# Image Data Compression

## Natural image statistics


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

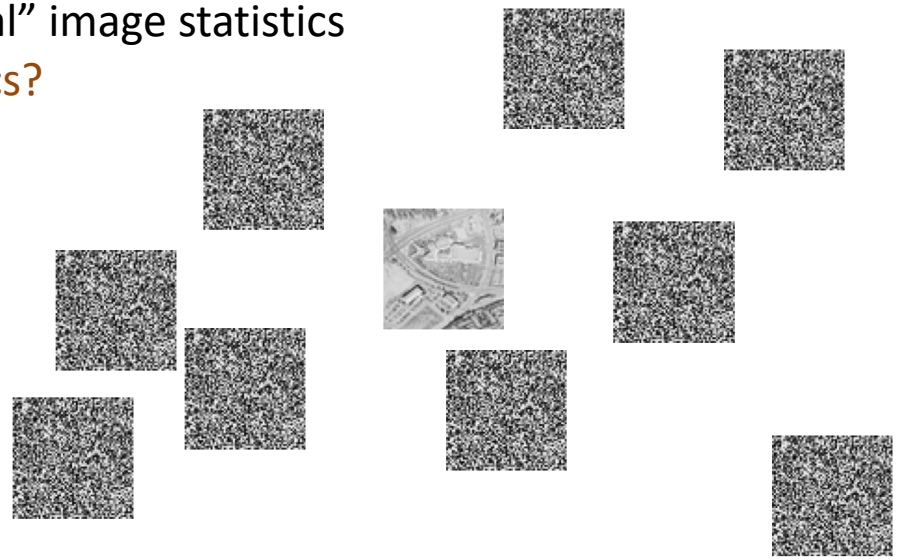
# Why do we need to study natural image statistics?

- Need good data models to have better compression methods
- Relatively recent field (the first comprehensive book on subject: 2009)
- Closely related to human visual system studies
  (conjecture: biological visual systems are “optimal” for the “natural” images)
- Studies facilitated by big image databases and available processing power
- May better understand differences of “unnatural” image statistics
- Why is it challenging to study the image statistics?

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

**A couple of (relatively) big numbers:**

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

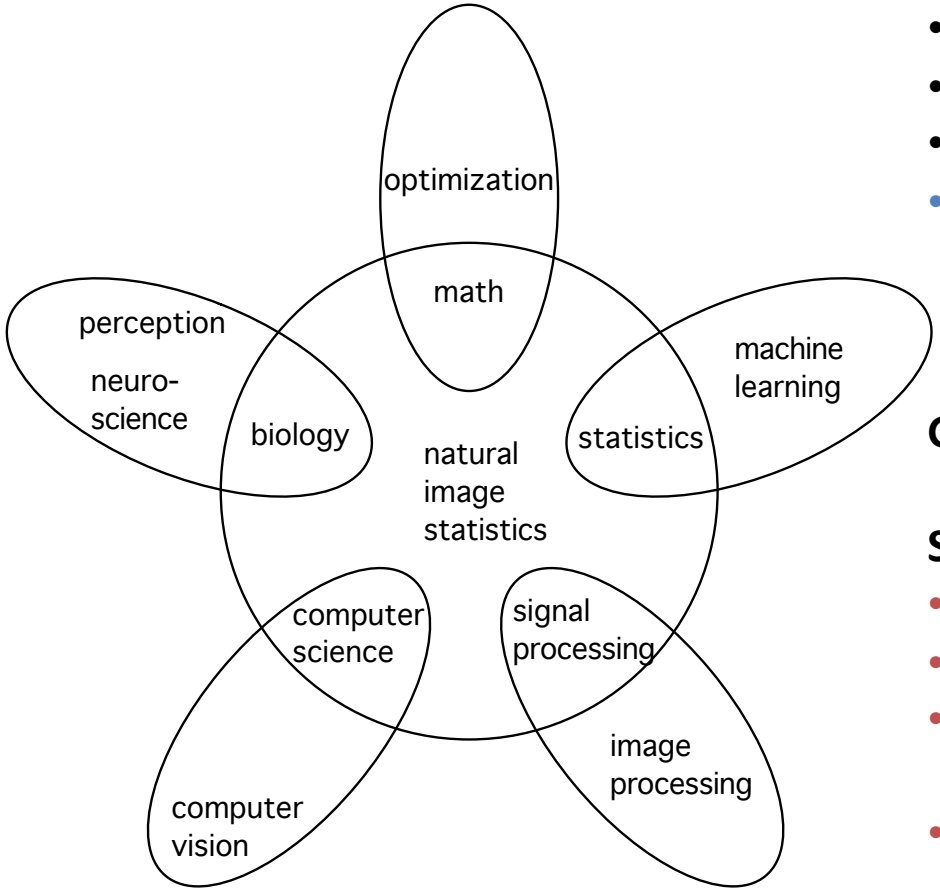


*“The distribution of natural images is complicated. Perhaps it is something like beer foam, which is mostly empty but contains a thin mesh-work of fluid which fills the space and occupies almost no volume. The fluid region represents those images which are natural in character.”*

[Ruderman 1996]


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Very multi-faceted research area




**Models of natural image data**

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

**Goal: transform images to "convenient" spaces**

**Some proposed criteria of "good" models:**

- "simplicity" ≈ reduced dimensionality
- "sparseness" ≈ compact representation
- "metabolic efficiency" ≈ energetically efficiently processed by neurons in brain
- "learning efficiency" ≈ easy to learn from unique images (each data sample is seen only once)
- "wiring length" ≈ require minimal communication length between neurons
- …

**It all starts with statistics…**


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Reminder of multivariate statistics (I)

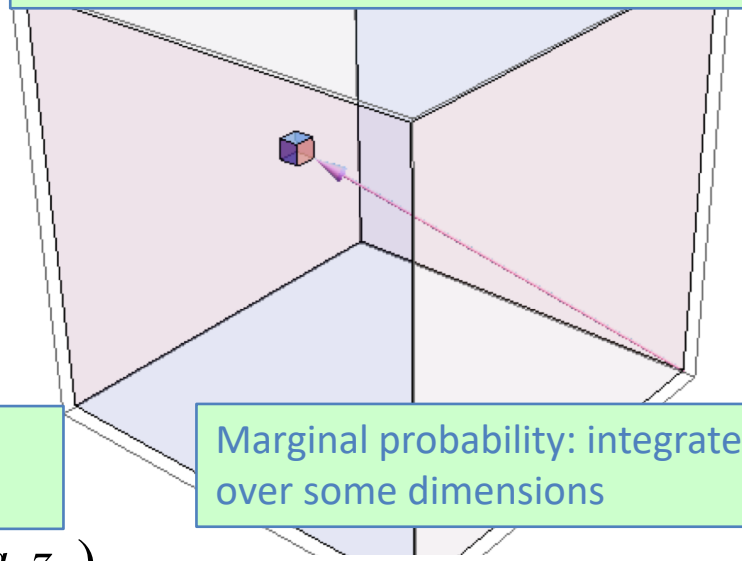
- Probability density function (PDF):

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

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

Slightly ambiguous, but shorter notation:

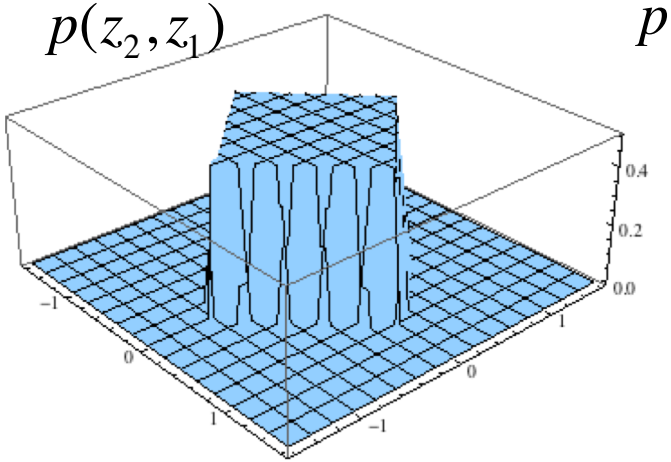
$$p(\vec{z}) \equiv p(z_1, z_2, \ldots, z_n) \equiv p_{\vec{z}}(\vec{z})$$



- Conditional vs marginal probability:

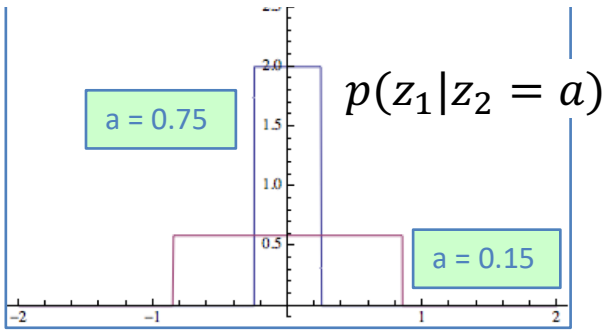
Joint probability: multi-dimensional PDF

$p(z_2, z_1)$



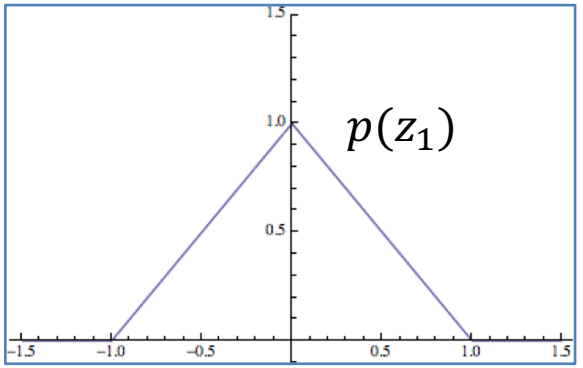

Conditional probability: fix some dimensions

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




Marginal probability: integrate over some dimensions

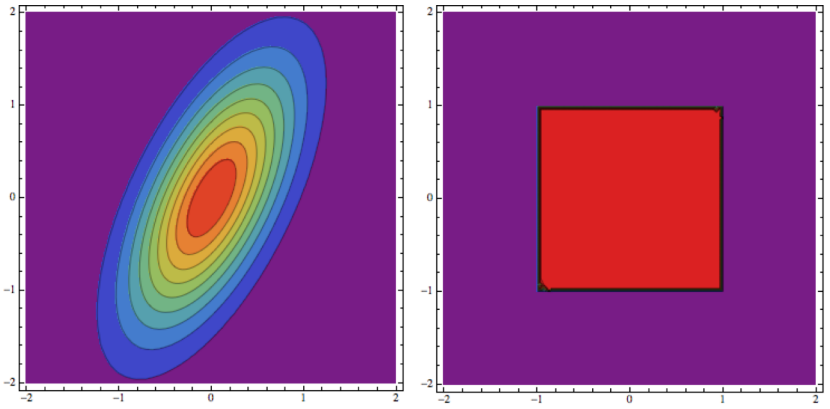
$$p(z_1) = \int p(z_1, z_2)\, dz_2$$



# Reminder of multivariate statistics (II)

- Variables $z_1$ and $z_2$ are independent iff their joint PDF factorizes:

$$p_{\vec{z}}(\vec{z}) \equiv p(z_1, z_2) = p(z_1)p(z_2)$$



- Expectation of a random vector (or its function):

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

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

1-st order statistic

**Note:** independent variables are uncorrelated, i.e.

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

but uncorrelated variables may still be dependent!!!

- Variance and covariance (1-dimensional case) :

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

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

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 (III)

unobserved parameters

observation

likelihood

a priori probability (prior)

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

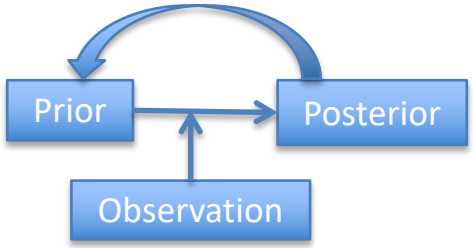
a posteriori probability (posterior)

normalization constant (independent of s)

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

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

Bayes' formula: incremental learning




**Parameter estimation** (usually performed in log space)

Constant, independent of α

- Bayes' formula: $\log p_\alpha(\alpha \mid z_1, z_2, \ldots, z_m) = \log p(z_1, z_2, \ldots, z_m \mid \alpha) + \log p(\alpha) - \log p(z_1, z_2, \ldots, z_m)$
- 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 (ML): $\alpha^*_{ML} = \underset{\alpha}{\operatorname{argmax}} \left[ \log p(z_1, z_2, \ldots, z_m \mid \alpha) \right]$

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 via least squares method:

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

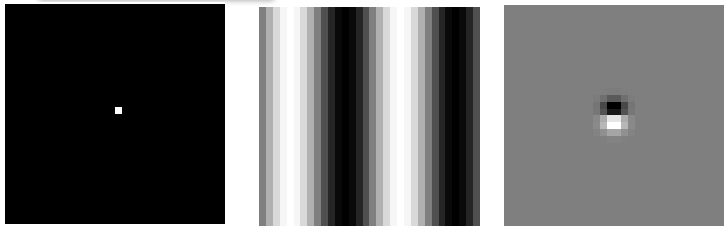
# Simple statistics of linear features

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

"Vectorize" image; gray-scale values

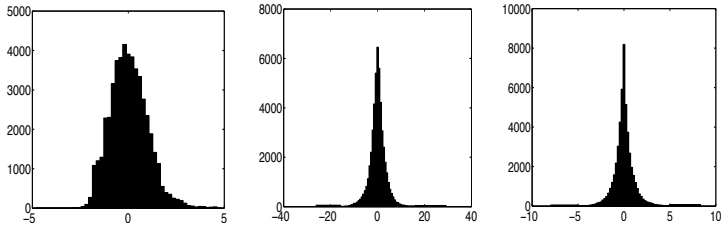
Basis functions, "features", can also be interpreted as images

- Examples of features:



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

- Histograms of filter outputs:



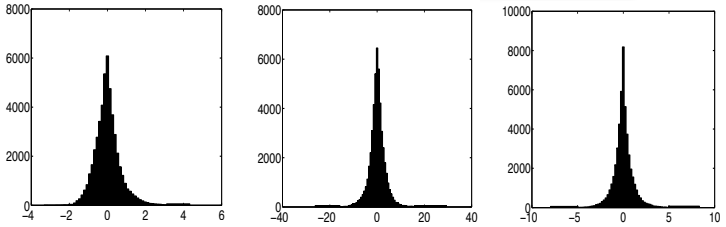
Two last histograms very non-Gaussian!

- Simplest image statistic: mean (DC component)
  (not to be confused with the 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 are orthogonal to mean value, Dirac filter is not orthogonal

**In what follows, always assume subtracted mean**

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

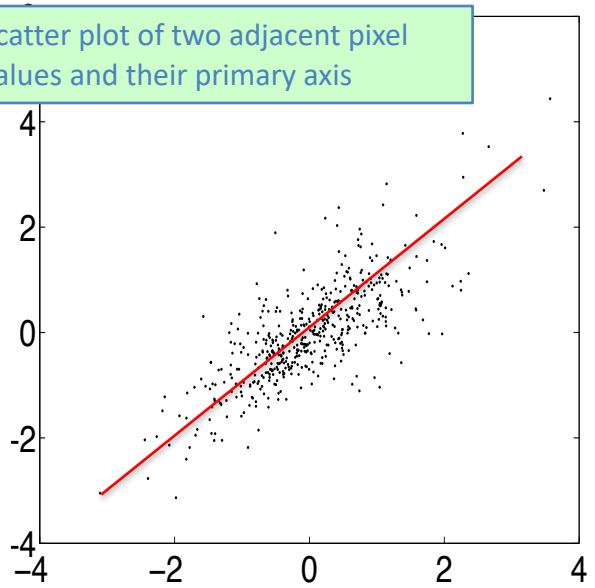
# Principal component analysis

- Same as KLT applied to whole n-dimensional patches
- Second-order structure (pixel-to-pixel correlations)
- Principal components “capture” the largest data variances
- Technical requirements: mean-subtracted images, limit norm of w by e.g. 1

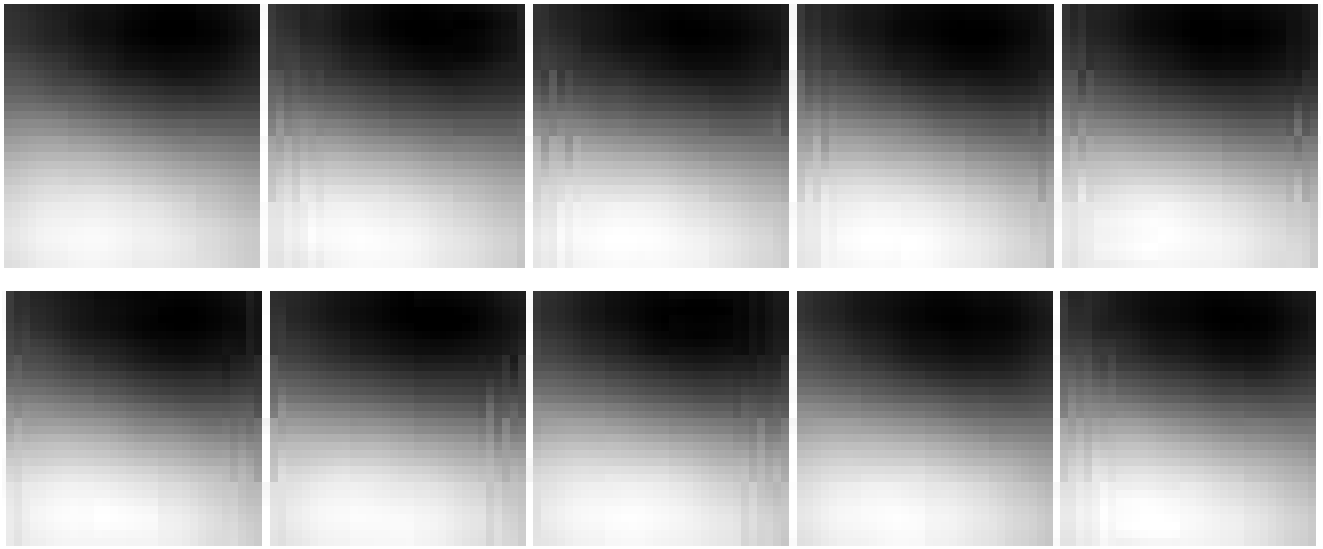
**Learning the first principal component:** max output variance in a given sample,

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

Scatter plot of two adjacent pixel values and their primary axis




First principal component computed for 32x32 patches for 10 different randomly sampled datasets:



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

# Principal component analysis (PCA) – further components

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

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

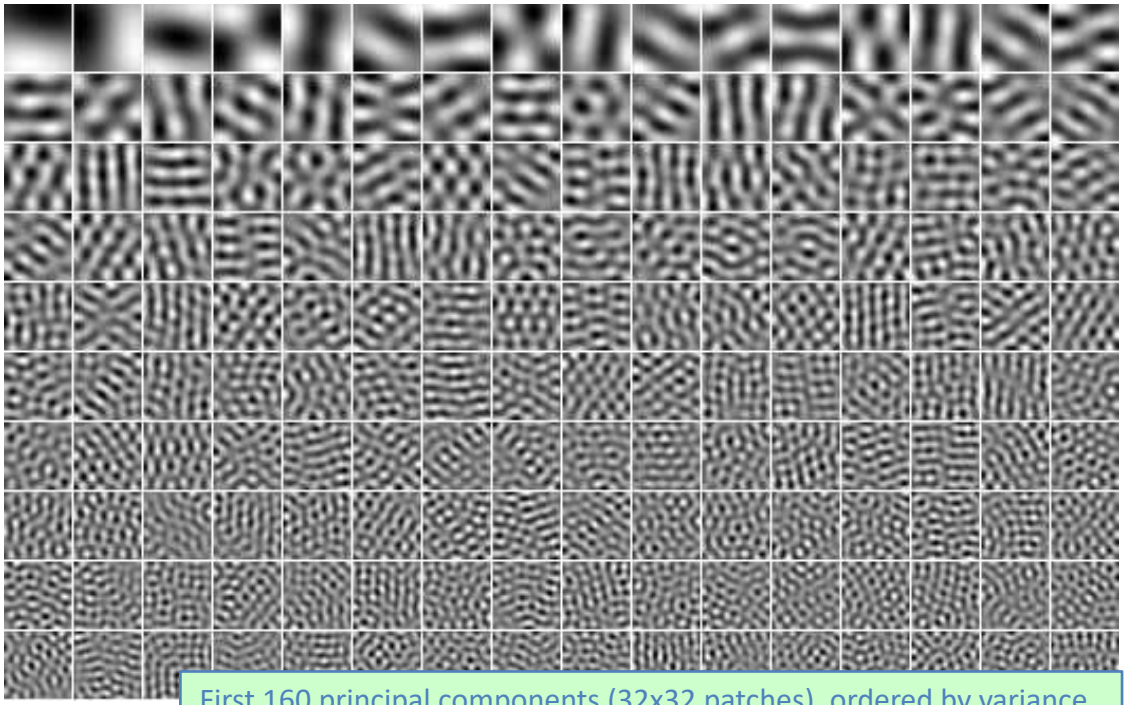
$$\vec{w}_j^* = \underset{\vec{w},\|\vec{w}\|_2=1,\ \vec{w}\perp\vec{w}_1^*,\ldots,\vec{w}_{j-1}^*}{\operatorname{argmax}} \left[ \frac{1}{T} \sum_{t=1}^{T} (\vec{w}^T \vec{z})^2 \right]$$

**Practical calculation:**

- Eigenvectors of covariance matrix (Eigenvalues give variance)
- Cf. Karhunen-Loève transform
- Alt: Fourier of correlation matrix

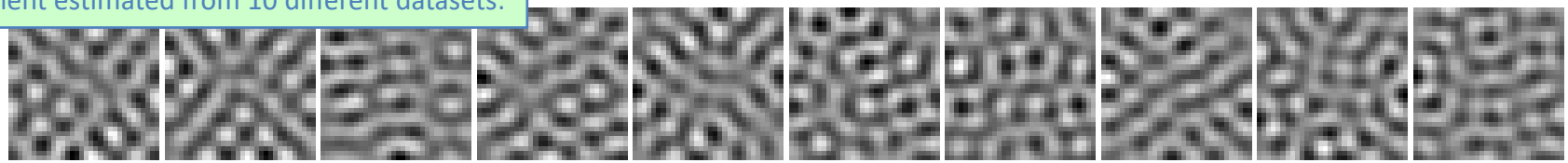
**Problems with PCA:**

- No reasonable property of images
- Similar variances for higher-order components -> unstable!
- Depends on random fluctuations



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

100th component estimated from 10 different datasets:



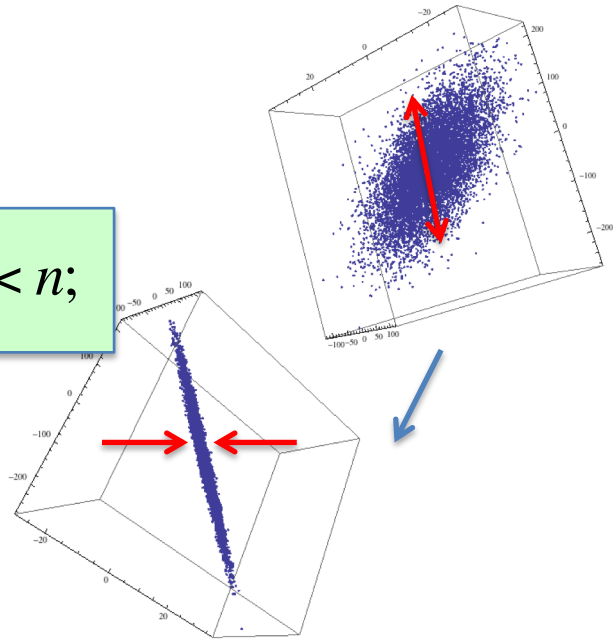

© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Why is PCA useful (I)?

**Dimension reduction with PCA**

- All pixel values (n = 32 x 32 = 1024) - too many variables!
- Apply a 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 must "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 the 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 (II)?

**Whitening by PCA – generic steps**

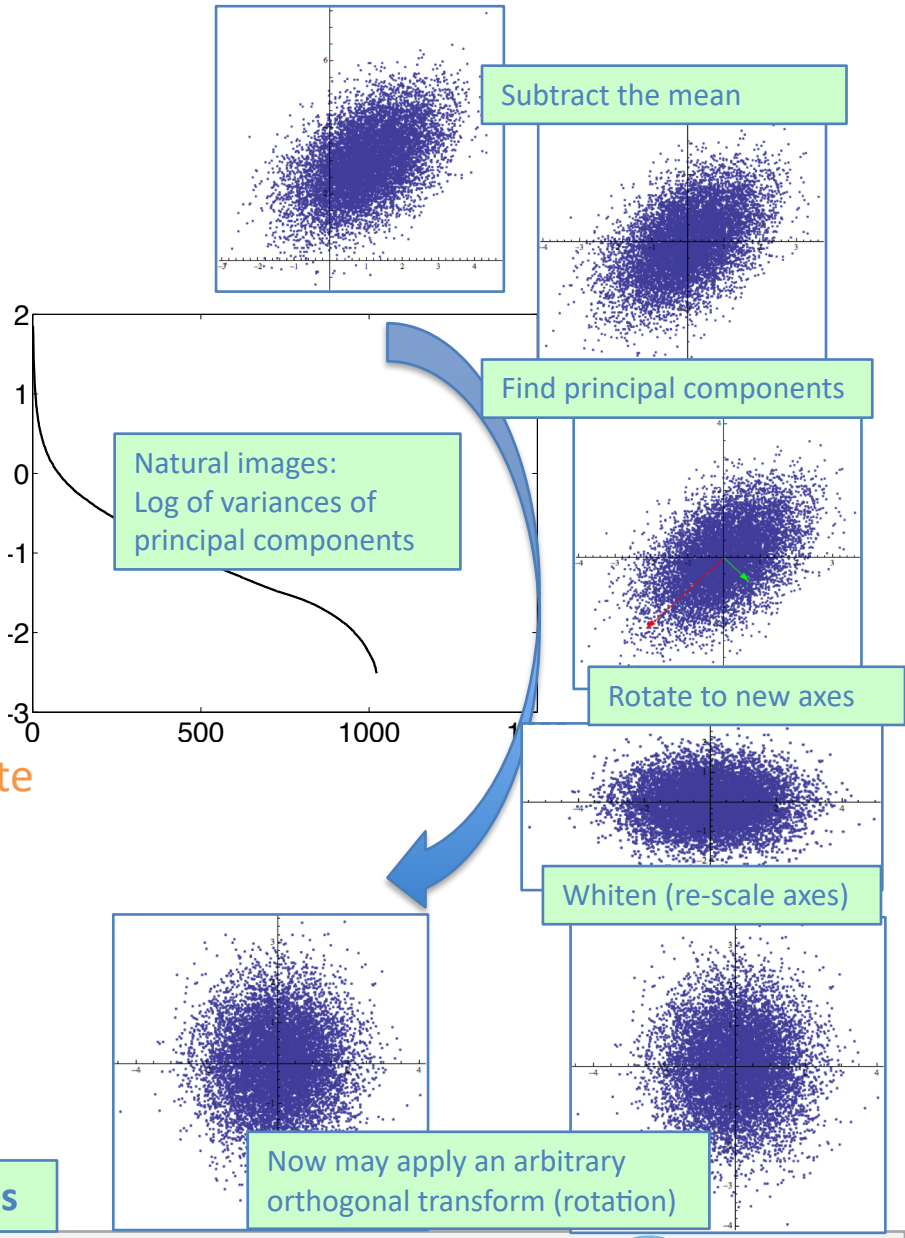
- Goal: transform an image to variables which are uncorrelated and have normalized variance:

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

- Solution: re-scale the principal components,

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

- Completely exploits and removes second-order information (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 the highest entropy – Gaussian: $\vec{y} \sim N(\vec{0}, I)$
- Zero correlation of features = orthogonality of their vectors in the whitened space




**Further analysis: study non-Gaussianity of natural images**

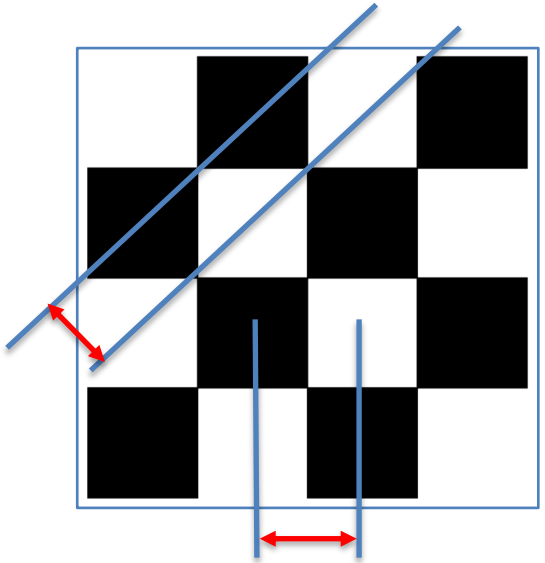


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Why is PCA useful (III)?

**Anti-aliasing of rasterized images by PCA:**

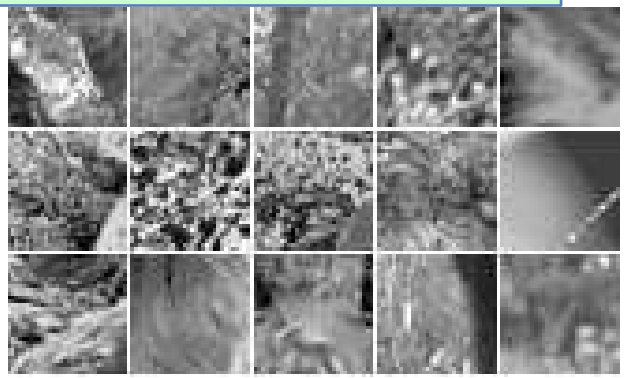
- Highest H or V frequencies can have only two phases
- Along diagonals, max frequency is √2 = 1.414… times higher
- Non-isotropic representation of isotropic natural images!
- Simple dimensional reduction by PCA simultaneously filters oblique high frequencies away, improves anisotropy
- PCA can also be formulated in Fourier space



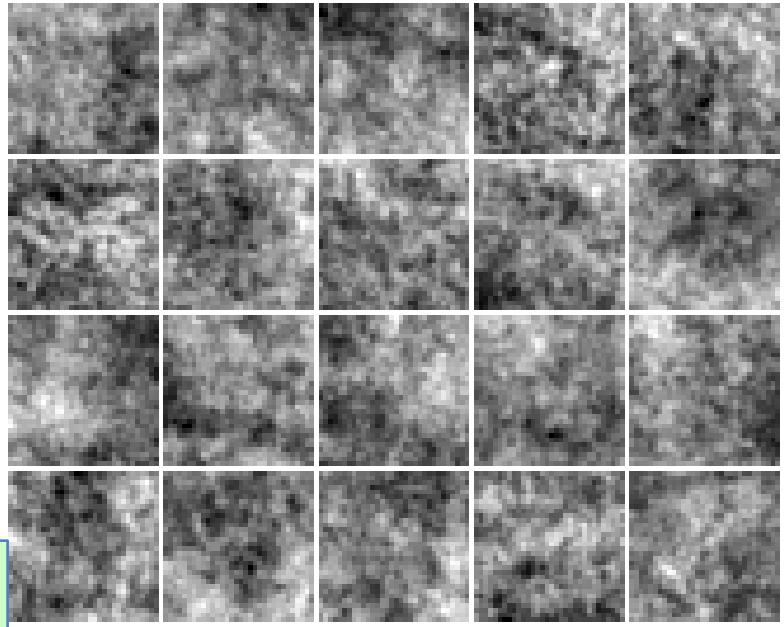
**PCA as a generative model:**

- Idea: generate random components from Gaussian distribution with $s_i$ according to estimated variances
- Perform inverse transform, i.e. “hallucinate” images
- Verdict: relatively poor generative quality!

Natural randomly sampled 32x32 images



Random 32x32 images generated from PCA model



# Beyond second-order statistics: sparse features

**Problems with PCA:**

- Features do not resemble neural receptive fields
- Generative model not extremely successful

**Need higher-order properties. E.g.: sparseness**

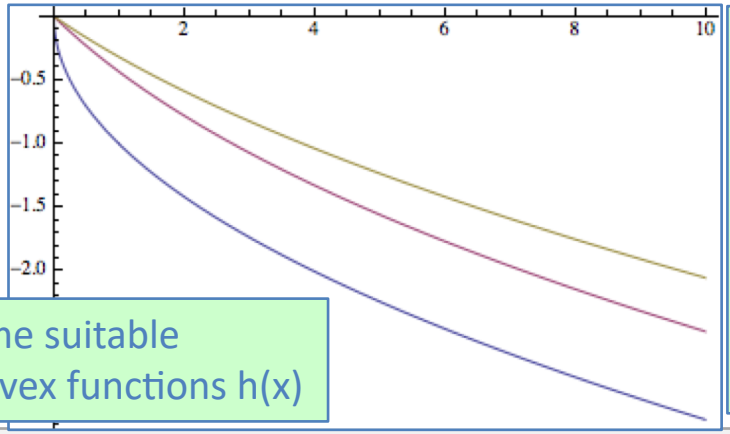
- Approximate definition: random variable components most of time are small ("rarely active")
- Not the same as small variance!

**May use many measures of sparseness**

- Maximize some non-linear function of squared variable,

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

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




Some suitable convex functions h(x)

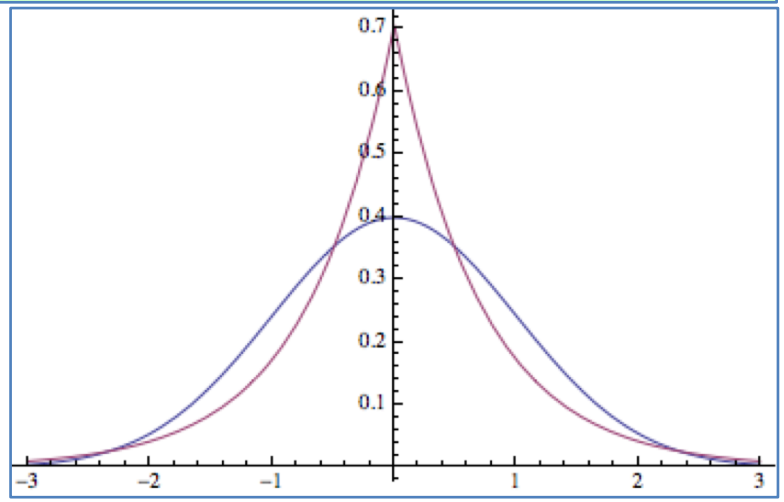
**Commonly used:**

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

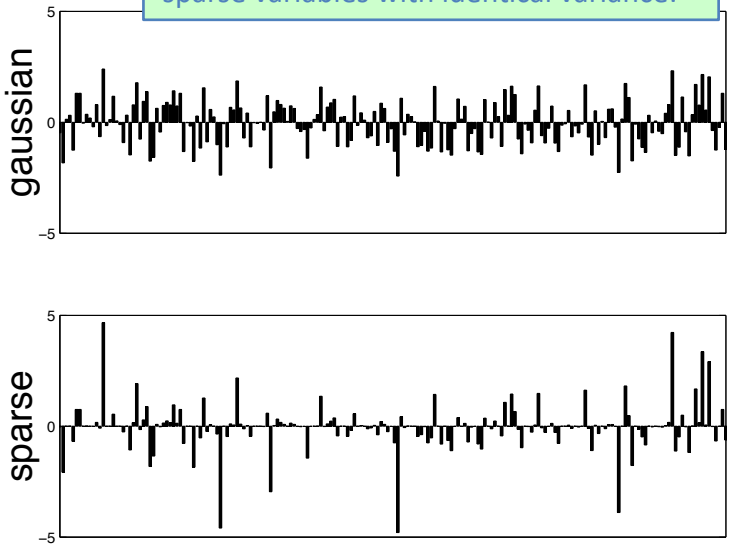
**Optimal measure:**

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

A Gaussian and some sparse PDFs with the same variance




Samples of normally distributed and sparse variables with identical variance:



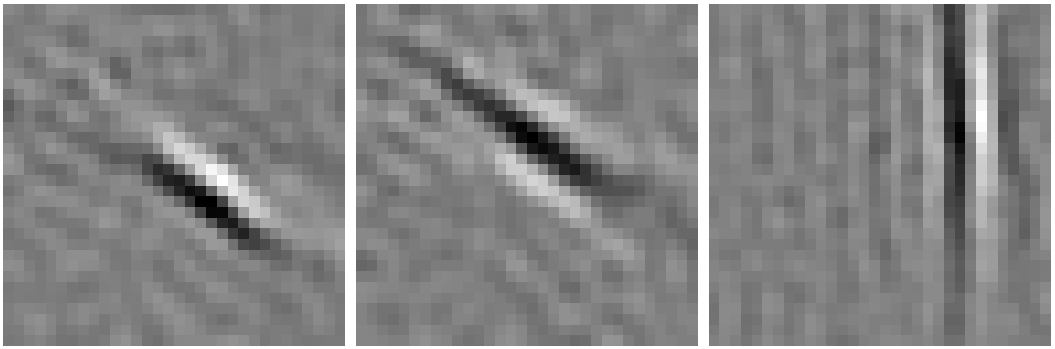
# Linear feature detectors with maximum sparseness

- Start from PCA-preprocessed (whitened) data
- Choose some convex function **h(x)**
- Learn one feature that maximizes sparseness:

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

**Results (finding a single sparse feature):**

- Features localized in space, frequency, orientation
- Resemble receptive fields of neurons
- Problem: many local maxima, all well-localized!



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

**Learning multiple sparse features**

- Start from different points (can find the same max many times!)

**A better method: "deflation"**

- Find local maxima satisfying some constraints
- One possible choice: subsequent features should be un-correlated with the previously found ones (= orthogonal in the 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:** 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 is limited by the pre-processing step (PCA)

**A small philosophical problem:**

Above: **Sparseness of <u>feature</u> =** only few feature values over sample images (as a function of t)

Really want: **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!

Under these conditions,

$$\sum_{i=1}^{n} h\left((\vec{v}_i^T\vec{s})^2\right)$$

measures the sparseness of a single image, sum over sample equivalent to the formula above

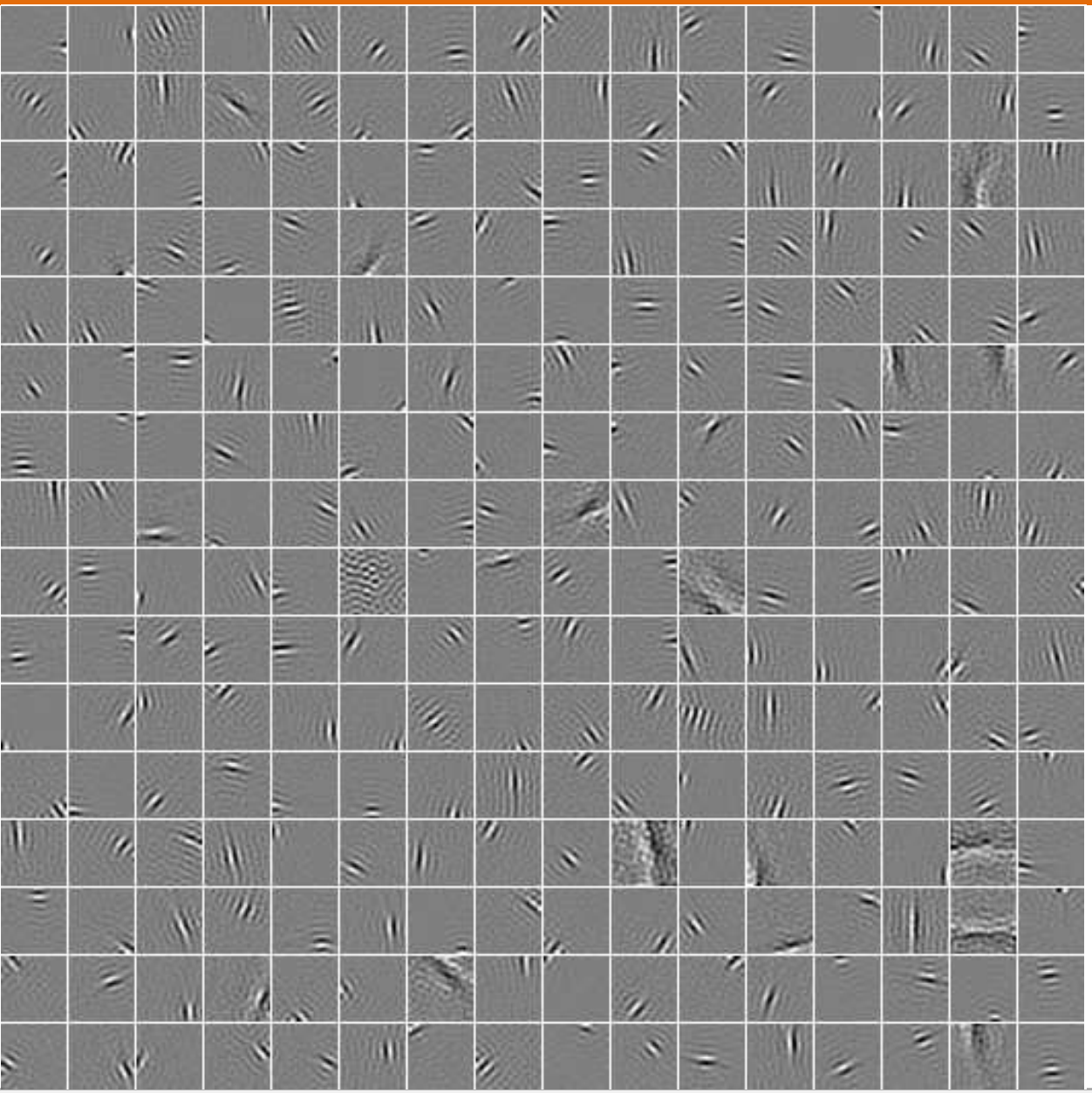
Both definitions equivalent if following conditions hold <u>for a typical image</u>:

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

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


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Sparse coding feature detectors from natural images



**Computation:**

- DB of 50000 32x32 patches
- No ordering (symmetric de-correlation)
- PCA reduction to **m = 256**
- FastICA algorithm to learn sparse features

**Features:**

- Localized in space
- Well-defined orientation
- Multi-scale (small and large)
- Similar to neural receptive fields, with distributions over frequency, phase, …
- Many entries are just shifted copies of other features

**Is that the best we can do?**


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Why is sparseness useful?

**So far:**

- A better statistical model of input data
- Efficient coding of images (many zeros after transform, easy to compress)
- Sparse features resemble receptive fields of simple cells in a biological visual system
- Some plausible explanation: neurons try to minimize their firing rates to save energy

**Still have open questions:**

- Sparseness measure was chosen ad hoc. What is the optimal function h(x)?
- Why was de-correlation (of many features) needed?
- What is the 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 is equivalent to finding maximally sparse features!

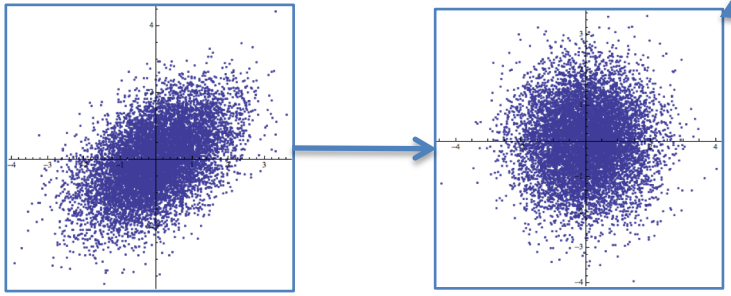
**Main idea:** assume that transformed coefficients are statistically independent over sample,

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


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Why does PCA not produce independent components?

**Gaussian distribution:** PCA and whitening

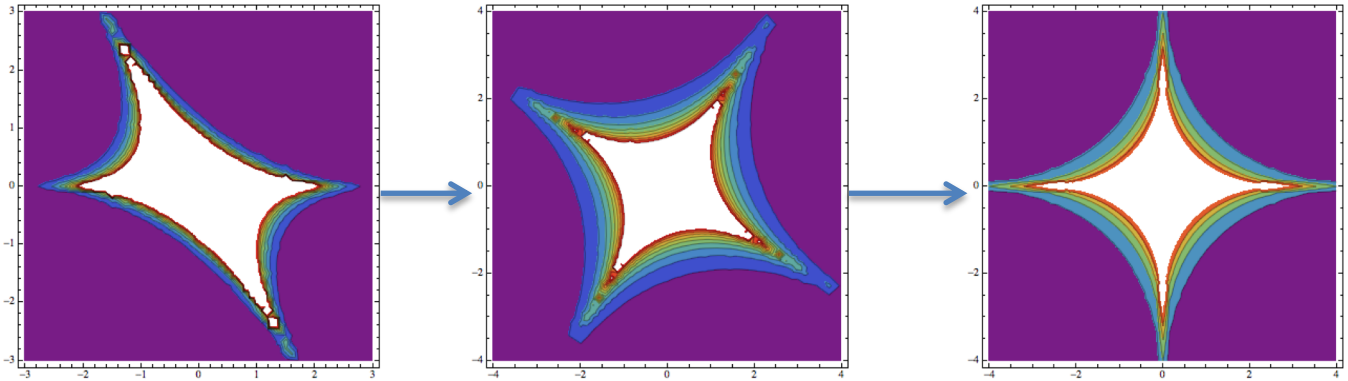


- All rotations are equivalent
- The PDF spherically symmetric
- Uncorrelated Gaussian variables are already independent!

**Using only second-order information:**

- The correlation matrix is symmetric, i.e. $n(n+1)/2$ values
- A transform matrix needs $n^2$ independent values
- Remaining DoFs: $n(n-1)/2$ rotation angles

**Sufficiently non-gaussian distribution:** PCA and whitening, followed by ICA:



**Benefits of ICA:**

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

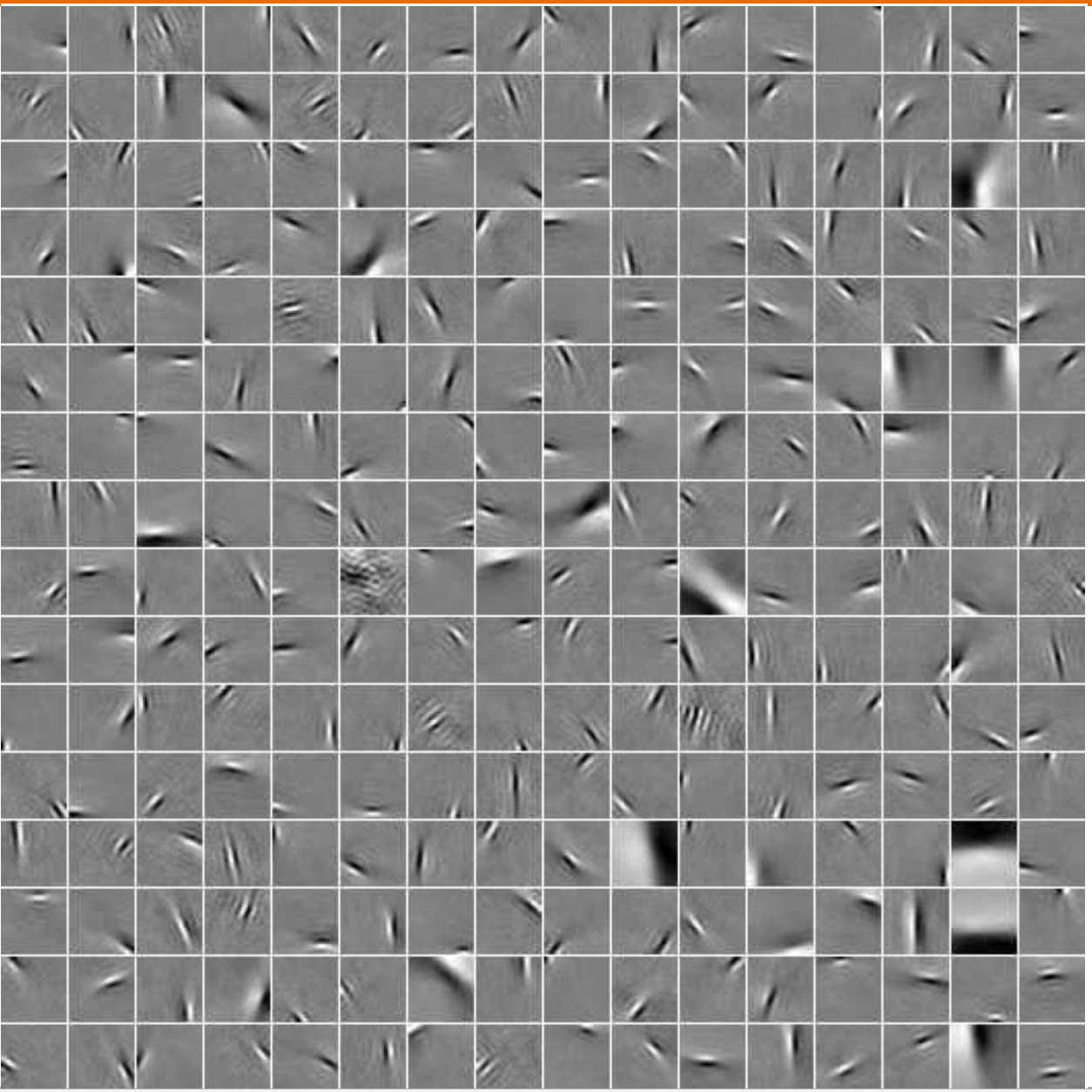
**ICA feature learning: maximum (log) likelihood estimation**

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

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

Many methods exist to find this maximum!

# ICA-based features learned from natural images



**Computation:**

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

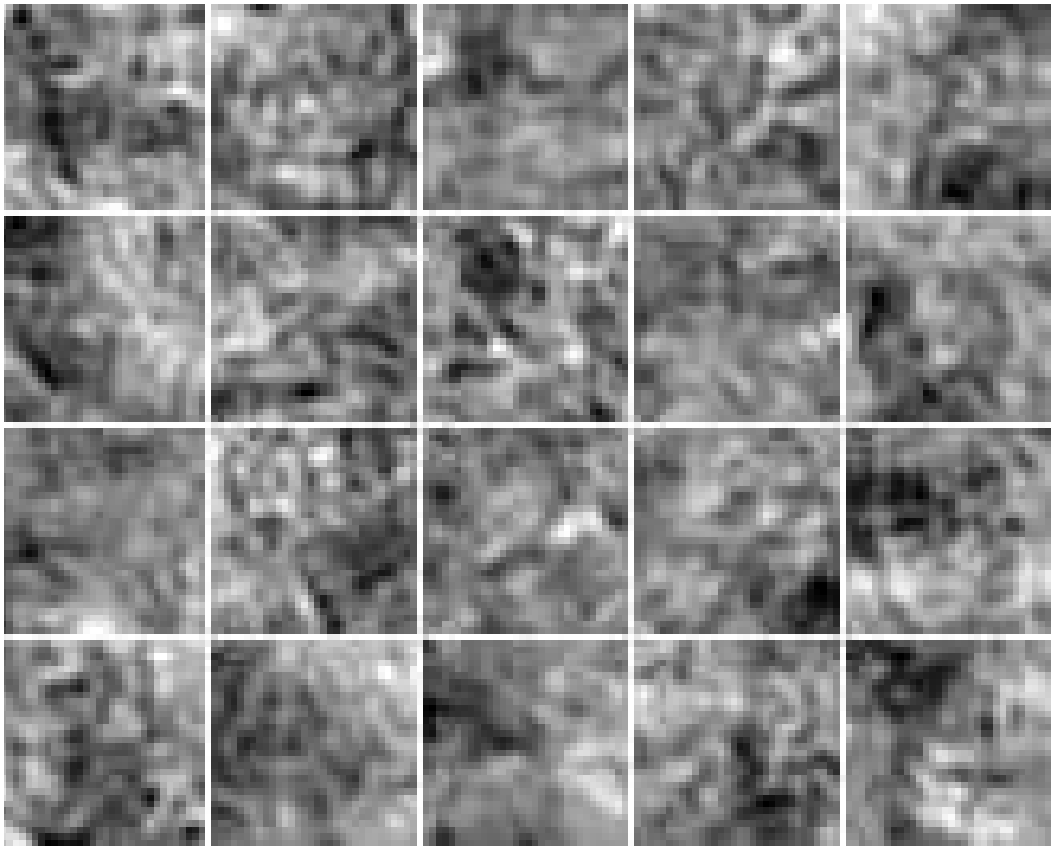
**Features:**

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

**Use as a generative model: sample "typical" natural images**


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT

# Image patches generated from the ICA model



**Computation:**

- Marginal distributions over each feature estimated from real images

**Results:**

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

**Further (current) research directions:**

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

**Still an open field, more work is required!**


© 2016-17 Alexey Pak, Lehrstuhl für Interaktive Echtzeitsysteme, Fakultät für Informatik, KIT