Computational Learning

Design of image operators

Junior Barrera

Nina S. T. Hirata

Marcel Brun

Instituto de Matemática e Estatística

University of São Paulo - Brazil

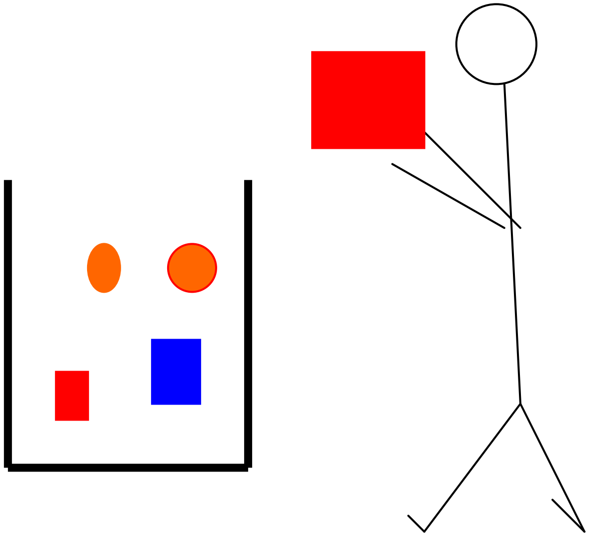
July/1999

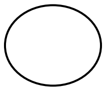
Outline

- Computational learning
- Design of W-operators
- Amount of data available
- Distribution of the domain
- Size of the window
- Constraints
- Gray scale and motion applications
- Conclusion

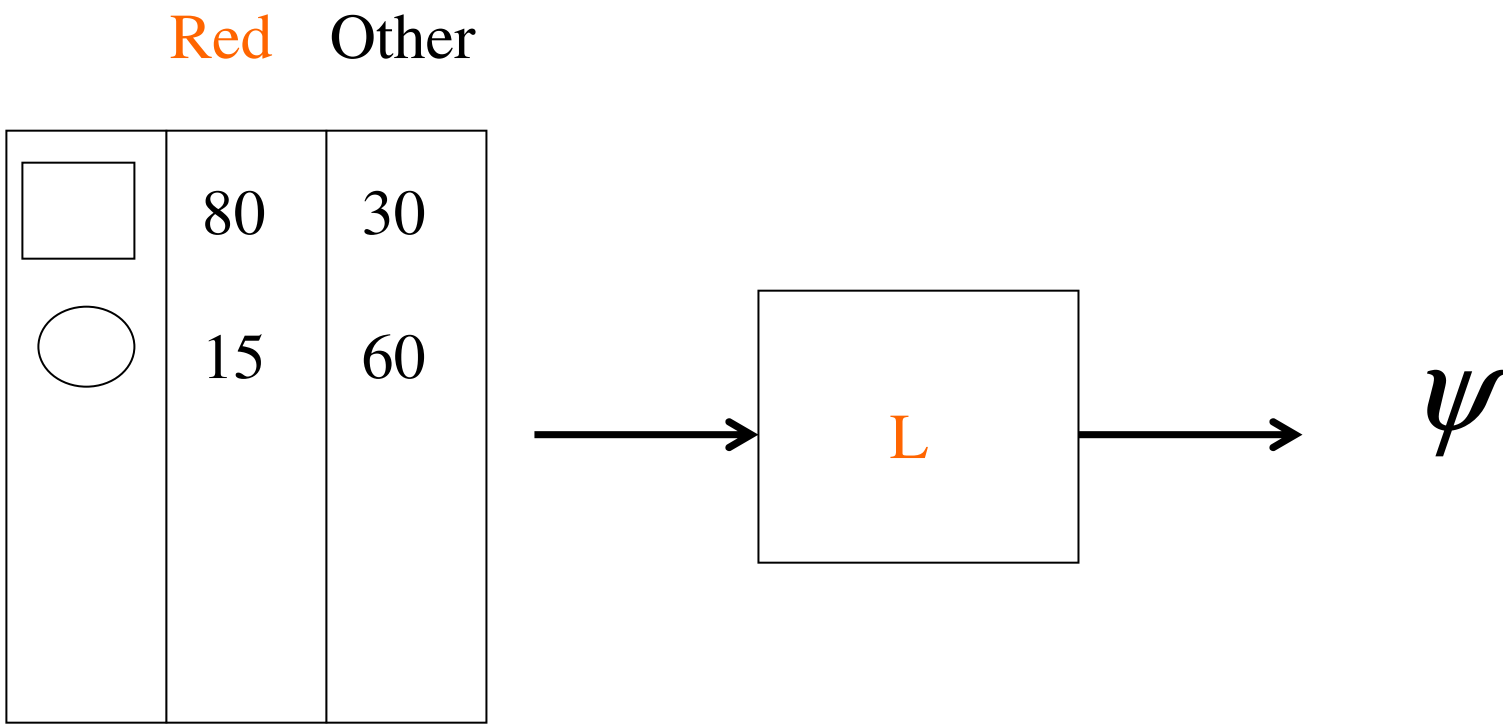
Learning a concept

- Domain: planar shapes
- concept: color RED



	Red	Other
	80	30
	15	60

Learning a concept



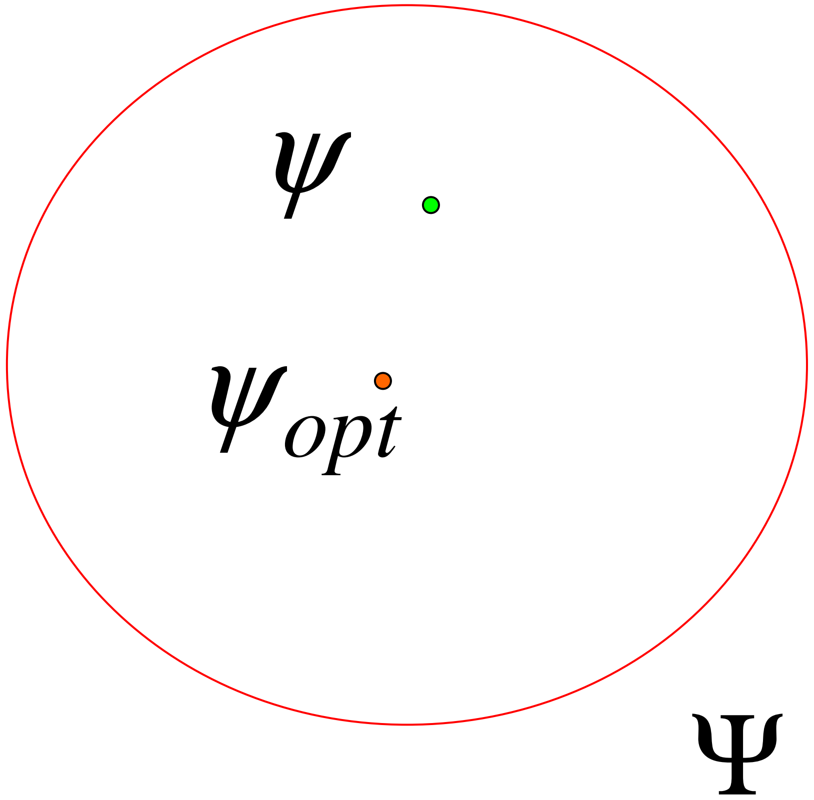
Terminology

- domain: objects with a random distribution
- concept: a set of objects of a given domain (or a binary function)
- teacher: he says if a generic object satisfies the concept, but he may make mistakes

Terminology

- example: an object classified by the teacher
- learning algorithm: gives an hypothesis for the concept from a collection of examples
- training data: examples used in the learning algorithm

The optimization problem



The optimization problem

ψ is a classifier on a random domain

$Er[\psi]$ is the error of ψ

ψ_{opt} is a classifier of minimum error

$$Er[\psi_{opt}] \leq Er[\psi], \forall \psi \in \Psi$$

Error measure

- **Design goal is to find a function with minimum risk.**
- **Risk (expected loss) of a function :**

$$R(\psi) = E[l(\psi(X), Y)]$$

X is a random set
Y is a binary random variable

- **Loss function**

$$l : \{0,1\} \times \{0,1\} \rightarrow R^+$$

Join probability

$$R(\psi) = \sum_{X,Y} l(\psi(X),Y)\, p(X,Y)$$

$p(X,Y)$ needs to be estimated from training data

MAE example

- ***Example*** **: MAE loss function**

$$l_{MAE}(a,b) = |a-b| \qquad a,b \in \{0,1\}$$

$$MAE\langle\Psi\rangle = E[|\psi(X) - Y|]$$

- **Optimal MAE function**

$$\psi(X) = \begin{cases} 1 & p(1,X) > p(0,X) \\ 0 & p(1,X) \le p(0,X) \end{cases}$$

Generalization

PAC learning

L is Probably Approximately Correct (PAC)

For $m > m(\varepsilon, \delta)$ examples

$$\Pr(|R(\psi) - R(\psi_{opt})| < \varepsilon) > 1 - \delta$$

$$\varepsilon, \delta \in (0, 1)$$

Efficiency

L should be computed in polynomial time

ψ should be represented in polynomial storage space

ψ should be executed in polynomial time

PAC learning procedure

1. Estimate $P(X,Y)$ ***from training images***

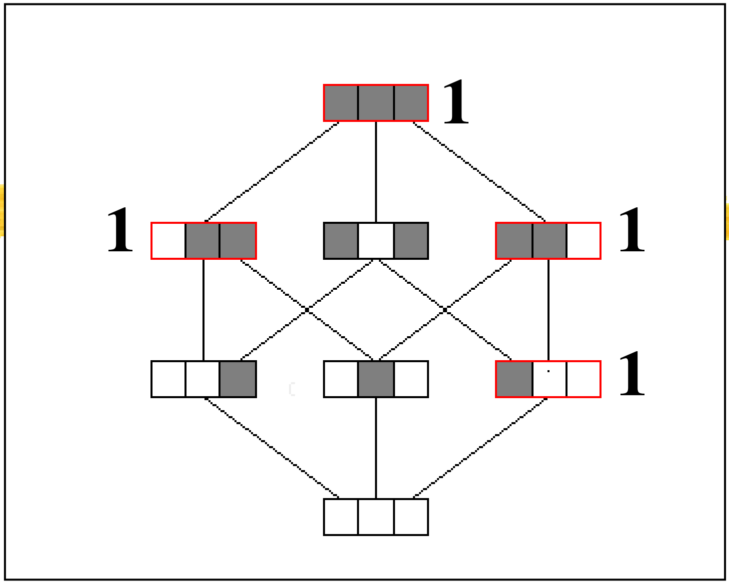
2. Attribute the binary label that minimizes the loss, for each observed shape

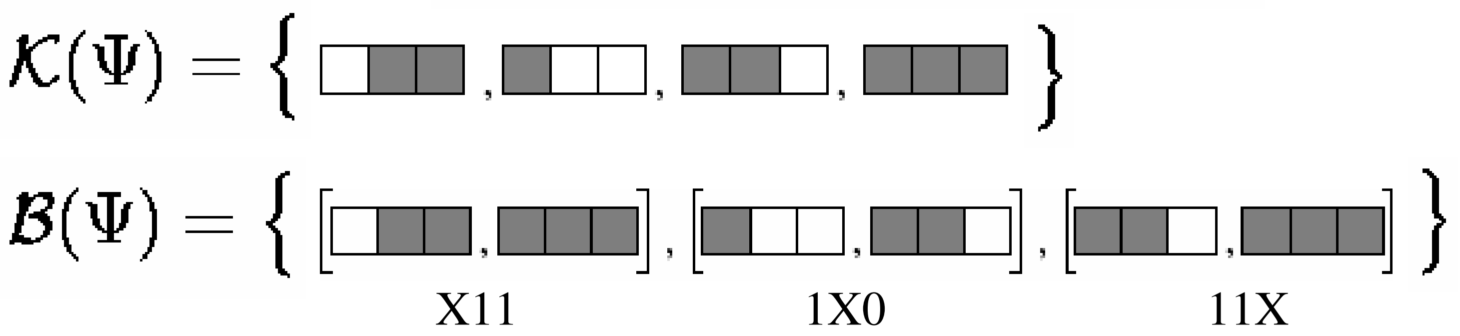
3. Make representation simplification (e.g., compute base), and attribute labels to non-observed shapes (generalization or prediction).

Example : Boolean function minimization, where non-observed shapes may be regarded as don't cares for minimization purpose.

Representation

Window $W = 1 \times 3$

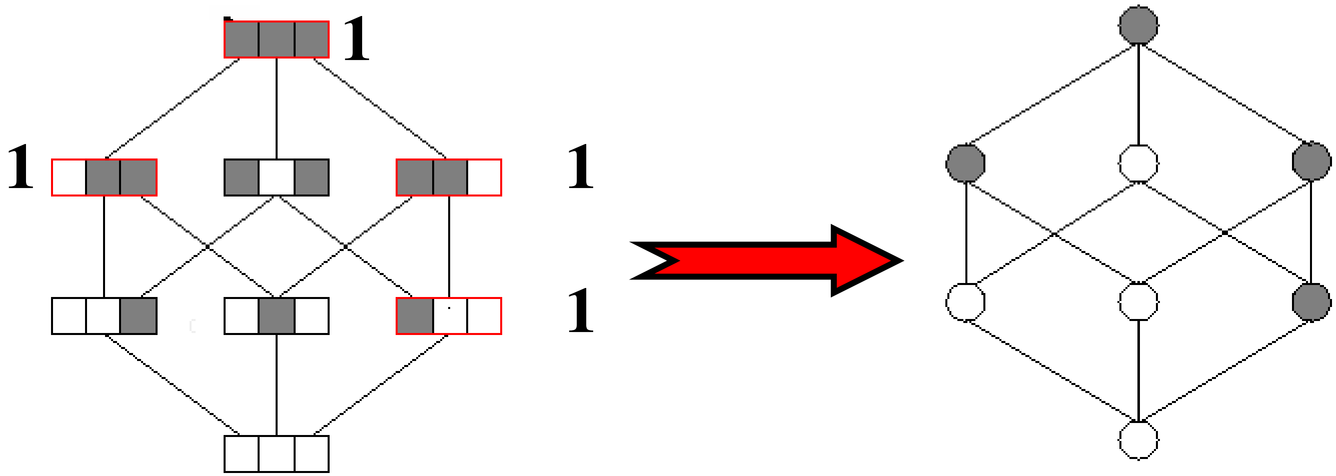




$$\psi = \lambda_{X11} \cup \lambda_{1X0} \cup \lambda_{11X}$$

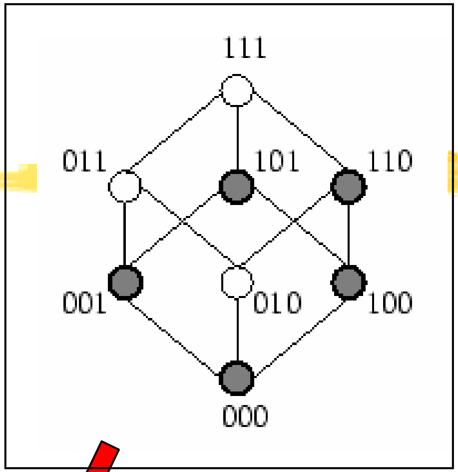
Pictorial representation

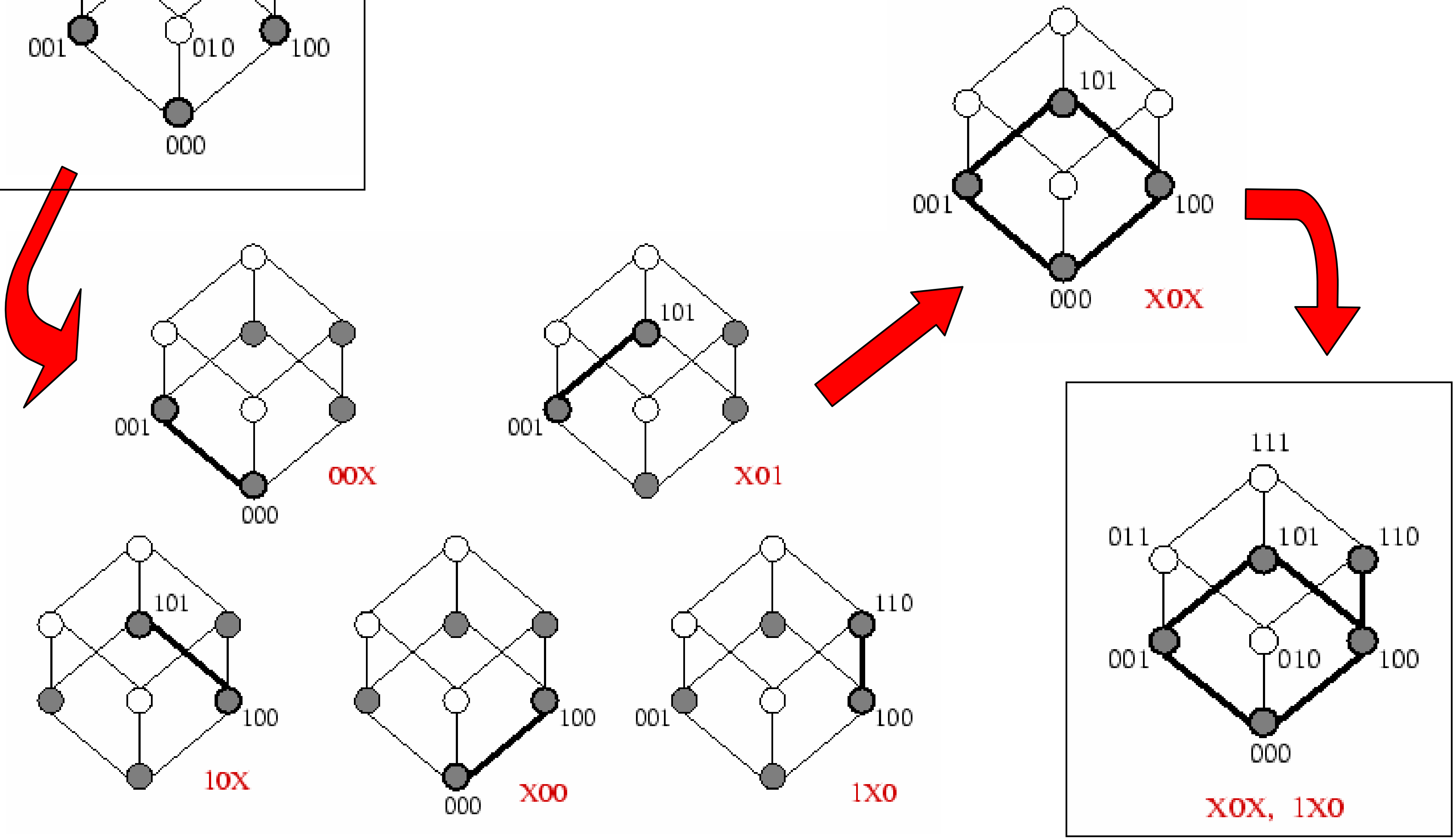
Simplified representation



dark = 1
white = 0

Minimization by Quine-McCluskey

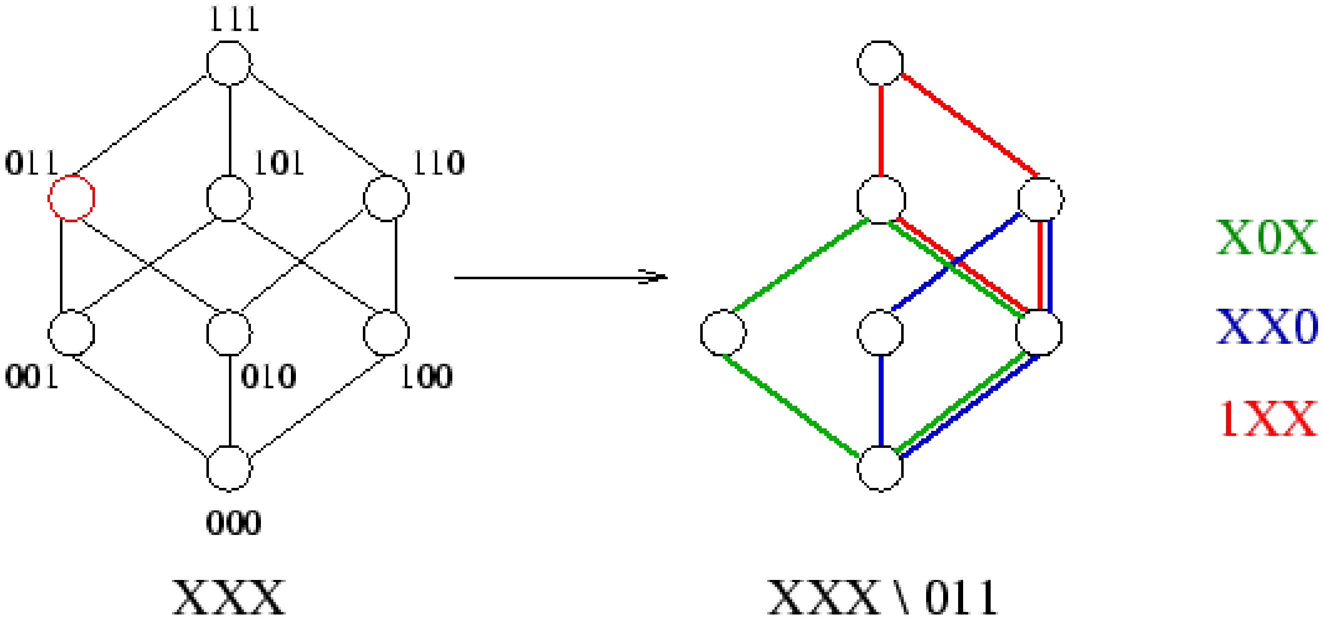




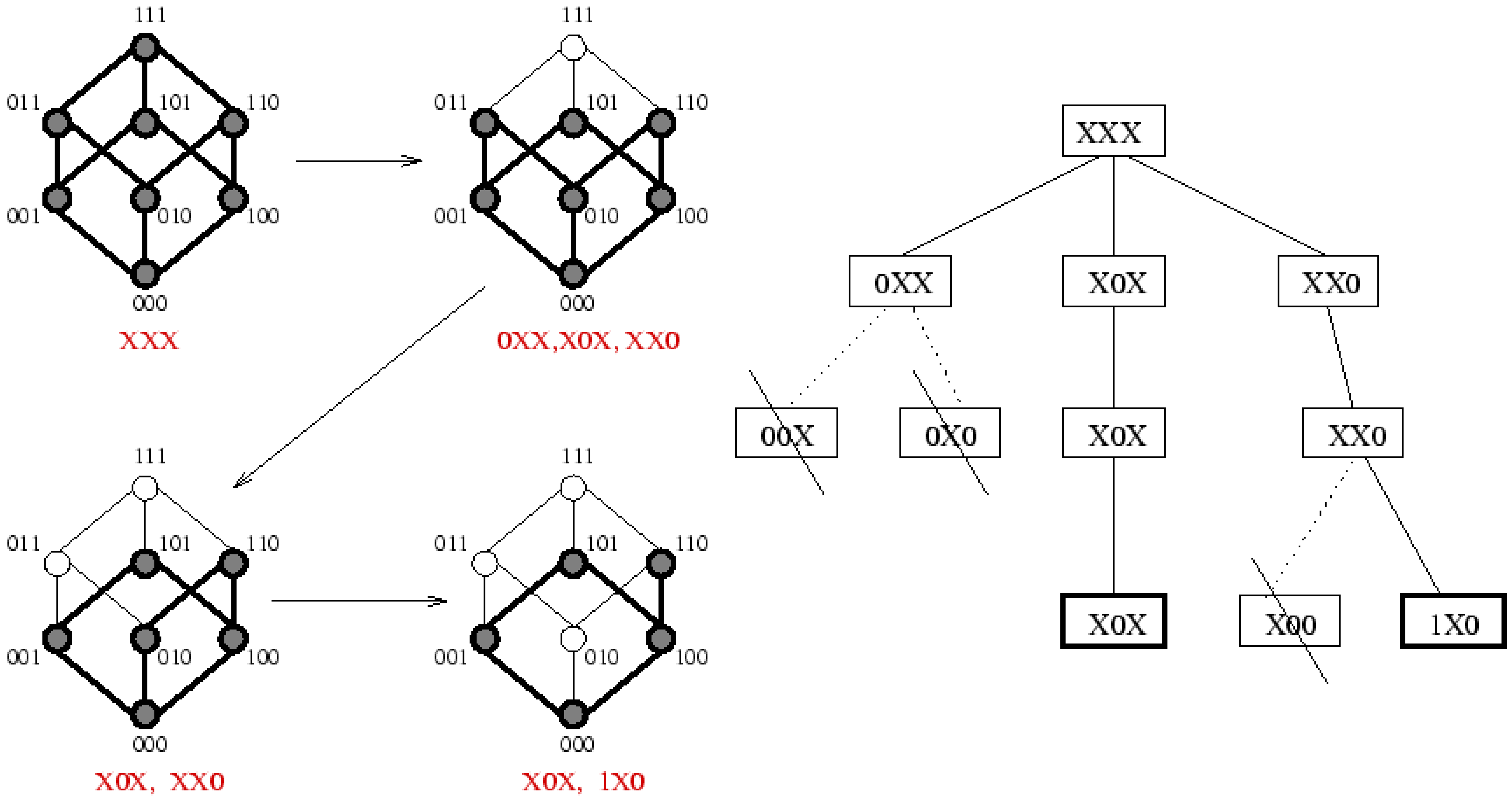
ISI (incremental splitting of intervals)

- Interval : $[A, B] \subseteq \mathcal{P}(W)$
- Splitting of *[A,B]* by **X**, $X \in [A, B]$

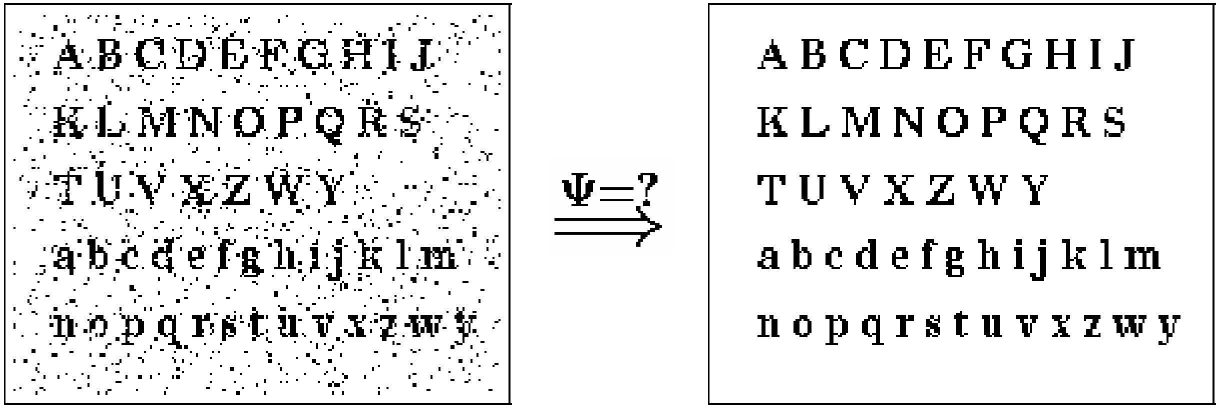
$$[A, B] \setminus X = \{[A, B \cap \{a\}^c] : a \in P \cap A^c\} \cup \{[A \cup \{b\}, B] : b \in P^c \cap B\}$$



ISI algorithm



The problem



observed ideal

Find an image operator that transforms the observed image to the respective ideal (or "close to the ideal") image.

Binary image operators

- **Binary image :** $f : E \rightarrow \{0,1\}$
- **Binary images can be understood as sets :**

$$f \longleftrightarrow S$$

$$x \in S \Leftrightarrow f(x) = 1 \quad \forall x \in E$$

$(\mathcal{P}(E), \subseteq)$ **is a complete Boolean lattice**

- **Binary image operators = set operators :**

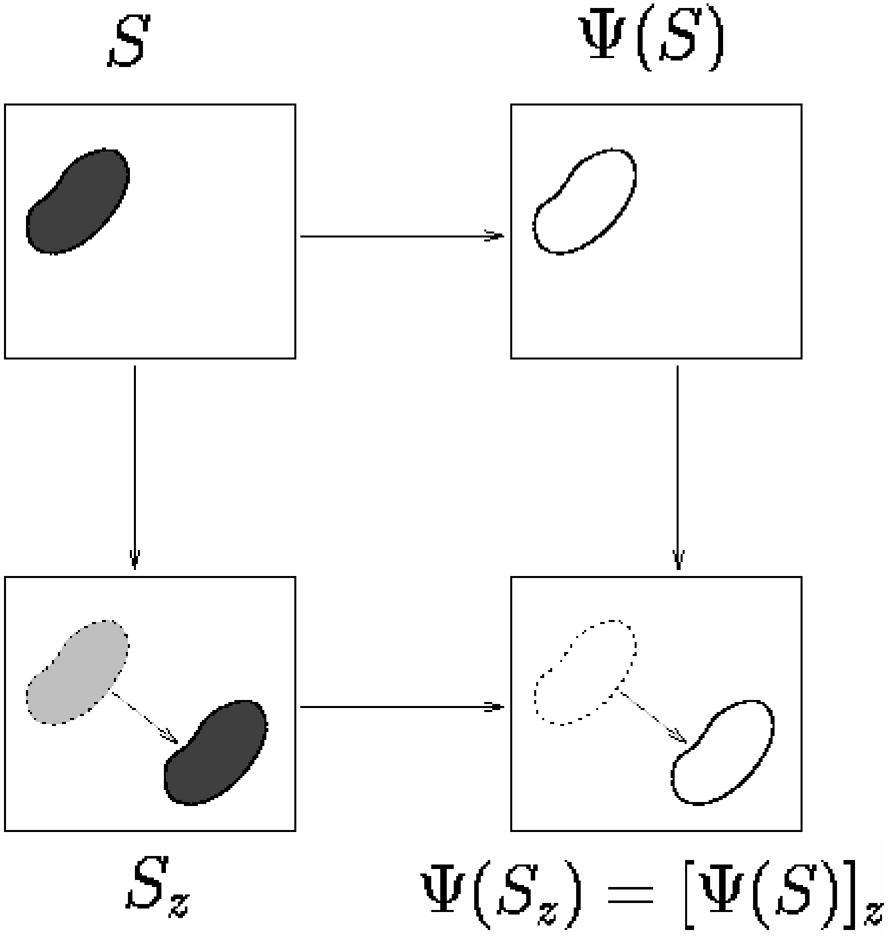
$$\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$$

Translation invariance

- Translation of S by z :

$$S_z = \{x + z : x \in S\}$$

- $\left\{ \begin{array}{l} \Psi : \mathcal{P}(E) \to \mathcal{P}(E) \text{ is} \\ \textbf{translation-invariant iff} \\ \Psi(S_z) = [\Psi(S)]_z \end{array} \right.$

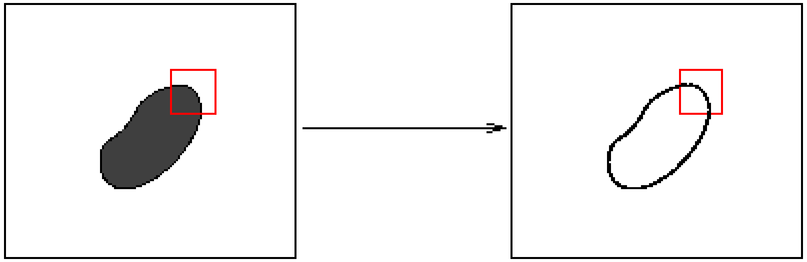


Local definition

Window : $W \subseteq E$

➡ **An image operator is locally defined within W iff**

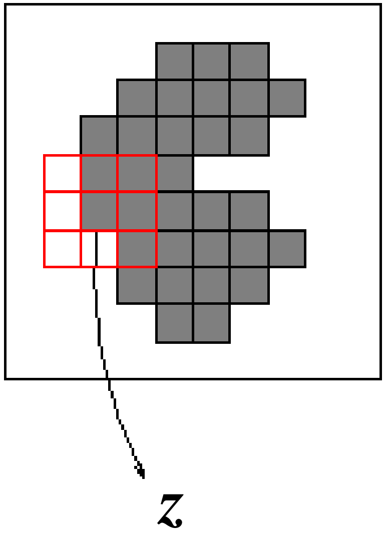
$$x \in \Psi(S) \Longleftrightarrow x \in \Psi(S \cap W_x)$$

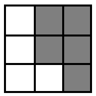


$S \cap W_z$ $\Psi(S \cap W_z)$

W-operators

- **Translation invariance**
  +
  local definition within W
  =
  W **-operators**



$\Psi(S)(z) = \psi\Big($  $\Big)$

- **W-operators are characterized by Boolean functions.**

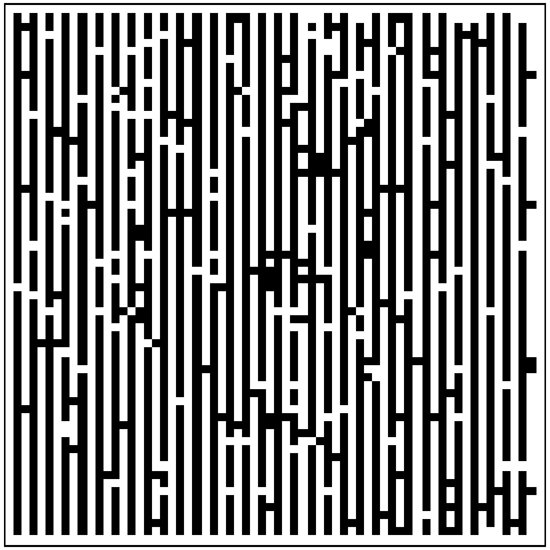
Statistical Hypothesis

X and Y are jointly stationary

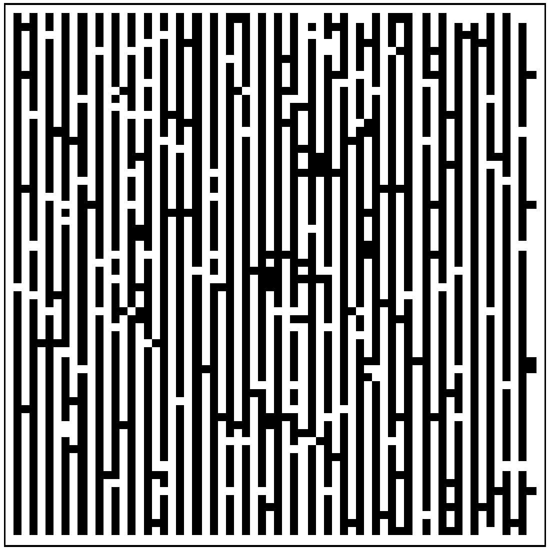
$$P(S \cap W_z, Y)$$

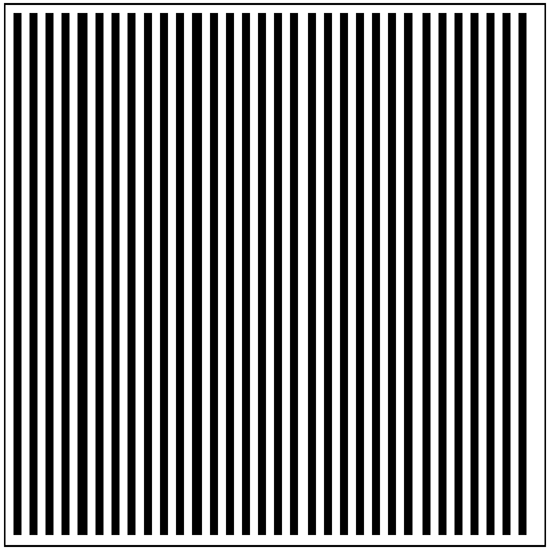
is the same for any z in E

Stationary Process

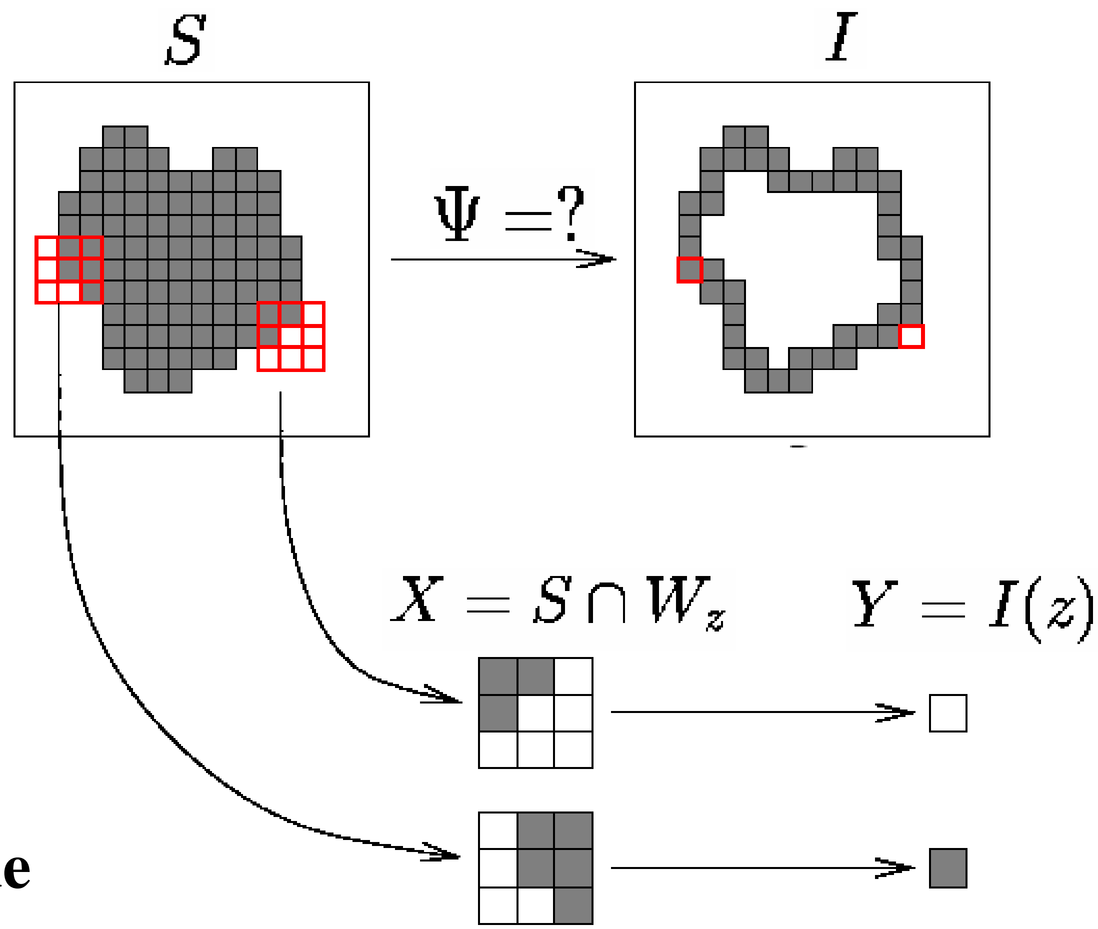


Join Stationary Process





Design procedure

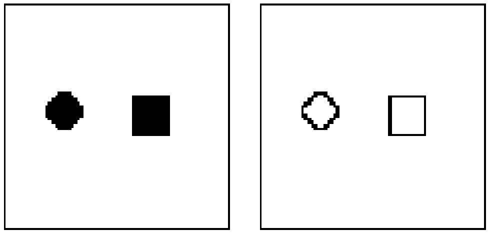


Uses

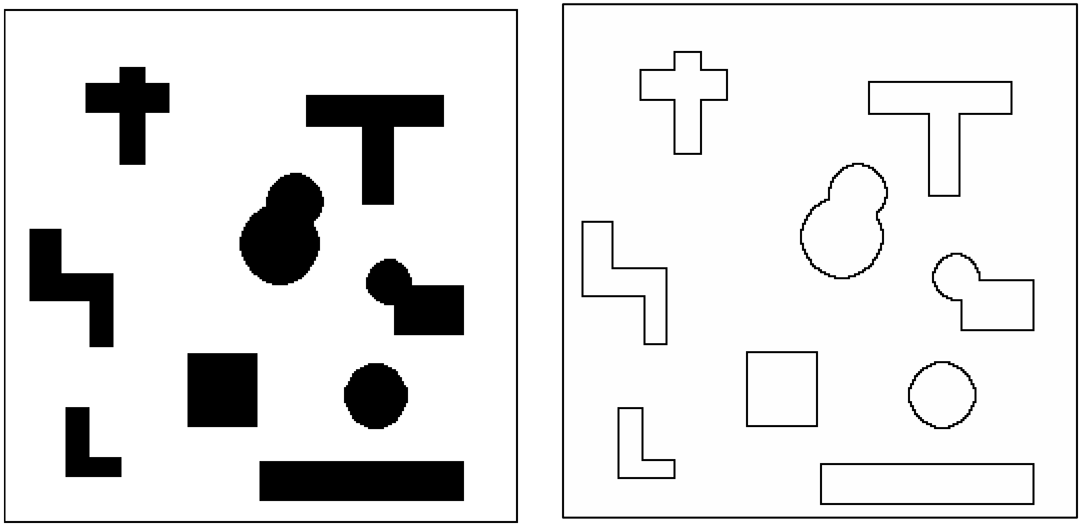
training data

learning technique

Edge detection



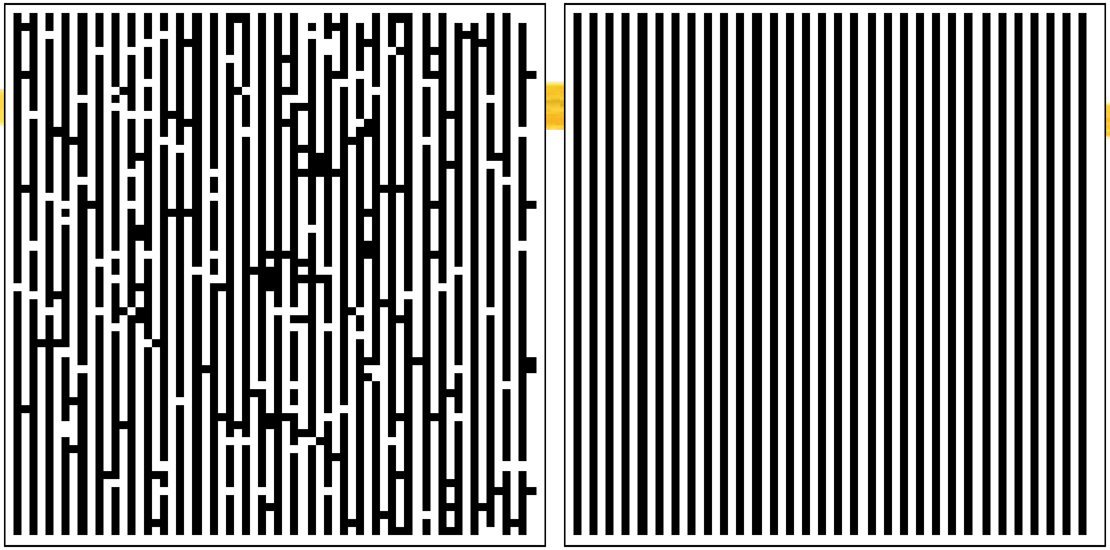
Training images



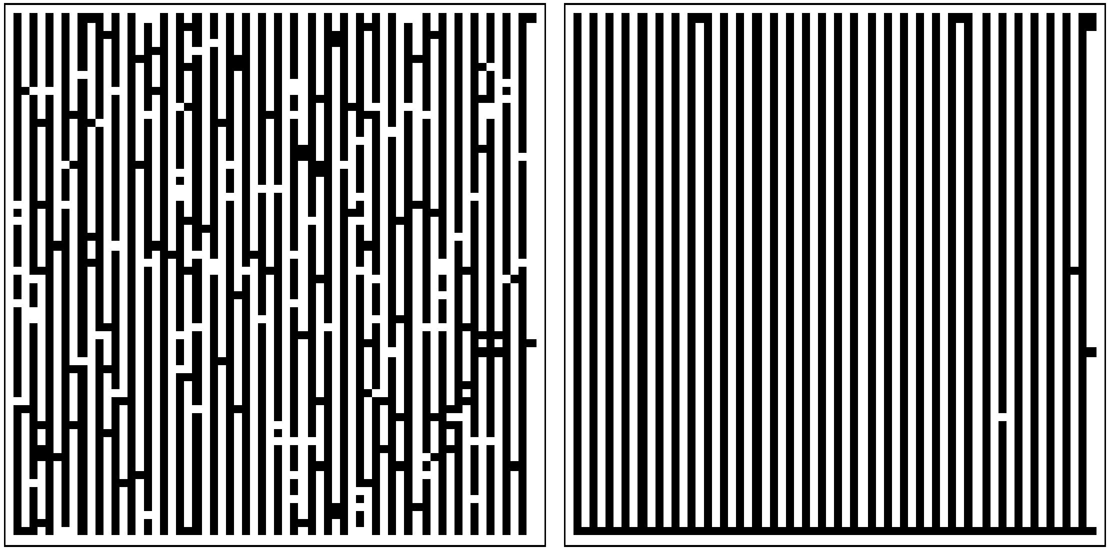
Test images

Noise filtering

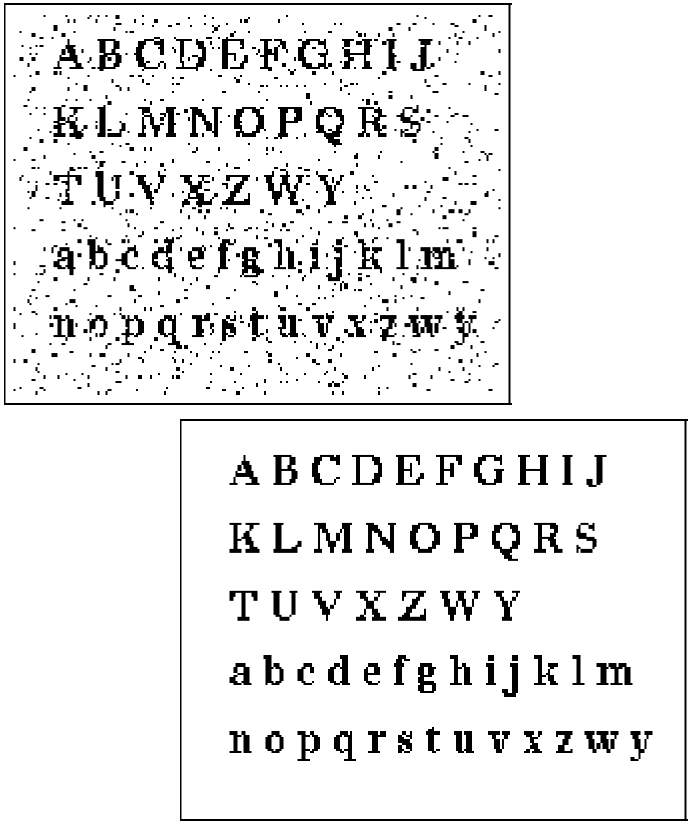
Training images



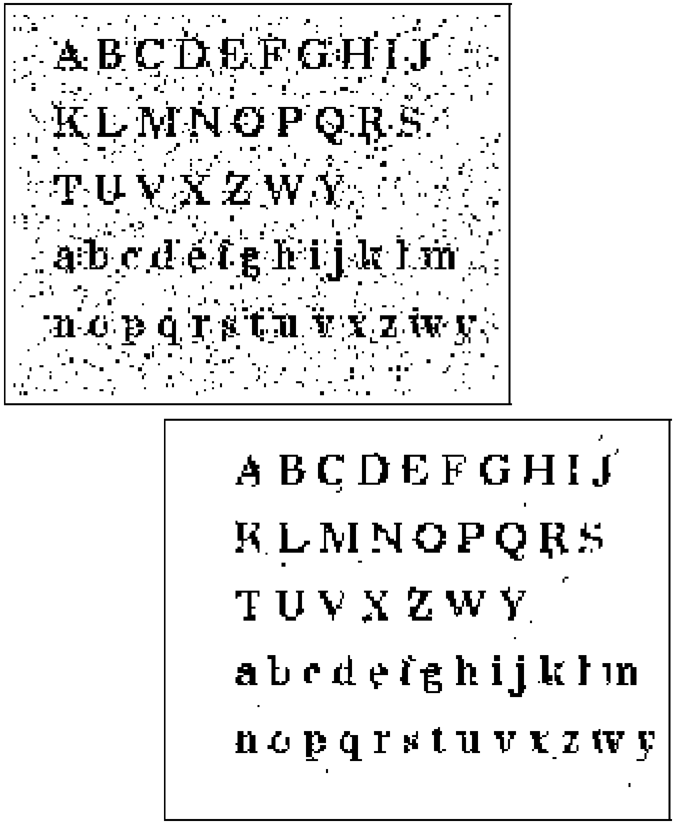
Test images



Noise removal

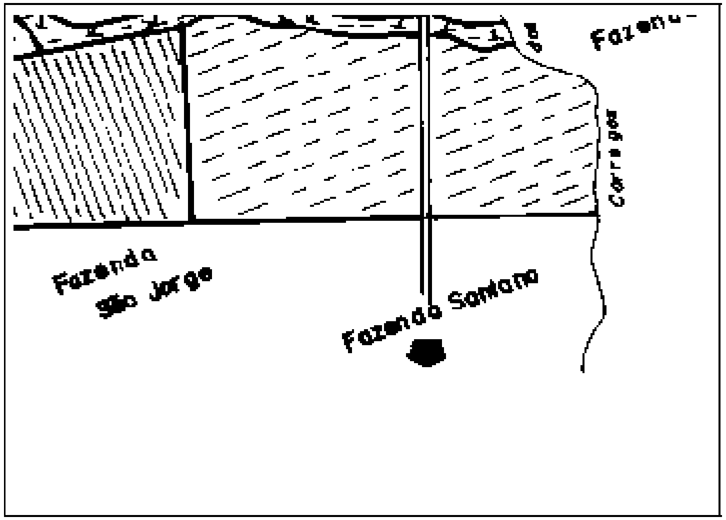


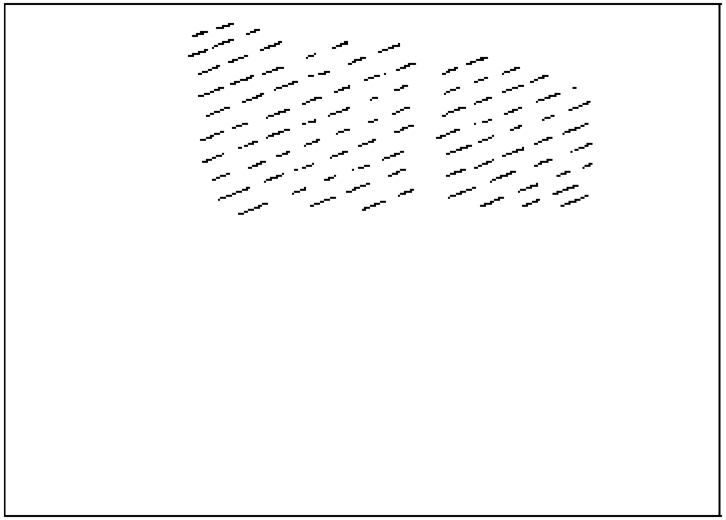
Training images



Test images

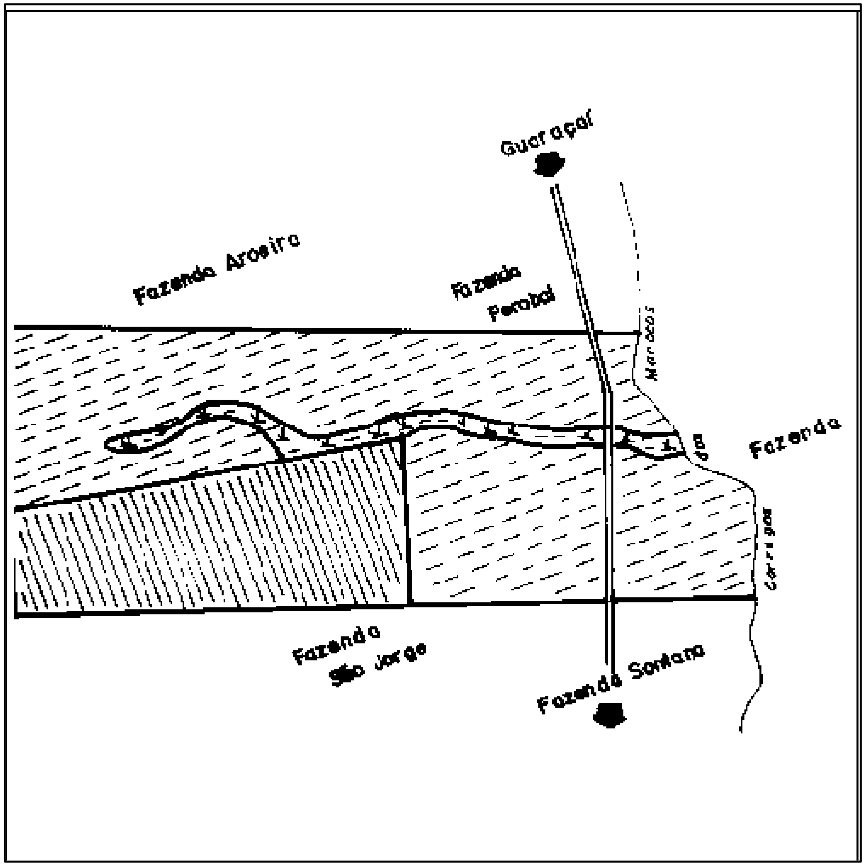
Texture extraction (1)

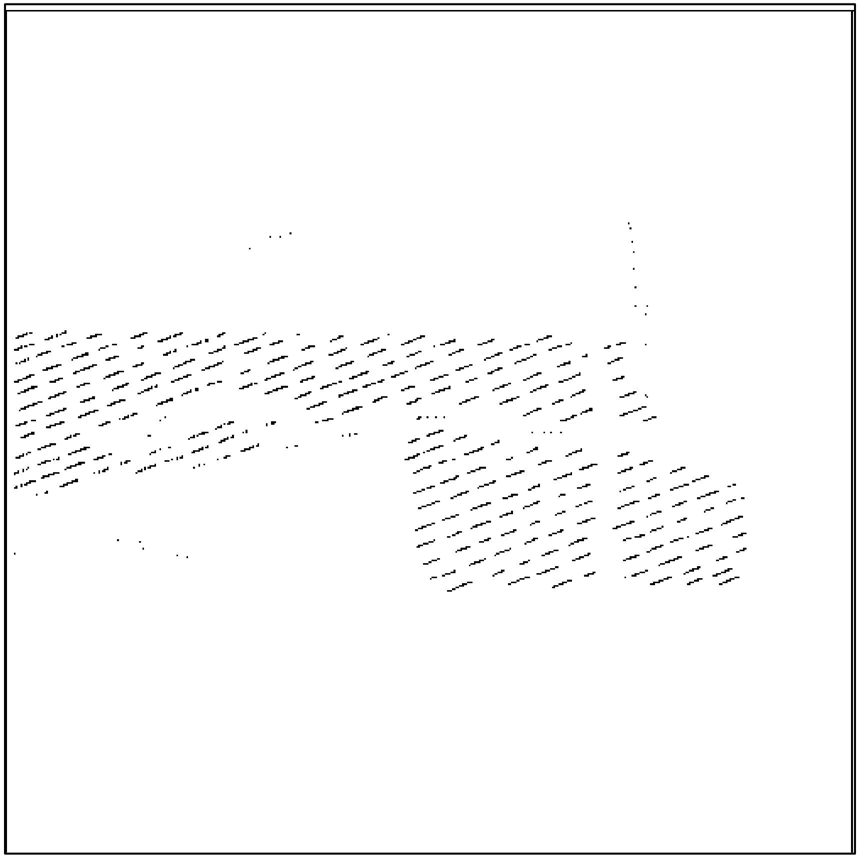




Training images

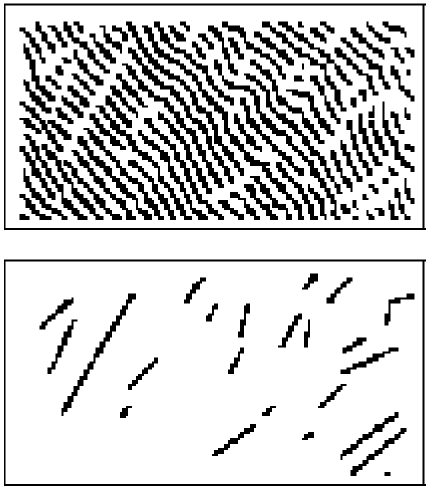
Texture extraction (2)



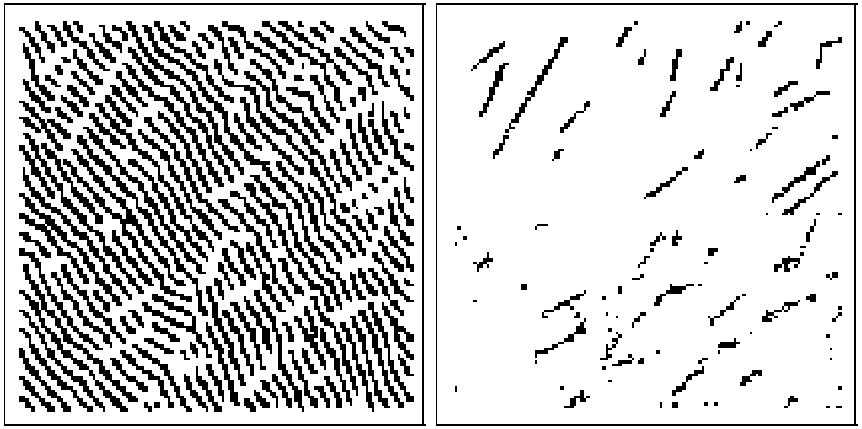


Test images

Example

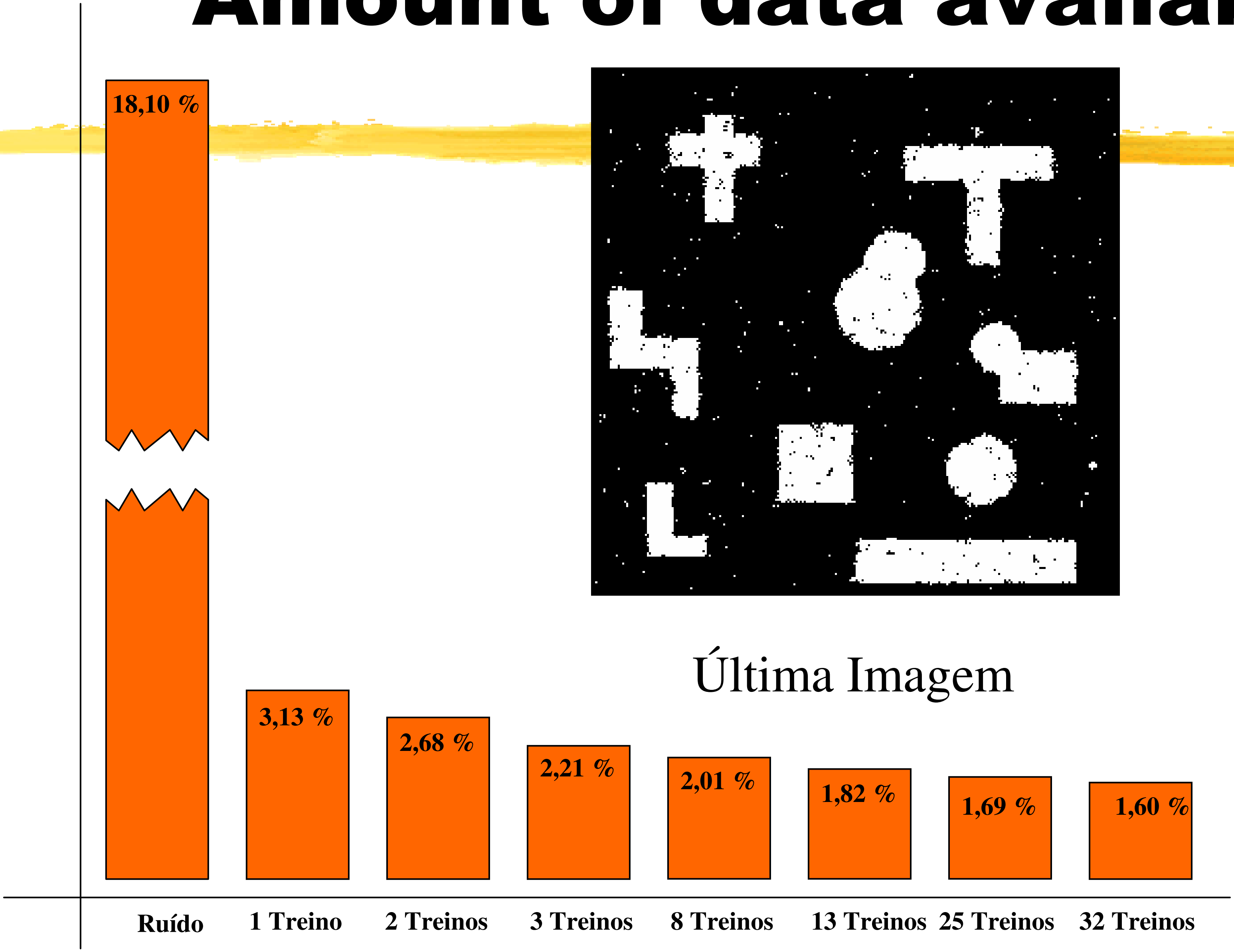


Training images



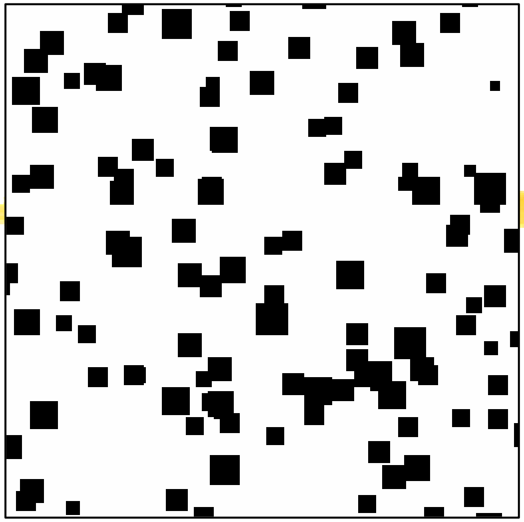
Test images

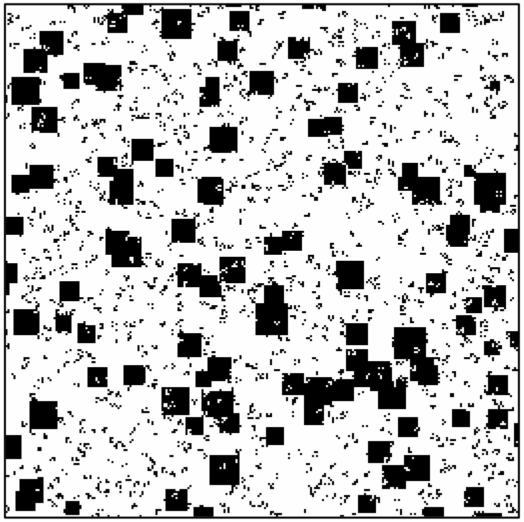
Amount of data available



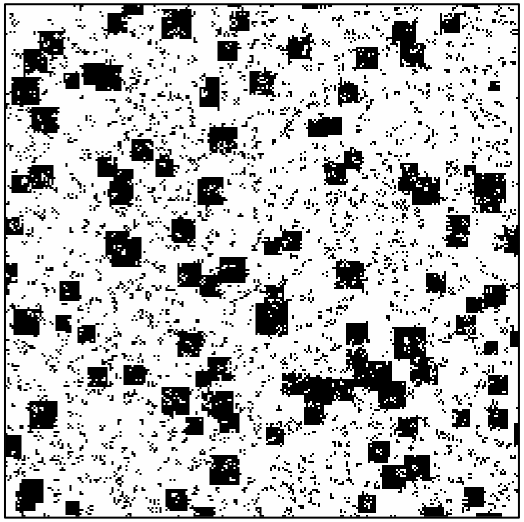
Última Imagem

Distribution

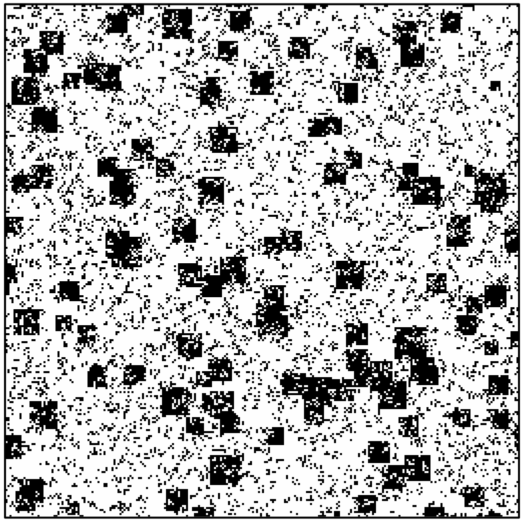




aditivo: 2%
subtrativo: 1%

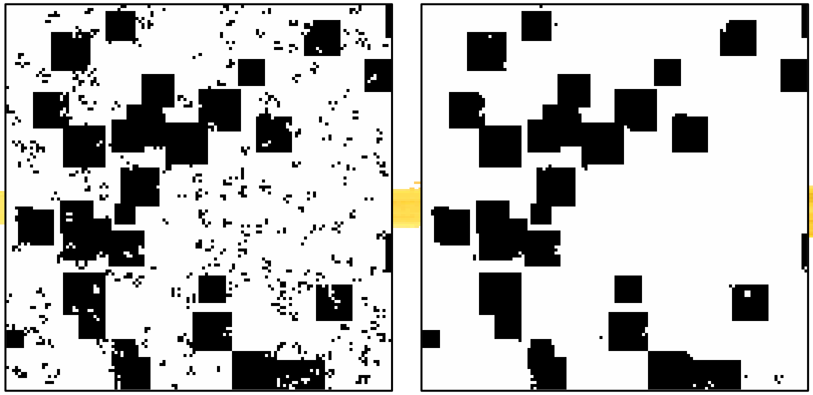


aditivo: 3%
subtrativo: 3%

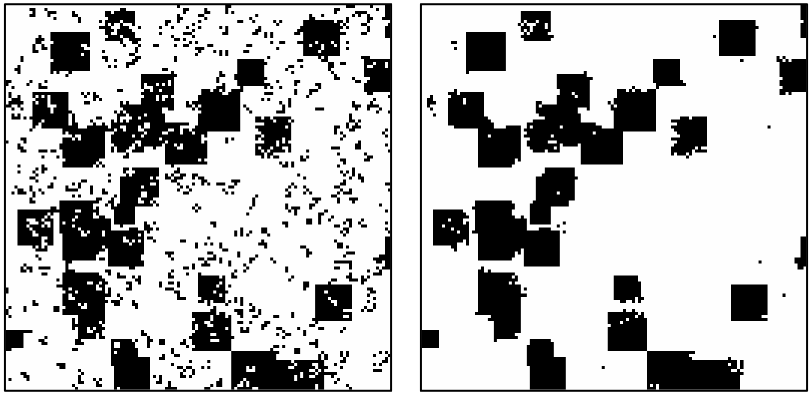


aditivo: 6%
subtrativo: 6%

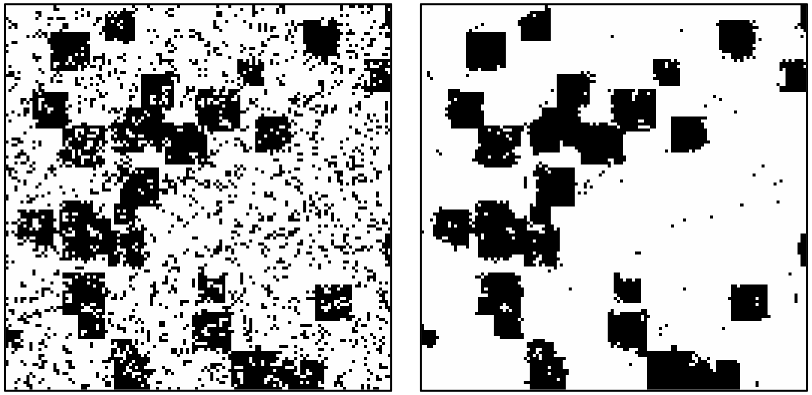
window 5x5, 6 training images



aditivo: 2%
subtrativo: 1%
padrões distintos : 140.060
em 1.548.384

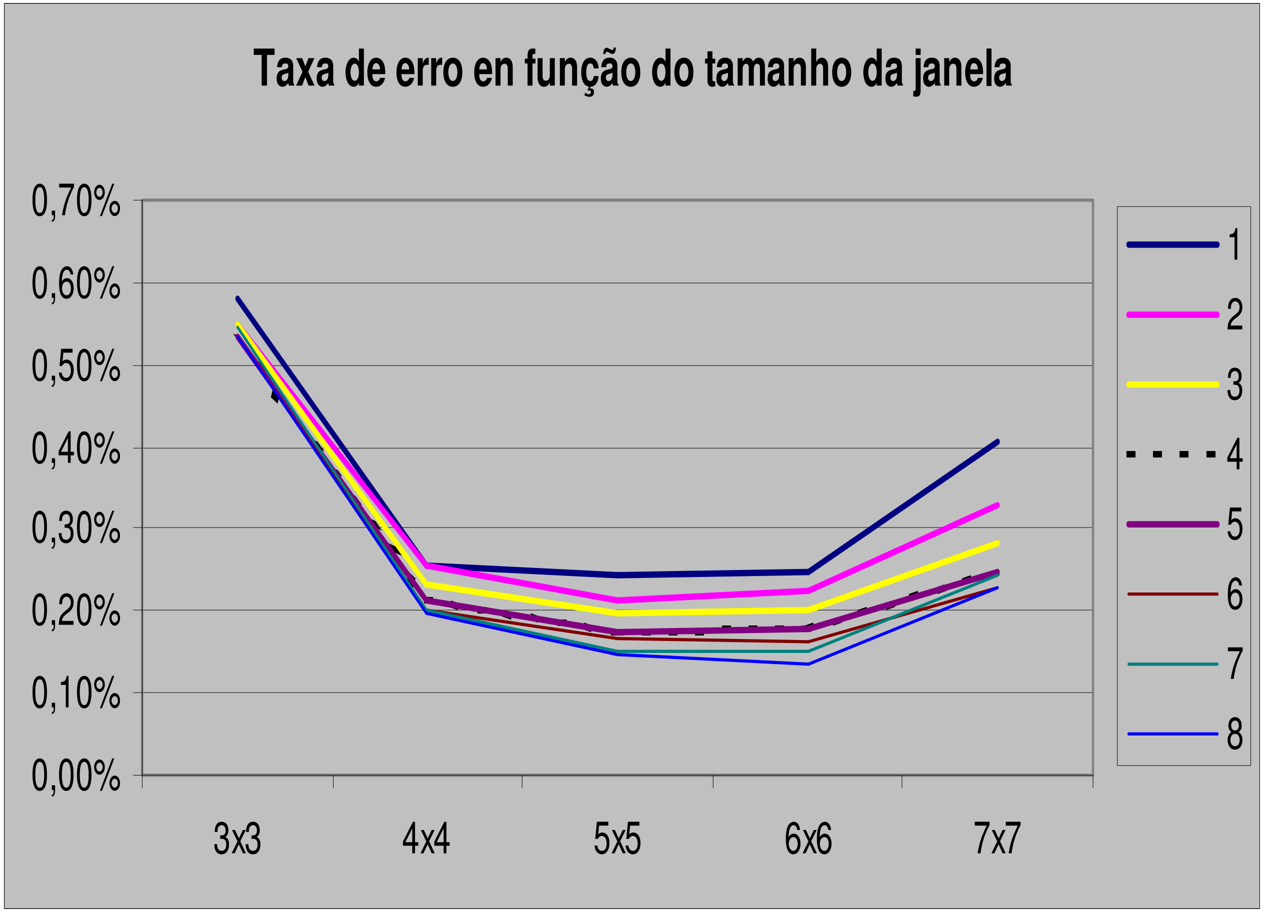


aditivo: 3%
subtrativo: 3%
padrões distintos : 266.743
em 1.548.384

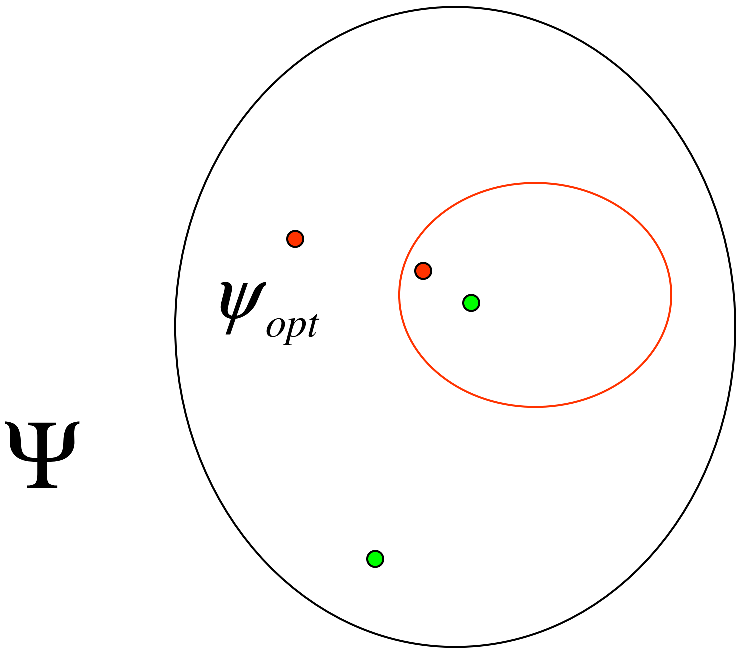


aditivo: 6%
subtrativo: 6%
padrões distintos : 487.494
em 1.548.384

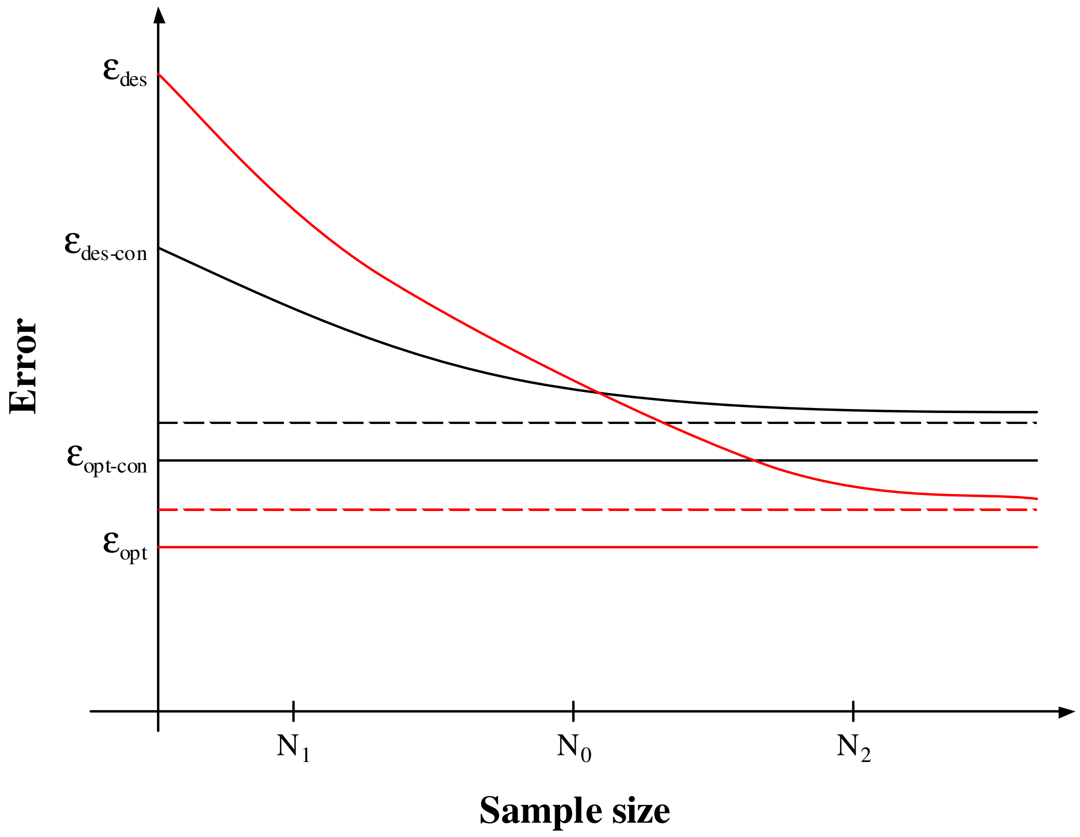
Size of the window



Size of the window



Size of the window



Difficulties

- **The space of *W*-operators is VERY large.**
- $$|W| = n \Longrightarrow \begin{cases} 2^{2^n} & W \text{ operators,} \\ 2^n & \text{conditional probabilities to be estimated} \end{cases}$$
- ***Consequences :***
  - Large amount of data (training images) are required for a good estimation of these parameters
  - Learning algorithm complexity increases

Difficulties

x1	x2	p(-1,x1,x2)	p(0,x1,x2)	p(1,x1,x2)	p(x1,x2)	y	Error
-1	-1	0.05	0.1	0.05	0.2	0	0.1
-1	0	0.03	0.03	0.04	0.1	1	0.06
-1	1	0.02	0.01	0.07	0.1	1	0.03
0	-1	0.01	0.01	0.03	0.05	1	0.02
0	0	0.03	0.01	0.01	0.05	-1	0.02
0	1	0.07	0.1	0.03	0.2	0	0.1
1	-1	0.04	0.06	0.1	0.2	1	0.1
1	0	0.03	0.01	0.01	0.05	-1	0.02
1	1	0.02	0.02	0.01	0.05	-1	0.03
							0.48

Ideal design

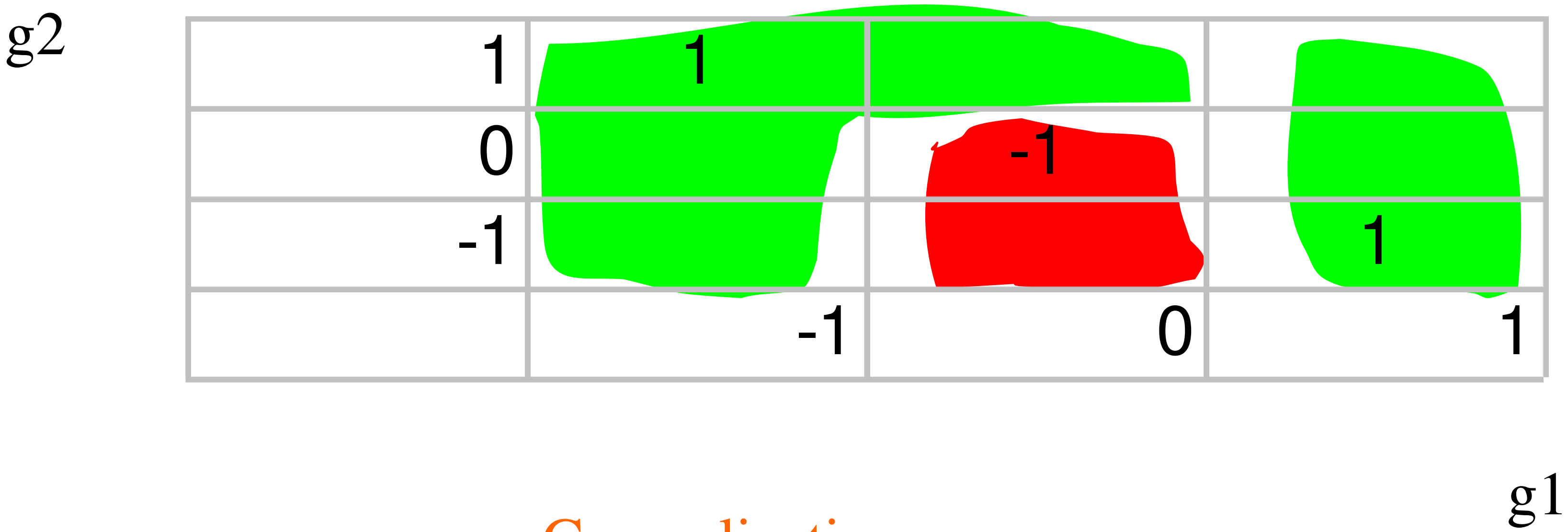
Difficulties

x1	x2	p(-1,x1,x2)	p(0,x1,x2)	p(1,x1,x2)	p(x1,x2)	y	Error
-1	-1						
-1	0						
-1	1	0.02	0.01	0.07	0.1	1	0.03
0	-1						
0	0	0.03	0.01	0.01	0.05	-1	0.02
0	1						
1	-1	0.04	0.06	0.1	0.2	1	0.1
1	0						
1	1						

Real design

Difficulties

g3 = f(g1,g2)



Generalization

Constraints

- **Structural Constraints**

  - impose maximum number of elements in the basis

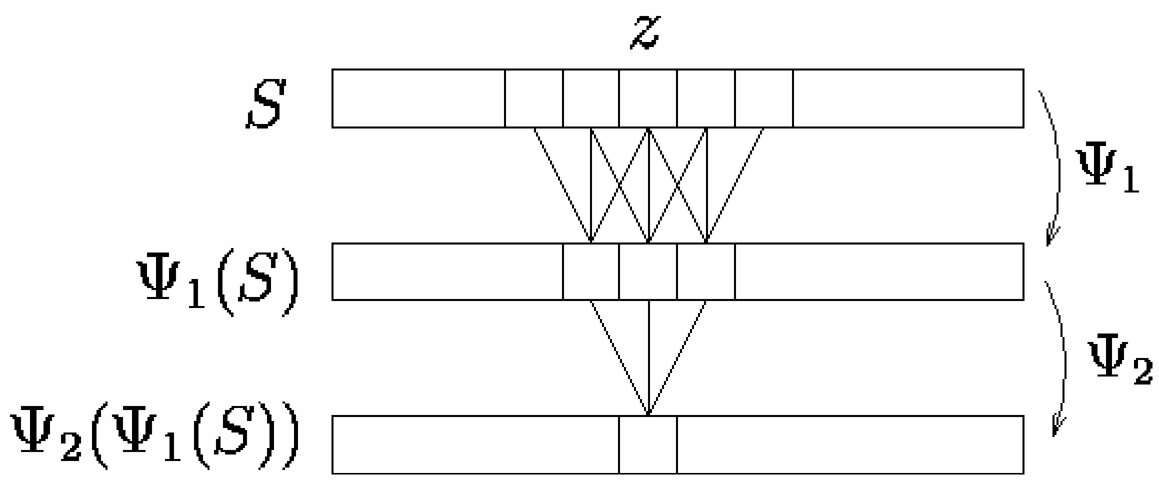
  - use alternative structural representations (e.g., sequential)

- **Algebraic constraints**

  - consider class of operators satisfying a given algebraic property (e.g., increasingness , idempotence, auto-dualism, etc)

Structural Constraint : iterative design (1)

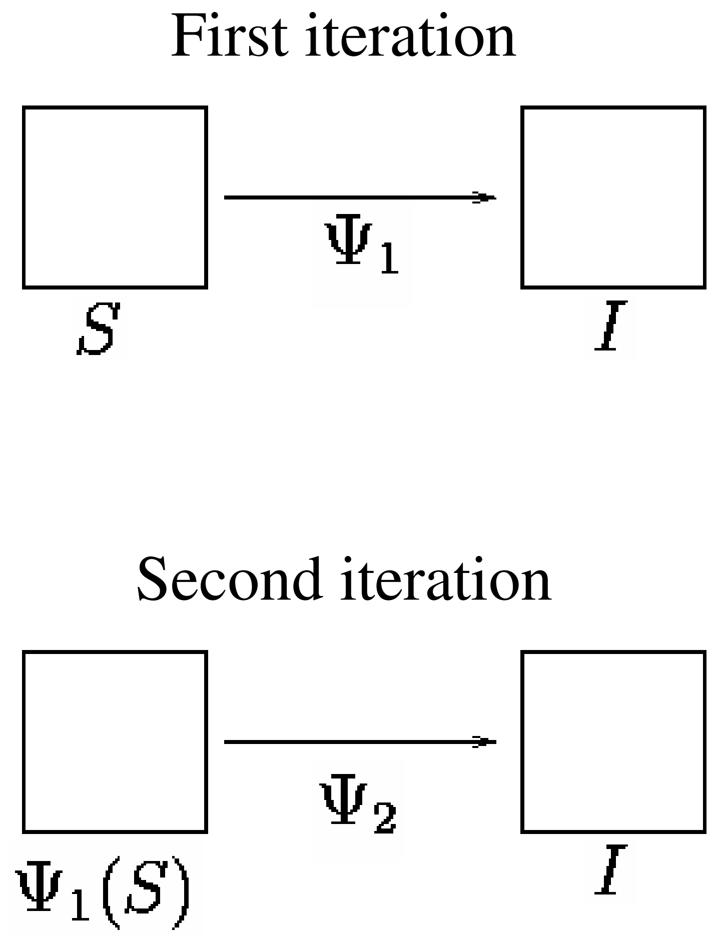
- *Motivation* : composition of operators over small windows produces an operator over a larger window



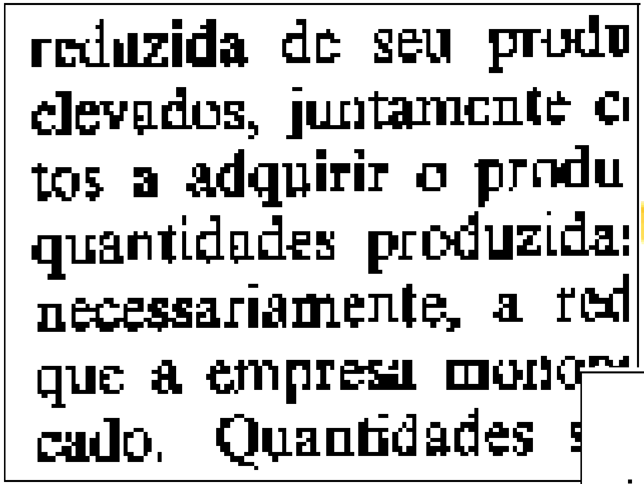
- $\Psi = \Psi_2(\Psi_1)$ is a $W \oplus W$-operator

Iterative design procedure

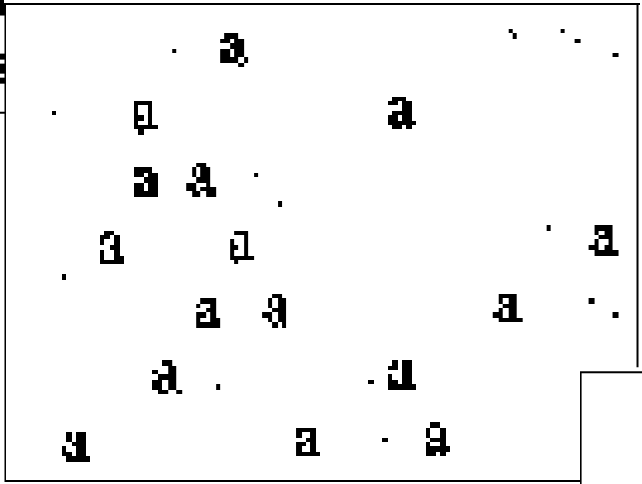
- **Successive application of the single iteration design procedure**



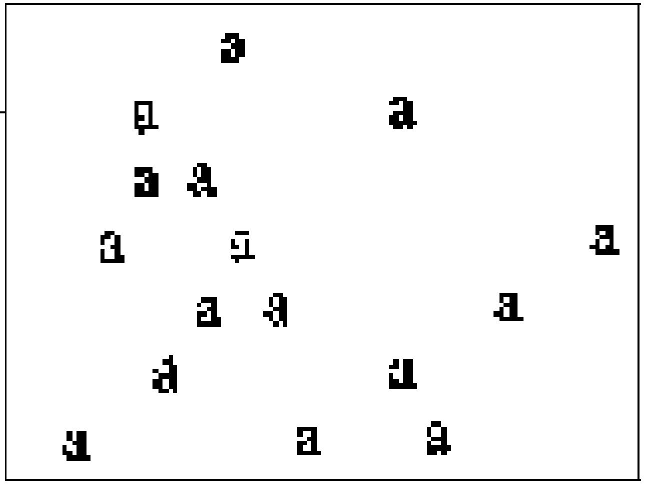
Application example



test image

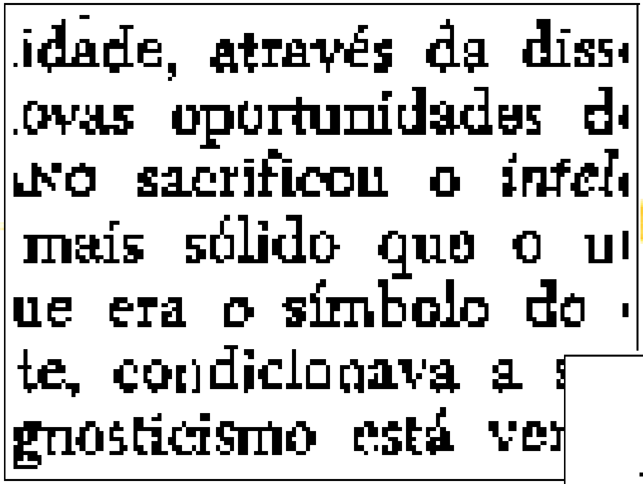


iteration 1

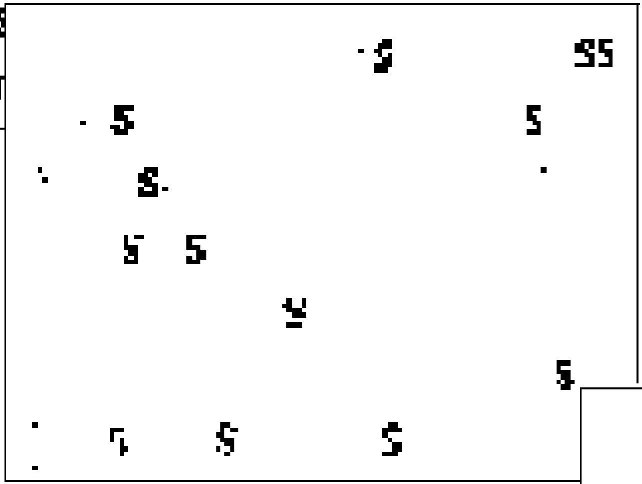


iteration 2

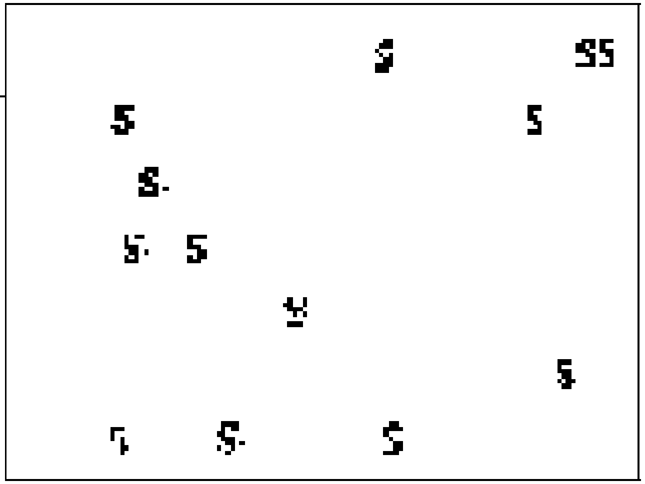
Application example



test image

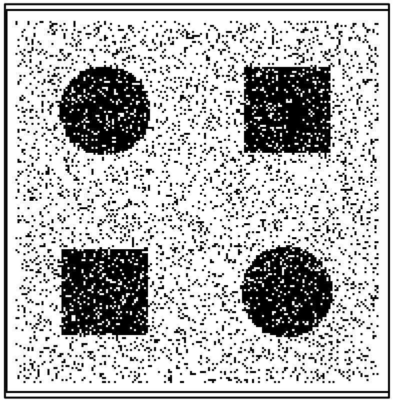


iteration 1

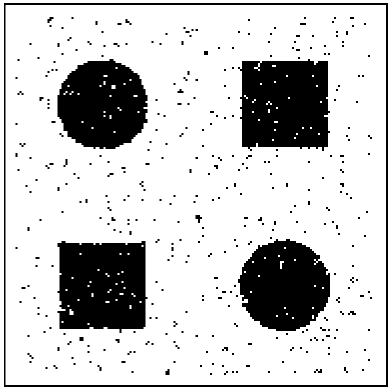


iteration 2

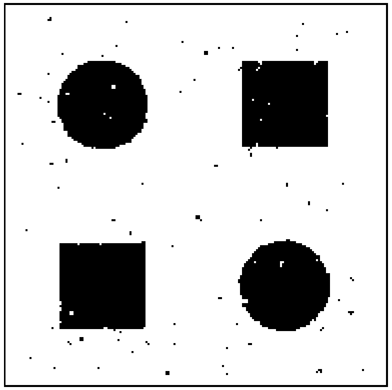
Application example



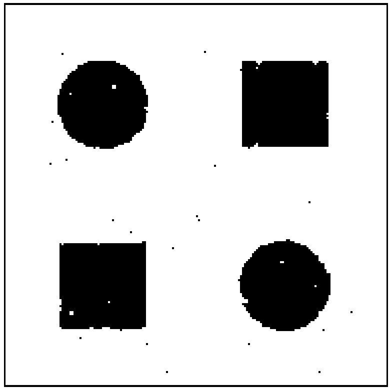
test image



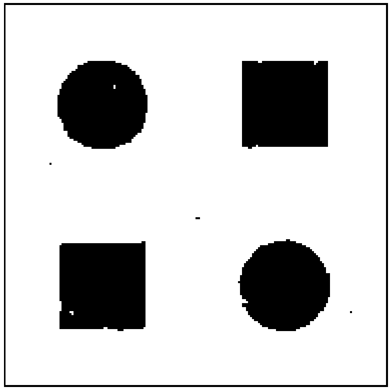
iteration 1



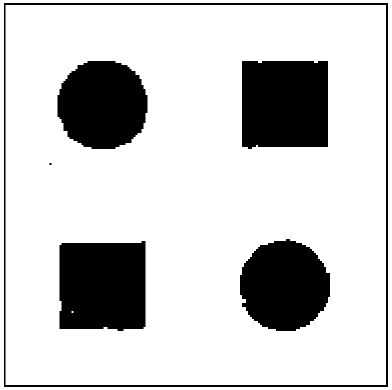
iteration 2



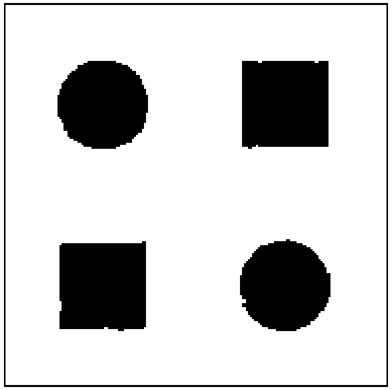
iteration 3



iteration 4

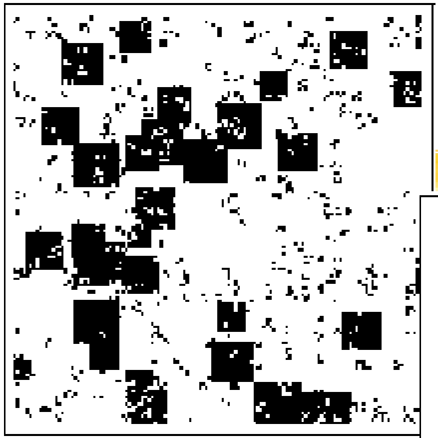


iteration 5



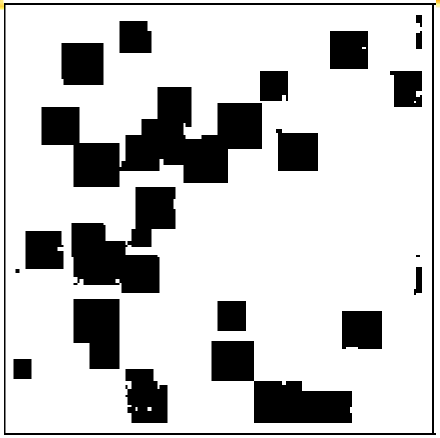
iteration 6

Application example

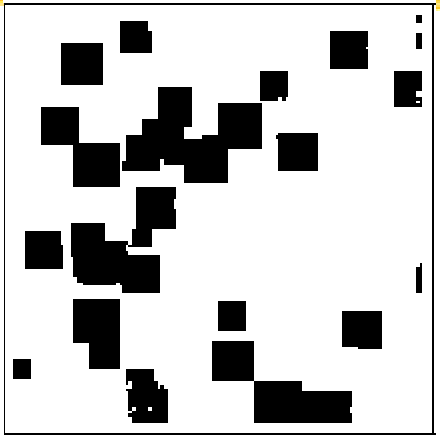


test image

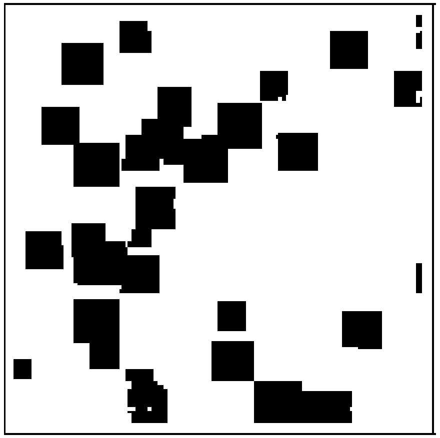
iteration 1



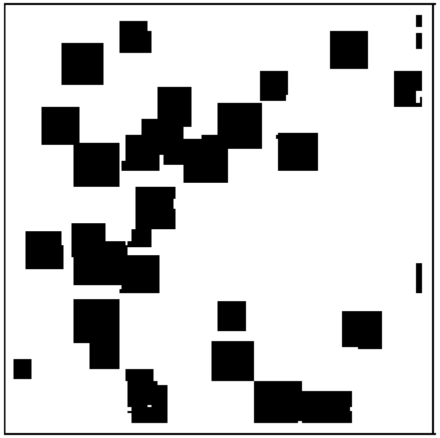
iteration 2



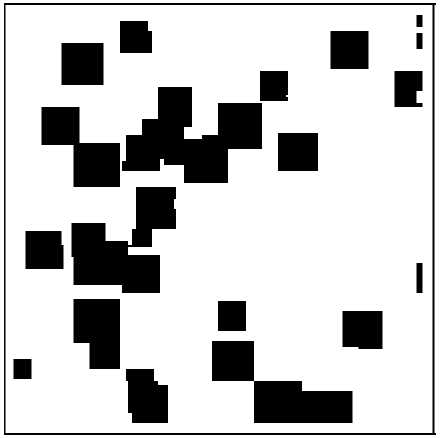
iteration 3



iteration 4

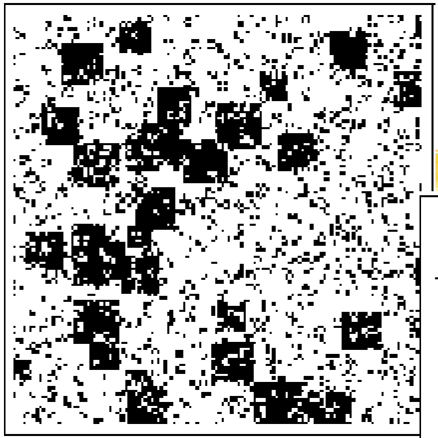


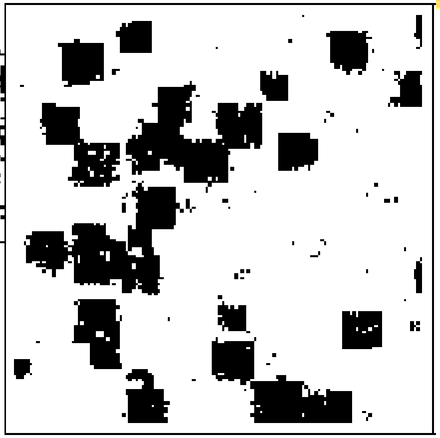
iteration 5

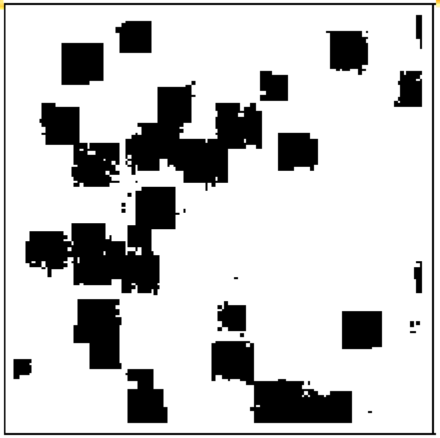


iteration 6

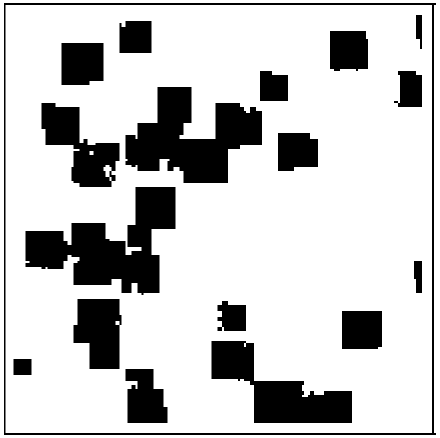
Application example

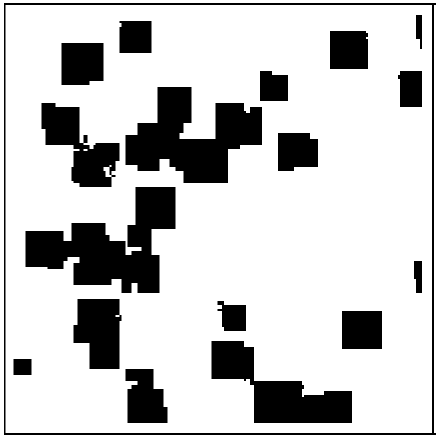

test image

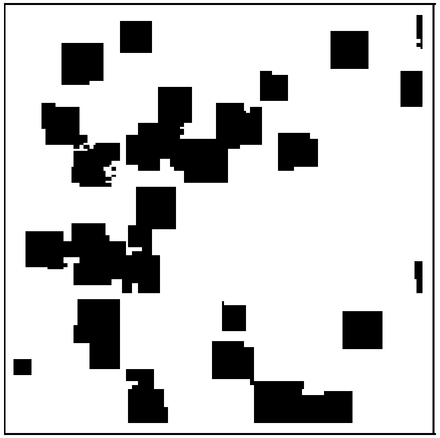

iteration 1


iteration 2


iteration 3


iteration 4


iteration 5


iteration 6

Algebraic constraints

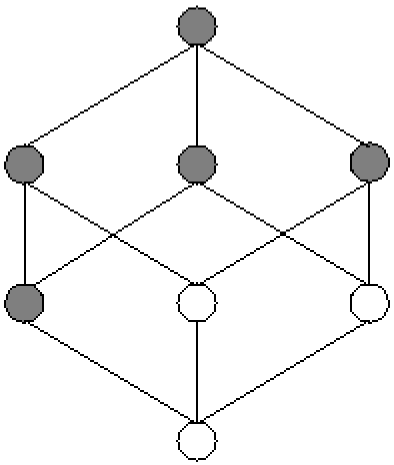
Design of operators based on the switching approach

- **estimate optimal W-operator**
- **switch value of the optimal W-operator in such a way that the resulting operator satisfies the algebraic constraint**

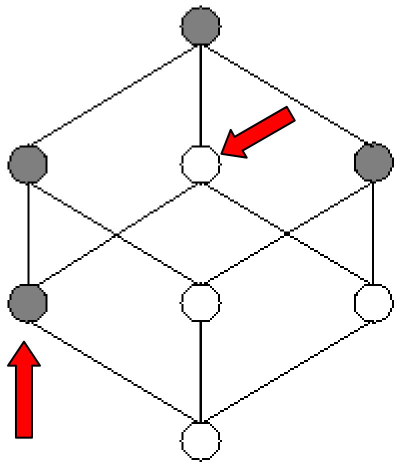
Algebraic constraints

Increasing *W*-operators

$$x \leq y \Rightarrow \psi(x) \leq \psi(y)$$



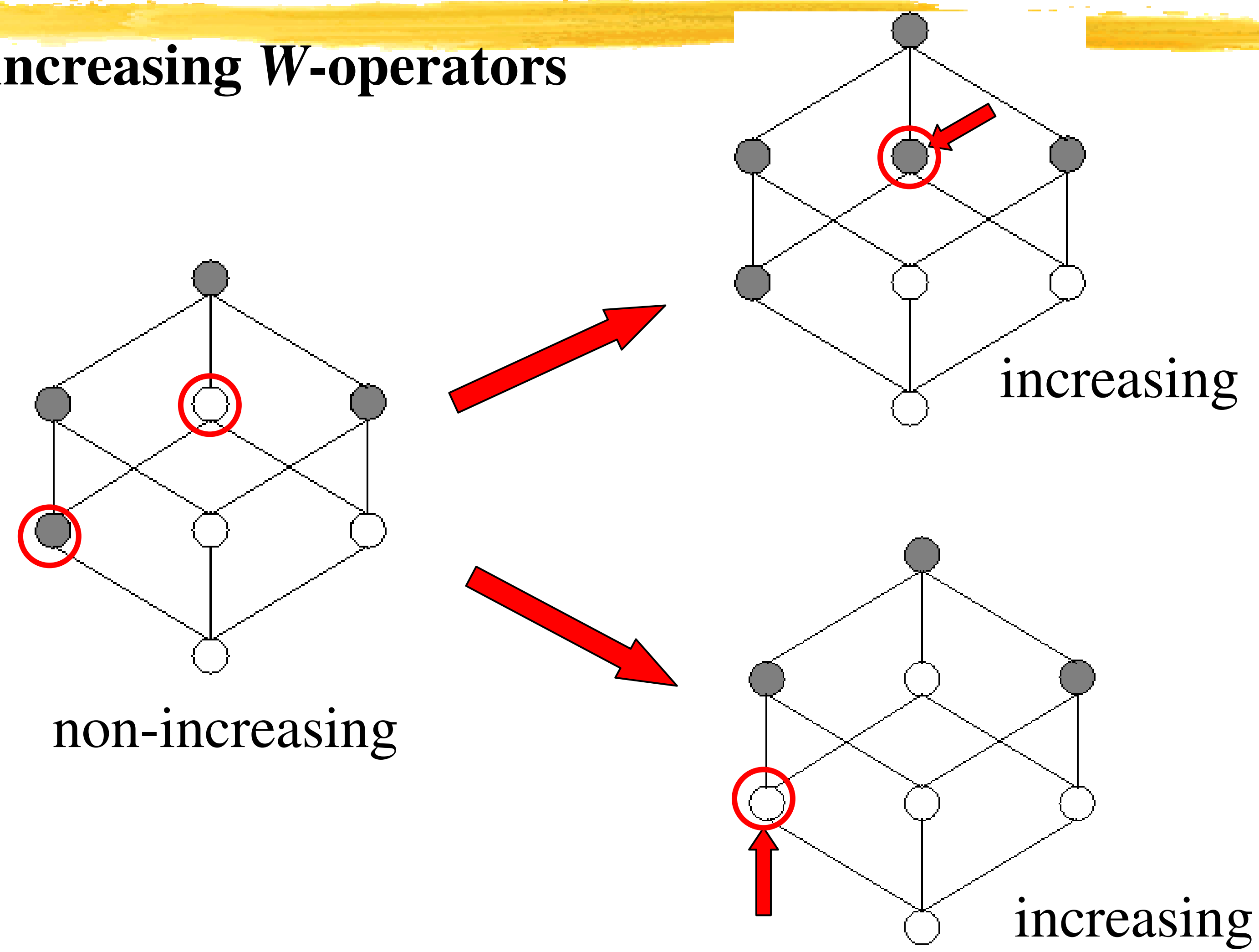
increasing



non-increasing

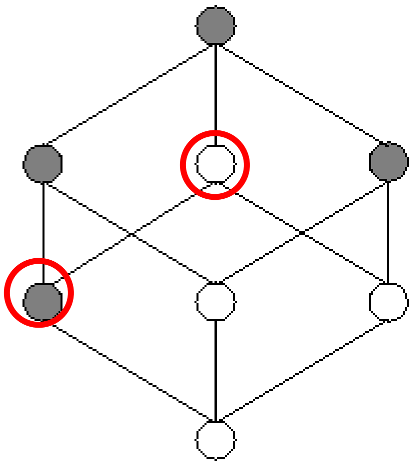
Switching approach

Design of increasing *W*-operators



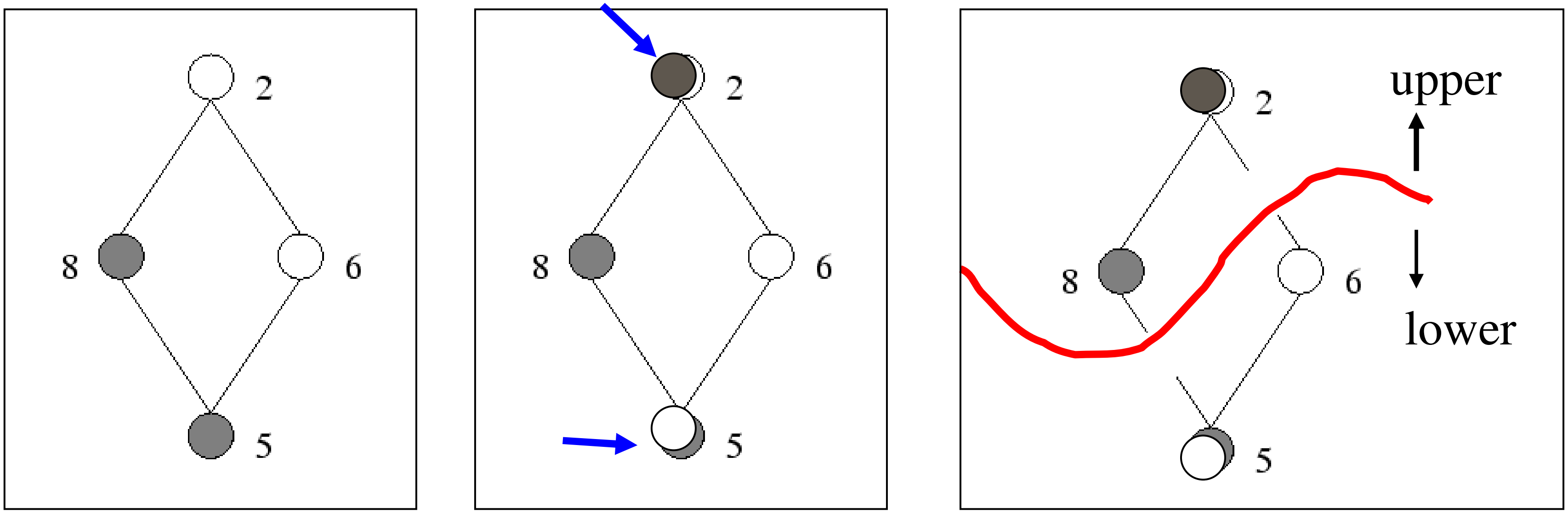
Switching approach

Inversion set



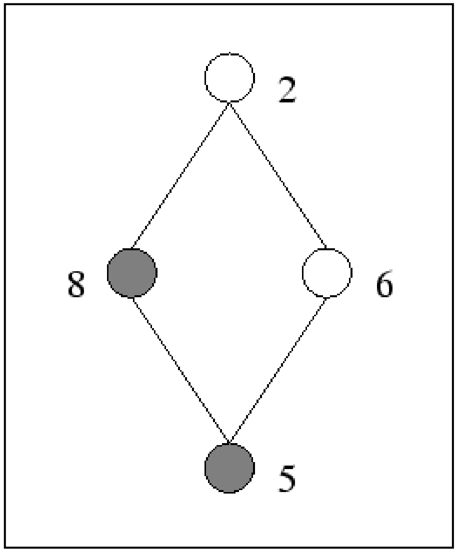
- Only inversion set elements need to be switched.
- Switching may increase risk
- There exists a switching cost (amount of risk increase due to the switching) for each element in the inversion set.
- *Goal* : find switching that minimizes overall risk increase

Switching → Partition

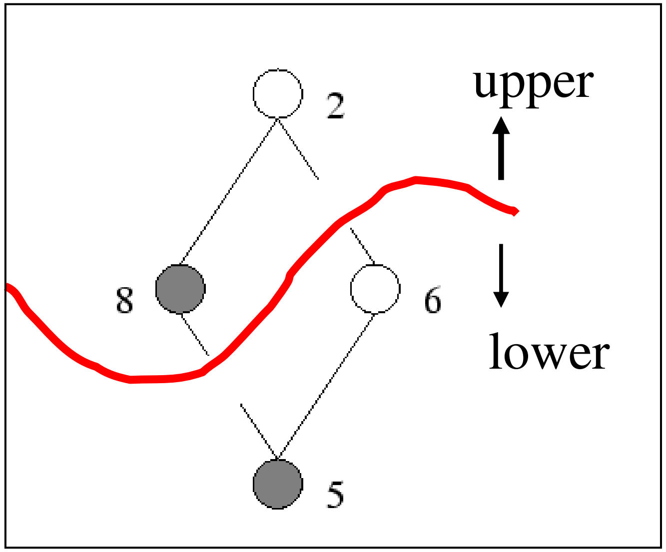


Inversion set After switching partition

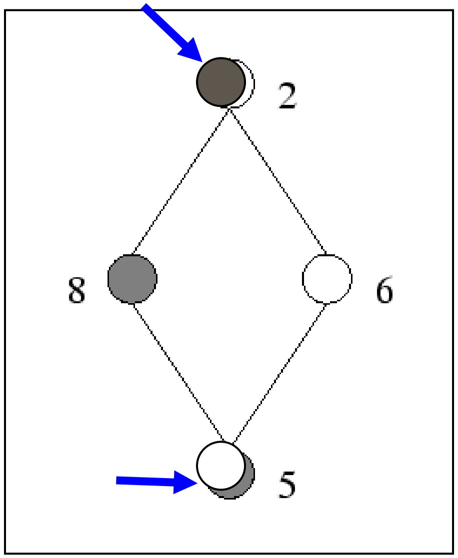
Partition ➡ Switching



Inversion set



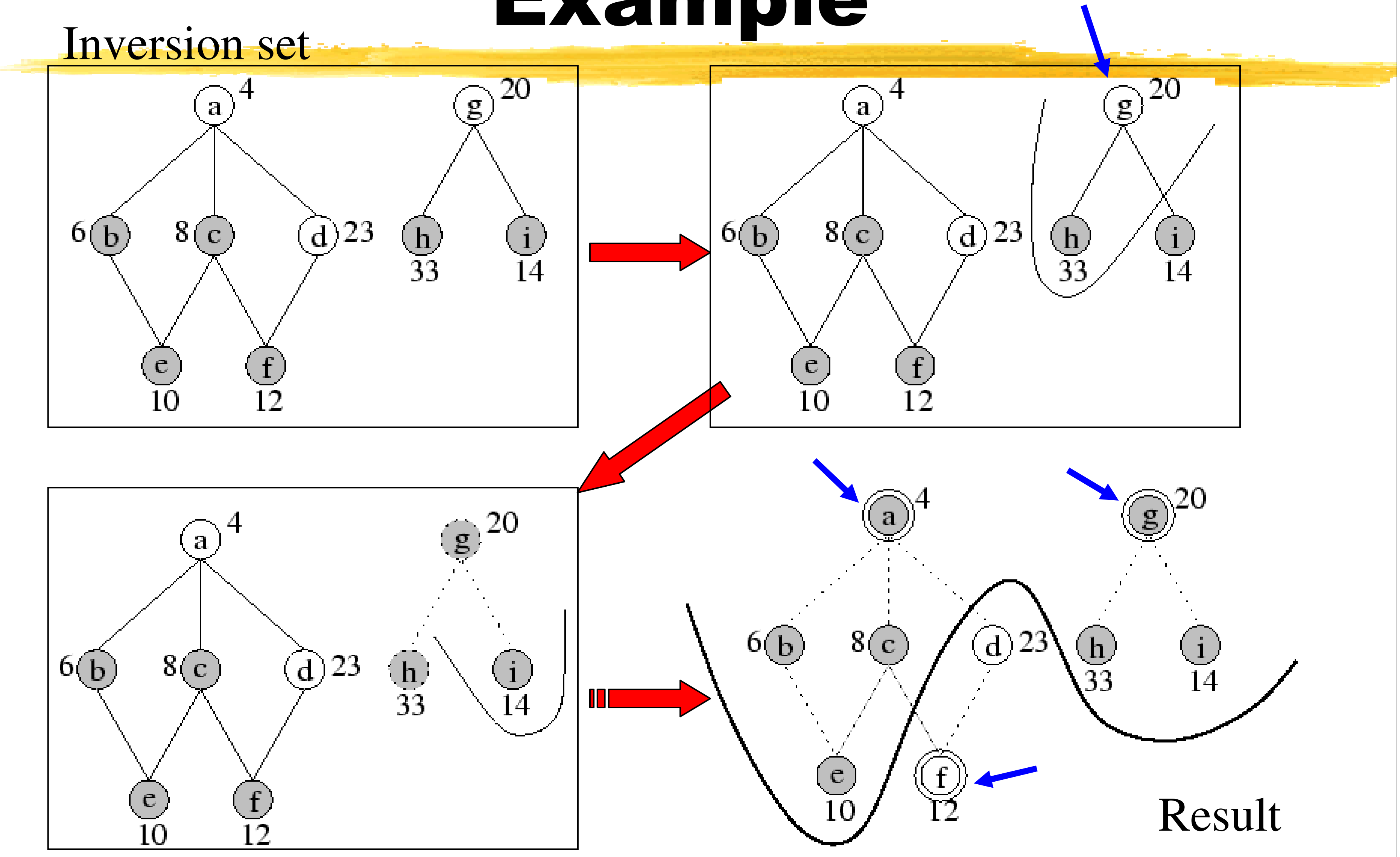
partition



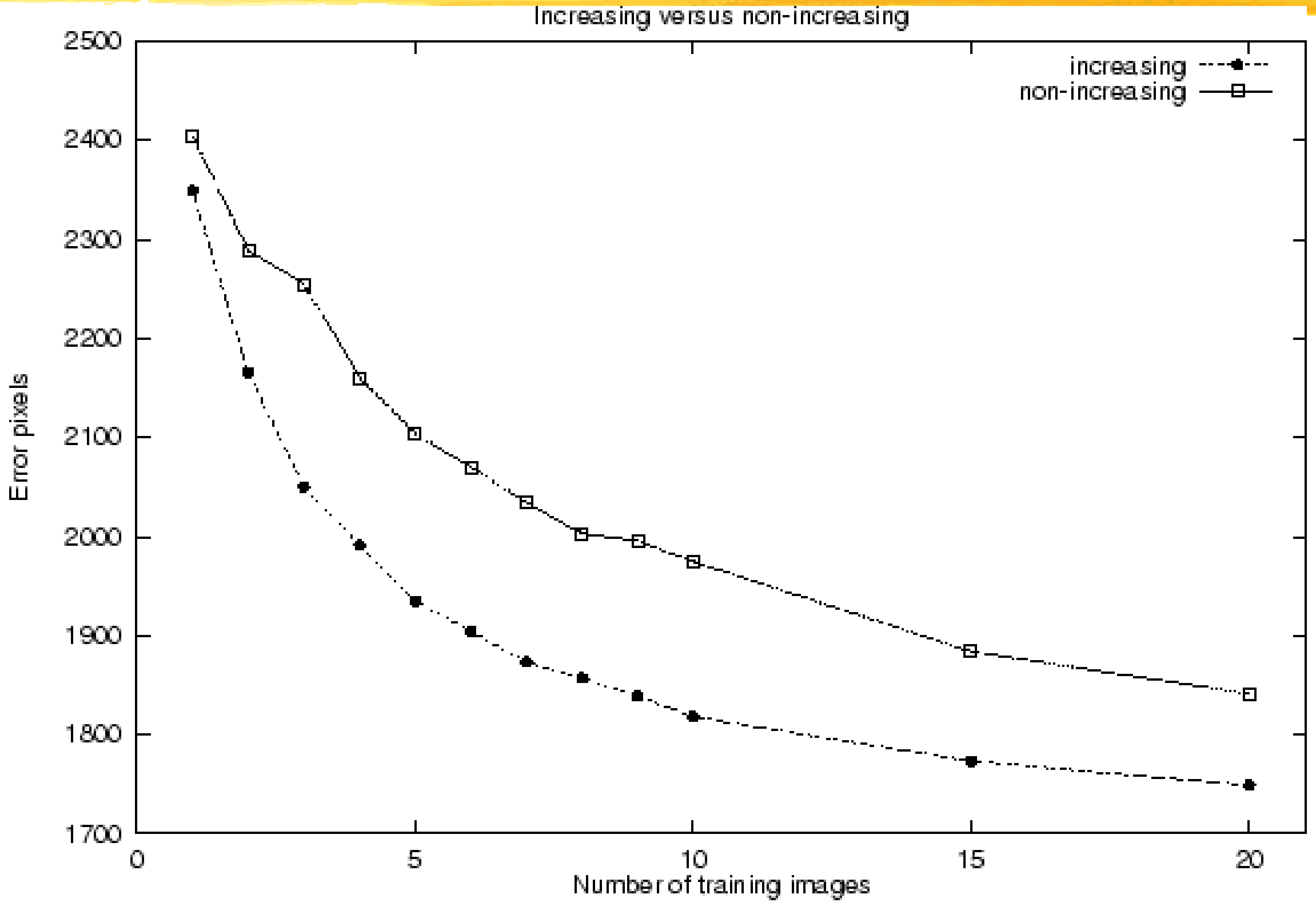
Implied switching

Example

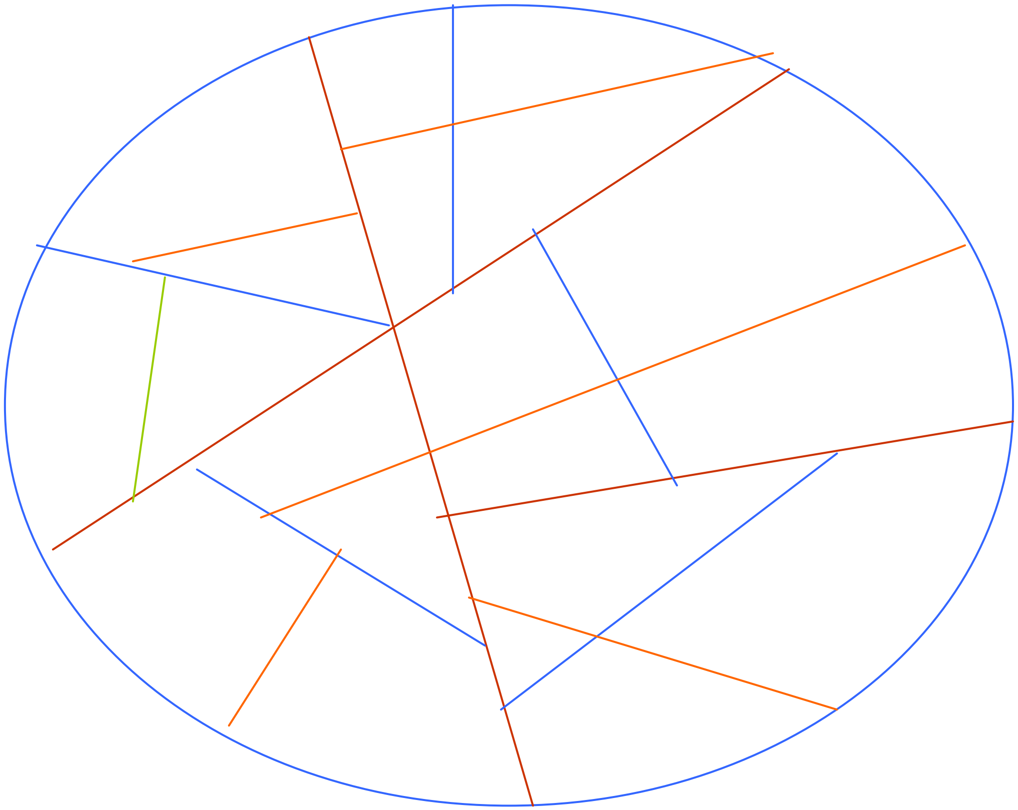
Inversion set



aditivo: 2%
subtrativo: 2%

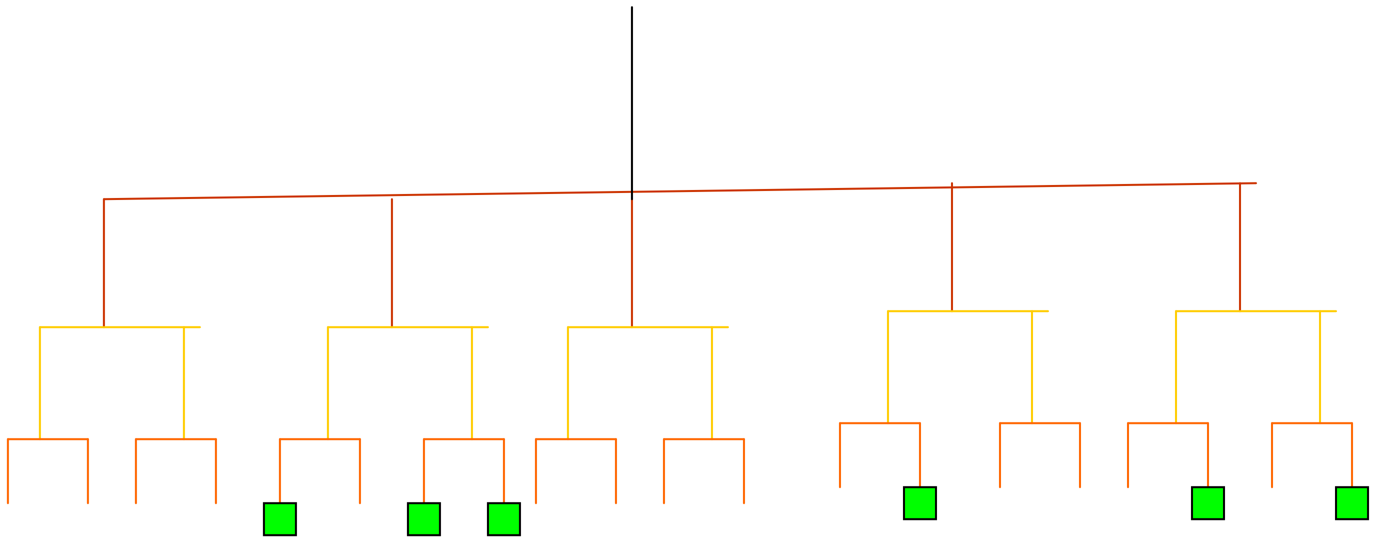


Hierarchical clustering

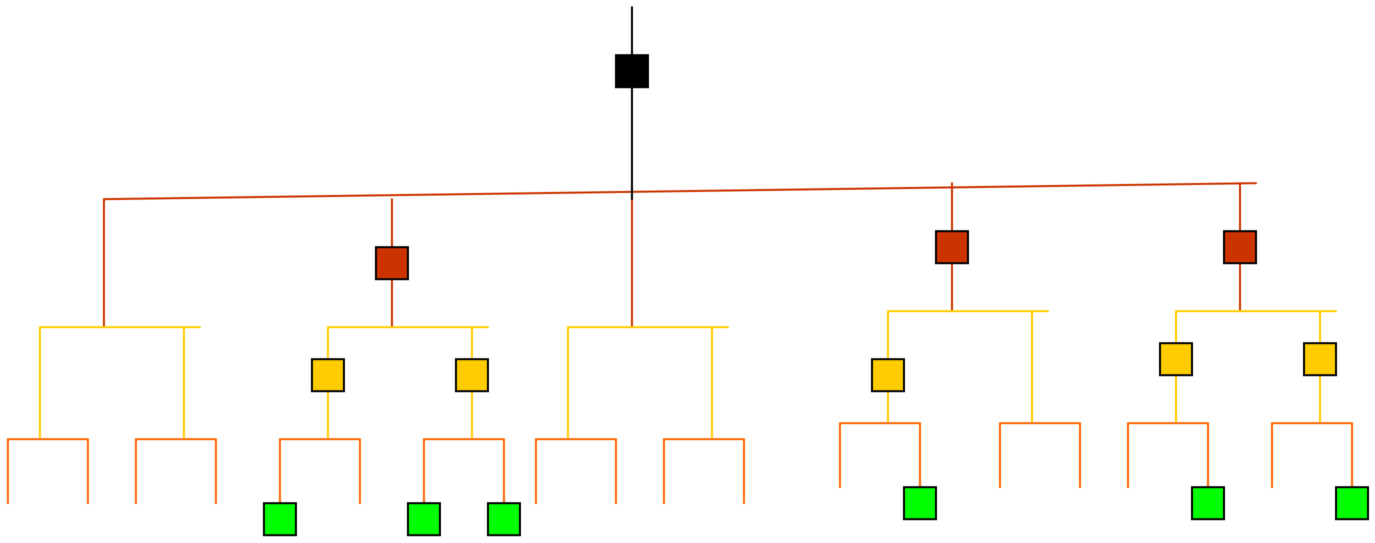


Shapes

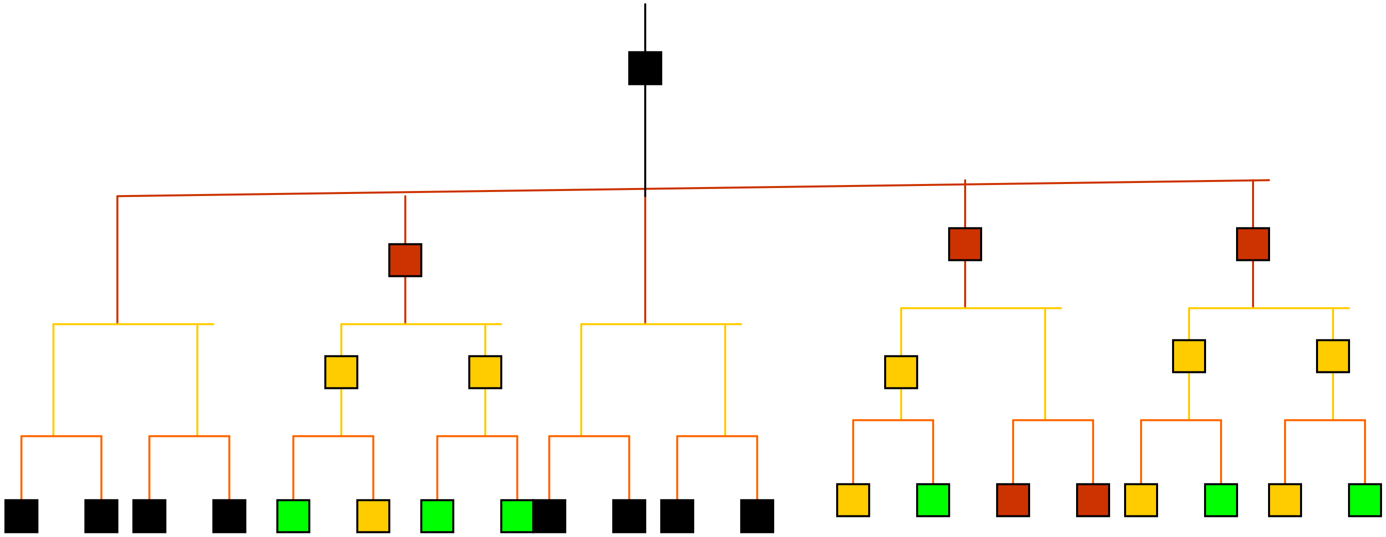
Hierarchical clustering



Hierarchical clustering



Hierarchical clustering



Gray-scale image operators

- **Gray-scale image :** $f : E \to K$ $\quad f \in K^E$

  $K = \{0, 1, \ldots, 255\}$

- **Gray-scale image operator :** $\Psi : K^E \to K^E$

- **Characteristic function :** $\psi : K^W \to K$

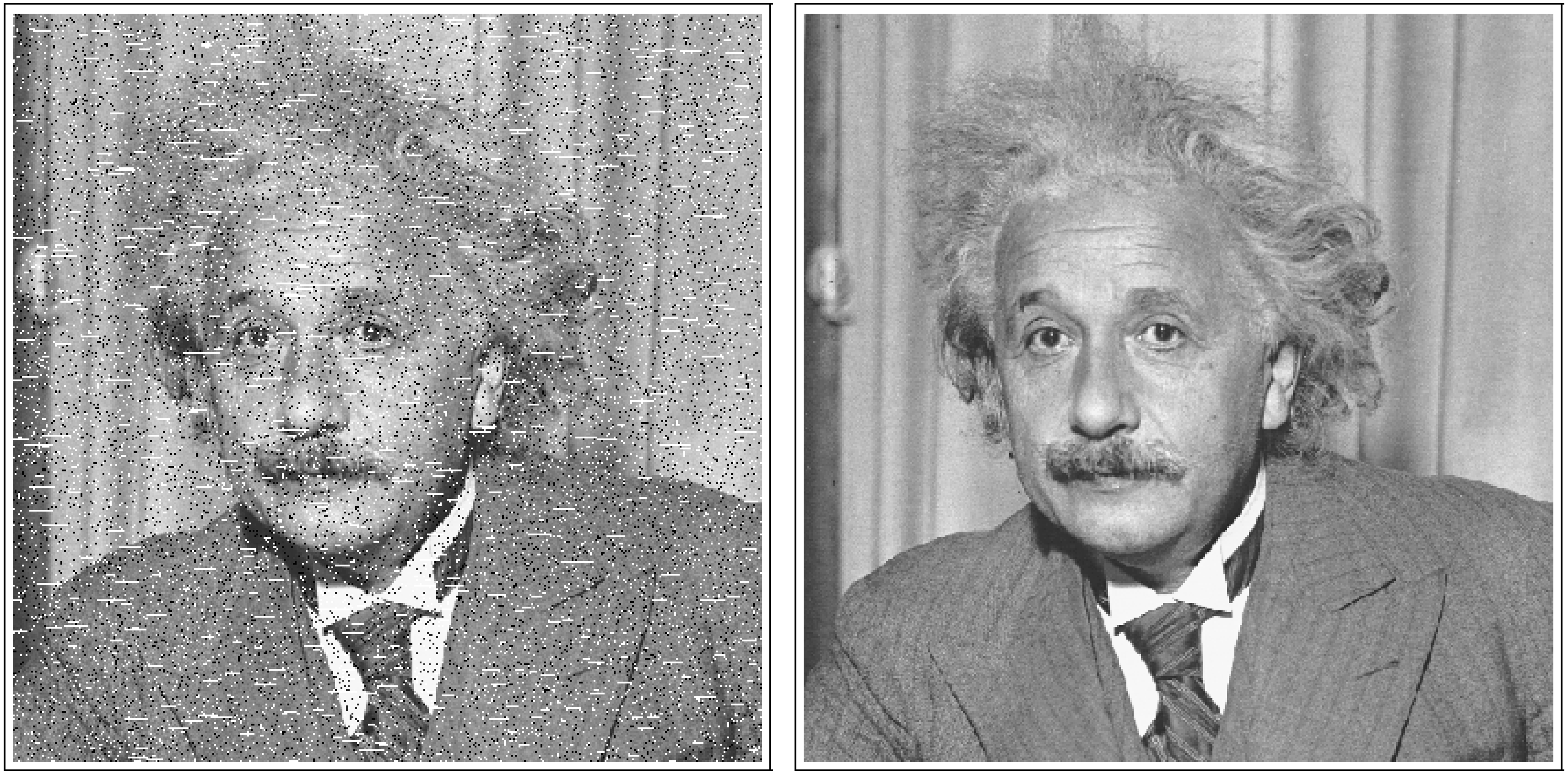
- **Design of gray-scale *W*-operators**

Same design procedure could be applied

$\Downarrow$

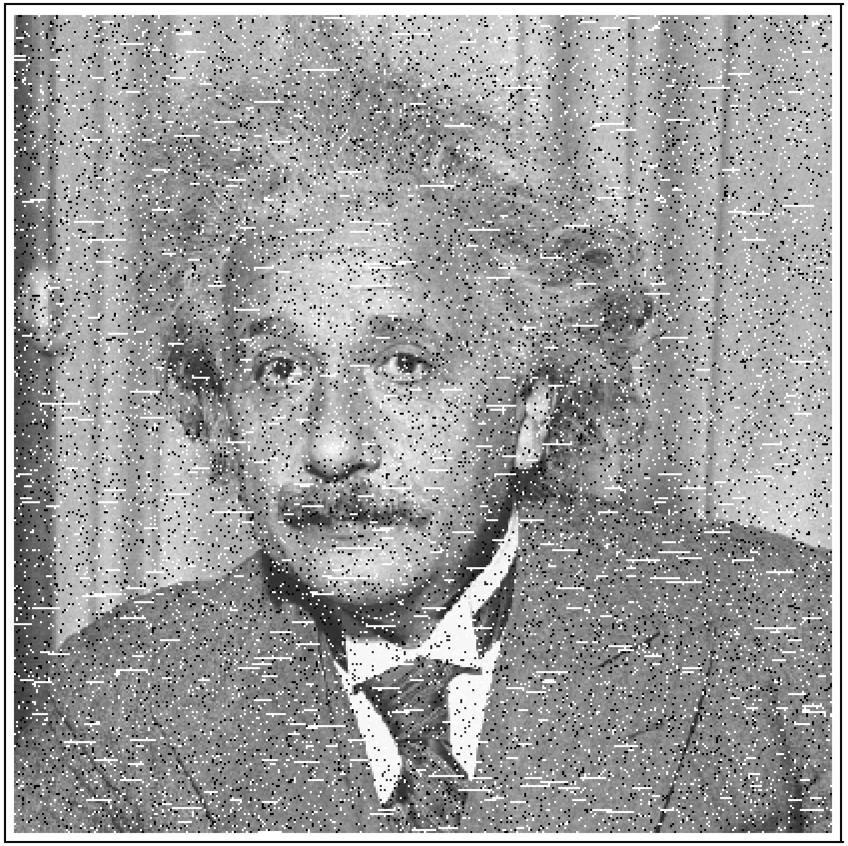
Computationally much more hard !!

Impulse noise removal (1)

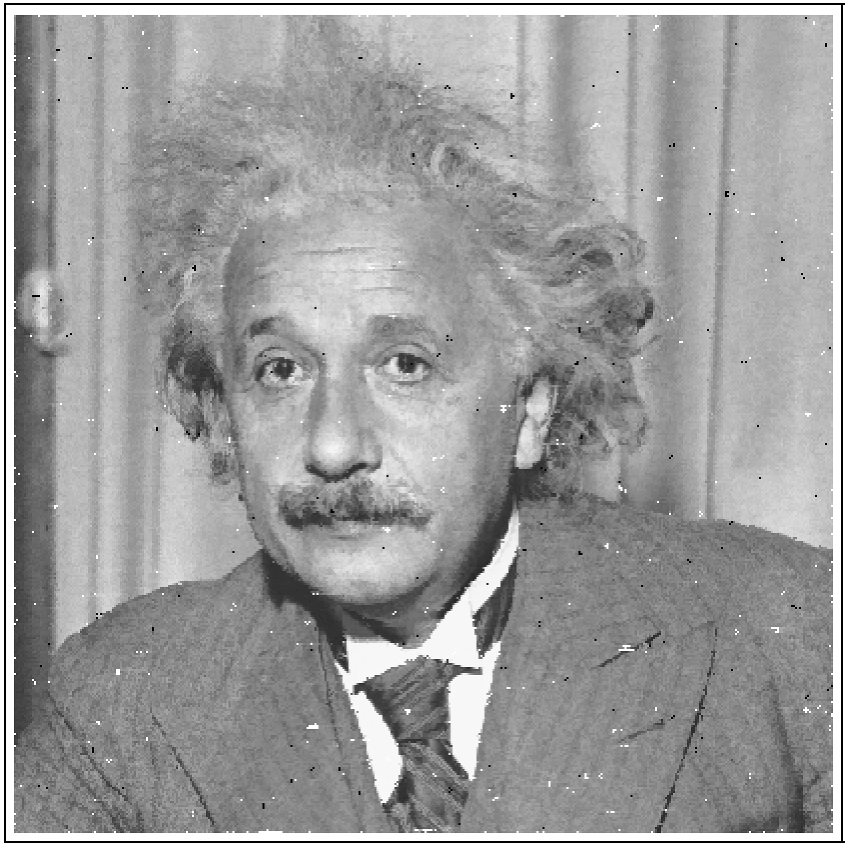


training images

Impulse noise removal (2)

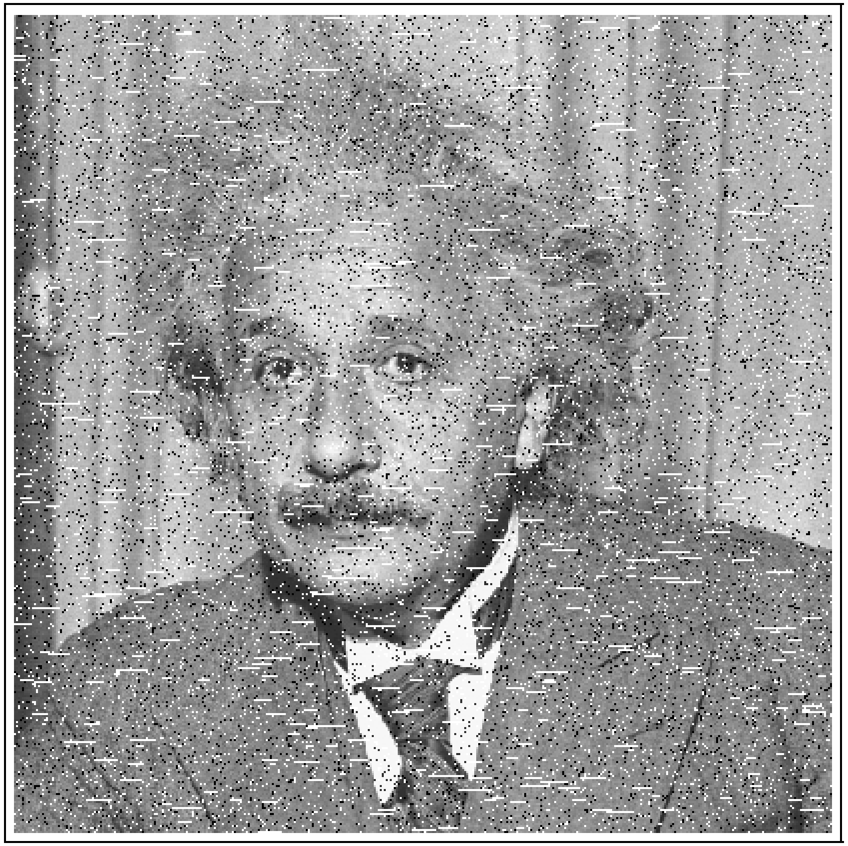


test image

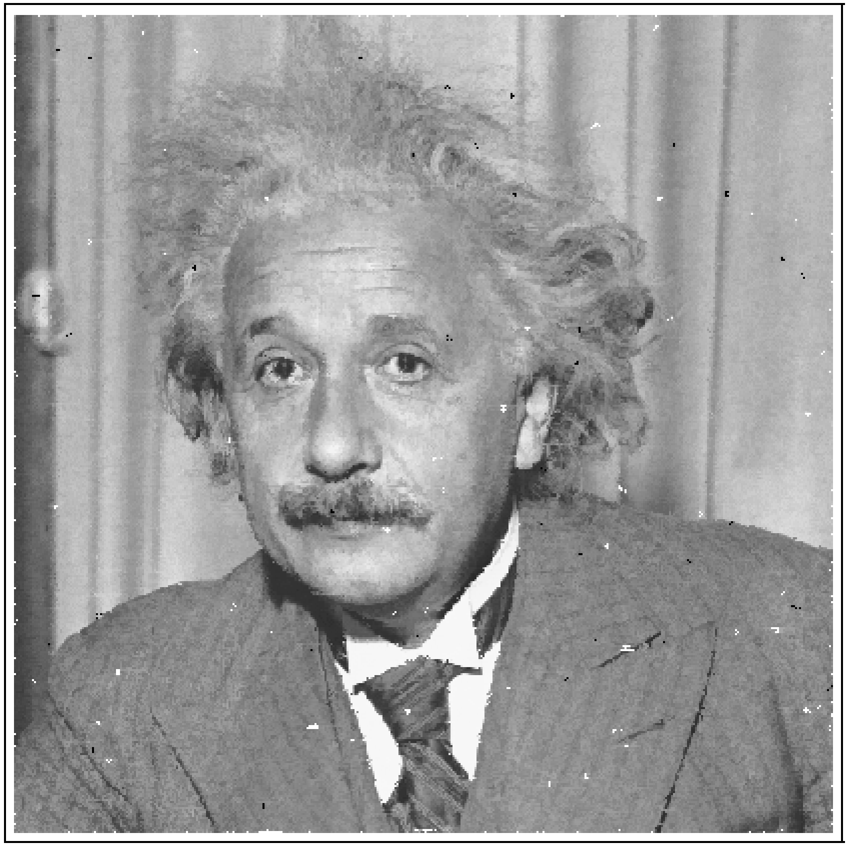


iteration 1

Impulse noise removal (3)

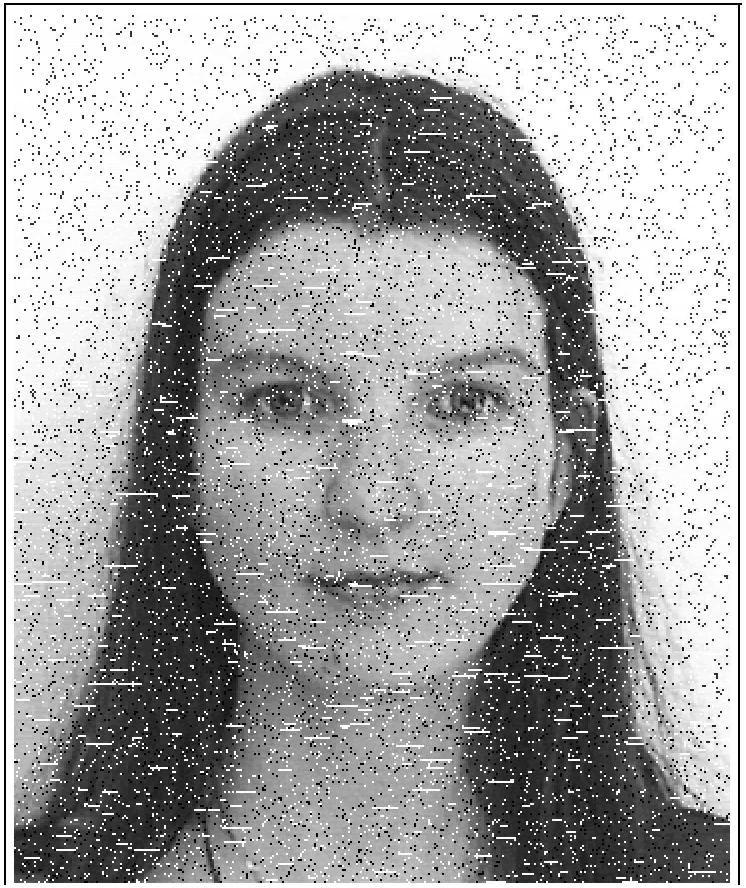


test image



iteration 5

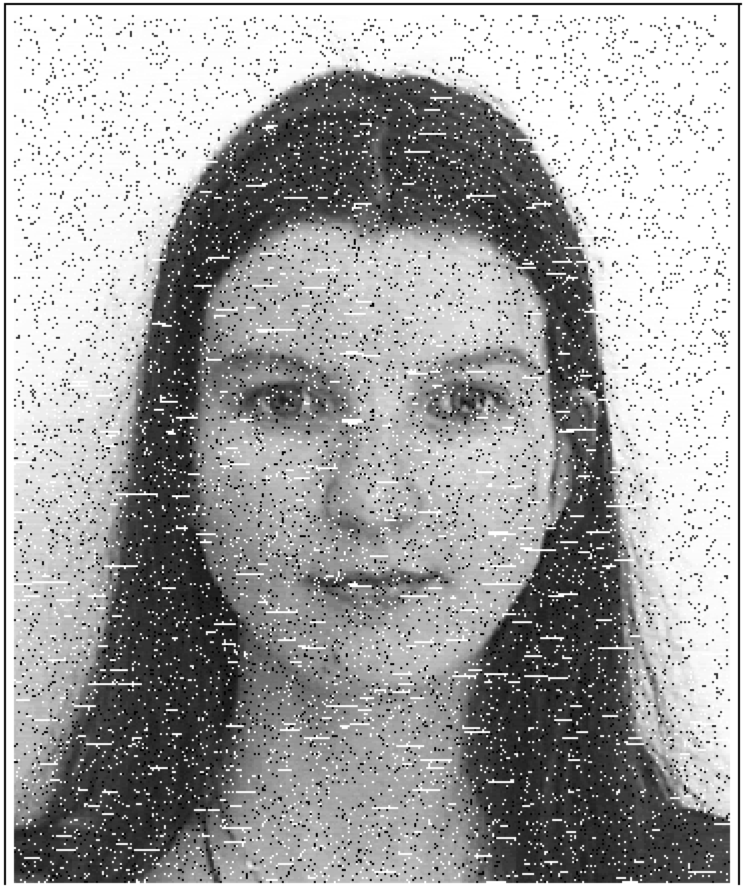
Robustness (1)



test image

iteration 1

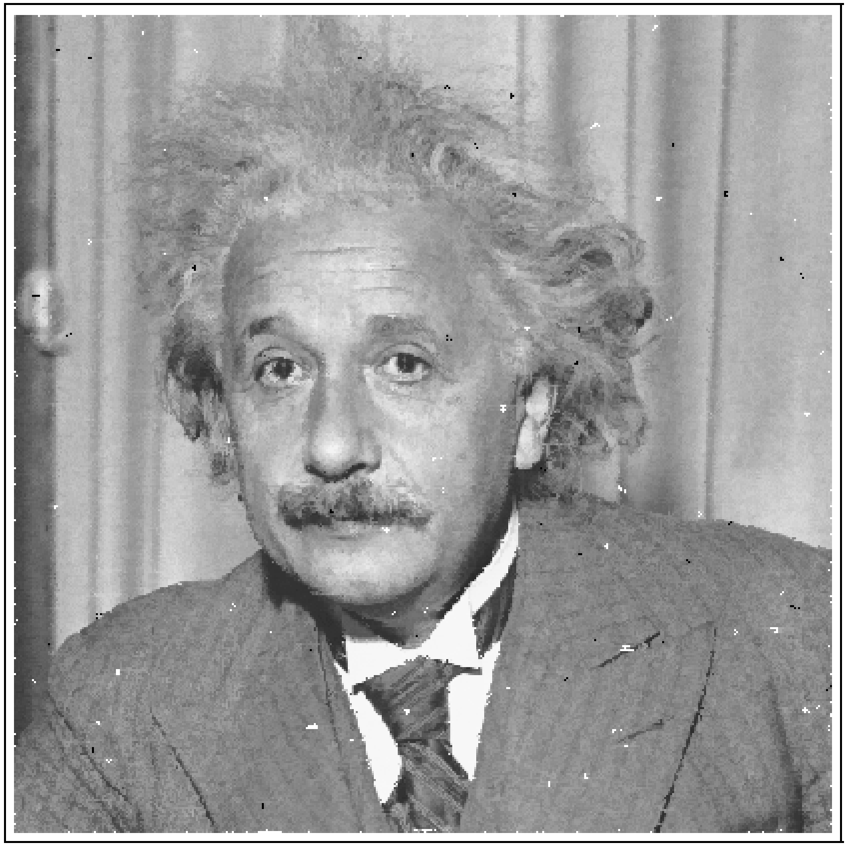
Robustness (2)



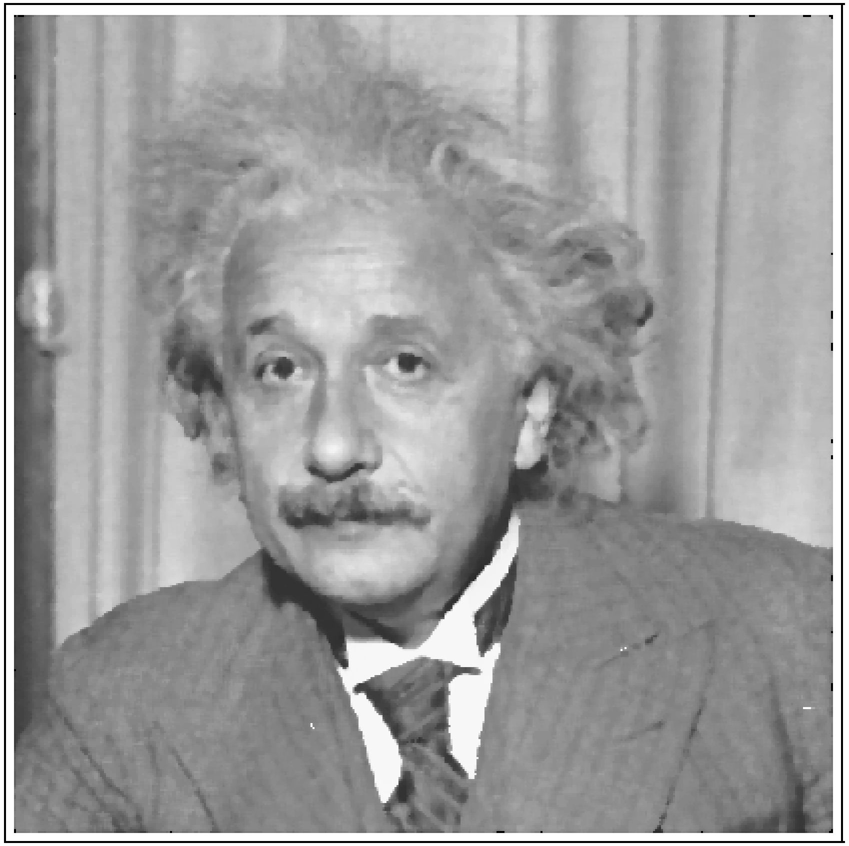
test image

iteration 5

Stack filter x median (1)



iteration 5



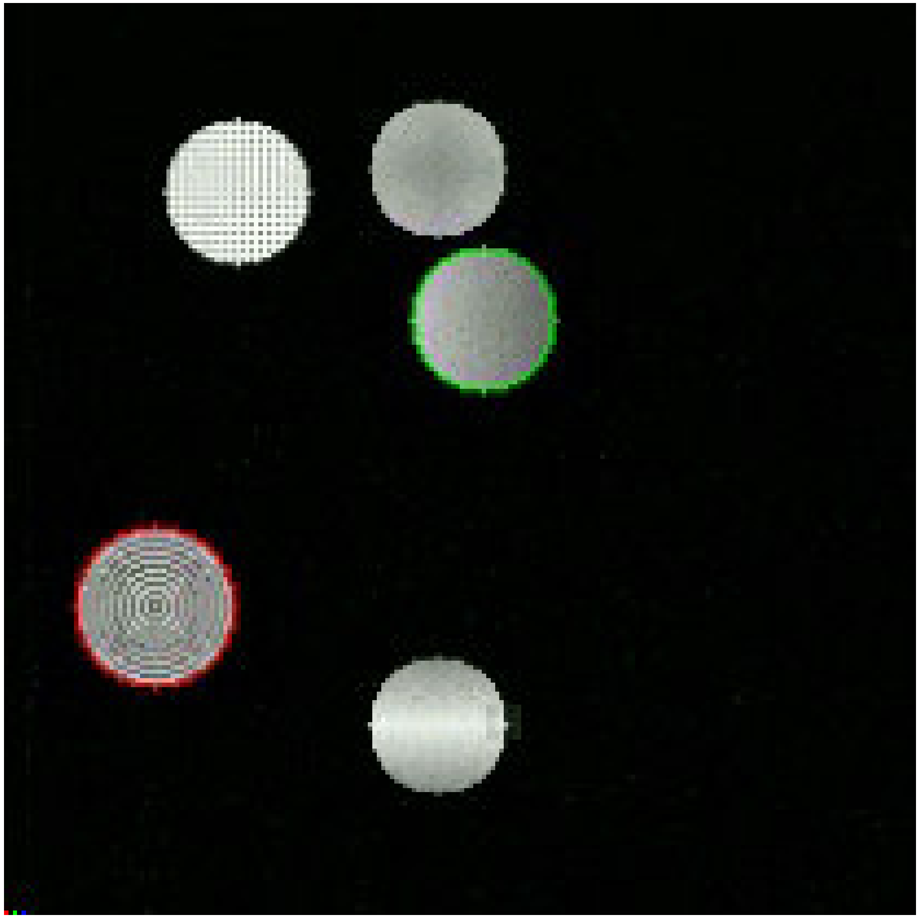
Median 5x5

Stack filter x median (2)

iteration 5

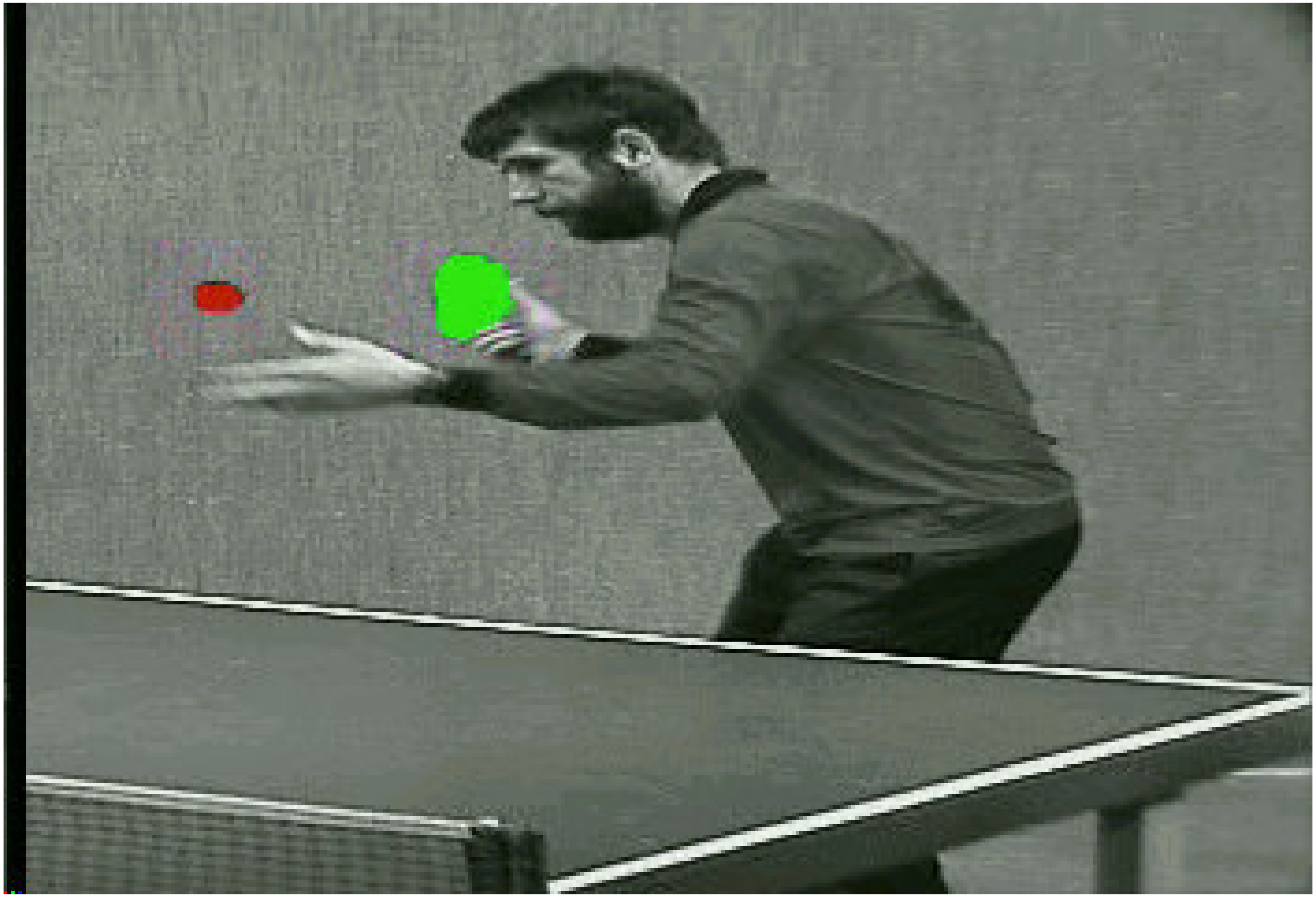
5x5 median

Motion Tracking



Motion Tracking

Motion Tracking



Conclusion

- A powerful tool to solve practical problems
- Hard problems requires modeling of prior knowledge
- Prior knowledge modeling implies complex problems in Statistics, Algebra and Combinatory