Minimum Description Length Principle: between Theory and Practice

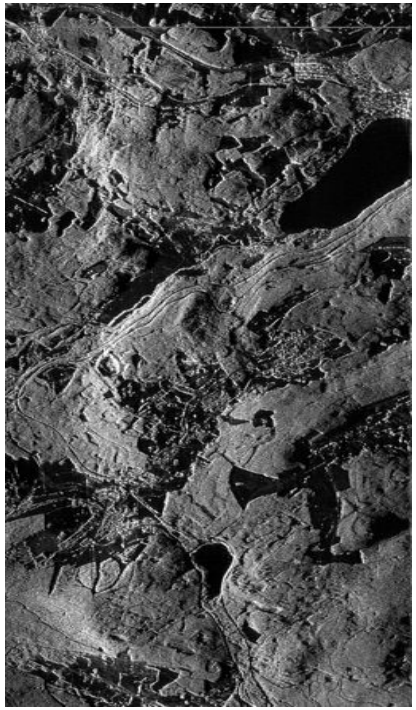
”Theory without practice is empty, practice without theory is blind”

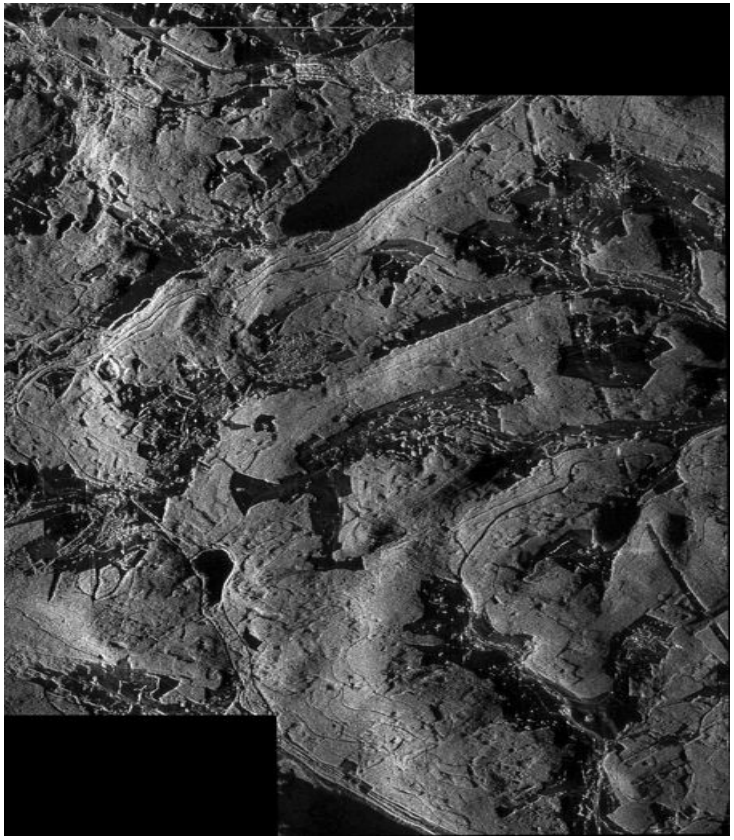
Prof. Alexey Potapov
ITMO University, Saint-Petersburg State University, AIDEUS
2015

AGI’15 @ Berlin

One practical task: image matching

- How to find correspondence between pixels of two images of the same scene?





Simplest approach: correlation

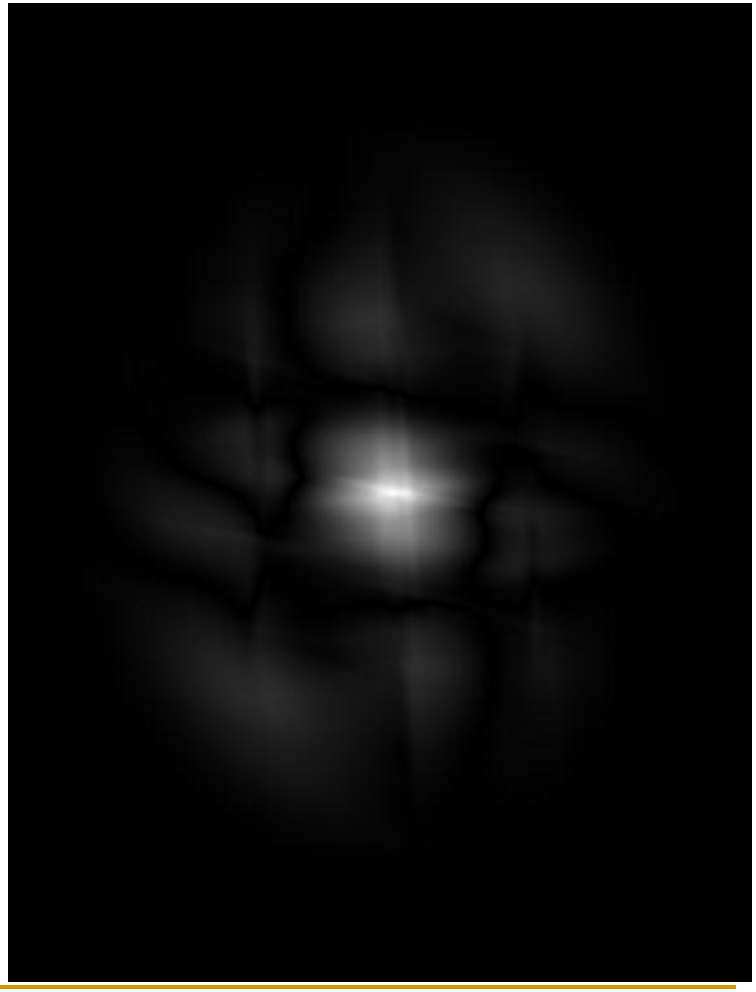
Least squares error

$$\varepsilon_{f_1,f_2}(\Delta x,\Delta y)=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left[f_1(x,y)-f_2(x+\Delta x,y+\Delta y)\right]^2$$

Correlation

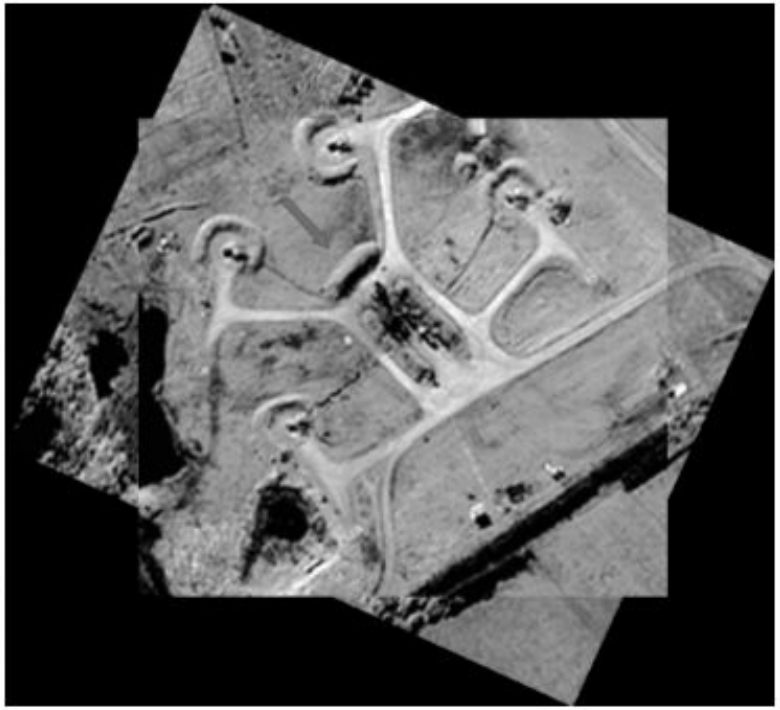
$$C_{f_1,f_2}(\Delta x,\Delta y)=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f_1(x,y)\,f_2(x+\Delta x,y+\Delta y)$$

Slightly more advanced: cross-correlation function calculated via Fourier Transform



Fourier-Mellin Transform

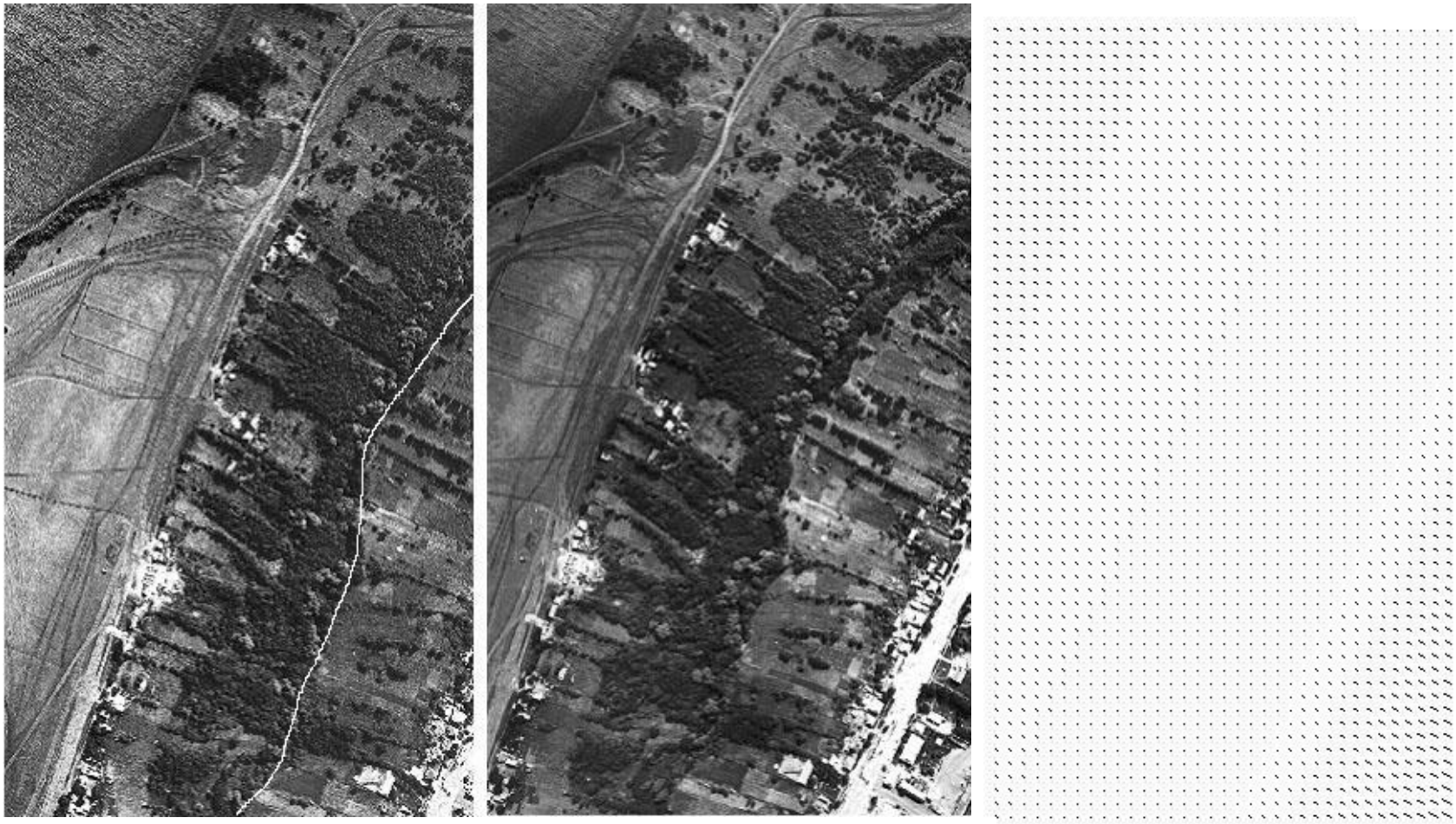
1. Amplitude spectrum
2. Log-polar transform
3. Cross-corr. Via Fourier
4. Find scale/ rotation
5. Compensate scale/ rotation
6. Cross.corr. to find shifts
7. Success!



Block Matching: Local displacement extension

1. Take local fragments around different points of pre-aligned images
2. Match them by correlation
3. Construct local displacement field

Resulting displacement field

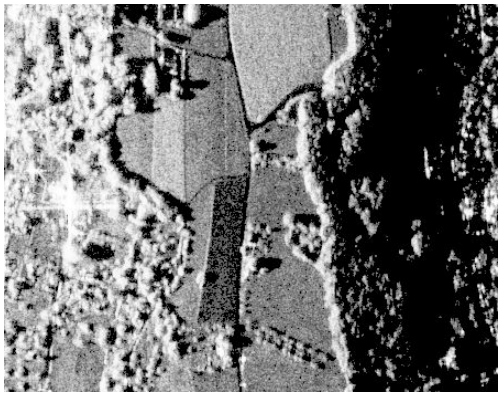


General solution for aerospace image matching!?

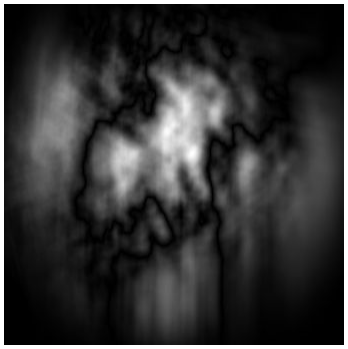
However…

Optical image

SAR image

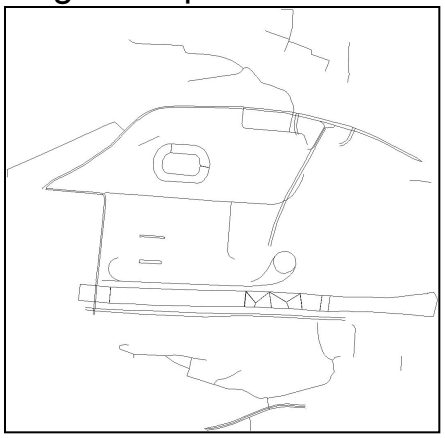


Cross-correlation field



Optical image

Digital map



Correlation?

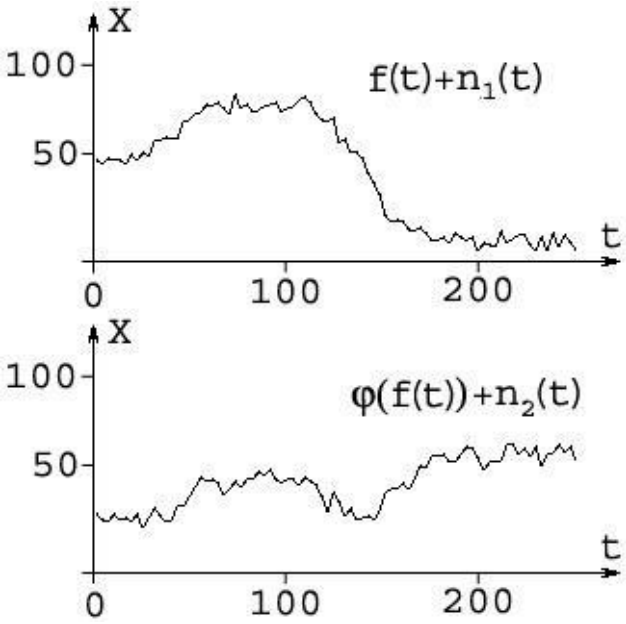
Many applications require matching images of different modalities

Criterion: Mutual Information

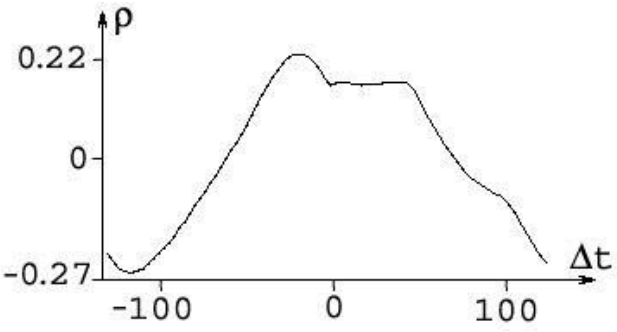
Mutual information $\mathrm{I}(X,Y) = E_{XY}\left[I(x;y)\right] = \sum_{x\in \mathrm{X}}\sum_{y\in \mathrm{Y}} P(x,y)\log_2 \frac{P(x,y)}{P(x)P(y)}$

No correlation => $E_{XY}\left[XY\right] - E_X\left[X\right]E_Y\left[Y\right] = 0$

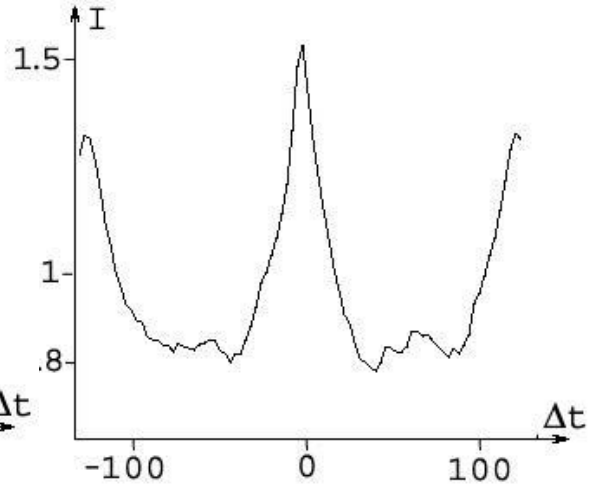
No mutual information => $E_{XY}\left[\varphi_1(X)\varphi_2(Y)\right] - E_X\left[\varphi_1(X)\right]E_Y\left[\varphi_2(Y)\right] = 0$



Cross correlation: degraded maximum



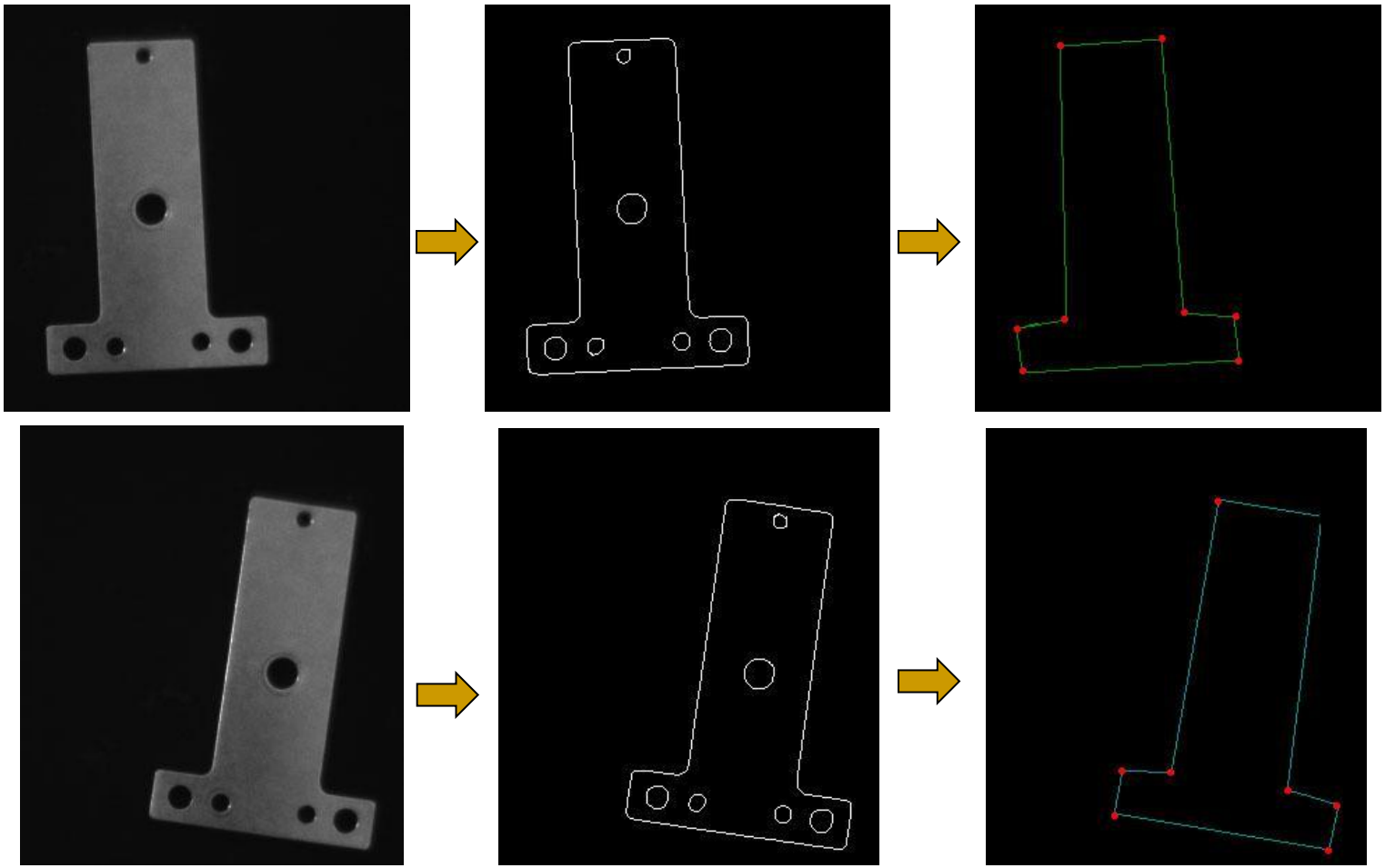
Mutual information: Ideal maximum



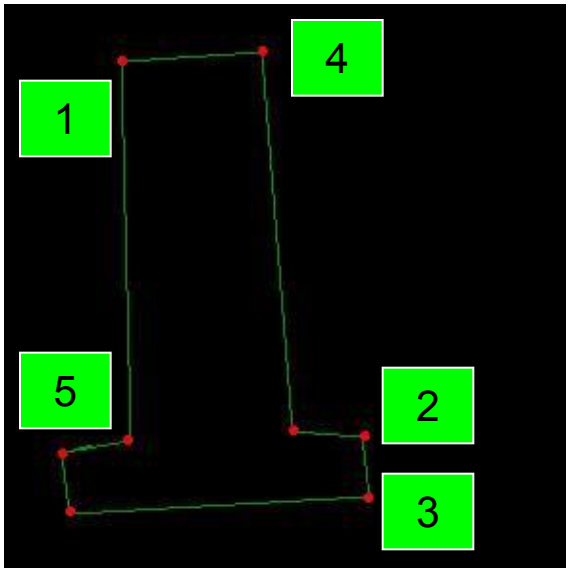
Unfortunately, it's difficult to compute and not applicable to vector maps

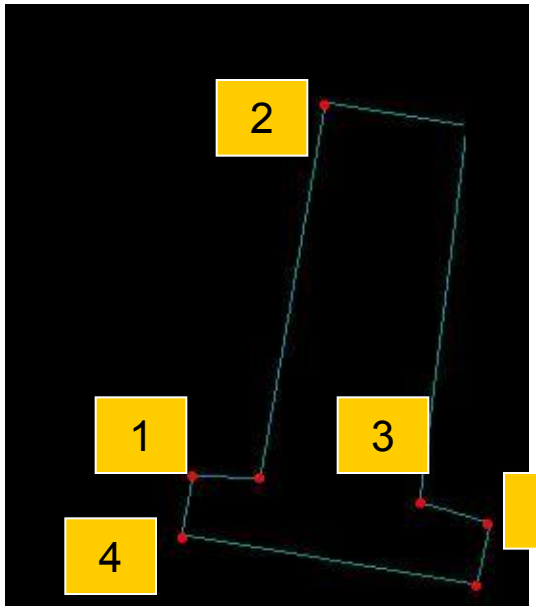
Viola P.A. Alignment by Maximization of Mutual Information: PhD thesis, MIT, Cambridge, Massachusetts. 1995. 156 p.

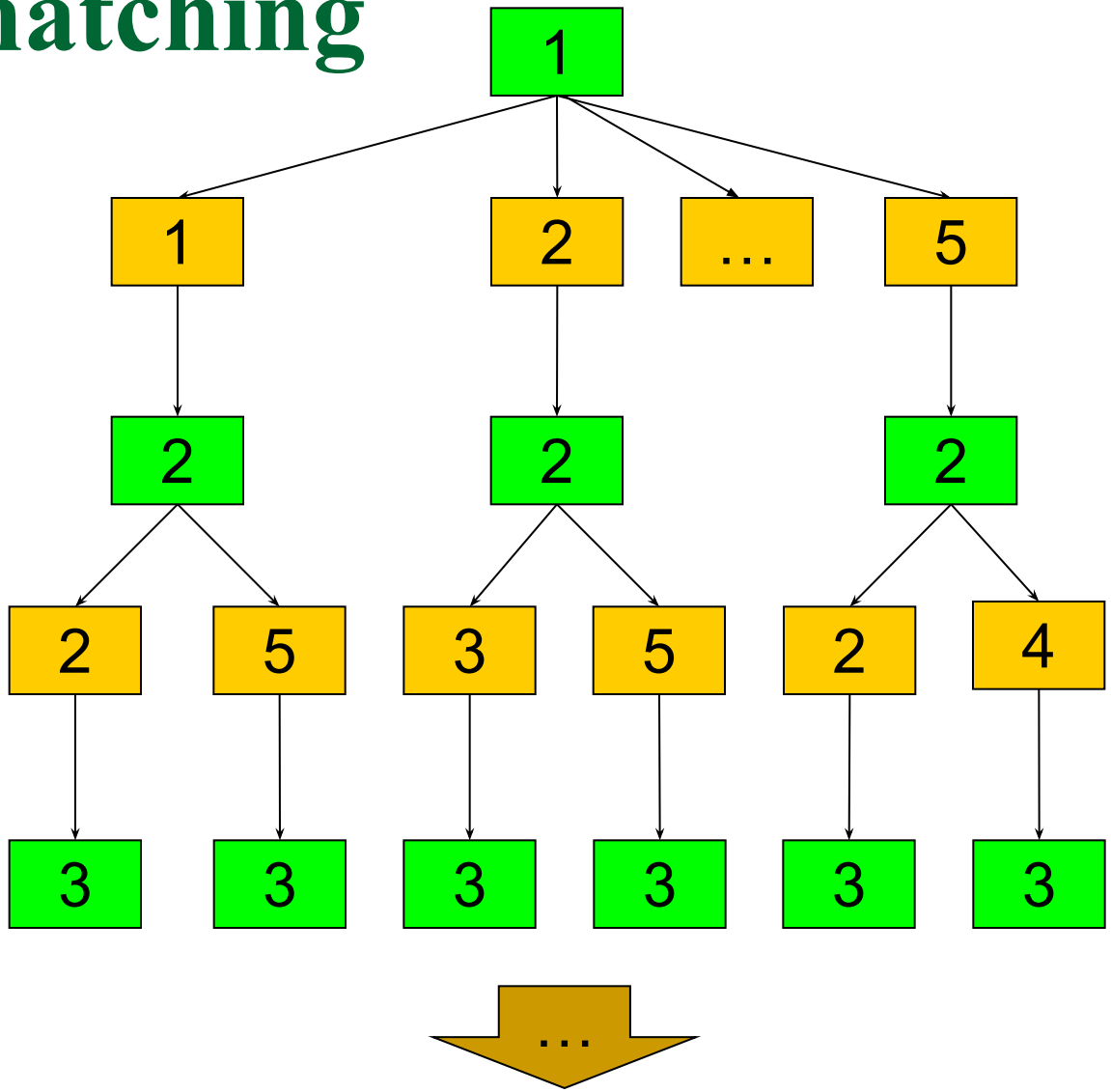
Invariant structural descriptions



Structural matching







More questions…

-How to estimate quality of structural correspondence?

-How to choose the group of transformations if it is not known?

-How to construct contours and structural elements optimally?

-How to choose the most adequate number of contours and structural elements?

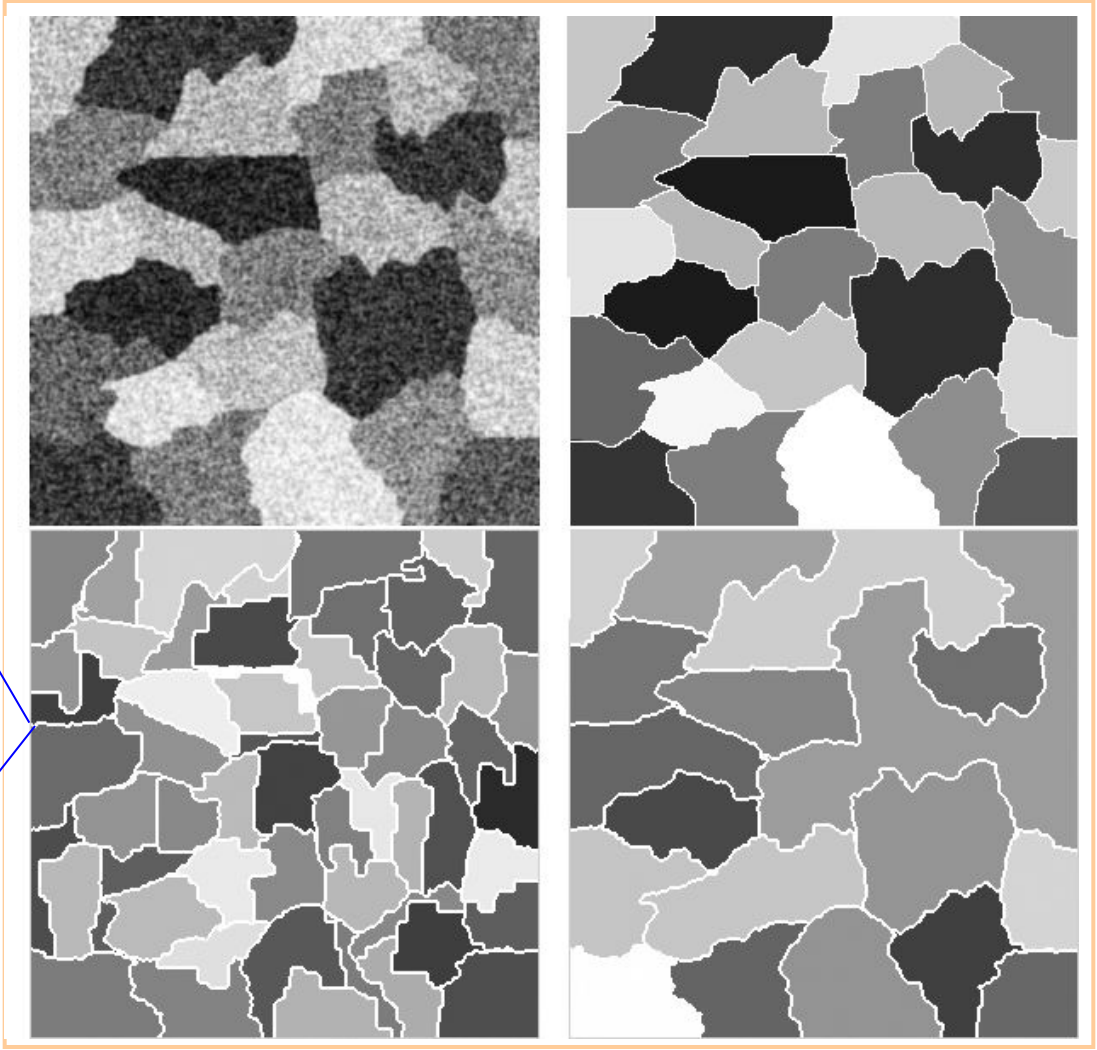
- Are precision criteria such as mean square error suitable? Or have they the same shortcomings as correlation?

MSE criterion: oversegmentation

Each region is described by average value

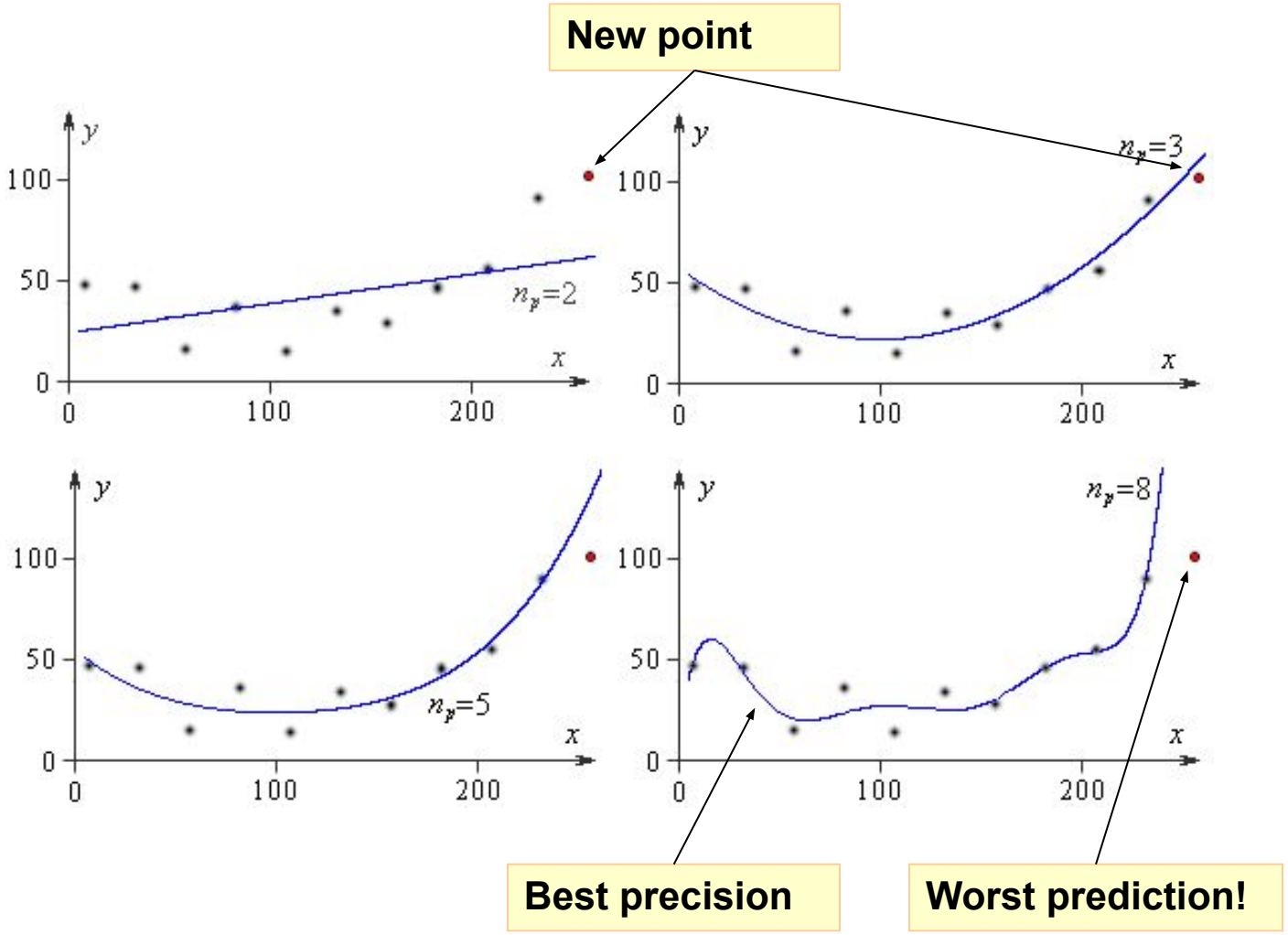
More precise

Over-segmentation!



Correct, but not the most precise description!

Functional approximation



Information-theoretic criterion

Again, criteria from information theory help:

- Mutual information can be extended for the task of matching structural elements
- In general, the minimum description length can be used for model selection

The best model is the model that minimizes the sum

- the description length (in bits) of the model,
- the description length (in bits) of data encrypted with help of the model (deviation of data from model).

Connection to Bayes' rule

Bayes rule: $P(H \mid D) = \dfrac{P(D \mid H)P(H)}{P(D)}$

- Posterior probability: $P(H \mid D)$
- Prior probability: $P(H)$
- Likelihood: $P(D \mid H)$

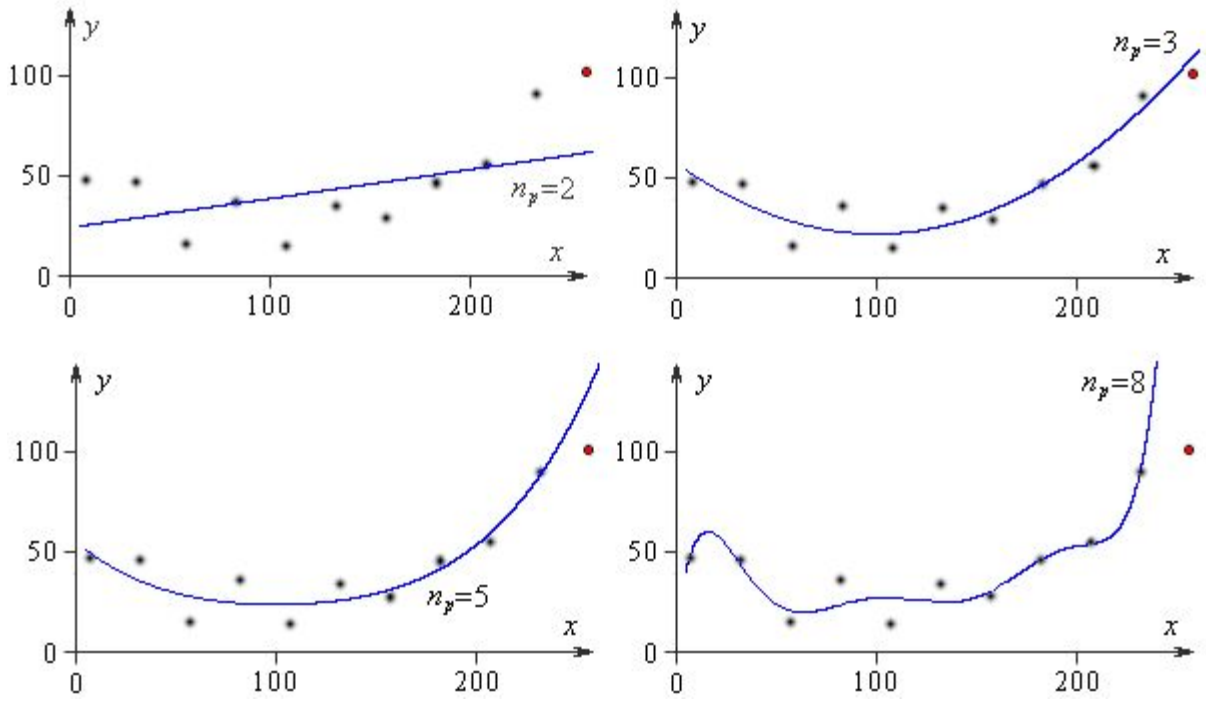
$$H^* = \arg\max_H P(H \mid D) = \arg\max_H P(H)P(D \mid H)$$

$$= \arg\min_H -\log P(H) - \log P(D \mid H)$$

- The description length of the model: –log $P(H)$
- The description length of data encrypted with the help of the model: –log $P(D \mid H)$

Application to function approximation

Too simple model



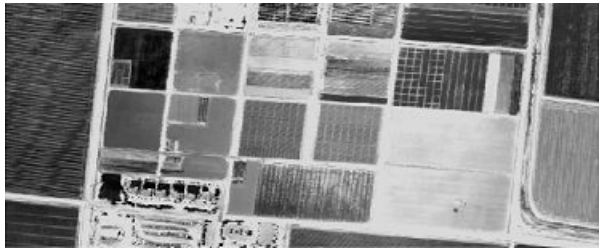
The best model is chosen as trade-off between precision and complexity

Too complex model

$$L = \underbrace{\frac{n_p}{2}\log_2 n}_{l(H)} + \underbrace{\frac{n}{2}\log_2 \frac{\varepsilon^2(\mathbf{w})}{n}}_{K(D|H)}$$

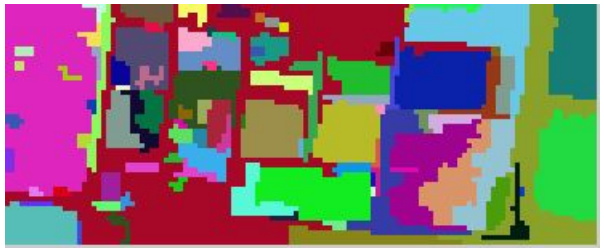
$$K(D \mid H) = -\log_2 P(D \mid H) = -\log_2 \prod P(\varepsilon_i(\mathbf{w}))$$

Application to image segmentation



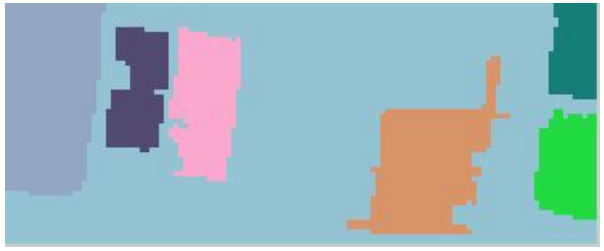
Initial image

N_{gr}=300; DL=4,5e+5



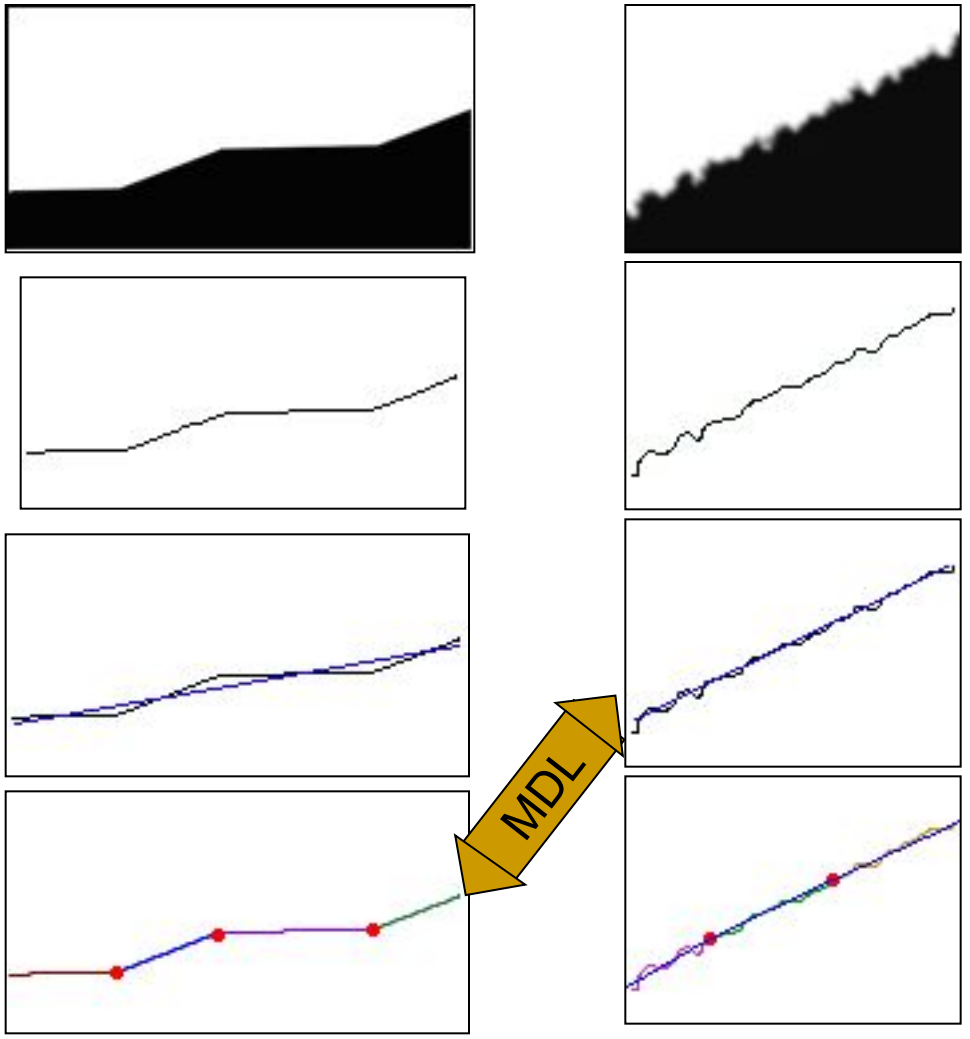
N_{gr}=100; DL=3,8e+5

N_{gr}=37; DL=3,7e+5



N_{gr}=7; DL=3,9e+5

Contour segmentation



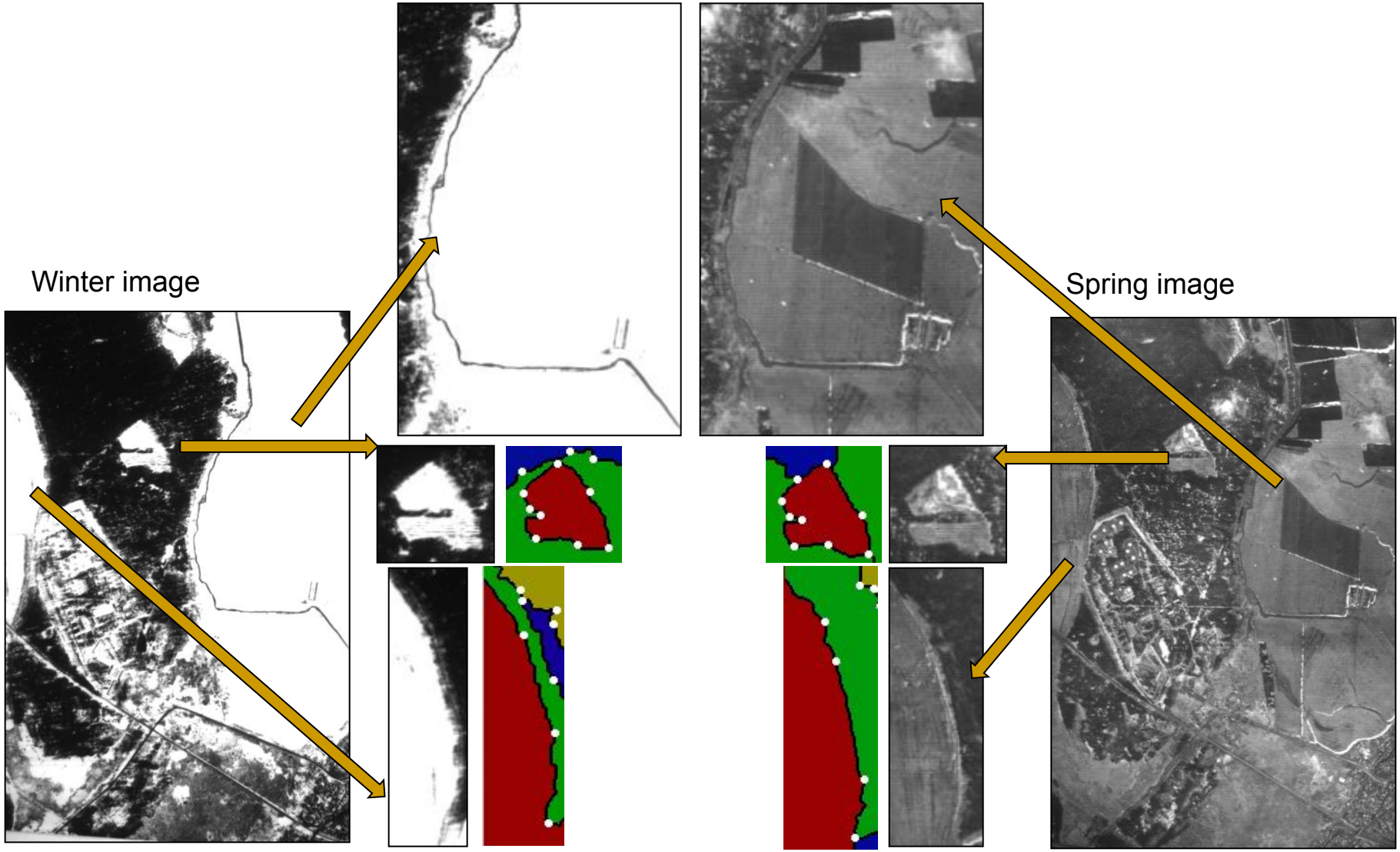
Images

Extracted contours

MSE-approximation with high threshold on dispersion

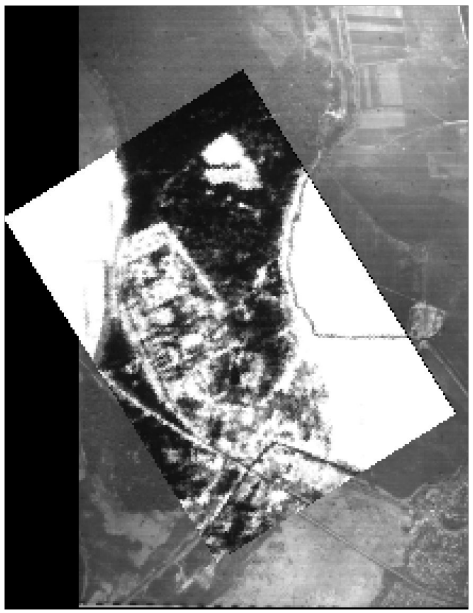
MSE-approximation with low threshold on dispersion

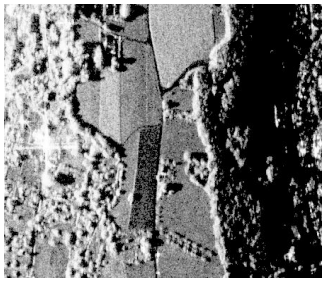
Full solution of invariant image matching

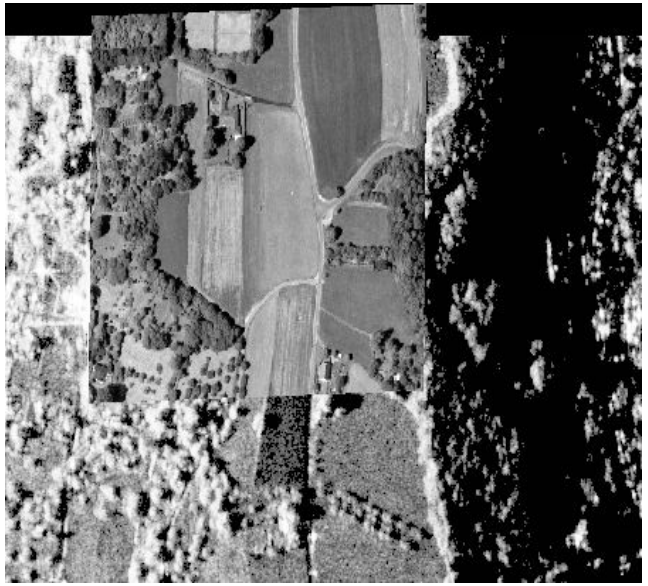


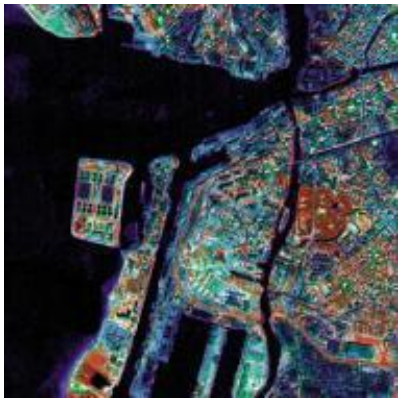
Potapov A.S. Image matching with the use of the minimum description length approach // Proc. SPIE. 2004. Vol. 5426. P. 164–175.

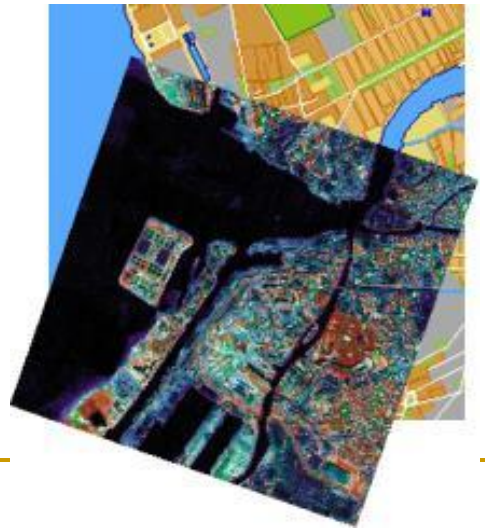
Successful matching





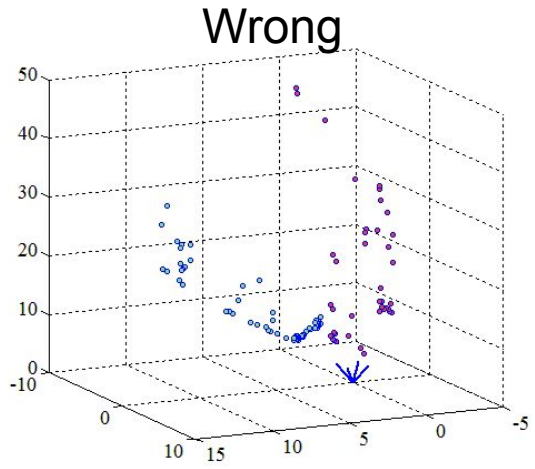


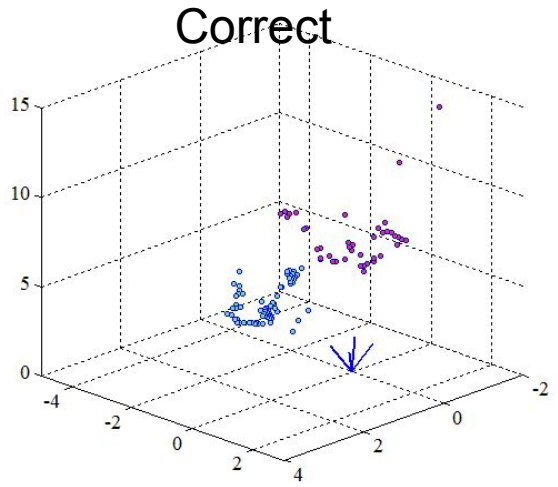




More applications of MDL

Correct separation into clusters for keypoint matching in dynamic scenes





Essential for correct estimation of a dynamic scene structure

A.N. Averkin, I.P. Gurov, M.V. Peterson, A.S. Potapov. Spectral-Differential Feature Matching and Clustering for Multi-body Motion Estimation // Proc. MVA2011 IAPR Conference on Machine Vision Applications. 2011. June 13-15, Nara, Japan. P. 173–176.

Various applications of MDL

- Pattern recognition, etc.:
  - Support-vector machines;
  - Discrimination functions;
  - Gaussian mixtures;
  - Decision forests;
  - ICA (as a particular case of MDL)
  - …
- Image analysis
  - Segmentation;
  - Object recognition and image matching;
  - Optical flow estimation;
  - Structural description of images;
  - Changes detection;
  - …
- Learning in symbolic domains, etc.

But wait… what about theory?

- MDL principle is used loosely
- Description lengths are calculated within heuristically defined coding schemes
- Success of a method is highly determined by the utilized coding scheme
- Is there some theory that overcomes this arbitrariness?

The theory behind MDL

- Algorithmic information theory

$$K_U(D) = \min_H [l(H) \mid U(H) = D], \quad H^* = \arg\min_H [l(H) \mid U(H) = D]$$

  - U – universal Turing machine
  - K – Kolmogorov complexity,
  - $l(H)$ – length of program H
  - H^* – best description/model of data D

- Two-part coding:

$$H^* = \arg\min_H [l(H) + K(D \mid H)]$$ if full model is separated into two parts

OR

$$H^* = \arg\min_H [l(H) + \log P(U(H) = D)]$$ if H is probabilistic program

- UTM defines the universal model space

Universal prediction

- Solomonoff's algorithmic probabilities
  - Prior probability

$$P_U(\alpha) = \sum_{p:U(p)=\alpha} 2^{-l(p)}$$

  - Predictive probability

$$P_U(\alpha' \mid \alpha) = P_U(\alpha\alpha') / P_U(\alpha)$$

- Universal distribution of prior probabilities dominates (with multiplicative factor) over any other distribution
- Bayesian prediction with the use of these priors converges in limit with prediction based on usage of true distribution

Solomonoff, R.: Algorithmic Probability, Heuristic Programming and AGI. In: Baum, E., Hutter, M., Kitzelmann, E. (eds). Advances in Intelligent Systems Research, vol. 10 (proc. 3rd Conf. on Artificial General Intelligence), pp. 151–157 (2010).

Universality of the algorithmic space

3.1415926535 8979323846 2643383279 5028841971 6939937510 5820974944 5923078164 0628620899 8628034825 3421170679 8214808651 3282306647 0938446095 5058223172 5359408128 4811174502 8410270193 8521105559 6446229489 5493038196 4428810975 6659334461 2847564823 3786783165 2712019091 4564856692 3460348610 4543266482 1339360726 0249141273 7245870066 0631558817 4881520920 9628292540 9171536436 7892590360 0113305305 4882046652 1384146951 9415116094 3305727036 5759591953 0921861173 8193261179 3105118548 0744623799 6274956735 1885752724 8912279381 8301194912 9833673362 4406566430 8602139494 6395224737 1907021798 6094370277 0539217176 2931767523 8467481846 7669405132 0005681271 4526356082 7785771342 7577896091 7363717872 1468440901 2249534301 4654958537 1050792279 6892589235 4201995611 2129021960 8640344181 5981362977 4771309960 5187072113 4999999 ……

```
int a=10000,b,c=8400,d,e,f[8401],g;
main() {for(;b-c;)f[b++]=a/5;
for(;d=0,g=c*2;c-=14, printf("%.4d",e+d/a),e=d%a)
for(b=c;d+=f[b]*a,f[b]=d%--g,d/=g--,--b;d*=b);}
```

By D.T. Winter

Grue Emerald Paradox

- Hypothesis No. 1: all emeralds are green
- Hypothesis No. 2: all emeralds are greu
  (that is green before 2050, and blue after this time)

- Likelihood of observation data equals
- How can we calculate prior probabilities of these two hypotheses?

Is it possible to ground prior probabilities?

- Probability theory allows to deduce one probability from another. But what are the initial probabilities?
- Universal priors work

Solomonoff R. Does Algorithmic Probability Solve the Problem of Induction? // Oxbridge Research, P.O.B. 391887, Cambridge, Mass. 02139. 1997.

Methodological usefulness

- Theory of universal induction answers the questions
  - What is the source of overlearning/ overfitting/ oversegmentation, etc.
  - Why is any new narrow learning method “yet another classifier”
  - Why are feed forwards neural networks not really “universal approximators”
  - And at the same time, why is “no free lunch theorem” not true

Gap between universal and pragmatic methods

- Universal methods
  - can work in arbitrary computable environment
  - incomputable or computationally infeasible
  - approximations are either inefficient or not universal
- Practical methods
  - work in non-toy environments
  - set of environments is highly restricted

=> Bridging this gap is necessary

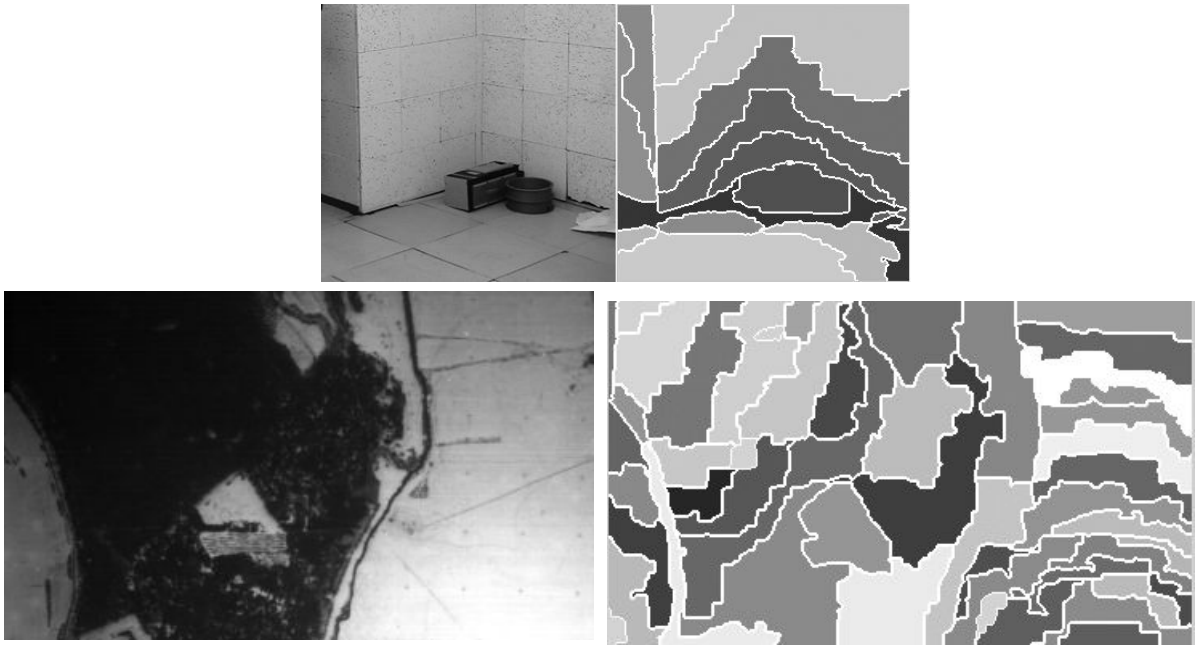
Choice of the reference UTM

- Unbiased AGI cannot be practical and efficient
- Dependence of the algorithmic probabilities on the choice of UTM appears to be very useful in order to put any prior information and to reduce necessary amount of training data
- UTM contains prior information

  => UTM can be optimized to account for posterior information

Limitations of narrow methods

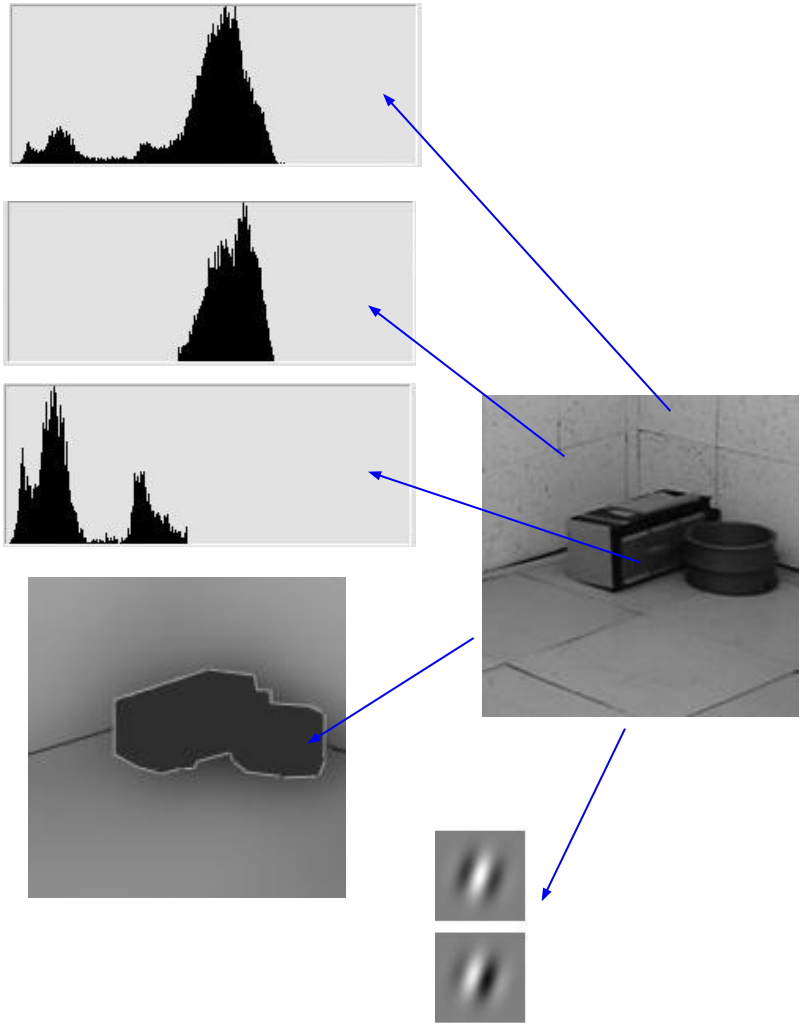
- Brightness segmentation can fail even with the MDL criterion



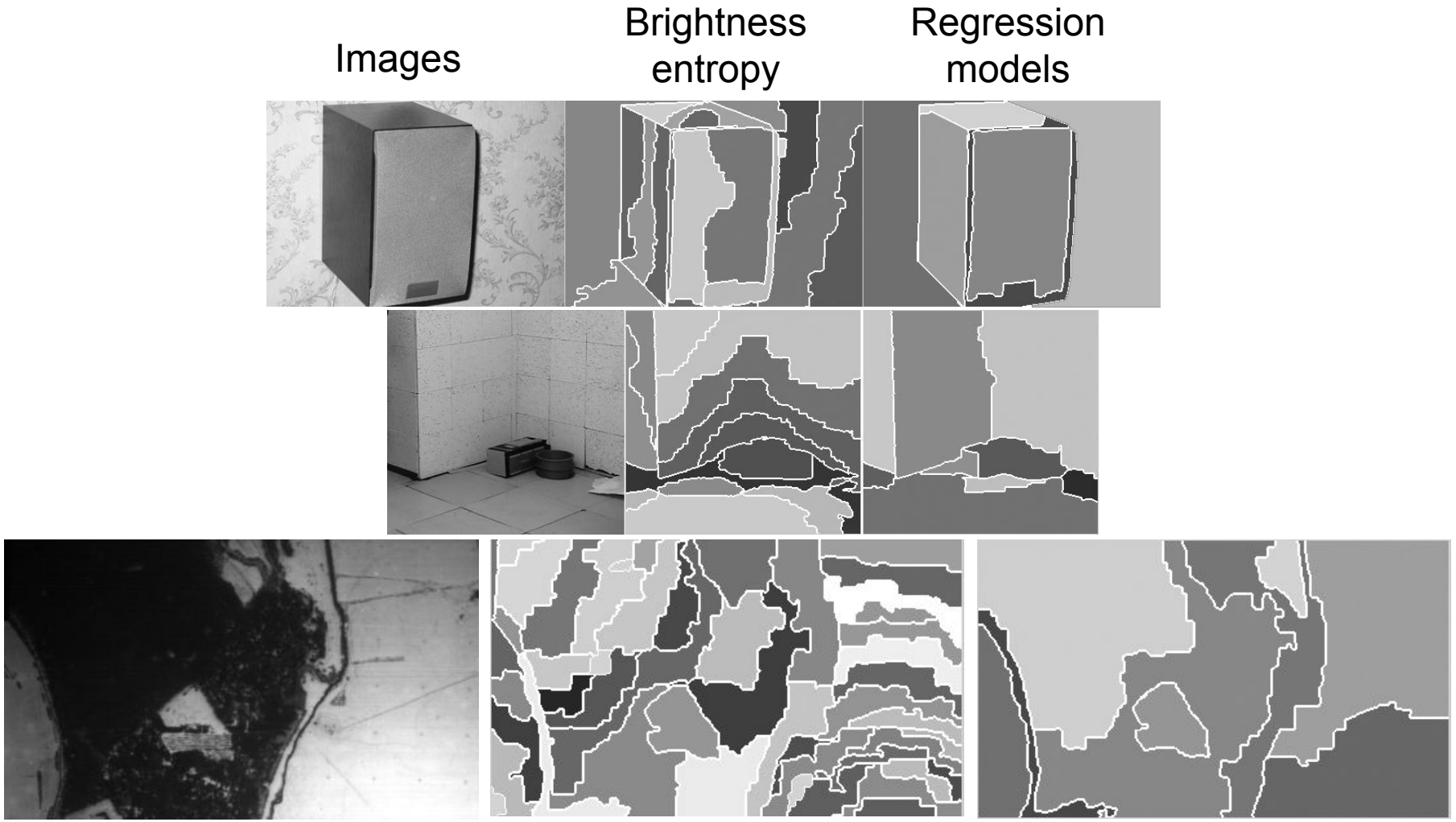
Essentially incorrect segments

More complex models…

1. Image is described as a set of independent and identically distributed samples of random variable (no segmentation).
2. Image is divided into regions; brightness values described independently within each region.
3. Second order functions are fit in each region, and brightness residuals are described as iid random variables.
4. Mixes of Gabor functions are used as regression models.

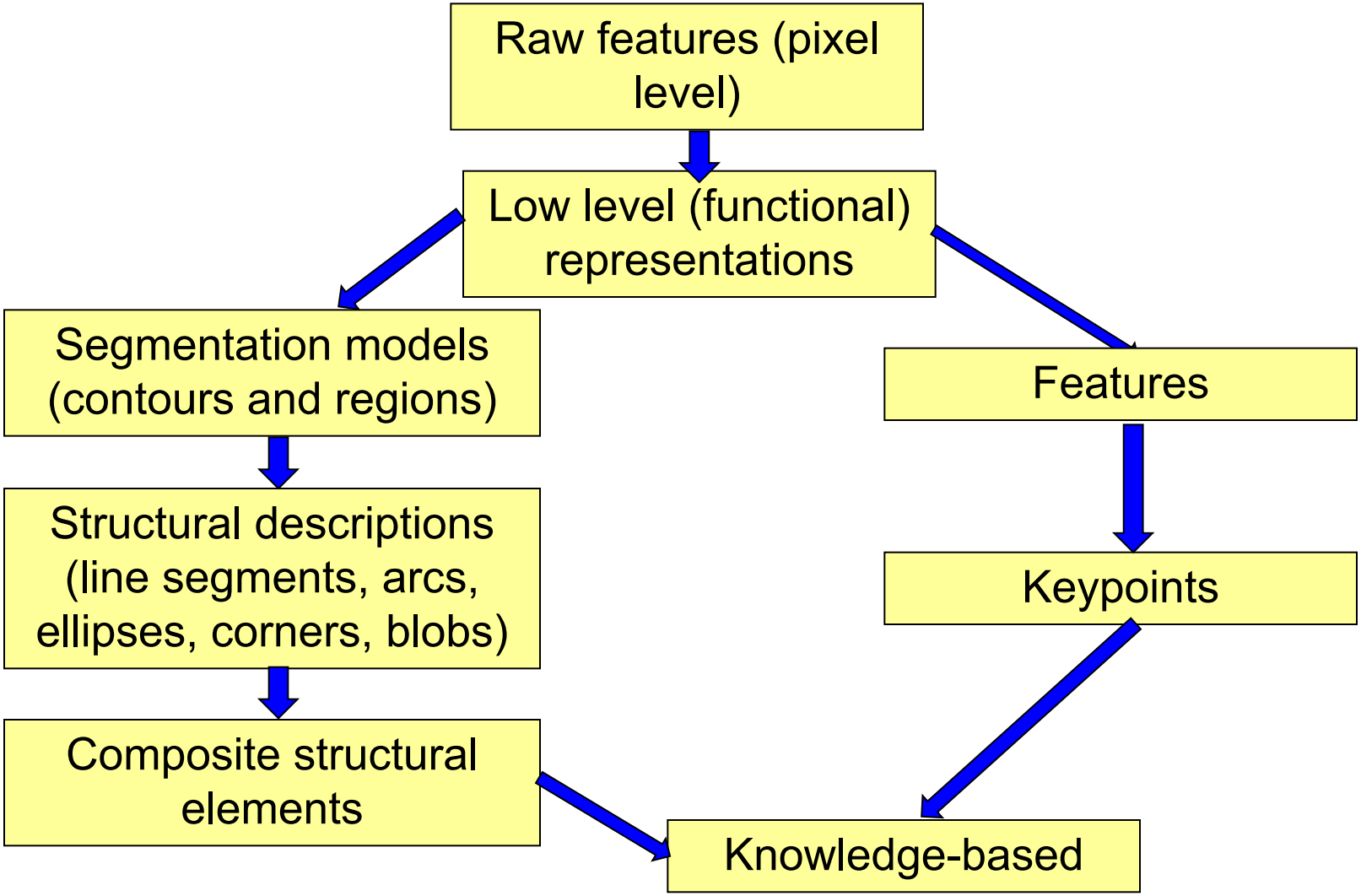


Comparison



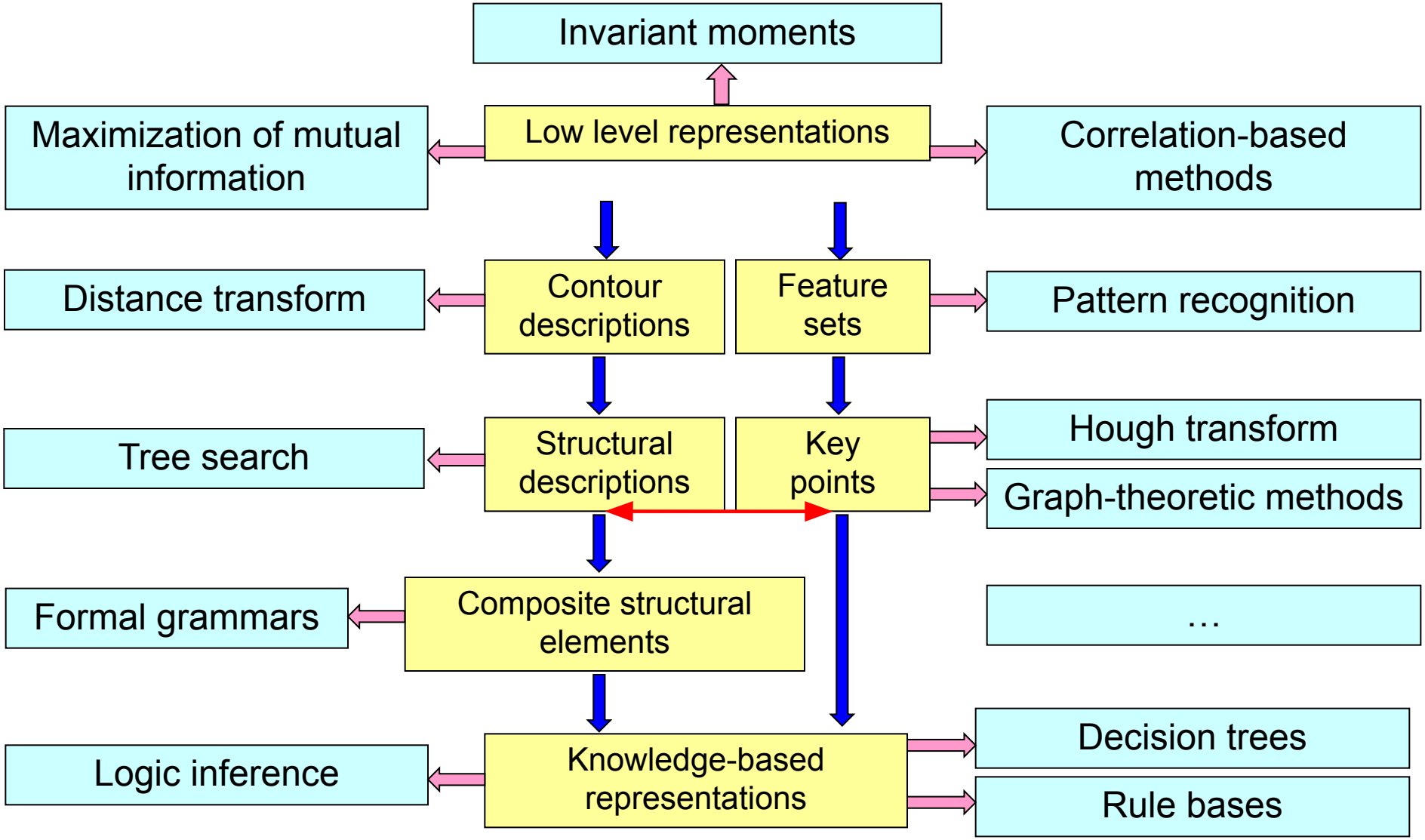
Potapov A.S., Malyshev I.A., Puysha A.E., Averkin A.N. New paradigm of learnable computer vision algorithms based on the representational MDL principle // Proc. SPIE. 2010. V. 7696. P. 769606.

Classes of image representations*



*Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. MIT Press (1982).

Example: image matching



But again… what about theory?

- MDL principle is used loosely
- Description lengths are calculated within heuristically defined coding schemes
- Success of a method is highly determined by the utilized coding scheme

- In computer vision and machine learning, some representation is used in every method

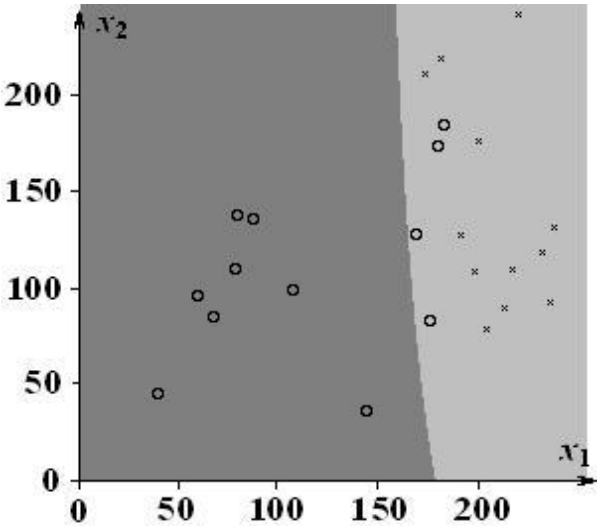
⇒ But how to construct the best representation?

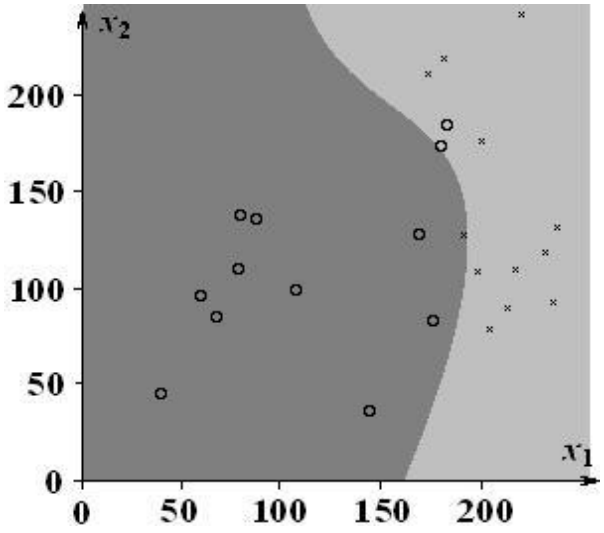
⇒ Representations correspond to ‘coding schemes’ in MDL applications. They should also be constructed on the base of strict criterion

⇒ But from what space and how?

Polynomial decision function

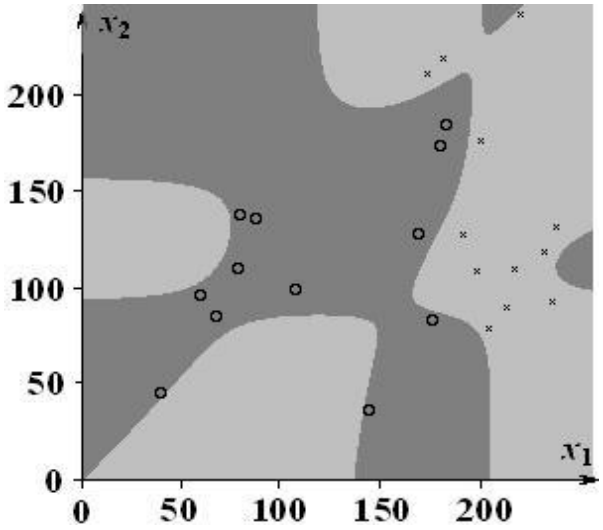
%(learn)=11.1
%(test)=5.4
L = 31.2 bit
Np=4

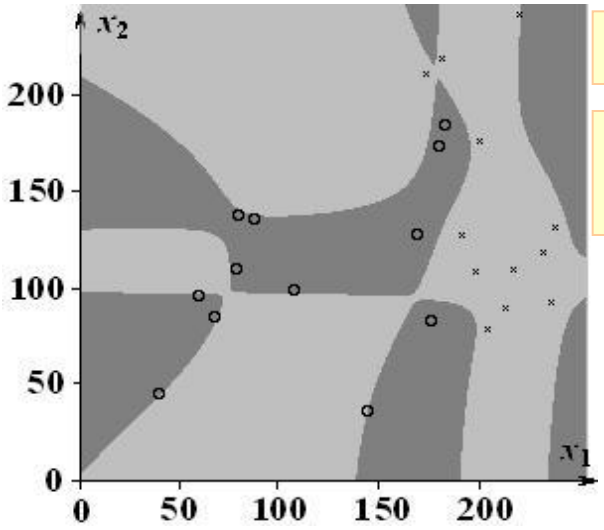




%(learn)=2.8
%(test)=3.6
L = 30.9 bit
Np=9

%(learn)=0.0
%(test)=8.6
L = 41.4 bit
Np=16



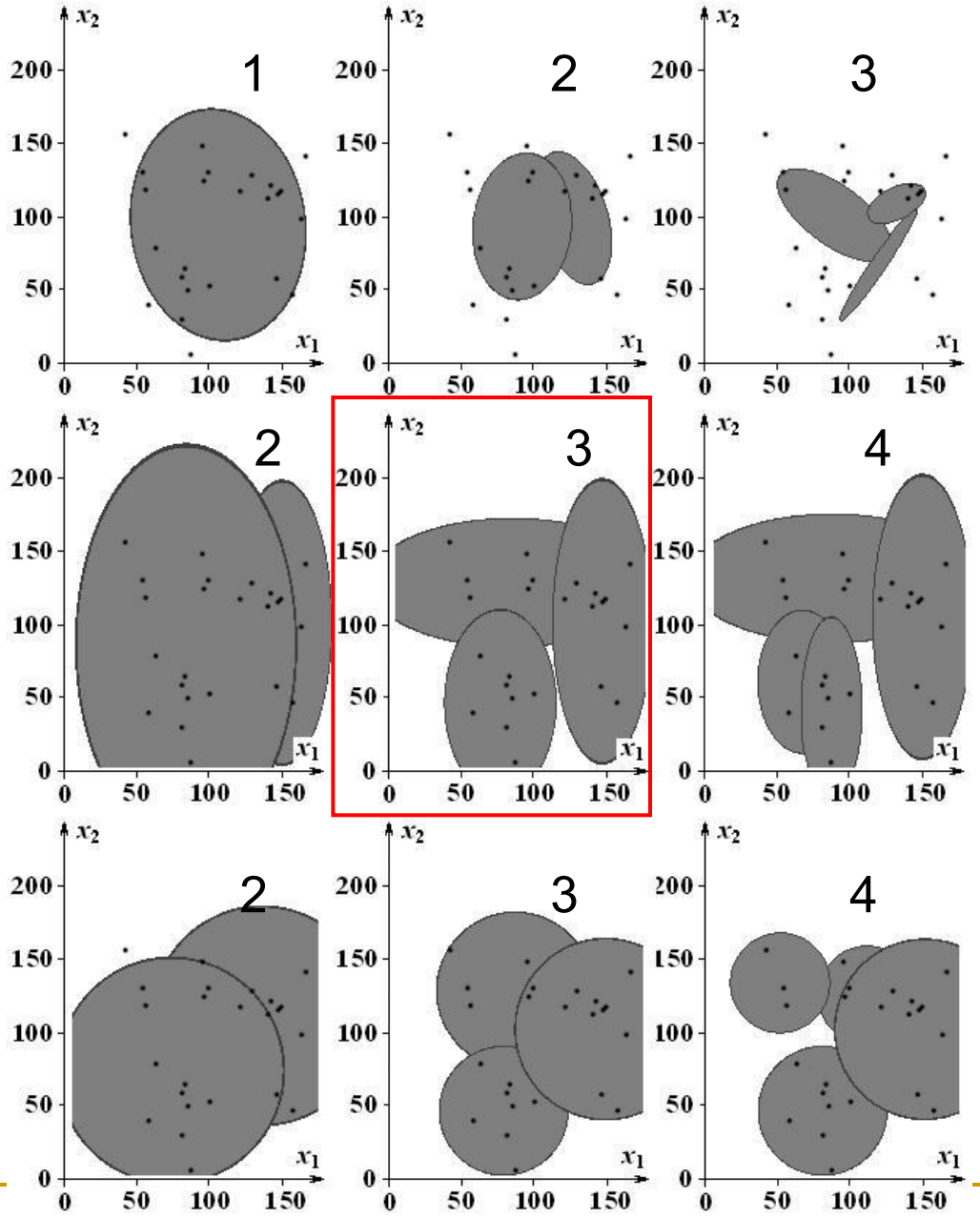


No outliers

Worst generalization!

%(learn)=0.0
%(test)=18.4
L = 62.0 bit
Np=25

Choosing between mixtures with different number of components and restrictions laid on the covariance matrix of normal distribution



L=834

L=855

L=855

L=838

L=817

L=826

L=857

L=826

L=823

Again, heuristic coding schemes

- Let’s switch back to theory

Universal Mass Induction

- Let $\{x_i\}_{i=1}^{n}$ be the set of strings
- An universal method cannot be applied to mass problems since typically

$$K_U(x_1 x_2 \ldots x_n) << \sum_{i=1}^{n} K_U(x_i)$$

where K is Kolmogorov complexity on universal machine U

- However, $K_U(x_1 x_2 \ldots x_n) \approx \min_S \left(l(S) + \sum_{i=1}^{n} K_U(x_i \mid S) \right)$ can hold
- One can search for models $y_i^* = \underset{y:S(y)=x_i}{\operatorname{argmin}} l(y)$ for each x_i independently within some best representation $S^* = \underset{S}{\operatorname{argmin}} \left(l(S) + \sum_{i=1}^{n} l(y_i^*) \right)$

Potapov, A., Rodionov, S.: Extending Universal Intelligence Models with Formal Notion of Representation. In: J. Bach, B. Goertzel, M. Iklé (Eds.): AGI'12, LNAI 7716, pp. 242–251 (2012).

Representational MDL principle

For example, image analysis tasks are mass problems: the same algorithm is applied to different images (or patterns) independently.

- Definition

Let *representation* for the set of data entities be such the program *S* for UTM *U* that for any data entity *D* the description *H* exists that *U*(*SH*)=*D*.

- Representational MDL principle
  - The best image description has minimum length within given representation
  - The best image representation minimizes summed description length of images from the given training set (and the length of representation itself).

Main advantage: applicable to any type of representation; representation is included into general criterion as a parameter.

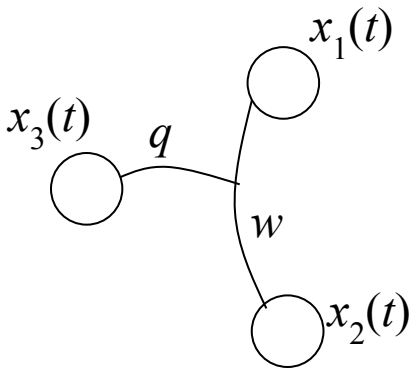
Possible usage of RMDL

- Synthetic pattern recognition methods*:
  - Automatic selection among different pattern recognition methods
- Selecting a representation that better fits the training sample from a specific domain either from a family of representations or from a fixed set of hand-crafted representations
- Improve data analysis methods for specific representations

* Potapov A.S. Synthetic pattern recognition methods based on the representational minimum description length principle // Proc. OSAV'2008. 2008. P. 354–362.

RMDL for optimizing ANN formalisms

- Considered extension of ANN representation

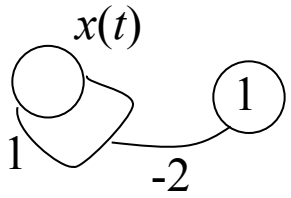


$$x_2'(t) = wx_1^{qx_3(t)+1}(t)$$

$$q = 0 \vee x_3 = 0 \Rightarrow x_2'(t) = wx_1(t)$$

$$qx_3 = 1 \Rightarrow x_2'(t) = wx_1^2(t)$$

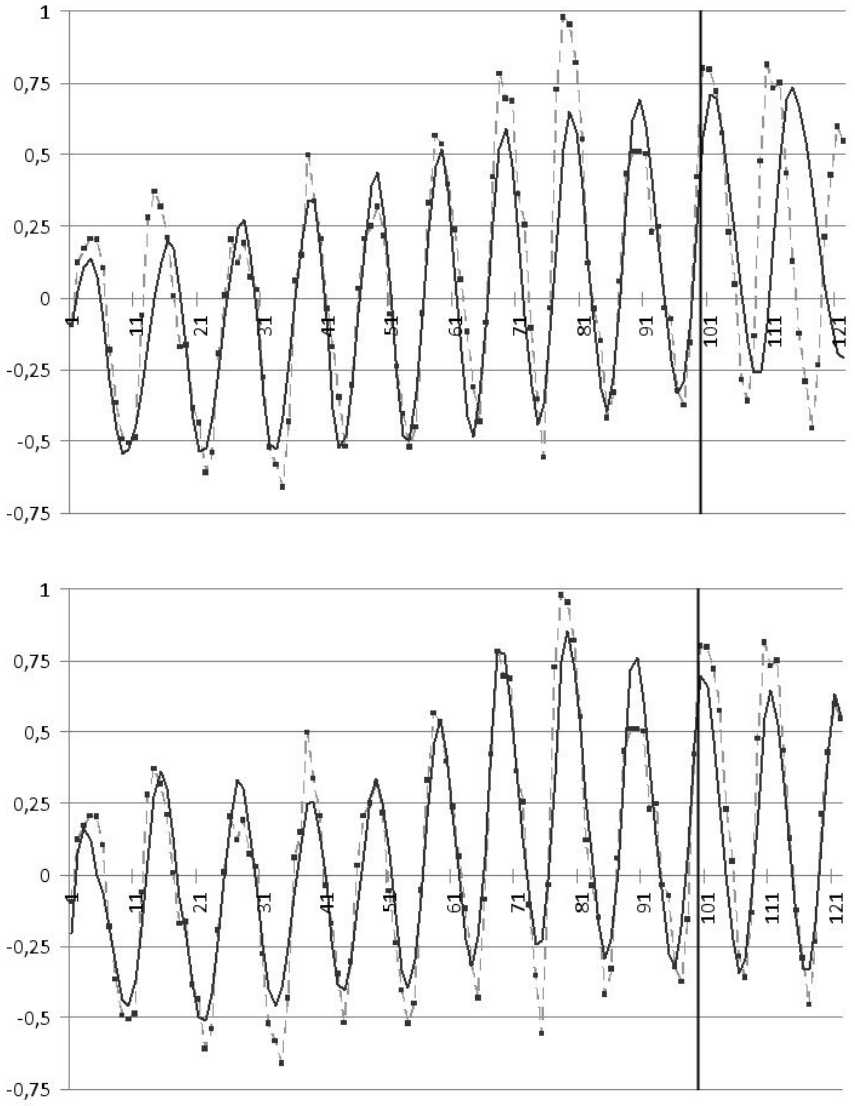
$$qx_3 = -2 \Rightarrow x_2'(t) = wx_1^{-1}(t)$$



$x'(t)=1/x(t)$
$x(t)=\ln(t)$

RMDL for optimizing ANN formalisms

- Experiments: Wolf annual sunspot time series
- Precision of forecasting depends on type of nonlinearity
- ANN with 4 neurons, 11 connections, and 2 second-order connections: MSE=220 (typical MSE: 214–625*)

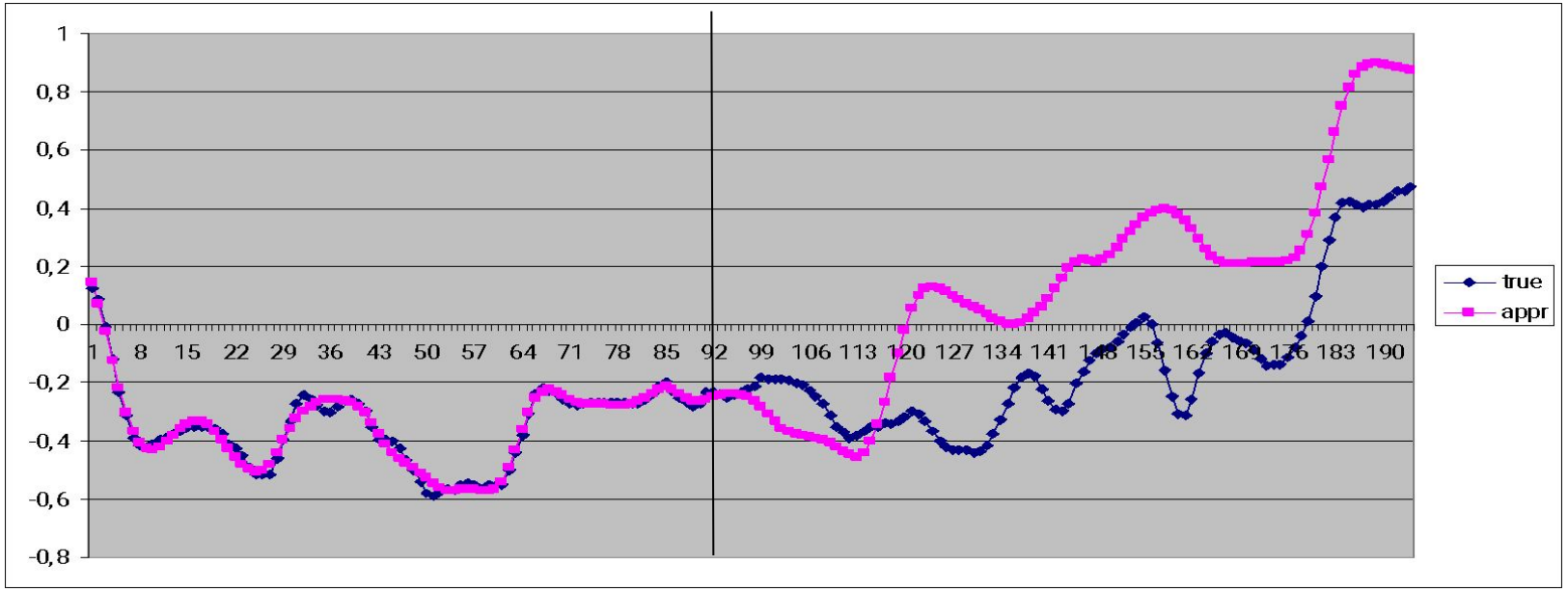


RMDL for optimizing ANN formalisms

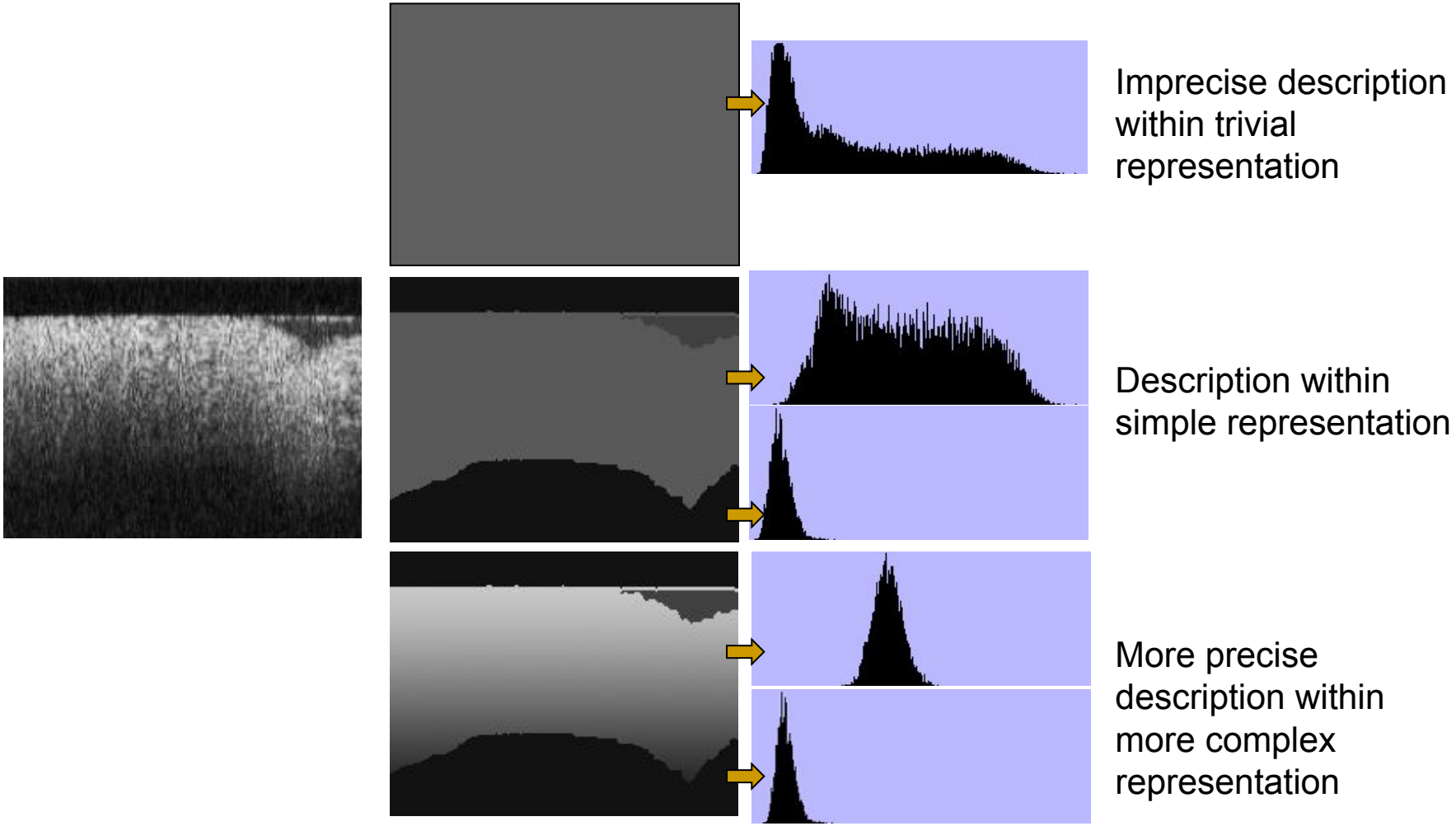
ANN type	RMDL, bits	error, %
Linear	651	15,8
Activation function	617	10,1
2nd–order connections	608	9,9

Test: Financial time series

Although we obtained an agreement between the short-term prediction precision and the RMDL criterion in average, one can agree with the statement: “MSE and NMSE are not very good measures of how well the model captures the dynamics”

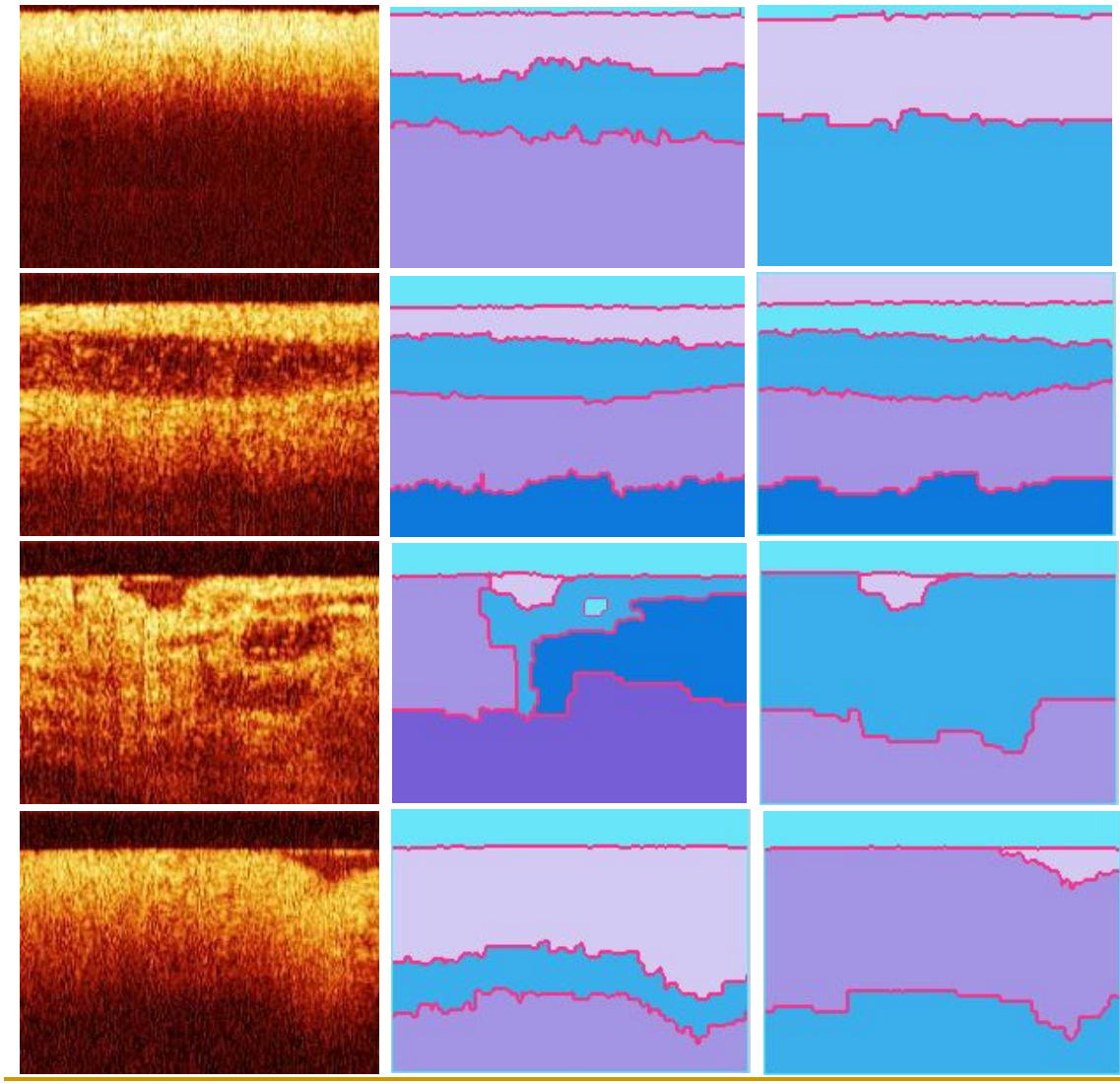


OCT image segmentation



Gurov I., Potapov A. Investigation of OCT Images Descriptions on the Base of Representational MDL Principle // Proc. MVA2009 IAPR Conference on Machine Vision Applications. P. 320–323.

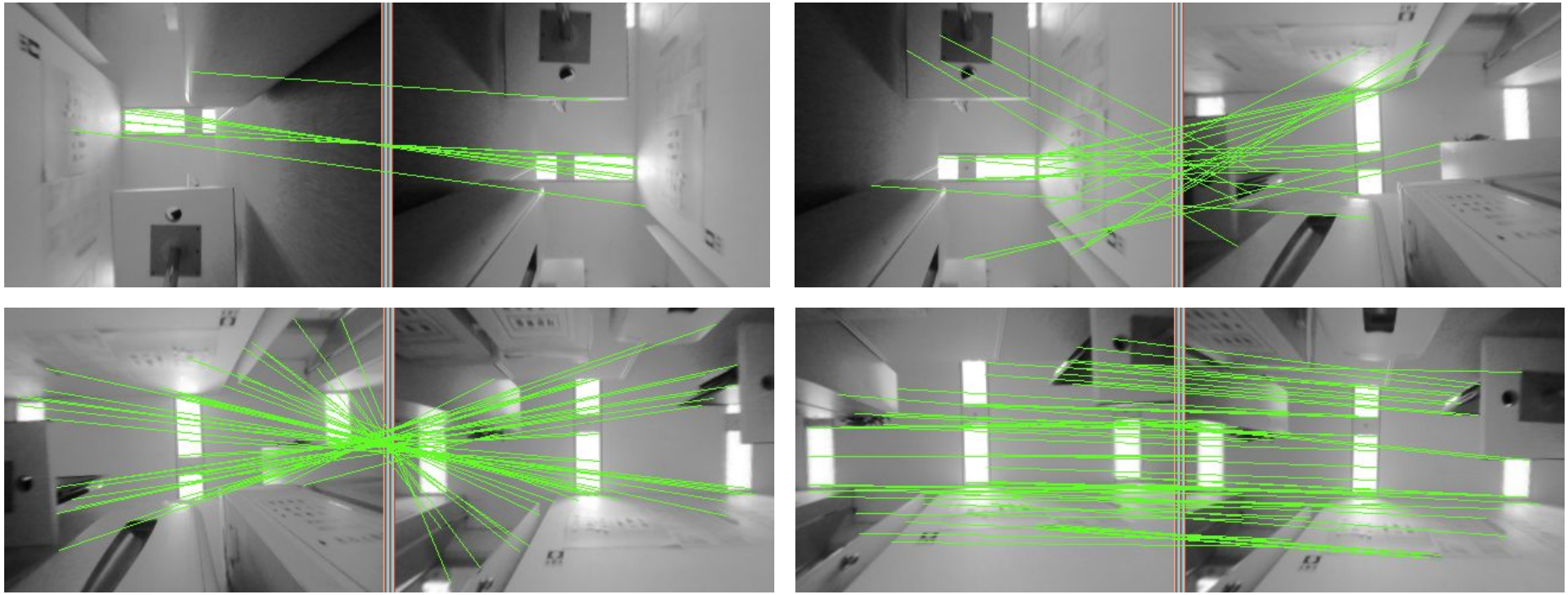
Segmentation results



Description length, bits	
S-0: 212204	
S-1: 184672	S-1: oversegmentation
S-2: 175096	S-2: correct detection of layers
S-0: 231201	
S-1: 212268	S-1 and S-2 are almost the same (and plausible) detection of thin layers
S-2: 207864	
S-0: 235566	
S-1: 219641	Differing segmentation results for a single thick layer (light absorption with depth causes regular reduction of brightness). Some inclusions are not detected.
S-2: 215066	
S-0: 236421	
S-1: 213015	S-1: odd layer is detected and inclusion is missed
S-2: 206204	S-2: plausible results of segmentation

Application to image feature learning

Training set with preliminarily matched key points using predefined hand-crafted feature transform

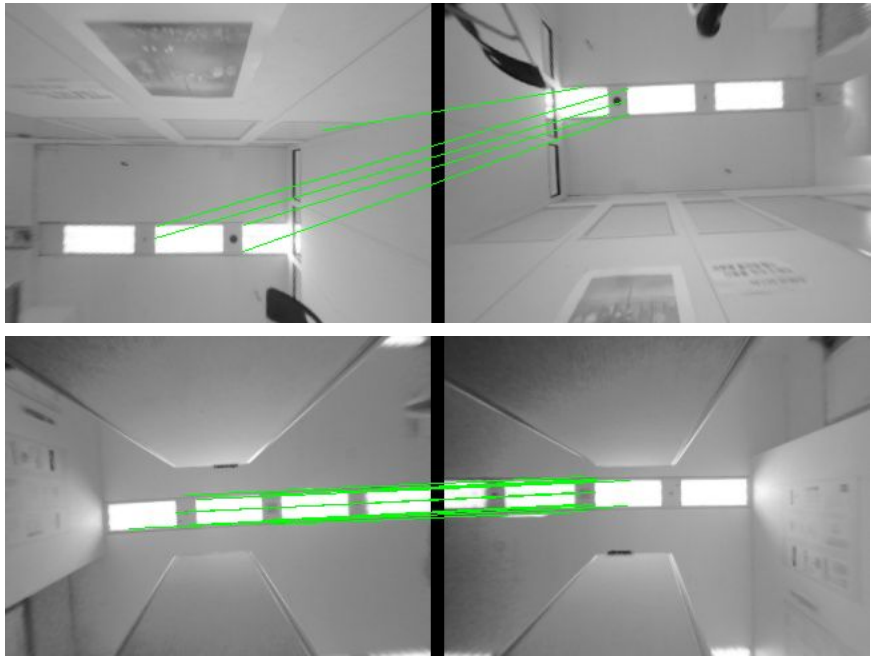


Example of some found linear feature transforms

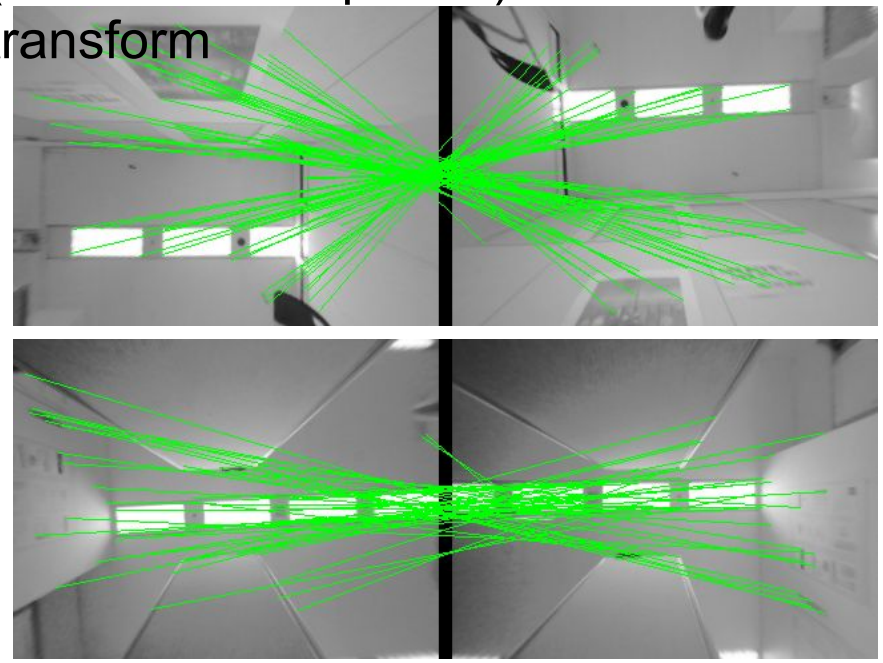
Example of some feature transforms for another environment

Results

Matching with predefined hand-crafted feature transform

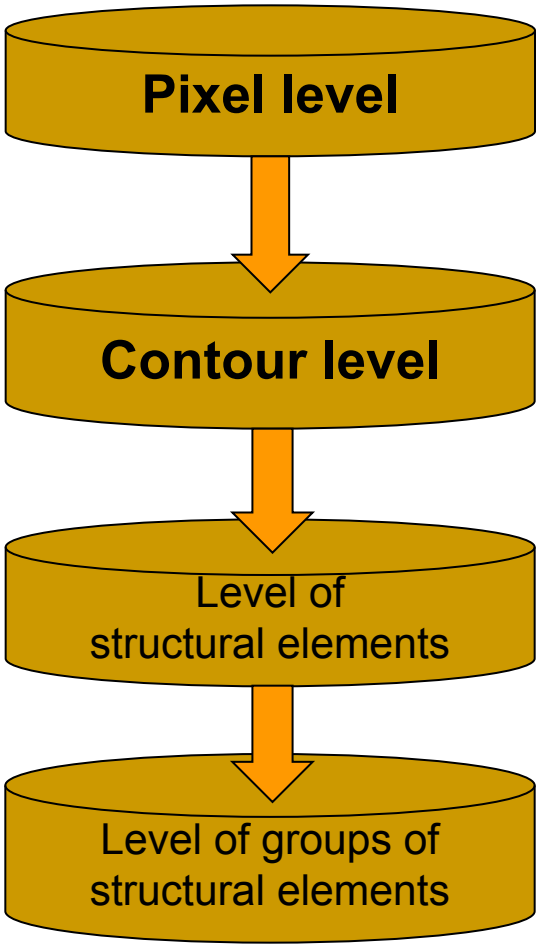


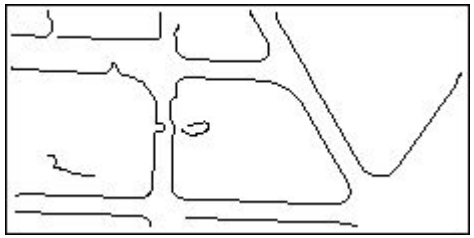
Matching with learned (environment-specific) feature transform

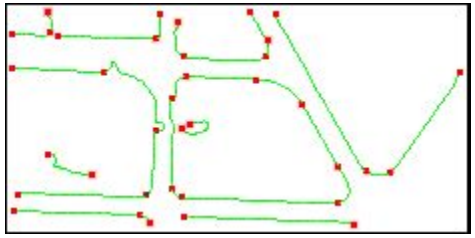


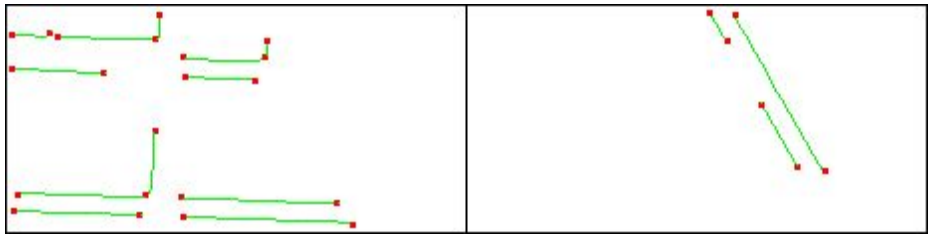
~50% of failures with predefined features were matched successfully with learned features (new images of the same environment were used)

Analysis of hierarchical representations

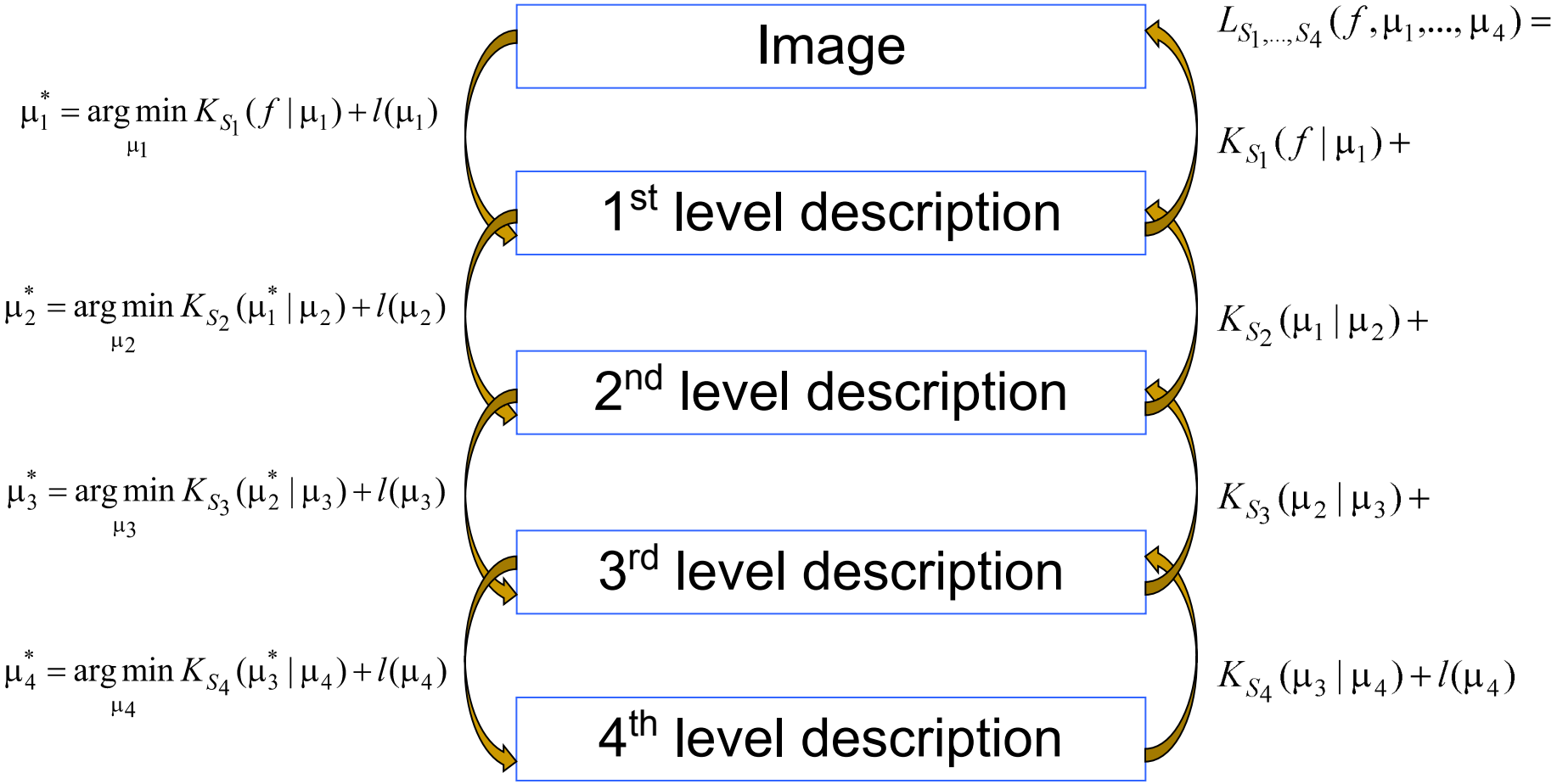




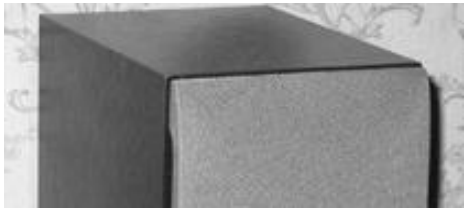


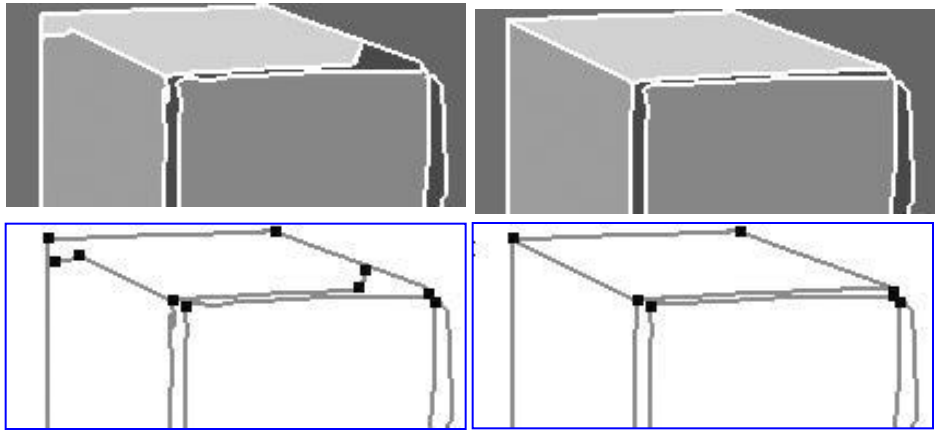


Adaptive resonance



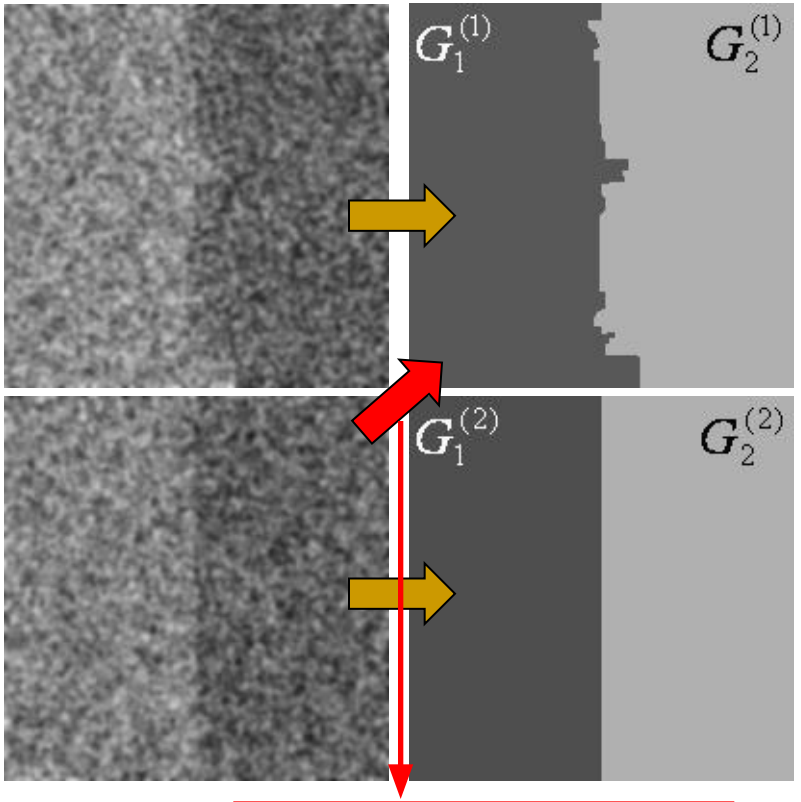
Implications





Independent optimi-zation of descriptions

Usage of integral description length



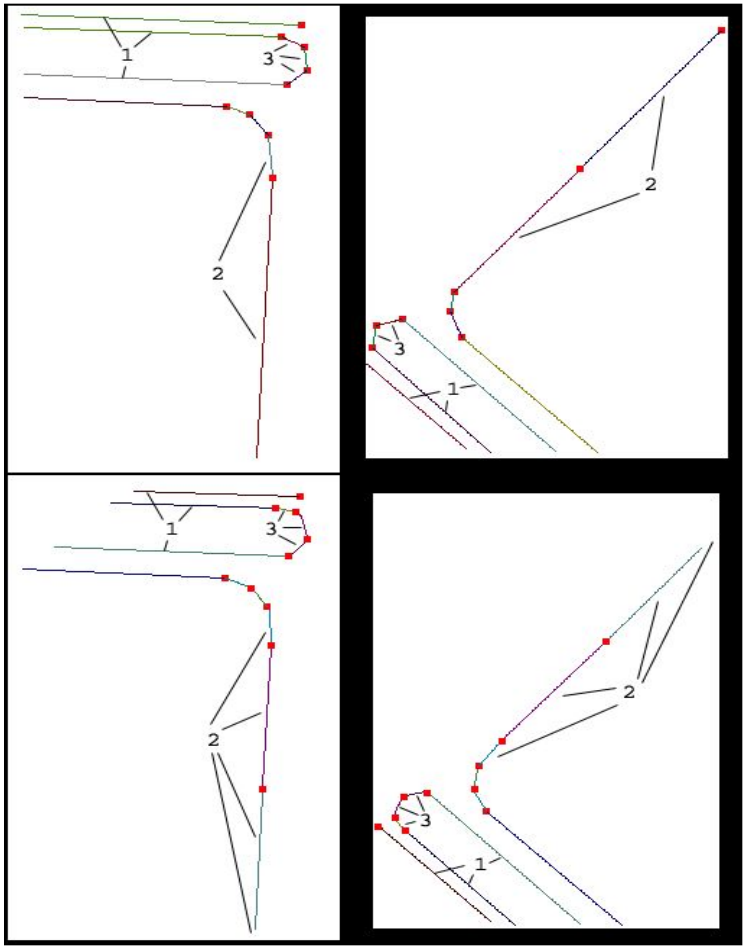
Without resonance

$$L_{S_1^{(H)}}(f,\mu_1,\mu_2,\mu_3,\mu_4) = K_{S_1^{(1)}}(f \mid \mu_1) + K_{S_1^{(1)}}(\mu_1 \mid \mu_2) +$$

$$+ K_{S_2^{(2)}}(\mu_2 \mid \mu_3) + K_{S_1^{(3)}}(\mu_3 \mid \mu_4) + K_{S_0^{(4)}}(\mu_4).$$

Adaptive resonance: matching as construction of common description

Initial structural elements of the first image



Initial structural elements of the second image

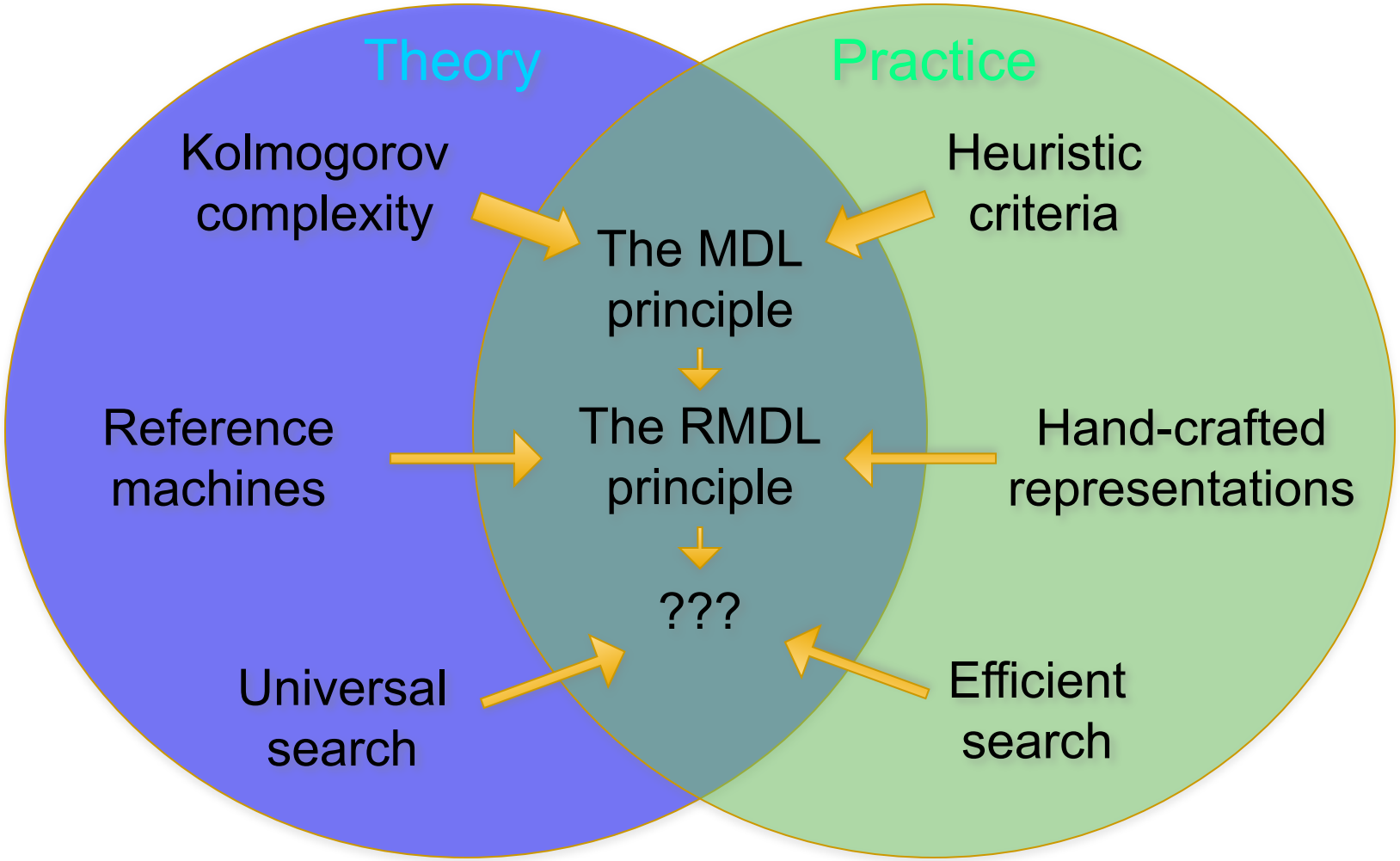
Fixed structural descriptions: same for both images

These descriptions slightly less precise, but w.r.t. images, but only one of them can be used instead of two

Learning representations

- Very difficult problem in Turing-complete settings
- Successful methods use efficient search and restricted families of representations
- Deep learning
  - Not universal
  - Compact (one-level ANNs should be exponentially larger than multi-level ANNs to represent some concepts => particular case of RMDL)
  - Higher expressive power or more efficient search than those of former methods

What is still missing?



Key Idea

- Humans create narrow methods, which efficiently solve arbitrary recurring problems
- Generality should be achieved not by a single uniform method solving any problem in the same fashion, but by automatic construction of (non-universal) efficient methods
- Program specialization is the appropriate concept*, which relates general and narrow intelligence methods
- However, no analysis of possible specialization of concrete models of universal intelligence has been given yet.

* Khudobakhshov, V.: Metacomputations and Program-based Knowledge Representation. In:K.-U. Kühnberger, S. Rudolph, P. Wang (Eds.): AGI'13, LNAI 7999, pp. 70–77 (2013).

Program Specialization

- Let $p_L(x,y)$ be some program (in some language L) with two arguments
- Specializer $spec_R$ is such program (in some language R) accepting p_L and x_0 that

$$(\forall y) spec_R(p_L, x_0)(y) = p_L(x_0, y)$$

  - $spec_R(p_L, x_0)$ is the result of deep transformation of p_L that can be much more efficient than $p(x_0, .)$

Futamura-Turchin projections

$$(\forall x) spec_R(intL, p_L)(x) = intL(p_L, x)$$

$$(\forall p_L, x) spec_R(spec_R, intL)(p_L)(x) = intL(p_L, x)$$

$$(\forall intL) spec_R(spec_R, spec_R)(intL) = comp_{L \to R}$$

Specialization of Universal Induction

- Universal mass induction consists of two procedures
  - Search for models

$$MSearch(S, x_i) \to y_i^* = \underset{y:S(y)=x_i}{\operatorname{argmin}} l(y)$$

  - Search for representations

$$RSearch(x_1, \ldots x_n) \to S^* = \underset{S}{\operatorname{argmin}} \left(l(S) + \sum_{i=1}^{n} l(y_i^*) \right)$$

- $MSearch(S, x)$ is executed for different x with same S
- This search cannot be non-exhaustive for any S, but it can be efficient for some of them
- One can consider computationally efficient projection $spec(MSearch, S)$: $(\forall x) spec(MSearch, S)(x) = MSearch(S, x)$

Approach to Specialization

- Direct specialization of *MSearch*(*S*, *x*) w.r.t. some given S^*
  - No general techniques for exponential speedup exists
  - And how to get *S*? *RSearch* is still needed
- Find $S'=spec(MSearch(S, x), S^*)$ simultaneously with S^*

  Main properties of *S*, *S*′: $(\forall x) S(S'(x)) = x$

  $$l(S) + \sum_i l(S'(x_i)) \to \min$$

- *S* is a generative representation (decoding)
- *S*′ is a descriptive representation (encoding)
  - *S*′ is also the result of specialization of the search for generative models, so in general it can include some sort of optimized search
- Simultaneous search for *S* and *S*′ will be referred to as *SS*′*-search*

Conclusion

- Attempts to build more powerful practical methods leaded us to utilization of the MDL principle that was heuristically applied for solving many tasks
- The MDL principle is very useful tool for introducing model selection criteria free from overfitting in the tasks of image analysis and pattern recognition
- We introduced the representational MDL principle to bridge the gap between universal induction and practical methods and used it to extend practical methods
- The remaining difference between universal and practical methods is in search algorithms. Specialization of universal search is necessary to automatically produce efficient methods

Thank you for attention!

Contact: potapov@aideus.com