MIXTURES OF PRODUCT COMPONENTS

Part II: Applications of Product Mixtures

Jiří Grim

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

January 2017

Available at: http://www.utia.cas.cz/people/grim

Cf. J. Grim: "Approximation of Unknown Multivariate Probability Distributions by Using Mixtures of Product Components: A Tutorial." *International Journal of Pattern Recognition and Artificial Intelligence*, (2017).
DOI: 10.1142/S0218001417500288

ÚTIA

Outline

1. Properties of Product Mixtures
   - Multivariate Bernoulli Mixture
   - Gaussian Product Mixture
   - Structural Mixture Models
2. Application: Statistical Pattern Recognition
   - General Solution: Bayes Formula
   - Example 1: Recognition of Numerals on a Binary Raster
   - Example 2: Recognition of Chess-Board Images
   - Example 3: Classification of Text Documents
3. Application: Prediction and Data Analysis Based on Mixture Models
   - Example 4: Texture Modeling by Stepwise Prediction
   - Example 5: Search for Textural Defects and Irregularities
   - Example 6: Evaluation of Screening Mammograms
   - Example 7: Digital Forensic Analysis of Image Data
   - Example 8: Image Inpainting by Local Prediction
   - Example 9: Interactive Statistical Model of Census Data
   - Example 10: Probabilistic Neural Networks
4. Literature

PRODUCT MIXTURES - RECAPITULATION

PURPOSE: **approximation of unknown probability distributions**

Computational properties of product distribution mixtures

- efficient estimation of multivariate distribution mixtures **(!)**
- suitable to approximate multi-modal, real-life probability distributions
- with increasing number of components the Gaussian mixtures approach the asymptotic accuracy of Parzen (kernel) estimates
- unlike Parzen estimates the product mixtures are optimally "smoothed" by the efficient EM algorithm
- directly available marginal probability distributions **(!)**
- the mixture parameters can be estimated from incomplete data
- enable the information controlled sequential decision-making
- product mixtures can be interpreted as probabilistic neural networks
- enable the structural optimization of probabilistic neural networks
- provide information criterion for the optimal feature subset

▸ Literature

EM Estimation of Multivariate Bernoulli Mixtures

COMPONENTS: products of univariate Bernoulli distributions

binary data: numerals on a binary raster, results of biochemical tests ...

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \mathcal{X}, \quad x_n \in \{0,1\}, \quad \mathcal{X} = \{0,1\}^N$$

$$F(\boldsymbol{x}|m) = F(\boldsymbol{x}|\boldsymbol{\theta}_m) = \prod_{n \in \mathcal{N}} f_n(x_n|\theta_{mn}) = \prod_{n \in \mathcal{N}} \theta_{mn}^{x_n}(1-\theta_{mn})^{1-x_n}$$

$$L = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} \log[\sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)], \qquad \mathcal{S} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}\}$$

EM iteration equations:

▸ Implementation Example

$$q(m|\boldsymbol{x}) = \frac{w_m F(\boldsymbol{x}|\boldsymbol{\theta}_m)}{\sum_{j=1}^{M} w_j F(\boldsymbol{x}|\boldsymbol{\theta}_j)}, \quad \boldsymbol{x} \in \mathcal{S}, \quad m = 1, 2, \ldots, M$$

$$w_m' = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} q(m|\boldsymbol{x}), \qquad \theta_{mn}' = \frac{1}{w_m' |\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} x_n q(m|\boldsymbol{x})$$

Remark: Product of a large number of parameters θ_{mn} may underflow.

EM Estimation of Gaussian Product Mixtures

COMPONENTS: Gaussian densities with diagonal covariance matrices

$$F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \prod_{n\in\mathcal{N}} \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\left\{ - \frac{(x_n - \mu_{mn})^2}{2\sigma_{mn}^2} \right\}, \quad \boldsymbol{x} \in \mathcal{X}$$

$$L = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} \log P(\boldsymbol{x}) = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} \log[\sum_{m\in\mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m)]$$

EM iteration equations: $(m \in \mathcal{M},\ n \in \mathcal{N})$

▸ Implementation Example

$$q(m|\boldsymbol{x}) = \frac{w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m)}{\sum_{j=1}^{M} w_j F(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}, \qquad \boldsymbol{x} \in \mathcal{S},$$

$$w_m^{'} = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} q(m|\boldsymbol{x}), \qquad \mu_{mn}^{'} = \frac{1}{w_m^{'}|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} x_n q(m|\boldsymbol{x})$$

$$(\sigma_{mn}^{'})^2 = \frac{1}{w_m^{'}|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} (x_n - \mu_{mn}^{'})^2 q(m|\boldsymbol{x}) = \frac{1}{w_m^{'}|\mathcal{S}|} \sum_{\boldsymbol{x}\in\mathcal{S}} x_n^2 q(m|\boldsymbol{x}) \ - \ (\mu_{mn}^{'})^2$$

no matrix inversion $\Rightarrow$ no risk of ill-conditioned matrices

ÚTIA

Structural Mixture Model (Grim et al. 1986, 1999, 2002)

binary structural parameters: $\phi_m = (\phi_{m1}, \ldots, \phi_{mN}) \in \{0,1\}^N$

$$F(\boldsymbol{x}|m) = \prod_{n \in \mathcal{N}} f_n(x_n|m)^{\phi_{mn}} f_n(x_n|0)^{1-\phi_{mn}},$$

$f_n(x_n|0)$: **fixed "background" distributions**, usually $f_n(x_n|0) = P_n^*(x_n)$

PRINCIPLE: if $\phi_{mn} = 0$ **then** $f_n(x_n|m)$ **is replaced by** $f_n(x_n|0)$

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} F(\boldsymbol{x}|m) w_m = F(\boldsymbol{x}|0) \sum_{m \in \mathcal{M}} G(\boldsymbol{x}|m, \phi_m) w_m,$$

$$G(\boldsymbol{x}|m, \phi_m) = \prod_{n \in \mathcal{N}} \left[\frac{f_n(x_n|m)}{f_n(x_n|0)} \right]^{\phi_{mn}}, \qquad F(\boldsymbol{x}|0) = \prod_{n \in \mathcal{N}} f_n(x_n|0) > 0$$

"the background distribution" $F(\boldsymbol{x}|0)$ **reduces in the Bayes formula:**

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})} = \frac{\sum_{m \in \mathcal{M}_\omega} G(\boldsymbol{x}|m, \phi_m) w_m}{\sum_{j \in \mathcal{M}} G(\boldsymbol{x}|j, \phi_j) w_j} \approx \sum_{m \in \mathcal{M}_\omega} G(\boldsymbol{x}|m, \phi_m) w_m$$

MOTIVATION: component-specific feature selection, "dimensionless" computation, structural neural networks.

ÚTIA

Structural EM Algorithm - Discrete Mixture

$f_n(x_n|m),\ x_n \in \mathcal{X}_n,\ n \in \mathcal{N}\ \approx$ **discrete probability distributions**

$$L = \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}} \log\Big[\sum_{m\in\mathcal{M}} G(\boldsymbol{x}|m,\phi_m)w_m\Big], \qquad G(\boldsymbol{x}|m) = \prod_{n\in\mathcal{N}}\left[\frac{f_n(x_n|m)}{f_n(x_n|0)}\right]^{\phi_{mn}}$$

EM iteration equations: $(m \in \mathcal{M}, n \in \mathcal{N}, \boldsymbol{x} \in \mathcal{S})$

$$q(m|\boldsymbol{x}) = \frac{G(\boldsymbol{x}|m,\phi_m)w_m}{\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j}, \qquad w_m^{'} = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} q(m|\boldsymbol{x})$$

$$f_n^{'}(\xi|m) = \sum_{\boldsymbol{x}\in\mathcal{S}} \delta(\xi, x_n)\frac{q(m|\boldsymbol{x})}{w_m^{'}|\mathcal{S}|},$$

structural optimization: $\phi_{mn}^{'} = 1$ for the R largest values $\gamma_{mn}^{'}$:

$$\gamma_{mn}^{'} = \sum_{\boldsymbol{x}\in\mathcal{S}} \frac{q(m|\boldsymbol{x})}{w_m^{'}|\mathcal{S}|}\log\left[\frac{f_n^{'}(x_n|m)}{f_n(x_n|0)}\right] = w_m^{'}\sum_{\xi_n\in\mathcal{X}_n} f_n^{'}(\xi_n|m)\log\frac{f_n^{'}(\xi_n|m)}{f_n(\xi_n|0)}$$

Remark: **The last sum is Kullback-Leibler information divergence.**

ÚTIA

Structural EM Algorithm - Gaussian Mixture

Gaussian densities: $f_n(x_n|\mu_{mn},\sigma_{mn}) = \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\left\{ -\frac{(x_n-\mu_{mn})^2}{2\sigma_{mn}^2} \right\}$

$$L = \frac{1}{|\mathcal{S}|} \sum_{x\in\mathcal{S}} \log\Big[\sum_{m\in\mathcal{M}} w_m \prod_{n\in\mathcal{N}} \left(\frac{f_n(x_n|\mu_{mn},\sigma_{mn})}{f_n(x_n|\mu_{0n},\sigma_{0n})} \right)^{\phi_{mn}} \Big],$$

EM iteration equations: $(m\in\mathcal{M}, n\in\mathcal{N}, \boldsymbol{x}\in\mathcal{S})$

$$q(m|\boldsymbol{x}) = \frac{G(\boldsymbol{x}|m,\phi_m)w_m}{\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j)w_j}, \qquad w_m' = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} q(m|\boldsymbol{x}),$$

$$\mu_{mn}' = \frac{1}{w_m'|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} x_n q(m|\boldsymbol{x}), \quad (\sigma_{mn}')^2 = \frac{1}{w_m'|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} x_n^2 q(m|\boldsymbol{x}) - (\mu_{mn}')^2,$$

structural optimization: $\phi_{mn}' = 1$ for the R largest values γ_{mn}' :

$$\gamma_{mn}' = \frac{w_m'}{2}\left[\frac{(\mu_{mn}'-\mu_{0n})^2}{(\sigma_{0n})^2} + \frac{(\sigma_{mn}')^2}{(\sigma_{0n})^2} - \log\frac{(\sigma_{mn}')^2}{(\sigma_{0n})^2} - 1 \right] = w_m' I(f_n'(\cdot|m), f_n(\cdot|0))$$

Remark: γ_{mn}' **is the Kullback-Leibler information divergence.**

ÚTIA

Statistical Approach to Pattern Recognition

$\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathcal{X}$: N-dimensional data vectors

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_J\}$: finite set of classes with the probabilities $p(\omega)$

$P(\boldsymbol{x}|\omega), \quad \omega \in \Omega$: class-conditional distributions (estimates)

BAYES FORMULA: class-probabilities $p(\omega|\boldsymbol{x})$ given a sample $\boldsymbol{x} \in \mathcal{X}$

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})}, \qquad P(\boldsymbol{x}) = \sum_{\omega \in \Omega} P(\boldsymbol{x}|\omega)p(\omega), \quad \boldsymbol{x} \in \mathcal{X}$$

BAYES DECISION FUNCTION: minimizes the probability of error

$$d(\boldsymbol{x}) = \omega_0 = \arg\max_{\omega \in \Omega}\{p(\omega|\boldsymbol{x})\} = \arg\max_{\omega \in \Omega}\{P(\boldsymbol{x}|\omega)p(\omega)\}$$

SOLUTION: estimation of class-conditional distributions $P(\boldsymbol{x}|\omega)$ given the training data sets $\mathcal{S}_\omega = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K_\omega)}\}, \omega \in \Omega$

Remark: Product components enable local feature selection (structural models) and sequential recognition.

Example 1: Recognition of Numerals on a Binary Raster

PROBLEM: recognition hand-written numerals

Database NIST SD19 contains about 400 000 numerals scanned from a binary raster, about 40 000 for each class; it is composed of 7 comparable parts written by the personnel of the US Bureau of Census, except for the part 4 written by scholars from Bethesda (worse quality)

TRAINING AND TEST DATA:

- **odd data vectors used for training** ($\sum |\mathcal{S}_\omega| = 201485$ **numerals**)
- **even data vectors used as test set** ($\sum |\mathcal{S}_\omega^T| = 201479$ **numerals**)
- numerals normalized to raster size 32x32 (i.e. dimenion $N = 1024$)
- data extension: three rotations of each numeral (-10,-5,+5 degrees)
- $\Rightarrow$ about 80 000 training resp. test numerals for each $\omega \in \Omega$

binary numerals:

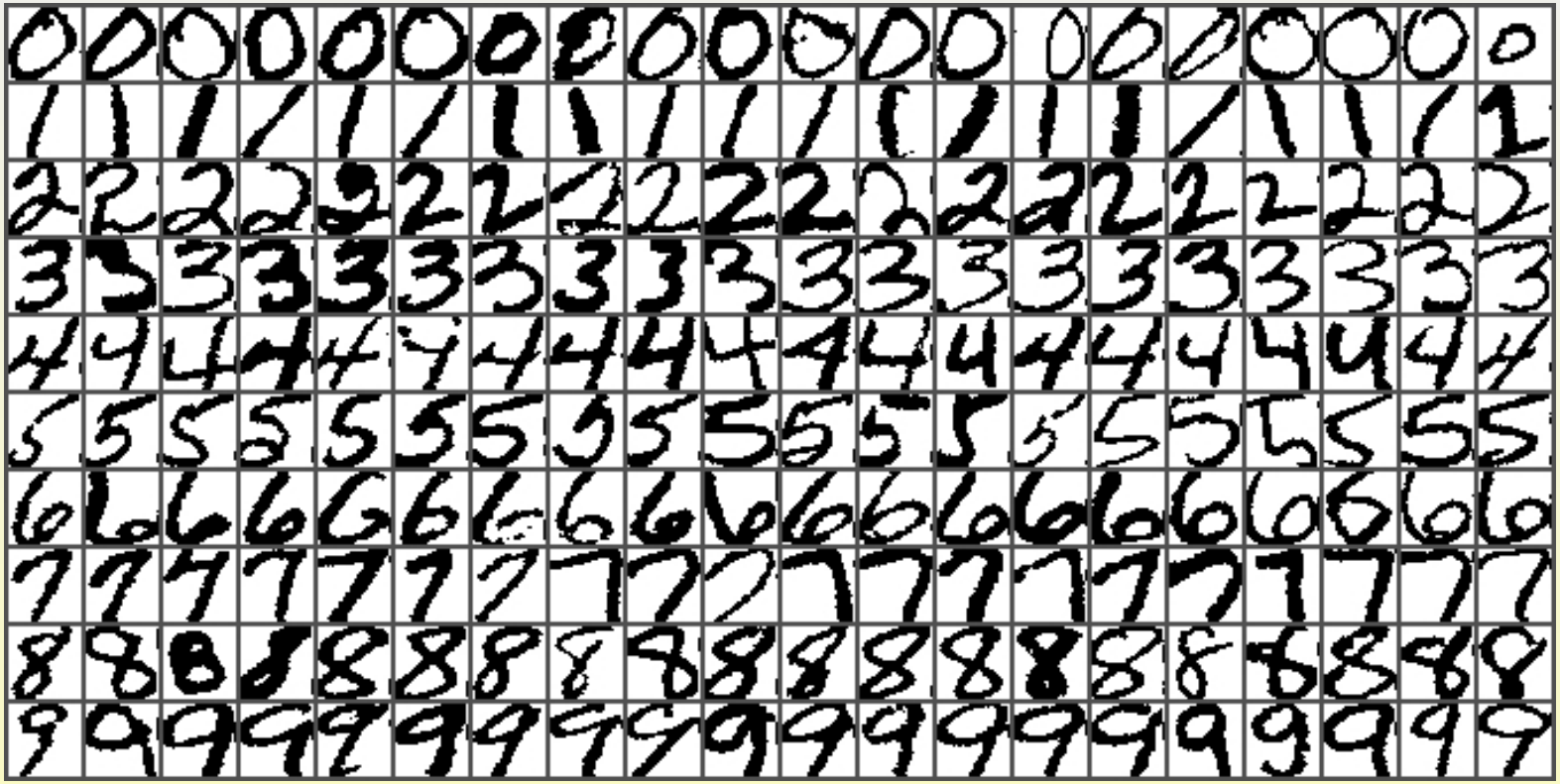
$x_n \in \{0, 1\}, \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_{1024}) \in \mathcal{X}, \quad \mathcal{X} = \{0, 1\}^{1024}$

number of classes: $|\Omega| = 10, \quad \Omega = \{\omega_0, \omega_1, \ldots, \omega_9\}$

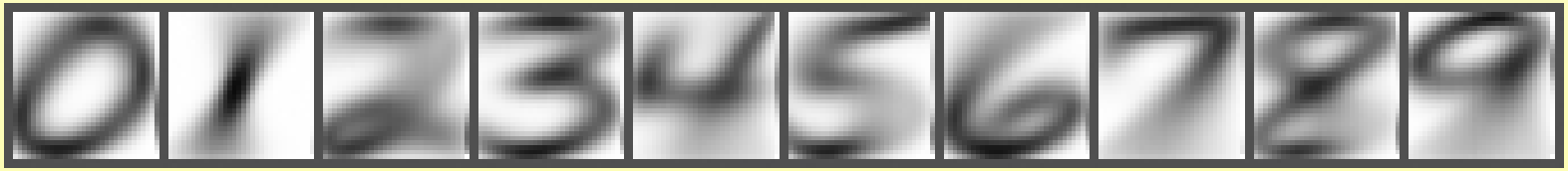
ÚTIA

Example 1: Hand-Written Numerals NIST SD19

Examples of numerals from the NIST SD19 normalized to 32x32 raster



“average numerals” (marginal probabilities of training data)



Example 1: Recognition of Numerals on a Binary Raster

SOLUTION: (Grim J., Hora J., 2010)

approximation of the class-conditional distributions $P(\boldsymbol{x}|\omega)$ by **structural Bernoulli mixtures** in the original binary space $\mathcal{X} = \{0,1\}^{1024}$

$$P(\boldsymbol{x}|\omega) = F(\boldsymbol{x}|0) \sum_{m \in \mathcal{M}_\omega} w_m \prod_{n \in \mathcal{N}} \left[\left(\frac{\theta_{mn}}{\theta_{0n}} \right)^{x_n} \left(\frac{1-\theta_{mn}}{1-\theta_{0n}} \right)^{1-x_n} \right]^{\phi_{mn}}, \quad \omega \in \Omega$$

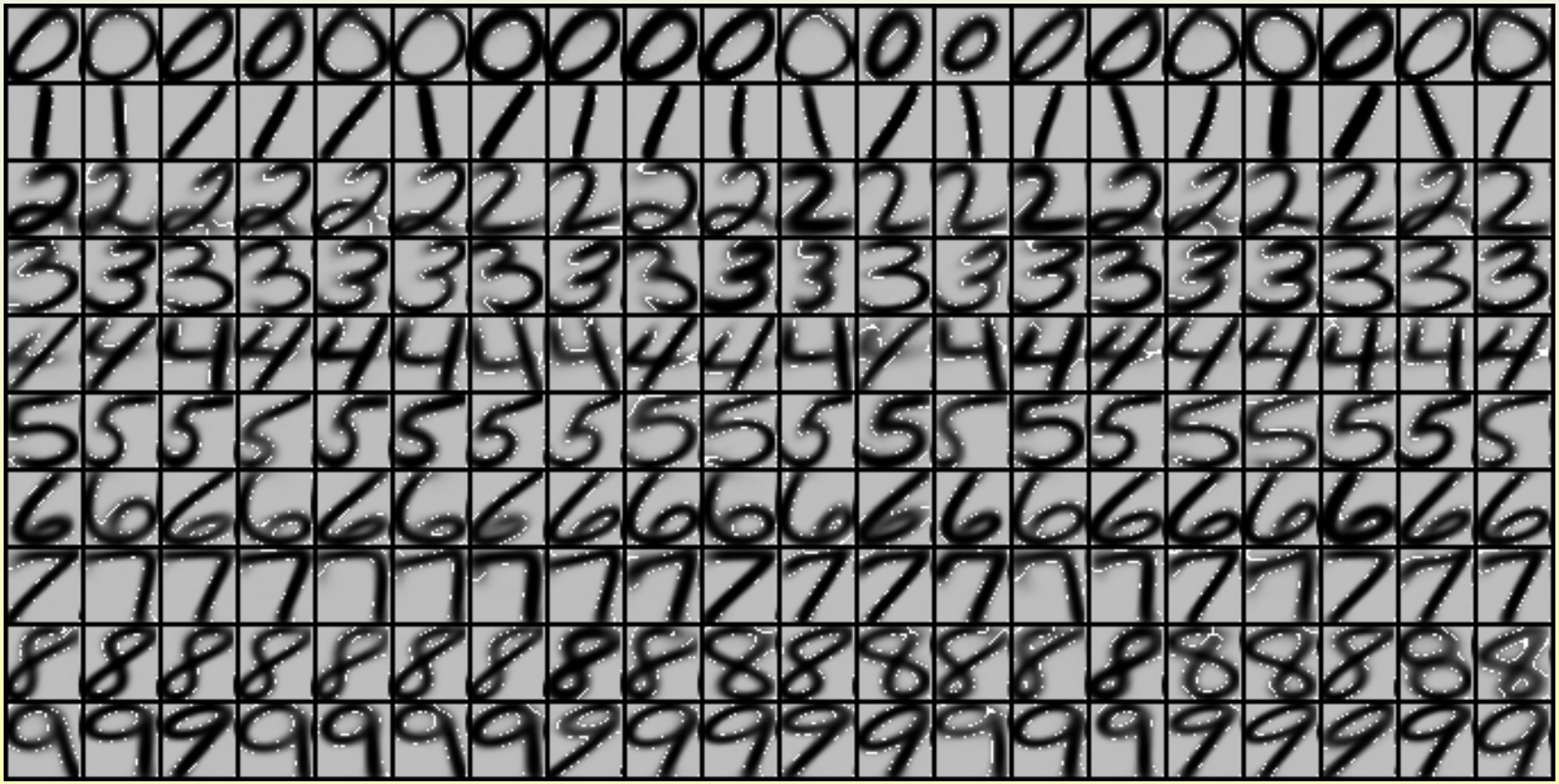
$$F(\boldsymbol{x}|0) = \prod_{n \in \mathcal{N}} \theta_{0n}^{x_n} (1-\theta_{0n})^{1-x_n}, \qquad \theta_{0n} = P\{x_n = 1\}$$

- $F(\boldsymbol{x}|0)$: fixed background distribution ($\boldsymbol{\theta}_0 \approx$ "mean" numeral)
- total number of components: $\sum_\omega |\mathcal{M}_\omega| = 1571$
- sum of non-zero structural parameters: $\sum_{m,n} \phi_{mn} = 1462373, \approx 90\%$
- random initial values: $\theta_{mn} \in \langle 0.1, 0.9 \rangle$
- stopping rule: the relative increment threshold $(L' - L)/L < 0.0001$)

Example 1: Recognition of Numerals on a Binary Raster

component parameters θ_{mn} as gray levels in raster arrangement

(white raster fields: "unused"variables specified by $\phi_{mn} = 0$)



Example 1: Recognition of Numerals on a Binary Raster

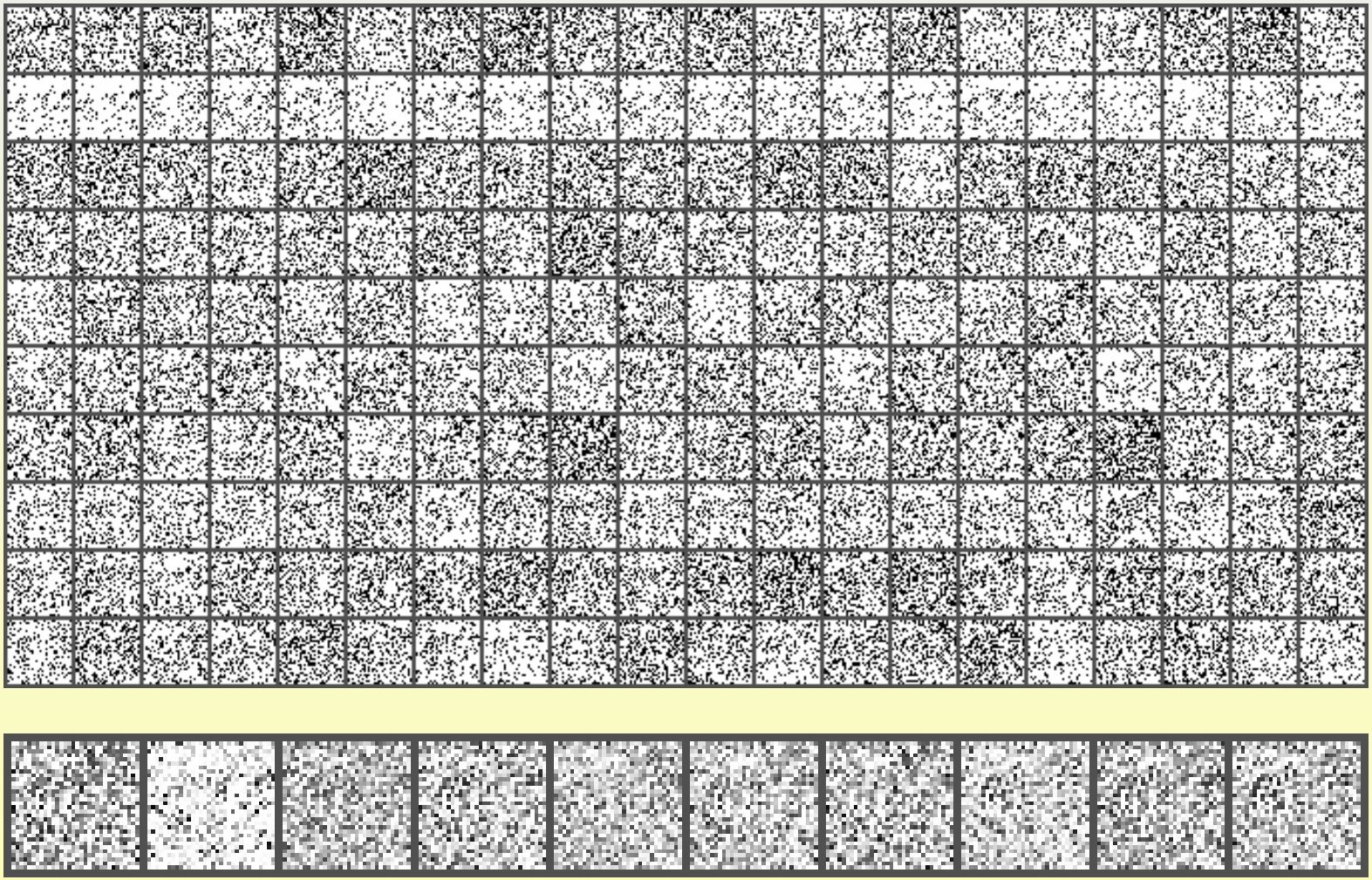
rows: classification of numerals from the given class
last column: classification error for the given class
last row: false positive classifications

CLASS	0	1	2	3	4	5	6	7	8	9	error
$\|\mathcal{S}_\omega\|$:	20182	22352	20038	20556	19577	18303	19969	20947	19790	19767	
0	19950	8	43	19	39	32	36	0	38	17	1.1%
1	2	22162	30	4	35	7	18	56	32	6	0.9%
2	32	37	19742	43	30	9	8	29	90	16	1.5%
3	20	17	62	20021	4	137	2	28	210	55	2.6%
4	11	6	19	1	19170	11	31	51	30	247	2.1%
5	25	11	9	154	4	17925	39	6	96	34	2.1%
6	63	10	17	6	23	140	19652	1	54	3	1.6%
7	7	12	73	10	73	4	0	20497	22	249	2.1%
8	22	25	53	97	30	100	11	11	19369	72	2.1%
9	15	13	25	62	114	22	3	146	93	19274	2.5%
false pos.:	197	139	537	396	352	462	148	328	665	699	1.84%

Mean classification error: 1.84%

Example 1: Example of a Raster Field Permutation

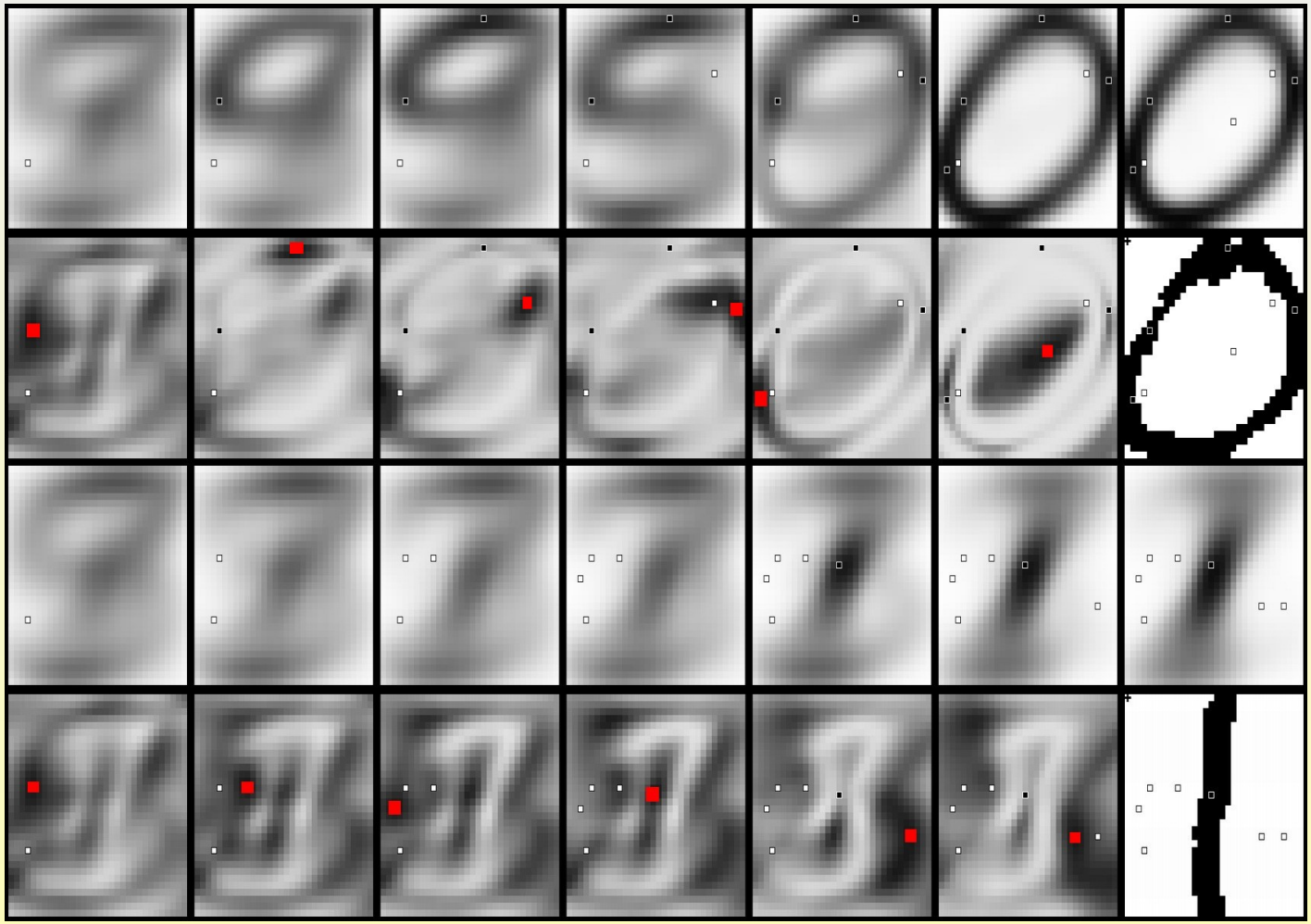
Examples of permutated numerals and related average images



Remark: Recognition accuracy of permutated numerals is identical.

Example 1: Optimal Sequential Recognition (Grim 2014)

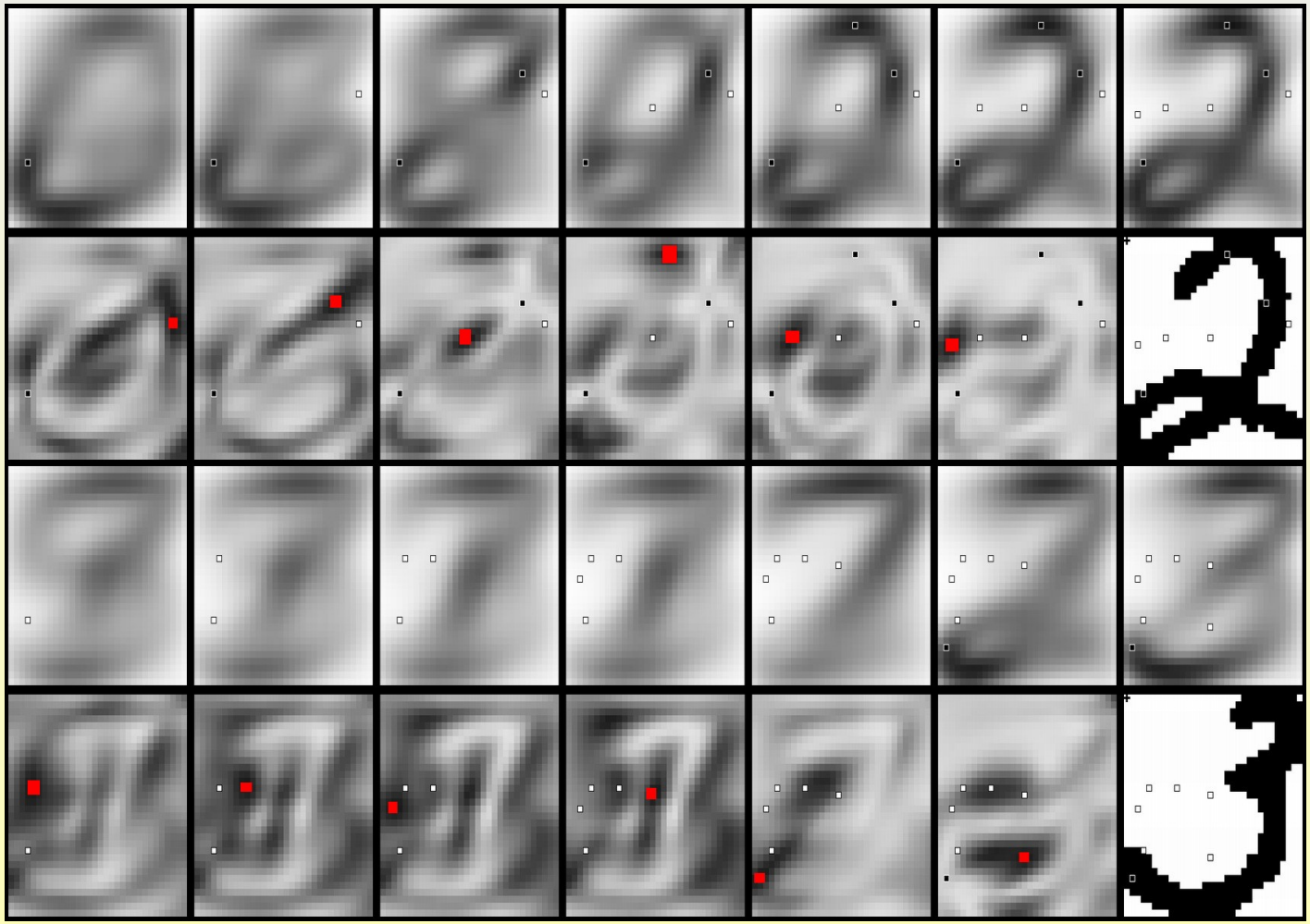
choice of the variable x_n by using conditional information $I_{x_D}(\mathcal{X}_n, \Omega)$:



Remark: Odd rows display the expected image, even rows show the raster field informativity and finally the hidden image.

Example 1: Optimal Sequential Recognition (Grim 2014)

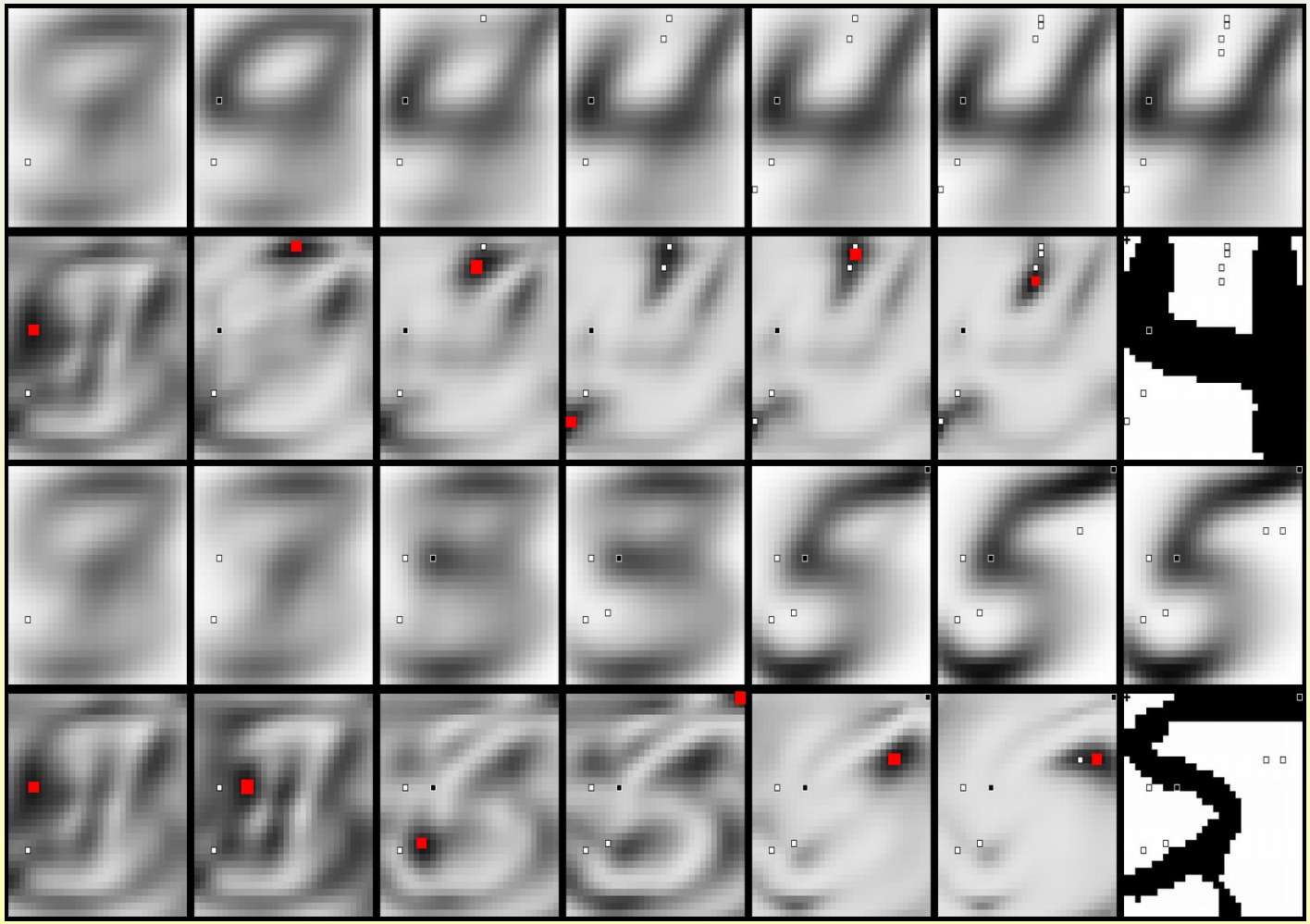
choice of the variable x_n by using conditional information $I_{x_D}(\mathcal{X}_n, \Omega)$:



Remark: The red squares denote the next optimally chosen field.

Example 1: Optimal Sequential Recognition (Grim 2014)

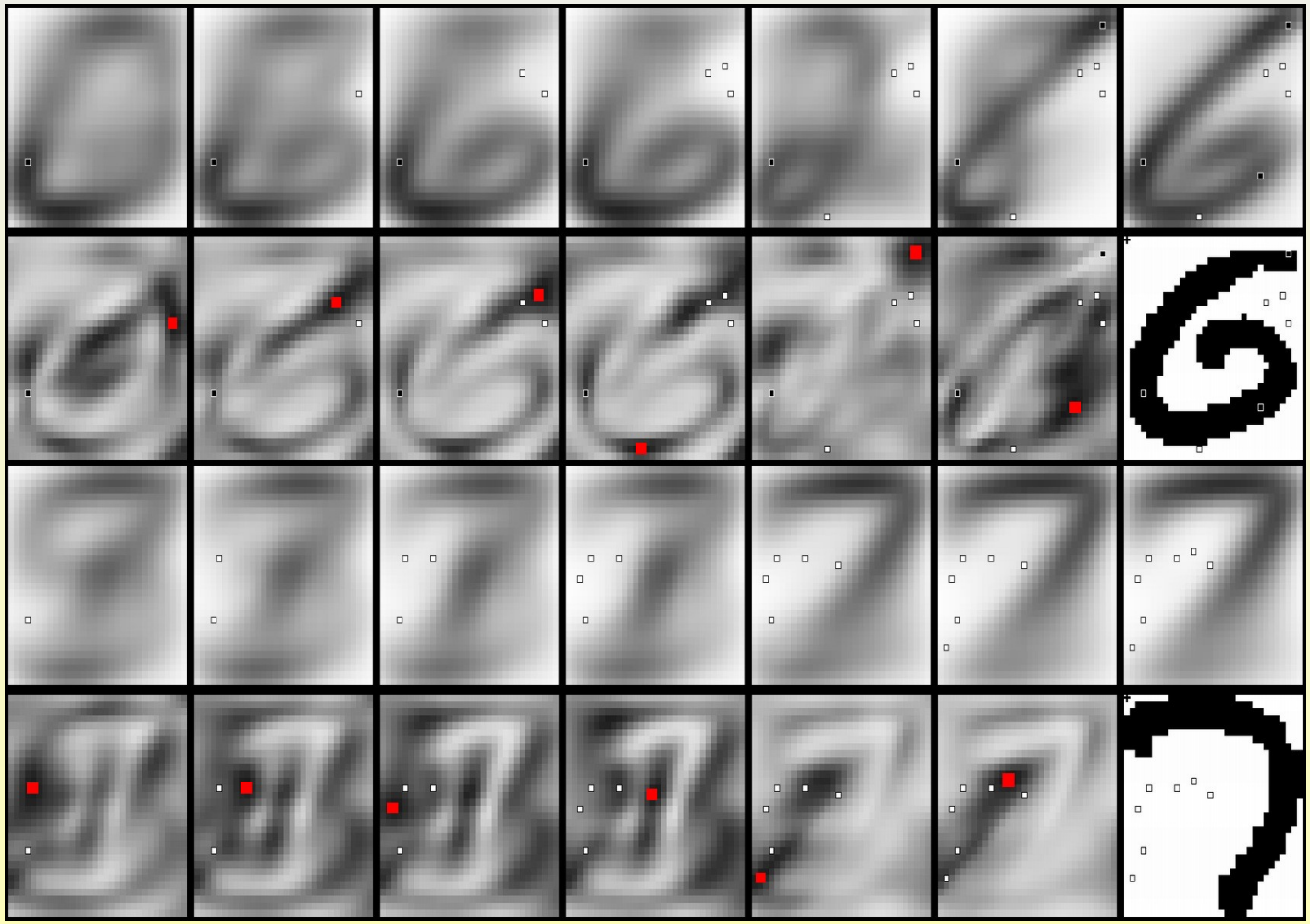
choice of the variable x_n by using conditional information $I_{x_D}(\mathcal{X}_n, \Omega)$:



Remark: The red squares denote the next optimally chosen field.

Example 1: Optimal Sequential Recognition (Grim 2014)

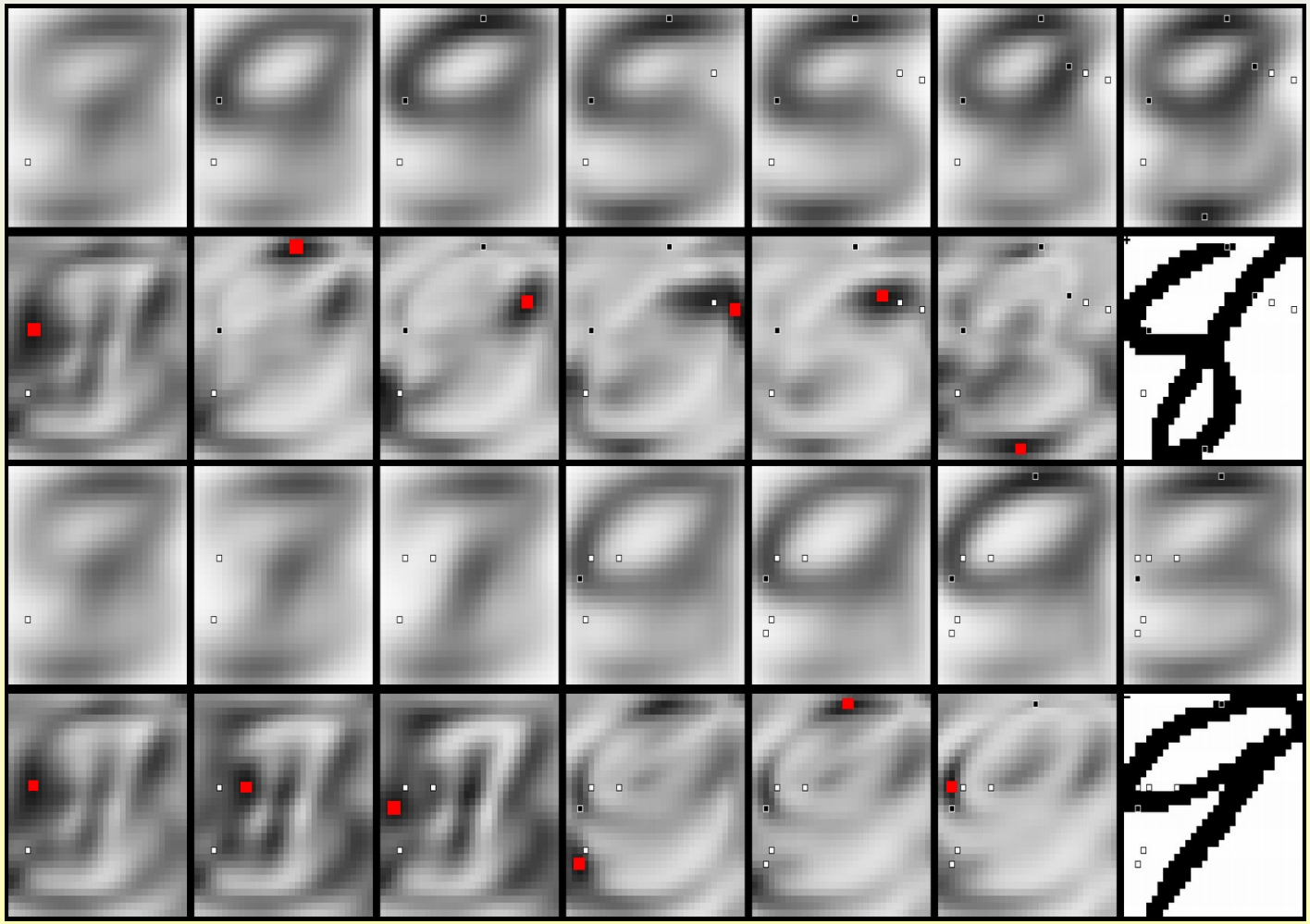
choice of the variable x_n by using conditional information $I_{x_D}(\mathcal{X}_n, \Omega)$:



Remark: Only seven pixels suffice to recognize the numeral correctly.

Example 1: Optimal Sequential Recognition (Grim 2014)

choice of the variable x_n by using conditional information $I_{\boldsymbol{x}_D}(\mathcal{X}_n, \Omega)$:



Remark: Only seven pixels suffice to recognize the numeral correctly.

Example 2: Recognition of Chess-Board Images

Problem:

recognition of two image classes generated by random moves of **rook** (class ω_1) resp. **knight** (class ω_2)

chessboard: 16 x 16 fields $\Rightarrow$ problem dimension: N = 256
number of random moves: until 10 different occupied fields

$$\boldsymbol{x} = (x_1, \ldots, x_{256}) \in \{0,1\}^{256}, \quad x_n \in \{0,1\}, \quad \sum_{n=1}^{256} x_n = 10, \quad \Omega = \{\omega_1, \omega_2\}$$

Properties of the problem:

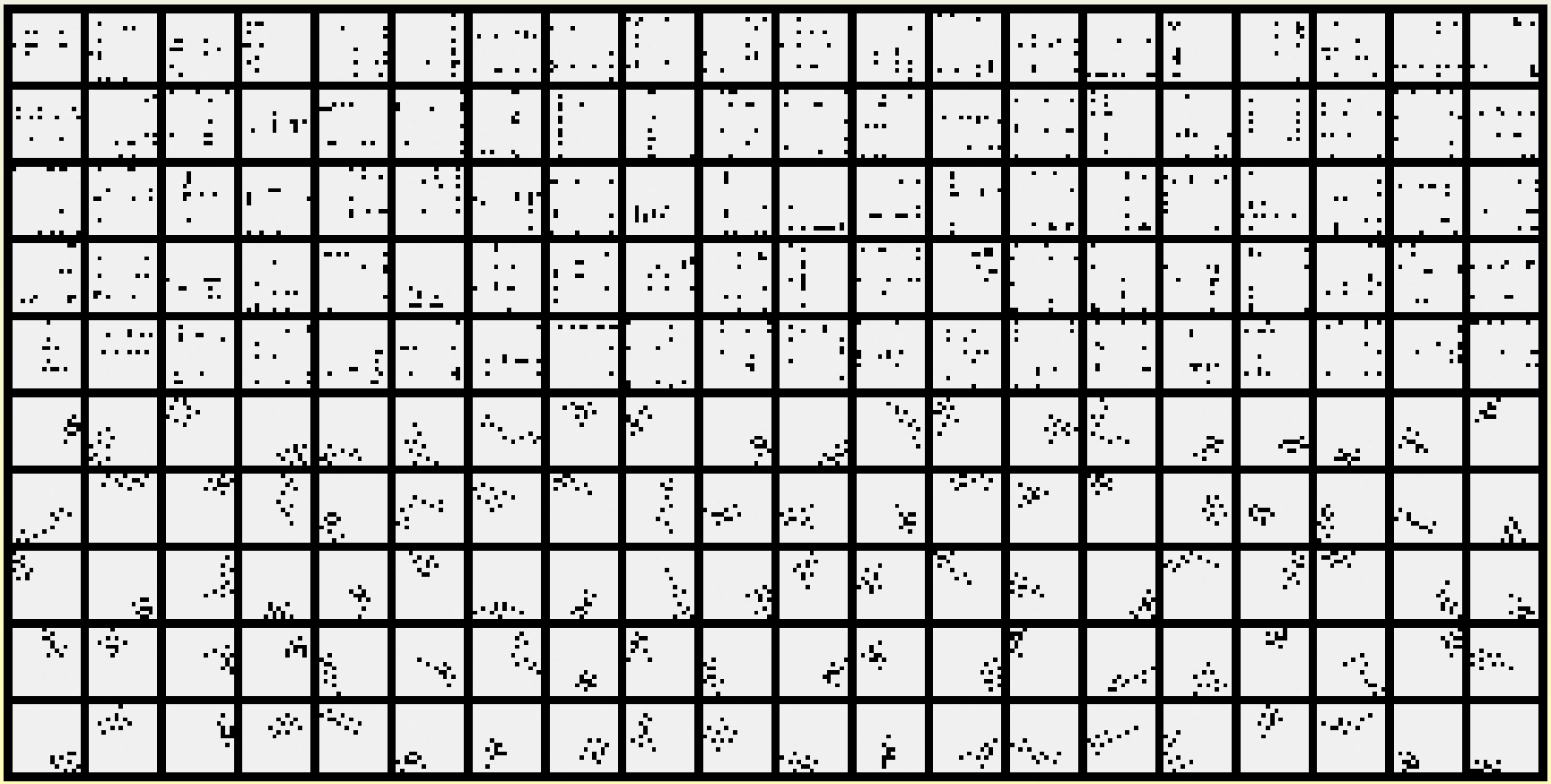
- statistically non-trivial problem, overlapping classes
- arbitrary large randomly generated training data
- arbitrary large randomly generated test data
- no simple informative features available

PROBLEM: Relation between model complexity and "overfitting".

Example 2: Recognition of Chess-Board Images

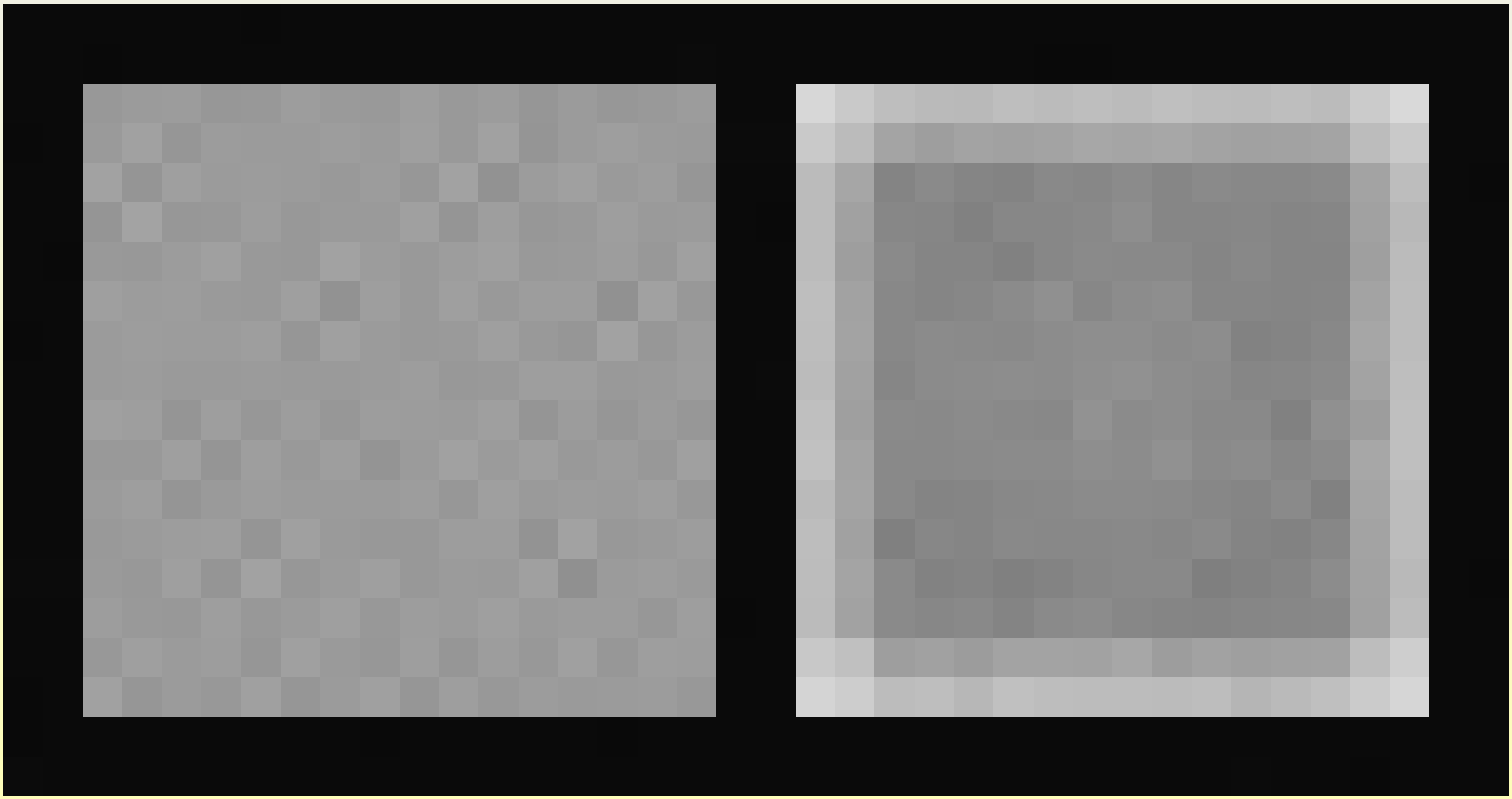
Examples of randomly generated images

class "rook" (upper part) resp. "knight" (lower part)



Example 2: Recognition of Chess-Board Images

Marginal probabilities for class "rook" (left) resp. "knight" (right) as gray-scale levels in raster arrangement



Example 2: Recognition of Chess-Board Images

SOLUTION: (Grim J., Hora J. 2010)

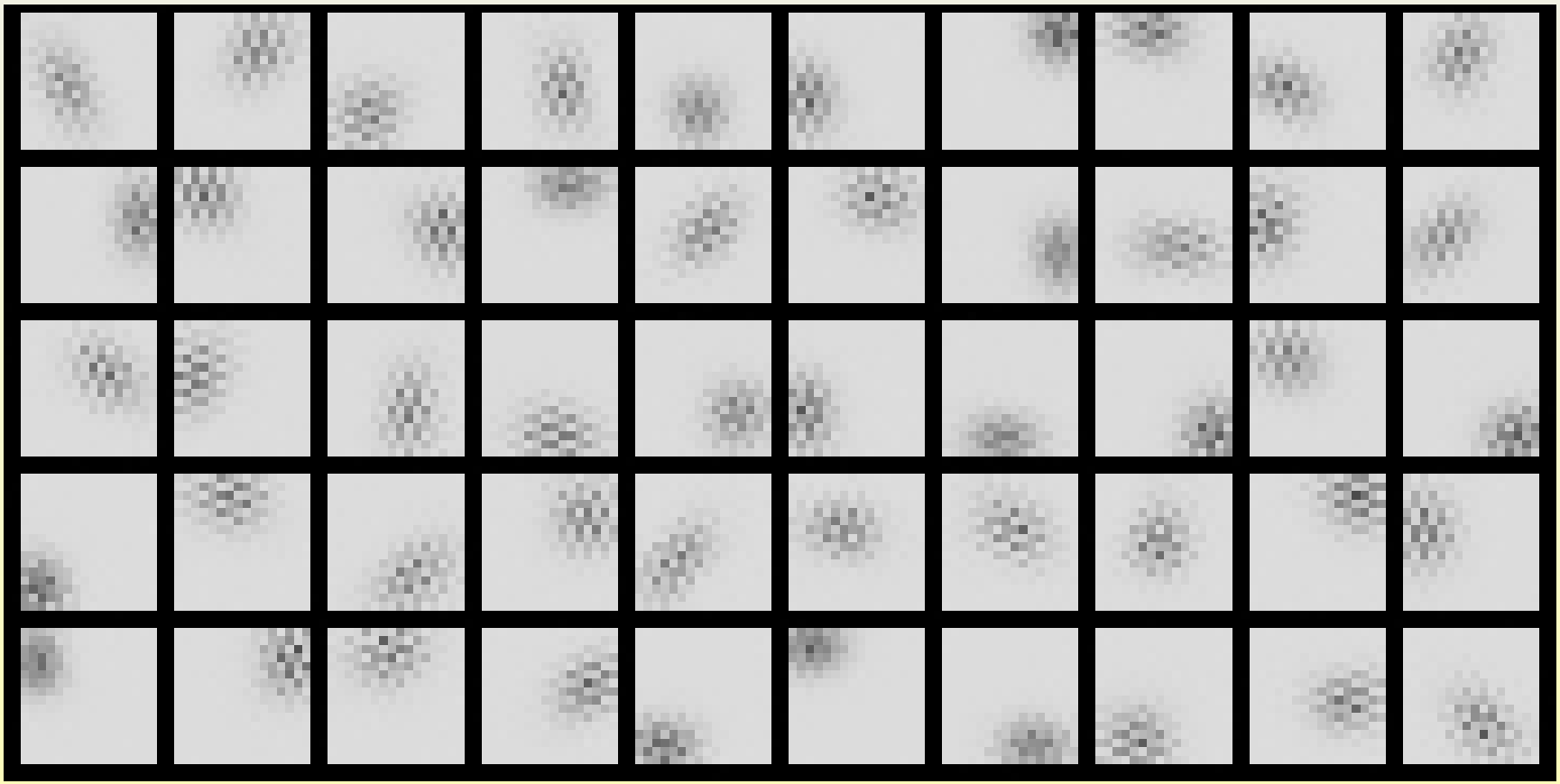
approximation of conditional distributions $P(\boldsymbol{x}|\omega_1), P(\boldsymbol{x}|\omega_2)$ by multivariate Bernoulli mixtures ($N = 256$)

$$P(\boldsymbol{x}|\omega) = \sum_{m \in \mathcal{M}_\omega} w_m \prod_{n=1}^{256} \theta_{mn}^{x_n} (1 - \theta_{mn})^{1 - x_n}, \quad x_n \in \{0, 1\}, \ \omega \in \Omega$$

- number of mixture components: $|\mathcal{M}_\omega| = 1, 2, 5, 10, 20, 50, 100, 200, 500$
- initial component weights identical: $w_m = 1/|\mathcal{M}_\omega|$
- training sample size: $|\mathcal{S}_\omega| = 1000, \ 10000, \ 100000$
- randomly initialized parameters θ_{mn} from interval $\langle 0.1, 0.9 \rangle$
- EM stopping rule: relative increment threshold $(L' - L)/L < 0.0001$

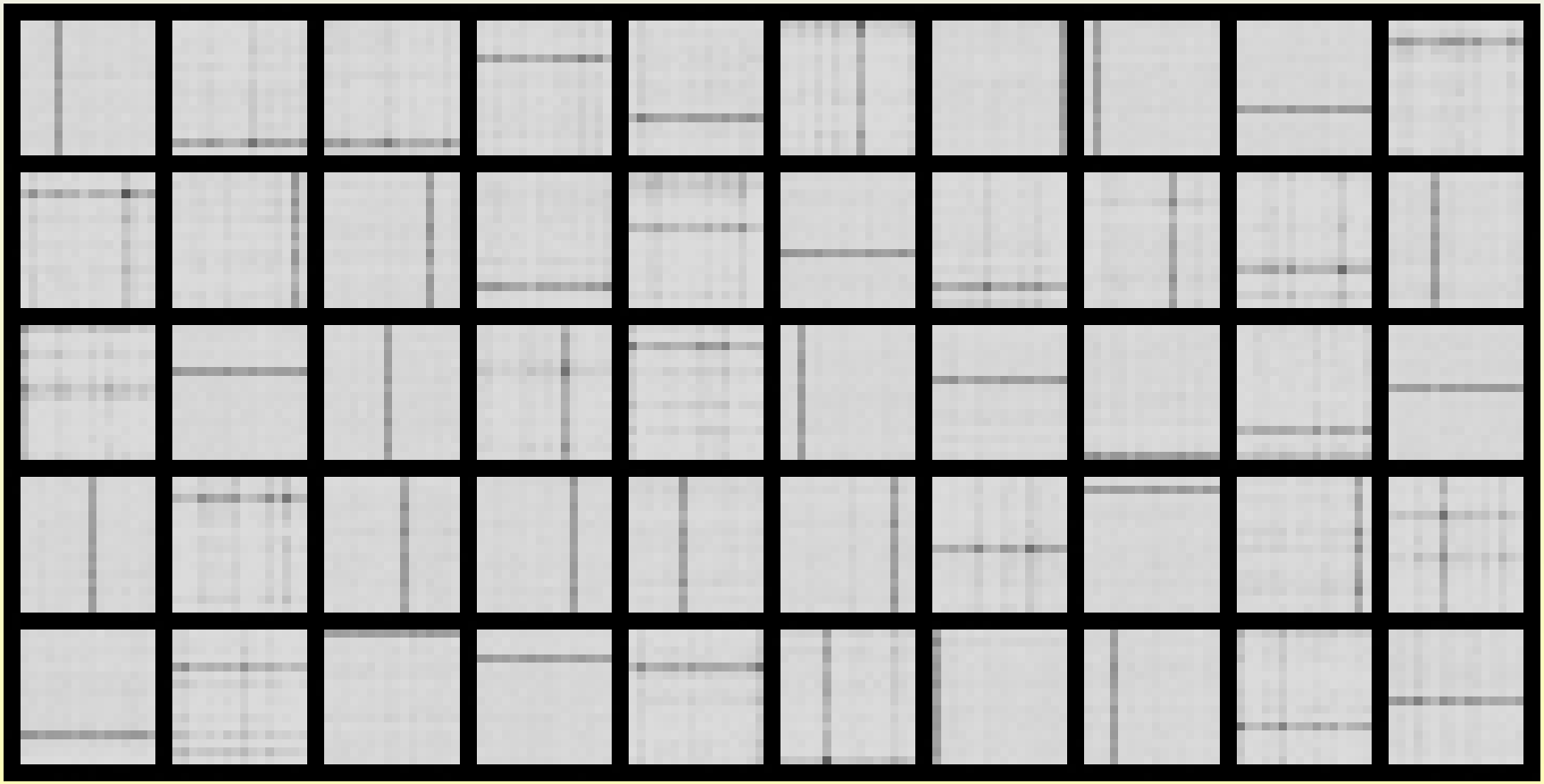
Example 2: Recognition of Chess-Board Images

Parameter estimates θ_{mn} for the class "knight" (50 components) as gray levels in raster arrangement



Example 2: Recognition of Chess-Board Images

Parameter estimates θ_{mn} for the class "rook" (50 components) as gray levels in raster arrangement



Example 2: Recognition of Chess-Board Images

RECOGNITION OF CHESS-BOARD IMAGES (error in %)

$\lvert\mathcal{M}_\omega\rvert$	**1 000**	200 000	**10 000**	200 000	**100 000**	200 000
1	**34.70**	41.56	**39.59**	40.38	**39.90**	40.02
2	**13.10**	15.83	**16.54**	16.65	**16.42**	16.48
5	**1.65**	7.72	**6.60**	7.00	**6.49**	6.60
10	**0.95**	9.21	**5.40**	5.90	**4.04**	4.34
20	**0.15**	8.76	**3.91**	4.90	**2.73**	2.89
50	**0.00**	9.35	**2.01**	4.54	**1.37**	1.90
100	**0.00**	11.02	**1.22**	5.57	**0.84**	1.68
200	**0.00**	15.40	**0.69**	8.35	**0.45**	1.92
500	**0.00**	17.77	**0.20**	14.66	**0.14**	3.76

training set (bold): $|\mathcal{S}_\omega^{train}| =$ 1000, 10 000, 100 000
independent test set: $|\mathcal{S}_\omega^{test}| =$ 200 000

Conclusion: **for a given training set there is an optimal model complexity**

ÚTIA

Example 2: Recognition of Chess-Board Images

RECOGNITION OF CHESS-BOARD IMAGES BY STRUCTURAL MIXTURE MODEL

$\lvert\mathcal{M}_\omega\rvert$	**1 000**	200 000	**10 000**	200 000	**100 000**	200 000
4	**13.80**	16.48	**25.53**	26.53	**16.91**	16.92
8	**6.45**	9.97	**10.96**	11.32	**7.28**	7.29
20	**5.70**	10.97	**5.14**	5.77	**4.70**	4.72
40	**8.65**	13.63	**4.20**	4.73	**3.29**	3.32
80	**4.20**	12.91	**6.12**	6.88	**1.91**	1.92
200	**0.25**	11.46	**3.36**	4.76	**1.83**	1.85
400	**0.00**	18.11	**3.54**	4.70	**3.10**	3.20
800	**0.00**	18.50	**3.88**	4.82	**5.42**	5.45
2000	**0.00**	18.75	**2.84**	6.39	**2.71**	2.73

Remark: The structural mixture model is less prone to "overfitting". The training set error (resubstitution, bold) is comparable with the independent test set error.

Example 3: Classification of Text Documents (Grim et al. 2008)

PROBLEM: **automatic classification of text documents**

text document:
$\boldsymbol{d} = \langle w_{i_1}, \ldots, w_{i_k} \rangle$ $\approx$ reduced to a list of terms from a vocabulary $\mathcal{V}$

vocabulary of terms: $\mathcal{V} = \{t_1, \ldots, t_N\}$ $\approx$ only informative terms
(by removing conjunctions, endings and rare terms from documents)

"bag of words" representation of documents
(defined by frequencies of vocabulary terms)

$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{d}) = (x_1, \ldots, x_N) \in \mathcal{X}$ $\approx$ vector of integers
the dimension of $\boldsymbol{x}$ is extreme $N \approx 10^4$ (!!)

x_n $\approx$ frequency of the vocabulary term t_n in the document $\boldsymbol{d}$
$|\boldsymbol{x}| = \sum_{n=1}^{N} x_n$ $\approx$ length of document $\boldsymbol{x}$

Remark: The "bag of words" representation ignores the order of words.

Example 3: Classification of Text Documents

probabilistic description:

$\mathcal{C} = \{c_1, \ldots, c_J\}$ $\approx$ set of document classes

$P(\boldsymbol{x}|c)p(c),\ c \in \mathcal{C}$ $\approx$ class-conditional distribution of documents

$p(c),\ c \in \mathcal{C}$ $\approx$ *a priori* probabilities of classes

"naive" Bayes classifier:

$$p(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)p(c)}{P(\boldsymbol{x})}, \qquad P(\boldsymbol{x}) = \sum_{c \in \mathcal{C}} p(c)P(\boldsymbol{x}|c)$$

assumes conditional independence of variables:

$$P(\boldsymbol{x}|c) = \prod_{n \in \mathcal{N}} f_n(x_n|c),\ c \in \mathcal{C}, \quad \mathcal{N} = \{1, \ldots, N\}$$

Remark: Naive Bayes classifier ignores the statistical relationship of vocabulary terms but more complex models did not improve the classification accuracy despite great effort.

Example 3: Classification of Text Documents

IDEA: approximation of $P(\boldsymbol{x}|c)$ by Poisson mixtures:

$$P(\boldsymbol{x}|c) = \sum_{m \in \mathcal{M}_c} w_m F(\boldsymbol{x}|\boldsymbol{\lambda}_m) = \sum_{m \in \mathcal{M}_c} w_m \prod_{n \in \mathcal{N}} f_n(x_n|\lambda_{mn})$$

$\lambda_{mn} \approx$ mean frequency of the term t_n in a document of length $|\boldsymbol{x}|$

component $F(\boldsymbol{x}|\boldsymbol{\lambda}_m)$ is defined as a product of Poisson distributions

probability of frequency x_n of the term t_n given the length $|\boldsymbol{x}|$:

$$f_n(x_n|\lambda_{mn}) = \frac{(\lambda_{mn})^{x_n}}{x_n!} \mathrm{e}^{-\lambda_{mn}}, \qquad (|\boldsymbol{x}| = \sum_{n=1}^{N} x_n)$$

given the length of document $|\boldsymbol{x}|$: $\Rightarrow \quad \theta_{mn} = \lambda_{mn}/|\boldsymbol{x}|$

$\theta_{mn} \approx$ **probability of occurrence of the term t_n in a document**

$$P(\boldsymbol{x}|c) = \sum_{m \in \mathcal{M}_c} F(\boldsymbol{x}|\boldsymbol{\theta}_m) w_m = \prod_{n \in \mathcal{N}} f_n(x_n|\theta_{mn}|\boldsymbol{x}|) = \prod_{n \in \mathcal{N}} \frac{(\theta_{mn}|\boldsymbol{x}|)^{x_n}}{x_n!} \mathrm{e}^{-\theta_{mn}|\boldsymbol{x}|}$$

Remark: Mixture of Poisson distributions has $M(N+1)$ parameters.
($\approx$ large number since the number of vocabulary terms N is large.)

CONCLUSION: In view of extreme dimensionality $N \approx 10^4$ the training data set is not large enough to estimate more complex models.

Example 3: Classification of Text Documents

"structural" multivariate Poisson mixtures: ▸ EM algorithm

$$P(\boldsymbol{x}|c) = \sum_{m \in \mathcal{M}_c} F(\boldsymbol{x}|\boldsymbol{\theta}_0) G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) w_m, \quad c \in \mathcal{C}$$

$F(\boldsymbol{x}|\boldsymbol{\theta}_0) \approx$ fixed "background" distribution common to all document classes

$$F(\boldsymbol{x}|\boldsymbol{\theta}_0) = \prod_{n \in \mathcal{N}} f_n(x_n|\theta_{0n}|\boldsymbol{x}|) = \prod_{n \in \mathcal{N}} \frac{(\theta_{0n}|\boldsymbol{x}|)^{x_n}}{x_n!} \mathrm{e}^{-\theta_{0n}|\boldsymbol{x}|}$$

$G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) \approx$ component functions, $\phi_{mn} \in \{0, 1\} \approx$ structural parameters

$$G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) = \prod_{n \in \mathcal{N}} \left[\frac{f_n(x_n|\theta_{mn}|\boldsymbol{x}|)}{f_n(x_n|\theta_{0n}|\boldsymbol{x}|)} \right]^{\phi_{mn}} = \prod_{n \in \mathcal{N}} \left[\left(\frac{\theta_{mn}}{\theta_{0n}} \right)^{x_n} \mathrm{e}^{(\theta_{0n} - \theta_{mn})|\boldsymbol{x}|} \right]^{\phi_{mn}}$$

"background" distribution $F(\boldsymbol{x}|\boldsymbol{\theta}_0)$ reduces in Bayes formula:

$$p(c|\boldsymbol{x}) = \frac{p(c) \sum_{m \in \mathcal{M}_c} G(\boldsymbol{x}|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m) w_m}{\sum_{c \in \mathcal{C}} p(c) \sum_{j \in \mathcal{M}_c} G(\boldsymbol{x}|\boldsymbol{\theta}_j, \boldsymbol{\phi}_j) f(j)}.$$

ÚTIA

Example 3: Classification of REUTERS Documents

text documents REUTERS:

8941 documents in **33** different classes

10105 vocabulary terms (without conjunctions, endings and rare terms)

6431 training documents, **2510** test documents
($\approx$ "APTE split" : small and multiply classified documents are omitted)

Experiment No.	**1**	**2**	**3**	**4**	**5**
# of components:	33	33	35	35	43
# of parameters:	333465	208366	285220	327184	201417
# of parameters [in %]:	100.0	62.5	80.6	92.5	46.4
Error count:	155	156	162	152	147
Mean Error [in %]:	6.17	6.21	6.45	6.07	5.86

Remark: The best classification accuracy (experiment 5) is only slightly better then the "naive" Bayes classifier error (experiment 1).

ÚTIA

Example 3: Classification of NEWSGROUPS documents

text documents "20 NEWSGROUPS":

19956 documents in **20** different comparably large classes

31826 vocabulary terms (without conjunctions, endings and rare terms)

13314 training documents, **6632** test documents
($\approx$ random partition without multiply classified documents)

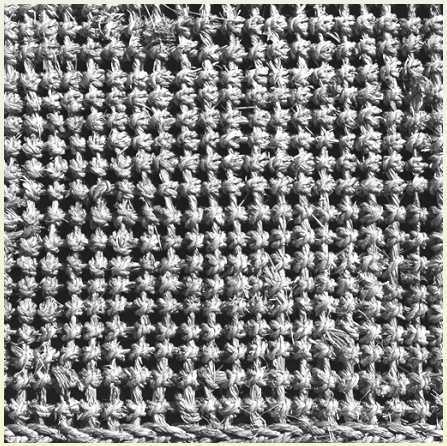
Experiment No.	**1**	**2**	**3**	**4**
# of components:	20	40	40	80
# of parameters:	636520	1204262	1102073	1024782
# of parameters [in %]:	100.0	94.6	86.6	40.2
Error count:	1406	1379	1370	1412
Mean Error [in %]:	21.20	20.79	20.66	21.29

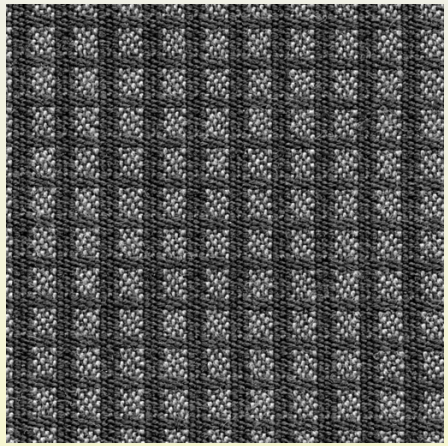
Remark: Classification errors differ in tens of documents only, the "naive" Bayes classifier error (experiment 1) is only slightly worse

Example 4: Texture Synthesis by Gaussian Mixture Models

grey-level textures: $Y = [y_{ij}]_{i=1}^{I}{}_{j=1}^{J}, \quad y_{ij} \in \{0, \ldots, 255\} \approx$ grey levels

Texture Examples: size 512×512 pixels, i.e. $I = J = 512$





Assumption of statistical "homogeneity":

We assume that textures can be described locally by means of statistical properties of internal pixels $x_1, \ldots, x_N$ of a suitably chosen sliding window.

$\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \approx$ internal pixels of a sliding window ($N \approx 10^2$)

$\mathcal{S} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}\} \approx$ data obtained by shifting the window

Remark: Windows may overlap $\Rightarrow$ data vectors $\boldsymbol{x} \in \mathcal{S}$ are not independent

ÚTIA

Example 4: Texture Synthesis by Gaussian Mixture Models

Idea of texture synthesis (Grim et al. 2003, 2004, 2005, 2006):

- estimation of statistical properties of internal pixels of a sliding window by means of a Gaussian mixture $P(\boldsymbol{x})$
- stepwise synthesis (prediction) of an arbitrary large texture image by means of conditional distributions derived from $P(\boldsymbol{x})$
- $\Rightarrow$ **a unique possibility of visual evaluation of model quality**

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \sum_{m \in \mathcal{M}} w_m \prod_{n \in \mathcal{N}} f_n(x_n|\mu_{mn}, \sigma_{mn})$$

$\mathcal{D} = \{j_1, \ldots, j_l\} \subset \mathcal{N} \quad \approx$ defined window part

$\mathcal{C} = \{i_1, \ldots, i_k\} = \mathcal{N} \setminus \mathcal{D} \quad \approx$ undefined window part

related marginal distributions:

$$\boldsymbol{x}_D = (x_{j_1}, \ldots, x_{j_l}) \in \mathcal{X}_D, \quad F(\boldsymbol{x}_D|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \Pi_{n \in D}\ f_n(x_n|\mu_{mn}, \sigma_{mn})$$

$$\boldsymbol{x}_C = (x_{i_1}, \ldots, x_{i_k}) \in \mathcal{X}_C, \quad F(\boldsymbol{x}_C|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \Pi_{n \in C}\ f_n(x_n|\mu_{mn}, \sigma_{mn})$$

ÚTIA

Example 4: Texture Synthesis by Gaussian Mixture Models

conditional distributions:

$$P_{C|D}(\boldsymbol{x}_C|\boldsymbol{x}_D) = \frac{P_{CD}(\boldsymbol{x}_C, \boldsymbol{x}_D)}{P_D(\boldsymbol{x}_D)} = \sum_{m\in\mathcal{M}} W_m(\boldsymbol{x}_D) F(\boldsymbol{x}_C|\boldsymbol{\mu}_{mC}, \boldsymbol{\sigma}_{mC})$$

$$W_m(\boldsymbol{x}_D) = \frac{w_m F(\boldsymbol{x}_D|\boldsymbol{\mu}_{mD}, \boldsymbol{\sigma}_{mD})}{\sum_{j\in\mathcal{M}} f(j) F(\boldsymbol{x}_D|\boldsymbol{\mu}_{jD}, \boldsymbol{\sigma}_{jD})} \quad \approx \quad \text{nearly binary}$$

PREDICTION: expected window part $\bar{\boldsymbol{x}}_C$ given the defined part $\boldsymbol{x}_D$:

$$\bar{\boldsymbol{x}}_C = \mathrm{E}_{C|D}\{\boldsymbol{x}_C|\boldsymbol{x}_D\} = \int \boldsymbol{x}_C P_{C|D}(\boldsymbol{x}_C|\boldsymbol{x}_D) d\boldsymbol{x}_C = \sum_{m\in\mathcal{M}} W_m(\boldsymbol{x}_D)\boldsymbol{\mu}_{mC} \approx \boldsymbol{\mu}_{m_0 C}$$

$\boldsymbol{\mu}_{m_0 C}$ $\approx$ "smoothed" tiles missing the high frequency details

SOLUTION: substitution of $\mu_{m_0 C}$ by the "most similar"parts of the original texture:

$$\boldsymbol{\mu}^*_m = \arg\min_{\boldsymbol{x}\in\mathcal{S}}\{\|\boldsymbol{\mu}_m - \boldsymbol{x}\|^2\}$$

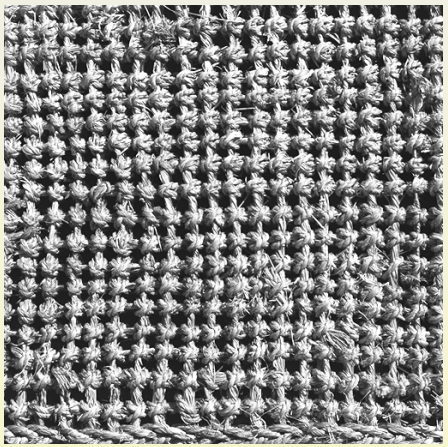
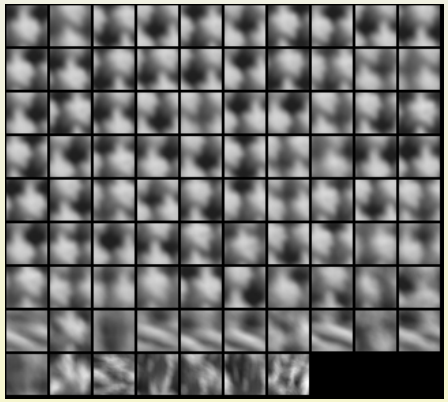
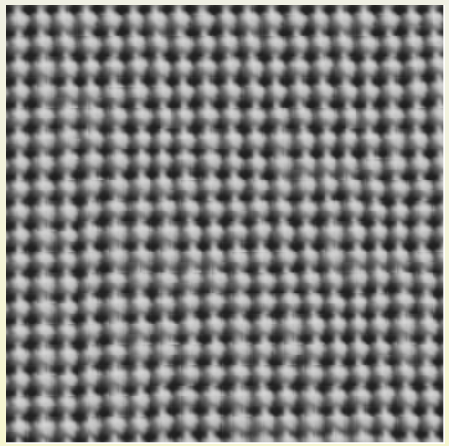
$\Rightarrow$ **"stochastic sampling"**

ÚTIA

Example 4: Texture Synthesis by Gaussian Mixture Models

synthesis of texture "ratan": prediction by component means μ_m

original texture component means μ_m prediction

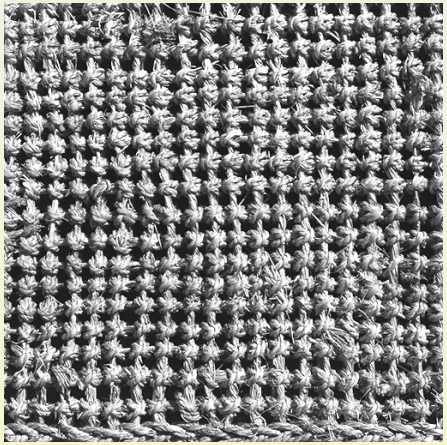
- image size: 512x512 pixels $\Rightarrow$ $|\mathcal{S}| \doteq 233000$
- sliding window size: 30x30 pixels, dimension N=900
- number of components: $|\mathcal{M}| = 80$
- number of EM iterations: $t = 15$

Remark: Component means μ_m displayed by grey-levels in window arrangement.

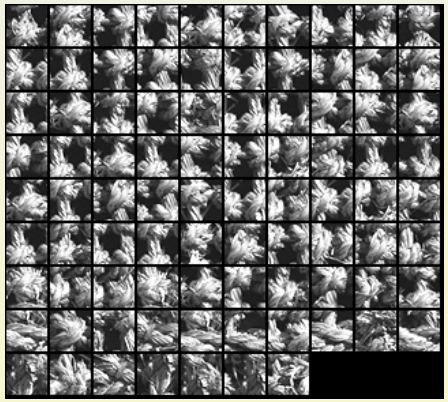
Example 4: Texture Synthesis by Gaussian Mixture Models

texture "ratan": prediction by using optimal tiles μ_m^*

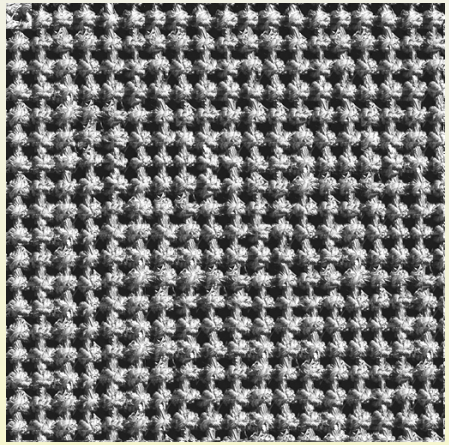
original texture



optimal "tiles"



sample synthesis



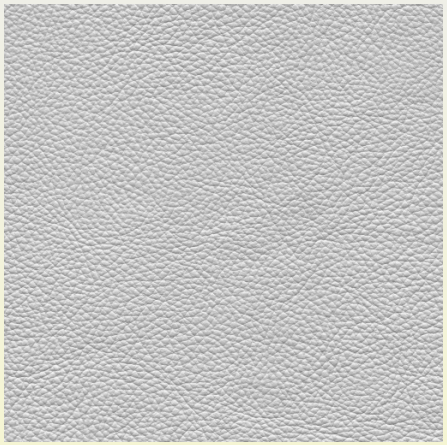
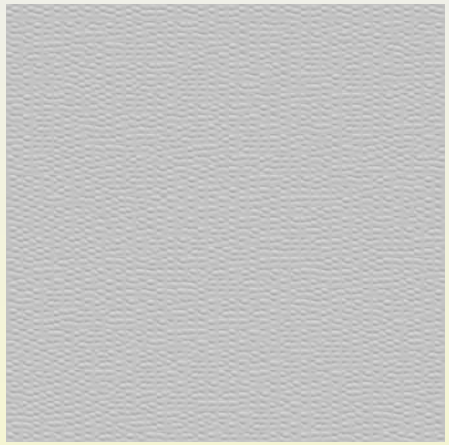
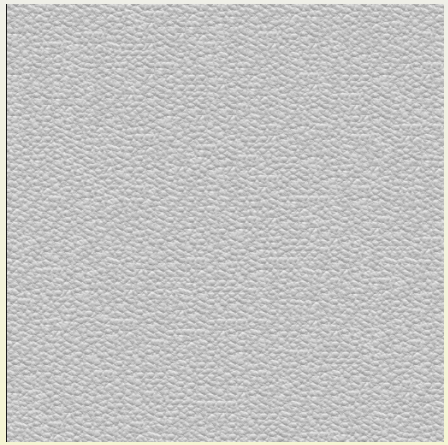
"realistic" synthesis: component means μ_m are substituted by similar pieces of the original texture μ_m^* optimally found by Eq.:

$$\mu_m^* = \arg\min_{x \in \mathcal{S}} \{\|x - \mu_m\|^2\}$$

Remark: Method of "stochastic sampling" is similar to texture synthesis based on sequential connecting of optimally found texture pieces.

Example 4: Texture Synthesis by Gaussian Mixture Models

texture "light leather": "tile" based prediction based on μ_m^*

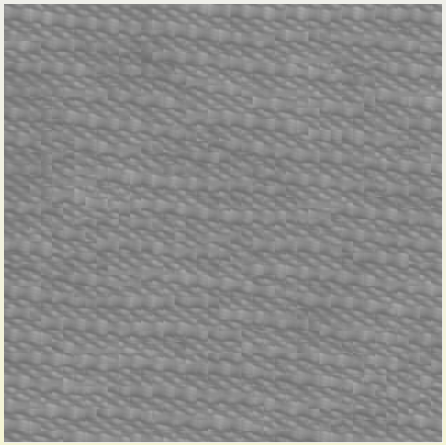
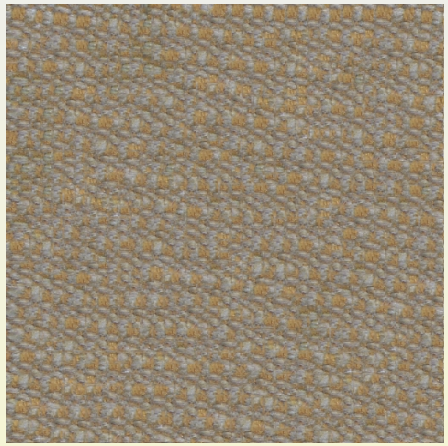
- sliding window size: 20x20 pixels, $|\mathcal{S}| \doteq 242000$ samples
- dimension: $N = 20x20 = 400$, number of components: $|\mathcal{M}| = 50$
- measure of component separation/overlap: $\bar{q}_{max} = 0.959$
- stepwise synthesis by step-size: 12 pixels

Remark: Synthesis by using a small step-size is not the best one, the optimal step-size corresponds approximately to the half of window-size.
($\approx$ Probably, the low-dimensional estimates are more reliable ?).

ÚTIA

Example 4: Texture Synthesis by Gaussian Mixture Models

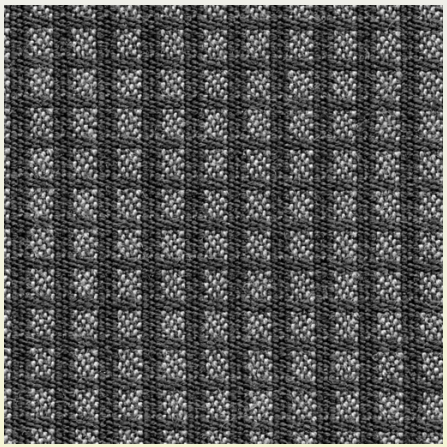
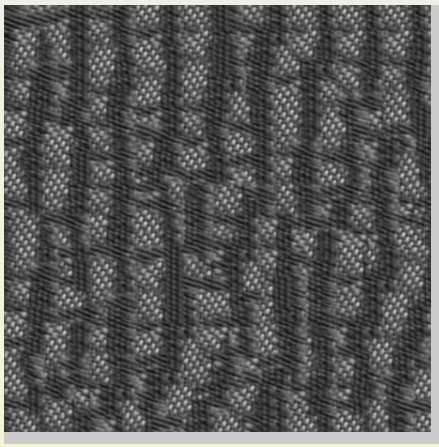
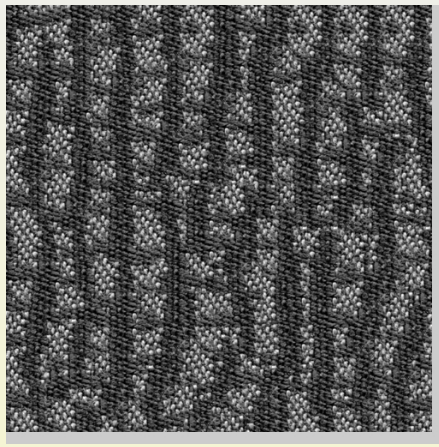
texture "rough cloth": "color tile" prediction based on μ_m^*

- sliding window size: 30*x*30 pixels
- number of samples (sliding window positions): $|\mathcal{S}| \doteq 232000$
- dimension: $N = 30x30 = 900$, number of components: $|\mathcal{M}| = 128$
- number of EM iterations: $t = 15$
- measure of component separation/overlap: $\bar{q}_{max} = 0.993$
- stepwise synthesis by step-size: 13 pixels
- component means μ_m substituted for prediction by optimally found pieces of the original color texture μ_m^*

Example 4: Texture Synthesis by Gaussian Mixture Models

texture "fabric": stochastic sampling based on the optimal "tiles"

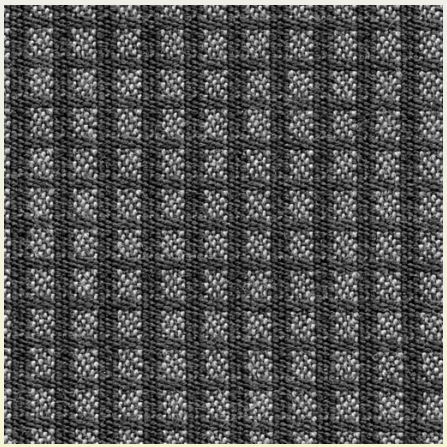
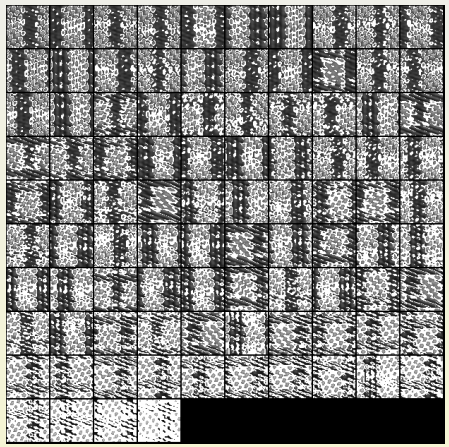
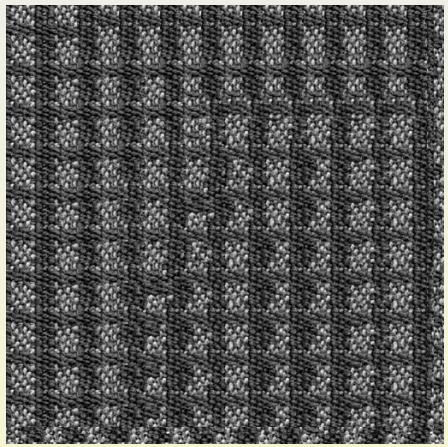




- dimension: $N = 30x30 = 900$, number of components: $|\mathcal{M}| = 90$
- number of samples (sliding window positions): $|\mathcal{S}| \doteq 232000$
- number of EM iterations: $t = 20$
- measure of component separation/overlap: $\bar{q}_{max} = 0.997$
- stepwise synthesis by step-size: 18 pixels

Remark: In case of window size $30x30$ pixels the micro-structure is described correctly but the reproduction of macro-structure fails.

Example 4: Texture Synthesis by Gaussian Mixture Models

high-dimensional structural model of the texture fabric

- **dimension:** $N = 60x60 = 3600$, number of components: $|\mathcal{M}| = 94$
- number of samples (sliding window positions): $|\mathcal{S}| \doteq 205000$
- measure of component separation/overlap: $\bar{q}_{\max} = 0.999$
- stepwise synthesis by step-size: 24 pixels
- for synthesis the white tile-pixels are replaced by "background"

Remark: The macro-structure of the texture fabric is reproduced in better quality then under small window-size $30x30$ pixels.

Example 5: Search for Textural Defects and Irregularities

gray scale texture: $\mathcal{Y} = [y_{ij}]_{i=1}^{I}{}_{j=1}^{J}, \quad y_{ij} \approx$ úrovně šedi $(\approx x_n)$

Assumption: homogeneity

Local statistical relationship between the interior window pixels should be shift-invariant.

window pixels in a given fixed order: $\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in R^N$

Method: (Grim et al. 2005)

Approximation of the probability density $P(\boldsymbol{x})$ by a Gaussian product mixture (diagonal covariance matrices in the components).

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \sum_{m \in \mathcal{M}} w_m \prod_{n \in \mathcal{N}} f_n(x_n|\mu_{mn}, \sigma_{mn})$$

$$f_n(x_n|\mu_{mn}, \sigma_{mn}) = \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\{-\frac{(x_n - \mu_{mn})^2}{2\sigma_{mn}^2}\}$$

ÚTIA

Example 5: Search for Textural Defects and Irregularities

IDEA:

In view of a successful texture synthesis by using local statistical models we assume that the mixture distribution $P(\boldsymbol{x})$ provides a sufficiently accurate statistical description of the window interior pixels and therefore the mixture model $P(\boldsymbol{x})$ can be used to evaluate the typicality of the window interior.

LOG-LIKELIHOOD: $\log P(\boldsymbol{x}) \approx$ measure of the window patch typicality
strongly depends on the gray level deviations

LOG-LIKELIHOOD RATIO: $\log P(\boldsymbol{x})/P_0(\boldsymbol{x})$
$\approx$ "structural" typicality of window patch $\boldsymbol{x}$

background distribution:

$$P_0(\boldsymbol{x}) = \prod_{n\in\mathcal{N}} f_n(x_n|\mu_{0n},\sigma_{0n}), \quad \mu_{0n} = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} x_n, \quad \sigma_{0n}^2 = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}\in\mathcal{S}} x_n^2 - \mu_{0n}^2$$

Remark: The marginal means and variances μ_{0n}, σ_{0n} are nearly identical for all image pixels $x_n, n \in \mathcal{N}$ $\Rightarrow$ the log-likelihood ratio is less sensitive to gray-level changes and is more influenced by the structural irregularities.

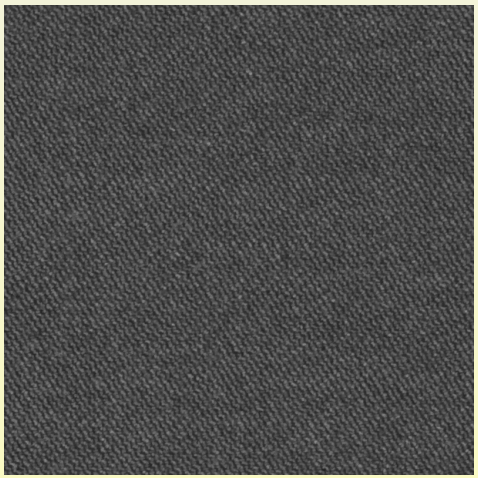
ÚTIA

Example 5: Search for Textural Defects and Irregularities

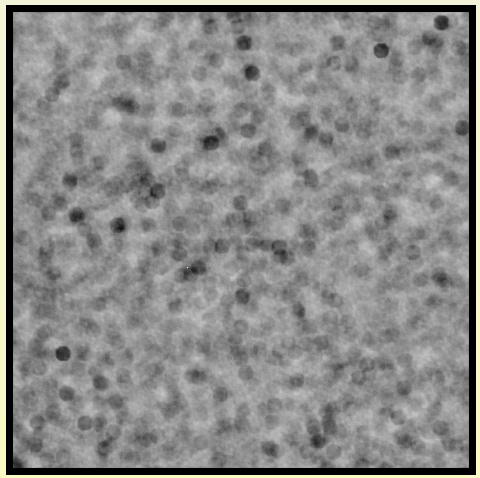
Local analysis of texture "cushion":

log-likelihood values $\log P(\boldsymbol{x})$ resp. $\log \frac{P(x)}{P_0(x)}$ displayed as gray levels at the central pixel of the sliding window

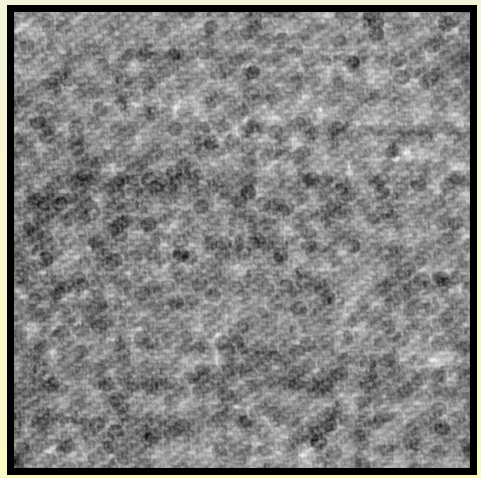
original image



L-likelihood



LR-likelihood



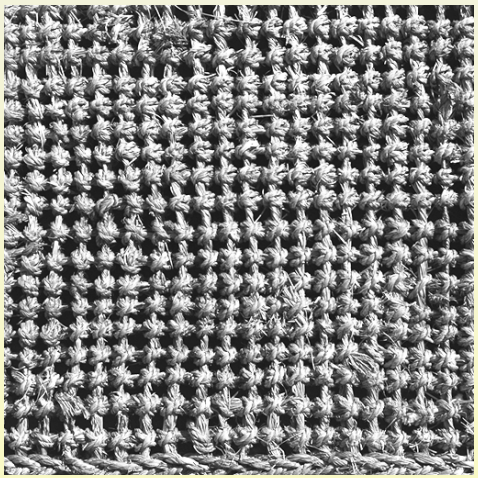
Remark: The log-likelihood values $\log P(\boldsymbol{x})$ are rather sensitive to gray-level changes. Thus, the hardly visible light pixels in the texture cushion (original image) are indicated as dark patches of window shape (medium image).

Example 5: Search for Textural Defects and Irregularities

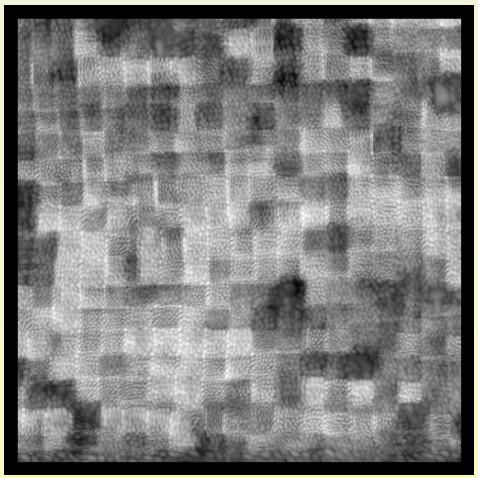
Local analysis of texture "ratan":

log-likelihood values $\log P(\boldsymbol{x})$ resp. $\log \frac{P(x)}{P_0(x)}$ displayed as gray levels at the central pixel of the sliding window:

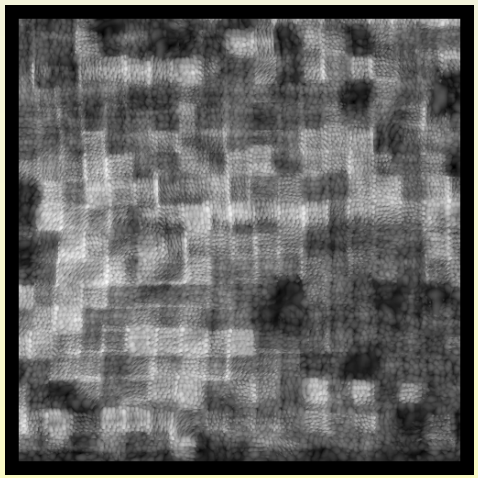
original image



L-likelihood



LR-likelihood



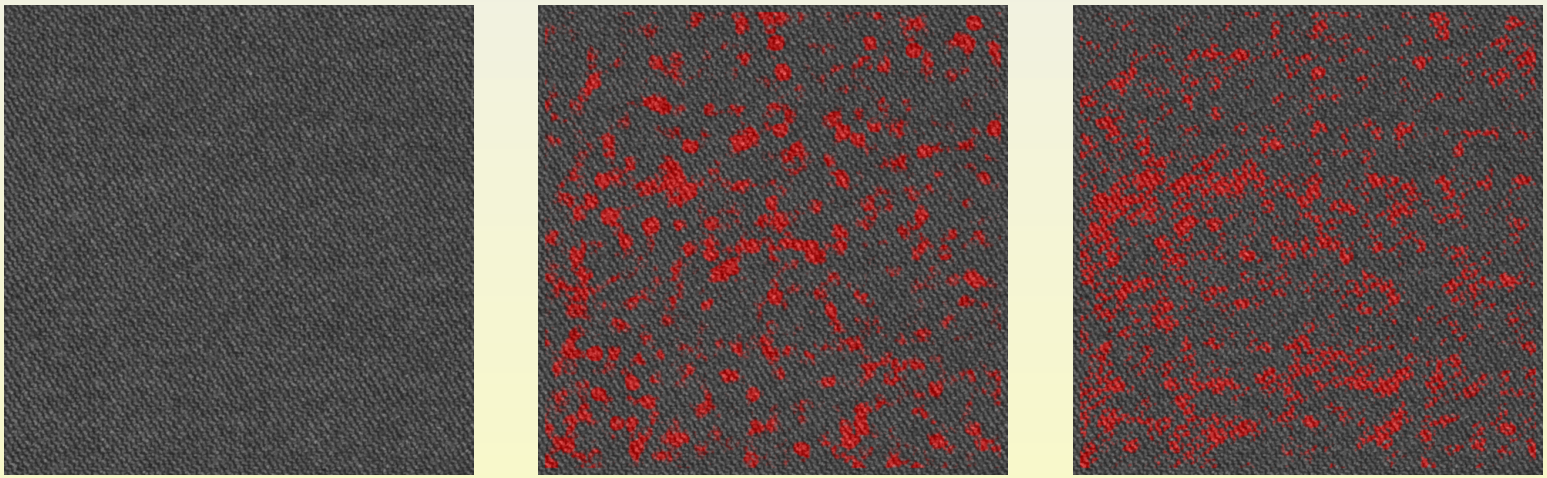
Remark: The log-likelihood ratio values $\log(P(\boldsymbol{x})/P_0(\boldsymbol{x}))$ are sensitive to structural deviations and less influenced by gray levels. Thus the irregularities of "ratan" (left) are better indicated on the right-hand image based on the log-likelihood ratio $\log(P(\boldsymbol{x})/P_0(\boldsymbol{x}))$.

ÚTIA

Example 5: Search for Textural Defects and Irregularities

Irregularity analysis of texture "cushion":

original image | L-likelihood | LR-likelihood



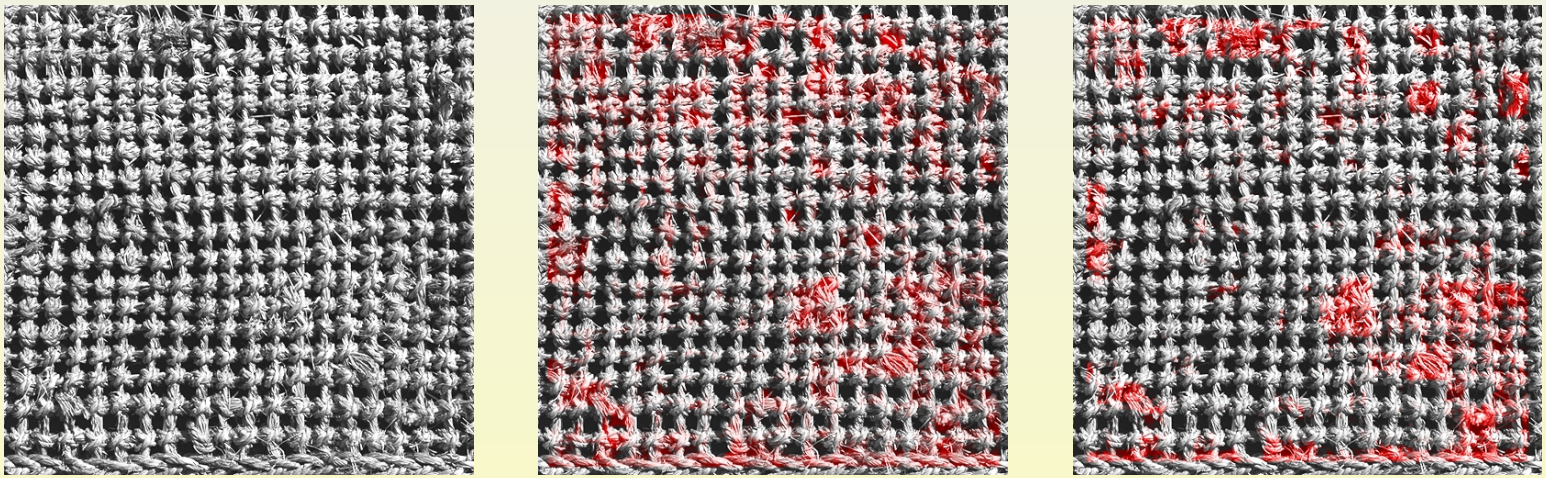
red color intensity: $\approx$ unusual (atypical) locations (defects) in texture

- medium image: $\approx$ low likelihood values: $\log P(\boldsymbol{x})$
- image on the right : $\approx$ low likelihood ratio values: $\log \frac{P(\boldsymbol{x})}{P_0(\boldsymbol{x})}$

Example 5: Search for Textural Defects and Irregularities

Irregularity analysis of texture "ratan":

original image | L-likelihood | LR-likelihood



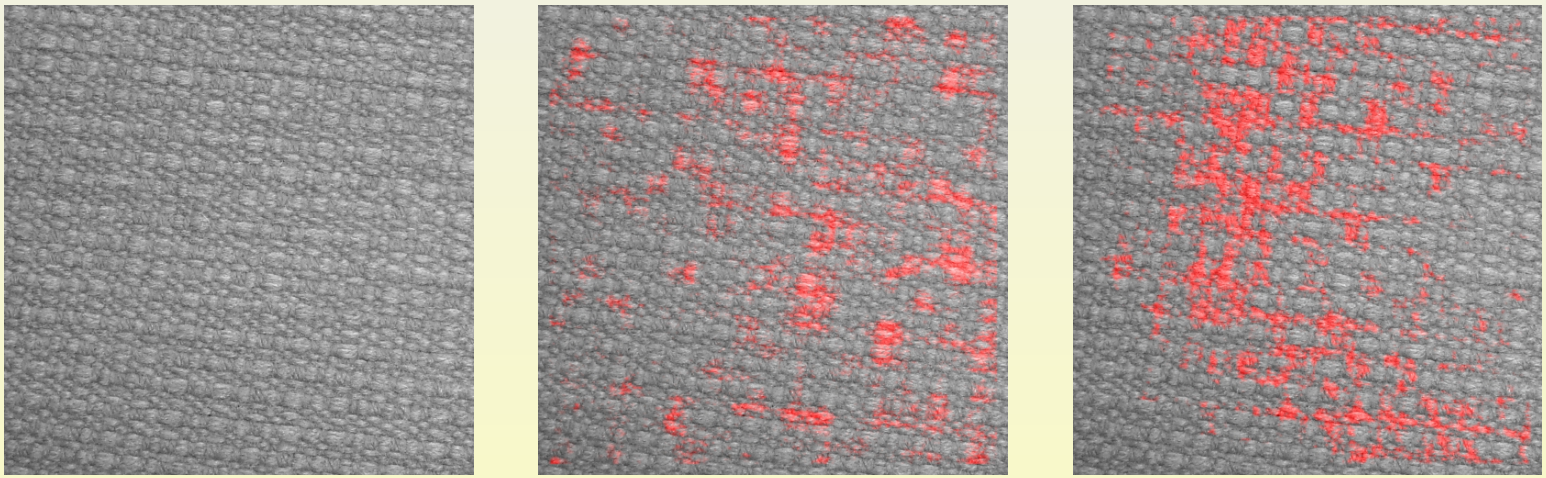
red color intensity: $\approx$ unusual (atypical) locations (defects) in texture

- medium image: $\approx$ low likelihood values: $\log P(\boldsymbol{x})$
- image on the right : $\approx$ low likelihood ratio values: $\log \frac{P(\boldsymbol{x})}{P_0(\boldsymbol{x})}$

Example 5: Search for Textural Defects and Irregularities

Irregularity analysis of texture "cloth":

original image | L-likelihood | LR-likelihood



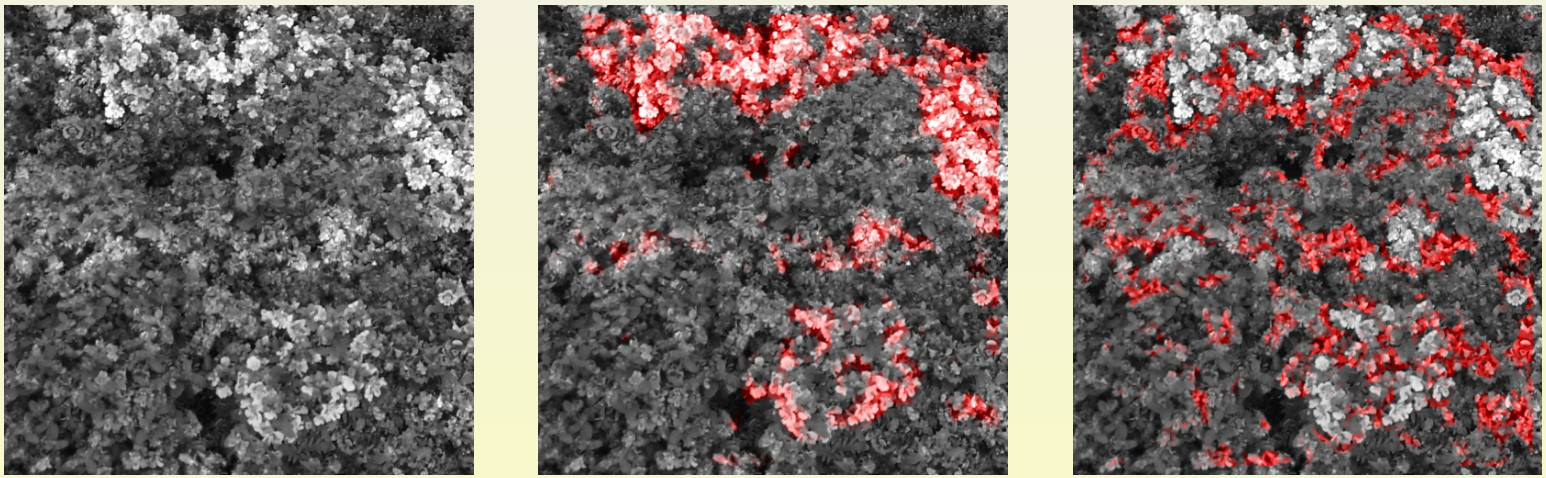
red color intensity: $\approx$ unusual (atypical) locations (defects) in texture

- medium image: $\approx$ low likelihood values: $\log P(\boldsymbol{x})$
- image on the right : $\approx$ low likelihood ratio values: $\log \frac{P(\boldsymbol{x})}{P_0(\boldsymbol{x})}$

Example 5: Search for Textural Defects and Irregularities

Irregularity analysis of texture "flowers":

original image | L-likelihood | LR-likelihood



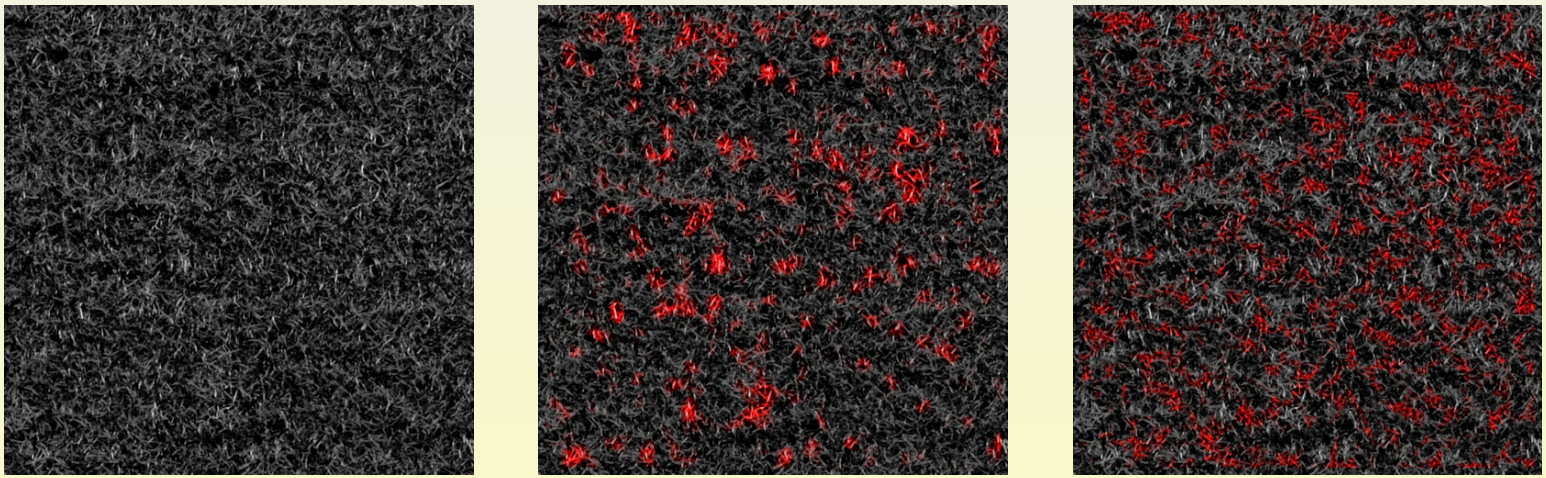
red color intensity: $\approx$ unusual (atypical) locations (defects) in texture

- medium image: $\approx$ low likelihood values: $\log P(\boldsymbol{x})$
- image on the right : $\approx$ low likelihood ratio values: $\log \frac{P(\boldsymbol{x})}{P_0(\boldsymbol{x})}$

Example 5: Search for Textural Defects and Irregularities

Irregularity analysis of texture "carpet":

original image | L-likelihood | LR-likelihood



red color intensity: $\approx$ unusual (atypical) locations (defects) in texturee

- medium image: $\approx$ low likelihood values: $\log P(\boldsymbol{x})$
- image on the right : $\approx$ low likelihood ratio values: $\log \frac{P(\boldsymbol{x})}{P_0(\boldsymbol{x})}$

Example 5: Texture Analysis by Local Statistical Models

- data vectors $\boldsymbol{x} \in \mathcal{S}$ generated by shifted window may overlap **(!)** and therefore they are not independent
- $\Rightarrow$ **basic principle of the maximum-likelihood estimates is violated**
- the data set $\mathcal{S}$ corresponds to a "trajectory" in the space $\mathcal{X}$ produced by the shifting window ($\Rightarrow$ it is not representative)
- **unlike other problems (recognition, texture synthesis, prediction) the estimated mixture $P(\boldsymbol{x})$ is applied only to the original data $\mathcal{S}$,** $\Rightarrow$ there is no risk of "overfitting"
- the log-likelihood criterion optimally "fits" the estimated mixture $P(\boldsymbol{x})$ to the original data set $\mathcal{S}$
- $\Rightarrow$ the estimated mixture $P(\boldsymbol{x})$ is well applicable to the original data $\boldsymbol{x} \in \mathcal{S}$ in view of the underlying estimation method
- $\Rightarrow$ the likelihood value $\log P(\boldsymbol{x})$ is well applicable as a measure of "typicality" of the vectors $\boldsymbol{x} \in \mathcal{S}$
- the missing independence of the vectors $\boldsymbol{x} \in \mathcal{S}$ is less relevant because the estimated mixture $P(\boldsymbol{x})$ is not applied to independent data

Example 6: Evaluation of Screening Mammograms

Statistical Data of Breast Cancer:

- breast cancer happens to about 8% of women during their lifetime
- occurrence of malignant findings in screening mammograms is only about 1 to 3 of 1000
- 5 to 10% of findings is proposed for surgical verification by biopsy
- about 60 to 80% of biopsies result in benign diagnoses ($\Rightarrow$ unnecessary physical trauma and emotional stress)
- retrospective examinations report about 10 to 20% false negative results of screening mammogram evaluation
- total number of screening mammograms evaluated worldwide in one year may be of order of millions

Meaning of Mammographic Screening:

early detection of malignant abnormalities by mammographic screening is the only effective tool to decrease the breast cancer mortality rates

Example 6: Evaluation of Screening Mammograms

Aim of Log-Likelihood Evaluation:

to emphasize mammographic lesions and facilitate diagnostic evaluation of screening mammograms

LOG-LIKELIHOOD IMAGE:

$\log P(\boldsymbol{x}) \approx$ measure of typicality of the window patch $\boldsymbol{x}$

Remark: low values of $\log P(\boldsymbol{x})$ displayed as dark pixels should indicate less-probable "unusual" or "suspect" locations of mammogram

LOG-LIKELIHOOD RATIO:

$\log P(\boldsymbol{x})/P_0(\boldsymbol{x}) \approx$ not applied

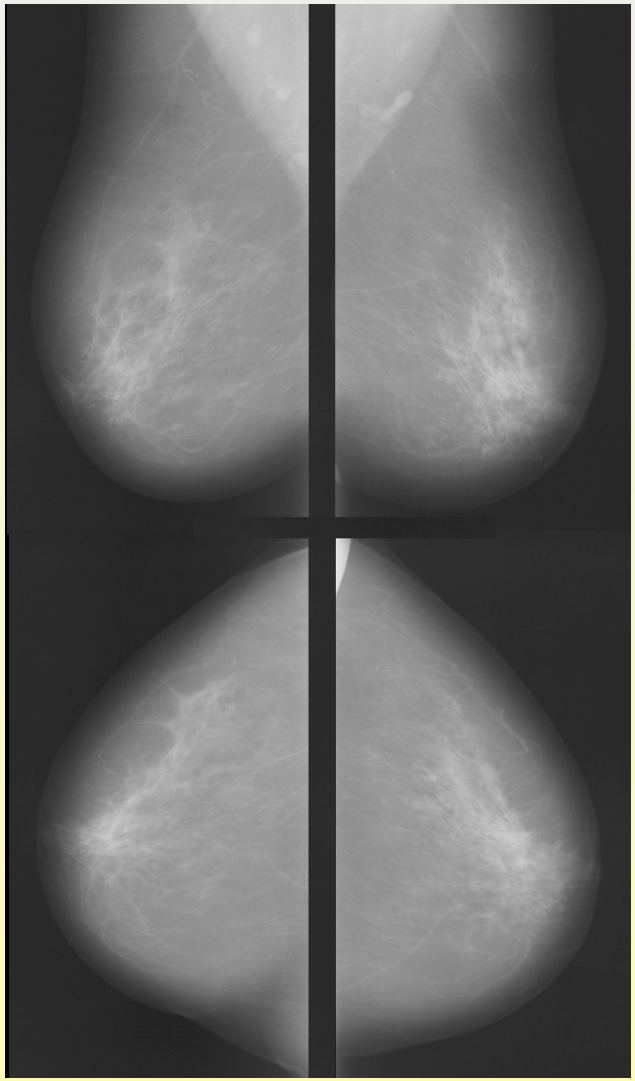
Remark: Log-Likelihood Ratio Image is not applied to screening mammograms because the grey-level deviations have diagnostic meaning and should not be suppressed.

Example 6: Computational Details of Evaluation

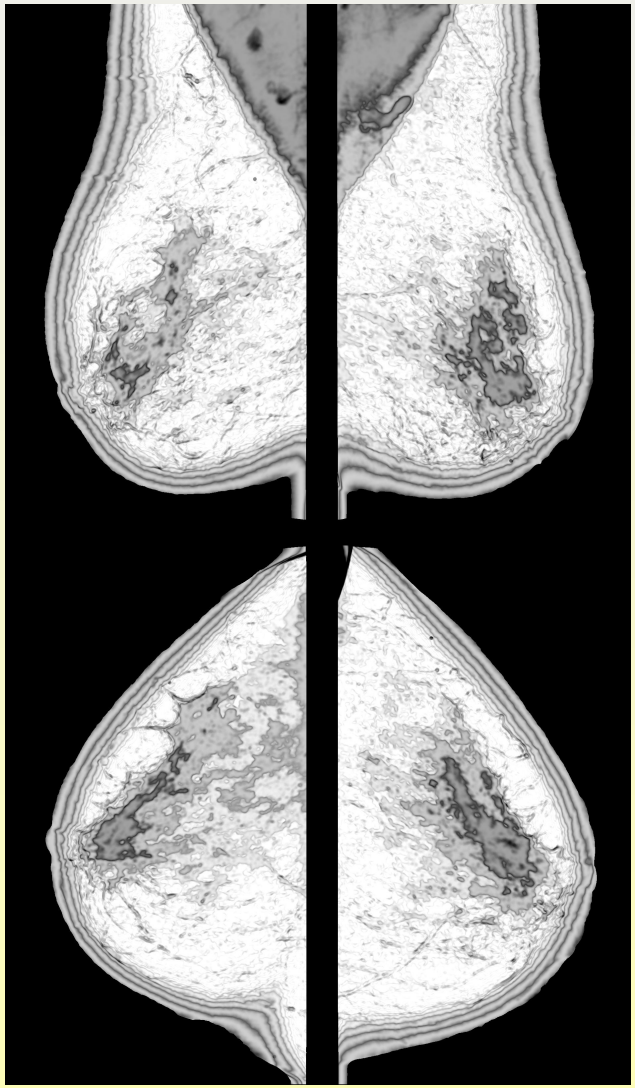
- source database: 2600 full mammograms of South Florida University **http://marathon.csee.usf.edu/Mammography/Database.html**
- four-view (full) mammogram: two medio-lateral and two cranio-caudal images
- mirror transform is applied to right-hand-side images to utilize the underlying symmetry (alternatively: both versions of each view)
- square window of size 13 x 13 pixels with cut-off corners, dimension of $\boldsymbol{x}$ is $N = 145$ $(= 169 - 4 \times 6)$
- $M = 36$ mixture components, randomly initialized parameters
- large data set $|\mathcal{S}| \approx 10^5 - 10^6$ obtained by scanning the image with the search window
- local statistical model is estimated from a single mammogram
- statistical model is invariant with respect to arbitrary linear transform of gray scale

Example 6: Original and Log-Likelihood Image

original image

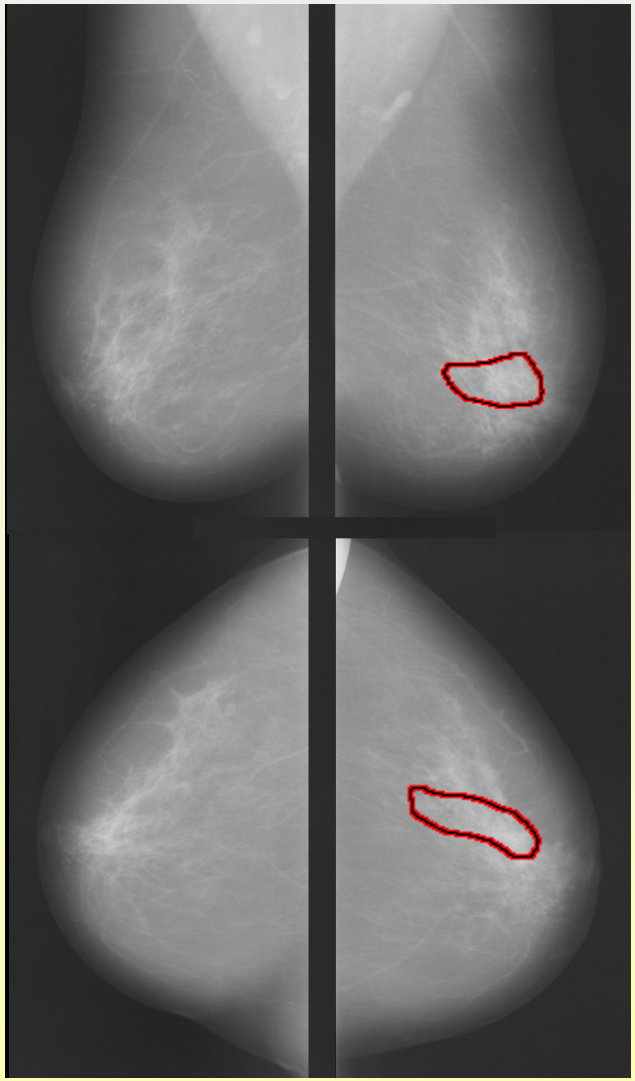


log-likelihood image

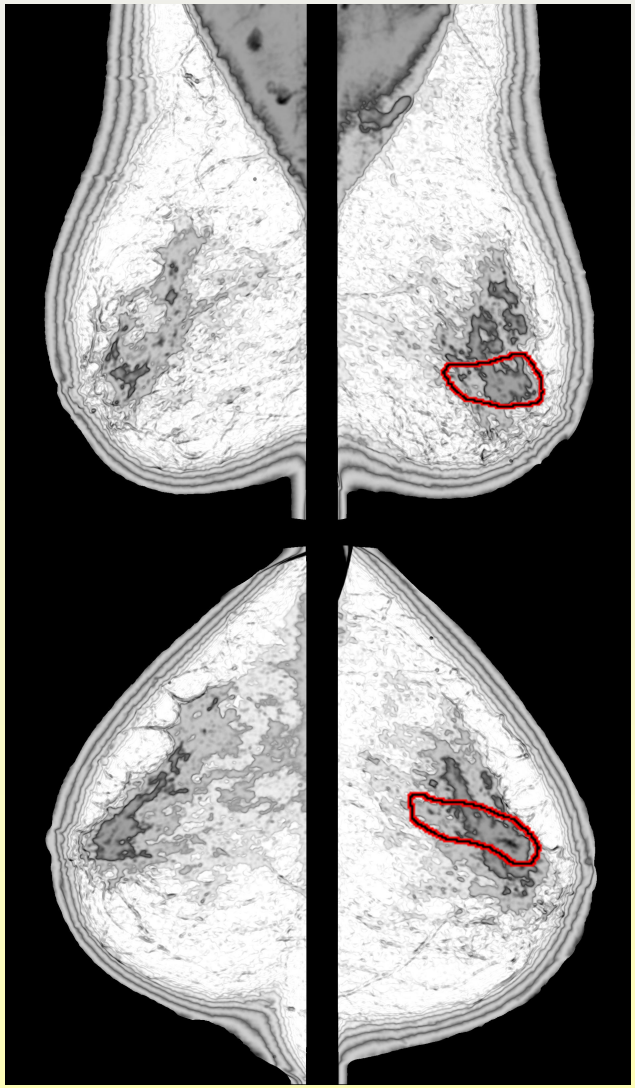


Example 6: Original and Log-Likelihood Image

original image

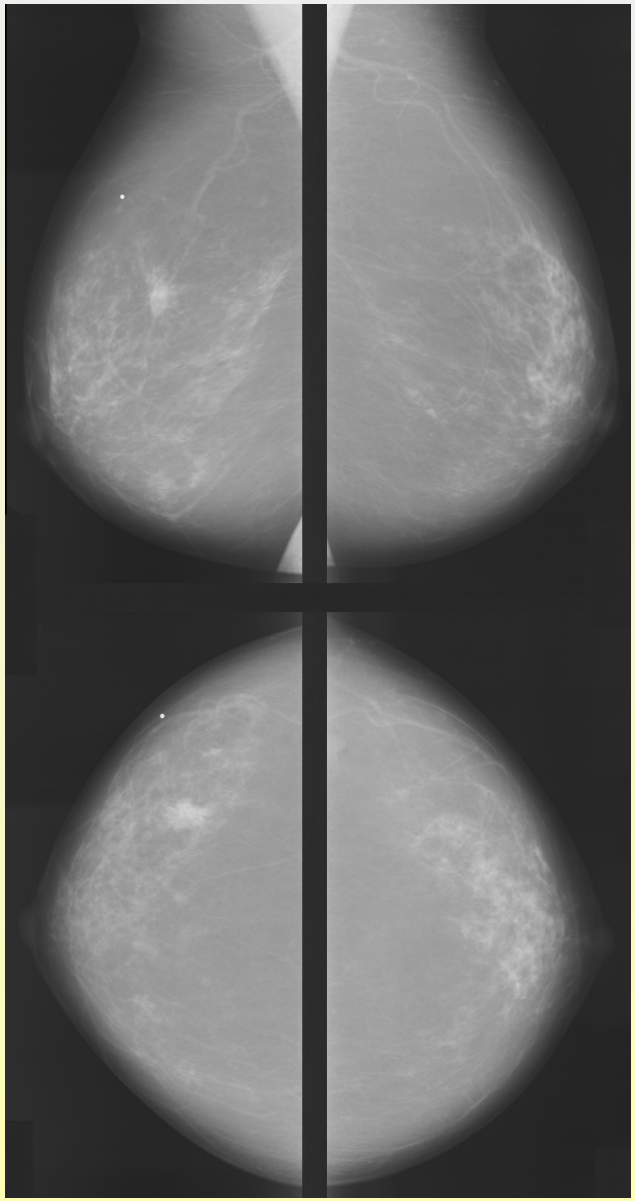


log-likelihood image

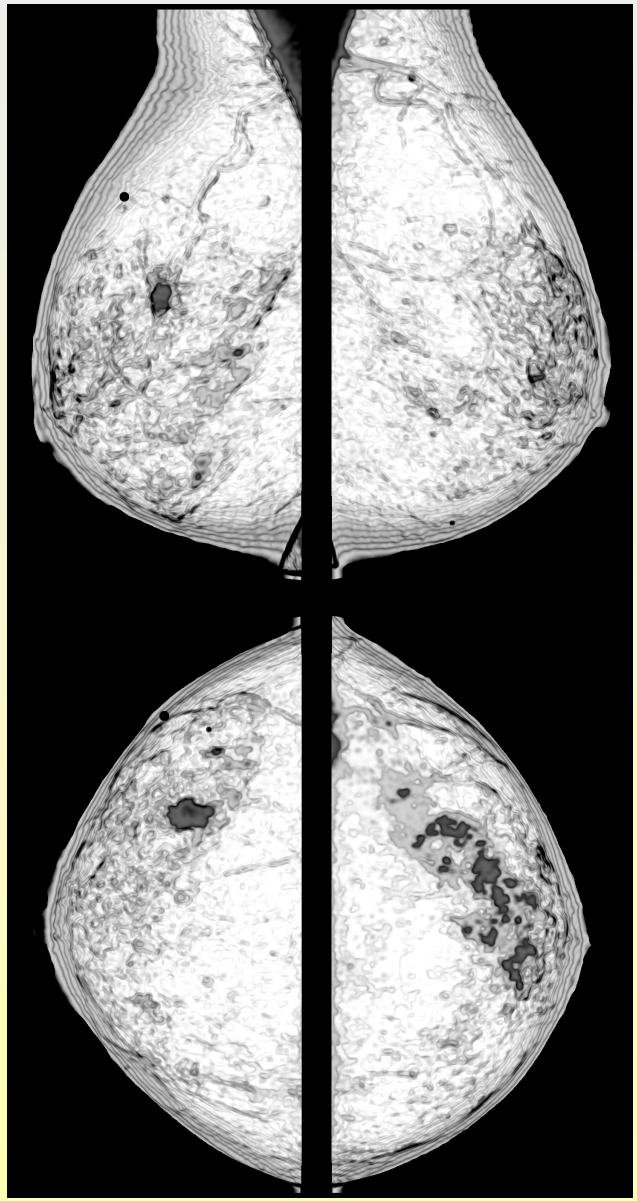


Example 6: Original and Log-Likelihood Image

original image

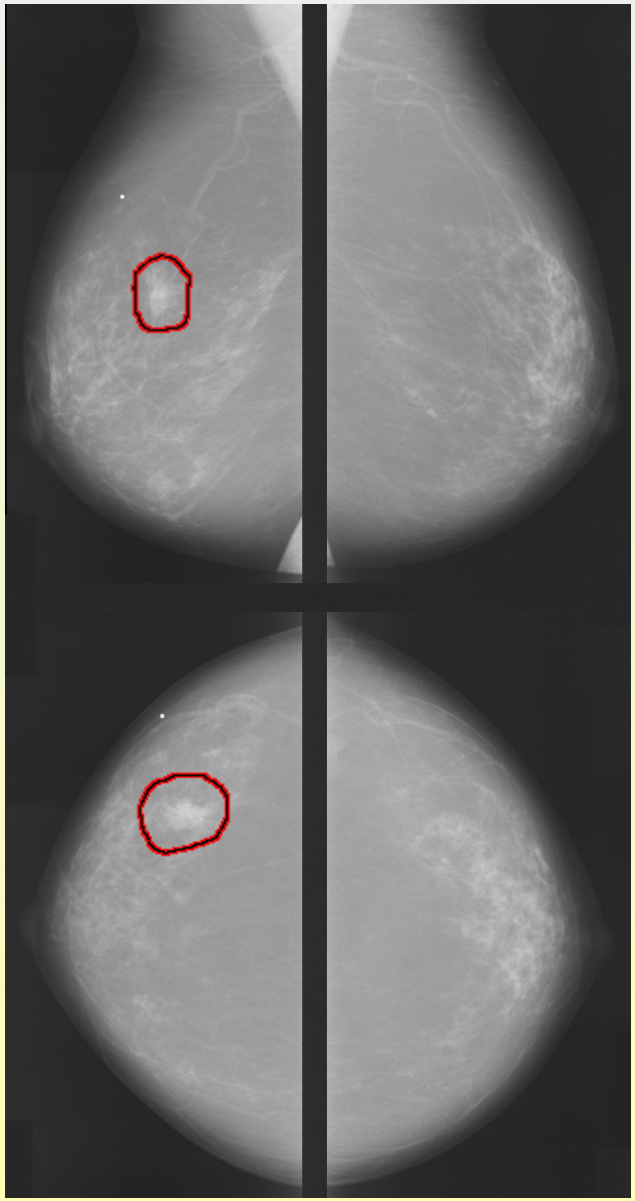


log-likelihood image

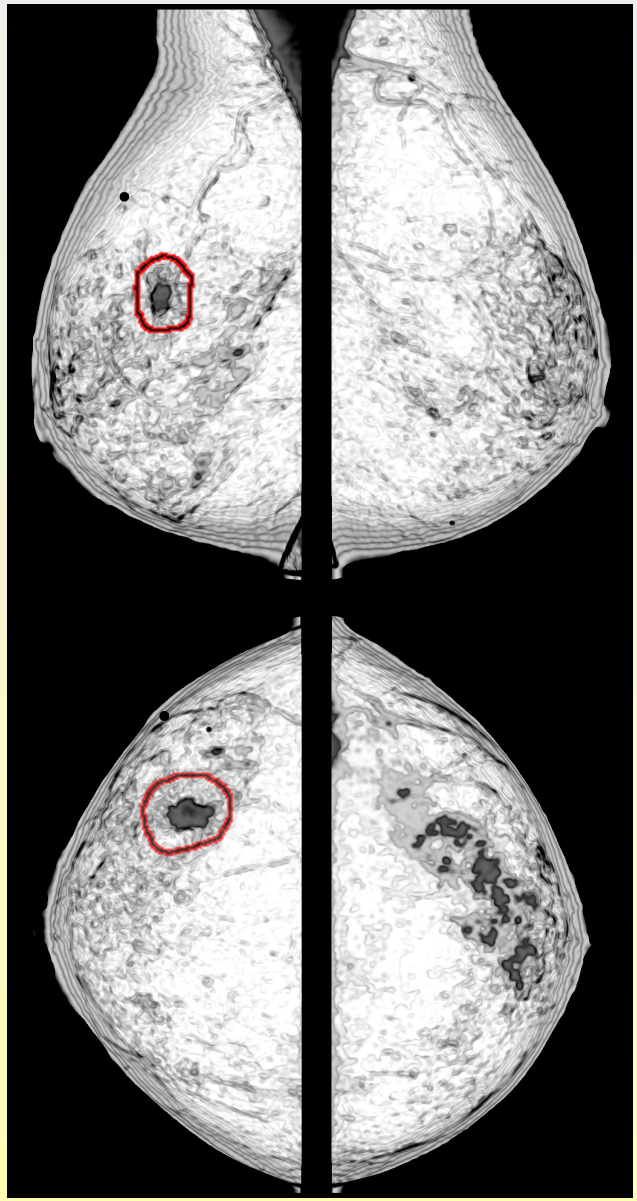


Example 6: Original and Log-Likelihood Image

original image

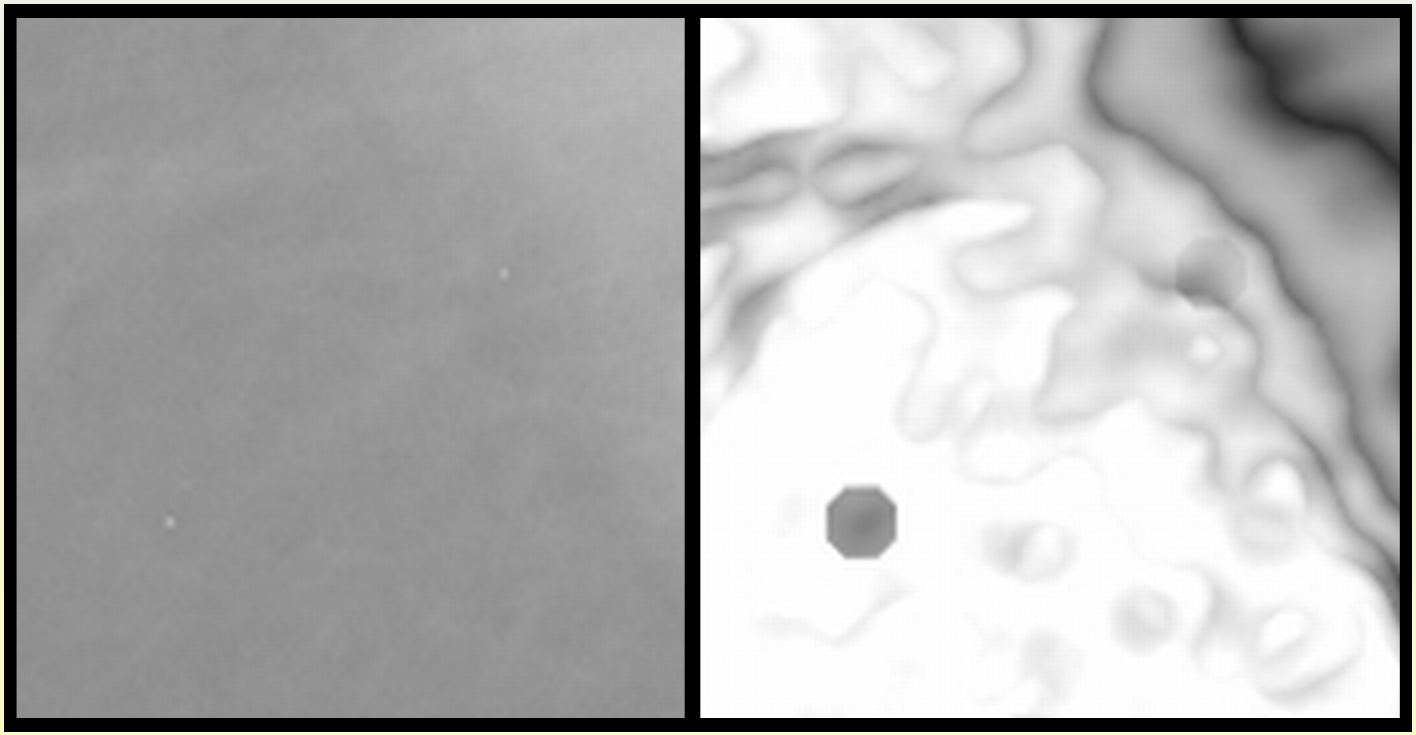


log-likelihood image



Example 6: Indication of "Micro-Calcifications" by Spots

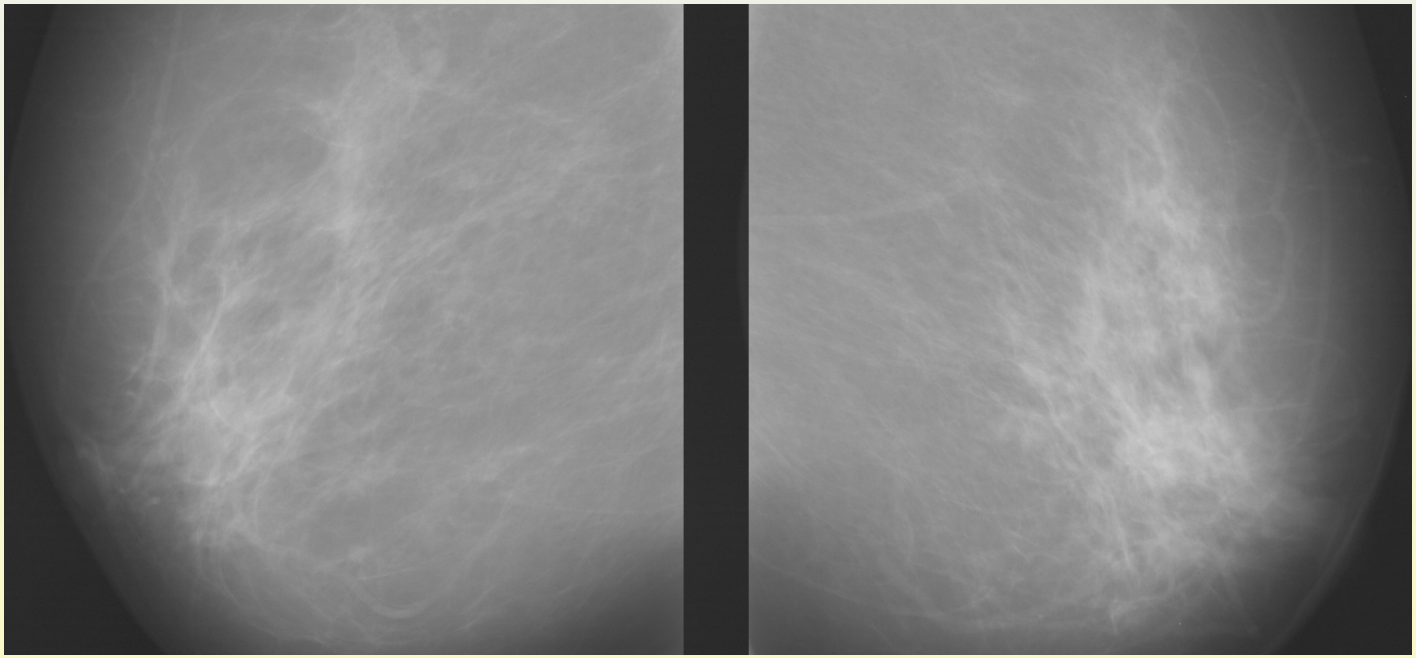
original micro-calcifications corresponding dark patches



Remark: Each position of the window containing a light pixel implies a lower value of $\log P(\boldsymbol{x})$. $\Rightarrow$ A light pixel is identified as a dark spot of window-size.

Example 6: Indication of "Masses" by Contour-Lines

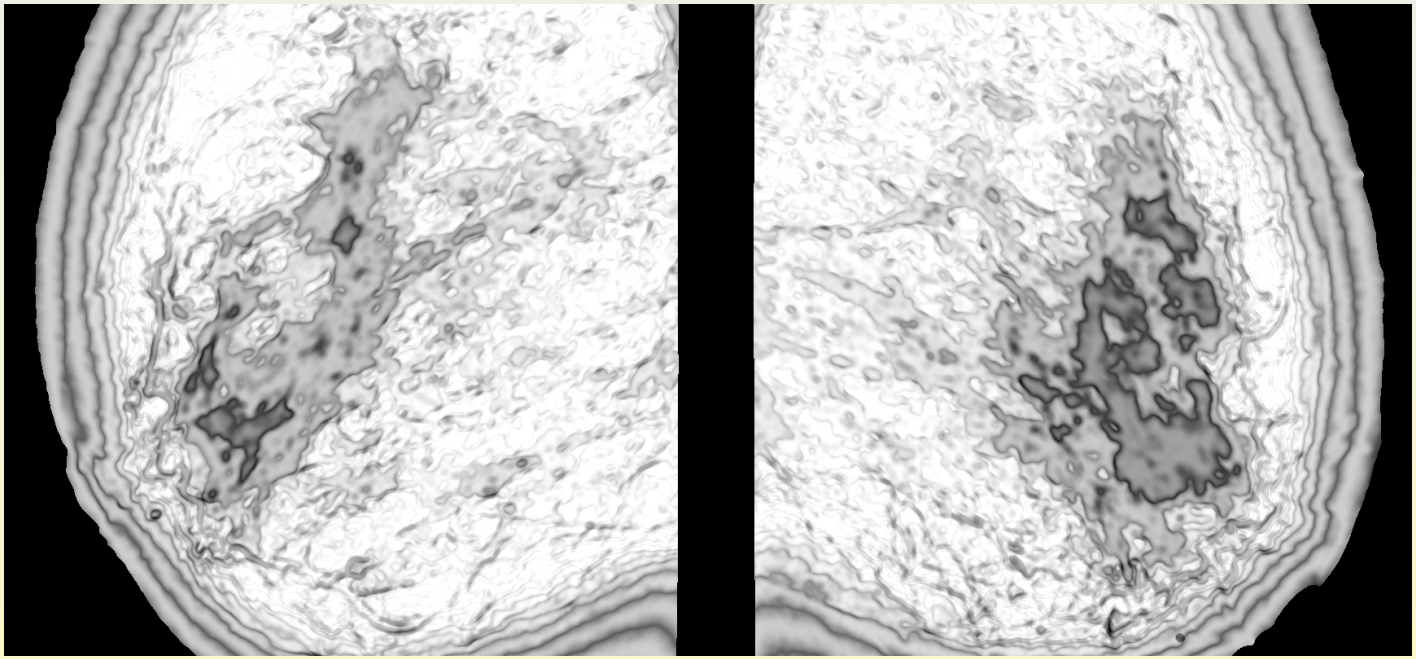
part of the screening mammogram containing suspect "masses"



Remark: The masses may be quite subtle, may have smooth boundaries and different shapes. Detection and classification of masses is more difficult than detection of micro-calcifications.

Example 6: Indication of "Masses" by Contour-Lines

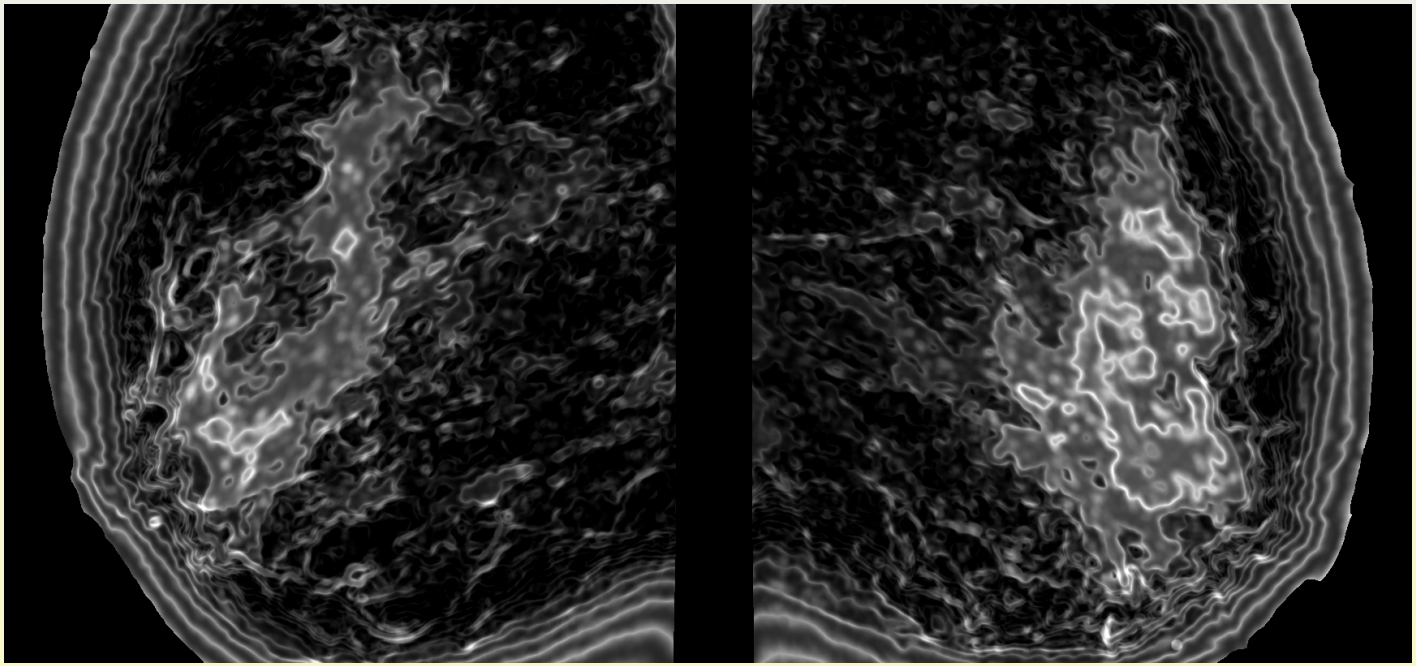
contour lines around "masses" and at the mammogram boundaries



Remark: Log-likelihood values $\log P(\boldsymbol{x})$ are typically dominated by a single component of the mixture which is most adequate to the underlying region. The "switching" of components at the boundaries of different regions accompanied by decreased log-likelihood values is responsible for the arising contour lines.

Example 6: Indication of "Masses" by Contour-Lines

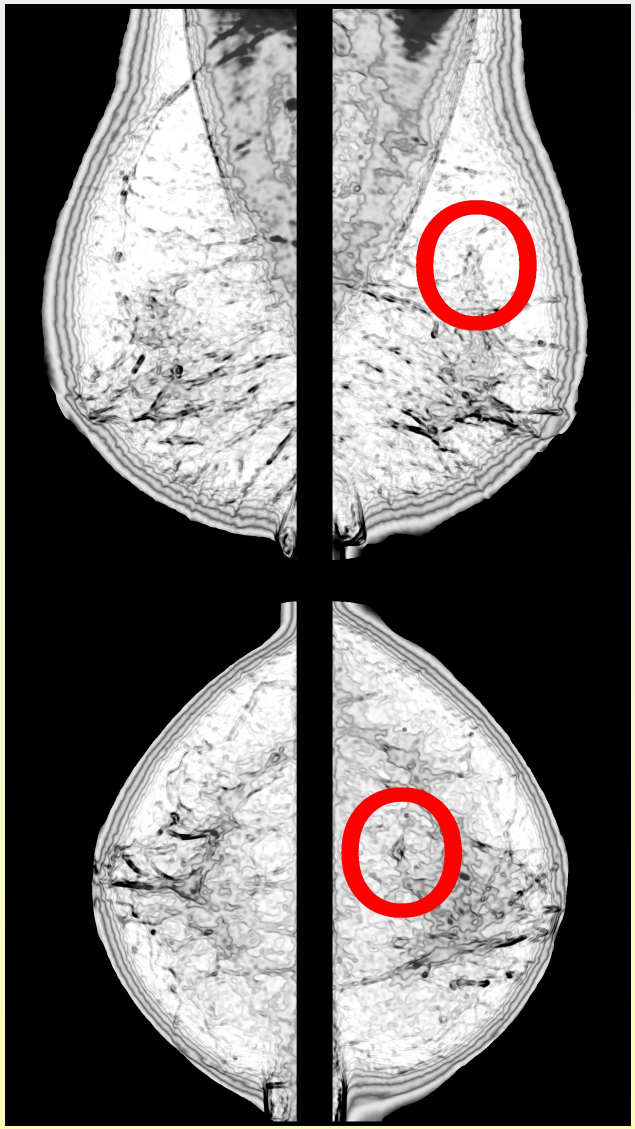
contour lines displayed by the inverse log-likelihood image



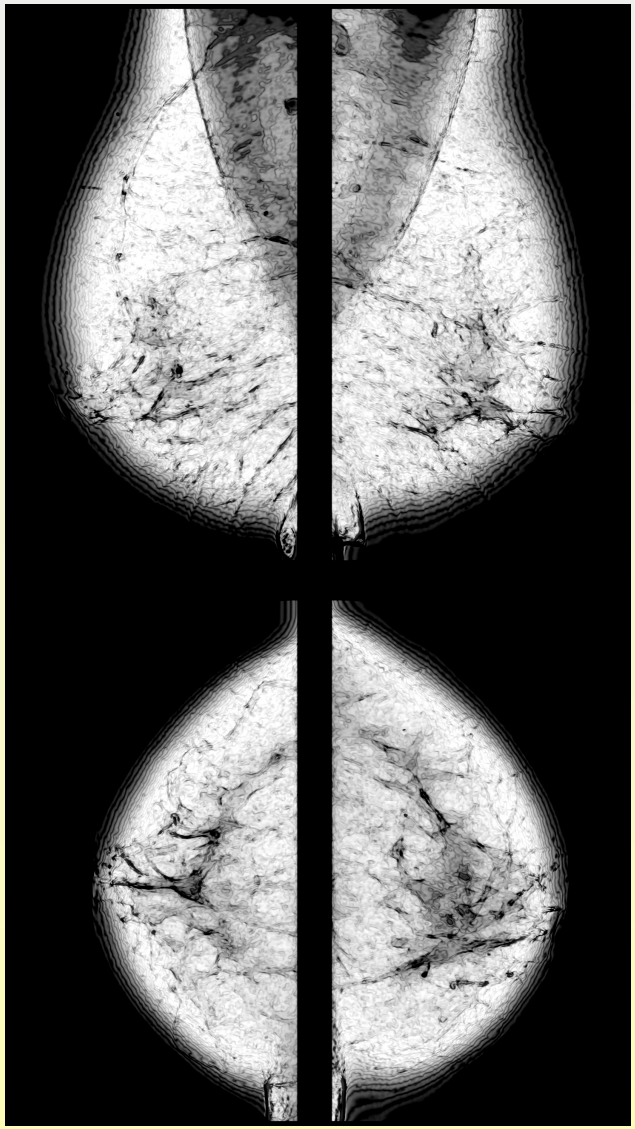
Remark: The most apparent demonstration of contour lines can be seen at the mammogram boundaries characterized by continuously decreasing grey levels. The contour lines may help to evaluate possible contralateral findings or multifocal lesions because regions having similar properties are easily identified visually.

Example 6: Rounded Malignant Mass

Gaussian mixture model



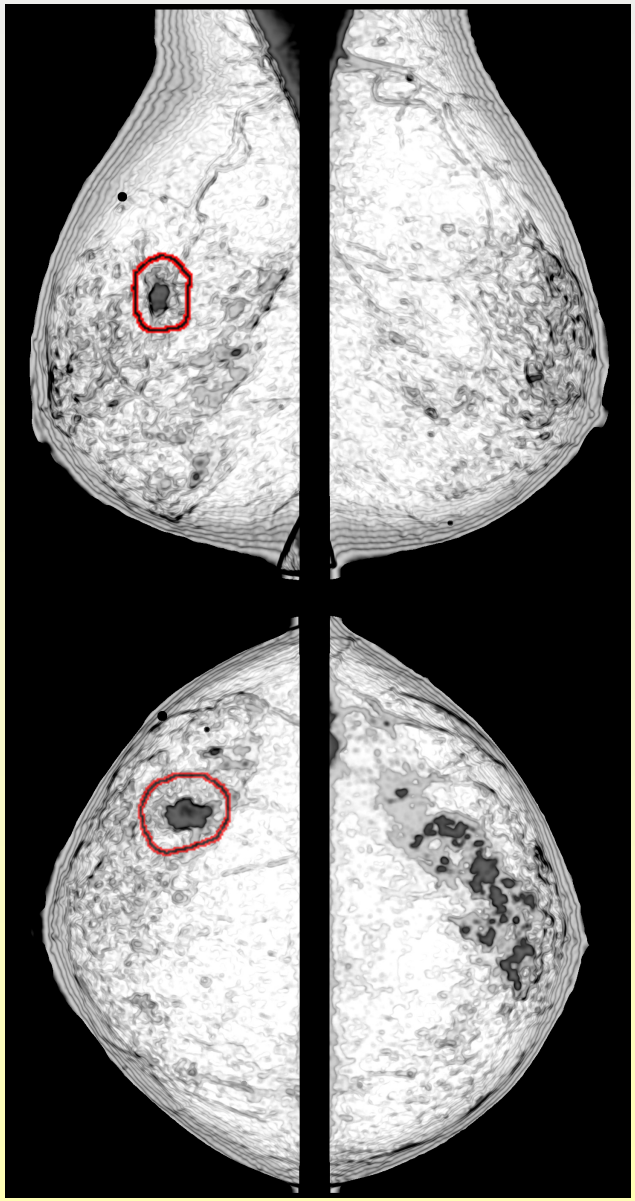
structural mixture model



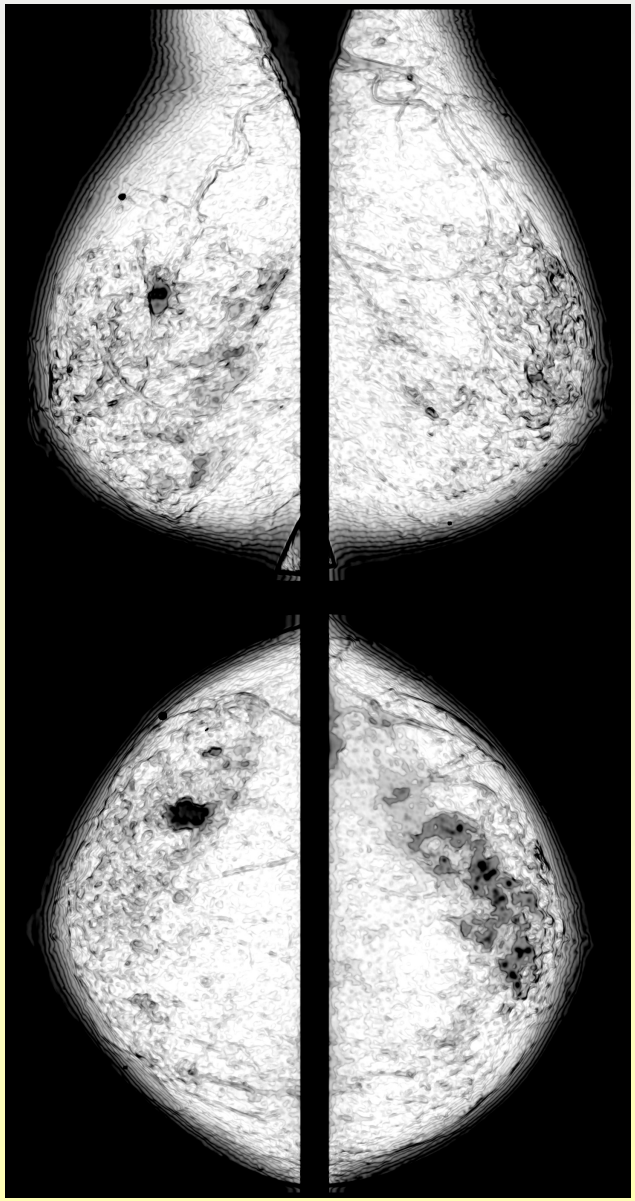
Remark: **Structural mixture model improves the visibility of lesions.**

Example 6: Segmentally distributed Microcalcification

Gaussian mixture model

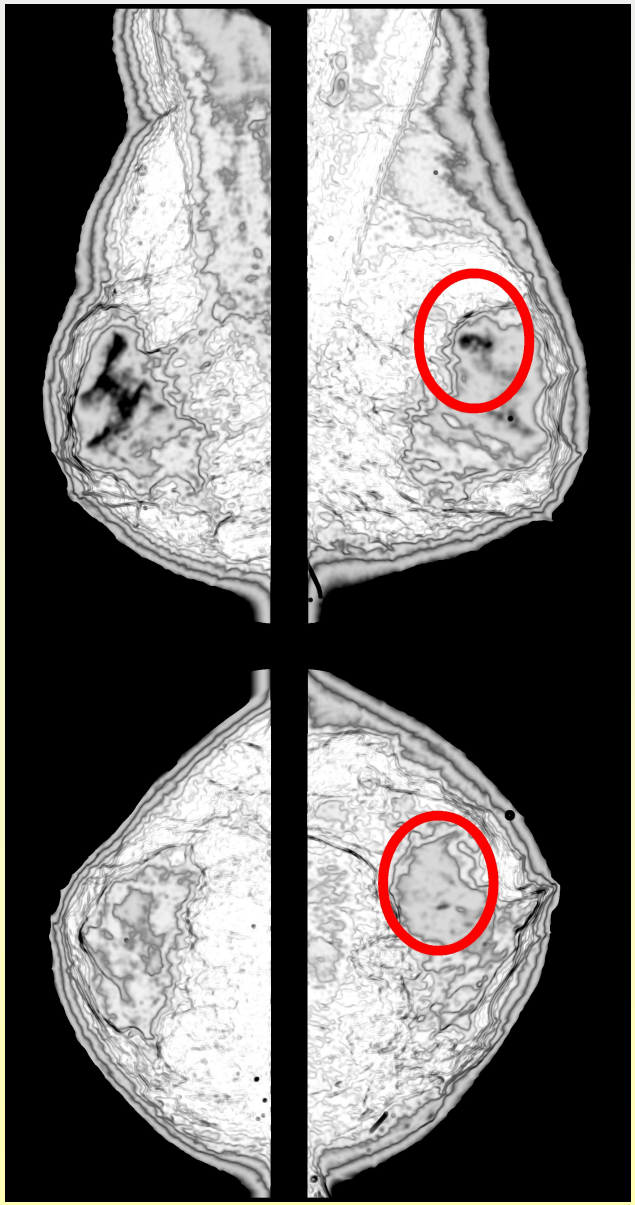


structural mixture model

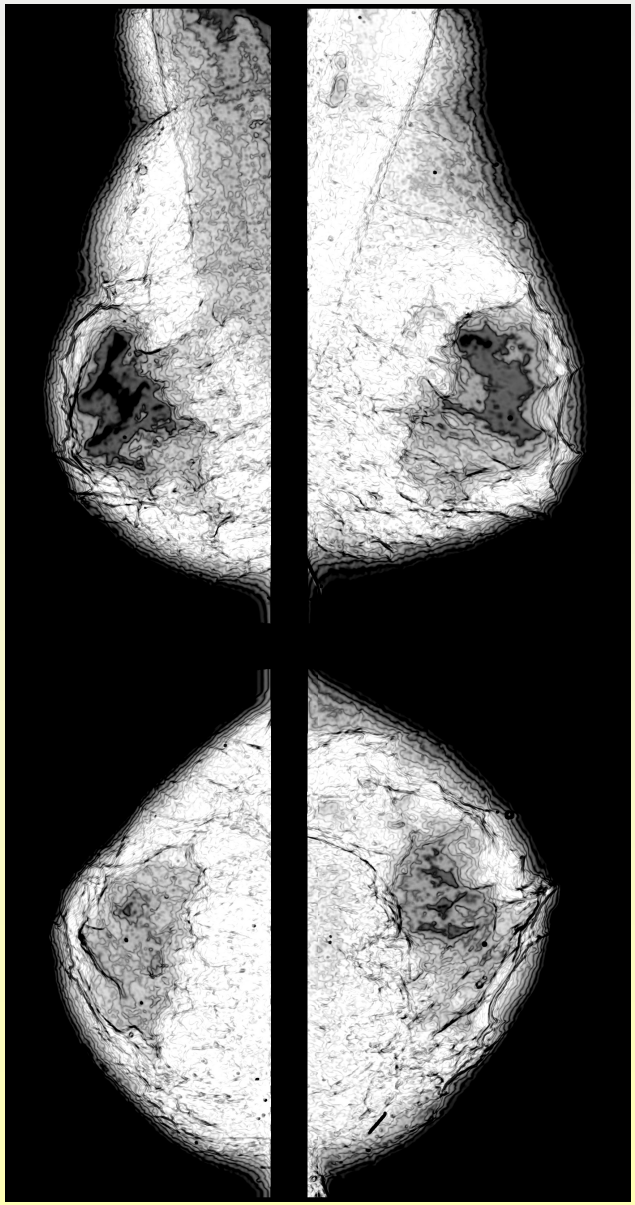


Example 6: Malignant Mass of Asymmetric Density

Gaussian mixture model



structural mixture model



Example 6: Evaluation of Screening Mammograms

SUMMARY:

- **aim:** to facilitate diagnostic evaluation
- **principle:** to emphasize lesions by statistical model as atypical locations
- **advantage:** log-likelihood image has a clear statistical interpretation
- **disadvantage:** the statistical model cannot include medical knowledge
- local mixture model is estimated from the evaluated mammogram only
- $\Rightarrow$ **each mammogram is evaluated individually**
- $\Rightarrow$ **the method need not be trained by other images**
- $\Rightarrow$ **high variability of mammograms is not relevant**
- "masses": emphasized as dark regions with contour lines
- "micro-calcifications": dark spots of the size and shape of window
- useful for evaluation of contra-lateral findings and multi-focal lesions
- **structural mixture model improves the visibility of lesions**

Example 7: Digital Forensic Analysis of Image Data

Specific problem of forensic analysis:

blind detection of possible traces of manipulated locations in images

examples of available methods:

- copy-move forgery detection
- identification of lighting inconsistencies
- detection of periodicity introduced by re-sampling
- evaluation of JPEG quantization artifacts
- detection of locally different statistical properties

STATE OF ART:

- available methods do not allow strict conclusions
- accuracy decreases with lossy compression formats
- results of detection are not always convincing
- only specific types of tampering may be identified

Example 7: Digital Forensic Analysis of Image Data

WE PROPOSE: (Grim et al. 2010)

detection of suspect regions by unusual local statistical properties

Motivation:

some specific features of images (spectral, textural) can be described locally by statistical properties of pixels in a small sliding window

digitized color image: $\mathcal{Z} = [\boldsymbol{z}_{ij}]_{i=1}^{I}{}_{j=1}^{J}$

$\boldsymbol{z}_{ij} = (z_{ij1}, z_{ij2}, z_{ij3}) \in \langle 0, 255 \rangle^3 \quad \approx$ three spectral values for each pixel

$\boldsymbol{x} \approx$ spectral RGB pixel values of the window in a fixed arrangement

$\boldsymbol{x} = (x_1, x_2, \ldots, x_N) \in \langle 0, 255 \rangle^N$

Idea:

- estimation of the multivariate probability density $P(\boldsymbol{x})$
- identification of untypical locations by low likelihood values $\log P(\boldsymbol{x})$

Example 7: Digital Forensic Analysis of Image Data

STATISTICAL MODEL: **Gaussian mixture of product components**

$$P(\boldsymbol{x}) = \sum_{m=1}^{M} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \sum_{m=1}^{M} w_m \prod_{n=1}^{N} f_n(x_n|\mu_{mn}, \sigma_{mn})$$

$$f_n(x_n|\mu_{mn}, \sigma_{mn}) = \frac{1}{\sqrt{(2\pi)}\sigma_{mn}} \exp\left\{ -\frac{(x_n - \mu_{mn})^2}{2\sigma_{mn}^2} \right\}$$

MODEL ESTIMATION: **by means of EM algorithm**

Invariance Property:

log-likelihood image is invariant with respect to arbitrary linear transform of the grey scale of the original image

REMARK: The component means $\boldsymbol{\mu}_m$ are computed as weighted averages of the sample vectors $\boldsymbol{x} \in \mathcal{S}$ (cf. EM algorithm) and therefore they are rather smooth without high frequency details. Thus, inserted image portion with suppressed high frequencies will be more probable.

Example 7: Digital Forensic Analysis of Image Data

$\log P(\boldsymbol{x}) \approx$ **measure of typicality of the window patch** $\boldsymbol{x}$
$\log P(\boldsymbol{x}) \approx$ **displayed as grey level at the central pixel of the window**

INTERPRETATION: **dark pixels** corresponding to the low values of $\log P(\boldsymbol{x})$ may indicate "untypical" or "suspect" locations of the image

Mechanisms of Forgery Detection:

- unusual spectral properties of small areas will be less probable
- unusual textural properties of small areas will be less probable
- blurred regions will appear more probable **(!)** because of missing high-frequency details

scaling of log-likelihood image: $\log P(\boldsymbol{x}) \in \langle \mu_0 - 2 * \sigma_0;\ \mu_0 + 2 * \sigma_0 \rangle$

REMARK: In high-dimensional spaces the density values $P(\boldsymbol{x})$ of adjacent windows may differ by several orders; therefore the log-likelihood values $\log P(\boldsymbol{x})$ are more suitable as a measure of typicality.

Example 7: Digital Forensic Analysis of Image Data

COMPUTATIONAL DETAILS OF NUMERICAL EXPERIMENTS:

- small square window of 5x5 pixels with trimmed corners
- (large windows tend to smooth out small details)
- 21 window pixels in three colors imply the model dimension N=63
- the estimated mixture density $P(\boldsymbol{x})$ describes the statistical properties of the 63 color sample values x_n of window patch
- training data set $\mathcal{S}$ is obtained by scanning the image with the search window
- the source texture images imply training data sets of size $|\mathcal{S}| \approx 10^6$
- number of components $M \approx 10^2$
- EM algorithm: random initialization, stopping rule: relative increment threshold ($\approx$ 10 - 20 iterations)
- **computing time:** picture: 3M pixels, model: M=20 components, dimension: N=63, 20 iterations $\approx$ 15 minutes (standard PC)

Example 7: Digital Forensic Analysis of Image Data

Original image including an inserted oval region in the left-upper part.

Example 7: Digital Forensic Analysis of Image Data

The oval part in the left-upper corner having somewhat different textural properties becomes distinctly lighter in the log-likelihood image.

Example 7: Digital Forensic Analysis of Image Data

Original image assembled from two parts by auto-stitch software.

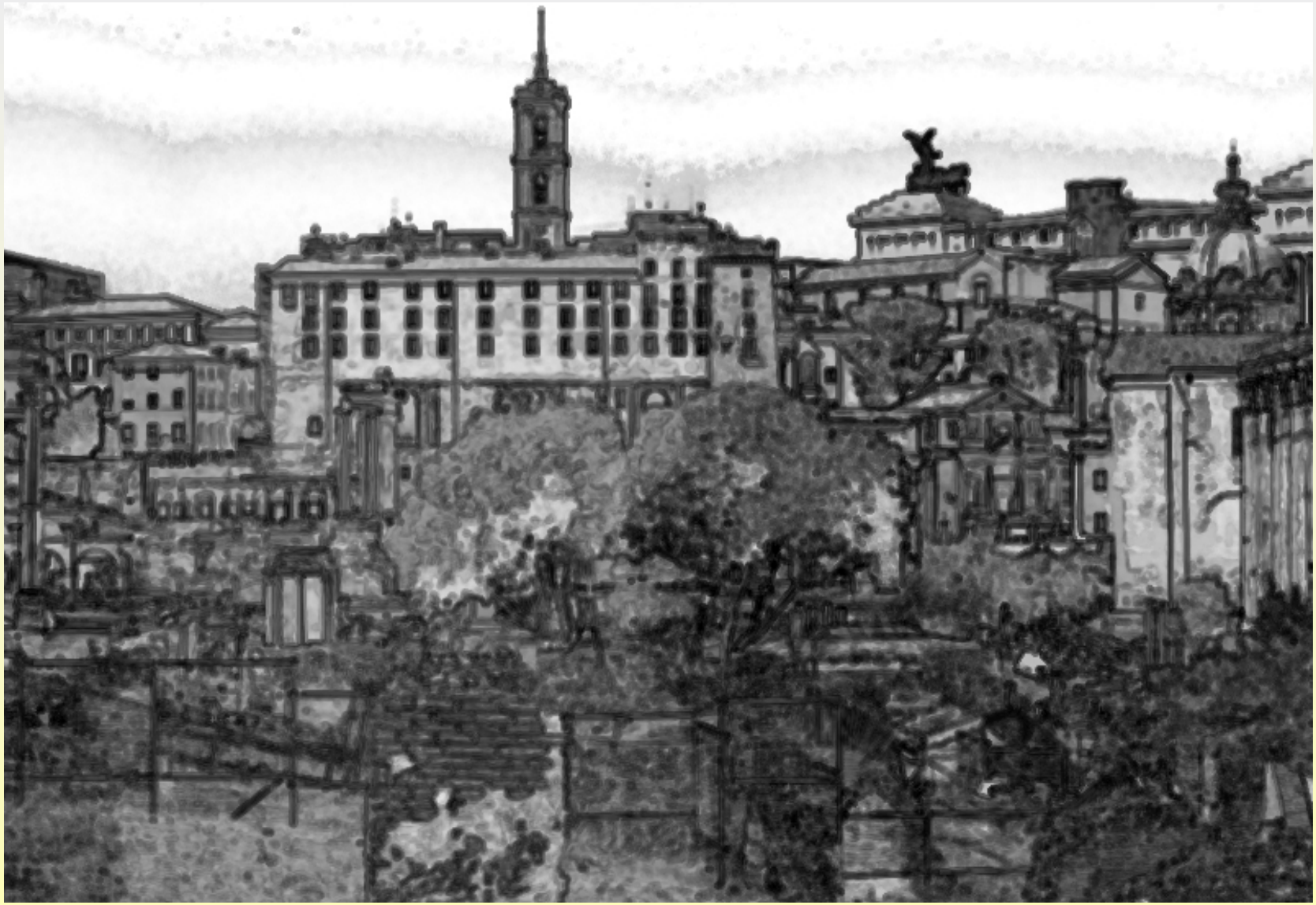
Example 7: Digital Forensic Analysis of Image Data

The slightly blurred left part becomes lighter in the log-likelihood image.

Example 7: Digital Forensic Analysis of Image Data

Original picture assembled from three parts by auto-stitch software.

Example 7: Digital Forensic Analysis of Image Data



The medium slightly blurred (incorrectly focused) part becomes lighter.

Example 7: Digital Forensic Analysis of Image Data

Concluding Remarks:

Properties of the Log-Likelihood Image:

- component means computed as weighted averages of data vectors are rather smooth
- log-likelihood image is invariant with respect to arbitrary linear transforms of the grey scales
- even small differences in brightness, resolution, frequency content or texture may cause visible changes in the log-likelihood image

Identification of Suspect Regions by Local Statistical Model:

- forgery detection by local statistical models is a blind method
- applicable to images of unknown origin without any prior information
- no specific type of image tampering is assumed
- capable to expose image manipulations of various kinds
- reasonably resistent to lossy information compression

Example 8: Image Inpainting by Local Prediction

PRINCIPLE: (Grim et al. 2008)

- estimation of local statistical image model as a Gaussian product mixture $P(\boldsymbol{x})$
- stepwise prediction of missing pixels by using conditional mixtures

known part of the sliding window:

$$\boldsymbol{x}_C = (x_{i_1}, \ldots, x_{i_k}), \quad C = \{i_1, \ldots, i_k\} \subset \mathcal{N}$$

conditional distribution for the missing pixel variable x_n:

$$P_{n|C}(x_n|\boldsymbol{x}_C) = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_C) f_n(x_n|\mu_{mn}, \sigma_{mn}), \quad n \notin C$$

$$W_m(\boldsymbol{x}_C) = \frac{w_m F_C(\boldsymbol{x}_C|\boldsymbol{\mu}_{mC}, \boldsymbol{\sigma}_{mC})}{\sum_{j=1}^{M} w_j F_C(\boldsymbol{x}_C|\boldsymbol{\mu}_{jC}, \boldsymbol{\sigma}_{jC})} \quad \approx \text{ nearly binary values}$$

PREDICTION: conditional expectation $\bar{x}_n$ given the defined part $\boldsymbol{x}_C$:

$$\bar{x}_n = \mathrm{E}_{n|C}\{x_n|\boldsymbol{x}_C\} = \int x_n P_{n|C}(x_n|\boldsymbol{x}_C) dx_n = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_C)\mu_{mn} \approx \mu_{m_0 n}$$

ÚTIA

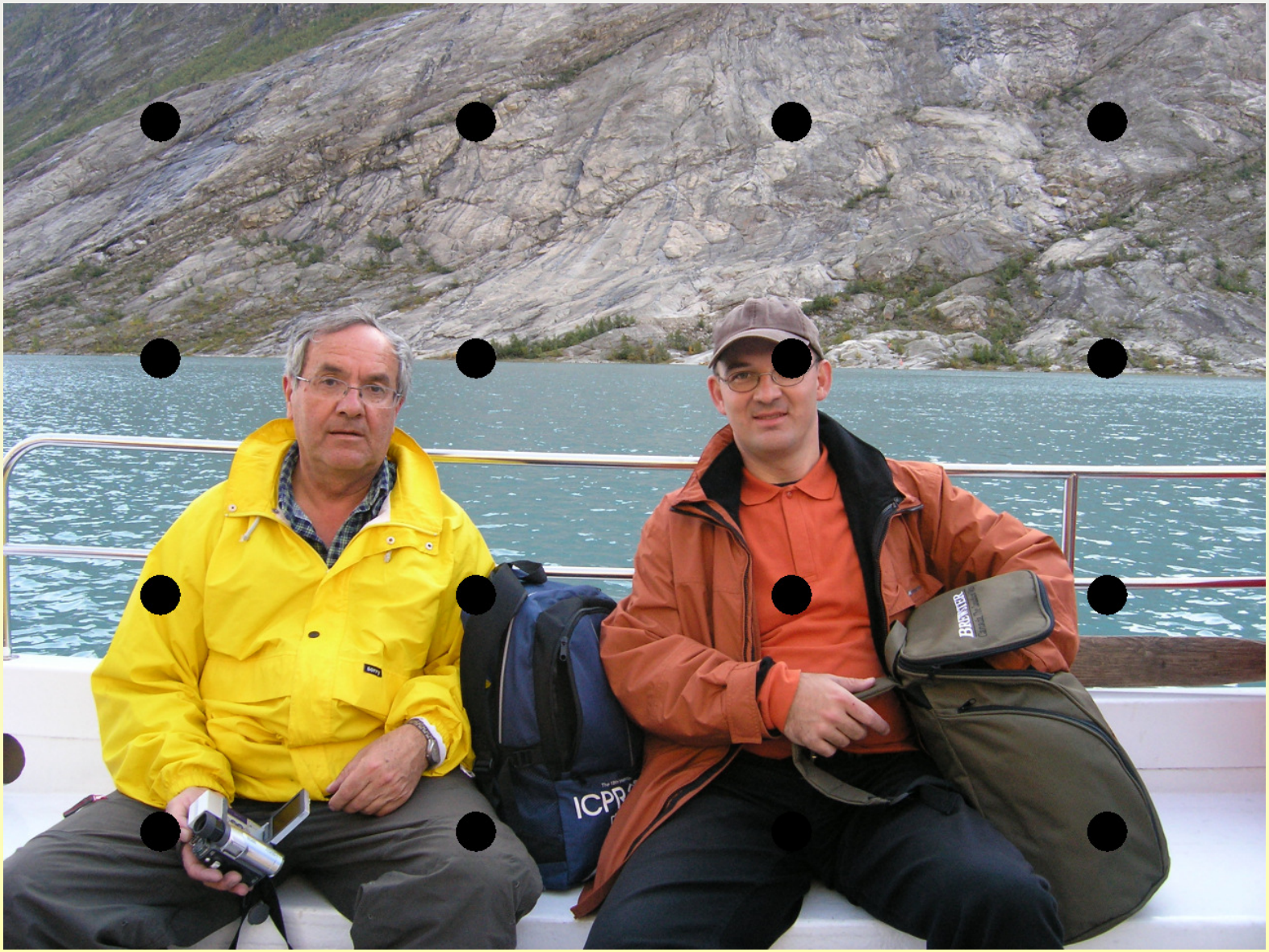
Example 8: Image Inpainting by Local Prediction

COMPUTATIONAL DETAILS OF NUMERICAL EXPERIMENTS:

- image size 1280 x 960 pixels
- square **sliding window** 7x7 pixel with cut-off corners includes 37 interior pixels
- **data set:** scanning the image by sliding window: $\Rightarrow \ |\mathcal{S}| \approx 10^6$ vectors
- three spectral components at 37 interior pixel $\Rightarrow$ **dimension** $\boldsymbol{x}$: N=111
- number of mixture components in different experiments: $M \approx 20 \div 80$
- randomly initialized components with uniform initial weights
- the stopping rule based on relative increment threshold $\Delta L \approx 10^{-3}$ (corresponds to cca 10 - 20 iterations)
- **model computing time:** cca $\approx$ 15 - 30 minutes (standard PC)
- stepwise prediction of missing parts completes the image in 3 to 5 iterations (depends on the form and size of missing parts)

Example 8:Image Inpainting by Local Prediction

damaged source image

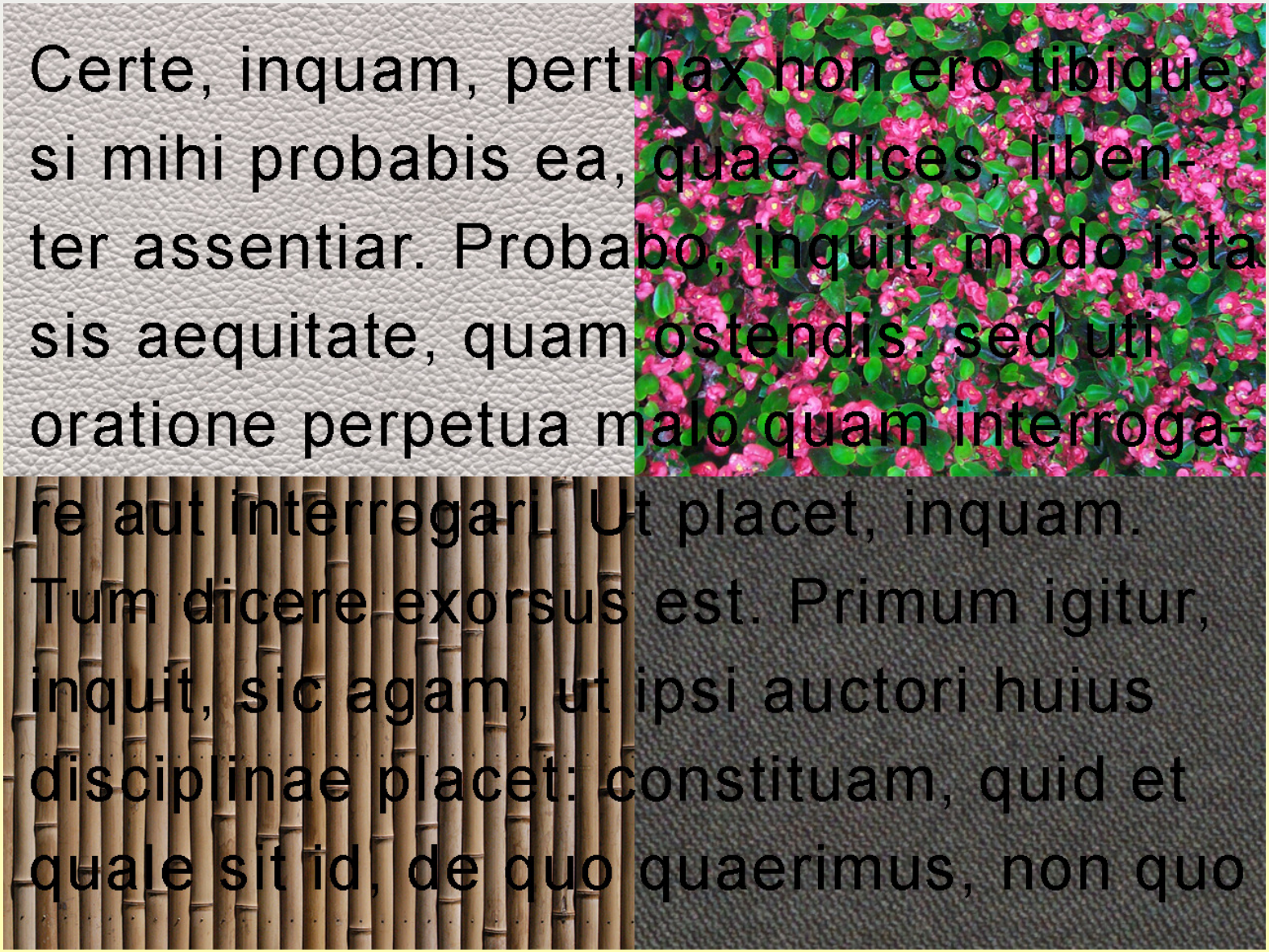


Example 5: Image Inpainting by Local Prediction

inpainted image

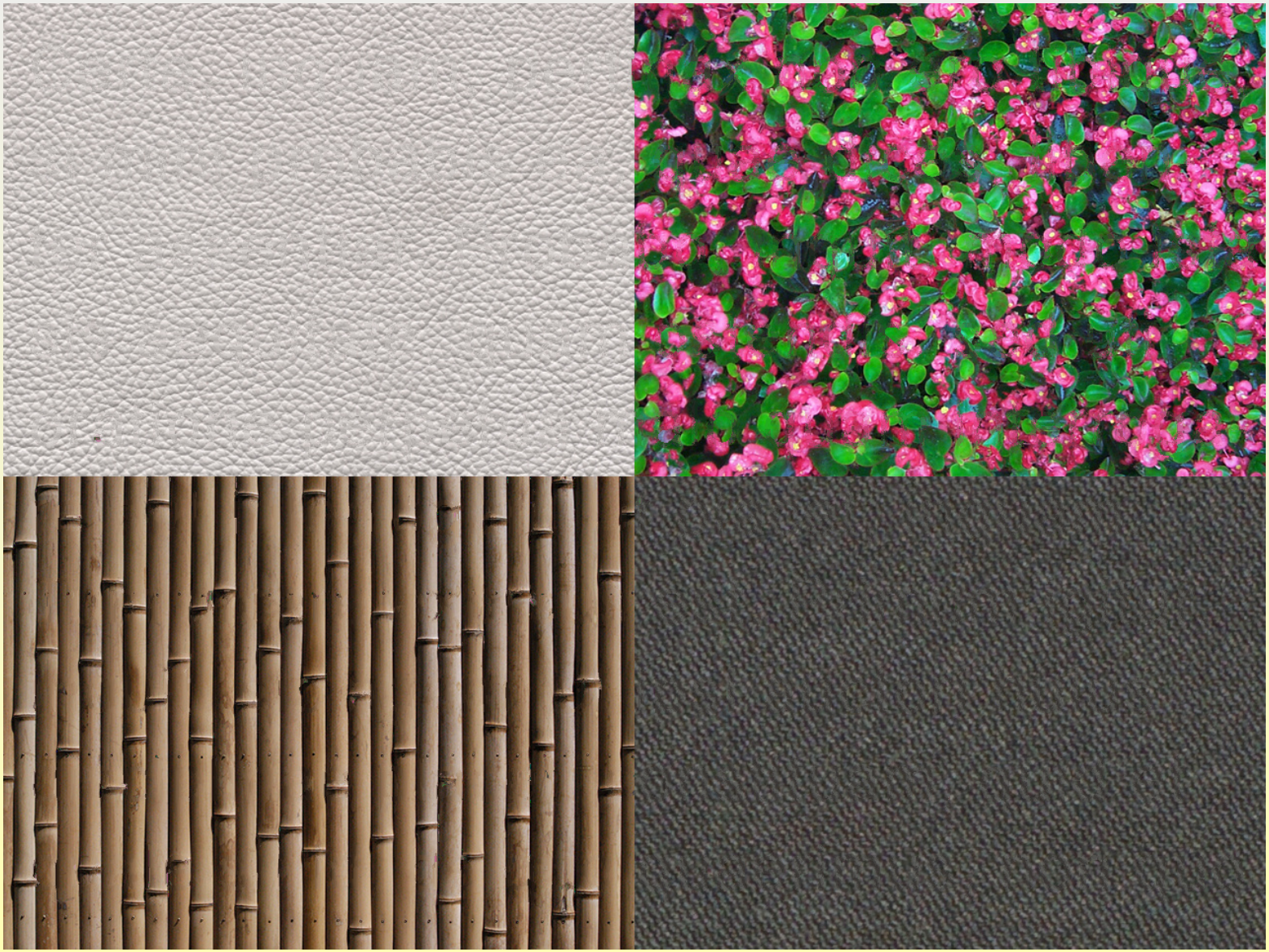
Example 8: Image Inpainting by Local Prediction

damaged source image



Example 8: Image Inpainting by Local Prediction

inpainted image



Example 8: Image Inpainting by Local Prediction

damaged source image

Example 8: Image Inpainting by Local Prediction

inpainted image

Example 9: Interactive Statistical Model of Census Data

PRINCIPLE: interactive information retrieval from statistical model
(Grim et al. 1992, 1995, 2001, 2004, 2009, 2010)

statistical model of the database:

$$P(\boldsymbol{x}) = \sum_{m=1}^{M} w_m F(\boldsymbol{x}|m) = \sum_{m=1}^{M} w_m \prod_{n=1}^{N} p_n(x_n|m)$$

conditional distributions $P_{n|C}(x_n|\boldsymbol{x}_C), n \notin C$ **given** $\boldsymbol{x}_C = (x_{i_1}, \ldots, x_{i_k})$**:**

$$P_{n|C}(x_n|\boldsymbol{x}_C) = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_C) f_n(x_n|m), \quad W_m(\boldsymbol{x}_C) = \frac{w_m F_C(\boldsymbol{x}_C|m)}{\sum_{j=1}^{M} w_j F_C(\boldsymbol{x}_C|j)}$$

Interactive statistical model:

- distributable without any confidentiality concerns
- $\Rightarrow$ **suitable for medical data**
- **advantage:** high accuracy of the model
- **advantage:** easily distributed because of information compression
- the model can be computed from incomplete data

Example 9: Interactive Statistical Model of Census Data

Questions in the statistical model of the 2001 Czech Census.

	Text of question	# of values	Missing in %	Entropy in %
1.	Region of residence	14	0.00	96.88
2.	Type of residence	3	0.00	32.92
3.	Economic activity	10	0.80	67.80
4.	Birth place (relatively)	6	1.95	74.65
5.	Religion	6	0.00	60.57
6.	Occupation type	14	3.89	68.33
7.	Sex	2	0.00	99.95
8.	Marital status	4	0.55	81.01
9.	Education	14	1.11	78.04
10.	Age	9	0.03	96.09
11.	Category of flat	5	0.53	27.81
12.	Bathroom	5	0.59	14.02
13.	Size of flat	7	0.64	80.62
14.	Internet and PC	4	2.85	49.11
15.	Legal relation to flat	9	0.39	72.43
16.	Gas supply	3	0.78	64.54
17.	Number of rooms over 8m^2	7	0.64	80.57
18.	Number of cars in household	4	3.39	71.32
19.	Number of persons in flat	6	0.00	93.79
20.	Vacational property	6	7.45	42.10
21.	Telephone in flat	5	1.80	80.88
22.	Water supply	4	0.35	8.02
23.	Type of heating	6	0.53	74.81
24.	Toilet	6	0.50	16.73

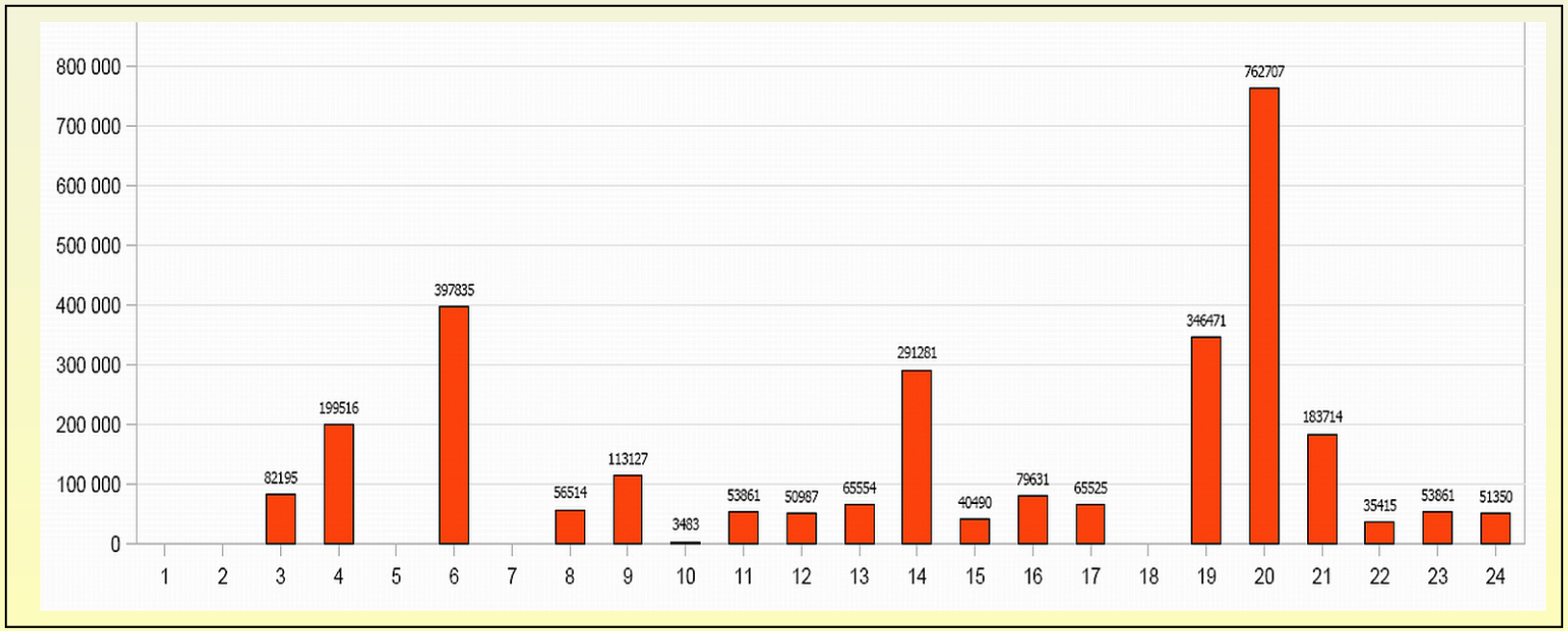
There are 10,230,060 respondents 1,524,240 incomplete records and 2,933,427 missing answers.

Example 9: Interactive Statistical Model of Census Data

Model estimation: (Grim et al. 2010)

- **estimation of parameters from incomplete data**
- **alternatively: estimation of missing data first**

Non-response frequency for individual questions.



Remark: The number of incomplete records is 1,524,240, the total number of missing values is 2,933,427.

Example 9: Interactive Statistical Model of Census Data

Accuracy of information retrieval from the statistical model

Evaluation of reproduction error for **26 mil.** combinations of at most five values x_n. (Only sub-populations greater then 1571 respondents are considered - cf. paper for details.)

$$\mathcal{A}_5 = \{\boldsymbol{x}_C = (x_{i_1}, \ldots, x_{i_5}) : N(\boldsymbol{x}_C) > 1571\}, \qquad N(\mathcal{A}_5) = 26\ 425\ 727$$

true sub-population size: $N(\boldsymbol{x}_C)$

estimated sub-population size: $P(\boldsymbol{x}_C)N$

mean absolute reproduction error E_a:

$$E_a = \frac{1}{N(\mathcal{A}_5)} \sum_{\boldsymbol{x}_C \in \mathcal{A}_5} |P(\boldsymbol{x}_C)N - N(\boldsymbol{x}_C)|, \qquad P(\boldsymbol{x}_C) = \sum_{m=1}^{M} w_m \prod_{j=1}^{5} p_{i_j}(x_{i_j}|m)$$

mean relative reproduction error E_r in %:

$$E_r = \frac{100}{N(\mathcal{A}_5)} \sum_{\boldsymbol{x}_C \in \mathcal{A}_5} \frac{|P(\boldsymbol{x}_C)N - N(\boldsymbol{x}_C)|}{N(\boldsymbol{x}_C)}$$

ÚTIA

Example 9: Interactive Statistical Model of Census Data

Relative and absolute error of the statistical census model

(24 variables, incomplete data, M=15000 components)

($\mathcal{A}_4 \approx$ combinations of up to four values, $\mathcal{A}_5 \approx$ combinations of up to five values)

Error Criterion	**Test $\mathcal{A}_4$**	**Test $\mathcal{A}_5$**
mean relative reproduction error in %:	**4.10**	**4.20**
standard deviation of the relative error:	6.37	5.84
maximum relative reproduction error in %:	368.62	368.62
mean absolute reproduction error:	460	338
standard deviation of the absolute error:	951	655
maximum absolute reproduction error:	45779	45779
Number of test subpopulations	**3 503 448**	**26 425 727**

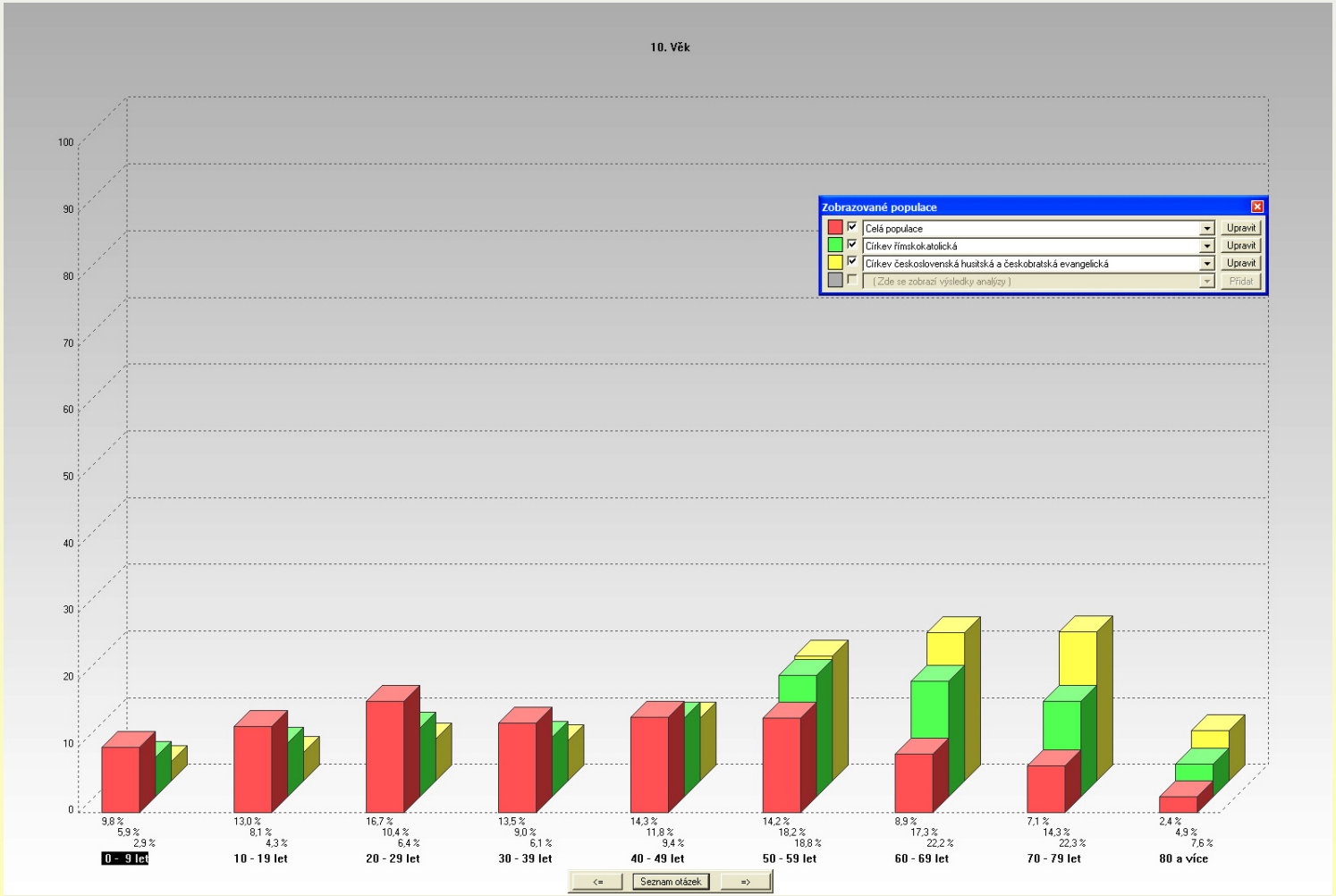
$\Rightarrow$ **the mean error of displayed histogram columns is 4.17%**

interactive model and publication: **http://ro.utia.cas.cz/census/**

ÚTIA

Example 9: Interactive Statistical Model of Census Data

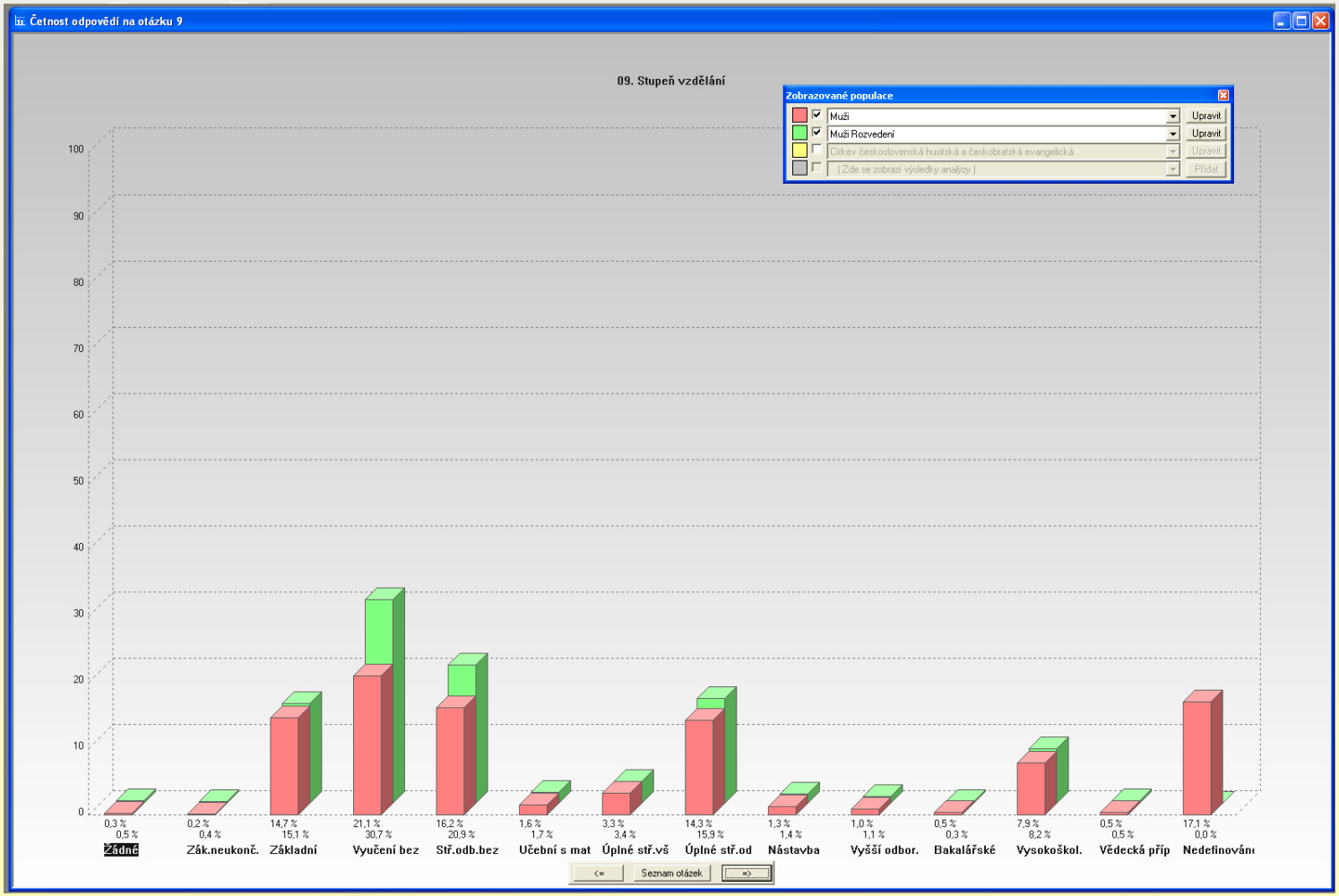
Age distributions of three different sub-populations



Remark: Analysis of small sub-populations is limited by model accuracy.

Example 9: Interactive Statistical Model of Census Data

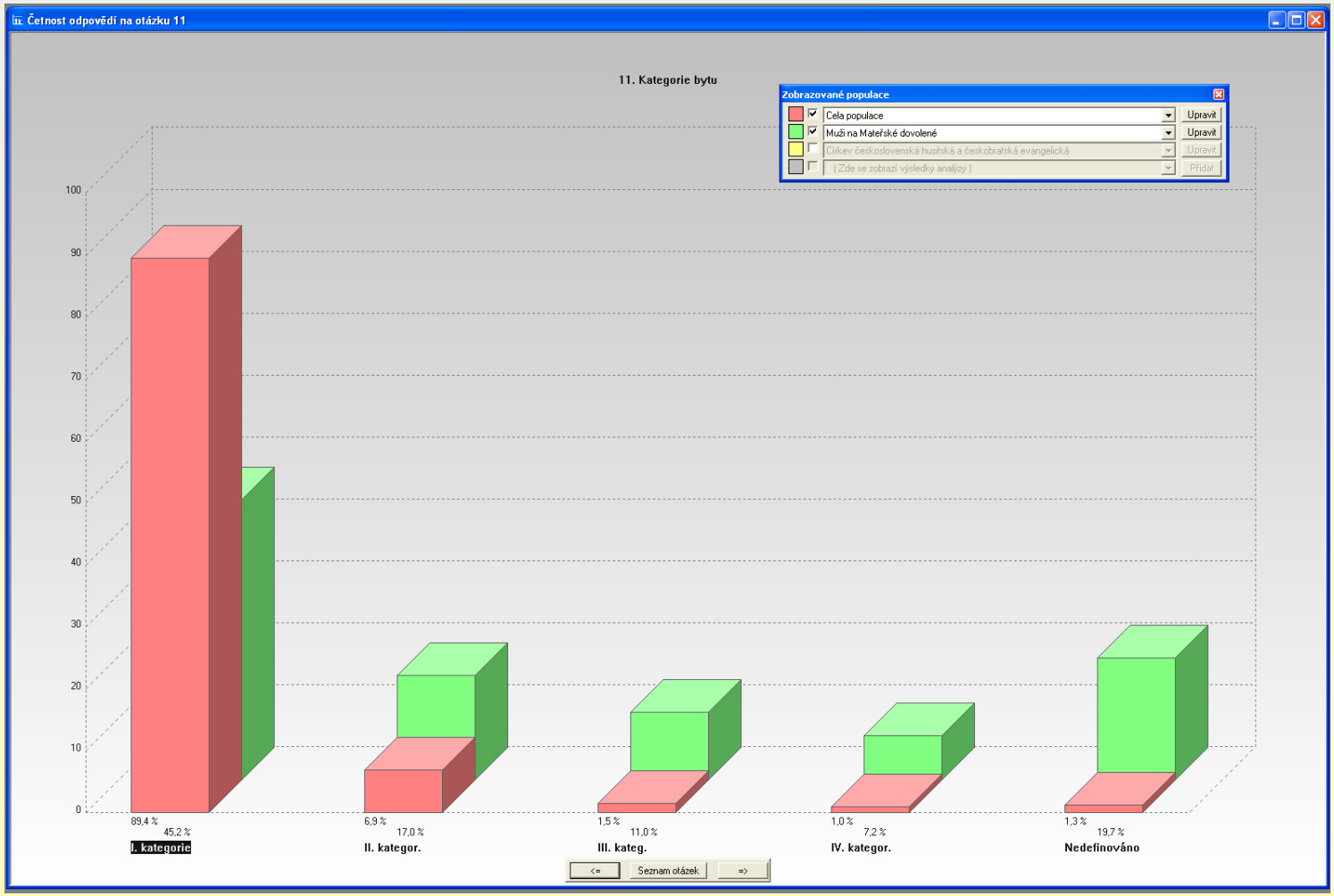
Education of divorced Men in Comparison with Whole Population



Conclusion: Less educated men divorce more frequently.

Example 9: Interactive Statistical Model of Census Data

Flat Quality of Men on Maternity Leave



Conclusion: Men on maternity leave come from poverty-stricken families and probably "solve" a difficult economical situation (unemployment).

Example 10: Probabilistic Neural Networks

Statistical Approach to Pattern Recognition

$\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathcal{X}$: N-dimensional binary data vectors

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_J\}$: finite number of classes

$P(\boldsymbol{x}|\omega)p(\omega), \quad \omega \in \Omega$: conditional distributions of classes

Bayes formula: to classify any given $\boldsymbol{x} \in \mathcal{X}$

$$p(\omega|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega)p(\omega)}{P(\boldsymbol{x})}, \qquad P(\boldsymbol{x}) = \sum_{\omega \in \Omega} P(\boldsymbol{x}|\omega)p(\omega)$$

Probabilistic Neural Networks (PNN, Grim et al. 1999-2012):

approximation of $P(\boldsymbol{x}|\omega)$ by mixtures of product components

$$P(\boldsymbol{x}|\omega) = \sum_{m \in \mathcal{M}_\omega} w_m F(\boldsymbol{x}|m), \quad F(\boldsymbol{x}|m) = \prod_{n \in \mathcal{N}} f_n(x_n|m), \quad \sum_{m \in \mathcal{M}_\omega} w_m = 1.$$

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} f(m)F(\boldsymbol{x}|m), \quad f(m) = p(\omega)w_m, \quad \mathcal{M} = \bigcup_{\omega \in \Omega} \mathcal{M}_\omega$$

Components $\approx$ **Neurons** $\approx$ complete interconnection of neurons

Output Layer: $p(\omega|\boldsymbol{x}) = \sum_{m \in \mathcal{M}_\omega} q(m|\boldsymbol{x}), \quad q(m|\boldsymbol{x}) = \frac{F(\boldsymbol{x}|m)f(m)}{\sum_{j \in \mathcal{M}} F(\boldsymbol{x}|j)f(j)}$

ÚTIA

Example 10: Probabilistic Neural Networks

structural mixture model: **incomplete interconnection**

$$y_m = \log q(m|\boldsymbol{x}) = \log f(m) + \sum_{n\in\mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)} - \log[\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j) f(j)]$$

$q(m\|\boldsymbol{x})$	$\approx$	probability of "spike" given the input pattern $\boldsymbol{x}$
$f(m)$	$\approx$	spontaneous activity of the m-th neuron
$\log \frac{f_n(x_n\|m)}{f_n(x_n\|0)}$	$\approx$	contribution of the input x_n to the activation of m-th neuron
$\log [\sum_{j\in\mathcal{M}} G(\boldsymbol{x}\|j,\phi_j) f(j)]$	$\approx$	common "norming" term (lateral inhibition)

"synaptical weight": $$\log \frac{f_n(x_n|m)}{f_n(x_n|0)} = \log \frac{f_n(x_n|m)}{P_n(x_n)} = \log \frac{q(m|x_n)}{f(m)}$$

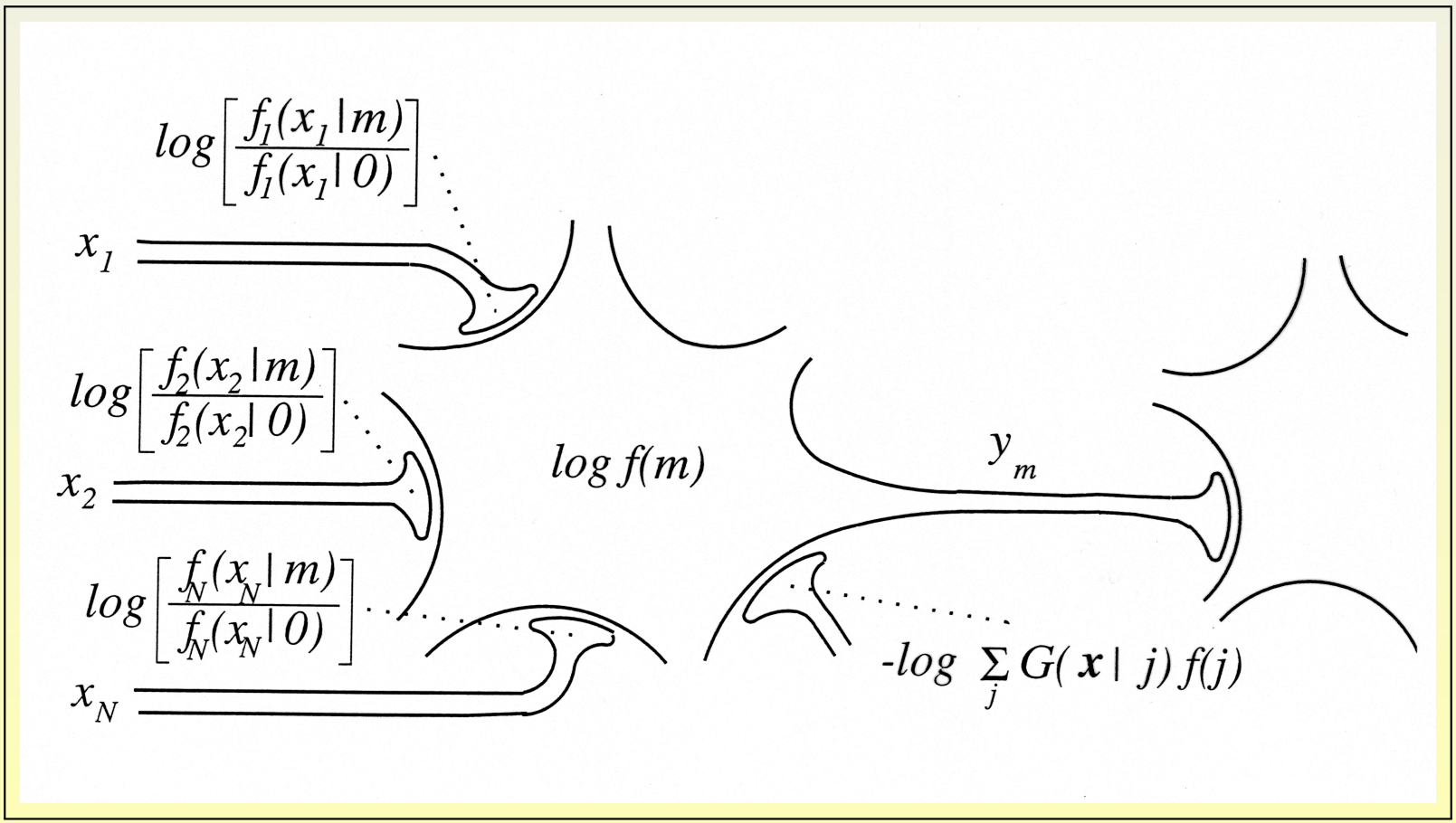
Hebb's postulate of learning (Hebb, 1949)

"When an axon of cell A ($\approx n$) is near enough to excite a cell B ($\approx m$) and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased."

Example 10: Probabilistic Neural Networks

probabilistic neuron: **interpretation of mixture components**

$$y_m = \log q(m|\boldsymbol{x}) = \log f(m) + \sum_{n\in\mathcal{N}} \phi_{mn} \log \frac{f_n(x_n|m)}{f_n(x_n|0)} - \log[\sum_{j\in\mathcal{M}} G(\boldsymbol{x}|j,\phi_j) f(j)]$$



Remark: The structure of PNN can be optimized by EM algorithm.

Literature 1/5

Grim J. (1992): A dialog presentation of census results by means of the probabilistic expert system PES, in *Proceedings of the Eleventh European Meeting on Cybernetics and Systems Research*, Vienna, April 1992, (Ed. R.Trappl), s. 997-1005, World Scientific, Singapore 1992. Paper Award

Grim J., Boček P. (1995): Statistical Model of Prague Households for Interactive Presentation of Census Data, In *SoftStat'95. Advances in Statistical Software 5*, s. 271 - 278, Lucius & Lucius: Stuttgart, 1996.

Grim J., (1996a): Design of multilayer neural networks by information preserving transforms. In: E. Pessa, M.P. Penna, A. Montesanto (Eds.), *Proceedings of the Third European Congress on System Science* (s. 977-982), Roma: Edizzioni Kappa.

Grim J., Pudil P., Somol P. (2000): Recognition of handwritten numerals by structural probabilistic neural networks. In: Proceedings of the Second ICSC Symposium on Neural Computation, Berlin, 2000. (Bothe H., Rojas R. eds.). ICSC, Wetaskiwin, 2000, pp 528-534. Paper Award

Back

Literature 2/5


Grim J. (2000): "Self-organizing maps and probabilistic neural networks". Neural Network World, 3(10): 407-415. Paper Award


Grim J., Boček P., Pudil P. (2001): Safe dissemination of census results by means of interactive probabilistic models. In: *Proceedings of the ETK-NTTS 2001 Conference*, (Hersonissos (Crete), June 18-22, 2001), Vol.2, s. 849-856, European Communities 2001.


Grim J., Haindl M. (2003): Texture Modelling by Discrete Distribution Mixtures. Computational Statistics and Data Analysis, 3-4 **41** 603-615


Grim J., Hora J., Pudil P. (2004): Interaktivní reprodukce výsledků sčítání lidu se zaručenou ochranou anonymity dat. *Statistika*, Vol. 84, No. 5, s. 400-414.


Haindl M., Grim J., Somol P., Pudil P., Kudo M. (2004): A Gaussian mixture-based colour texture model. In: *Proc. of the 17th International Conference on Pattern Recognition.* IEEE, Los Alamitos 2004, s. 177-180.

Back

Literature 3/5


Grim J., Somol P., Haindl M., Pudil P. (2005): A statistical approach to local evaluation of a single texture image. In: Proc. of the 16-th Annual Symposium PRASA 2005. (Nicolls F. ed.). University of Cape Town, 2005, s. 171-176.


Haindl M., Grim J., Pudil P., Kudo M. (2005): A Hybrid BTF Model Based on Gaussian Mixtures. In: Texture 2005. Proceedings of the 4th International Workshop on Texture Analysis. (Chantler M., Drbohlav O. eds.). IEEE, Los Alamitos 2005, s. 95-100.


J. Grim, M. Haindl, P. Somol, and P. Pudil. (2006): A subspace approach to texture modelling by using Gaussian mixtures. In *Proc. of the 18th Int. Conf. ICPR 2006*, Eds. B. Haralick, T.K. Ho), s. 235–238, 2006.


J. Grim, P. Somol, M. Haindl, and P. Pudil, (2006): Color texture segmentation by decomposition of Gaussian mixture model, In *Progress in Pattern Recognition, Image Analysis and Applications*, vol. 19, No. 4225, s. 287–296.


Grim J., Hora J. (2007): Recurrent Bayesian Reasoning in Probabilistic Neural Networks. *Artificial Neural Networks – ICANN 2007*, Ed. Marques de Sá et al., LNCS 4669, s. 129–138, Berlin: Springer

Back

Literature 4/5


Grim, J. (2007): Neuromorphic features of probabilistic neural networks. *Kybernetika.*, 5 **43** 697–712


Grim, J. (2014). Sequential pattern recognition by maximum conditional informativity. *Pattern Recognition Letters*, Vol. 45C, pp. 39-45. *http:// dx.doi.org/10.1016/j.patrec.2014.02.024* Paper Award


Grim J., Hora, J. (2008): Iterative principles of recognition in probabilistic neural networks. *Neural Networks*, Special Issue, 6 **21**, 838–846 Paper Award


Grim J., Hora J., Somol P., Boček P., Pudil, P. (2009): Interaktivní statistický model dat ze sčítání lidu v ČR v r. 2001. Statistika. Roč. 89, č. 4, s. 285-299


Grim J. (2008): Extraction of Binary Features by Probabilistic Neural Networks. In: *Artificial Neural Networks - ICANN 2008 Part II*, Springer: Berlin, LNCS **5164** 52–61


Grim J., Novovičová J., Somol P. (2008): Structural poisson mixtures for classification of documents. ICPR 2008: 1-4, http://dx.doi.org/10.1109/ICPR.2008.4761669

Back

Literature 5/5


Grim J., Somol P., Pudil P., Míková I., Malec M. (2008): Texture Oriented Image Inpainting based on Local Statistical Model. In: Proc. 10th IASTED Conf. on Signal & Image Processing, SIP 2008. Calgary : ACTA Press, 2008 - (Cristea, P.), s. 15-20.


Grim J., Somol P., Haindl M., Danes J. (2009): Computer-Aided Evaluation of Screening Mammograms Based on Local Texture Models. *IEEE Transactions on Image Processing 18(4): 765-773* Paper Award


Grim J., Hora J., Boček P., Somol P. and P. Pudil (2010): Statistical Model of the 2001 Czech Census for Interactive Presentation. *Journal of Official Statistics*. Vol. 26, No. 4, pp. 673–694.


Grim J., Hora, J. (2010): Computational Properties of Probabilistic Neural Networks. In: *Artificial Neural Networks - ICANN 2010 Part II*, Springer: Berlin, LNCS **5164** 52–61


Grim J., Somol P., Pudil P. (2010): Digital Image Forgery Detection by Local Statistical Models. In: Proc. 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Los Alamitos, California, IEEE computer society, 2010 - (Eds. Echizen, I. et al.) s. 579-582. Paper Award

Back

ÚTIA

EM algorithm for Multivariate Bernoulli Mixtures

basic EM algorithm in C++: **mixture of Bernoulli distributions**

```
//    Estimation of Multivariate Bernoulli Mixture by means of EM algoritmu
//=============================================================================
//short     X[NN];                  //  binary data vector
//int       NN;                     //  dimension of binary vectors
//int       MM;                     //  number of mixture components
//double    P[MM][NN],SP[MM][NN];   //  mixture parameters and related estimates
//double    W[MM],SW[MM];           //  component weights and related estimates
//double    FX[MM];                 //  component values for a given vector X[NN]
//double    FXM,SWM,Q,SUM,SWM;      //  auxiliary variables
//int       N,M,IT,ITERMAX;         //  auxiliary variables

for(IT=1; IT<=ITERMAX; IT++)
//*****************************************************************************
{  for(M=0; M<MM; M++) {SW[M]=0.0; for(N=0; N<NN; N++) SP[M][N]=0.0;}
   Q=0.0;
   for(J=1;J<=JJ;J++)                //  cycle over all data vectors X
   { READ(X);  SUM=0.0;              //  to read X from the input data set
     for(M=0; M<MM; M++)
     {  FXM=W[M];
        for(N=0; N<NN; N++) if(X[N]==1) FXM*=P[M][N]; else FXM*=(1-P[M][N]);
        FX[M]=FXM;  SUM+=FXM;
     }  // end of M-loop
     Q=Q+log(SUM);
     for(M=0; M<MM; M++)
     {  G=FX[M]/SUM; SW[M]+=G;  for(N=1; N<=NN; N++) if(X[N]==1) SP[M][N]+=G;
     }  //  end of M-loop
   }  //  end of J-loop
   Q=Q/JJ;
   for(M=0; M<MM; M++)              //  to compute the new parameter estimates
   {  SWM=SW[M];   W[M]=SWM/JJ;  for(N=0; N<NN; N++) P[M][N]=SP[M][N]/SWM;
   }  // end of M-loop
   print(IT,Q);
}  //  end of IT-loop
//*****************************************************************************
printf("\n End of the EM algorithm\n\n");
```

Remark: In case of small dimension NN only !.

Back

EM algorithm for Gaussian Product Mixtures

basic EM algorithm in C++: **multidimensional Gaussian product mixture**

```
//   Estimation of the Gaussian product mixture by means of EM algorithm
//=============================================================================
//int IT,N,M;  long K;  double F,G,FXM,SWM,SUM,FMAX,Q0;    // global variables
//double   X[DNN];                      // real data vector
//double   FX[DMM],W[DMM],SW[DMM];      // components, weights, weight estimates
//double   C[DMM][DNN], A[DMM][DNN];    // component means and variances
//double   SC[DMM][DNN],SA[DMM][DNN];   // new estimates of means and variances
for(IT=1; IT<=ITMAX; IT++)              // iteration loop
//*****************************************************************************
{ Q=0.0
  for(M=1; M<=MM; M++)                  // logarithmic parameters and initial values
  {  SW[M]=RMIN;           F=log(W[M]+RMIN)-NN2LN2PI;
     for(N=1; N<=NN; N++) {F-=log(A[M][N]); SC[M][N]=RMIN; SA[M][N]=RMIN;}
     W[M]=2*F;                          // to simplify the evaluation of exponents
  }  // end of M-loop
  for(I=1;I<=K;I++)                     // cycle over all data vectors X
  {  READ(X);  FMAX=-RMAX;
     for(M=1; M<=MM; M++)               // evaluation of the logarithm of components
     {  FXM=W[M]; for(N=1; N<=NN; N++) {F=(X[N]-C[M][N])/A[M][N]; FXM-=F*F;}
        FXM/=2.0f;    FX[M]=FXM;  if(FXM>FMAX) FMAX=FXM;
     } // end of M-loop
     SUM=0.0;
     for(M=1; M<=MM; M++)               // to compute the component values and P(X)
     {  FXM=FX[M]-FMAX;  if(FXM>MINLOG) {FXM=exp(FXM); SUM+=FXM;} else FXM=0.0;
        FX[M]=FXM;
     }  // end of M-loop
     Q=Q+log(SUM)+FMAX;                 // to compute the log-likelihood criterion
     for(M=1; M<=MM; M++)
     {  G=FX[M]/SUM;  SW[M]+=G;
        for(N=1; N<=NN; N++) {F=X[N]; SC[M][N]+=G*F; SA[M][N]+=G*F*F;}
     }  //  end of M-loop
  }  //  end of K-loop
  Q/=K;
  for(M=1; M<=MM; M++)                  //  to compute the new parameter estimates
  {  SWM=SW[M];   W[M]=SWM/K;
     for(N=1; N<=NN; N++)
     {  F=SC[M][N]/SWM;   C[M][N]=F;  A[M][N]=sqrt(SA[M][N]/SWM-F*F);
     }  //  end of N-loop
  }   //  end of M-loop
  printf("\nIT=%2d   Q=%15.7lf  \n",IT,Q);
  //***************************************************************************
} //  end of IT-loop
```

Back

Example 3: Classification of Text Documents

log-likelihood criterion:

$$L = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} \log[\sum_{m \in \mathcal{M}_c} G(\boldsymbol{x}|\boldsymbol{\theta}_m, \phi_m) w_m], \qquad \mathcal{S}_c = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K_c}\}$$

EM algorithm:

$$q(m|\boldsymbol{x}) = \frac{G(\boldsymbol{x}|\boldsymbol{\theta}_m, \phi_m) w_m}{\sum_{j \in \mathcal{M}_c} G(\boldsymbol{x}|\boldsymbol{\theta}_j, \phi_j) f(j)}, \qquad m \in \mathcal{M}_c,\ n \in \mathcal{N},\ \boldsymbol{x} \in \mathcal{S}_c$$

$$\tilde{x}_n^{(m)} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} x_n q(m|\boldsymbol{x}), \qquad |\bar{\boldsymbol{x}}|^{(m)} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} |\boldsymbol{x}| q(m|\boldsymbol{x})$$

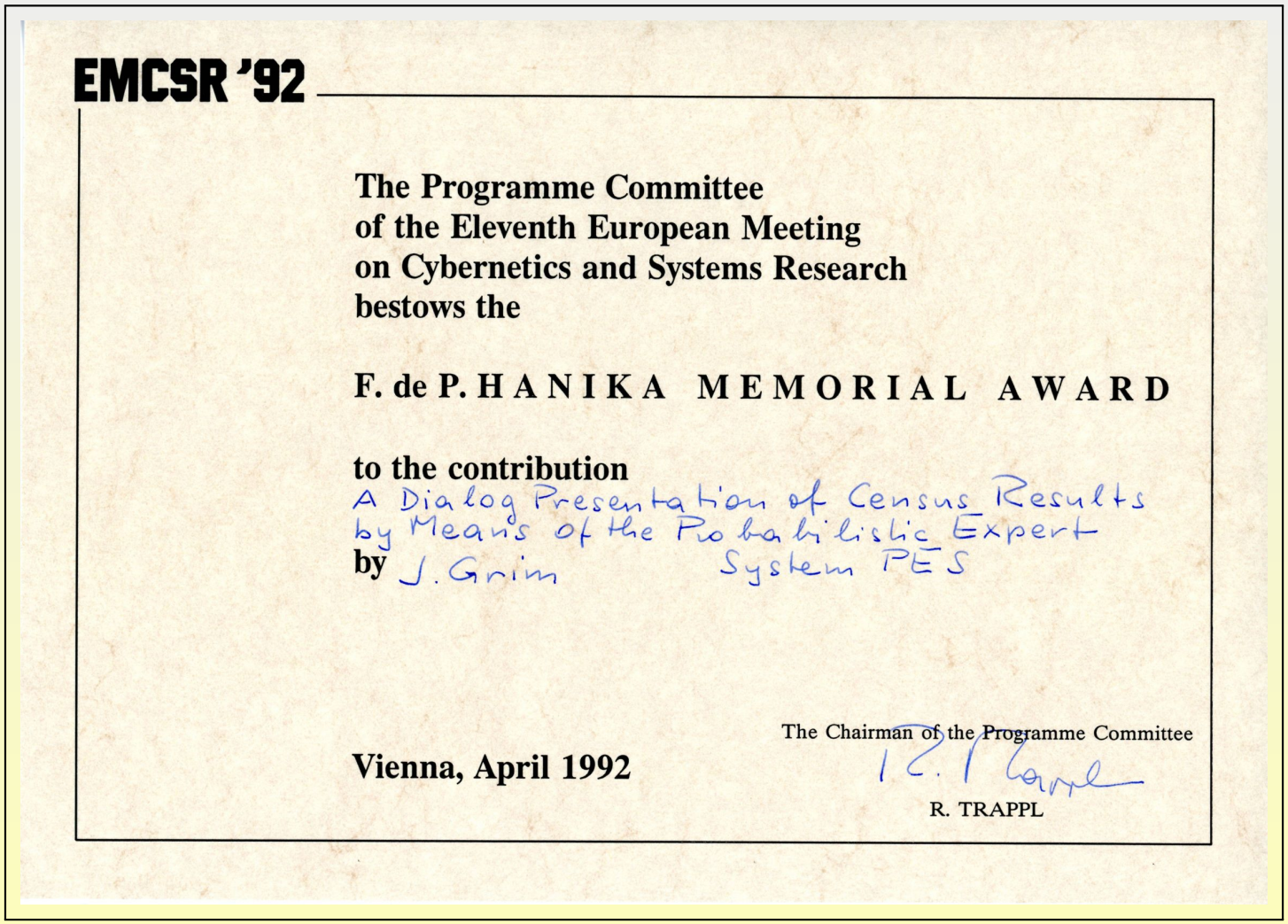
$$w_m^{'} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} q(m|\boldsymbol{x}), \qquad \theta_{mn}^{'} = \frac{\tilde{x}_n^{(m)}}{|\bar{\boldsymbol{x}}|^{(m)}}$$

$$\phi_{mn}^{'} = \begin{cases} 1, & \gamma_{mn}^{'} \in \Gamma_r^{'}, \\ 0, & \gamma_{mn}^{'} \notin \Gamma_r^{'}, \end{cases}, \qquad \gamma_{mn}^{'} = \tilde{x}_n^{(m)} \log \frac{\theta_{mn}^{'}}{\theta_{0n}} + |\bar{\boldsymbol{x}}|^{(m)} (\theta_{0n}^{'} - \theta_{mn})$$

$\Gamma_r^{'}$ is the set of r largest values $\gamma_{mn}^{'}$

Back

Paper Award



EMCSR '92

The Programme Committee
of the Eleventh European Meeting
on Cybernetics and Systems Research
bestows the

F. de P. HANIKA MEMORIAL AWARD

to the contribution
A Dialog Presentation of Census Results
by Means of the Probabilistic Expert
by J. Grim System PES

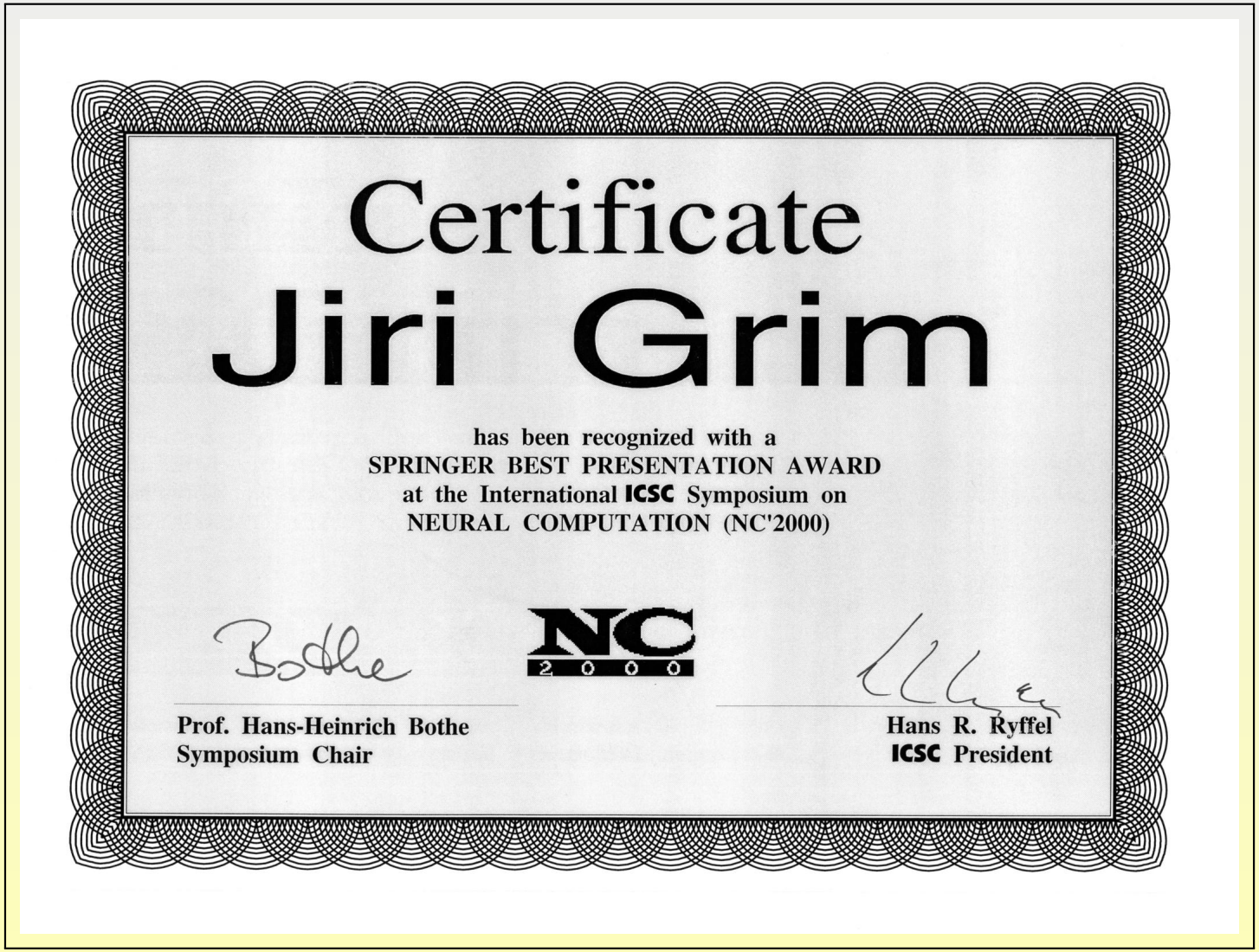
Vienna, April 1992

The Chairman of the Programme Committee
R. TRAPPL

Eleventh European Meeting on Cybernetics and Systems Research, Vienna, April 1992

Back

Paper Award



Certificate

Jiri Grim

has been recognized with a
SPRINGER BEST PRESENTATION AWARD
at the International ICSC Symposium on
NEURAL COMPUTATION (NC'2000)

NC 2000

Prof. Hans-Heinrich Bothe
Symposium Chair

Hans R. Ryffel
ICSC President

Second ICSC Symposium on Neural Computation, Berlin, 2000

Back

Paper Award

Ústav teorie informace a automatizace
Akademie věd České republiky v.v.i.

Uděluje v soutěži o Nejlepší publikaci 2009

2. cenu

Jiřímu Grimovi

za práci

Computer-Aided Evaluation of Screening Mammograms Based on Local Texture Models

IEEE Trans. on Image Processing, vol. 18, No. 4, pp. 765–773, ISSN 1057–7149, April 2009.

(spoluautoři Somol P., Haindl M., Daneš J.)

ředitel ústavu

IEEE Transactions on Image Processing 18(4): 765-773, 2009

Back

Paper Award

Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing
IIH-MSP-2010
October 15-17, 2010, Darmstadt, Germany

Best Paper Award Certificate

in recognition of the professional excellence of the paper

Digital Image Forgery Detection by Local Statistical Models

by

Jiri Grim, Petr Somol

&

Pavel Pudil

Xiamu Niu
General Chair
Harbin Institute of Technology
China

Alexander Nouak ,
Conference executive Chair
Fraunhofer IGD
Germany

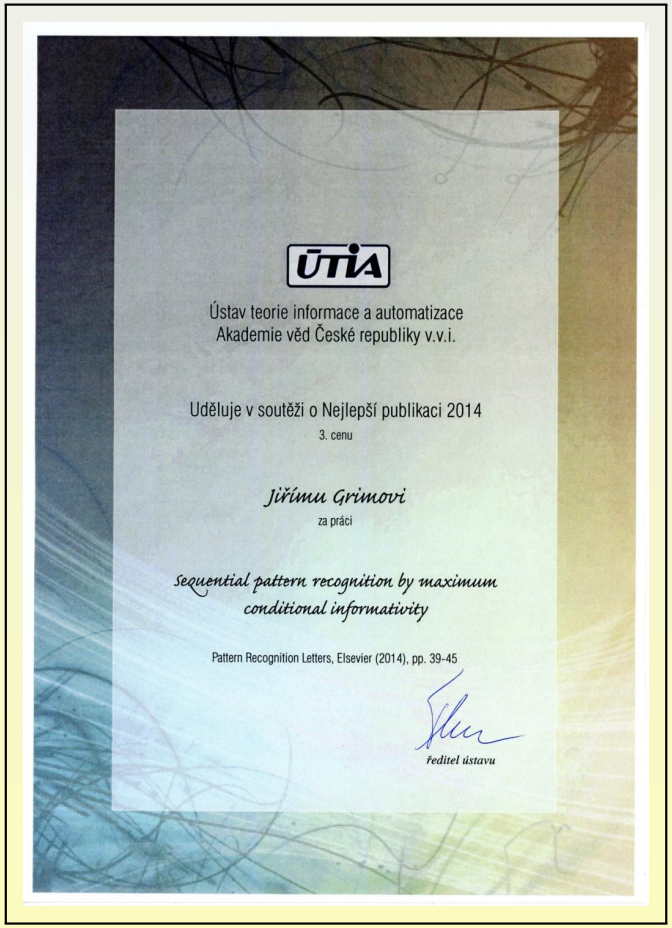
Jeng-Shyang Pan
Program Committee Chair
National Kaohsiung University of Applied Sciences
Taiwan

Isao Echizen,
Program Committee Chair
National Institute of Informatics
Japan

Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP Darmstadt, 2010

Back

Paper Award



Pattern Recognition Letters, Vol. 45C, pp. 39-45, 2014

Back

Paper Award

ÚTIA

Ústav teorie informace a automatizace
Akademie věd České republiky

Uděluje v soutěži o Nejlepší publikaci 2008

2. cenu

Jiřímu Grimovi

za práci

Iterative principles of recognition in probabilistic neural networks

Neural Networks 21 (2008), 838-846.
(spoluautor Hora J.)

ředitel ústavu

Neural Networks, 21(6): 838–846, 2008

Back

Paper Award

ÚTIA

Ústav teorie informace a automatizace
Akademie věd České republiky

Cena za nejlepší článek

udělena v roce 2000

Jiřímu Grimovi

za práci

Self-Organizing Maps
and Probabilistic Neural Networks

Neural Network World No. 3, 2000, 407-415

ředitel ústavu

Neural Network World, 3(10): 407-415, 2000

Back