Image Data Compression

Natural image statistics

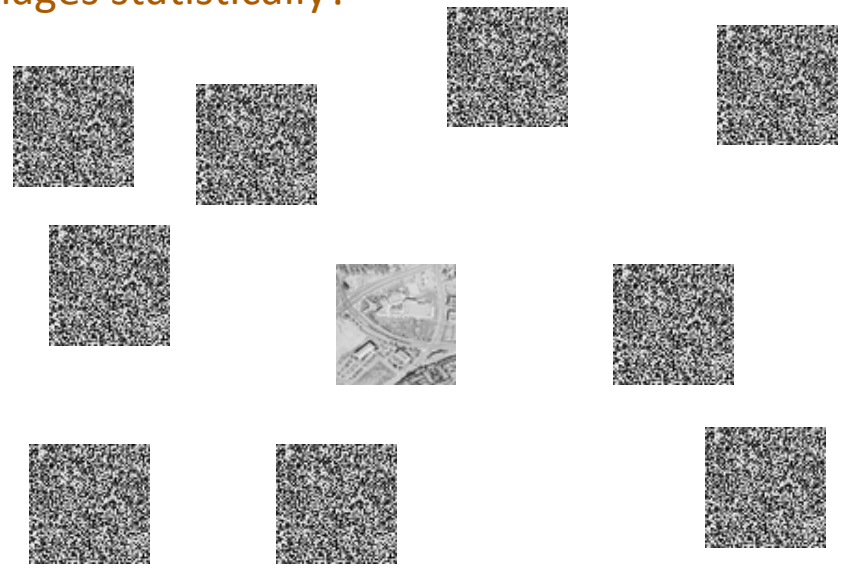
What is natural image statistics?

- Need good data models to have better compression methods!
- Relatively recent field (in 2009 first comprehensive book on subject)
- Closely related to human visual system studies
  (conjecture: biological visual systems are “optimal” for their natural images)
- Facilitated by big image databases and available processing power
- Are there really any challenges in studying images statistically?

Following book:
[Hyvärinen, Hurri, Hoyer Natural Image Statistics, Springer 2009]

A few [relatively] big numbers:

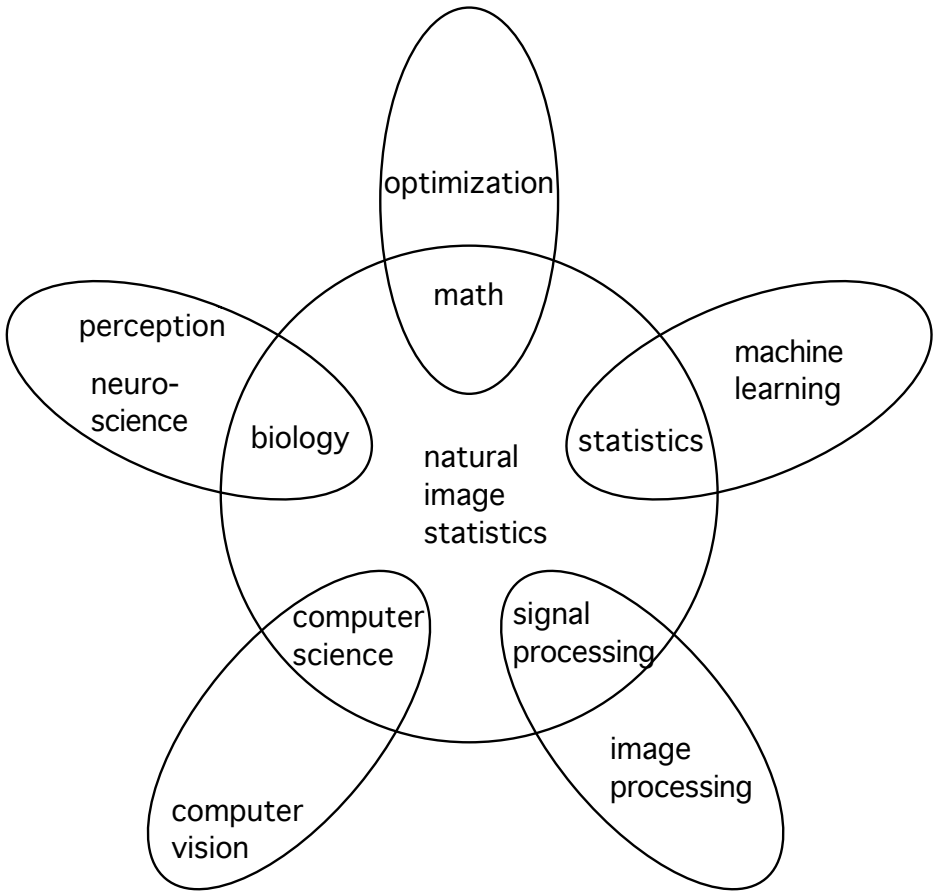
- Seconds since Big Bang: 10^{17}
- Atoms in the Universe: 10^{80}
- Entropy (white noise images):
  H = 65 x 65 x 8 = 33800 bits
  # of 65 x 65 8-bit gray-scale images:
  $2^H \approx 10^{10000}$
- Natural scenes are redundant, estimated entropy ~ 0.4 H,
- => one out of 10^{6000} white noise images has basic statistics of natural scenes



“The distribution of natural images is complicated. Perhaps it is something like beer foam, which is mostly empty but contains a thin mesh-work of fluid which fills the space and occupies almost no volume. The fluid region represents those images which are natural in character.”

[Ruderman 1996]

Very multi-faceted research area



Models of natural image data

- Physical (generative) imaging models
- Non-linear manifold of natural images
- Non-parametric sample-based models
- Biologically-inspired neural networks
- Simple models reproducing increasingly more complex statistics of natural images (unsupervised learning!)

Model: transform to "convenient" space

Some criteria of "good" models:

- "simplicity" ≈ reduced dimensionality
- "sparseness" ≈ compact representation
- "metabolic efficiency" ≈ efficient wrt processing energy in brain
- "learning efficiency" ≈ easy to learn from unique images (seen only once)
- "wiring length" ≈ minimal communication between neurons
- …

Reminder of multivariate statistics

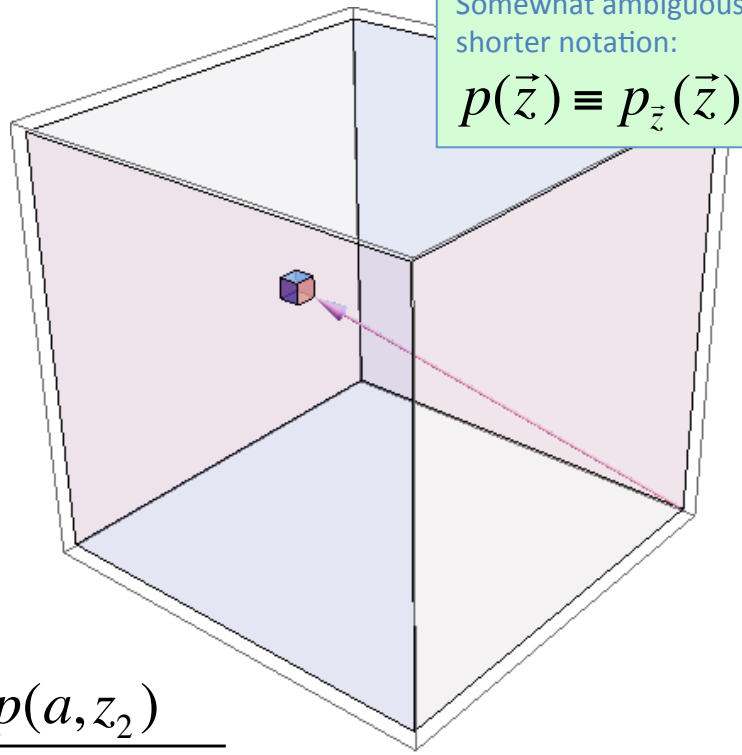
- Probability density function (PDF):

$$p_{\vec{z}}(\vec{a}) = \lim_{\Delta z \to 0} \frac{P(z_i \in [a_i, a_i + \Delta z] \forall i = 1, \ldots, n)}{\Delta z^n}$$

$$\int p_{\vec{z}}(\vec{a})\, d\vec{a} = 1$$

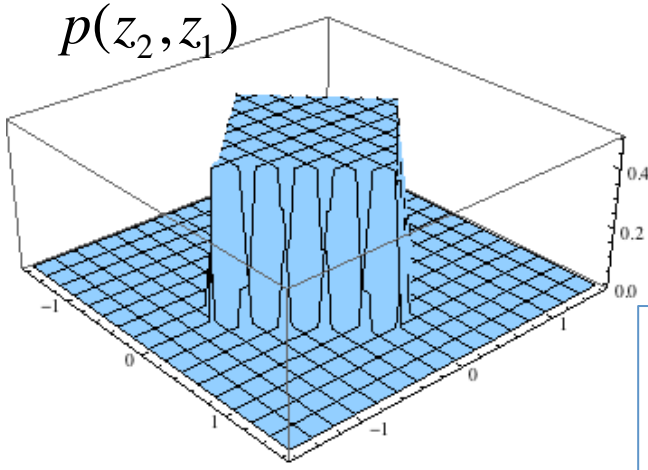
Somewhat ambiguous shorter notation:

$$p(\vec{z}) \equiv p_{\vec{z}}(\vec{z})$$

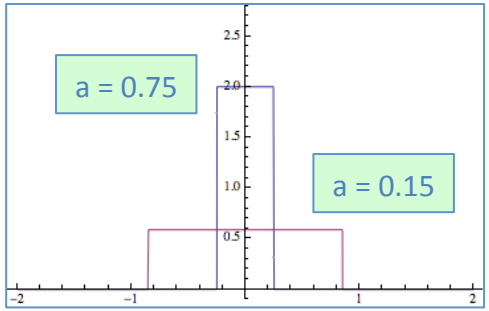


- Conditional and marginal probability:

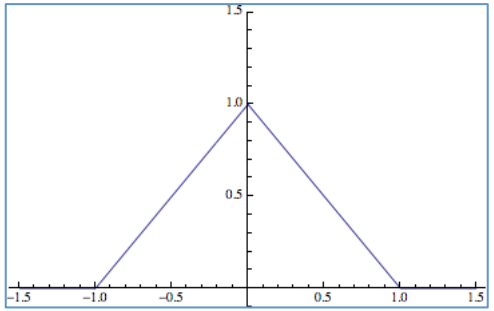
$p(z_2, z_1)$



$$p(z_2 \mid z_1 = a) = \frac{p(a, z_2)}{\int p(a, z_2)\, dz_2}$$



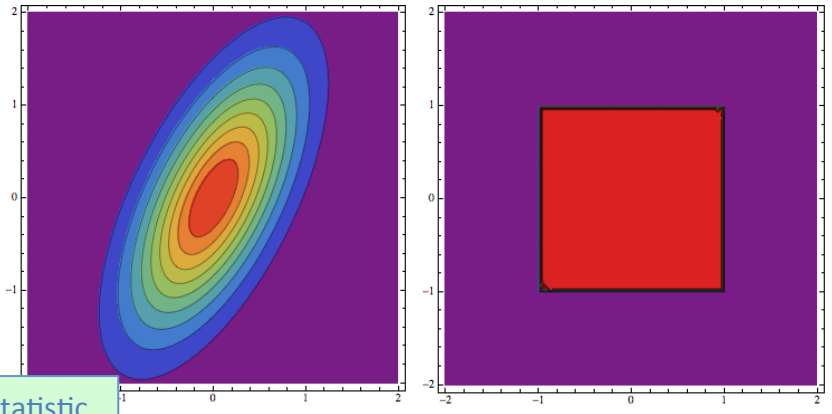
$$p(z_1) = \int p(z_1, z_2)\, dz_2$$



Reminder of multivariate statistics (continued)

- The variables z_1 and z_2 are independent iff their joint PDF factorizes:

$$p(z_1, z_2) = p(z_1)p(z_2)$$



- Expectation of random vector / function of RV:

$$E\{\vec{z}\} \equiv \langle \vec{z} \rangle = \int p_{\vec{z}}(\vec{a})\, d\vec{a}$$

$$E\{g(\vec{z})\} = \int g(\vec{a}) p_{\vec{z}}(\vec{a})\, d\vec{a}$$

1-st order statistic

Independent variables are uncorrelated:

$$E\{g_1(\vec{z}_1)g_2(\vec{z}_2)\} = E\{g_1(\vec{z}_1)\}E\{g_2(\vec{z}_2)\}$$

However, uncorrelated variables may still be dependent!

- Variance and covariance (1D):

$$\mathrm{var}\{z\} = E\{(z - \langle z \rangle)^2\}$$

$$\mathrm{cov}\{z_1, z_2\} = E\{z_1 z_2\} - \langle z_1 \rangle \langle z_2 \rangle$$

2-nd order statistics

- Covariance matrix (nD):

$$C(\vec{z}) = \begin{pmatrix} \mathrm{cov}\{z_1, z_1\} & \mathrm{cov}\{z_1, z_2\} & \dots & \mathrm{cov}\{z_1, z_n\} \\ \mathrm{cov}\{z_2, z_1\} & \mathrm{cov}\{z_2, z_2\} & \dots & \dots \\ \dots & & \dots & \dots \\ \mathrm{cov}\{z_n, z_1\} & \dots & \dots & \mathrm{cov}\{z_n, z_n\} \end{pmatrix} = E\{\vec{z}\vec{z}^T\} - \langle \vec{z} \rangle \langle \vec{z} \rangle^T$$

Reminder of multivariate statistics (continued further)

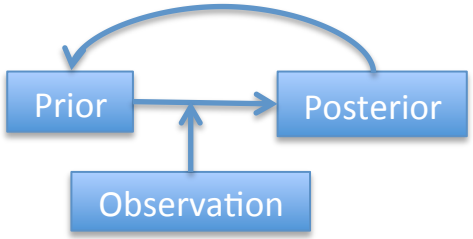
$$p_{\vec{s}}(\vec{s} \mid \vec{z}) = \frac{p_{\vec{z}}(\vec{z} \mid \vec{s}) p_{\vec{s}}(\vec{s})}{\int p_{\vec{z}}(\vec{z} \mid \vec{s}) p_{\vec{s}}(\vec{s}) ds}$$

- unobserved parameters
- observation
- likelihood
- a priori probability (prior)
- a posteriori probability (posterior)
- normalization constant (independent of s)

Non-informative prior:
e.g. uniform distribution,

$$p_{\vec{s}}(\vec{s}) = c$$

Incremental learning



Parameter estimation (log space)

Constant, independent of α

- BR: $\log p_{\alpha}(\alpha \mid z_1, z_2, \ldots, z_m) = \log p(z_1, z_2, \ldots, z_m \mid \alpha) + \log p(\alpha) - \log p(z_1, z_2, \ldots, z_m)$
- Maximum a posteriori probability (MAP): $\alpha^*_{MAP} = \underset{\alpha}{\operatorname{argmax}} \left[\log p(z_1, z_2, \ldots, z_m \mid \alpha) + \log p(\alpha) \right]$
- Maximum likelihood: $\alpha^*_{ML} = \underset{\alpha}{\operatorname{argmax}} \left[\log p(z_1, z_2, \ldots, z_m \mid \alpha) \right]$

Equivalent to MAP with constant prior

Example: 1D Gaussian

$$p_z(z \mid \alpha) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(z-\alpha)^2}{2} \right]$$

- Observe: $z_1, z_2, \ldots, z_m$, iid: $p_z(z_1, z_2, \ldots, z_m \mid \alpha) = p(z_1 \mid \alpha) \cdot \ldots \cdot p(z_m \mid \alpha)$
- Task: estimate α (assume flat prior)

Likelihood:

$$\log p(z_1, \ldots, z_m \mid \alpha) = -\frac{1}{2} \sum (z_i - \alpha)^2 + const$$

Solution with least squares method:

$$\alpha^*_{ML} = \frac{1}{m} \sum z_i$$

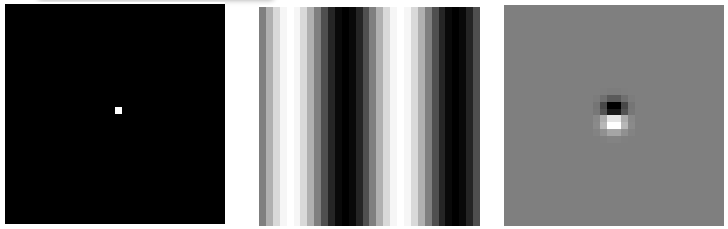
Statistics of linear features

- One patch (e.g. 32 x 32 pixels) - random vector $\vec{z}$ (n = 1024 components)
- Patch at random location in random image from DB = one observation
- Apply discrete linear transform: $s_i = \vec{w}_i^T \vec{z}$

"Vectorize" image; gray-scale values

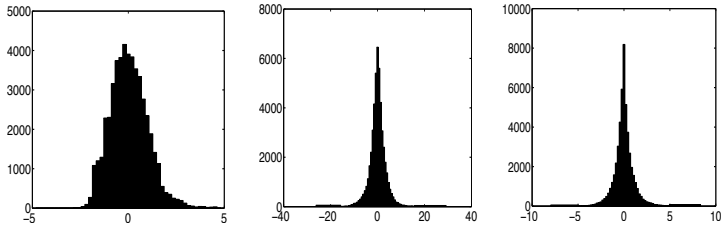
Basis functions, "features", can also be interpreted as images

- Examples of features:



- Dirac filter (pixel value)
- Grating detector (cosine)
- Gabor edge detector

- Histograms of filter outputs:



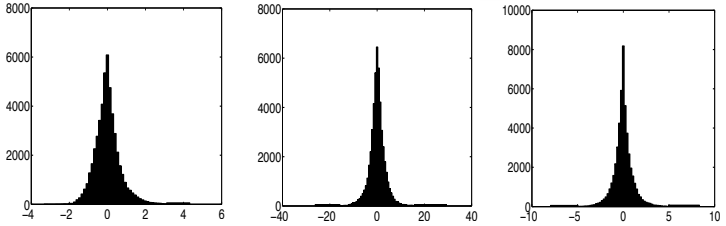
Two last histograms very non-Gaussian!

- Simplest image statistic: mean (DC component)
  (not to be confused with mean over sample!)

$$\hat{\vec{z}} = \vec{z} - \frac{1}{n}\sum_{i=1}^{n} z_i$$

The only fist-order structure

- Mean-subtracted histograms:



Grating and edge detectors orthogonal to mean value, Dirac filter - not

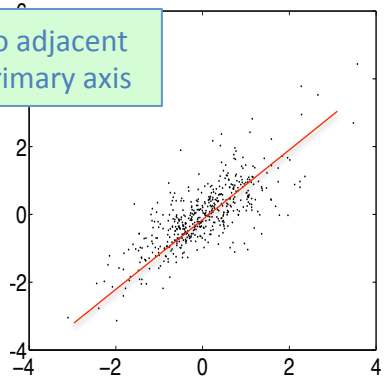
In what follows, assume subtracted mean

General strategy: remove trivial parts that are well-understood, study what remains

Principal component analysis

- KLT applied to whole patches
- Second-order structure (pixel-to-pixel correlations)
- Principal components “explain” as much variance in data as possible
- Technical requirement: mean-free images, limit norm of w to e. g. 1

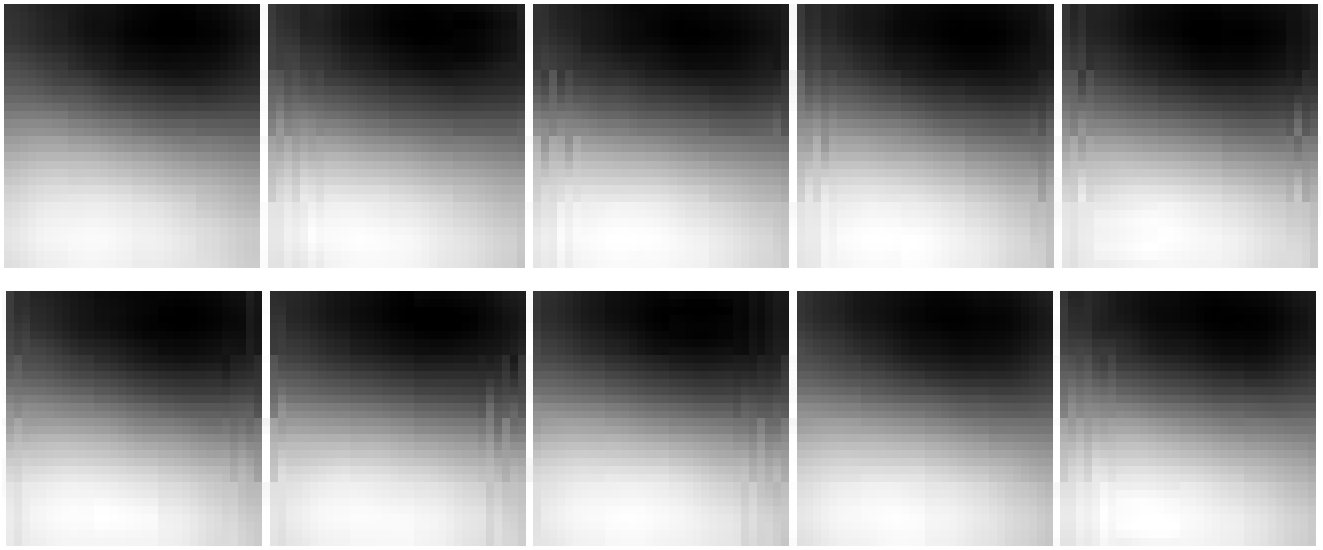
Scatter plot of two adjacent pixels and their primary axis



Learning first principal component: max output variance in given sample,

$$\vec{w}^{*} = \underset{\vec{w},\|\vec{w}\|_2=1}{\operatorname{argmax}} \left[\frac{1}{T} \sum_{t=1}^{T} (\vec{w}^{T} \vec{z})^{2} \right]$$

First principal component computed for windows of 32x32 pixels for 10 different randomly sampled datasets



Seems to be rather stable, but not extremely useful per se…

Principal component analysis – further components

Deflation-type definition: all following vectors maximize variance in the orthogonal subspace of previous ones

- Resulting vectors are orthogonal
- Resulting projections are uncorrelated
- Equiv. to frequency separation!

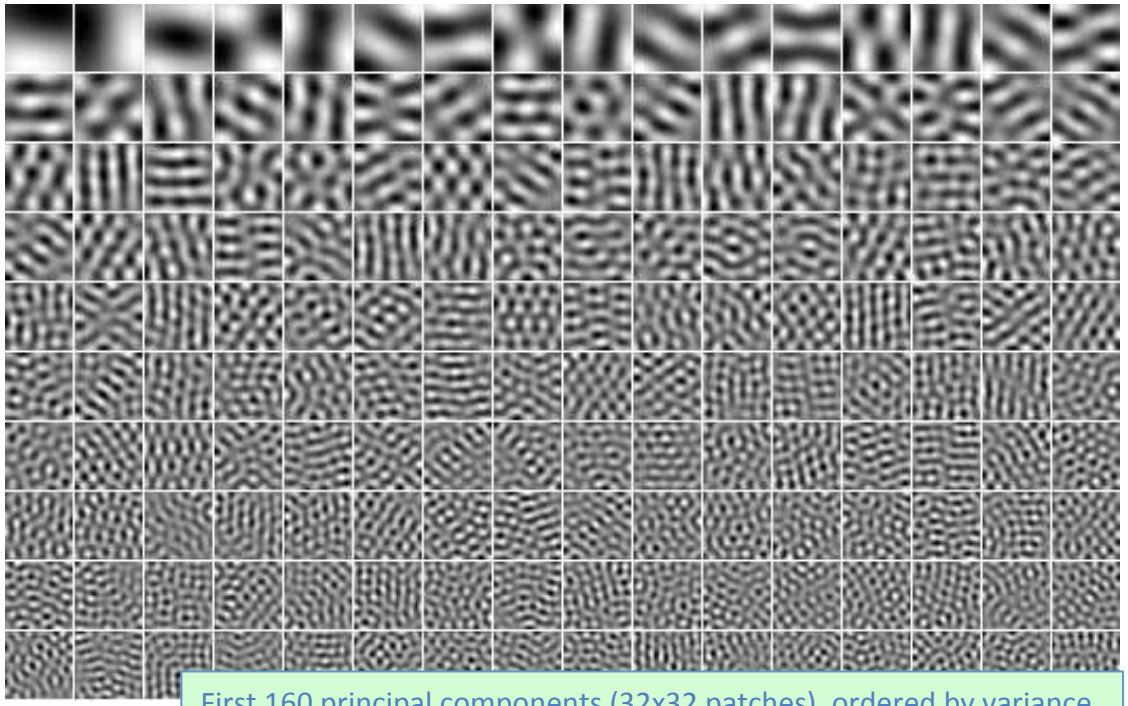
$$\vec{w}_j^* = \underset{\vec{w},\|\vec{w}\|_2=1,\ \vec{w}\perp\vec{w}_1^*,\ldots,\vec{w}_{j-1}^*}{\operatorname{argmax}} \left[\frac{1}{T} \sum_{t=1}^{T} (\vec{w}^T \vec{z})^2 \right]$$

Practical calculation:

- Eigenvectors of correlation mtx (Eigenvalues give variance)
- See Karhunen-Loève transform
- Alt: Fourier of correlation mtx

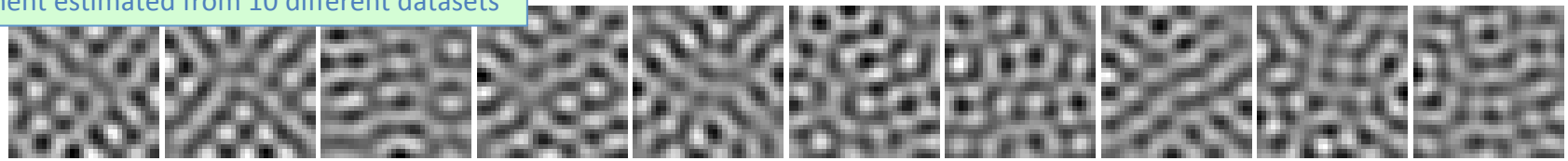
Problems with PCA:

- No reasonable property of images
- Almost same variance for higher components -> poorly defined!
- Depends on random fluctuations



First 160 principal components (32x32 patches), ordered by variance

100th component estimated from 10 different datasets

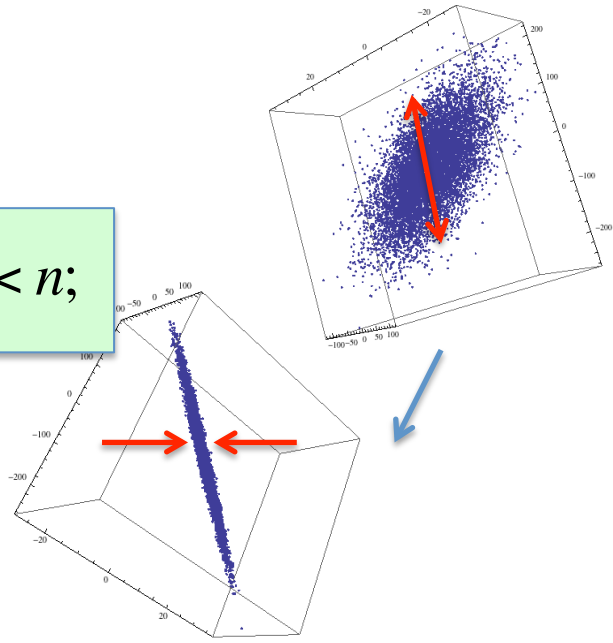


Why is PCA useful?

Dimension reduction by PCA

- All pixel values (n = 32 x 32 = 1024) - too many DoF
- By linear transform, reduce to m < n values:

$$\vec{z} = (z_1, \ldots, z_n)^T, \; s_i = \vec{w}_i^T \vec{z}, \; z_j = \vec{a}_j^T \vec{s}, \; i = 1, \ldots, m, \; m < n;$$



- New variables: “preserve information”; in other words:
- In reconstructing z, minimize average squared error:

$$E\left\{ \left\| \vec{z} - \sum_i \vec{a}_i \cdot s_i \right\|^2 \right\} \rightarrow \min$$

- Assumptions: orthonormal transformation basis,

$$\vec{w}_i^T \vec{w}_j = \delta_{ij}$$

- **Solution**: take m first principal components!
- Defined up to arbitrary rotation in m-dim space
- Non-unique definition of directions, but more stable principal subspace (spanned by principal directions)

For natural images: ~ 10% of dimensions are enough!

Proof for 1st principal component:

$$s = \vec{w}^T \vec{z}, \Delta = E\left\{ \left\| \vec{z} - s \cdot \vec{w} \right\|^2 \right\}$$

$$\Delta = E\left\{ \left\| \vec{z} \right\|^2 \right\} + E\left\{ (\vec{w}^T \vec{z}) \cdot (\vec{w}^T \vec{z}) \cdot \left\| \vec{w} \right\|^2 \right\} - 2E\left\{ (\vec{w}^T \vec{z}) \cdot (\vec{w}^T \vec{z}) \right\}$$

$$\left\| \vec{w} \right\|^2 = \vec{w}^T \vec{w} = 1,$$

$$\Delta = E\left\{ \left\| \vec{z} \right\|^2 \right\} - \vec{w}^T E\left\{ \vec{z} \cdot \vec{z}^T \right\} \vec{w} = \mathrm{var}(\vec{z}) - \vec{w}^T C(\vec{z}) \vec{w}$$

$$\min \Delta:$$

$$L = \vec{w}^T C(\vec{z}) \vec{w} - \lambda \cdot (\vec{w}^T \vec{w} - 1),$$

$$\frac{dL}{d\vec{w}} = 2 \cdot \vec{w}^T C(\vec{z}) - 2\lambda \cdot \vec{w}^T = 0, \Leftrightarrow \left[C(\vec{z}) - \lambda I \right] \vec{w} = 0.$$

Why is PCA useful?

Whitening by PCA

- Transform to variables which are uncorrelated and have unit variance:

$$E\{s_i s_j\} = \delta_{ij}$$

- Solution: re-scale principal components,

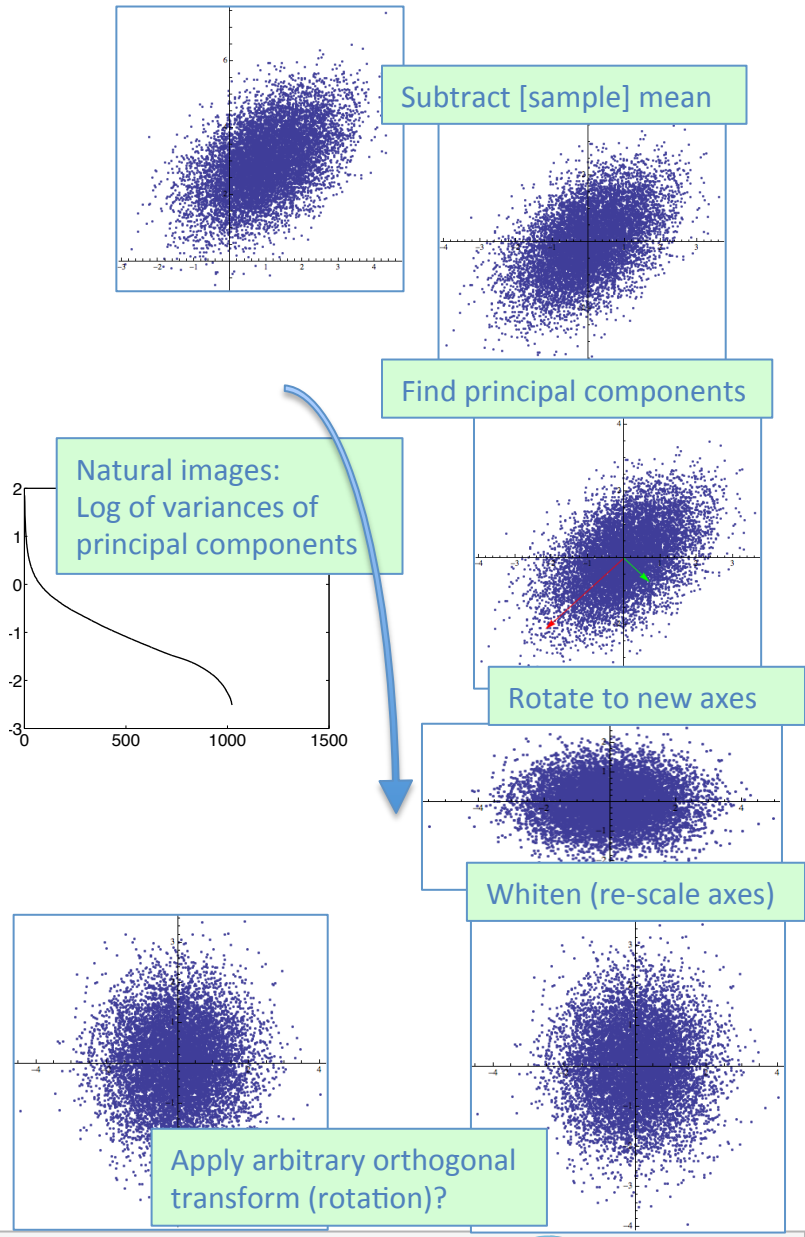
$$y_i = \frac{s_i}{\sqrt{\mathrm{var}(s_i)}}$$

- Completely removes second-order information! (i.e. correlations and variances)
- Standard pre-processing of statistical data
- There exist many whitening transforms; in fact, any orthogonal rotation of whitened data is white
- White distribution with highest entropy: Gaussian

$$\vec{y} \sim N(\vec{0}, I)$$

- Zero correlation of features = orthogonality of their vectors in whitened space

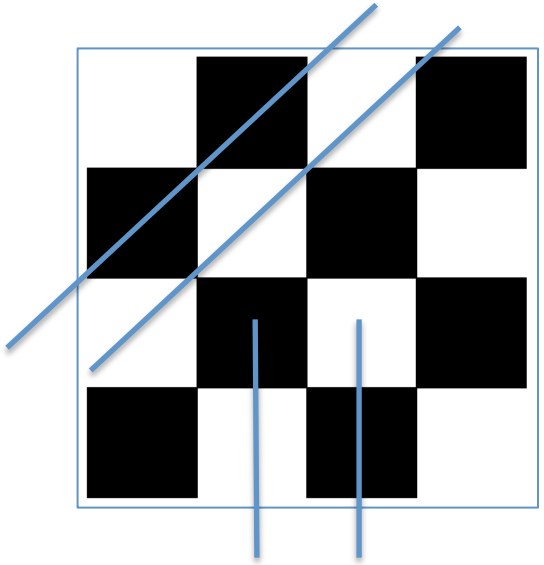
Further analysis: non-Gaussianity of natural images



Why is PCA useful?

Anti-aliasing by PCA:

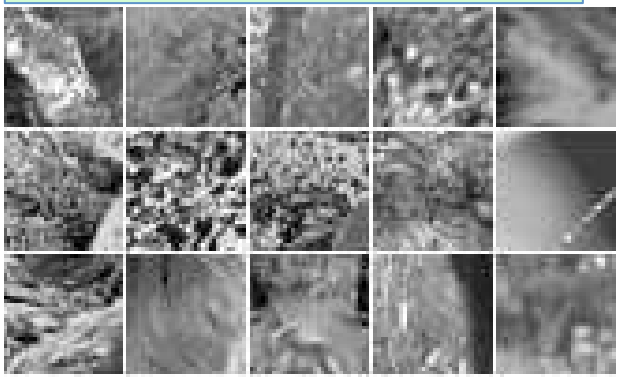
- Highest H or V frequencies can have only two phases
- Along diagonal, max frequency is √2 times higher
- Non-isotropic representation of isotropic natural images!
- Simple dimensional reduction by PCA simultaneously filters oblique frequencies away!
- PCA can be formulated in Fourier space

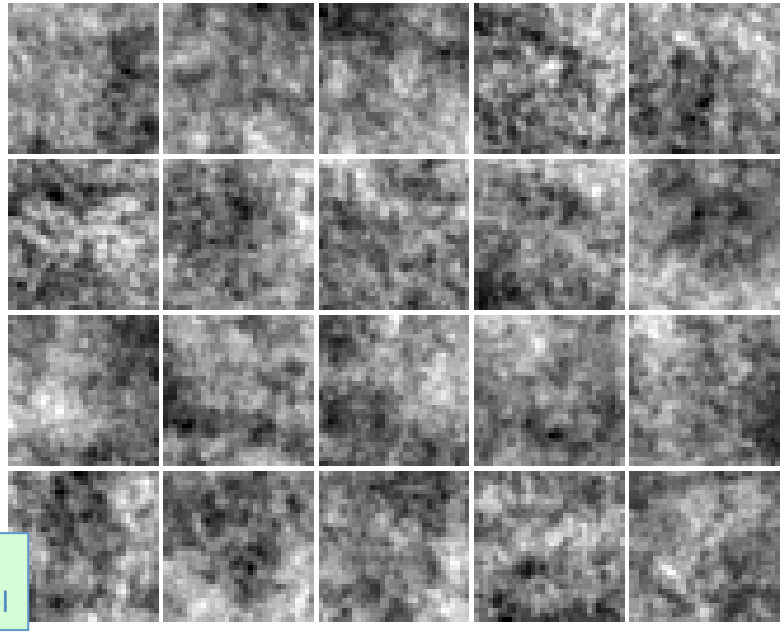


PCA as generative model

- Generate random components from Gaussian distribution with s_i according to estimated variances
- Perform inverse transform, i.e. reconstruct images

Natural randomly sampled 32x32 images





20 random 32x32 images generated from PCA model

Beyond second-order statistics: sparse features

- PCA features do not resemble neural receptive fields
- PCA is not very successful generative model

One choice of higher-order property: sparseness

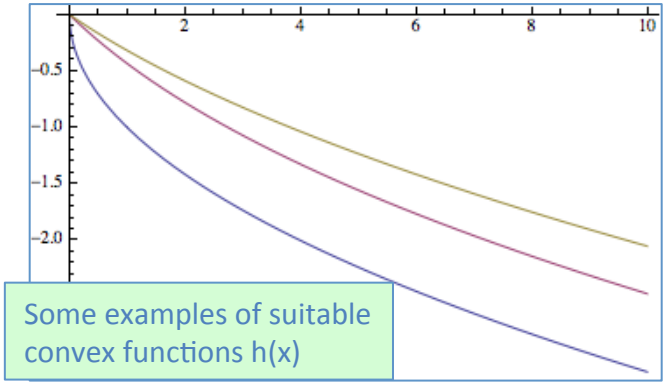
- Approximate definition: random variable most of time very close to zero ("rarely active")
- Not the same as small variance!

Measure of sparseness

- Maximize non-linear function of squared variable,

$$E\{h(s^2)\} \rightarrow \max$$

- Need h(x) to be convex: near zero – higher weights



Some examples of suitable convex functions h(x)

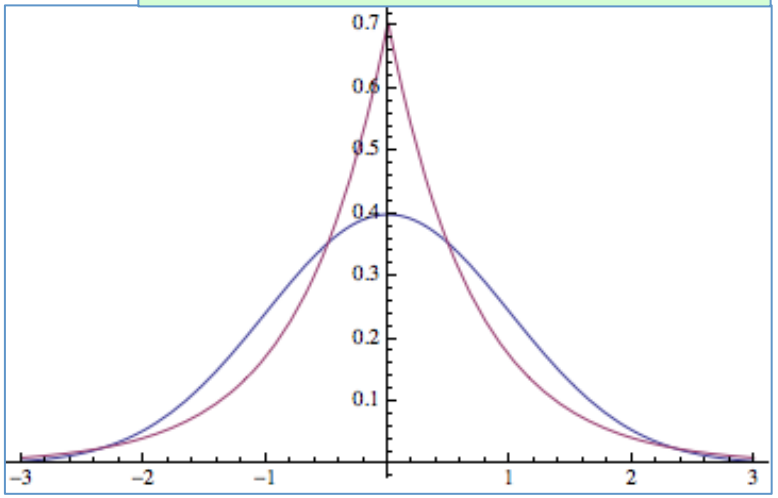
Commonly used:

- kurtosis (4-th moment),
- $-x^{1/2}$
- $-\log \cosh (x^{1/2})$
- $-(x + e)^{1/2}$

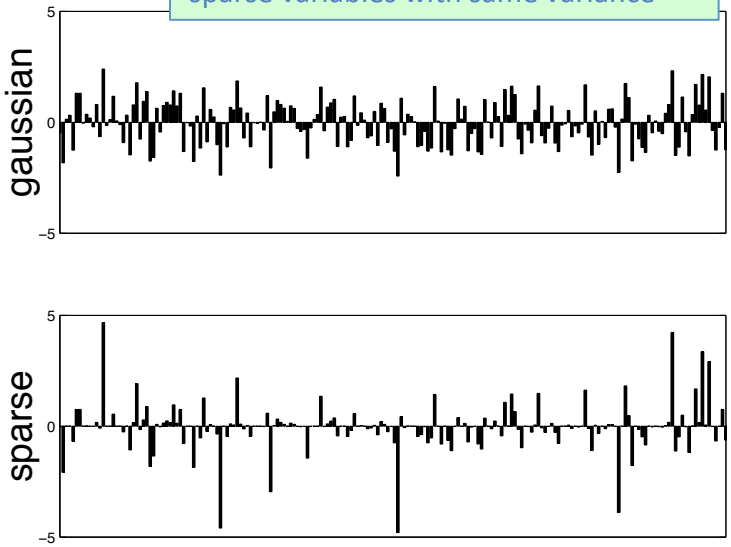
Optimal measure:

- $\log p(x^{1/2})$

Gaussian and sparse PDFs with same variance



Samples of normally distributed and sparse variables with same variance



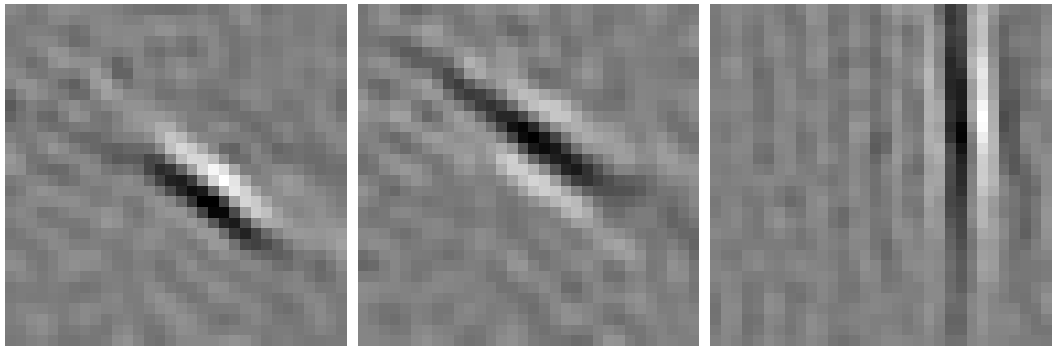
Linear feature detectors with maximum sparseness

- Start from PCA-pre-processed data
- Choose convex function h(x)
- Learn one feature maximizing sparseness:

$$\vec{v}^* = \arg\max_{\vec{v},\|\vec{v}\|=1} E\left\{h\left((\vec{v}^T\vec{s})^2\right)\right\}$$

Results (single feature)

- Features localized in space, frequency, orientation
- Resembles receptive fields of neurons
- Many local maxima, all well-localized!



Weights found by sparseness maximization in natural images, different initialization (i.e. local maxima)

Learning multiple sparse features

- Could start from different points (can find same max many times!)
- Better method: deflation-based
- Find local maxima satisfying some constraints
- Possible choice: subsequent features un-correlated with previously found ones (= orthogonal in whitened space),

$$\vec{v}_j^* = \arg\max_{\vec{v},\|\vec{v}\|=1} E\left\{h\left((\vec{v}^T\vec{s})^2\right)\right\},$$

$$E\left\{(\vec{z}^T\vec{v}_j^*)(\vec{z}^T\vec{v}_i^*)\right\} = 0, \forall 1 \le i < j$$

- Leads to gradual deterioration of features (too strict constraints);
- Last vectors have too little space to optimize

Symmetric de-correlation

Better yet solution: maximize sum of individual sparseness measures, under constraints of unit variance and symmetric de-correlation:

$$\{\vec{v}_1,\ldots,\vec{v}_n\}^* = \underset{\{\vec{v}_1,\ldots,\vec{v}_n\}}{\operatorname{argmax}} \sum_{i=1}^{n} E\left\{h\left((\vec{v}_i^T\vec{s})^2\right)\right\}$$

$$E\left\{(\vec{v}_i^T\vec{s})(\vec{v}_j^T\vec{s})\right\} = \delta_{ij}$$

Number of features limited by pre-processing step (PCA)

Small philosophic problem:

Above: **Sparseness of <u>feature</u>** = single feature values over sample images (as function of t)
Wanted: **Sparseness of <u>representation</u>** = feature values over index i for a single image

- Similar to spoken language: many words, each phrase contains only a few of them
- Number of features can exceed dimensionality of data!

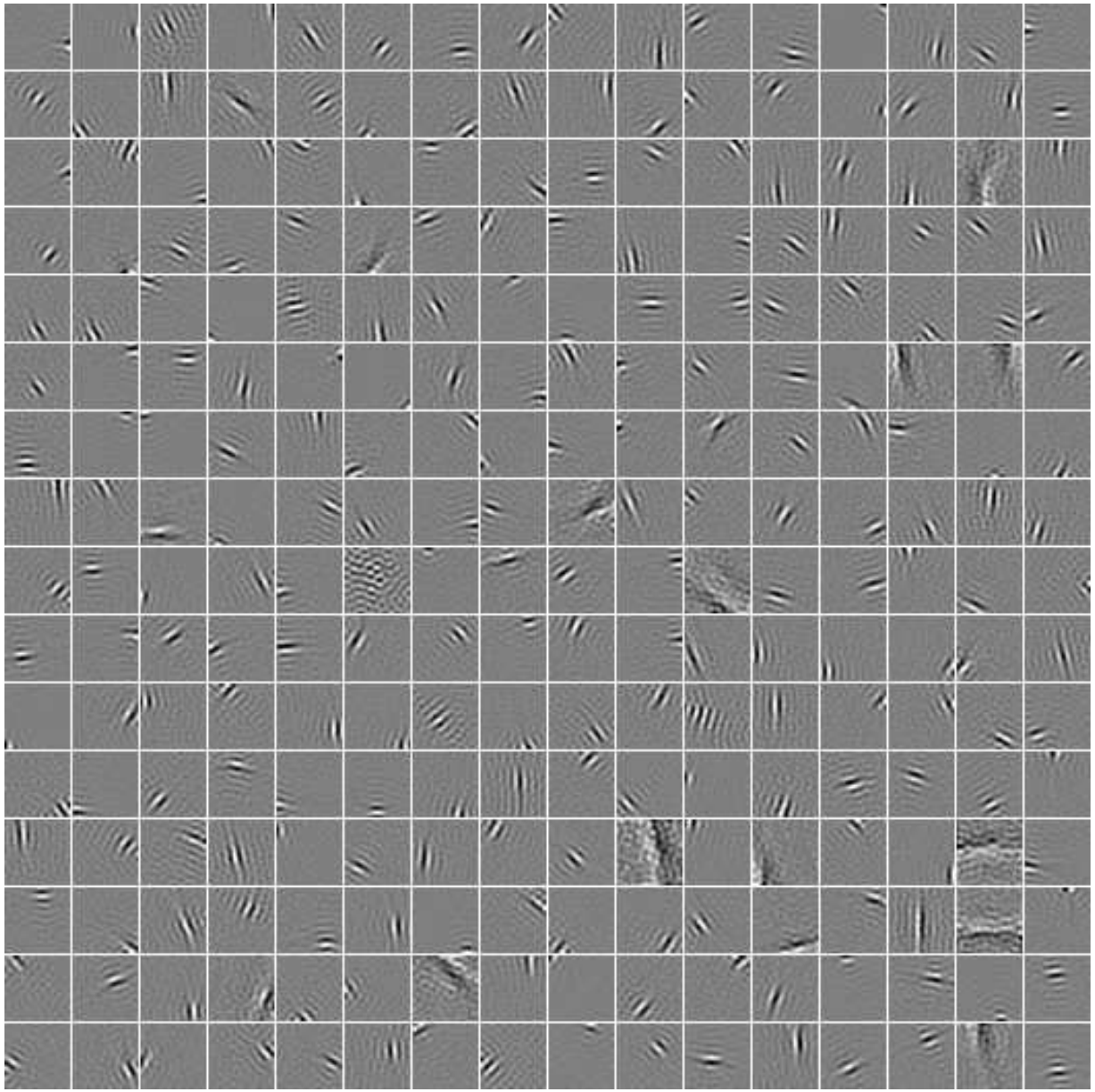
Both definitions are equivalent if following conditions hold <u>for a single typical image</u>:

- The mean of features is approximately zero
- Mean of square of features equals approximately one

Under these conditions, $\sum_{i=1}^{n} h\left((\vec{v}_i^T\vec{s})^2\right)$ measures sparseness of single image, and sum over sample is equivalent to formula above

This holds if features are statistically independent and have identical distribution (iid)

Sparse coding feature detectors from natural images



Computation:

- DB of 50000 32x32 patches
- No ordering (symmetric DC)
- PCA reduction to D = 256
- FastICA algorithm

Features:

- Localized in space
- Well-defined orientation
- Multi-scale (small and large)
- Can be studied as neural RFs, with distributions over frequency, phase, …
- Many entries just shifted copies of each other

Is that the best we can do?

Why is sparseness useful?

So far:

- Better statistical model of input data
- Efficient coding of images (many zeros after transform, easy to compress)
- Sparse features resemble RFs of simple cells in visual system
- Some reasonable explanation: neurons trying to minimize firing rate to save energy

Still have open questions:

- Sparseness measure was ad hoc. What is optimal function h(x)?
- Why was de-correlation (of many features) needed?
- What is true Bayesian prior distribution of images?

Further refinement: **generative models**

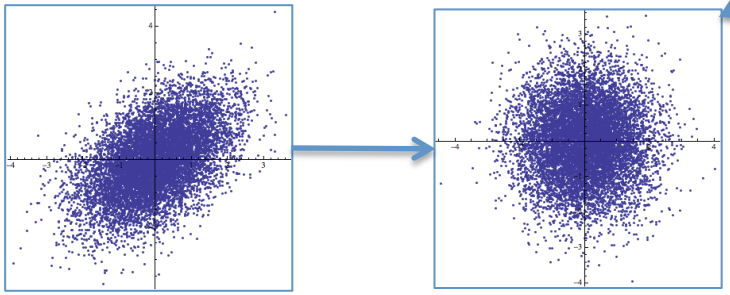
- Observed data (images) generated as transformations of hidden (latent) variables
- Specific flavor of model: Independent Component Analysis (ICA)
- ICA in some specific formulation equivalent to finding maximally sparse features!

Main idea: assume transformed coefficients statistically independent over sample,

$$s_i = \vec{w}_i^T \vec{z}, \quad p(s_1,\ldots,s_n) = p_1(s_1)\cdot\ldots\cdot p_n(s_n), \mathrm{var}(s_i) = 1$$

Why does PCA not produce independent components?

Gaussian distribution: PCA and whitening

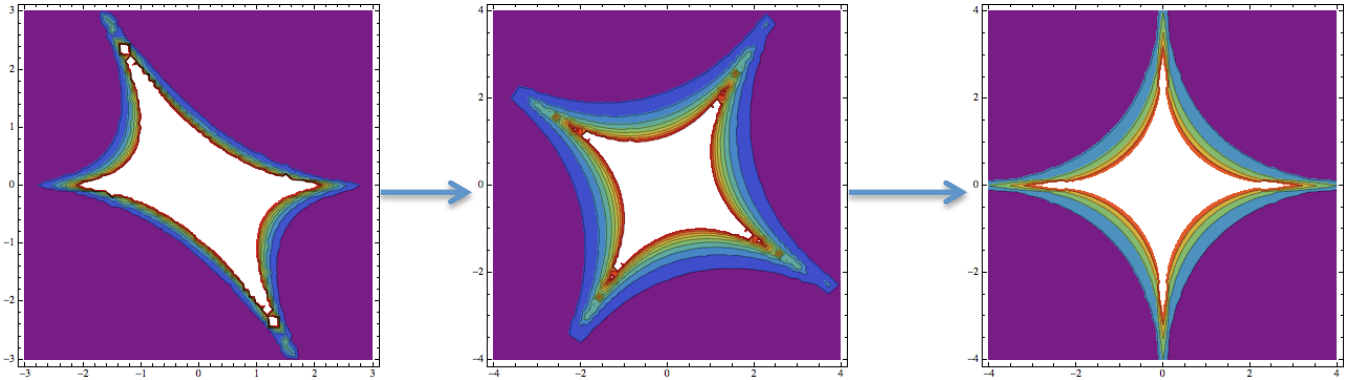


- All rotations equivalent
- PDF spherically symmetric
- Uncorrelated Gaussian variables are already independent!

Using only second-order information:

- Correlation matrix symmetric, i.e. $n(n+1)/2$ values
- Transform matrix needs n^2 independent values
- Remaining: $n(n-1)/2$ rotation angles

Sufficiently non-gaussian distribution: PCA and whitening, then ICA step



Benefits of ICA:

- Strong relation to sparseness
- Provides optimal sparseness measure
- Justifies de-correlation
- PDF for Bayesian inference

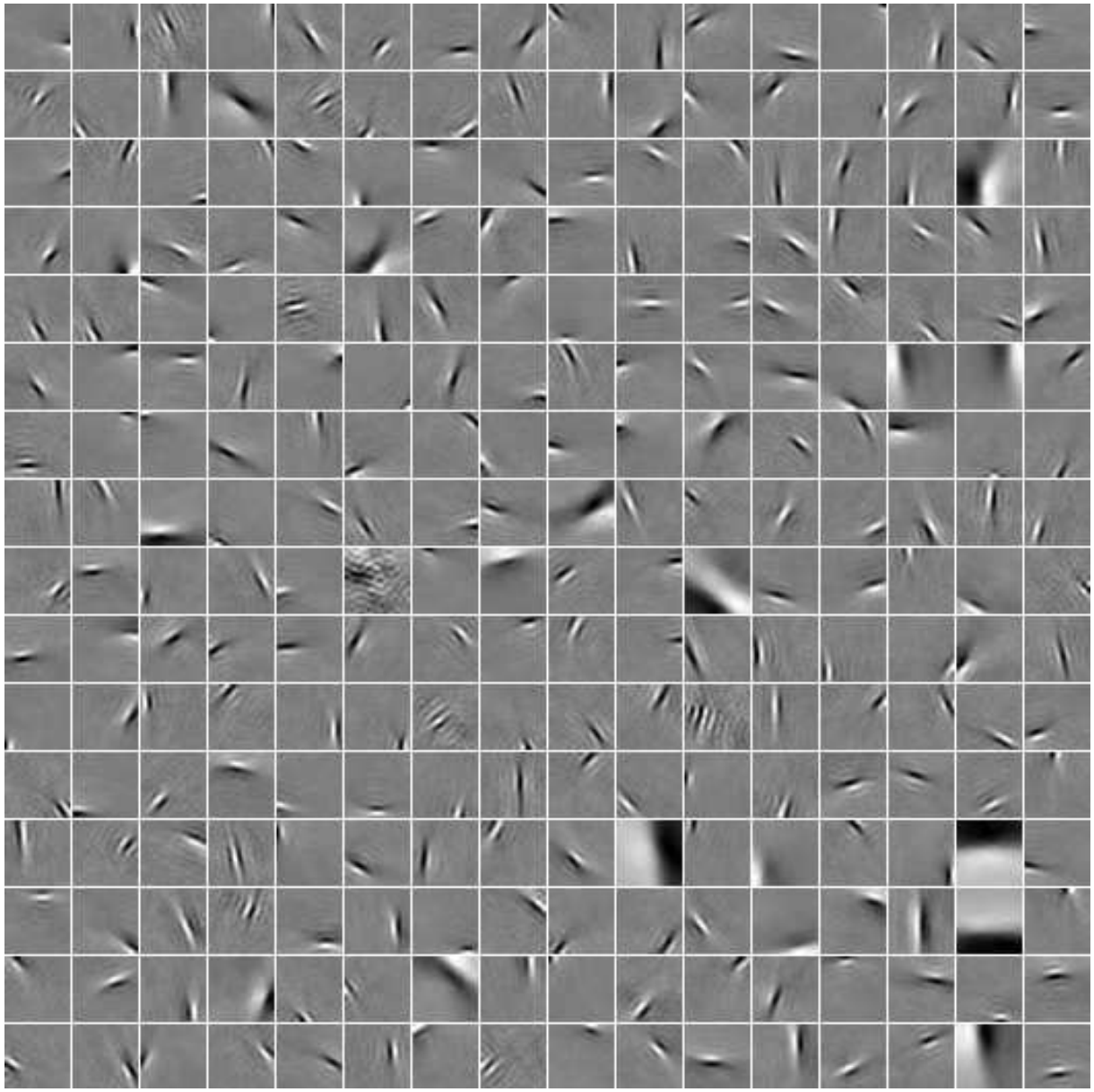
ICA feature learning: maximum (log) likelihood estimation

$$s_i = \vec{v}_i^T \vec{z}, p(s_1,\ldots,s_n) = p_1(s_1)\cdot\ldots\cdot p_n(s_n), \Rightarrow p(\vec{z}) = \left|\det(V)\right| \cdot \prod_{i=1}^{n} p_i(\vec{v}_i^T \vec{z})$$

$$\{\vec{v}_1,\ldots,\vec{v}_n\}^* = \arg\max_{\vec{v}_1,\ldots,\vec{v}_n} \left[\log\left(\left|\det(V)\right|\right) + \sum_i \log\left(p_i(\vec{v}_i^T \vec{z})\right) \right]$$

Many methods exist to find exactly this maximum!

ICA generating basis from natural images



Computation:

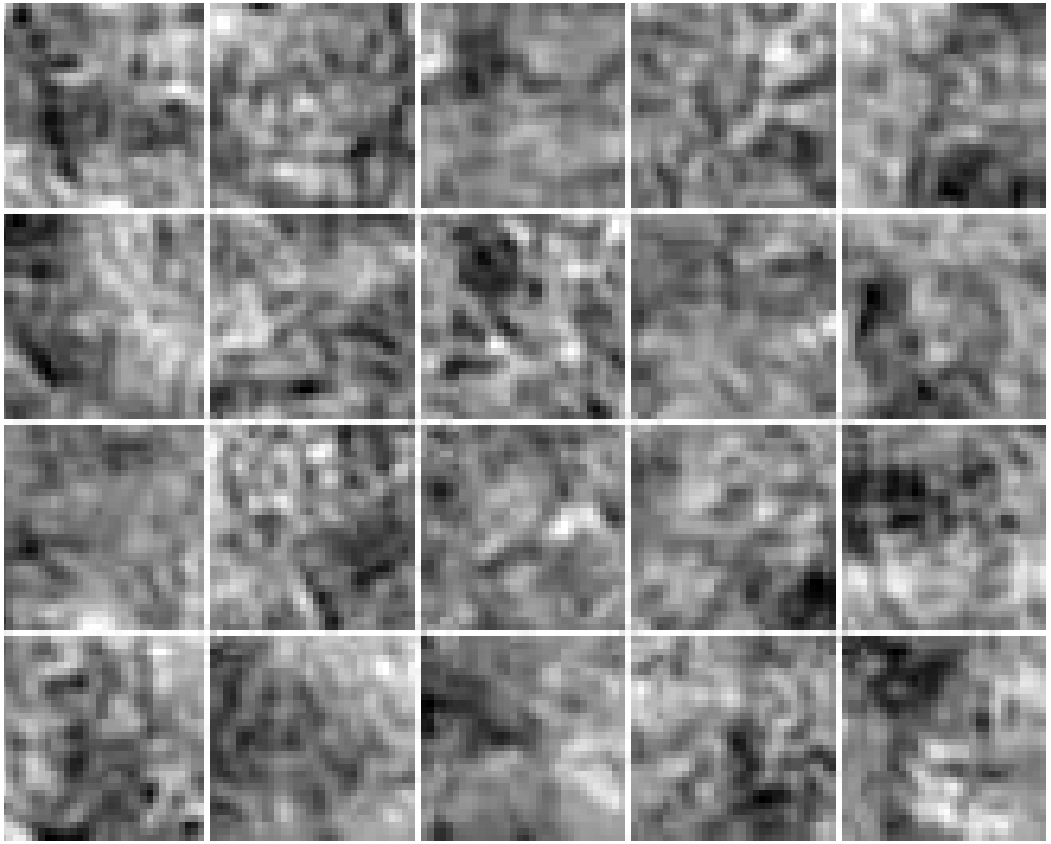
- DB of 50000 32x32 patches
- No ordering (symmetric DC)
- PCA reduction to D = 256

Features:

- Similar to sparse features
- Same properties wrt localization as before
- Frequency channels seem to be statistically independent!
- Extracted features still exhibit some dependence!

Can be used to generate "typical" images!

Image patched generated from ICA model



Computation:

- Marginal distributions over each feature estimated from real images

Results:

- Better than PCA: edge-like structures
- Still, not extremely "natural" ☺

Further research directions:

- Minimum-entropy coding
- Over-complete bases
- Non-negative models
- Non-linear features, energy detectors (cf. complex cells)
- Independent Subspace Analysis (ISA)
- Multi-layer models
- Modeling extra-striate cortex (V2, …)
- Markov Random Field models
- …

Still an open field, more work is required!

Merry Christmas and happy New Year!

Next lecture: 08.01.2014