Course Material for CS 754 (Advanced Image Processing, IITB)
Course Instructor: Ajit Rajwade

# Statistics of Natural Images

# Motivation

- Number of possible 200 x 200 images (of 256, i.e. 8 bit intensity levels) = 256^40000 = 2^320000 = 10^110000.
- This is several trillion times the number of atoms in the universe (10^90).
- Only a tiny subset of these are plausible as natural images.

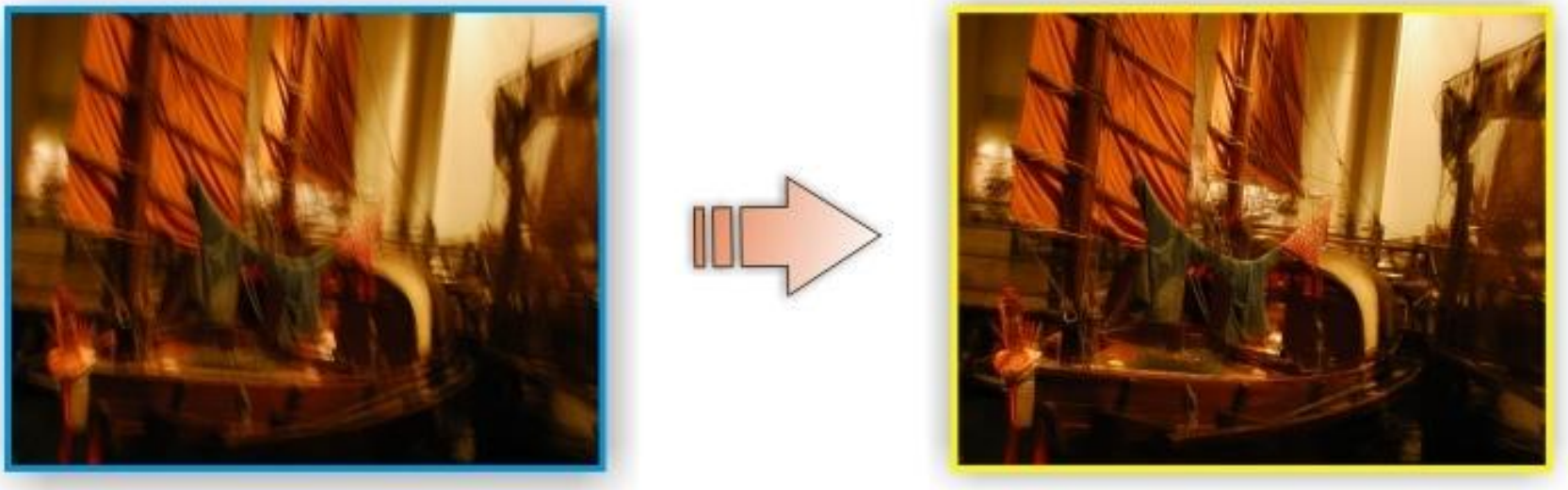
# Why study statistics of natural images?

- Useful for many computer vision or image processing applications

1. Image denoising, deblurring, filling of missing pixels in images (inpainting)
2. Image compression
3. Classification into image categories

Sample State of the art result: Gaussian Noise sigma = 15

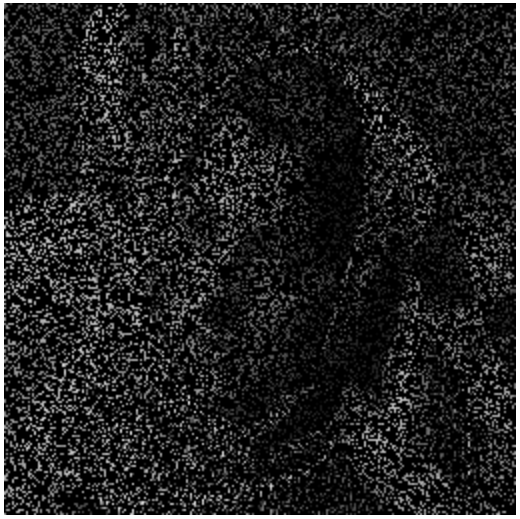
Motion deblurring



http://www.cse.cuhk.edu.hk/leojia/projects/motion_deblurring/

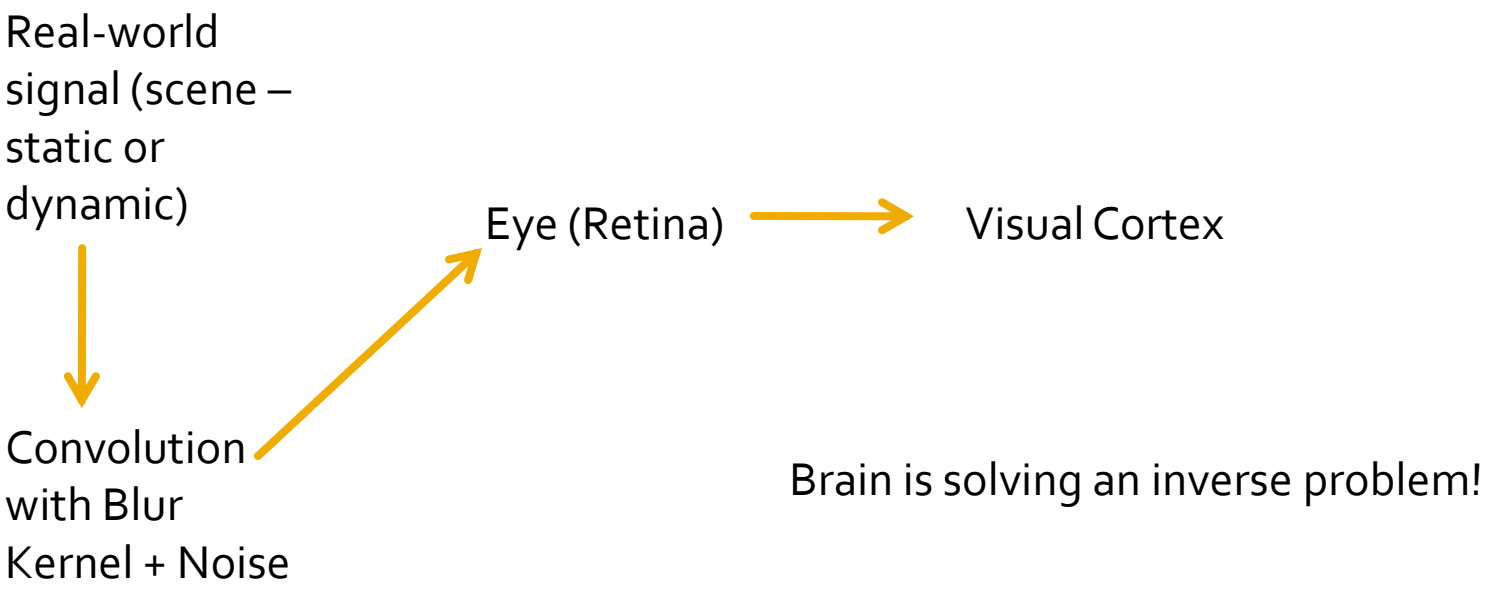
# Inpainting


Since 1699, when French explorers
landed at the great bend of the
Mississippi River and celebrated
the first Mardi Gras in North America,
New Orleans has brewed a fascinating
melange of cultures. It was French,
then Spanish, then French again, then
sold to the United States. Through all
these years, and even into the 1900s,
others arrived from everywhere:
Acadians (Cajuns), Africans, indige-



# Why study statistics of natural images?

- Natural image statistics also help us understand the human visual system better.



# What are these amazing statistical properties?

- Power law
- Distribution of the DCT coefficients or wavelet coefficients of images or image patches
- Relationships between these coefficients
- Many more!

# Some background

- We know that an image is a 2D array of intensity values stored at pixels.
- But images can be conveniently represented in the frequency domain as well.

# Discrete Fourier transform

- Given a 2D discrete signal (image) $f(x,y)$ of size $W_1$ by $W_2$, its DFT is given as:

$$F_d(u,v) = \frac{1}{\sqrt{W_1 W_2}} \sum_{x=0}^{W_1-1} \sum_{y=0}^{W_2-1} f(x,y) \exp(-j2\pi(ux/W_1 + vy/W_2))$$

$$f(x,y) = \frac{1}{\sqrt{W_1 W_2}} \sum_{u=0}^{W_1-1} \sum_{v=0}^{W_2-1} F(u,v) \exp(j2\pi(ux/W_1 + vy/W_2))$$

- Here the image is being represented as a linear combination of complex exponentials of different frequencies.

# 2D Discrete Cosine Transform

$$F(u,v) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(n,m)a_{NM}^{unvm}$$

$$f(n,m) = \sum_{u=0}^{N-1} F(u,v)\tilde{a}_{NM}^{unvm}$$

Here the image is being represented as a linear combination of the **cosine** functions of different frequencies.

$$DCT:$$

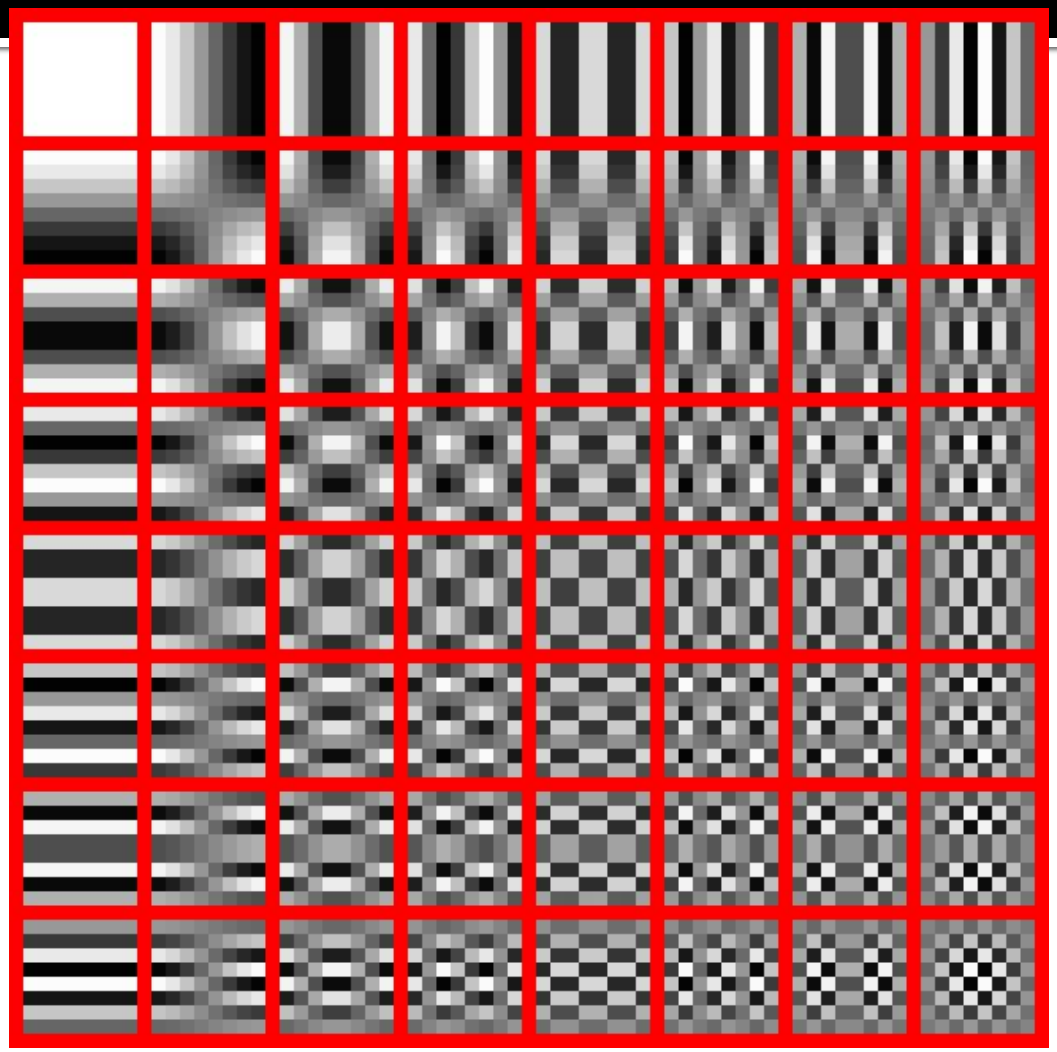
$$a_{NM}^{unvm} = \alpha(u)\alpha(v)\cos\left(\frac{\pi(2n+1)u}{2N}\right)\cos\left(\frac{\pi(2m+1)v}{2M}\right), u = 0\ldots N-1, v = 0\ldots M-1$$

$$\alpha(u) = \sqrt{1/N}\ (u=0), \text{else } \alpha(u) = \sqrt{2/N}$$

$$\alpha(v) = \sqrt{1/M}\ (v=0), \text{else } \alpha(v) = \sqrt{2/M}$$

$$\tilde{a}_{NM}^{unvm} = a_{NM}^{unvm}$$

# How do the DCT bases look like? (2D-case)



The DCT transforms an 8×8 block of input values to a linear combination of these 64 patterns. The patterns are referred to as the two-dimensional DCT *basis vectors*, and the output values are referred to as *transform coefficients*. Here each basis vector is reshaped to form an image.

Note: An image patch (size 8 x 8) can be represented as the linear combination of these 64 patterns.

http://en.wikipedia.org/wiki/JPEG

# (1) Power law for natural images

- The squared-amplitudes of the frequency components of an average natural image undergo rapid decay w.r.t. frequency pair magnitude $f$

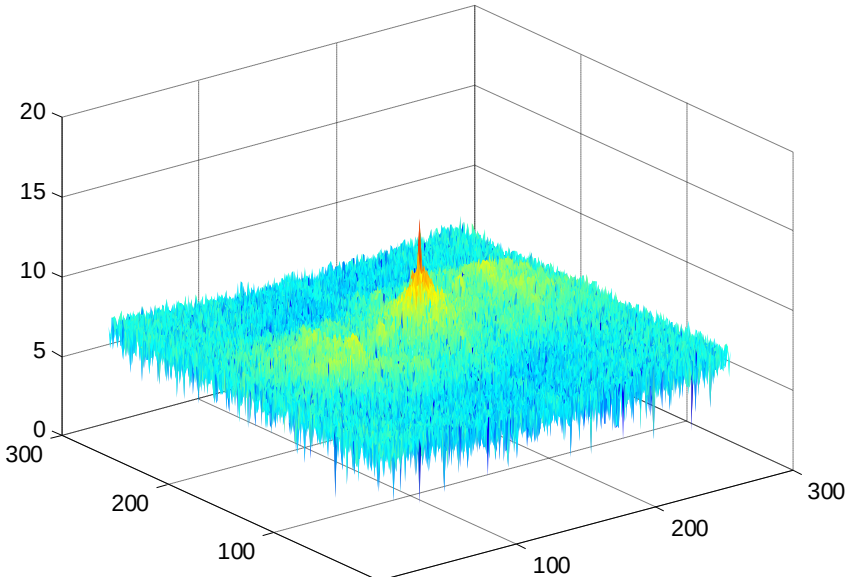
Expected value of squared magnitude of Fourier transform coefficient at frequency (u,v). Expectation is over all natural images.

$$E(|S(u,v)|^2) = A|f|^{\alpha-2} \quad \text{where} \quad |f| = \sqrt{u^2+v^2};\ f = (u,v)$$

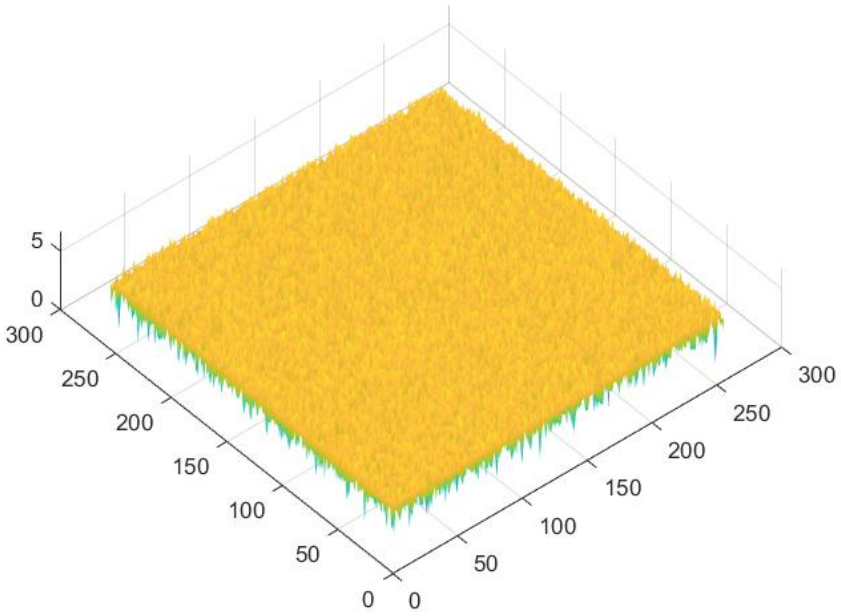
where $\alpha$ is a small number between 0 and 1 (usually around 0.19 for natural images), $A$ is a constant.

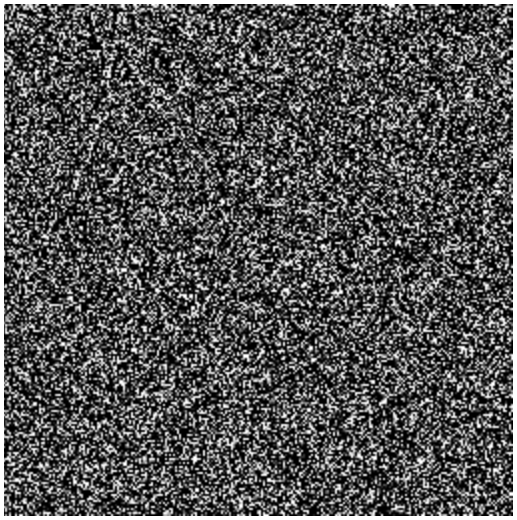
- This power law holds true across a range of scales (resolution) of the image.

Spectrum of Barbara
*log(abs(fftshift(fft2(im)+1)*

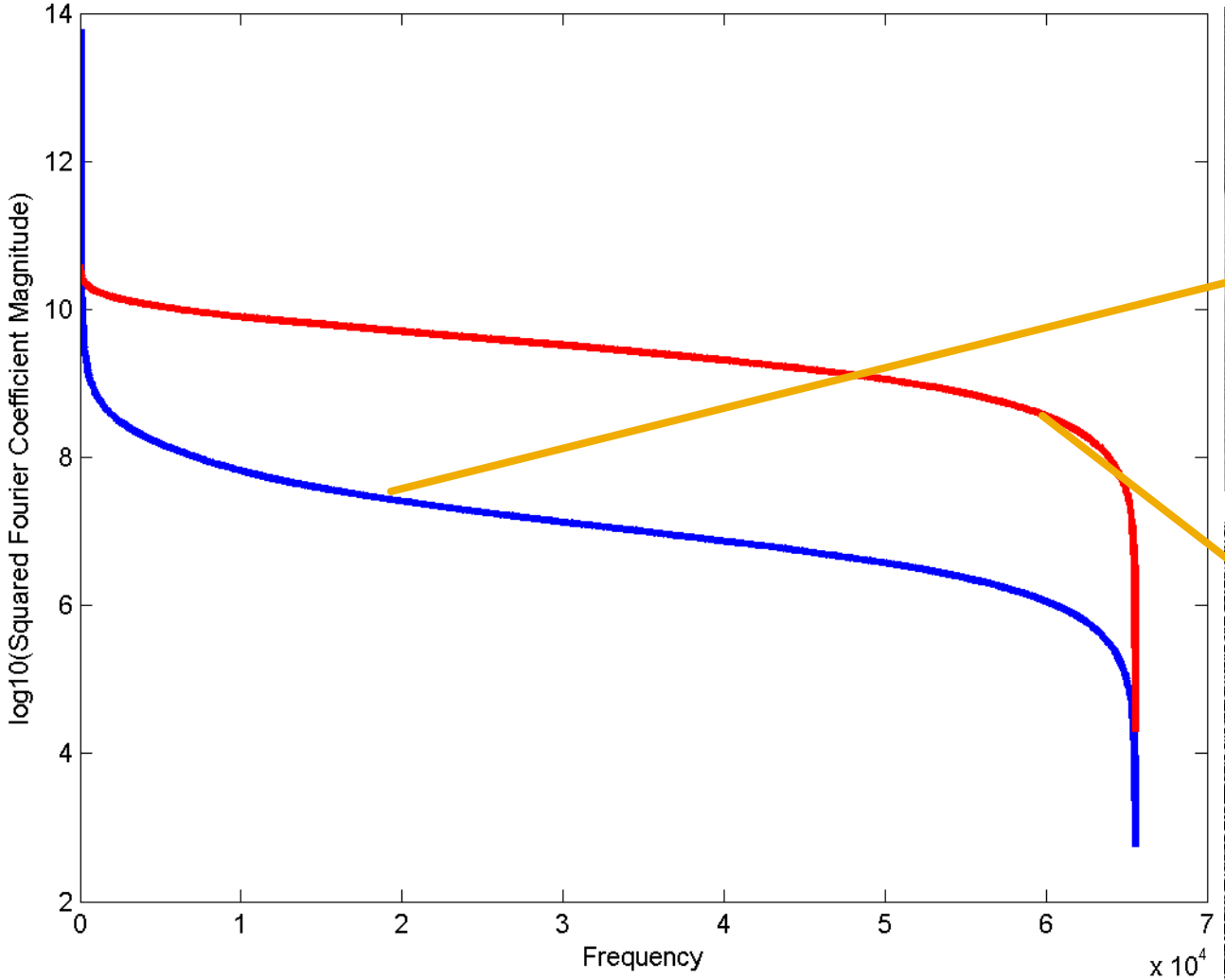


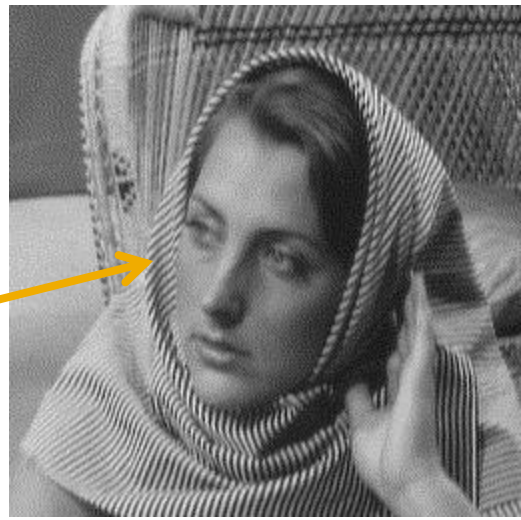

Spectrum of Noise
*log(abs(fftshift(fft2(im)+1)*

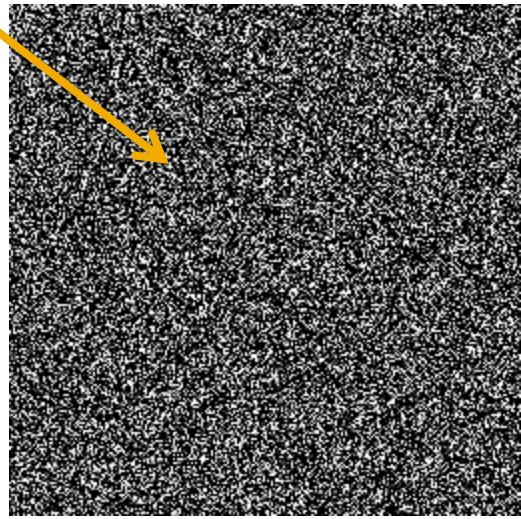


# (1) Power Law


log10(Squared Fourier Coefficient Magnitude)
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
Frequency
x 10^4

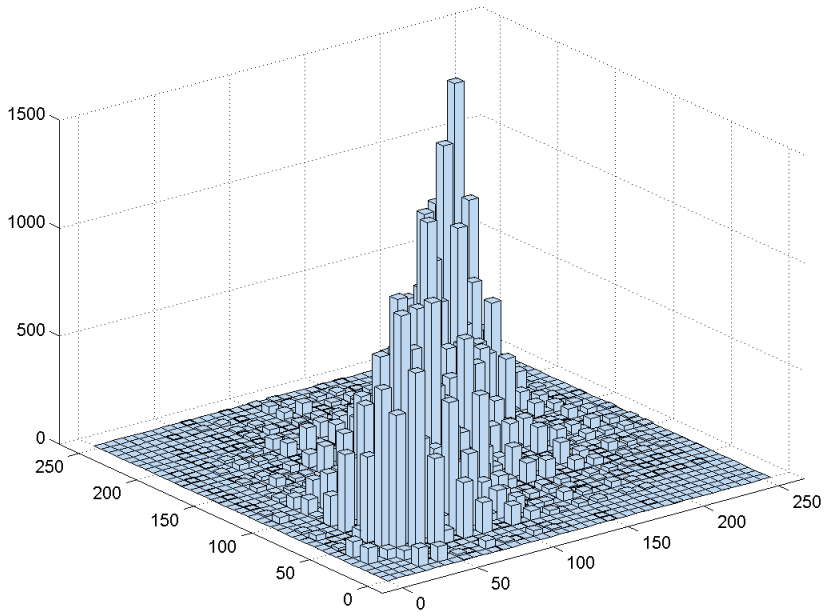


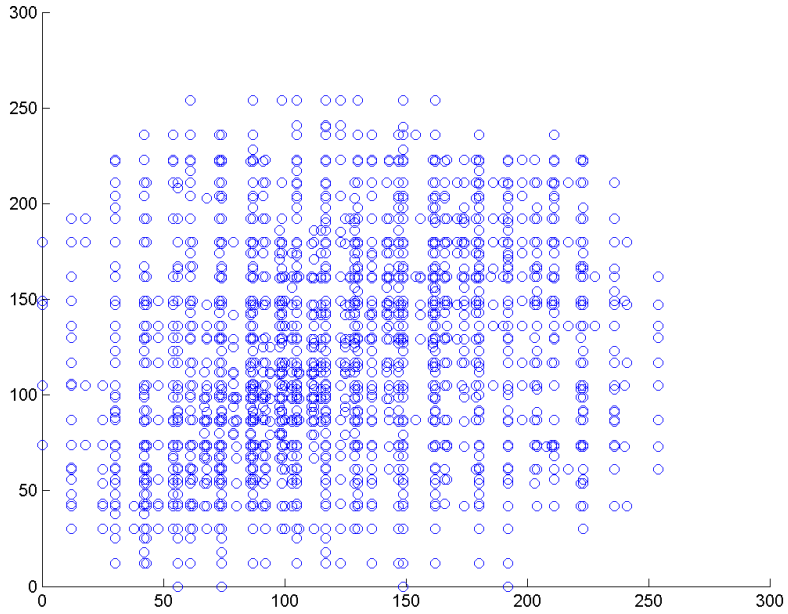


# (1) Power Law

- Power law is tremendously useful for (lossy) image compression.
- Image energy concentrated in just first few Fourier coefficients.
- Remaining coefficients are small and can be ignored (i.e. considered to be 0) without much loss of information.
- This principle is used by the JPEG algorithm (DCT coefficients), usually at a patch (block) level (patch-size of 8 x 8, usually).

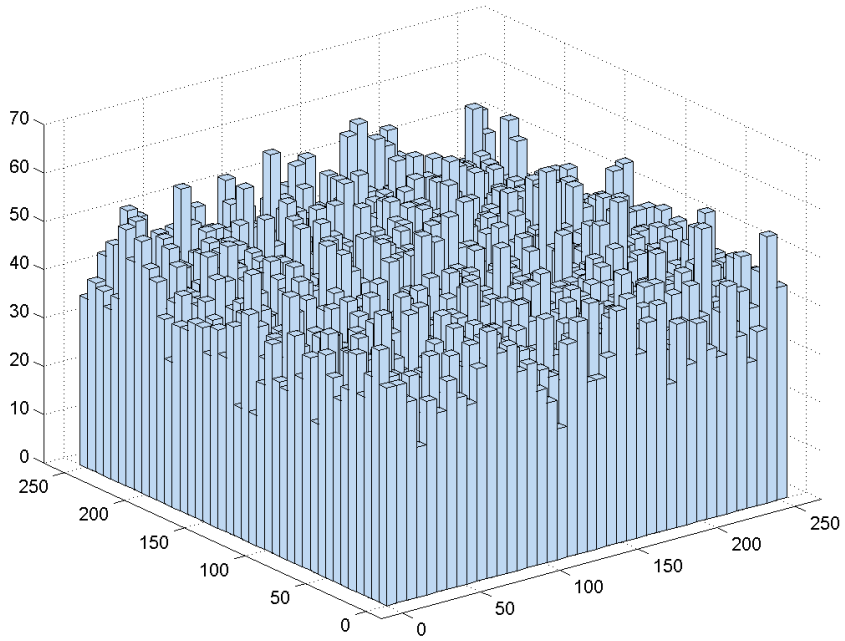
# (2) Joint statistics: pixel and an immediate neighbor

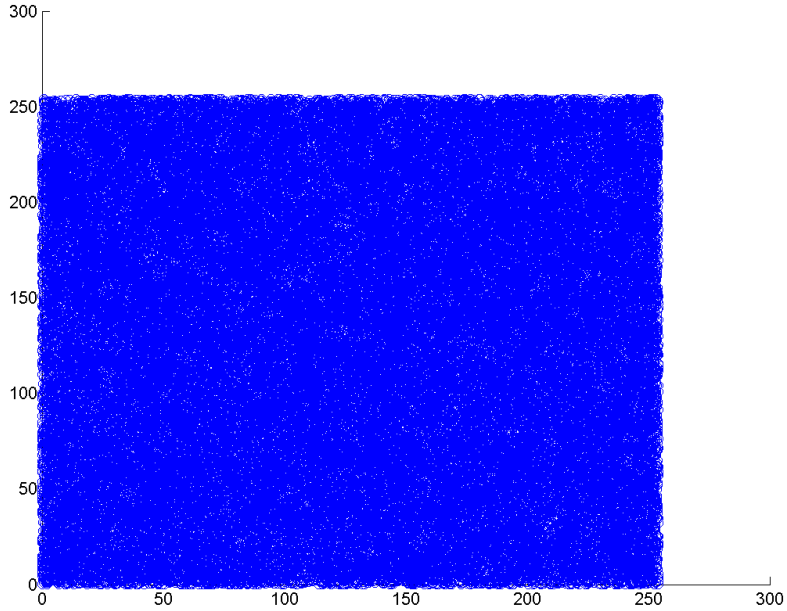



What do you think the corresponding plot for a pure noise image would look like?

# (2) Joint statistics: pixel and an immediate neighbor




What do you think the corresponding plot for a pure noise image would look like?

# (2) Joint statistics of pixel and an immediate neighbor

- The histogram shows a high degree of correlation between the values of a pixel and one of its immediate neighbors.

http://www.cse.iitb.ac.in/~ajitvr/CS663/DCT_PCA/

# (2a) Joint statistics of a pixel and nearby pixels: Experiment

- Suppose you extract $M \sim 100{,}000$ small-sized (8 x 8) patches from a set of images.
- Compute the column-column and row-row correlation matrices.

$$\mathbf{C_C} = \frac{1}{M-1}\sum_{i=1}^{M}\mathbf{P}_i\mathbf{P}_i^T = \frac{1}{M-1}\sum_{i=1}^{M}\sum_{j=1}^{8}P_i(:,j)P_i(:,j)';$$

$$\mathbf{C_R} = \frac{1}{M-1}\sum_{i=1}^{M}\mathbf{P}_i^T\mathbf{P}_i = \frac{1}{M-1}\sum_{i=1}^{M}\sum_{j=1}^{8}P_i(j,:)'P_i(j,:);$$

- The correlation values in the matrix decrease almost in geometric progression with respect to distance – for short distances.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.0000 | 0.9902 | 0.9795 | 0.9733 | 0.9682 | 0.9639 | 0.9604 | 0.9570 |
| 0.9902 | 1.0005 | 0.9908 | 0.9795 | 0.9734 | 0.9684 | 0.9643 | 0.9605 |
| 0.9795 | 0.9908 | 1.0010 | 0.9908 | 0.9796 | 0.9735 | 0.9689 | 0.9646 |
| 0.9733 | 0.9795 | 0.9908 | 1.0005 | 0.9904 | 0.9793 | 0.9735 | 0.9686 |
| 0.9682 | 0.9734 | 0.9796 | 0.9904 | 1.0004 | 0.9903 | 0.9794 | 0.9734 |
| 0.9639 | 0.9684 | 0.9735 | 0.9793 | 0.9903 | 1.0001 | 0.9903 | 0.9793 |
| 0.9604 | 0.9643 | 0.9689 | 0.9735 | 0.9794 | 0.9903 | 1.0004 | 0.9904 |
| 0.9570 | 0.9605 | 0.9646 | 0.9686 | 0.9734 | 0.9793 | 0.9904 | 1.0002 |

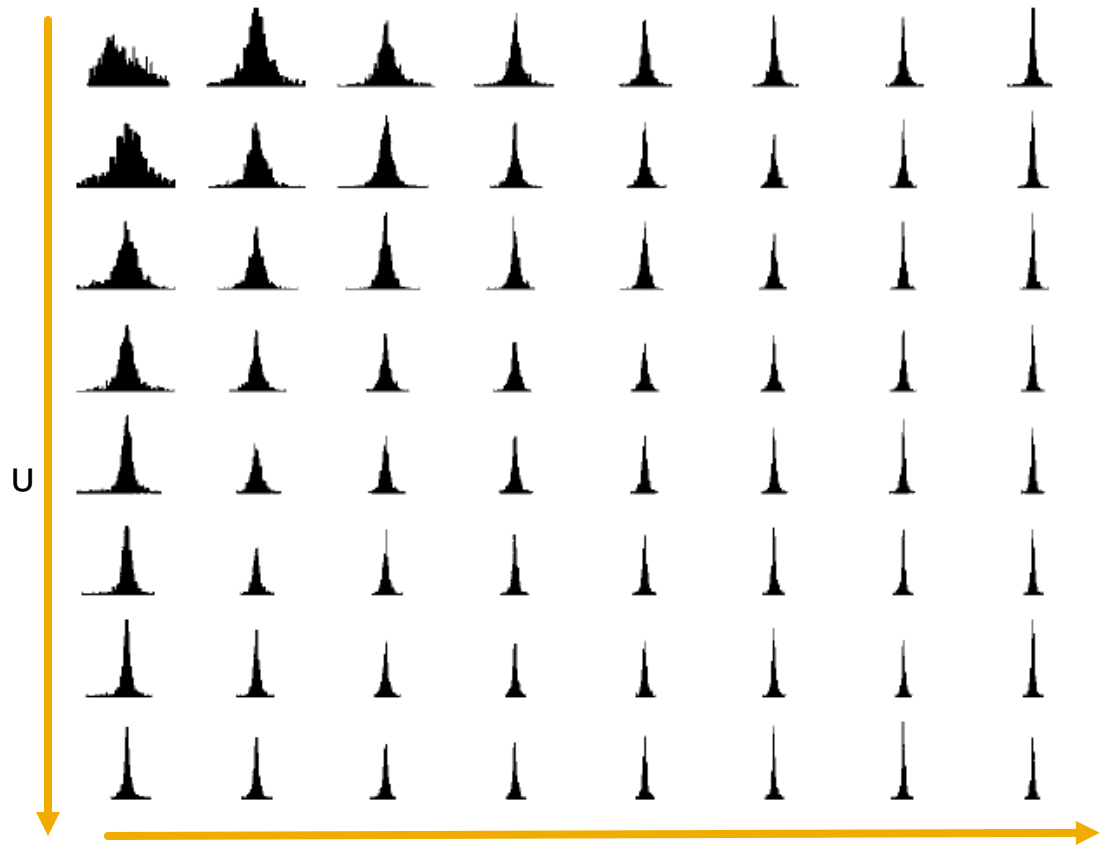
CR/CR(1,1) -
Notice it can be approximated by the form shown two slides before, with ρ~0.99

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.0000 | 0.9888 | 0.9770 | 0.9704 | 0.9648 | 0.9599 | 0.9554 | 0.9510 |
| 0.9888 | 1.0004 | 0.9891 | 0.9768 | 0.9703 | 0.9646 | 0.9596 | 0.9548 |
| 0.9770 | 0.9891 | 1.0004 | 0.9886 | 0.9764 | 0.9698 | 0.9640 | 0.9587 |
| 0.9704 | 0.9768 | 0.9886 | 0.9994 | 0.9878 | 0.9755 | 0.9687 | 0.9627 |
| 0.9648 | 0.9703 | 0.9764 | 0.9878 | 0.9986 | 0.9870 | 0.9746 | 0.9676 |
| 0.9599 | 0.9646 | 0.9698 | 0.9755 | 0.9870 | 0.9978 | 0.9861 | 0.9734 |
| 0.9554 | 0.9596 | 0.9640 | 0.9687 | 0.9746 | 0.9861 | 0.9967 | 0.9847 |
| 0.9510 | 0.9548 | 0.9587 | 0.9627 | 0.9676 | 0.9734 | 0.9847 | 0.9951 |

CC/CC(1,1) -
Notice it can be approximated by the form shown two slides before, with ρ~0.9888

# (3) Distribution of DCT coefficients

- Due to the JPEG standard, the DCT is widely used in image processing.
- The DCT is performed on 8 x 8 blocks.
- The distribution of the DCT coefficients of a fixed frequency value $(u,v)$ across different blocks has a peculiar shape – see next slide.




DCT coefficients computed for small patches. The distribution is represented by means of a histogram per coefficient. The samples to build the distribution for each coefficient come from the image patches – for each of which the DCT was computed.

*Image source: Lam and Goodman, "A Mathematical Analysis of the DCT coefficient distributions for images", IEEE Transactions on Image Processing, 2000.*
*https://www.researchgate.net/publication/3327260_A_mathematical_analysis_of_the_DCT_coefficient_distributions_for_images*

# (3) Distribution of DCT coefficients

- The shapes of these histograms can be approximated to a high degree of accuracy by means of some distributions with a precise parametric form.

- In particular, a Laplacian distribution has been experimentally shown to be a very good fit for all except the DC coefficient ($u = v = 0$).

# Segway: Generalized Gaussian Distribution

Gaussian

$$p(x;\mu,\sigma^2)=\frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

Generalized Gaussian

$$p(x;\mu,\sigma^2,\beta)=\frac{\beta e^{-(|x-\mu|/\sigma)^\beta}}{2\sigma\Gamma(1/\beta)}$$

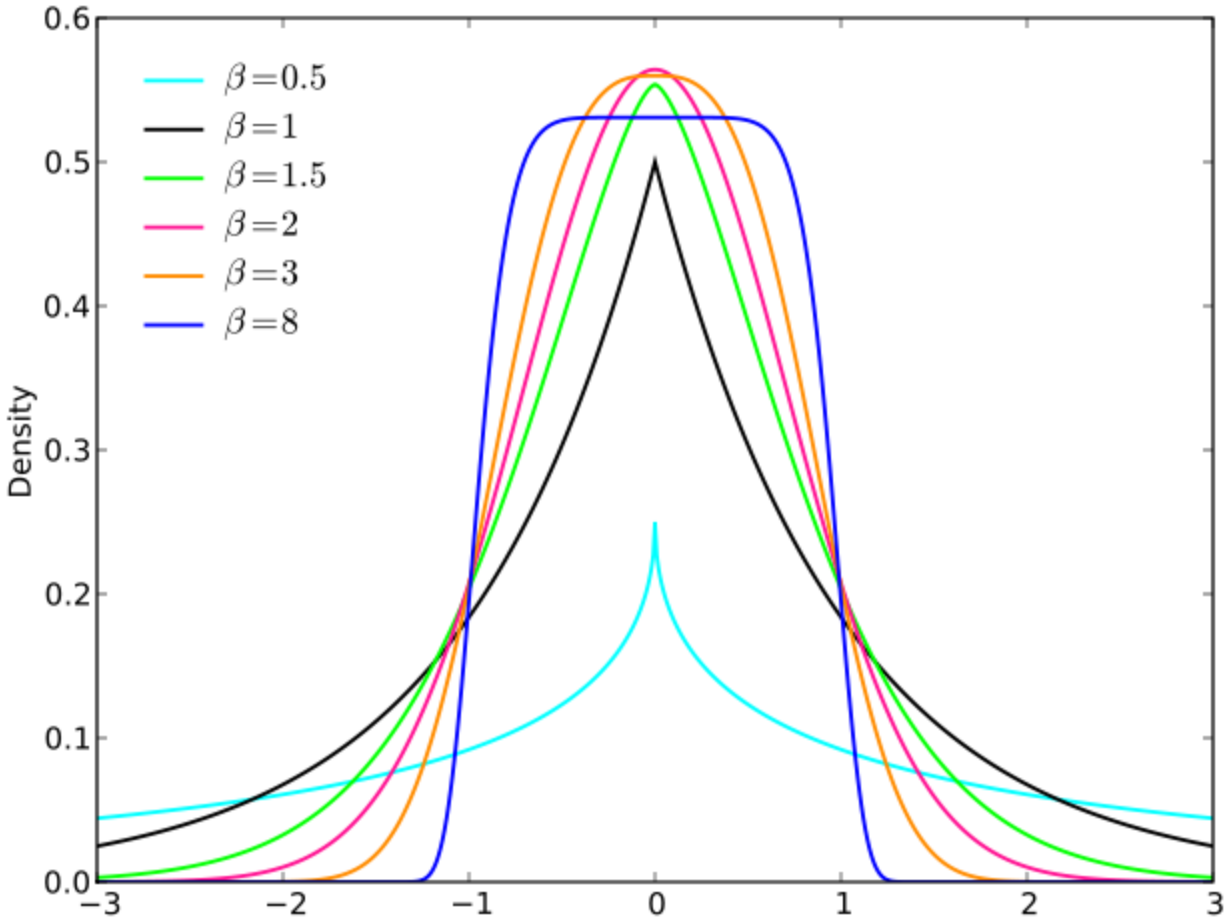
Shape parameter

Scale parameter

$\beta=2$ → Gaussian

$\beta=1$ → Laplacian

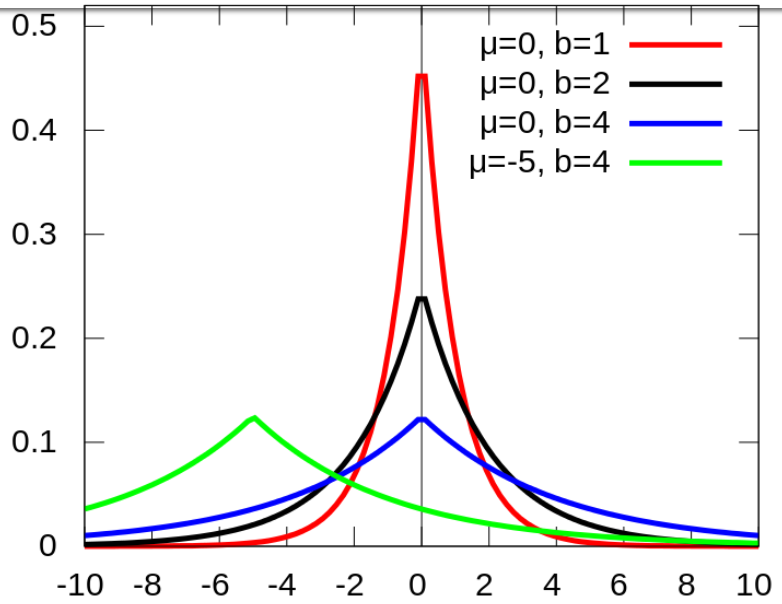
$\beta\to\infty$ → Uniform

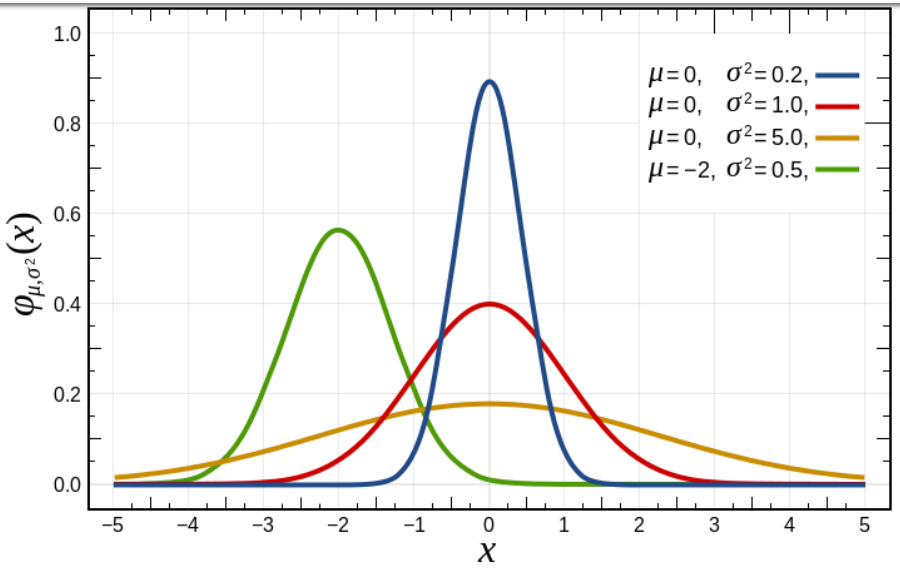



Generalized Gaussian Distributions: The Laplacian is a GGD with beta = 1.

https://en.wikipedia.org/wiki/Generalized_normal_distribution

# Laplacian versus Gaussian distribution




$$p_X(x) = \frac{1}{2b}\exp\left(\frac{-|x-\mu|}{2b}\right)$$

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

The probability density of a sample decreases as the sample value moves farther away from the mean. But for the Gaussian distribution, this decrease is much quicker (why?).

# (3) Distribution of DCT coefficients

- Is there an explanation as to why the DCT coefficients have a Laplacian distribution?
- Consider the formula for a DCT coefficient:

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)a_{NM}^{uxvy}$$

$$DCT:$$

$$a_{NM}^{uxvy} = \alpha(u)\alpha(v)\cos\left(\frac{\pi(2x+1)u}{2N}\right)\cos\left(\frac{\pi(2y+1)v}{2M}\right), u = 0...N-1, v = 0...M-1$$

$$\alpha(u) = \sqrt{1/N}\ (u=0), \text{else}\ \alpha(u) = \sqrt{2/N}$$

$$\alpha(v) = \sqrt{1/M}\ (v=0), \text{else}\ \alpha(v) = \sqrt{2/M}$$

$$\tilde{a}_{NM}^{uxvy} = a_{NM}^{uxvy}$$

By Lindeberg's central limit theorem, the distribution of F(*u*,*v*) should be Gaussian – the weighted summation of identically distributed random variables can be approximated as Gaussian.

# (3) Distribution of DCT coefficients: Lindeberg's central limit theorem

- Consider independent random variables $X_k$, $1 \leq k \leq n$ – with mean $\mu_k$ and std. dev. $\sigma_k$ respectively. Here $n = N \times N$.
- Define $s_n^2 = \sum_{k=1}^{n} \sigma_k^2$ .

https://en.wikipedia.org/wiki/Lindeberg%27s_condition

- If the following condition holds, then the distribution of the random variable $Z_n$ converges to N(0,1) as $n \rightarrow \infty$:

$$\forall \varepsilon > 0, \lim_{n\to\infty} \frac{\sum_{k=1}^{n} E\left[(X_k - \mu_k)^2 1_{\{|X_k - \mu_k| > \varepsilon s_n\}}\right]}{s_n^2} = 0$$

$$Z_n = \frac{\sum_{k=1}^{n} (X_k - \mu_k)}{s_n}$$

# (3) Distribution of DCT coefficients: Lindeberg's central limit theorem

- If we consider all pixels in a $N$ x $N$ ($n=N^2$) patch to be random variables with variance $\sigma^2$, then we have:

$$\sigma_k^2 = Var\left( \alpha(u)\alpha(v)\cos\left(\frac{\pi(2x+1)u}{2N}\right)\cos\left(\frac{\pi(2y+1)v}{2N}\right) f_k \right)$$

$$\leq \frac{1}{n} Var(f_k) = \sigma^2 / n; \quad n = N^2$$

Spatial index $k$ corresponding to location (x,y)

Random variable $X_k$

- Clearly the sufficient condition for Lindeberg's CLT is satisfied in our scenario! This is because the variance of no one r.v. dominates over the sum $s_n$.

# (3) Distribution of DCT coefficients

- Note that the distribution is Gaussian even if the pixel values in a patch are spatially correlated – as long as the correlation coefficient is less than 1.
- A small number of elements in the weighted summation is enough for the Gaussian distribution via the central limit theorem.
- But note that this is under the assumption that the pixel values are identically distributed.
- This assumption is true within a patch but not true across patches – as the **variance** of the intensity values in different patches can be very different!

# (3) Distribution of DCT coefficients

- In terms of mathematical equations:

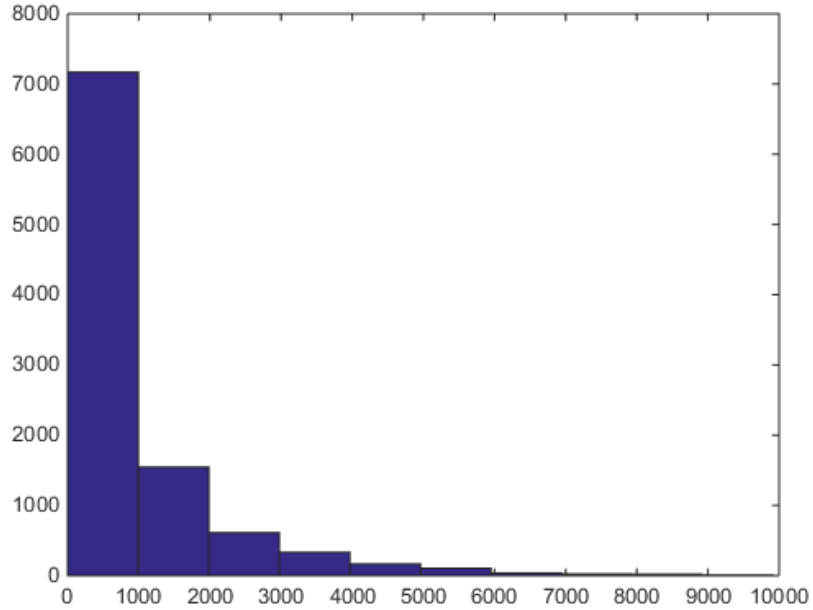
$$p(F(u,v)\,|\,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(F(u,v))^2}{2\sigma^2}\right)$$

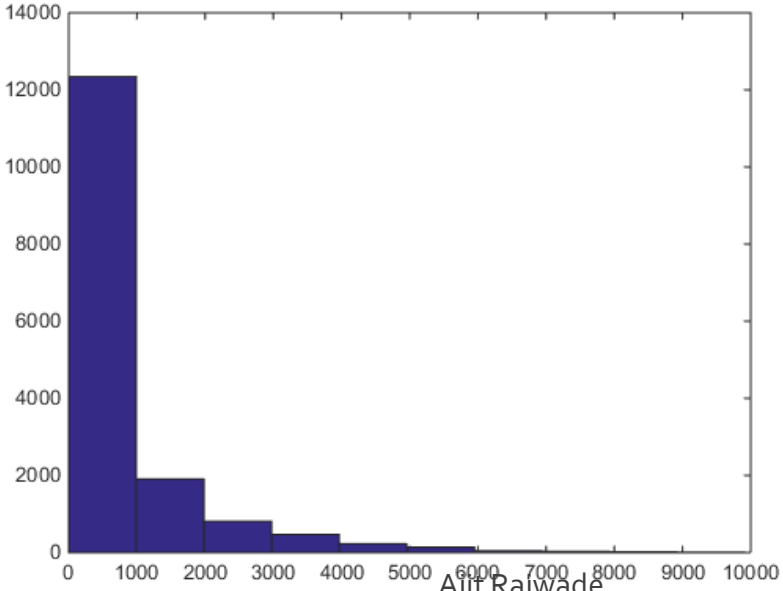
$$\therefore\ p(F(u,v)) = \int_0^\infty p(F(u,v)\,|\,\sigma^2)p(\sigma^2)d\sigma^2$$

If $\sigma^2$ = variance of pixel intensity values, then $\mathrm{Var}(F(u,v)) = \sigma^2$

- What can we say about $p(\sigma^2)$?

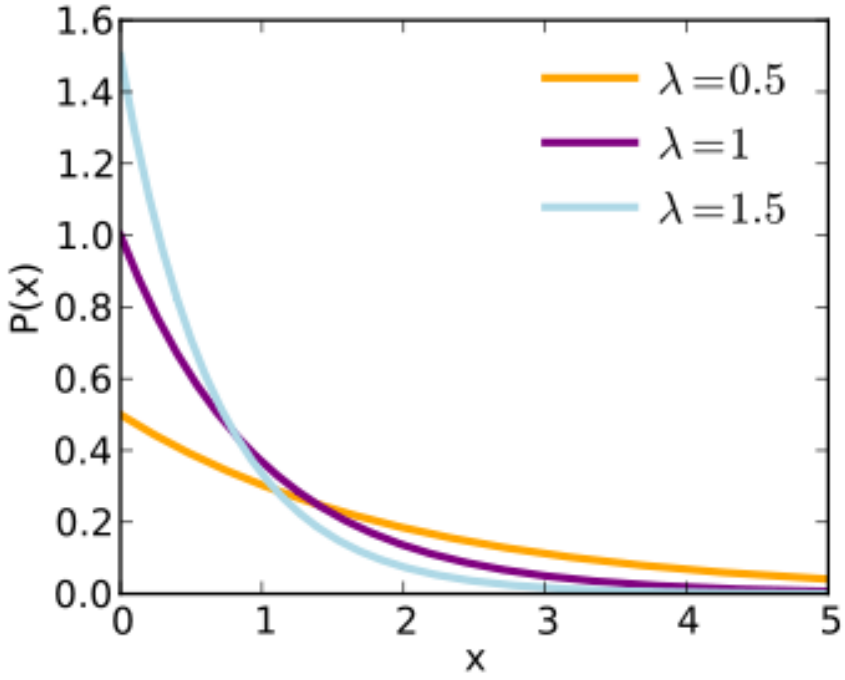
Histogram of patch variance values – collected over all non-overlapping patches of size 8 x 8 from a grayscale image




# (3) Distribution of DCT coefficients

- What can we say about $p(\sigma^2)$?
- It can be modelled quite closely by an exponential distribution.




$$p(\sigma^2) = \lambda \exp(-\lambda \sigma^2)$$

https://en.wikipedia.org/wiki/Exponential_distribution

# (3) Distribution of DCT coefficients

- So let us complete the math!
- We have

$$\therefore p(F(u,v)) = \int_0^\infty p(F(u,v)\,|\,\sigma^2)p(\sigma^2)d\sigma^2$$

$$= \int_0^\infty \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(F(u,v))^2}{2\sigma^2}\right)\lambda\exp(-\lambda\sigma^2)d\sigma^2$$

$$= \lambda\sqrt{\frac{2}{\pi}}\int_0^\infty \exp\left(-\frac{(F(u,v))^2}{2\sigma^2}-\lambda\sigma^2\right)\lambda d\sigma$$

$$= \frac{\lambda}{2}\sqrt{\frac{2}{\pi}}\sqrt{\frac{\pi}{\lambda}}\exp\left(-2\sqrt{\frac{\lambda\,|\,F(u,v)\,|^2}{2}}\right)$$

$$= \frac{\sqrt{2\lambda}}{2}\exp\left(-\sqrt{2\lambda}\,|\,F(u,v)\,|\right)$$

$$\int_0^\infty \exp\left(-ax^2 - b/x^2\right)dx = \frac{1}{2}\sqrt{\frac{\pi}{a}}\exp\left(-2\sqrt{ab}\right)$$

This is a Laplacian distribution!

# (3) Distribution of DCT coefficients

- The widths of the Laplacian distribution of the coefficients however decrease with increase in frequency – go back a few slides.
- Why so?
- Remember that each DCT coefficient is a **weighted** summation of the intensity values.
- The weights are of the following form:

$$\cos((2x+1)\pi u/n)\cos((2y+1)\pi v/n)$$

# (3) Distribution of DCT coefficients

- The weights are of the following form:

$$\cos((2x+1)\pi u/n)\cos((2y+1)\pi v/n)$$

- At lower frequencies, the weights applied to adjacent pixels are close in value, and this increases the magnitude of the summation.
- At higher frequencies, the weights applied to adjacent pixels are of similar magnitude but opposite sign, due to which spatially correlated values nullify each other giving smaller-valued summations. Remember: most images are **piecewise smooth**!
- Hence the average value of the high frequency coefficients is often small in magnitude, and the average value of low frequency coefficients is large in magnitude.
- This also affects the variances and second order uncentralized moments.

# (3) Distribution of DCT coefficients

- The DCT coefficients of small patches of images are quite sparse – very few of them have significant value and the rest have value close to 0.

- The distribution is heavy-tailed, i.e. the probability of values significantly larger than the mean does not go to zero as quickly as a Gaussian distribution would predict!

# (3) Distribution of DCT coefficients

- If we assumed a different prior for $p(\sigma^2)$, note that the final distribution will be different from a Laplacian.

- In general, the DCT coefficients of natural images can be modelled as a Generalized Gaussian distribution with shape parameter less than or equal to 1.

# (4) Distribution of Wavelet coefficients

- We will study first a very simple form of the wavelet transform – called the Haar wavelet.
- The Haar wavelet is a sequence of rescaled square-shaped functions which together form an orthonormal basis.
- The wavelet transform basically involves expressing the image as a linear combination of Haar wavelet basis functions.

# (4) Distribution of Wavelet coefficients

- The basic Haar wavelet functions are shown below:

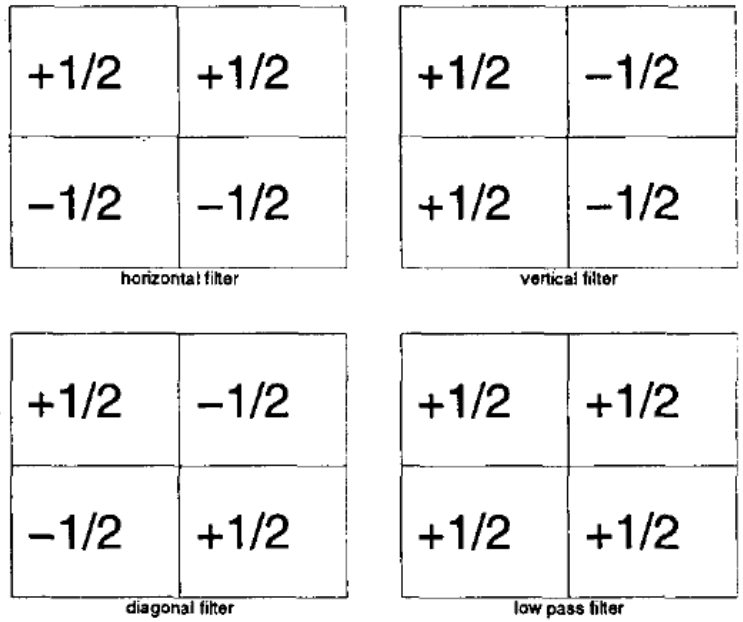



Figure 5: Haar Filters

Corresponding orthonormal basis (each column is a basis vector)

$$\begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{pmatrix}$$

*Image source: Huang & Mumford, Statistics of Natural Images and Models, CVPR 1999*

http://www.dam.brown.edu/ptg/MDbook/Huangthesis.pdf

# (4) Distribution of Wavelet coefficients

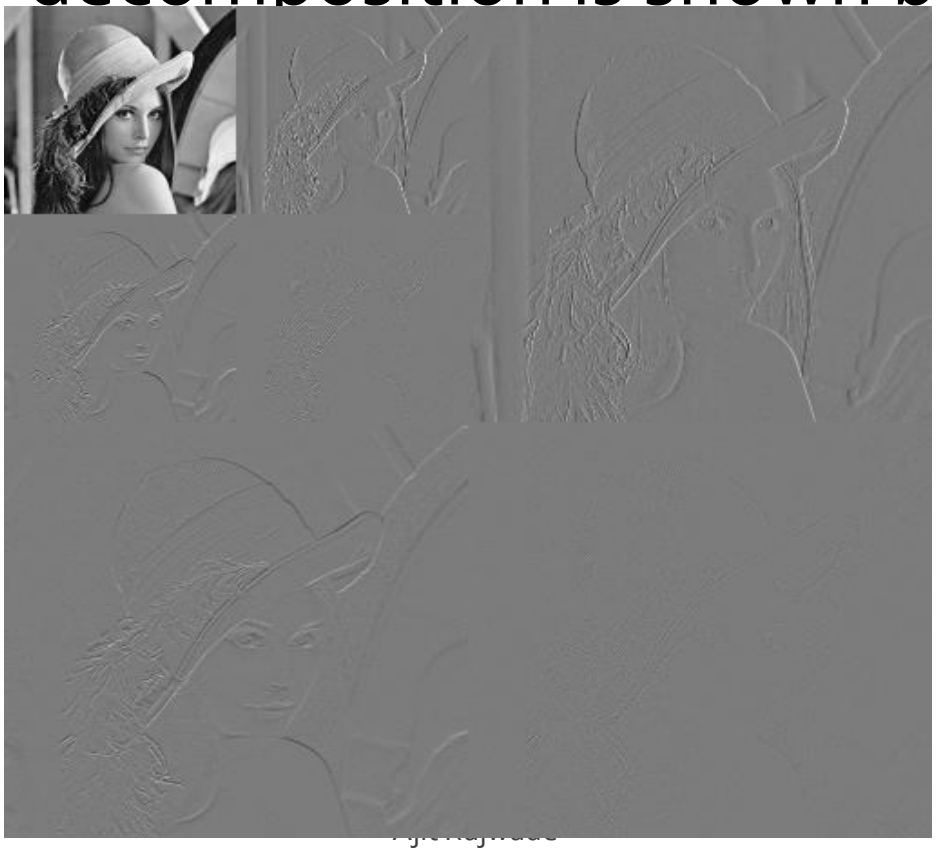
- The wavelet transform is computed in a multi-scale manner.
- Imagine you had a 64 x 64 image.
- Take each 2 x 2 patch (non-overlapping) and compute the four wavelet coefficients – low-pass, vertical, horizontal and diagonal.

# (4) Distribution of Wavelet coefficients

- Each set of coefficients can be organized to form a 32 x 32 image – called a *sub-band* image.
- This is called as a *level 1 wavelet decomposition*.
- The low-pass sub-band can then be subjected to a second-level wavelet decomposition to generate 16 x 16 sub-band images.
- The level two 16 x 16 low-pass sub-band can then be subjected to a level 3 wavelet decomposition, and so on till a maximum level 6 (yielding a single coefficient).

# (4) Distribution of Wavelet coefficients

- A sample level-two Haar wavelet decomposition is shown below:



# (4) Distribution of Wavelet coefficients

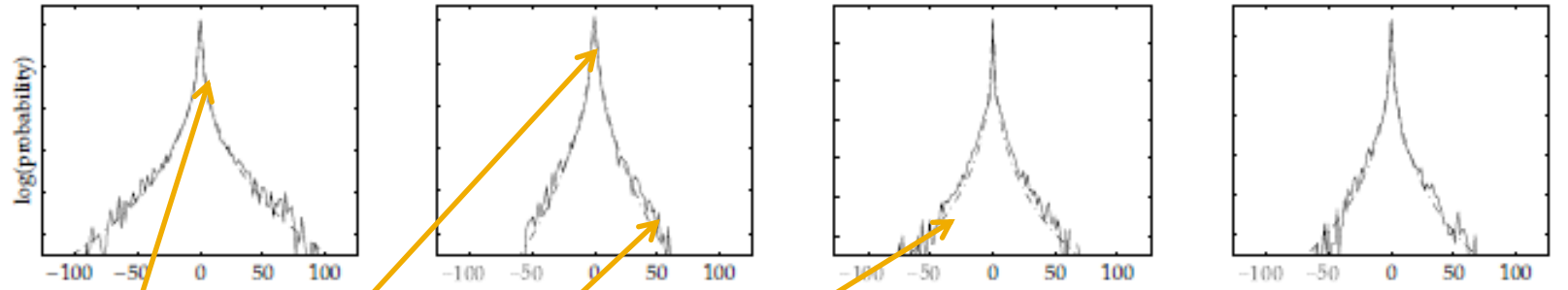



**Figure 1.** Examples of 256-bin coefficient histograms for vertical bands of four images ("Boats", "Lena", "CTscan", and "Toys"), plotted in the log domain. Also shown (dashed lines) are fitted model densities corresponding to equation (1). Below each histogram is the sample kurtosis (fourth moment divided by squared variance), and the relative entropy of the model.

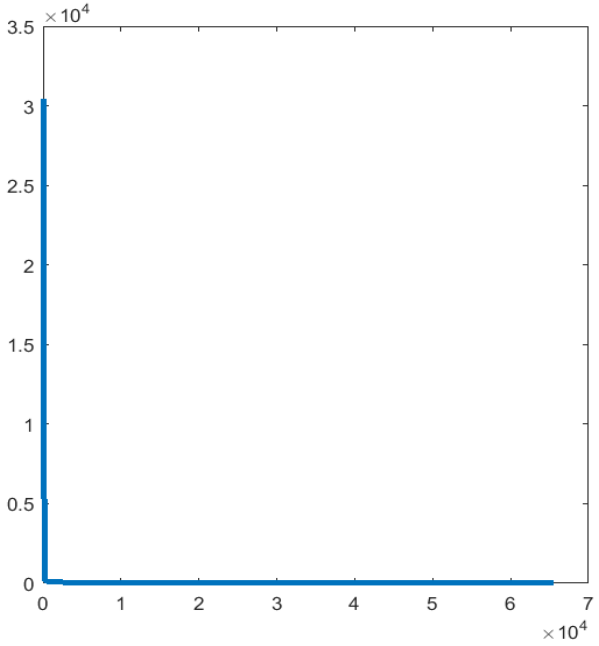
Smoother regions: larger coefficient values

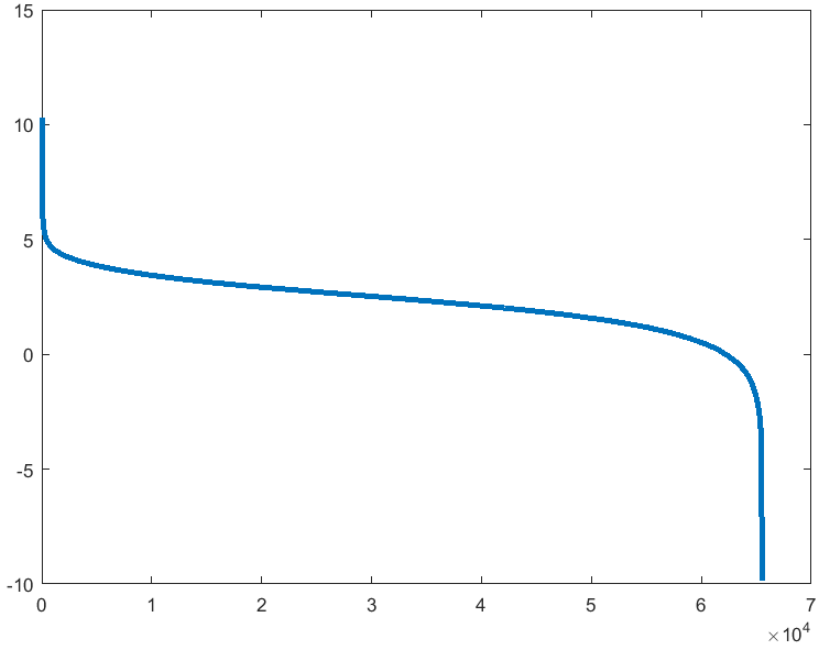
Textured regions/edges: smaller values

*Image source: Simoncelli, Bayesian denoising of visual images in the wavelet domain, 1999, [http://www.cns.nyu.edu/pub/lcv/simoncelli98e.pdf](http://www.cns.nyu.edu/pub/lcv/simoncelli98e.pdf)*

# (4) Distribution of Wavelet coefficients

- Follow exponential decay rule (like Fourier coefficients) – small coefficients can be ignored, the remaining can be coded.



512 x 512 Barbara image

Image reconstructed from top 80,000 largest DCT coefficients

# (4) Distribution of Wavelet coefficients

- The significant wavelet coefficients can be (say) Huffman encoded using their histogram.
- This can be used in image compression algorithms.
- **But you can do even better!**

# (4a) Joint Statistics of Haar wavelet coefficients

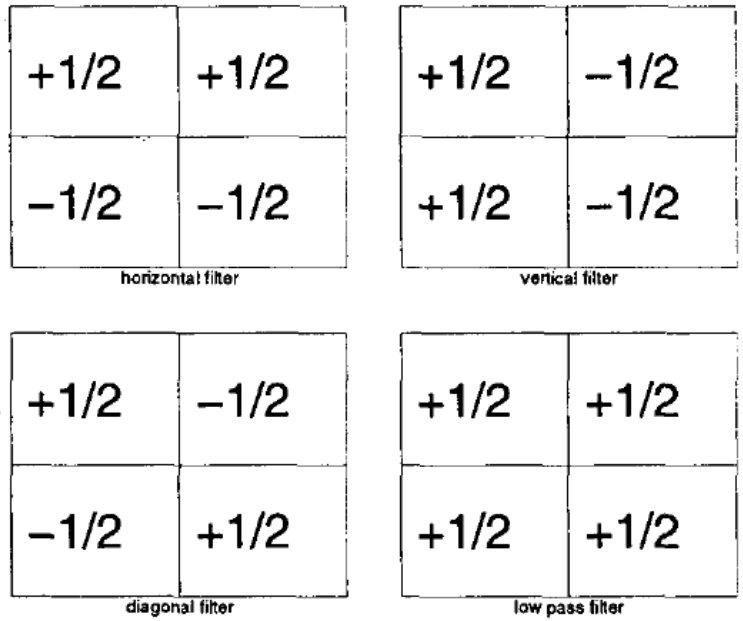


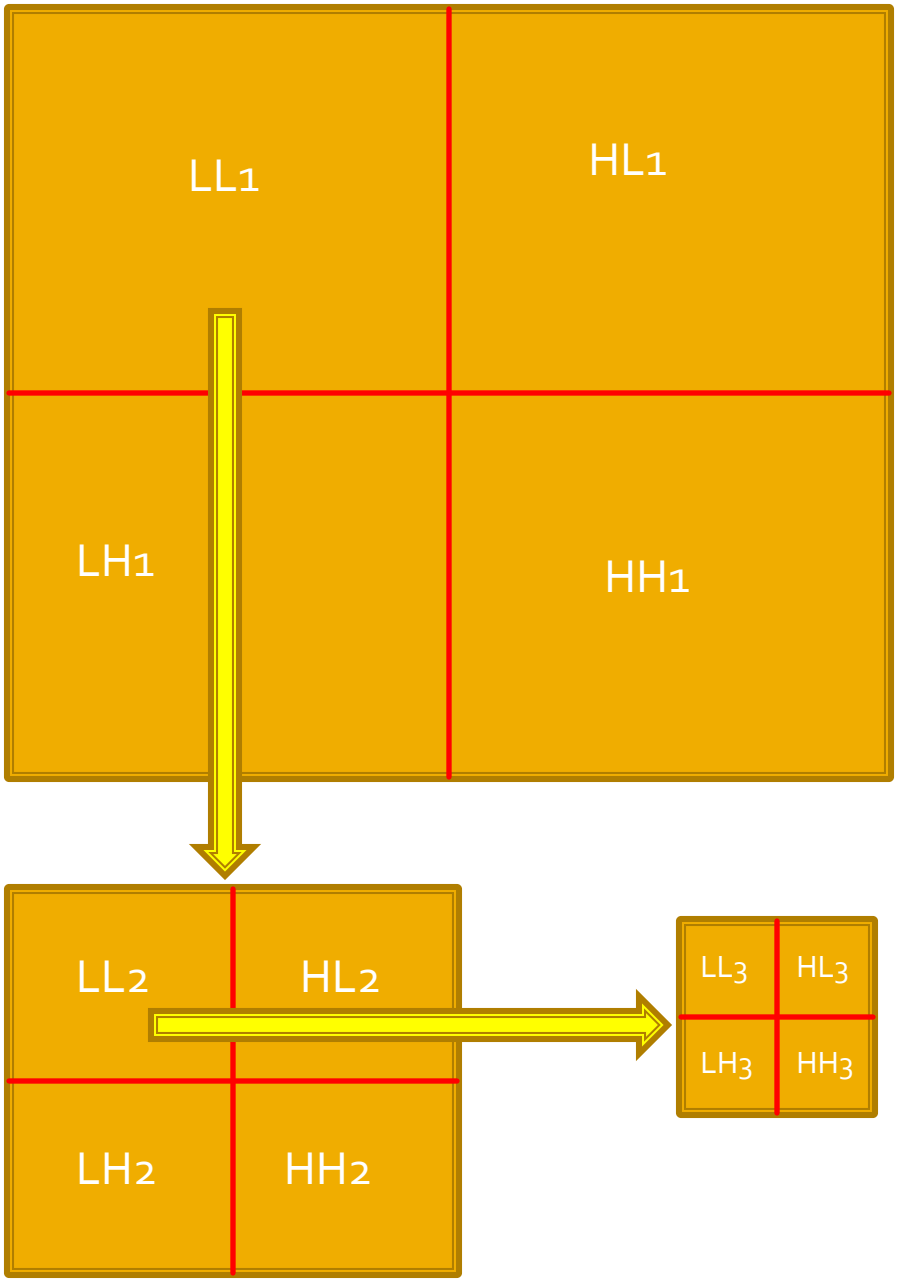

Figure 5: Haar Filters

*Image source: Huang & Mumford, Statistics of Natural Images and Models, CVPR 1999*

Coefficients computed at multiples scales of the Haar wavelet pyramid

Concept of **parent**, **child**, **sibling** and **cousin** coefficients (all are called wavelet sub-bands). **Sibling** = adjacent spatial locations in a sub-band, **cousins** = same spatial location at adjacent orientations.

http://www.dam.brown.edu/ptg/MDbook/Huangthesis.pdf




Three-level wavelet decomposition of an image

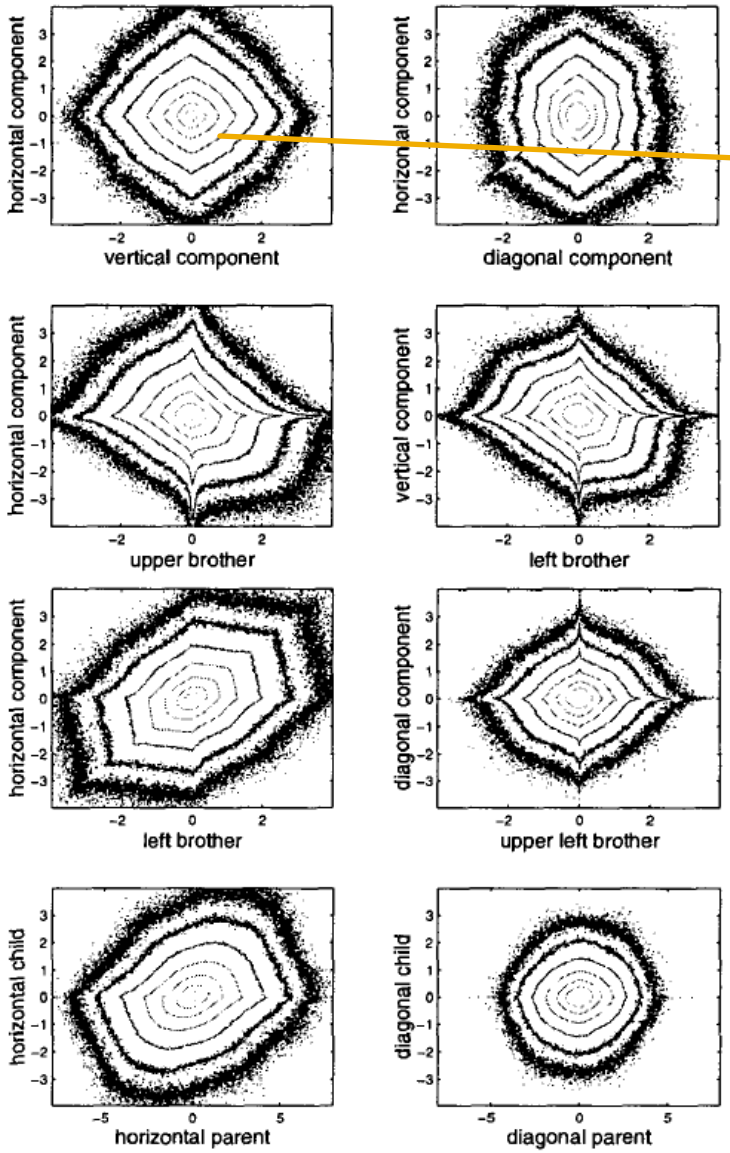



Figure 8: Contour Plot of the log(histogram) of several wavelet coefficient pairs

$$|x|+|y|=1$$

$$f(ch, cv) = e^{C_1+C_2(|ch|+|cv|)^{\alpha}}$$

These joint statistics reveal that wavelet coefficients are NOT independent

*Image source: Huang & Mumford, Statistics of Natural Images and Models, CVPR 1999*

http://www.dam.brown.edu/ptg/MDbook/Huangthesis.pdf

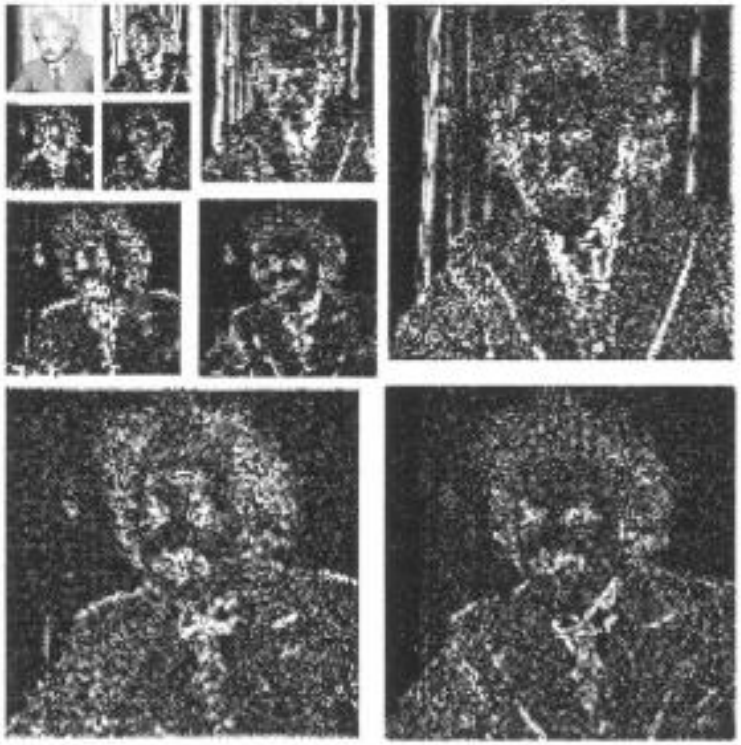
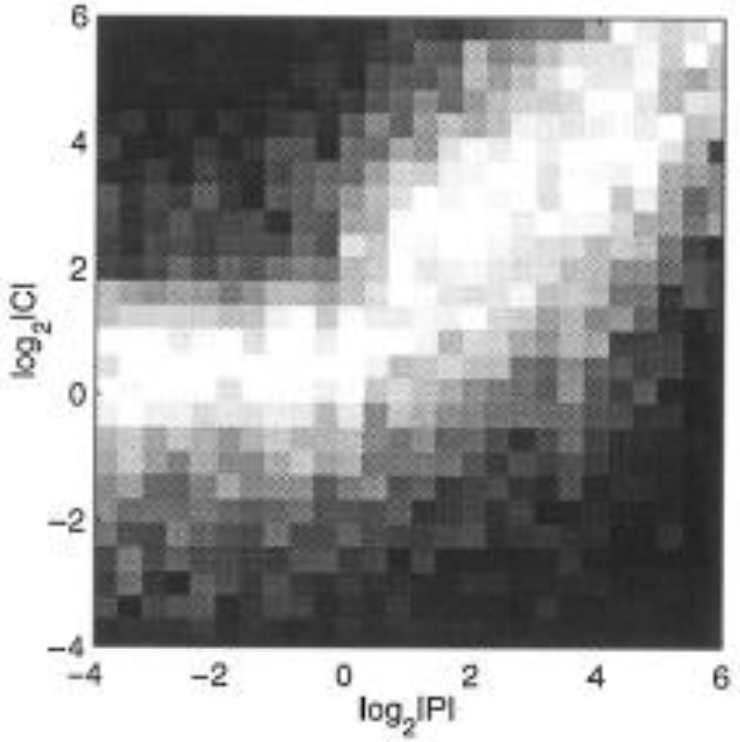
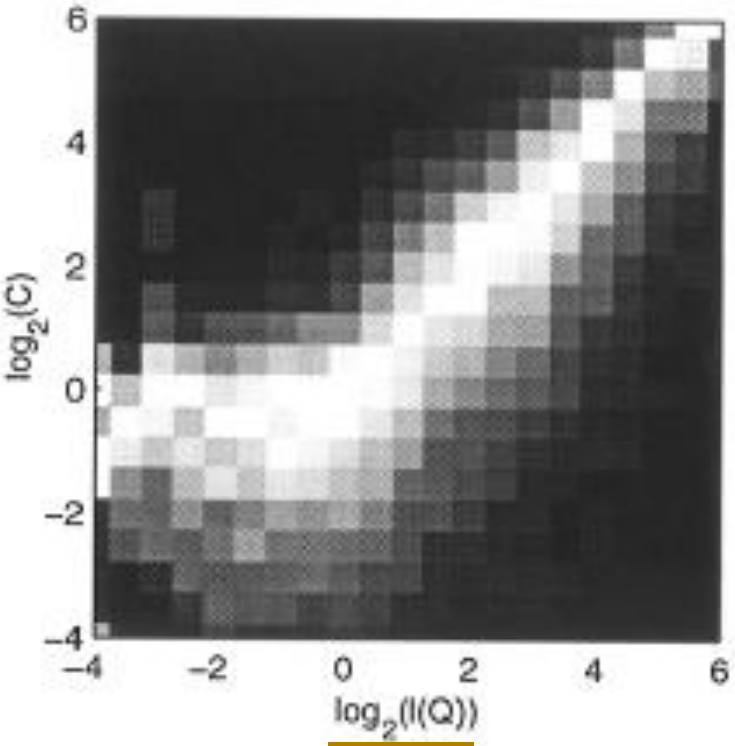


*Image source: Buccigrossi et al, Image Compression via Joint Statistical Characterization in the Wavelet Domain.* [http://www.cns.nyu.edu/pub/lcv/buccigrossi97a.pdf](http://www.cns.nyu.edu/pub/lcv/buccigrossi97a.pdf)

Fig. 3. Coefficient magnitudes of a wavelet decomposition. Shown are absolute values of subband coefficients at three scales, and three orientations of a separable wavelet decomposition of the Einstein image. Also shown is the lowpass residual subband (upper left). Note that high-magnitude coefficients of the subbands tend to be located in the same (relative) spatial positions.

Large magnitude coefficients tend to occur at neighboring spatial locations within a sub-band, or at the same locations in sub-bands of adjacent scale/orientation, or at related spatial locations in parent and child sub-bands.



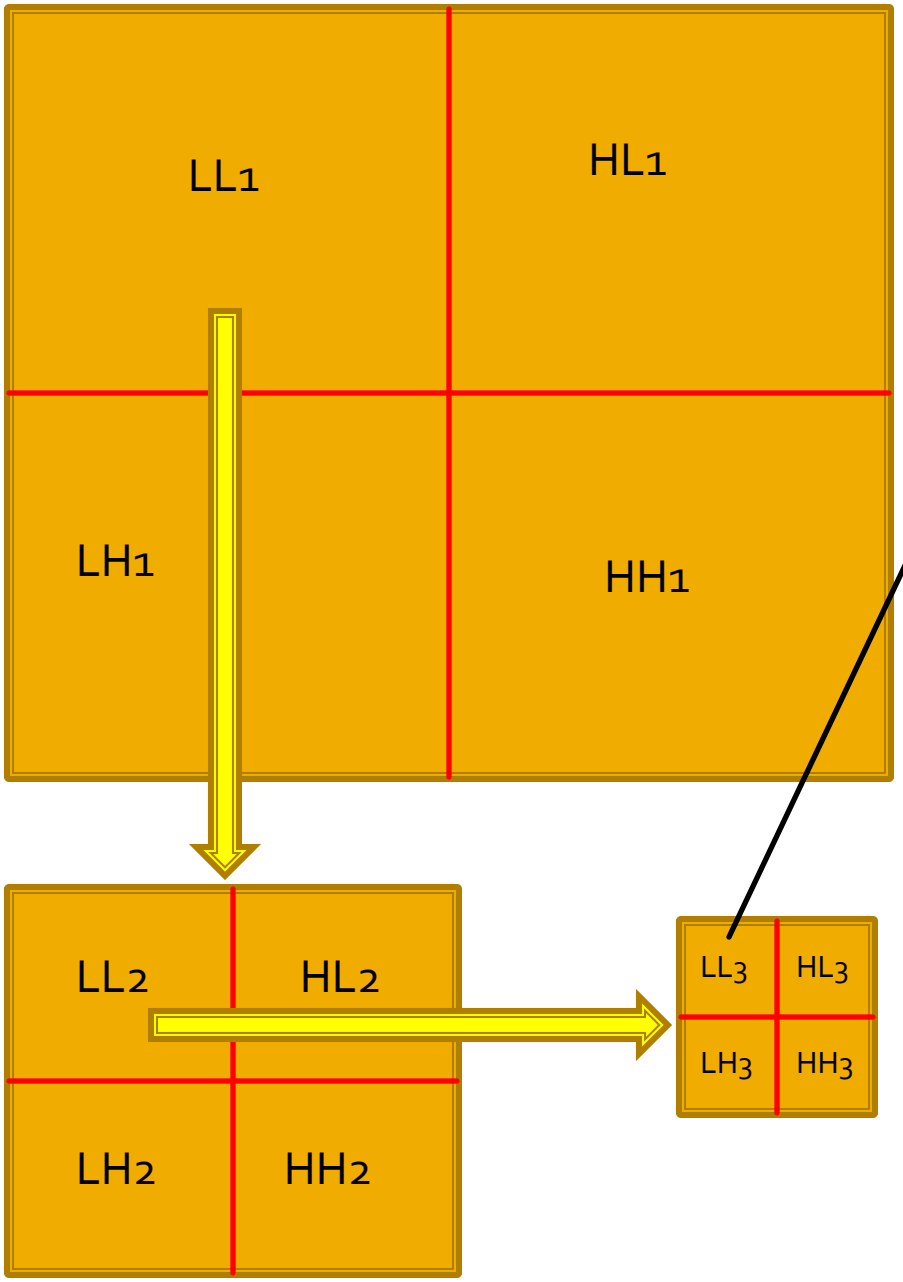




Joint histogram of logarithms of absolute value of child and parent coefficients (also true for coefficients at adjacent orientations) – features like prominent edges or textures have large magnitude coefficients at multiple scales

Joint histogram of logarithm of squared value of child coefficient, and logarithm of square of linear combination of squared values of coefficients from neighboring sub-bands

$$l(\vec{Q}) \equiv \vec{w} \cdot \vec{Q} = \sum_k w_k Q_k$$

$$\vec{w} = \mathcal{E}(\vec{Q}\vec{Q}^T)^{-1} \cdot \mathcal{E}(C' \cdot \vec{Q})$$




$$c(x,y) \approx \sum_{k=1}^{\#neighbors} w_k Q_k^2(x_k, y_k)$$

Wavelet coefficient from neighboring sub-band (at corresponding location)

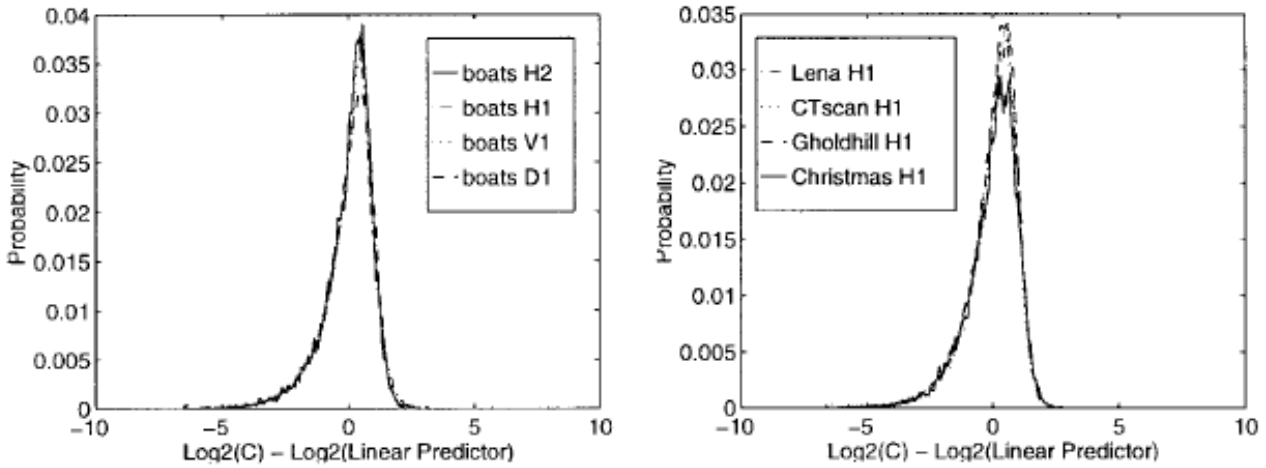



Fig. 6. Comparison of conditional distributions in the log domain of different subbands and images. Distributions were normalized (in the log domain) to have mean zero and variance one. Left: comparison of distributions for different subbands of the boats image. Right: comparison of distributions for different images (Lena, Goldhill, CT scan, Christmas).

Conditional distributions of a wavelet coefficient (log absolute) given linear combinations of its neighbors (log absolute) – the shapes are robust across images! The plots are mean and variance normalized.

This redundancy means that we need not store all the wavelet coefficients – we can just predict some coefficients directly given their neighbors using the linear model that was fit. **Useful for lossy image compression!**

# Why study Natural Image Statistics: Bayesian Framework

$$y = \mathcal{H}x + \varepsilon$$

Observation

Known operator

Unknown (to be determined) signal

noise

$p(y \mid x)$ → Likelihood

$p(x)$ → **Prior Model on signal**

Posterior probability

Bayes rule

$$p(x \mid y) = \frac{p(y \mid x)\,p(x)}{p(y)} \propto p(y \mid x)\,p(x)$$

# Why study Natural Image Statistics: Bayesian Framework

$$y = \mathcal{H}x + \varepsilon$$

Assuming zero-mean i.i.d. (additive) Gaussian noise model, the likelihood is as follows:

NOTE: Likelihood derived from the assumed noise model

$$p(y \mid x) \propto \exp\left( -\frac{\|y - \mathcal{H}x\|^2}{2\sigma^2} \right)$$

# Bayesian Framework: To Estimate x

- Maximum a posteriori (MAP) estimate:

$$\hat{x} = \arg\max_x p(x \mid y) = \arg\max_x p(y \mid x)\, p(x) / p(y)$$

$$= \arg\max_x p(y \mid x)\, p(x)$$

As y does not affect maximization w.r.t. x

The MAP estimate asks the following question: Given the observation y, what x is the most likely, taking into account that we have prior information on x in the form of p(x)?

If p(x) were a uniform distribution (or effectively we had no prior information about x), then MAP reduces to maximizing p(y|x) – which is called the maximum likelihood estimate.

# Bayesian Framework: To Estimate x

- Minimum mean square error (MMSE) estimate:

$$\hat{x} = \arg\min_z \int \|x - z\|^2 p(x \mid y) dx$$

Prior is important!

$$\therefore \hat{x} = \frac{\int x p(x \mid y) dx}{\int p(x \mid y) dx} = \frac{\int x p(y \mid x) p(x) dx}{\int p(y \mid x) p(x) dx} = E(x \mid y)$$

Integrate to 1

# Simple Example: 1

Prior

$$p(x=10)=0.7$$

$$p(x=15)=0.3$$

Likelihood

$$y=x+\varepsilon$$

$$\varepsilon \sim N(0,\sigma), \sigma = 2$$

Observed Value of $y$ = 14. Determine $x$ given $y$ and the knowledge of the noise model (likelihood) and prior on $x$.

# Simple Example :1

$$\widehat{x}_{MAP} = \arg\max_x \; p(x \mid y) = \arg\max_x \; p(y \mid x)\, p(x)$$

$$= \arg\max[0.7 \times e^{-(14-10)(14-10)/(2\times 4)}, 0.3 \times e^{-(14-15)(14-15)/(2\times 4)}]$$

$$= \arg\max[0.0947, 0.2647]$$

$$= 15$$

$$\widehat{x}_{MMSE} = \frac{\int x p(y \mid x)\, p(x) dx}{\int p(y \mid x)\, p(x) dx}$$

$$= \frac{10(e^{-2})(0.7) + 15(e^{-1/8})(0.3)}{(e^{-2})(0.7) + (e^{-1/8})(0.3)} = 13.6825$$

# Simple Example: 2

$$x \sim N(1,2), \mu_x = 1, \sigma_x = 2$$

$$y = x + \varepsilon$$

$$\varepsilon \sim N(0,3), \mu_y = 0, \sigma_y = 3$$

Observed Value of y = 2. Determine x given y and the knowledge of the noise model (likelihood) and prior on x.

# Simple Example: 2

$$\hat{x}_{MAP} = \arg\max_x \; p(x \mid y) = \arg\max_x \; p(y \mid x)\, p(x)$$

$$= \arg\max_x \; e^{-(y-x)^2/(2*3*3)} e^{-x^2/(2*2*2)}$$

$$= \arg\max_x \; e^{-(x-\hat{\mu})^2/(2\hat{\sigma}^2)}$$

$$= \hat{\mu}(why\,?)$$

$$\hat{\mu} = \frac{\mu_x \sigma_y^{\,2} + \mu_y \sigma_x^{\,2}}{\sigma_x^{\,2} + \sigma_y^{\,2}} = \frac{1*4+0}{4+9} = \frac{4}{13}$$

# Simple Example: 2

$$\widehat{x}_{MMSE} = \frac{\int x p(y \mid x) p(x) dx}{\int p(y \mid x) p(x) dx}$$

$$= \frac{\int x e^{-(x-\hat{\mu})/(2*\hat{\sigma}^2)} dx}{\int e^{-(x-\hat{\mu})/(2*\hat{\sigma}^2)} dx} = \frac{\hat{\sigma}\sqrt{2\pi}\,\hat{\mu}}{\hat{\sigma}\sqrt{2\pi}} = \hat{\mu} = \frac{4}{13}$$

When the likelihood and prior are both Gaussian, the MAP and MMSE estimates are equal.

# Maximum Likelihood Estimation

$$\widehat{x}_{ML} = \arg\max_x \; p(y \mid x)$$

If there is only one observation sample available (assume Gaussian noise), what is the maximum likelihood estimate of *x*?

If there are some *N* observation samples available (under Gaussian noise), what is the maximum likelihood estimate of *x*?

These two examples were very simple and involved scalar quantities. In future lectures, we will use more complex examples, where the unknown quantity $x$ will be multivariate – in fact, it will be an image.

We will study the applications of natural image statistics in the following applications:

(1) Image denoising and deblurring
(2) Scene categorization
(3) Image denoising (another flavour)
(4) Reflection Removal

# Application in Denoising or Deblurring

# Application in Denoising

- Consider the following noise model:
  $$y = x + n, n \sim N(0, \sigma)$$

- Given *y*, and knowing σ, determine the underlying image *x*.

- Exploit the prior (fact) that the image *x* has DCT coefficients which are Laplacian distributed.

# Application in Denoising

- Let the DCT coefficients be given as follows:

$$\theta = U^T x \text{ where } U \text{ is the orthonormal 2D DCT matrix}$$

- So the estimation problem is

$$\hat{\theta} = \arg\max_\theta \; p(y \mid U\theta)\, p(\theta)$$

$$= \arg\max_\theta \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - U\theta\|_2^2}{2\sigma^2} \right) \frac{\lambda}{\sqrt{2}} \exp\left( -\sqrt{2}\lambda \|\theta\|_1 \right)$$

$$= \arg\max_\theta \log\left( \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - U\theta\|_2^2}{2\sigma^2} \right) \frac{\lambda}{\sqrt{2}} \exp\left( -\sqrt{2}\lambda \|\theta\|_1 \right) \right)$$

$$= \arg\min_\theta \left( \frac{\|y - U\theta\|_2^2}{2\sigma^2} \right) + \lambda \|\theta\|_1$$

$$= \arg\min_\theta \left( \|y - U\theta\|_2^2 \right) + \hat{\lambda} \|\theta\|_1$$

# Application in Deblurring

- Consider the following noise model:

$$y = h * x + n = \mathbf{H}x + n, n \sim N(0, \sigma)$$

- Given *y*, and knowing **H** and σ, determine the underlying image *x*.

- Exploit the prior (fact) that the image *x* has DCT coefficients which are Laplacian distributed.

# Application in Deblurring

- Let the DCT coefficients be given as follows:

$$\theta = U^T x \text{ where } U \text{ is the orthonormal 2D DCT matrix}$$

- So, the estimation problem is

$$\hat{\theta} = \arg\max_\theta \, p(y \mid \mathbf{H}U\theta)\, p(\theta)$$

$$= \arg\max_\theta \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - \mathbf{H}U\theta\|_2^2}{2\sigma^2} \right) \frac{\lambda}{\sqrt{2}} \exp\left( -\sqrt{2}\lambda \|\theta\|_1 \right)$$

$$= \arg\max_\theta \log\left( \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - \mathbf{H}U\theta\|_2^2}{2\sigma^2} \right) \frac{\lambda}{\sqrt{2}} \exp\left( -\sqrt{2}\lambda \|\theta\|_1 \right) \right)$$

$$= \arg\min_\theta \left( \frac{\|y - \mathbf{H}U\theta\|_2^2}{2\sigma^2} \right) + \lambda \|\theta\|_1$$

$$= \arg\min_\theta \left( \|y - \mathbf{H}U\theta\|_2^2 \right) + \hat{\lambda} \|\theta\|_1$$

# Application in deblurring

- The convolution kernel *h* is assumed to be known here – we will not immediately deal with the case where it is unknown!

- The convolution of *x* with *h* can be equivalently represented by the product of a circulant matrix **H** derived from *h*, with the signal vector *x*.

# Application in deblurring

- A circulant matrix is a matrix where each row is a right circular shift of its preceding row in the following form:

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{n-1} & . & . & . & h_1 \\ h_1 & h_0 & h_{n-1} & . & . & h_2 \\ & & & & & \\ & & & & & \\ & & & & & \\ h_{n-1} & h_{n-2} & & & h_1 & h_0 \end{pmatrix}$$

# Gaussian instead of Laplacian prior

- What would happen if you imposed a Gaussian prior on the DCT coefficients?

$$\hat{\theta} = \arg\max_\theta \, p(y \mid U\theta)\, p(\theta)$$

$$= \arg\max_\theta \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - U\theta\|_2^2}{2\sigma^2} \right) \frac{1}{\hat{\sigma}\sqrt{2\pi}} \exp\left( -\frac{\|\theta\|^2}{2\hat{\sigma}^2} \right)$$

$$= \arg\max_\theta \log\left( \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\|y - U\theta\|_2^2}{2\sigma^2} \right) \frac{1}{\hat{\sigma}\sqrt{2\pi}} \exp\left( -\frac{\|\theta\|^2}{2\hat{\sigma}^2} \right) \right)$$

$$= \arg\min_\theta \left( \frac{\|y - U\theta\|_2^2}{2\sigma^2} + \frac{\|\theta\|^2}{2\hat{\sigma}^2} \right)$$

$$= \arg\min_\theta \left( \|y - U\theta\|_2^2 \right) + \hat{\lambda}\|\theta\|^2 \text{ where } \hat{\lambda} = (\sigma / \hat{\sigma})^2$$

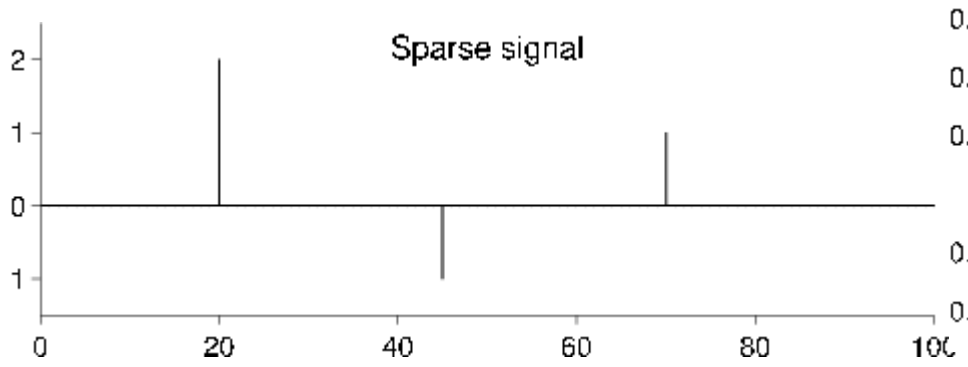
# Gaussian instead of Laplacian prior

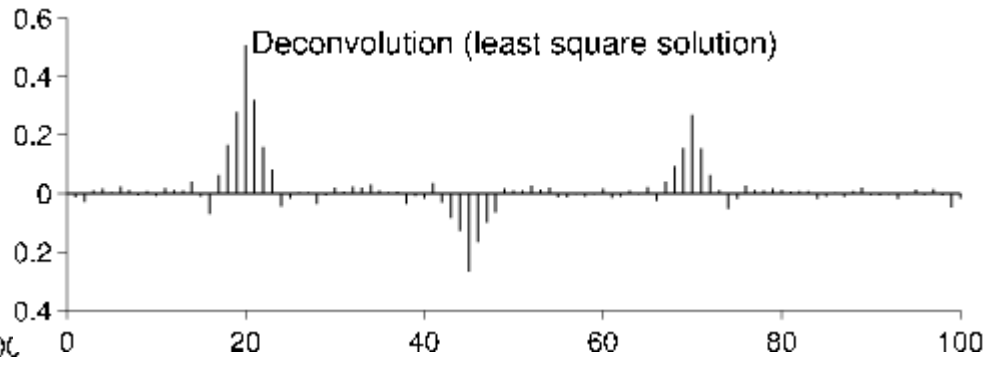
- Taking derivative w.r.t. θ, we get:

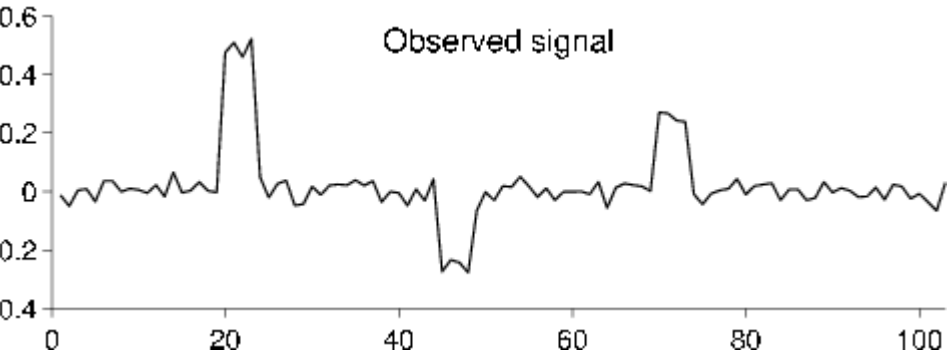
$$\hat{\theta}_i = \frac{(U^T y)_i}{1+\hat{\lambda}} = \frac{\widehat{\sigma}^2}{\sigma^2 + \widehat{\sigma}^2}(U^T y)_i$$

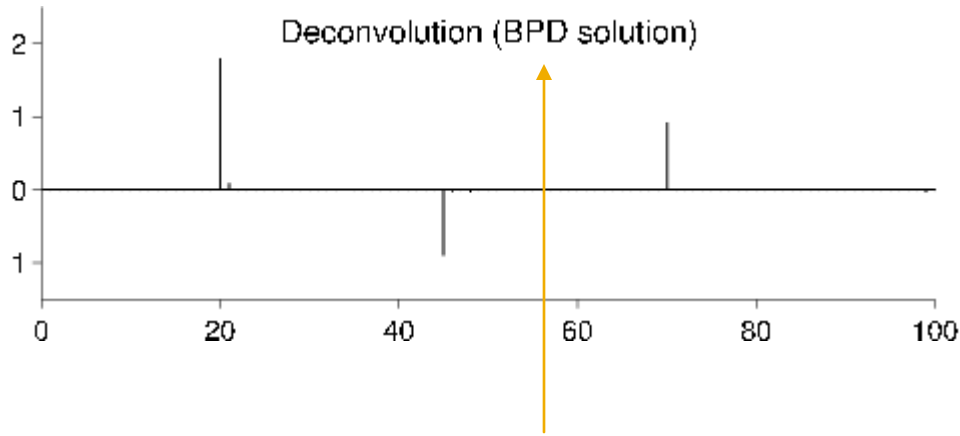
This is the Wiener filter which we have seen last semester! The Wiener filter is the optimal linear filter regardless of the signal prior, which is what we proved in CS 663. For Gaussian likelihood and Gaussian prior, the Wiener filter is the optimal filter (among linear as well as non-linear) in a MAP or MMSE sense.

- However for natural images or image patches, the Laplacian prior on the DCT or wavelet coefficients, is better suited – and yields better results in denoising.




i.e. solution with Laplacian prior

# Limitation of this model

- For some images, a GGD with shape parameter less than 1 is more suitable to model the DCT coefficients than a Laplacian.
- In such cases, the optimization problem becomes:

$$J(\boldsymbol{\theta}) = \left\| \mathbf{y} - \mathbf{A}\boldsymbol{\theta} \right\|_2^2 + \lambda \left\| \boldsymbol{\theta} \right\|_p, \; p < 1$$

- The problem however is that this is a non-convex optimization problem – and hence the local minima are different from the global minimum.
- With Laplacian or Gaussian priors, the problems were convex. Many convex problems have efficient solutions.

# Limitation of this model

- As pointed out in class, the DC coefficient is often not well modeled by the Laplacian distribution.
- The model also assumes that all DCT coefficients (or wavelet coefficients) are statistically independent – which is not true always as we shall see later.

# Statistical Compressed Sensing

- This is another view of compressed sensing based on Bayesian statistics.
- Consider compressive measurements of the form: $y = \Phi x + \eta,$

$$y \in R^m, \Phi \in R^{m \times n}, x \in R^n, m << n,$$

$$\eta \in R^m, \eta \sim N(0, \sigma^2 I_{m \times m})$$

- Suppose $x \in N(0, \Sigma)$.

# Statistical Compressed Sensing

- Consider the MAP solution for x given y and Φ:

$$\hat{x} = \arg\max_x \; p(y \mid x, \Phi)\, p(x)$$

$$= \arg\max_x \frac{1}{\sigma^{m/2}(2\pi)^{m/2}} \exp\left( -\frac{\|y - \Phi x\|_2^2}{2\sigma^2} \right) \frac{1}{|\Sigma|^{0.5}(2\pi)^{n/2}} \exp\left( -x^t \Sigma^{-1} x / 2 \right)$$

$$= \arg\max_x \log\left( \frac{1}{\sigma^{m/2}(2\pi)^{m/2}} \exp\left( -\frac{\|y - \Phi x\|_2^2}{2\sigma^2} \right) \frac{1}{|\Sigma|^{0.5}(2\pi)^{n/2}} \exp\left( -x^t \Sigma^{-1} x / 2 \right) \right)$$

$$= \arg\min_x \left( \frac{\|y - \Phi x\|_2^2}{2\sigma^2} \right) + x^t \Sigma^{-1} x / 2 + cons\tan ts$$

$$= (\Phi^T \Phi / (2\sigma^2) + \Sigma^{-1} / 2)^{-1} \Phi^T y / (2\sigma^2)$$

# Statistical Compressed Sensing

- Consider the MAP solution for x given y and $\Phi$:

$$\hat{x} = (\Phi^T \Phi / \sigma^2 + \Sigma^{-1})^{-1} \Phi^T y / (\sigma^2)$$
$$= (\Sigma - \Sigma \Phi^T [\sigma^2 I + \Phi \Sigma \Phi^T]^{-1}) \Phi^T \Sigma y / (\sigma^2)$$

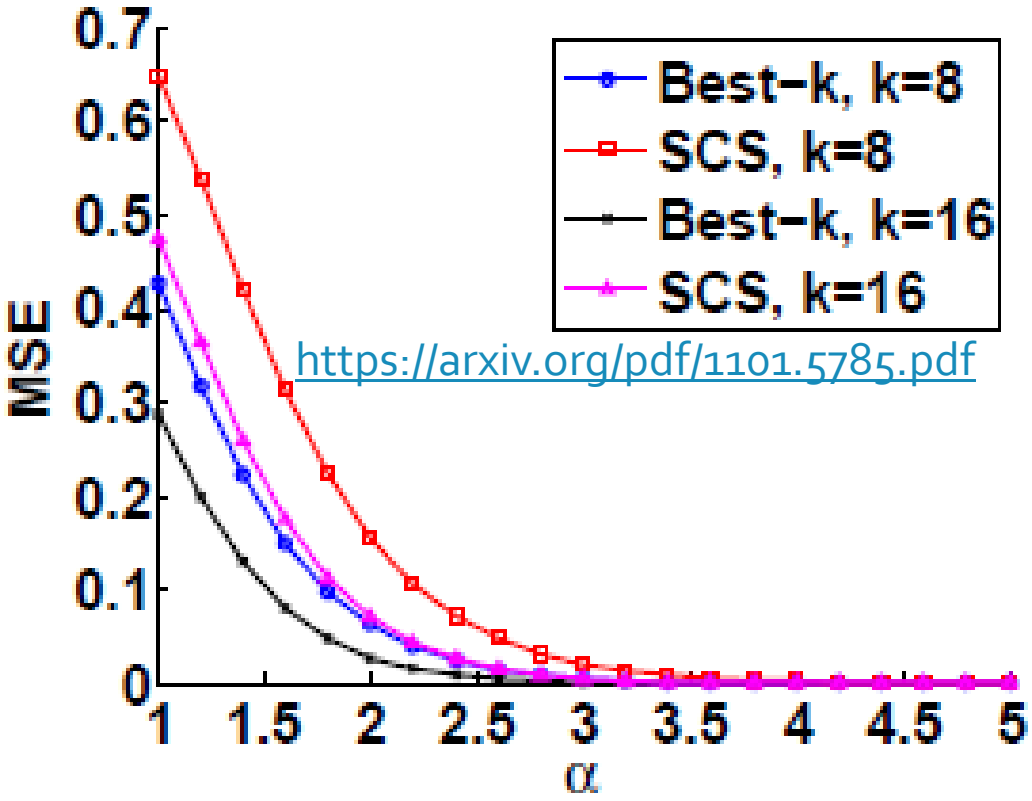
The latter expression follows by the Woodbury matrix identity.

https://en.wikipedia.org/wiki/Woodbury_matrix_identity

# Statistical Compressed Sensing

- Since the likelihood and prior are both Gaussian, this is also the MMSE solution.
- Unlike traditional compressed sensing, this solution is in closed form!
- Hence better reconstruction speed!

# Results




https://arxiv.org/pdf/1101.5785.pdf

Best-$k$ = best $k$-term approximation obtained by keeping the $k$ largest (absolute value) entries of $\mathbf{x}$ and setting the rest to 0. This is for $\mathbf{x} \sim N(\mathbf{0},\mathbf{C})$ where C is a covariance matrix with decaying eigenvalues.

$k = m$ = # of measurements

**Assumption:**
Eigen-values in the covariance matrix for the signal (i.e. Σ) are of the form: $i^{-\alpha}$ where $1 \le i \le n$. The larger the value of α, the lower the reconstruction error.

**The decay of the eigenvalues of the covariance matrix is the equivalent of signal sparsity or compressibility in an appropriate orthonormal basis.**

For example, a signal with identity covariance matrix would not yield itself to good reconstruction via this method.

# Gaussian assumption on signals?

- Statistical compressed sensing has been applied to patch-based compressed sensing (where have we encountered this?)
- Is the Gaussian assumption valid on patches?
- Not a single Gaussian, but a mixture of Gaussians – also called a Gaussian mixture model (GMM).

# Gaussian assumption on signals?

- The probability density function of an image patch is expressed as a GMM, i.e. as follows:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p_k N(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}), \boldsymbol{x} \in \mathrm{R}^{\mathrm{n}}, \boldsymbol{\mu_k} \in \mathrm{R}^{\mathrm{n}}, \sum_{k=1}^{K} p_k = 1$$

$\boldsymbol{\Sigma_k}$ is a $n$ x $n$ covariance matrix (positive semidefinite)

- In fact, GMMs are known to be universal PDF estimators.

- That is any smooth PDF can be approximated to an arbitrary degree of accuracy using an appropriate number of Gaussians.

# Gaussian assumption on signals?

- Choice of K is a bit of an art, though there are techniques for it (eg: cross-validation).
- Too large a K leads to overfitting, too small a value of K leads to underfitting.
- A GMM could be fit to a bag of some N small-sized patches using an algorithm called Expectation Maximization (EM).
- You will study EM in a machine learning or non-parametric statistics course.

# GMMs for compressed sensing

- Perform the GMM fitting for a large bag of patches.

- Consider compressive measurements **y** of a patch **x** as follows:

$$y = \Phi x + \eta,$$
$$y \in R^m, \Phi \in R^{m \times n}, x \in R^n, m << n,$$
$$\eta \in R^m, \eta \sim N(0, \sigma^2 I_{m \times m})$$

# GMMs for compressed sensing

- From **y**, we find component-wise MAP estimates (for each of the *k* different mixture components):

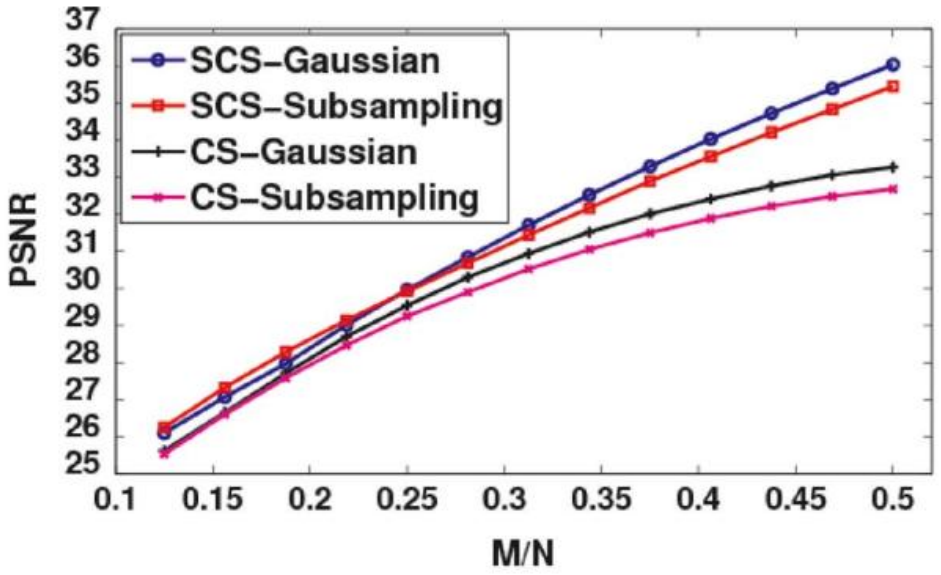
$$\hat{x}_k = \left( \frac{\Phi^T \Phi}{2\sigma^2} + \Sigma_k^{-1} \right)^{-1} \left( \frac{\Phi^T y}{2\sigma^2} + \Sigma_k^{-1} \mu_k \right)$$

- Which of these estimates do we select? The j-th one as per the following:

$$j = \arg\min_k \left\| y - \Phi \hat{x}_k \right\|^2 / (2\sigma^2) + (\hat{x}_k - \mu_k)^t \Sigma_k^{-1} (\hat{x}_k - \mu_k) + \log |\Sigma_k|$$

# GMMs for compressed sensing

- This is called the MAP step.
- In some variants, the K mixture components (means, covariance matrices and mixing values) are re-estimated from the signal reconstructions in the previous step.
- This step is called the ML step (maximum likelihood).
- These steps are repeated till convergence, i.e. till when the signal estimates do not change much.
- This is called the MAP-EM or the MAP-ML or the Max-Max algorithm.



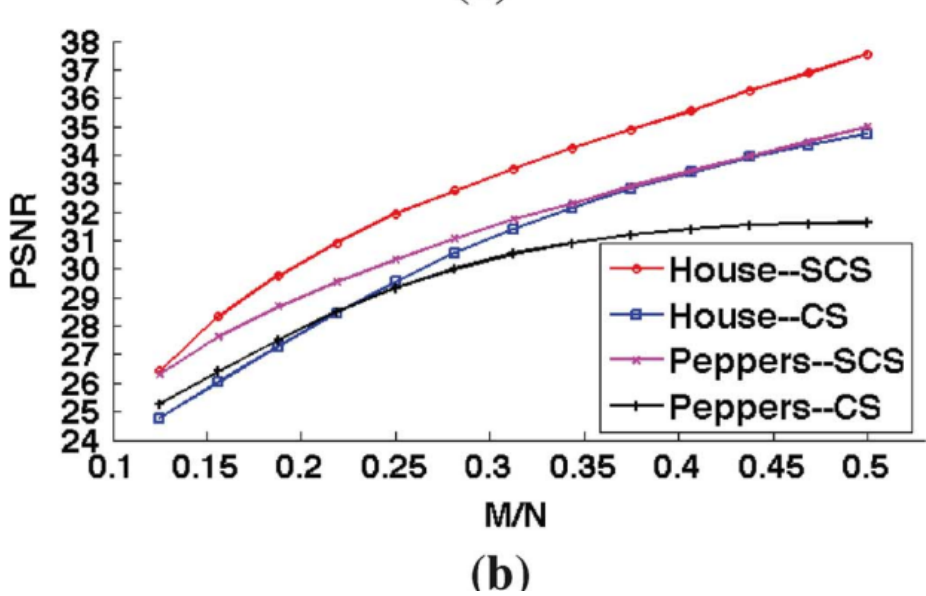




Fig. 5. (a) PSNR (dB) versus sampling rate for SCS and CS using Gaussian and random subsampling sensing matrices on image patches extracted from Lena. (b) PSNR (dB) versus sampling rate for SCS and CS using Gaussian sensing matrices on image patches extracted from House and Peppers.

Yu and Sapiro, Statistical Compressed Sensing of Gaussian Mixture Models, IEEE TSP 2011

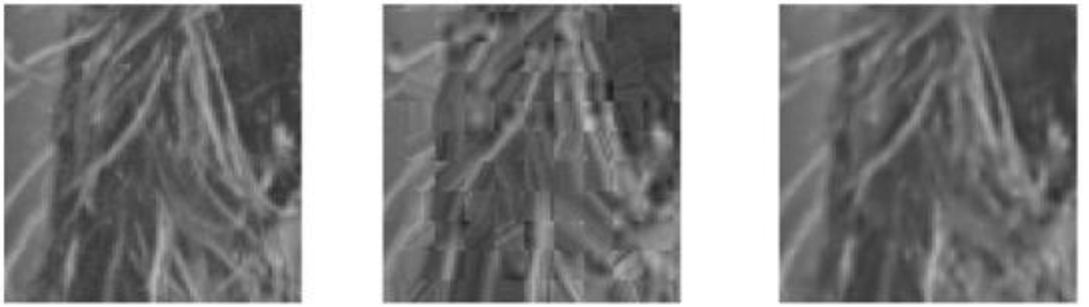




Fig. 7. From left to right. Zoomed crops from Lena, reconstructed images by SCS using Gaussian sensing matrices and nonoverlapping reconstruction, and by SCS using subsampling random matrices and overlapping reconstruction. The image is sensed on *nonoverlapped* patches at a sampling rate of $M/N = 0.25$. Local PSNRs are reported.

Yu and Sapiro, Statistical Compressed Sensing of Gaussian Mixture Models, IEEE TSP 2011

# Application in Scene Categorization

# Problem statement

- Consider a set of different scene categories.
- Given an image belonging to one of those categories, classify to which one it belongs.
- Next slide shows average images of different categories – with the main object (if any) chosen at a fixed scale and images translated such that the main object is in the center.

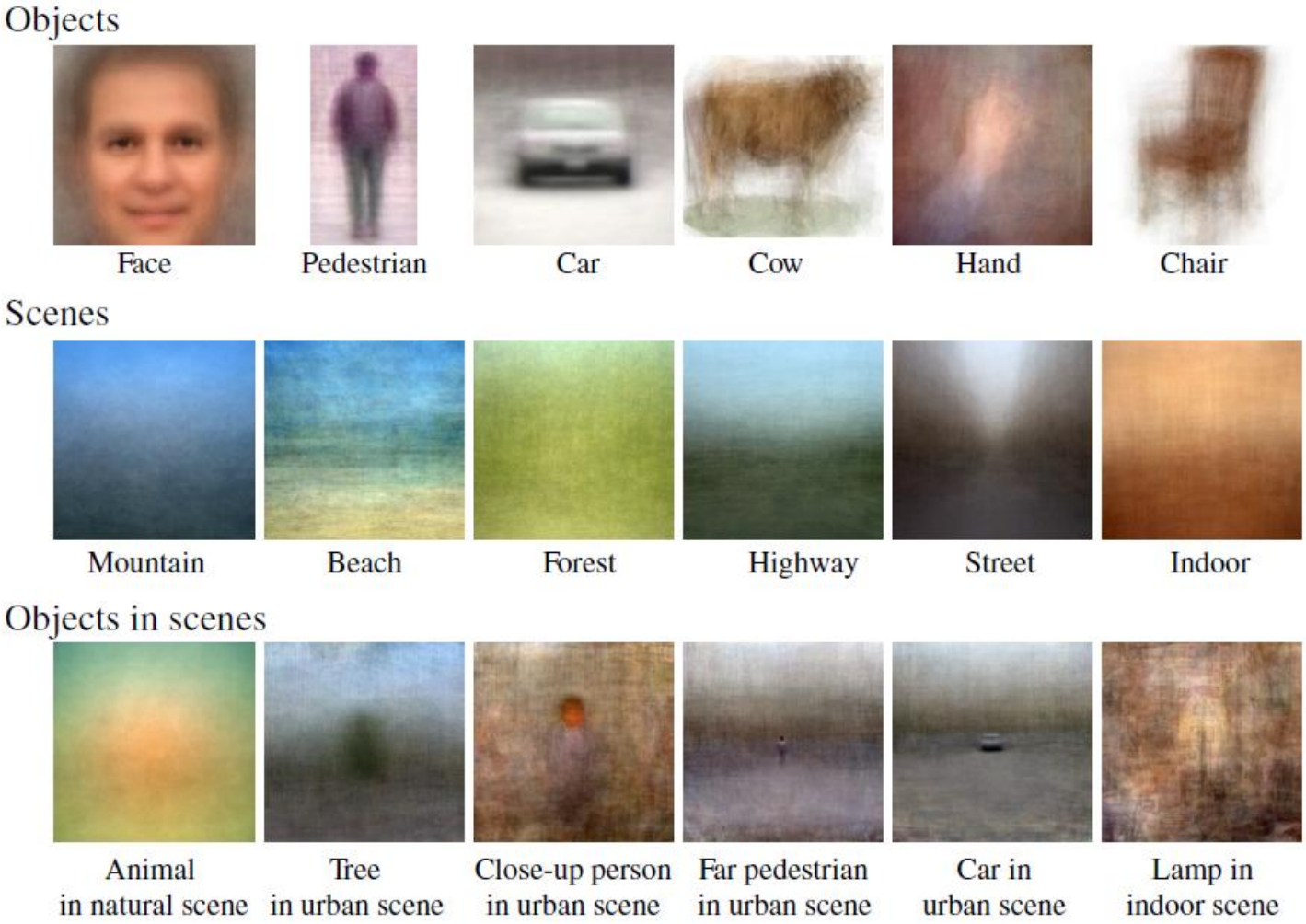



**Figure 1.** Averaged pictures of categories of objects, scenes and objects in scenes, computed with 100 exemplars or more per category. Exemplars were chosen to have the same basic level and viewpoint in regard to an observer. The group objects in scenes (third row) represent examples of the averaged peripheral information around an object centred in the image.

*Image source: Torralba and Olvia, Statistics of Natural Image Categories, Network, 2003*

# Man-made versus natural scenes

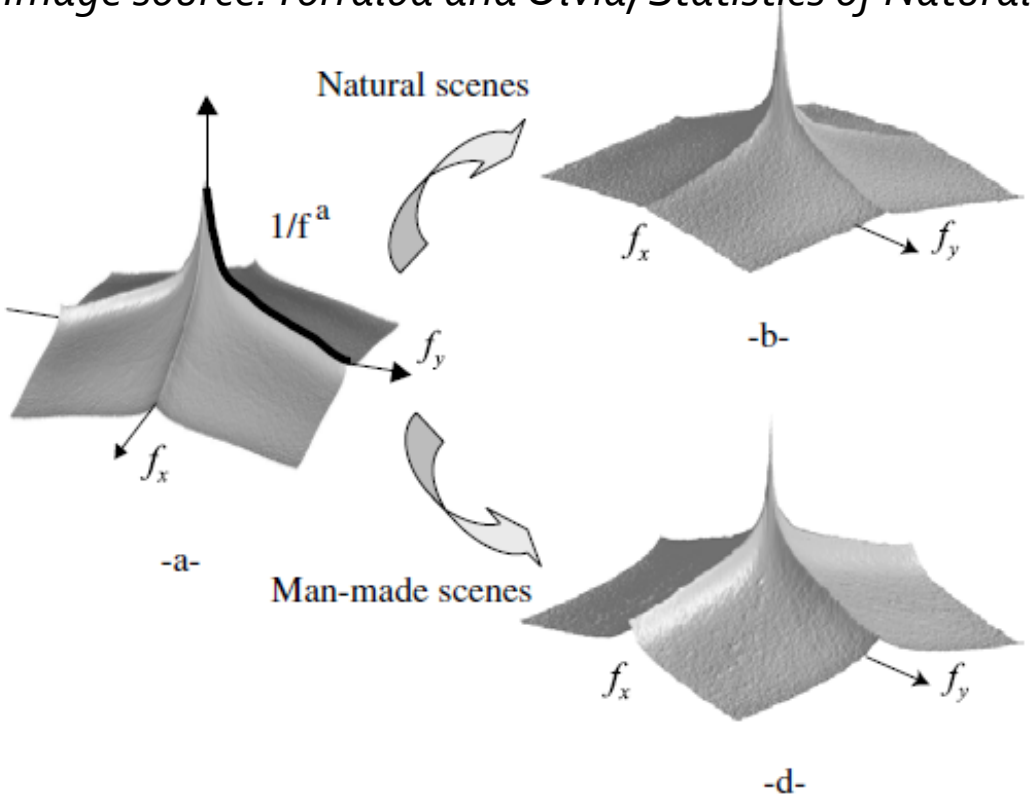
- We know the power law for an average image:

$$E(|S(u,v)|^2) = A|f|^{\alpha-2} \quad \text{where } f = \sqrt{u^2+v^2}$$

- But across image categories, the plots of the Fourier spectrum vary considerably even though they obey the power law.

- In man-made scenes, the horizontal and vertical edges are typically dominant – in natural scenes, they aren't.

*Image source: Torralba and Olvia, Statistics of Natural Image Categories, Network, 2003*



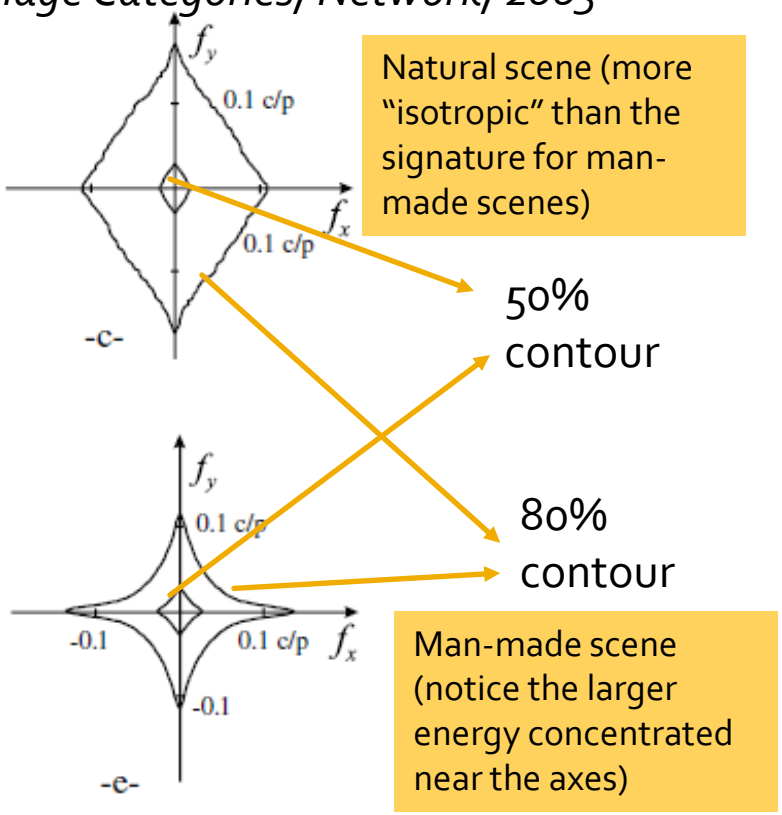




**Figure 2.** (a) Mean power spectrum averaged from 12 000 images (vertical axis is in logarithmic units). Mean po[b]pectra computed with 6000 pictures of man-made scene[d]nd 6000 pictures of natural scenes (c) and (e) are their respective spectral signatures. The contour plots represent 50 and 80% of the energy of the spectral signature. The contour is selected so that the sum of the components inside the section represents 50% (and 80%) of the total. Units are in cycles per pixel (cf also Baddeley 1996).

X% contour means that the energy inside the contour is X%. Energy refers to sum of the squares of the amplitudes of the Fourier components inside the contour. How is the contour computed? The Fourier coefficients are sorted in descending order of squared magnitude and the first set of coefficients amounting to X% of energy are collected together to create this contour.

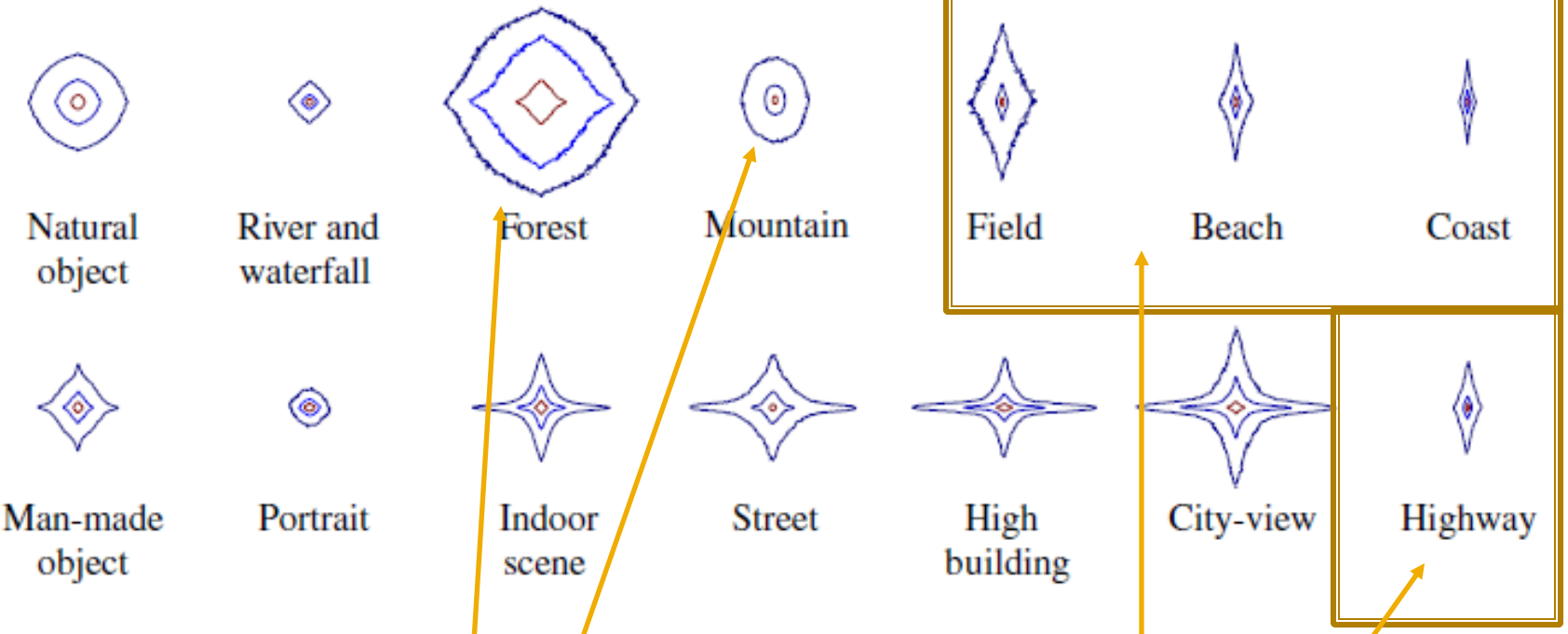



**Figure 3.** Spectral signatures of 14 different image categories. Each spectral signature is obtained by averaging the power spectra of a few hundred images per category. The contour plots represent 60, 80 and 90% of the energy of the spectral signatures (energy is obtained by adding the square of the Fourier components). The size of the spectral signature is correlated with the slope ($\alpha$). A large value of $\alpha$ produces a fast decay of the energy at high spatial frequencies, which produces a smaller contour. The overall shape is a function of both $\alpha(\theta)$ and $A(\theta)$.

More isotropic spectral signatures!

Domination by the horizon (strong horizontal edges)

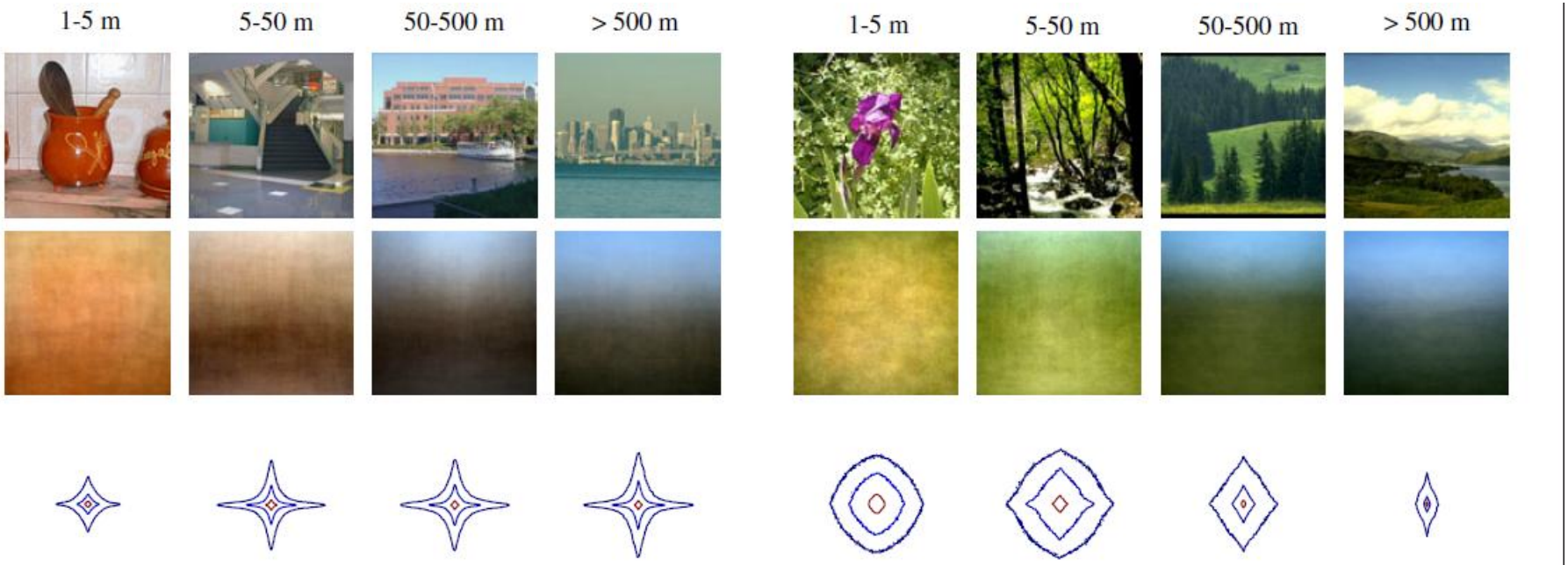



**Figure 4.** Averaged spatial images and spectral signatures as a function of scene scale. Scene scale refers to the mean distance between the observer and the principal elements that compose the scene. Each image average and spectral signature was calculated with 300–400 images.

Average spectral signatures for scenes at different scales.

Close-ups of man-made scenes are dominated by smooth surfaces, as opposed to close-ups of natural scenes. Smooth surfaces = dominance of low frequency information! Hence the spectra of man-made scenes have a dominance of low-frequency content – as is evident in the plots above.

*Image source: Torralba and Olvia, Statistics of Natural Image Categories, Network, 2003*

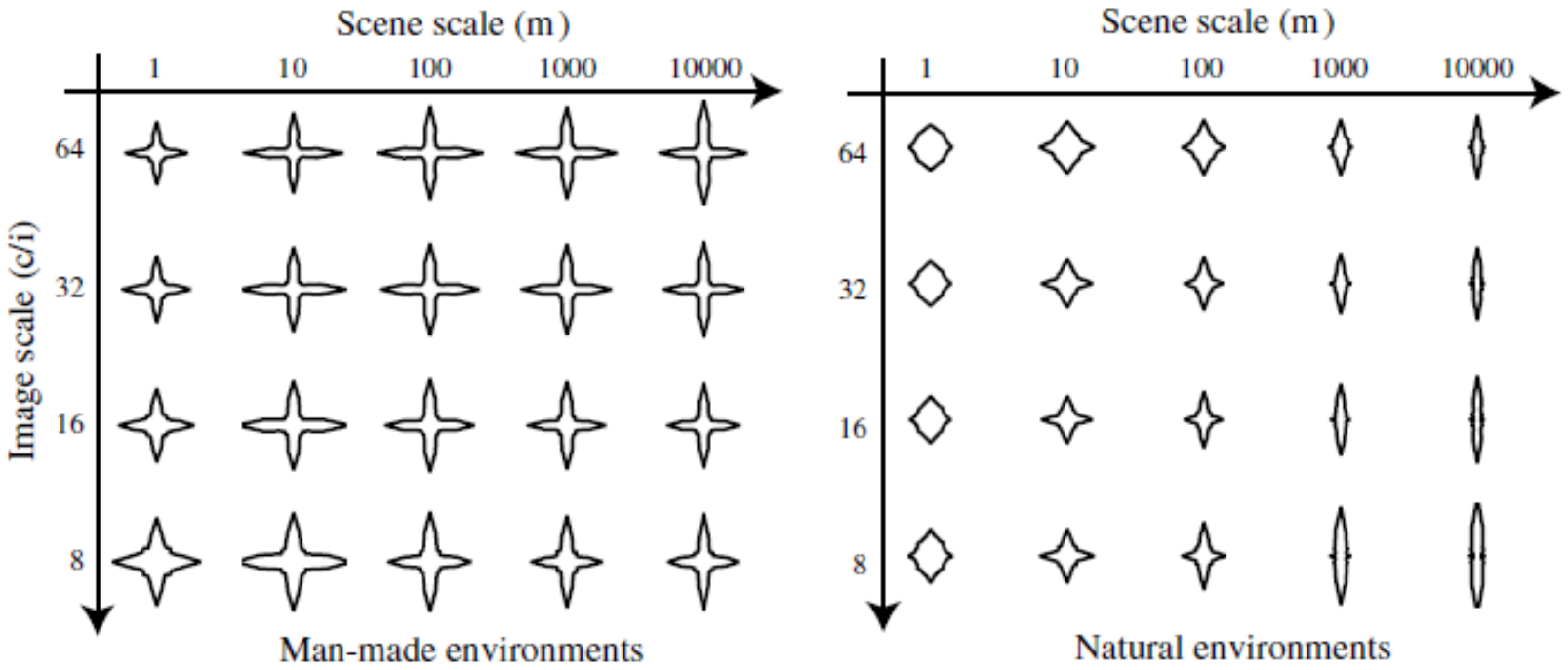



**Figure 5.** Polar plots of responses of multiscale oriented Gabor filters. The magnitude of each orientation corresponds to the total output energy averaged across the entire image. The energies are normalized across image scale by multiplying by a constant so that noise with $1/f$ amplitude spectrum has the same polar plots at all image scales.

**Scene scale:** average scene depth
**Image scale:** related to its resolution (different scales generated by upsampling the image by a factor of 2 successively)

*Image source: Torralba and Olvia, Statistics of Natural Image Categories, Network, 2003*

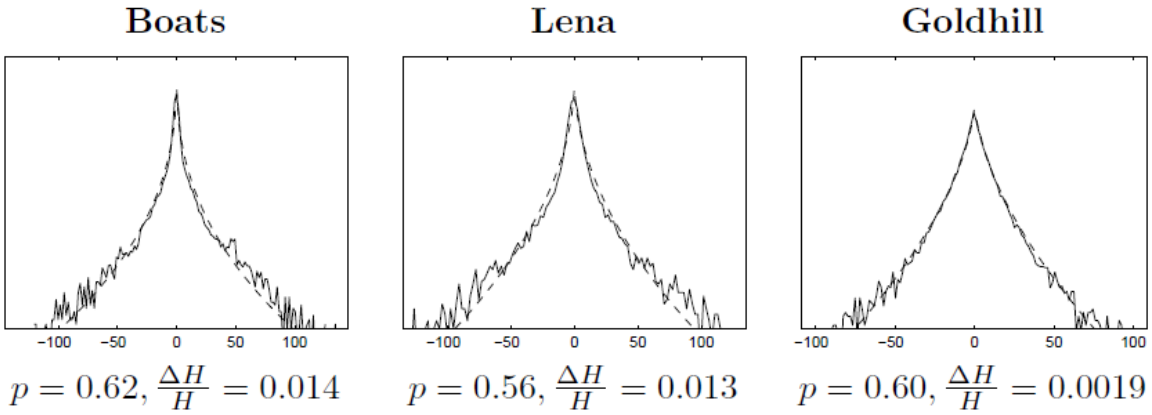
# Scene classification

- So how can we apply the observations about scene spectra to scene classification?
- We can apply PCA or LDA on such signatures.

# Application of Dependencies between wavelet coefficients: in denoising

# Recall 1

- Wavelet coefficients of images are Laplacian distributed!




- The various wavelet coefficients are not statistically independent.

Large wavelet coefficients tend to occur near each within the same sub-band.

And at the same relative spatial locations in sub-bands at adjacent scales or orientations

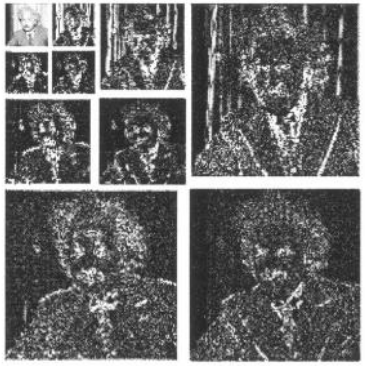


Fig. 3. Coefficient magnitudes of a wavelet decomposition. Shown are absolute values of subband coefficients at three scales, and three orientations of a separable wavelet decomposition of the Einstein image. Also shown is the lowpass residual subband (upper left). Note that high-magnitude coefficients of the subbands tend to be located in the same (relative) spatial positions.

# Wavelet coefficient dependency

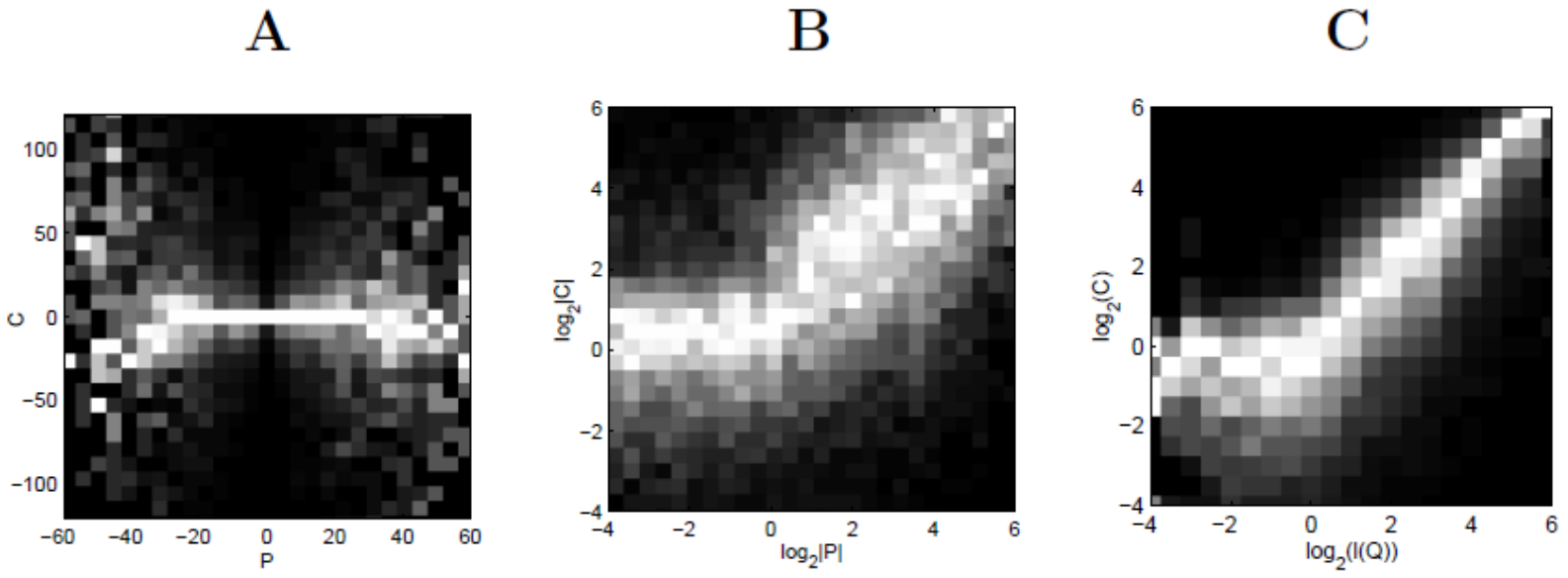



**FIGURE 6.** Conditional histograms for a fine scale horizontal coefficient. Brightness corresponds to probability, except that each column has been independently rescaled to fill the full range of display intensities. **A:** Conditioned on the parent (same location and orientation, coarser scale) coefficient. Data are for the "Boats" image. **B:** Same as **A**, but in the log domain. **C** Conditioned on a linear combination of neighboring coefficient magnitudes.

# Wavelet coefficient dependency

- The conditional density of the child wavelet coefficient ($c$) given the parent ($p$) (figure 6A,B two slides before) reveals:

1. $E(c|p) = 0$ for all values of $p$.
2. They are not independent statistically – because the variance of $c$ depends on the value of $p$.
3. The right side of the conditional density of the log of the squared coefficient is unimodal and concentrated on a unit slope line. In fact, $E(c^2|p^2)$ is proportional to $p^2$.
4. Left side shows $c$ being constant (not dependent on $p$).

- This pattern is also observed for siblings (adjacent spatial locations), cousins (same spatial location, adjacent orientations).
- This pattern is robust across a wide range of images.

# Wavelet coefficient dependency

- So how do we model this mathematically?
- Here is one statistical model:

$$\mathcal{P}(c \mid \vec{p}) = \mathcal{N}\left(0; \sum_k w_k p_k^2 + \alpha^2\right).$$

Neighbors of coefficient *c*
(some cousins & siblings, parent)

$$c^2 \approx \sum_k w_k p_k^2 + \alpha^2$$

Can be obtained by least squares method

$$c^2 \approx \sum_k w_k p_k^2 + \alpha^2$$

Can be obtained by least squares method

$$c_i^2 \approx \sum_k w_k p_{ik}^2 + \alpha^2, i = \text{index for coefficient } (1 \le i \le n)$$

$$\therefore \mathrm{c} \approx wP + \alpha^2, c \in R^{1\times n}, w \in R^{1\times K} \text{ (assuming K neighbors)}$$

$$P \in R^{K\times n}$$

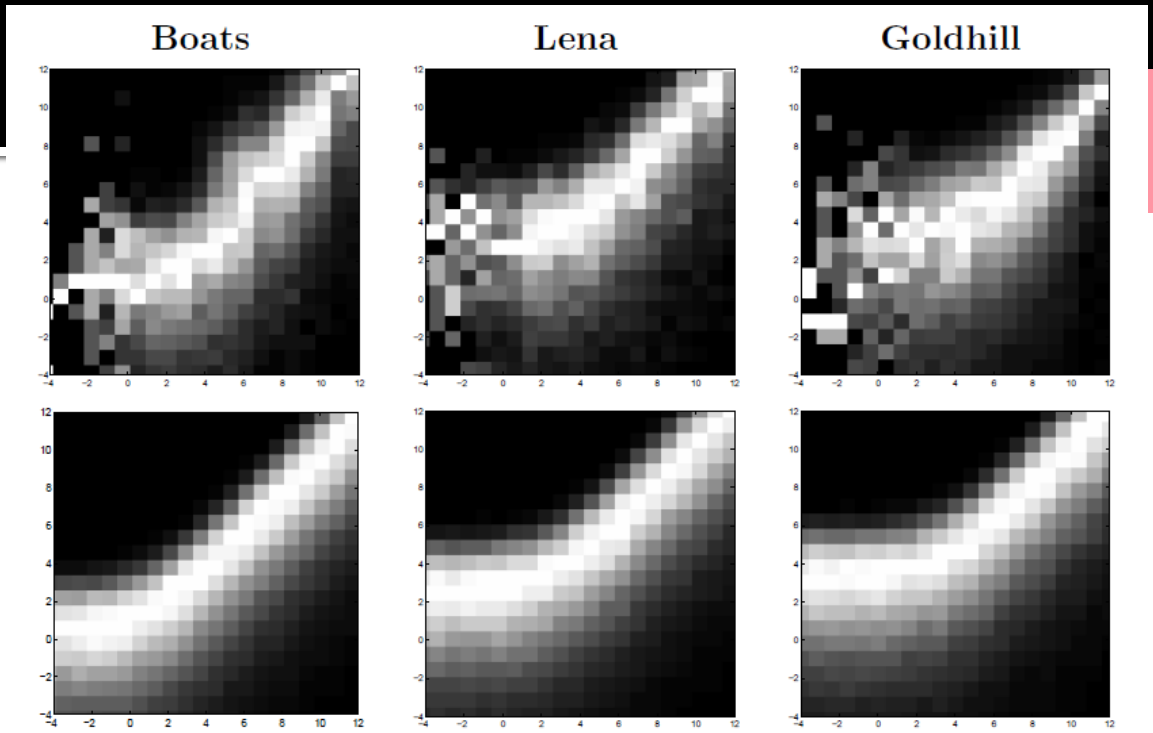
Bringing in $\alpha^2$, we have:

$$\mathrm{c} \approx (w\ \alpha^2)\begin{pmatrix} P \\ 1 \end{pmatrix} = \hat{w}\hat{P}, \hat{w} \in R^{1\times(K+1)}, \hat{P} \in R^{(K+1)\times n}$$

$$\therefore \hat{w} = c\hat{P}^T(\hat{P}\hat{P}^T)^{-1}$$

Least squares estimate of w

# Wavelet coefficient dependency




$$\mathcal{P}\left(c \mid \vec{p}\right) = \mathcal{N}\left(0; \sum_k w_k p_k^2 + \alpha^2\right).$$

**FIGURE 7. Top**: Examples of log-domain conditional histograms for the second-level horizontal subband of different images, conditioned on an optimal linear combination of coefficient magnitudes from adjacent spatial positions, orientations, and scales. **Bottom**: Model of equation (1.8) fitted to the conditional histograms in the left column. Intensity corresponds to probability, except that each column has been independently rescaled to fill the full range of intensities.

# Application to denoising

- We have already seen the formula:

$$\hat{\theta}_i = \frac{(U^T y)_i}{1+\hat{\lambda}} = \frac{\widehat{\sigma}^2}{\sigma^2 + \widehat{\sigma}^2}(U^T y)_i \qquad y = x + n = U\theta + n$$

- The same formula is applicable here with the following modification:

$$\widehat{\sigma}^2 = \sum_k w_k p_k^2 + \alpha^2$$

$$\therefore \hat{\theta}_i = \frac{\sum_k w_k p_k^2 + \alpha^2}{\sigma^2 + \sum_k w_k p_k^2 + \alpha^2}(U^T y)_i$$

# Application to denoising

- But this is a chicken and egg problem – because we do not know the values of $\{w_k\}$ or $\alpha$ beforehand!
- But we can estimate these values by minimizing:

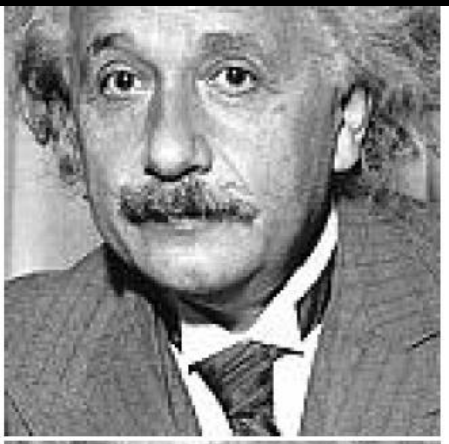
$$\{\hat{w}, \hat{\alpha}\} = \arg \min_{\{\vec{w}, \alpha\}} \mathbb{E} \left[ \hat{c}^2 - \sum_k w_k \hat{p}_k^2 - \alpha^2 \right]^2 .$$

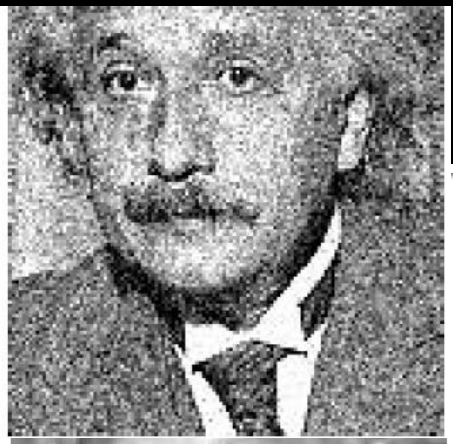
- The values of $\{p_k\}$ are obtained from a denoising algorithm that ignores wavelet coefficient dependency, e.g. using

$$\hat{\theta}_i = \frac{\hat{\sigma}^2}{\sigma^2 + \hat{\sigma}^2} (U^T y)_i$$

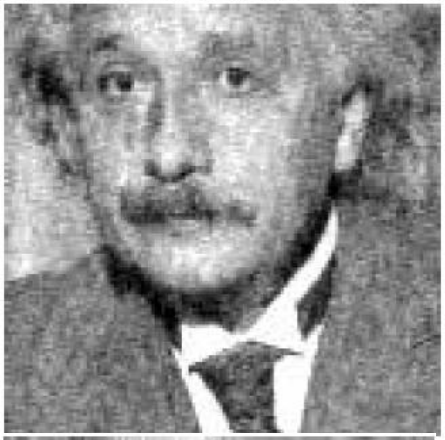
and then used for estimating $\{w_k\}$ or $\alpha$.

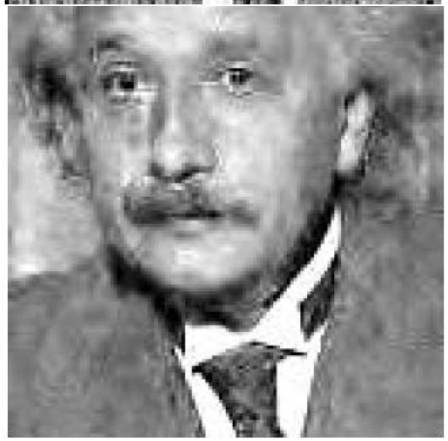
# Sample results



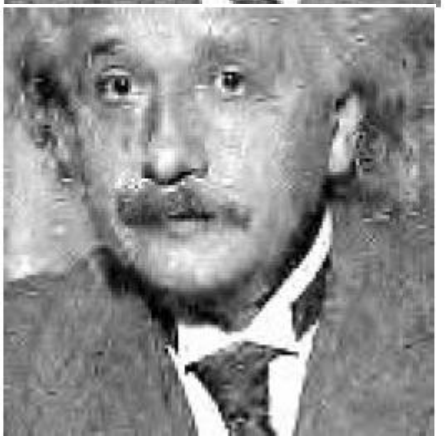


(Left) Original and (Right) noisy image





(Left) Marginal MAP with independent Gaussian prior and (Right) new model



MMSE estimator using GGD prior on wavelet coefficients

# User-assisted reflection removal

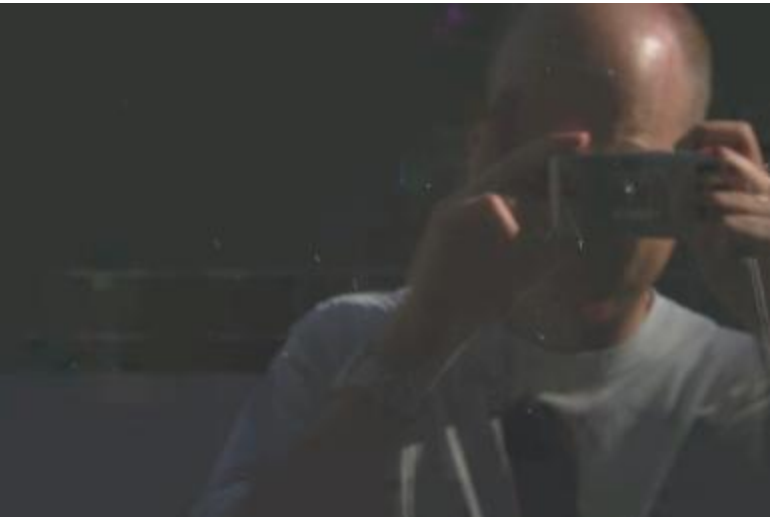
# Problem statement

- A picture of the scene outside taken through a glass window contains undesirable reflections – of the scene on the side of the photographer.
- The reflection can be weakened using a polarizer in front of the camera lens but that is unavailable in all cameras and it is unwieldy to use.

$$I(x,y) = I_1(x,y) + I_2(x,y)$$

To estimate $I_1, I_2$ given $I$

This is an ill-posed problem, as there are infinitely many $I_1$ and $I_2$ that could sum up to I.



# Method for reflection removal: user-assisted

- Allow a user to mark a set of points $S_1$ which the user thinks belong to $I_1$.
- And another set of points $S_2$ which the user thinks belong to $I_2$.
- Intuitively, these points will belong to strong edges in the two images.

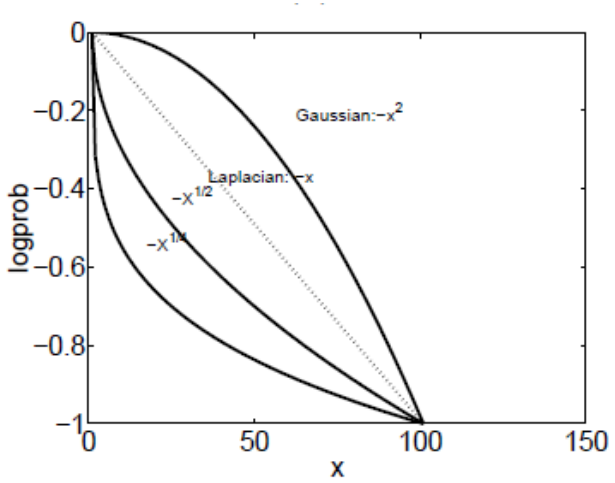
# Method for reflection removal: user-assisted

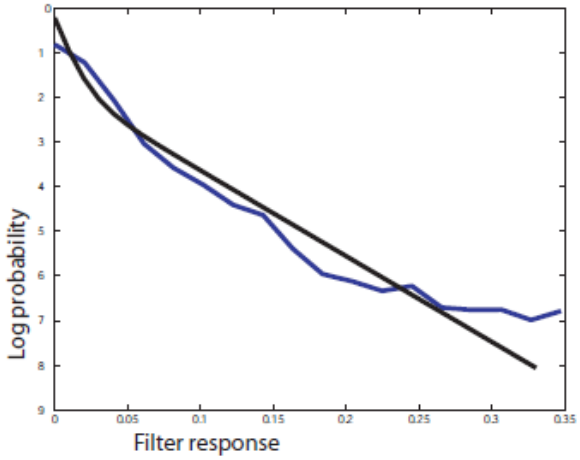
- Find $I_1$ and $I_2$ such that:

1. $I_1$ and $I_2$ sum up to I
2. The gradient of $I_1$ at points in $S_1$ should match the gradient of I at those points.
3. The gradient of $I_2$ at points in $S_2$ should match the gradient of I at those points.

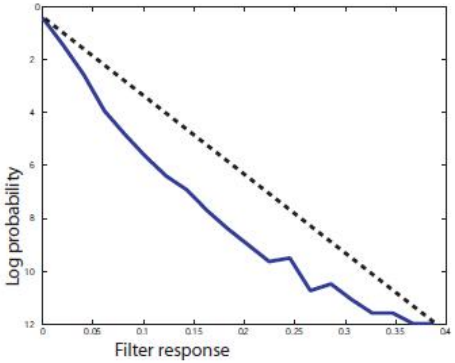
# Method for reflection removal: statistical model

- Exploit a statistical property of a natural image.
- The gradients are sparse!



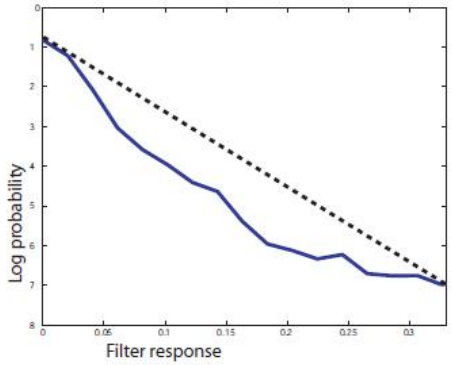



(a)

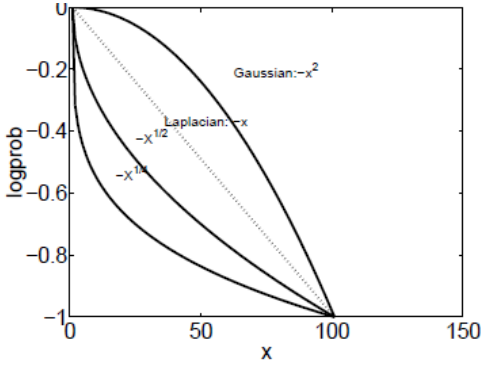



(b)

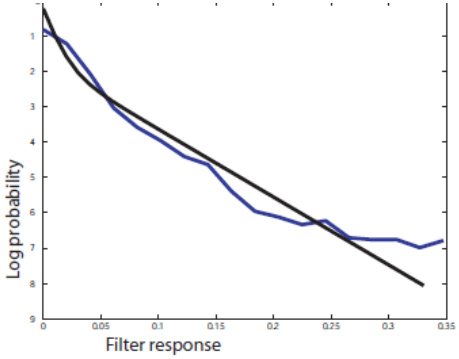
(c)




(d)




(e)




(f)

The gradients (in this case, the y derivatives of the intensity) are sparse!

# Method for reflection removal: statistical model

- One possible statistical model for the gradients of the image is the following:

$$p(z) \propto \frac{1}{s_1} e^{-(|z|/s_1)^q}$$

$z$ = gradient value,
$q$ = shape parameter

- But the gradients can be computed in $x$ and $y$ directions, and we can have first as well as second order derivatives.
- Each choice of derivative will be called as a gradient filter – of which we will have many.

# Method for reflection removal: statistical model

- Note that each gradient filter is applied to every pixel location of image I.
- The output of filter *k* at pixel location *i* on image $I_1$ will be denoted as $f_{ik,1}$.
- We will assume statistical independence of all these filter outputs.
- So the overall model is:

$$\prod_{i,k} p(f_{ik,1}), \text{where } f_{ik,1} = (f_k * I_1)_i$$

# Method for reflection removal: statistical model

- Recall that we have $I = I_1 + I_2$ where $I_1$ and $I_2$ are both unknown.
- Gradients of both images are sparse and independent of each other.
- So we have to maximize wr.t. $I_1$ and $I_2$ :

$$\prod_{i,k} p(f_{ik,1}) p(f_{ik,2})$$

- Equivalent to minimizing w.r.t. $I_1$ and $I_2$ :

$$\sum_{i,k} -\log(p(f_{ik,1})) - \log(p(f_{ik,2})) = \sum_{i,k} \rho(f_{ik,1}) + \rho(f_{ik,2})$$

$$\rho(x) = (|x|/s_1)^q = -\log p(x)$$

# Method for reflection removal: statistical model

- But we also have an important constraint that $I_1$ and $I_2$ sum up to I.
- So the objective function becomes:

$$E(I_1) = \sum_{i,k} \rho(f_{ik,1}) + \rho\left((f_k * (I - I_1))_i\right)$$

- But that's not enough. We need to enforce that the gradients of the given image I and the estimated $I_1$ agree at all points in set $S_1$.
- Likewise for $I_2$ and $S_2$.

# Method for reflection removal: statistical model

- So the objective function becomes:

$$E(I_1) = \sum_{i,k} \rho\left((f_k * I_1)_i\right) + \rho\left((f_k * (I - I_1))_i\right) +$$

$$\lambda \sum_{i \in S_1, k} \rho\left(\left(f_k * I\right)_i - \left(f_k * I_1\right)_i\right) + \lambda \sum_{i \in S_2, k} \rho\left(\left(f_k * I\right)_i - \left(f_k * I_2\right)_i\right)$$

$$= \sum_{i,k} \rho\left((f_k * I_1)_i\right) + \rho\left((f_k * (I - I_1))_i\right) +$$

$$\lambda \sum_{i \in S_1, k} \rho\left(\left(f_k * I\right)_i - \left(f_k * I_1\right)_i\right) + \lambda \sum_{i \in S_2, k} \rho\left(f_k * I_1\right)_i$$

Because $I_2 = I - I_1$.

# Optimization algorithm

- Given the statistical model for the gradient filter outputs, the function $\rho$ is non-convex if $q < 1$ (it is convex if $q = 1$).
- The optimization procedure for this is not very easy.
- The authors use a method called **iteratively reweighted least squares (IRLS)**.

# Segway: Least squares method

- Consider the solution to the following problem:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^t (\mathbf{y} - \mathbf{X}\boldsymbol{\beta});$$

$$\mathbf{y} \in R^n, \mathbf{X} \in R^{n \times m}, \boldsymbol{\beta} \in R^m, m \leq n$$

$$\rightarrow \boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- This is a least squares problem, and it has a well-known pseudo-inverse based solution.

- Now we will look at some flavours of least squares.

# Segway: Weighted least squares method

- Now consider the solution to the following problem:

$$\arg\min_{\boldsymbol{\beta}} \left\| \mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\|^2 = \arg\min_{\beta} (\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}))^t (\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}))$$

$$\rightarrow \boldsymbol{\beta} = (\mathbf{X}^T \mathbf{W}^2 \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^2 \mathbf{y}$$

- Here **W** is a *n* x *n* diagonal matrix containing weights which give different levels of importance to each entry of **y**.
- The solution of this is again in terms of a pseudo-inverse.

# Segway: Least *p*-norm problem

- Consider the solution to the following problem:

$$\arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^p = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} | y_i - \mathbf{X_i}\boldsymbol{\beta} |^p, p \leq 1$$

$$\mathbf{y} \in R^n, \mathbf{X} \in R^{n \times m}, \boldsymbol{\beta} \in R^m, m \leq n$$

- This has no known closed form solution!

- Instead an iterative procedure has been proposed – called IRLS.

# Segway: IRLS

- The IRLS at step $t$+1 involves a weighted least

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} | w_i^{(t)} (y_i - \mathbf{X_i}\boldsymbol{\beta}) |^2 = (\mathbf{X}\mathbf{W}(t)^2\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}(t)^2\mathbf{y}$$

Weight for point $i$ at iteration $t$

Diagonal matrix of weights at iteration $t$ (for all points)

- At t = 1, the weights are set to 1.
- The weights are updated as follows:

$$w_i^{(t+1)} = | y_i - \mathbf{X_i}\boldsymbol{\beta}^{(t+1)} |^{(p-2)/2}, \mathbf{W}(t+1) = diag(\{w_i^{(t+1)}\})$$

- This is done till convergence.

# Segway: IRLS

- The weights are updated as follows:

$$w_i^{(t+1)} = | y_i - \mathbf{X_i}\boldsymbol{\beta}^{(t+1)} |^{(p-2)/2}$$

- Why these weights? Simply because the problem can be re-written as follows:

$$\arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^p = \arg\min_{\beta} \sum_{i=1}^{n} | y_i - \mathbf{X}_i\boldsymbol{\beta} |^{p-2} (y_i - \mathbf{X}_i\boldsymbol{\beta})^2, p \leq 1$$

$$\mathbf{y} \in R^n, \mathbf{X} \in R^{n \times m}, \boldsymbol{\beta} \in R^m, m \leq n$$

# Back to the Optimization algorithm

- The objective function is:

$$E(I_1) = \sum_{i,k} \rho((f_k * I_1)_i) + \rho((f_k * (I - I_1))_i) +$$

$$\lambda \sum_{i \in S_1, k} \rho((f_k * I - f_k * I_1)_i) + \lambda \sum_{i \in S_2, k} \rho((f_k * I_1)_i)$$

- It can be expressed as:

$$E(v) = \sum_{j=1}^{4} \sum_{k} \rho(\mathbf{A_{j,k}} \mathbf{v} - \mathbf{b_{j,k}}), \mathbf{v} = \mathrm{vec}(I_1)$$

$$\mathbf{A_{j,k}} = \text{matrix corresponding to filter } f_k$$

# Optimization algorithm

1. Set weights $\mathbf{w_j^{(0)}} = 1$

2. $t = 1$

3. Repeat till convergence:

(a) $\overline{\mathbf{A}} \leftarrow \sum_j \mathbf{A_j^T W^{(t-1)} A_j}, \overline{\mathbf{b}} \leftarrow \sum_j \mathbf{A_j^T} W^{(t-1)} \mathbf{b_j}, \mathbf{x}^{(t)} \leftarrow$ solution of $\overline{\mathbf{A}}\mathbf{x} = \overline{\mathbf{b}}$

(b) $\mathbf{W}^{(t)} \leftarrow diag(\mathbf{w_j^{(t)}}), \mathbf{w_j^{(t)}} \leftarrow \frac{1}{\mathbf{u_j}} \frac{d\rho}{d\mathbf{u_j}}$ where $\mathbf{u_j} \leftarrow \overline{\mathbf{A}}\mathbf{x}^{(t)} - \overline{\mathbf{b}}$,

$$\rho(\mathbf{u_j}) = \log\left( \sum_{l=1}^{2} \frac{\pi_l}{2s_l} e^{-|u_j|/s_l} \right)$$

(c) $t = t + 1$

# Method for reflection removal: actual statistical model used in the paper

- The statistical model for the gradients of the image is chosen to be the following:

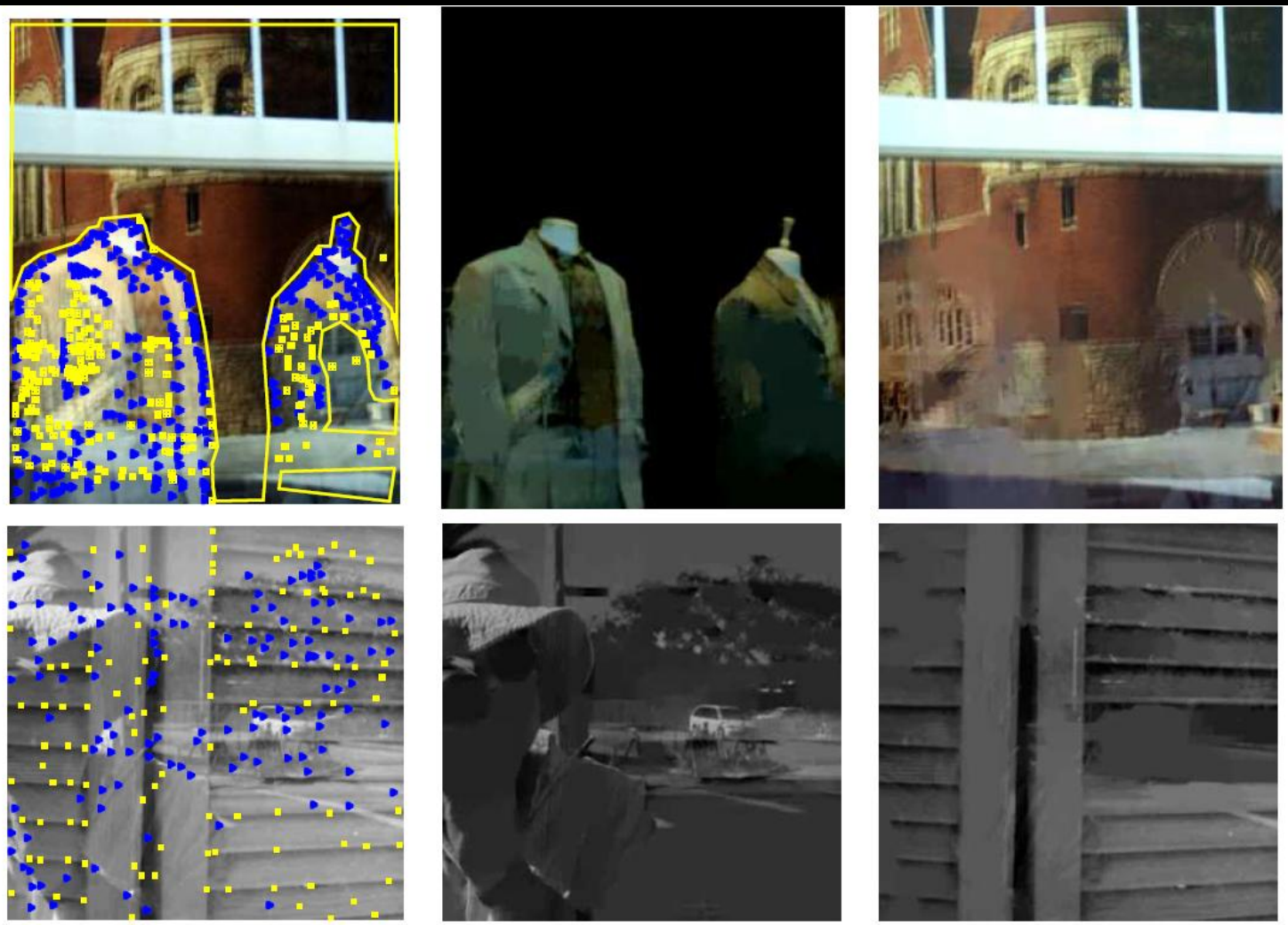
$$p(z) = \frac{\pi_1}{2s_1} e^{-|z|/s_1} + \frac{\pi_2}{2s_2} e^{-|z|/s_2}$$

$$\rho(z) = -\log p(z)$$

z = gradient value

- This is a mixture of two Laplacian distributions and it is seen to be sparser than a single Laplacian.

# Sample results



Input | Output layer 1 | Output layer 2

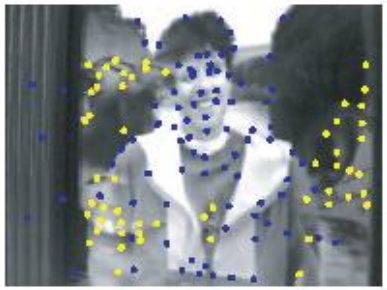
Laplacian prior

Laplacian prior

Sparse prior

Sparse prior

Comparison: Laplacian and Sparse (mixture of two Laplacians) priors



Laplacian prior

Laplacian prior

Sparse prior

Sparse prior

Comparison: Laplacian and Sparse (mixture of two Laplacians) priors

Input Laplacian prior Gaussian prior

Fig. 5. A very simple image with two labeled points. The Laplacian prior gives the correct decomposition for this image while the Gaussian prior prefers to split edges into two low contrast edges.

(a) (b)

Fig. 6. Gaussian prior results using the labels in the second column of fig4.

Comparison: Laplacian and Gaussian priors. Notice the much better results For the Laplacian prior as compared to the Gaussian prior.

# Summary

- Motivation for studying statistics of natural images – applications and Bayesian framework
- Statistics: power law, marginal and joint distributions of wavelet coefficients
- Applications: denoising, deblurring, scene categorization