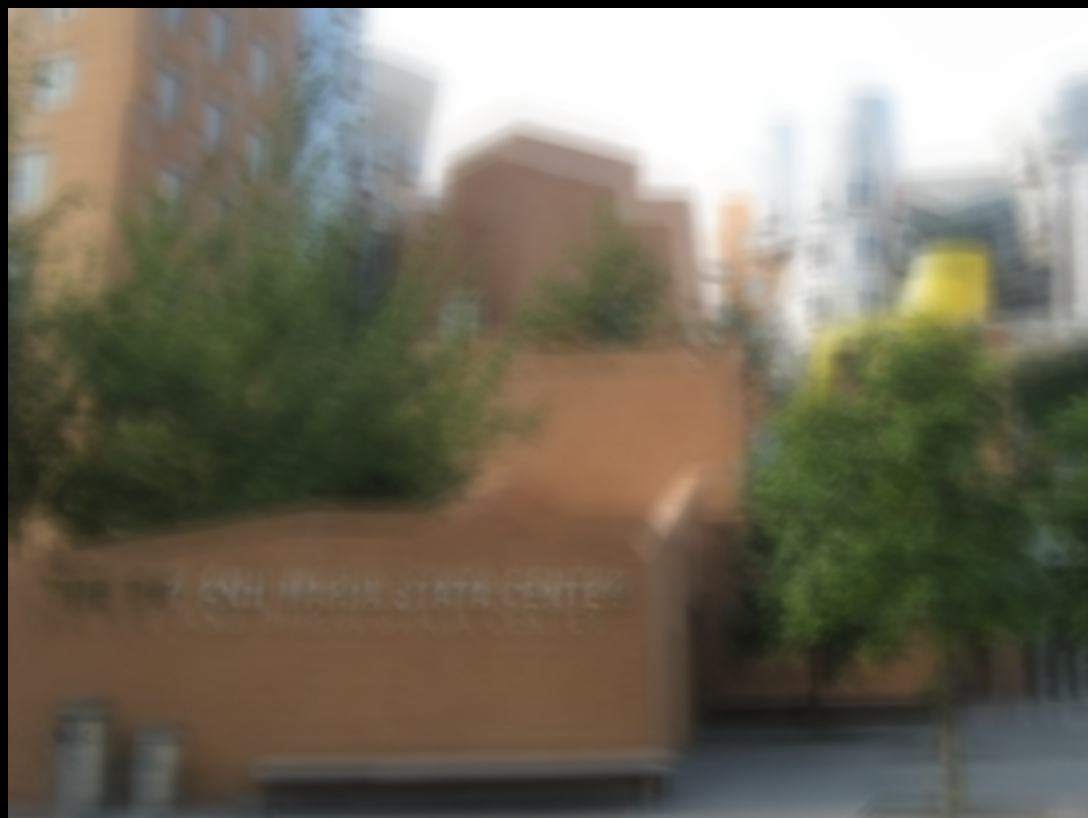


# Let's take a photo

---



Blurry result



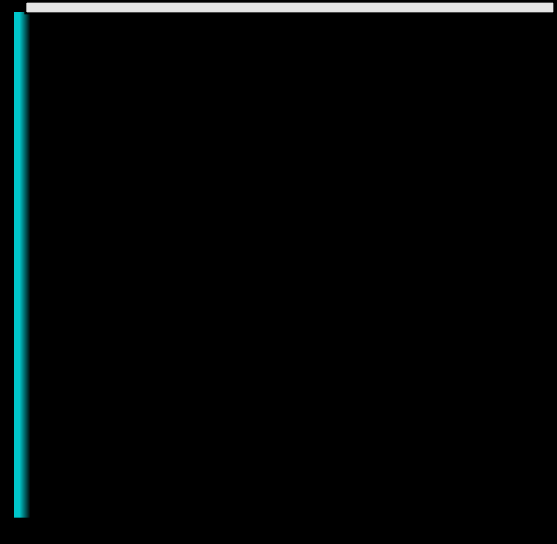
# Slow-motion replay

---



# Slow-motion replay

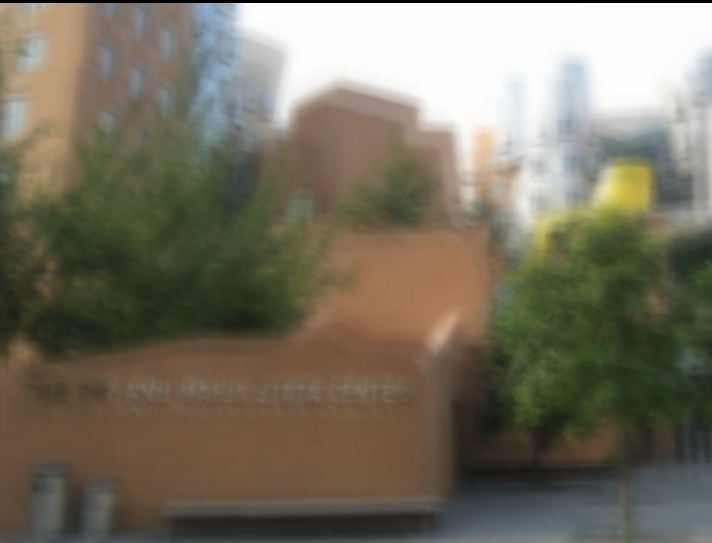
---



Motion of camera

# Image formation process

---



Blurry image

Input to algorithm

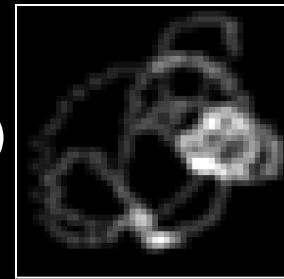
=



Sharp image

Desired output

⊗



Blur kernel

Convolution operator

**Model is approximation**

# Why is this hard?

---

Simple analogy:

11 is the product of two numbers.

What are they?

No unique solution:

$$11 = 1 \times 11$$

$$11 = 2 \times 5.5$$

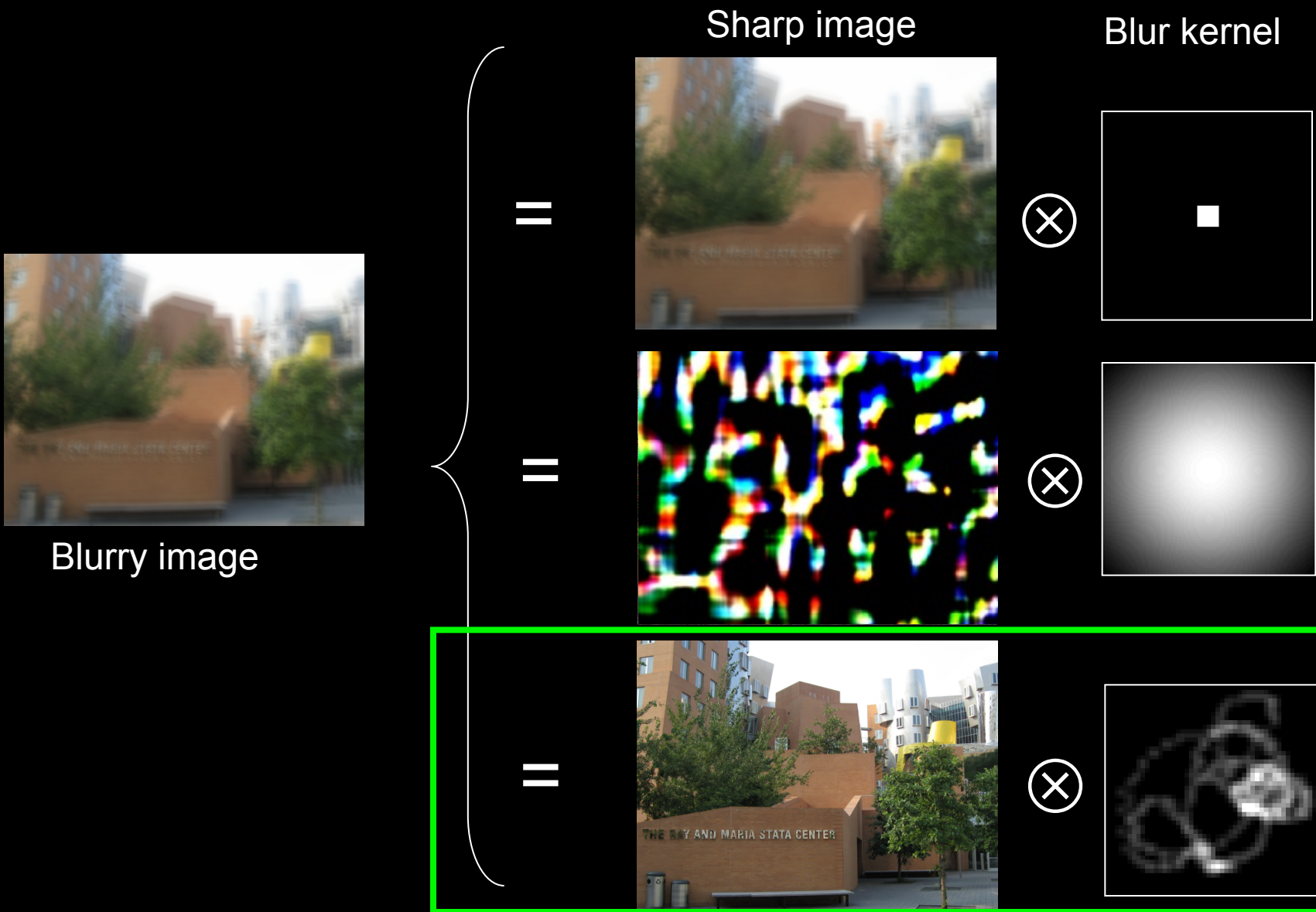
$$11 = 3 \times 3.667$$

etc.....

**Need more information !!!!**

# Multiple possible solutions

---

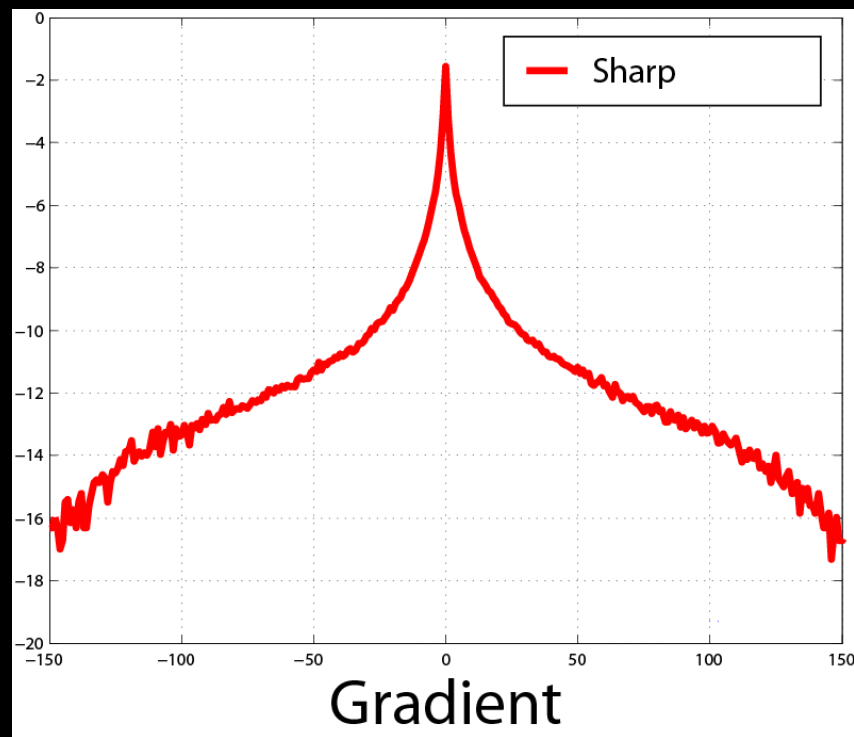


# Natural image statistics

---

Characteristic distribution with heavy tails

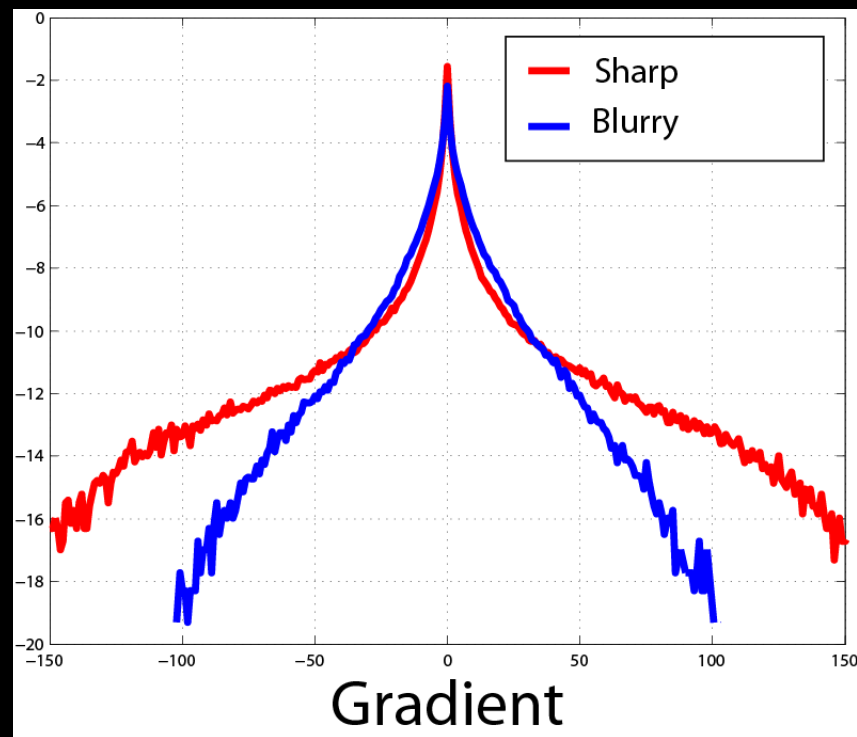
Histogram of image gradients



# Blurry images have different statistics

---

## Histogram of image gradients

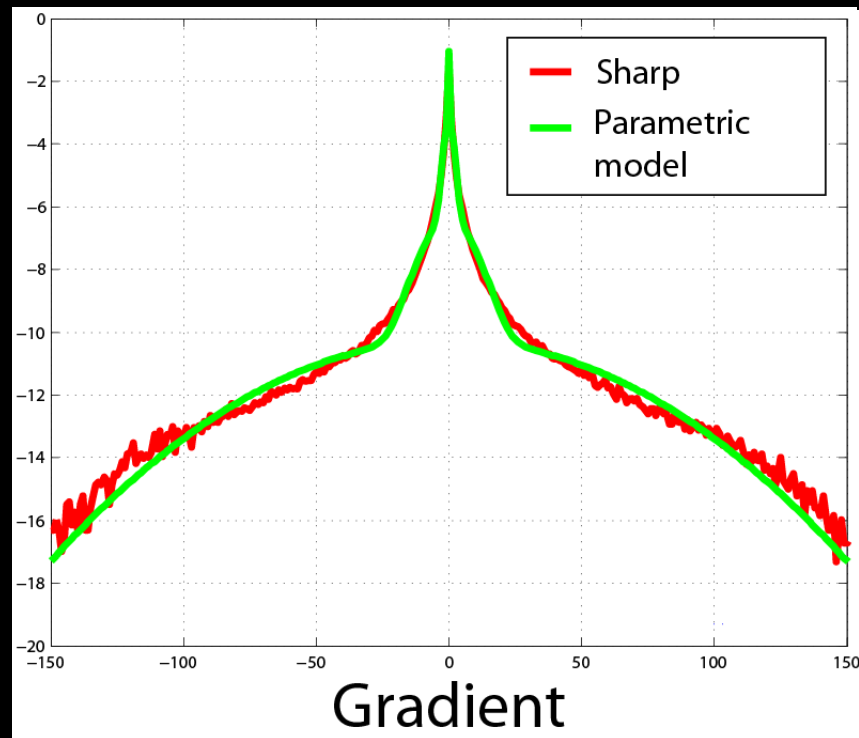




# Parametric distribution

---

## Histogram of image gradients



Use parametric model of sharp image statistics

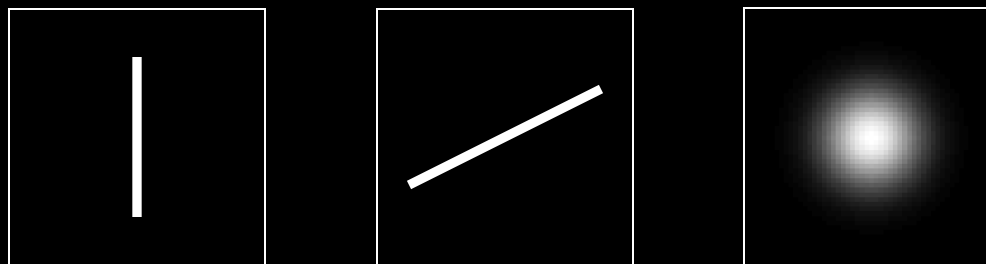
Slides R. Fergus

# 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



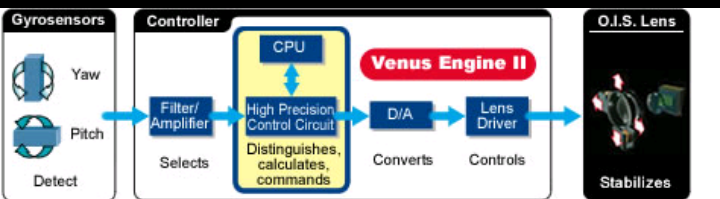
Assumed forms of blur kernels

- Image constraints are frequency-domain power-laws

# Existing work on image deblurring

## Hardware approaches

### Image stabilizers



### Dual cameras



Ben-Ezra and  
Nayar 2004

### Coded shutter



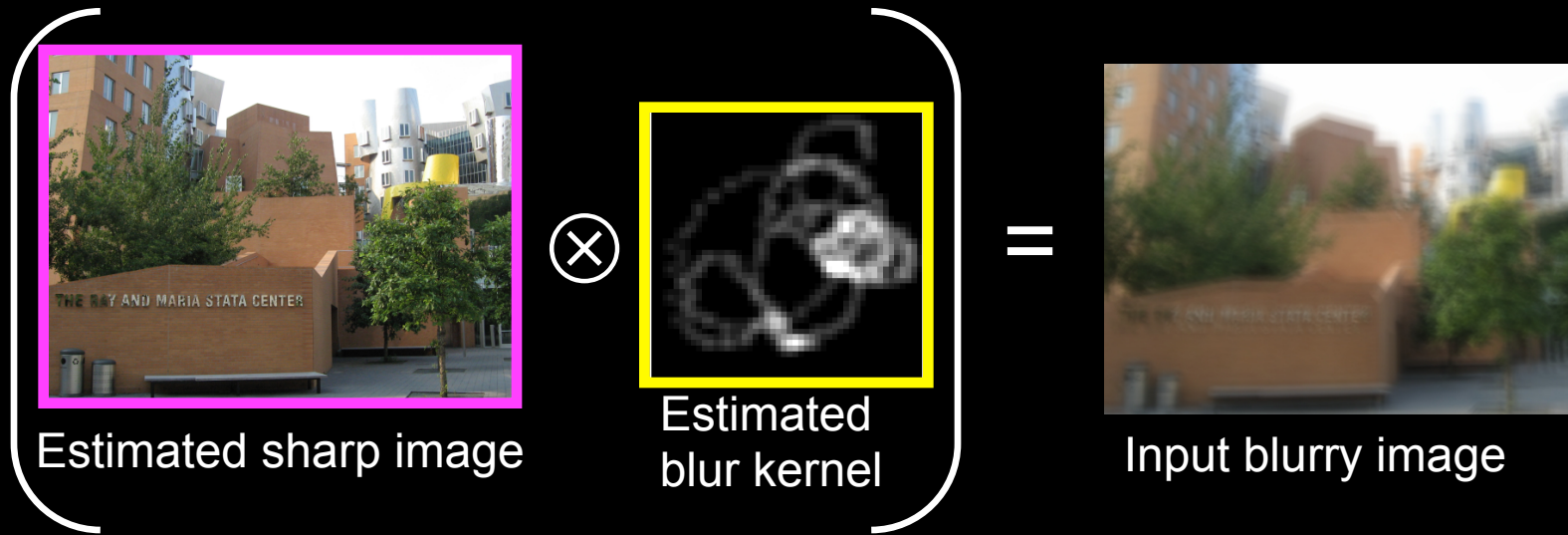
Raskar et al.  
SIGGRAPH 2006

Our approach can be combined with these hardware methods

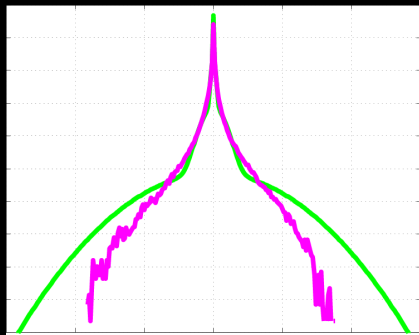
# Three sources of information

---

## 1. Reconstruction constraint:



## 2. Image prior:



Distribution of gradients

## 3. Blur prior:



Positive  
&  
Sparse

# Three sources of information

---

$y$  = observed image

$b$  = blur kernel

$x$  = sharp image

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

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

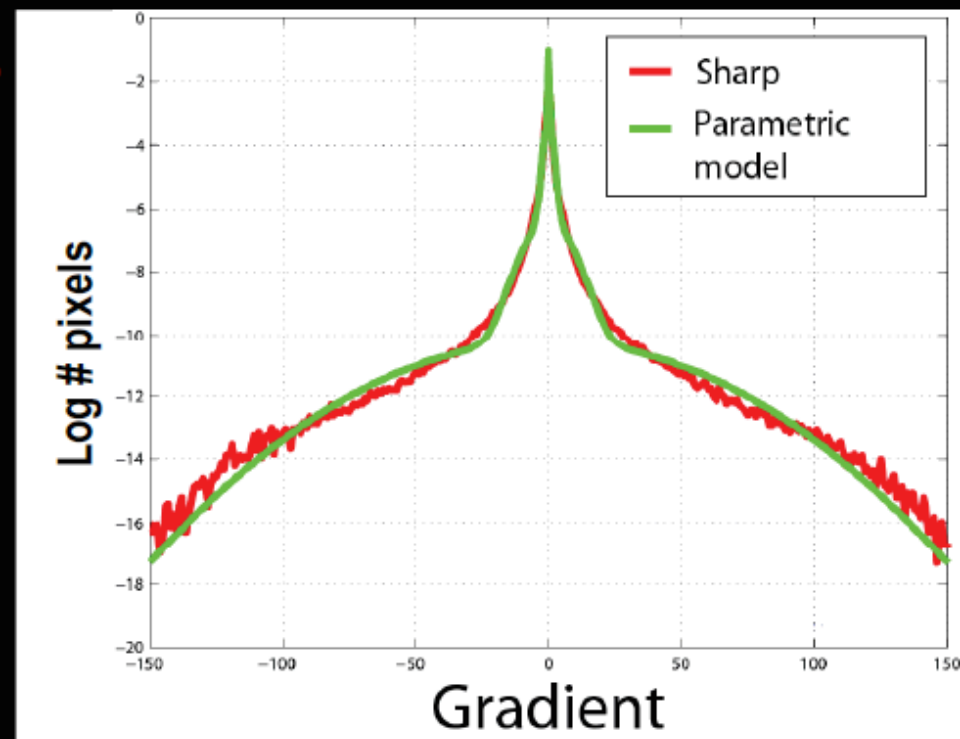
$$p(x) = \prod_i \sum_{c=1}^C \pi_c \mathcal{N}(f(x_i) | 0, s_c^2)$$

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

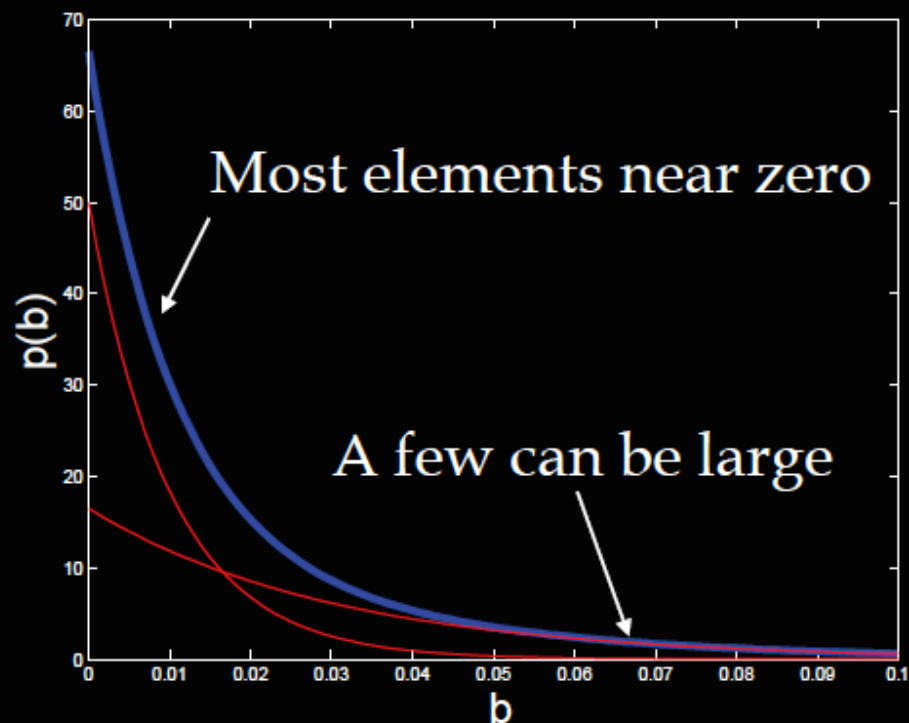
$$p(b) = \prod_j \sum_{d=1}^D \pi_d \mathcal{E}(b_j | \lambda_d)$$

#### 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:

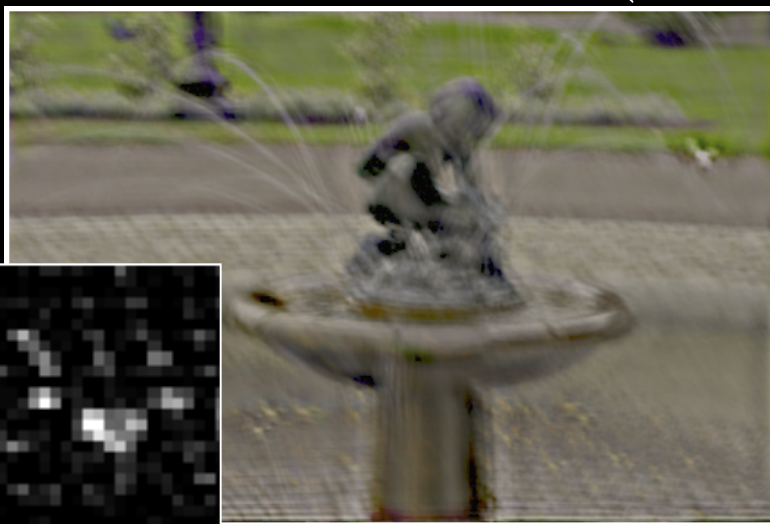
- Combine 3 terms into an objective function
- Run conjugate gradient descent
- This is Maximum a-Posteriori (MAP)

# Results from MAP estimation

Input blurry image



Maximum a-Posteriori (MAP)



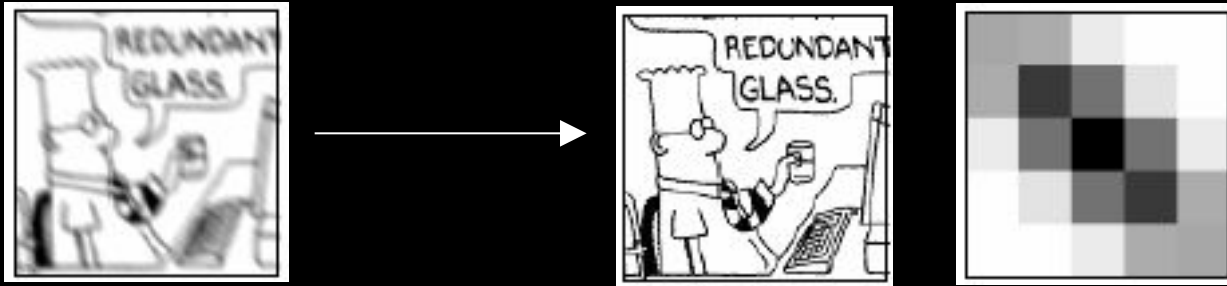
Our method: Variational Bayes



# 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

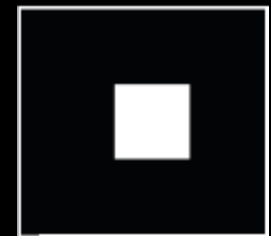
Initialize 3x3 blur kernel



Blurry patch

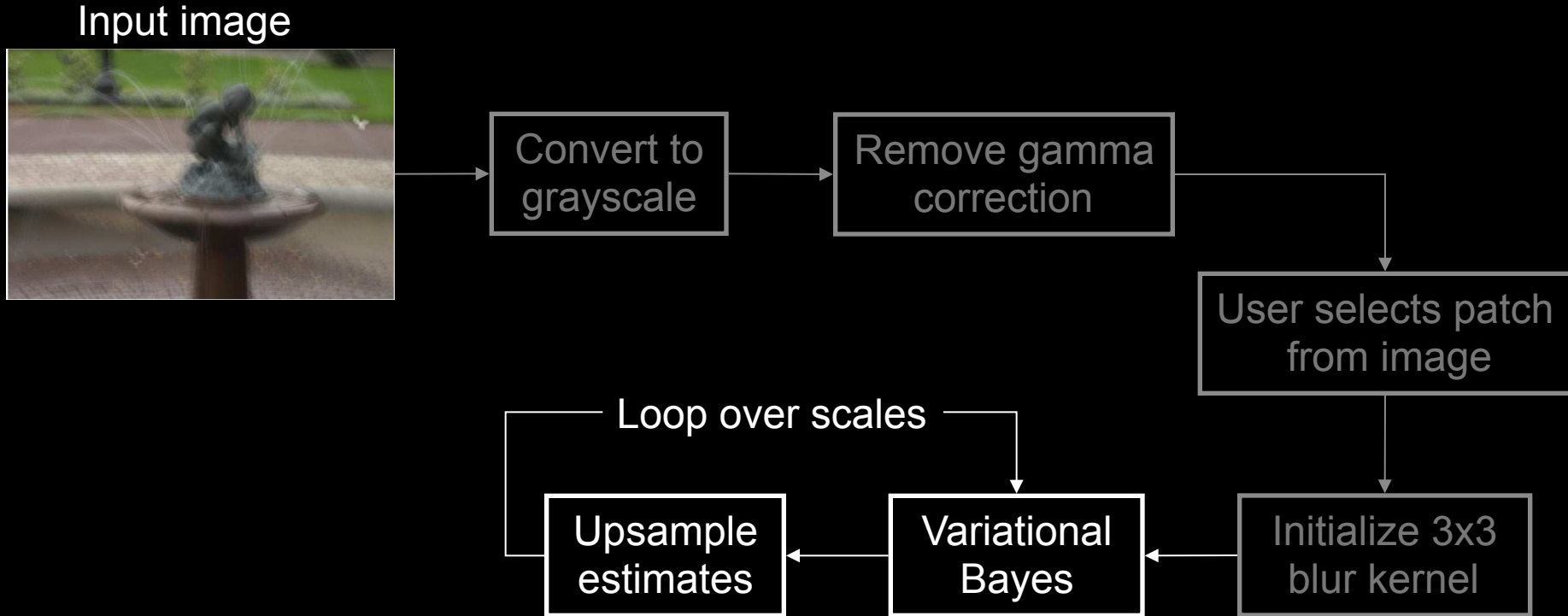


Initial image estimate

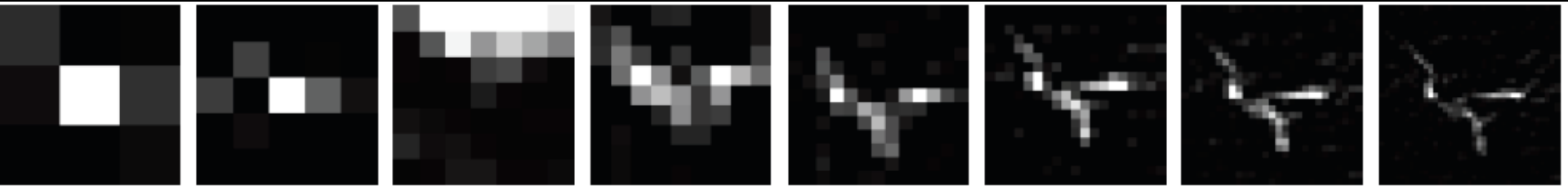


Initial blur kernel

# Inferring the kernel: multiscale method



Use multi-scale approach to avoid local minima:



# Image Reconstruction

Input image



Convert to grayscale

Remove gamma correction

User selects patch from image

Loop over scales

Upsample estimates

Variational Bayes

Initialize 3x3 blur kernel

Full resolution blur estimate

Non-blind deconvolution  
(Richardson-Lucy)



Deblurred image



# Results on real images

---

Submitted by people from their own photo collections

Type of camera unknown

Output does contain artifacts

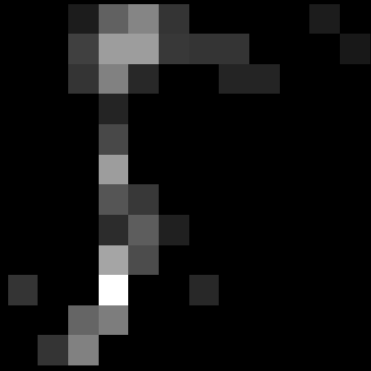
- Increased noise
- Ringing

Compares well to existing methods

Original photograph



Blur kernel



Our output



Matlab's deconvblind



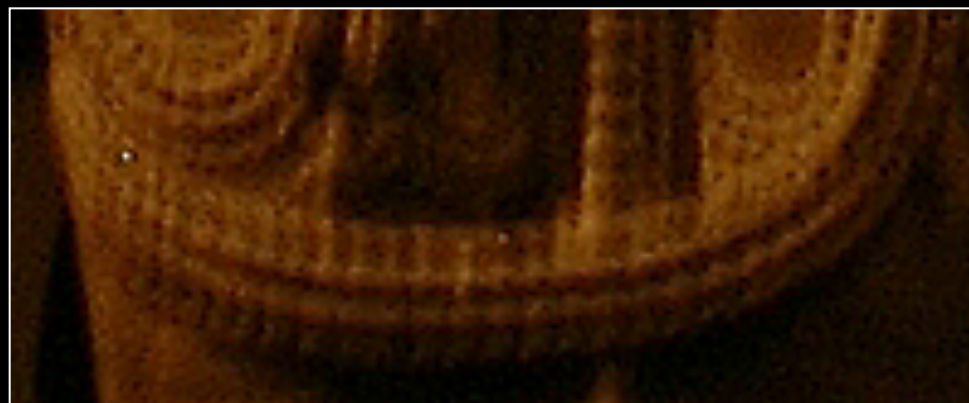
# Close-up of garland

---

Original



Matlab's  
`deconvblind`



Our output



Original photograph



# Matlab's deconvblind



Photoshop sharpen more

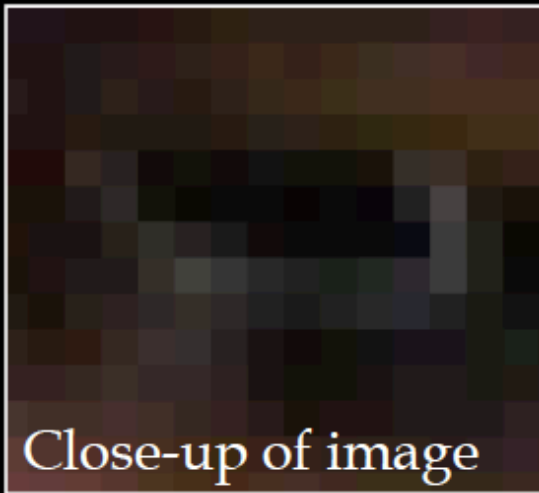
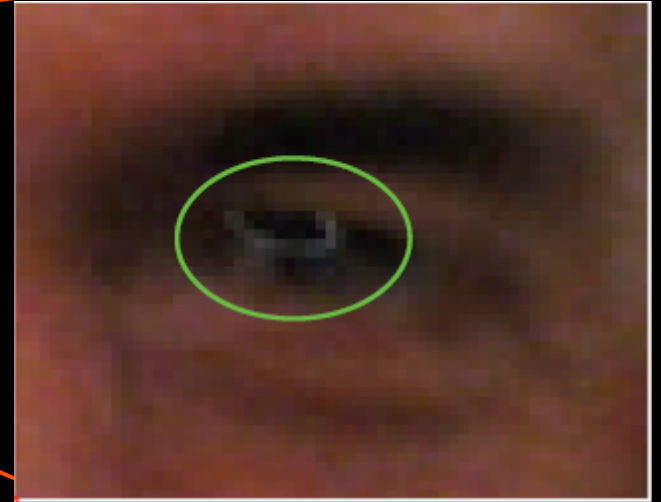
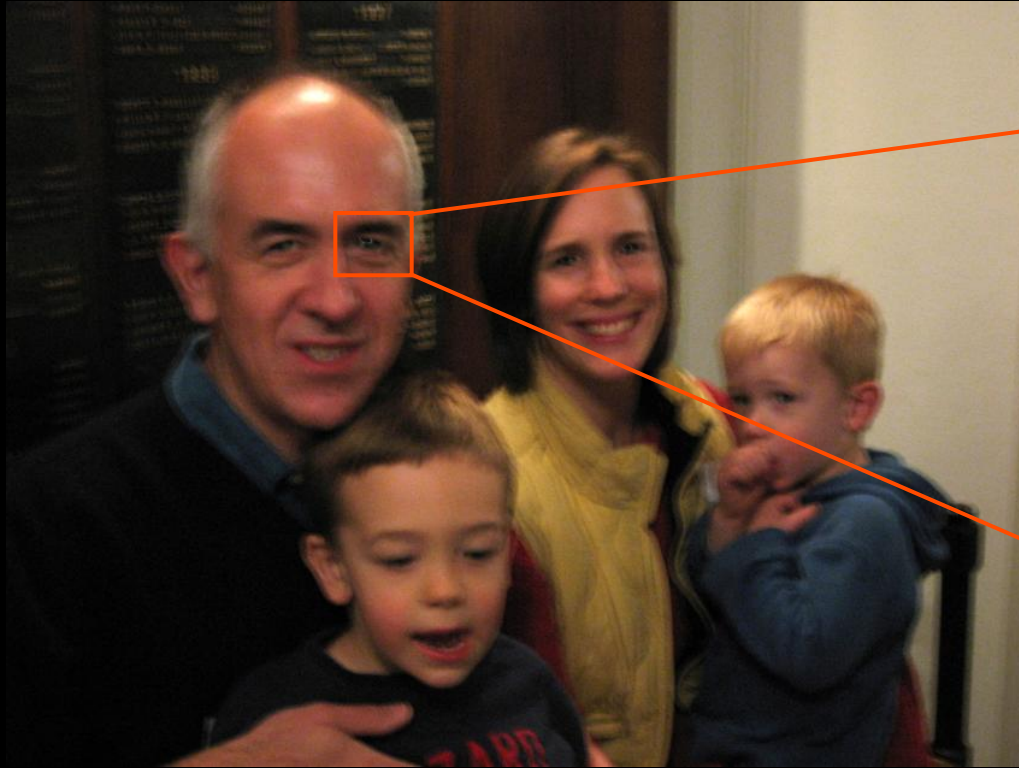




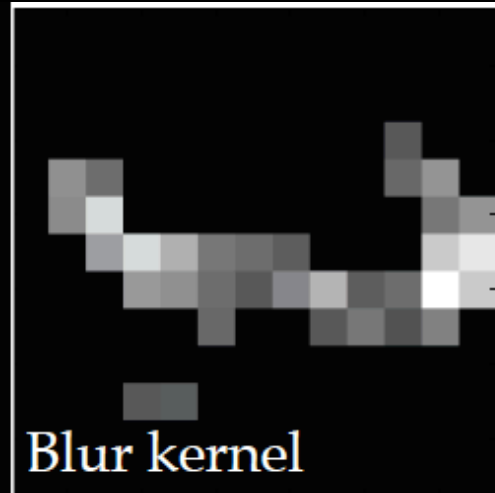
Our output

Blur kernel





Close-up of image



Blur kernel



Close-up of our output

# Original photograph



# Our output

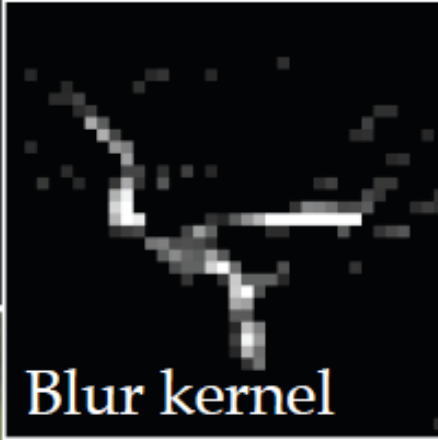


Blur kernel

# Original photograph



Our output



# Matlab's deconvblind

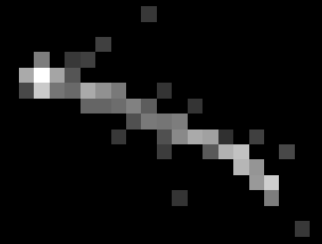


Original photograph





Our output



Blur kernel

# Close-up of bird

---

Original



Unsharp mask



Our output



Original photograph



Blur kernel

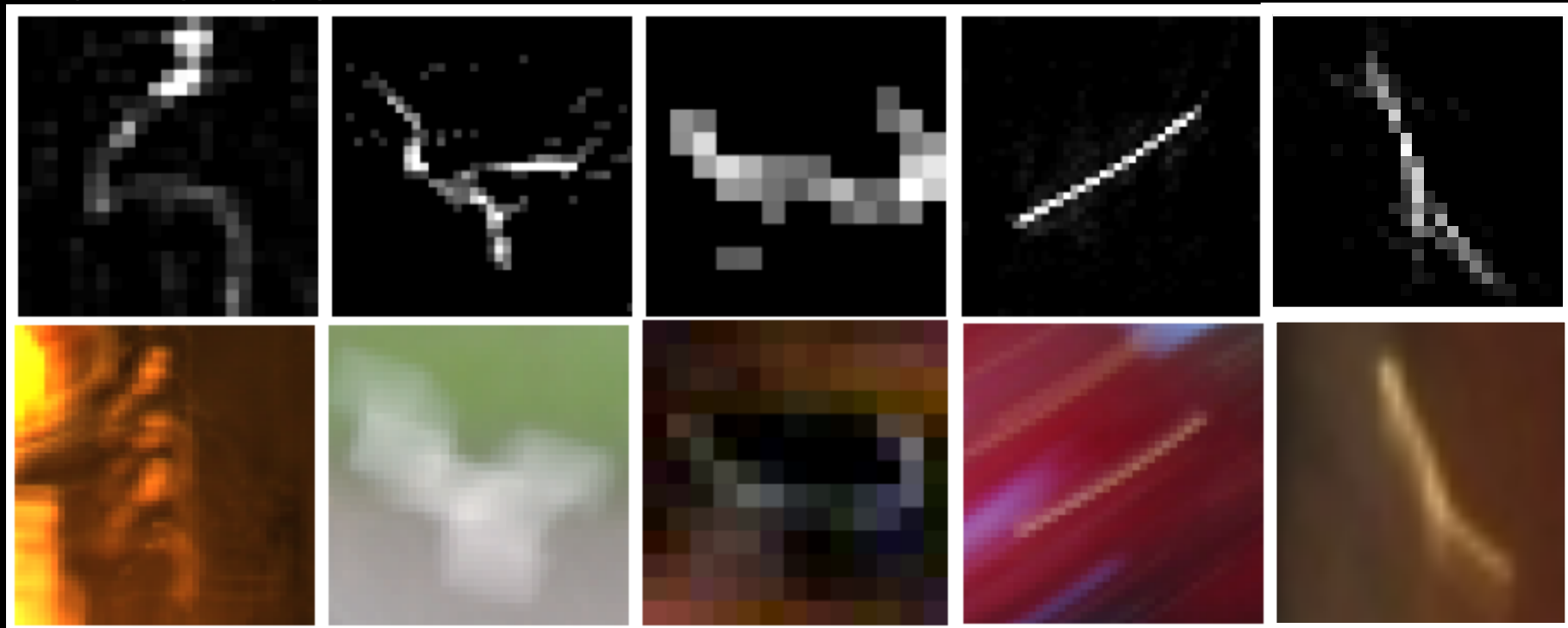
output



# Image artifacts & estimated kernels

---

## Blur kernels



## Image patterns

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

# Bayesian methods

