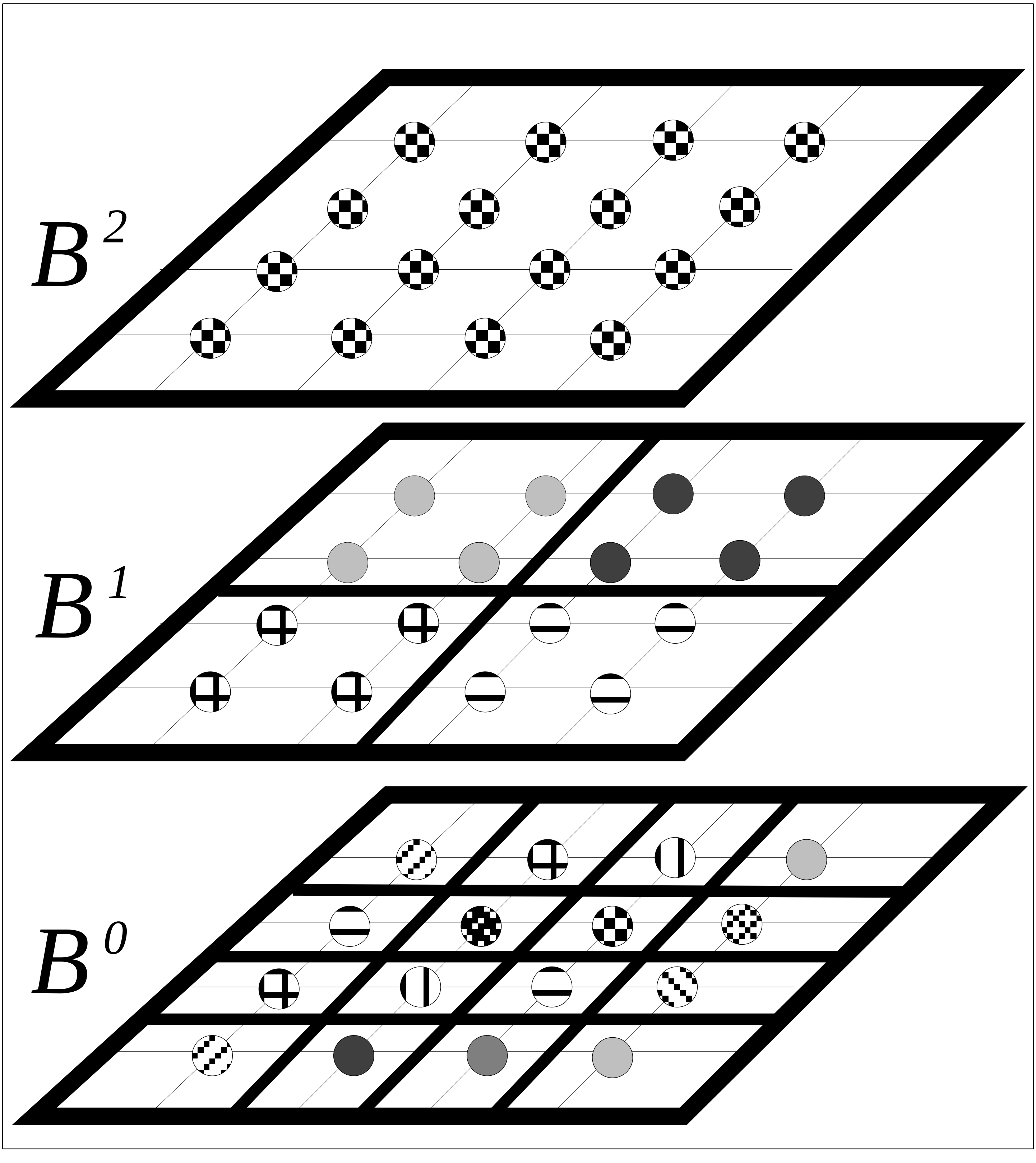
Image Segmentation and Parameter Estimation in a Markovian Framework

Zoltan Kato

Multiscale Model



Multiscale Model

To generate the multiscale version of the monogrid model:

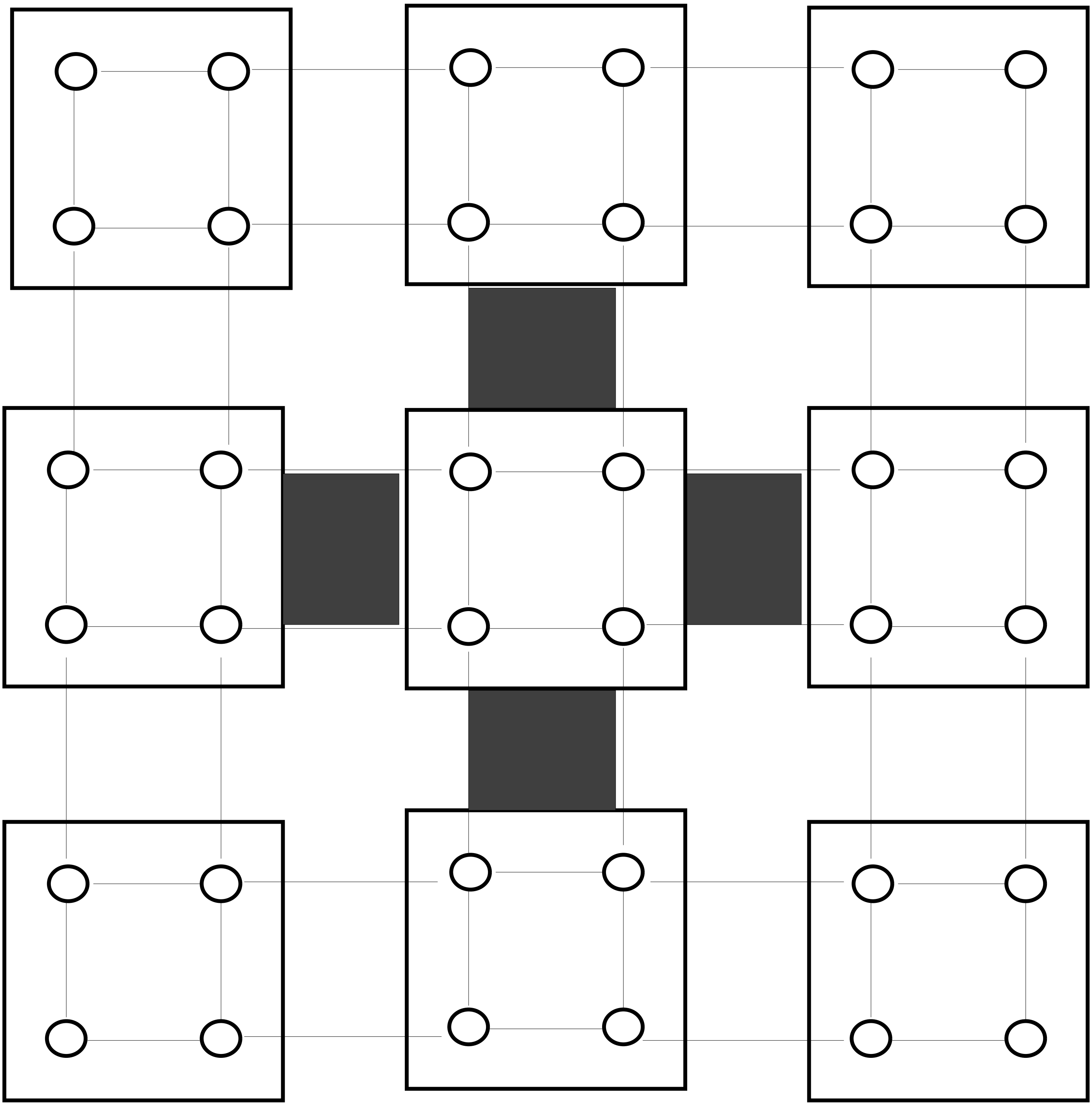
- divide the initial grid into blocks of size $n \times n$ (2×2 here)

- associate the same label to the pixels of a block

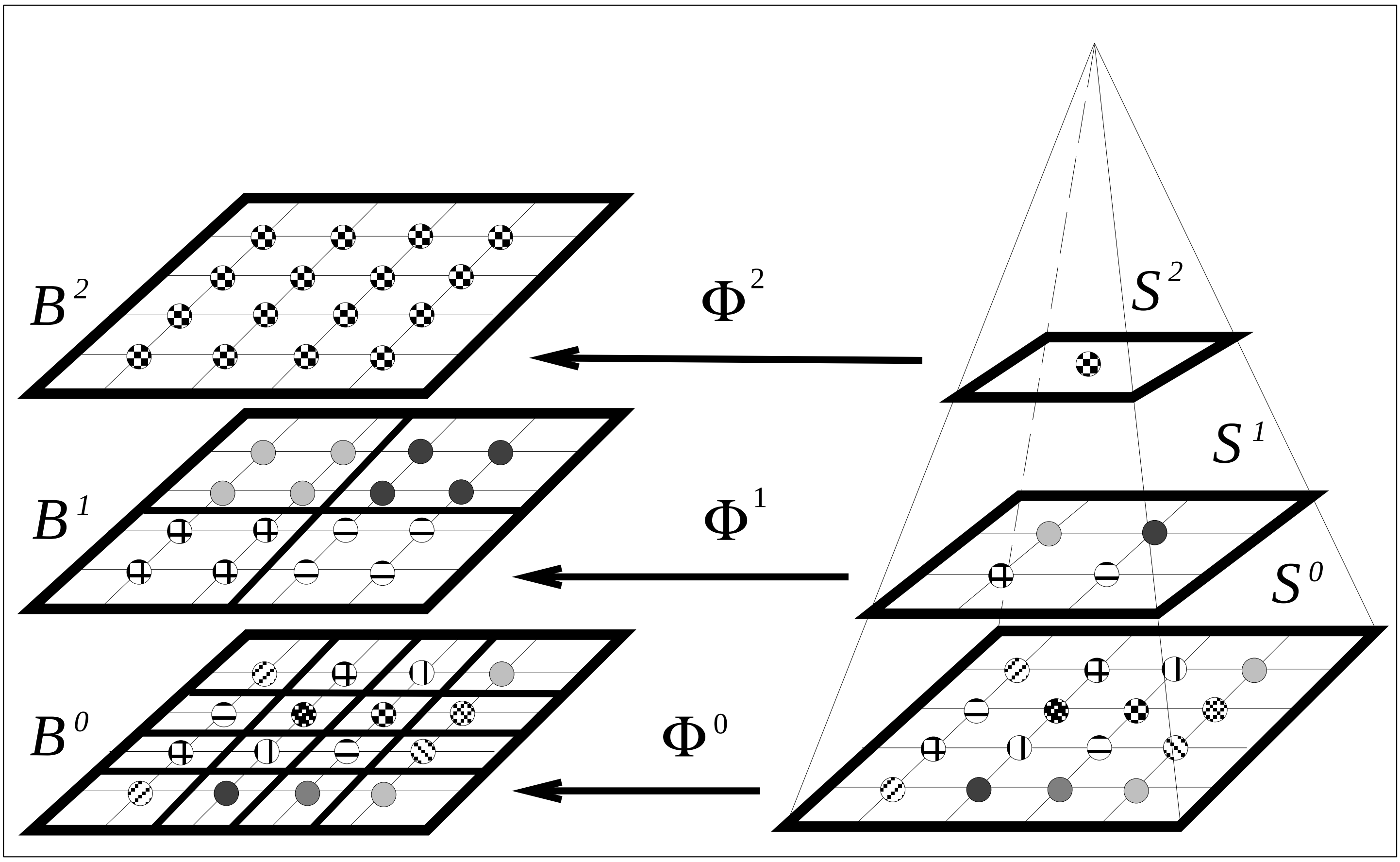
- coarser scales are defined similarly with blocks $n^i \times n^i$ ($i = 2, 3, \ldots$)

Neighborhood structure on blocks

We have the same neighborhood structure as on the initial grid:



Equivalent Model

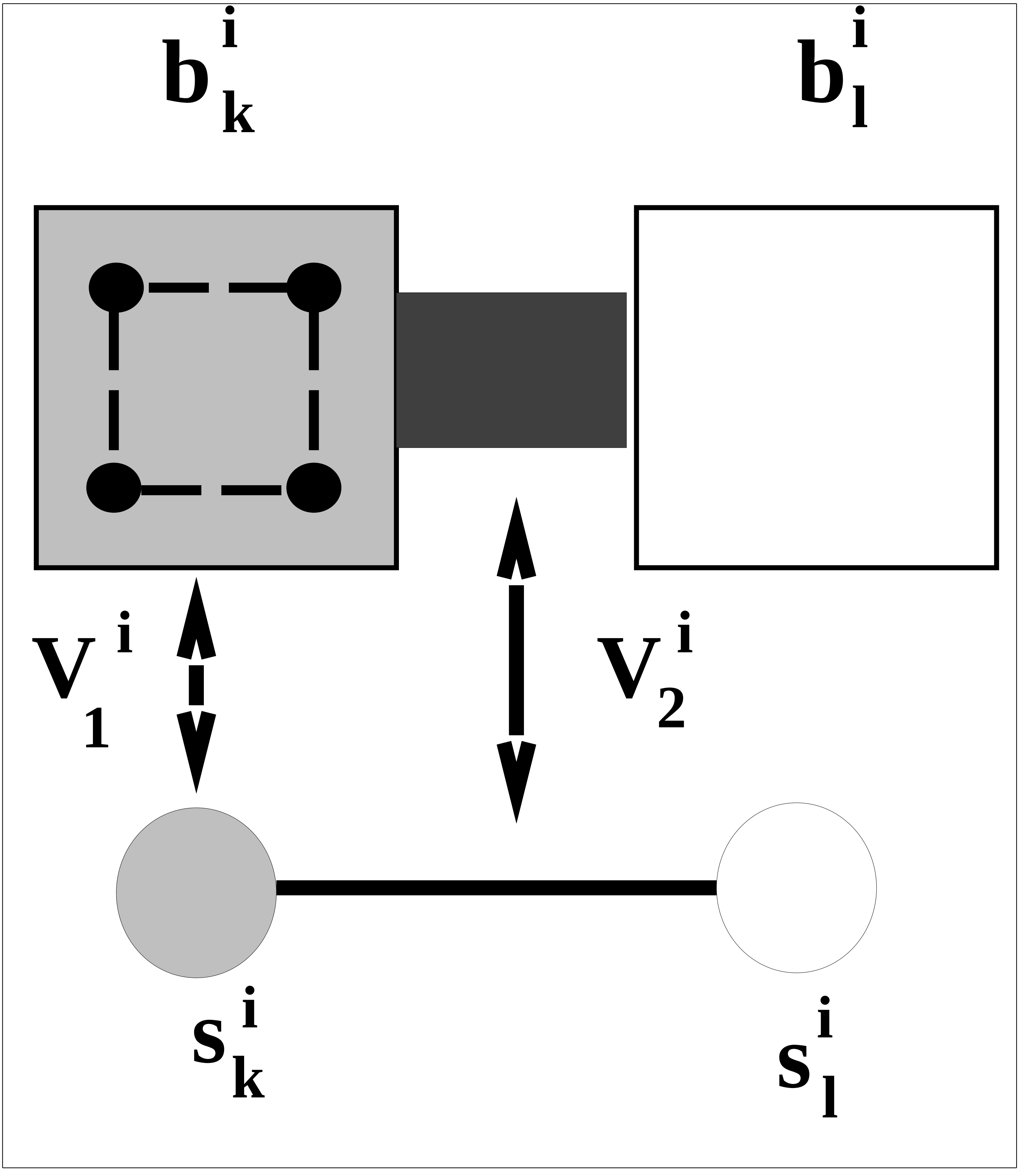


- To each block, we associate a unique site having the common label of the corresponding block.

- These sites form a coarse grid isomorphic to the corresponding scale

- The isomorphism is just a projection of the coarse label field to the fine grid.

Energy function on coarse grids



Energy function on coarse grids

- The first order clique-potentials V_1^i are the sum of the potentials of cliques included in the corresponding block:

$$V_1^i(\omega^i_{s^i}, \mathcal{F}) = \sum_{s \in b^i_{s^i}} V_1(\omega_s, f_s) - p^i \beta$$

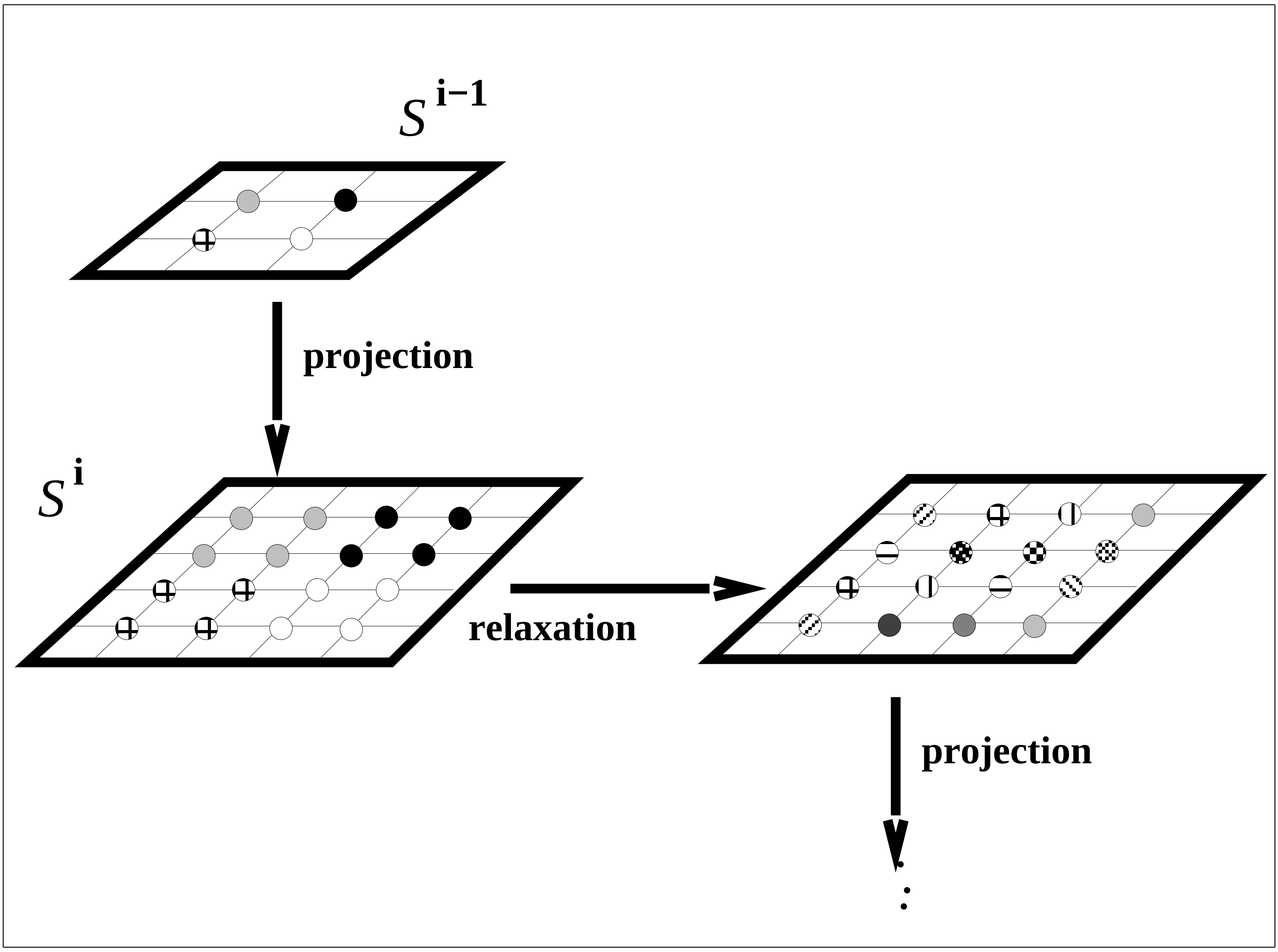
- The second order potentials V_2^i are the sum of potentials siting astride the two corresponding blocks:

$$V_2^i(\omega_r, \omega_s) = \begin{cases} -q^i\beta & \text{if } \omega_r = \omega_s \\ +q^i\beta & \text{if } \omega_r \neq \omega_s \end{cases}$$

p^i is the number of second order cliques included in a block and q^i is the number of cliques between two neighbor blocks

$$\begin{aligned} p^i &= 2n^i(n^i - 1) \\ q^i &= n^i \end{aligned}$$

Multiscale Relaxation Scheme



Using a top-down startegy in the pyramid:

1. The problem is solved at a higher level

2. The lower level is initialized by the projection of the resulting labeling.

Advantages:

+ At coarser grids, the state space has only a few elements $\Longrightarrow$ fast convergence properties on a sequential machine.

+ For deterministic relaxation methods (ICM. . .), the final result is improved.

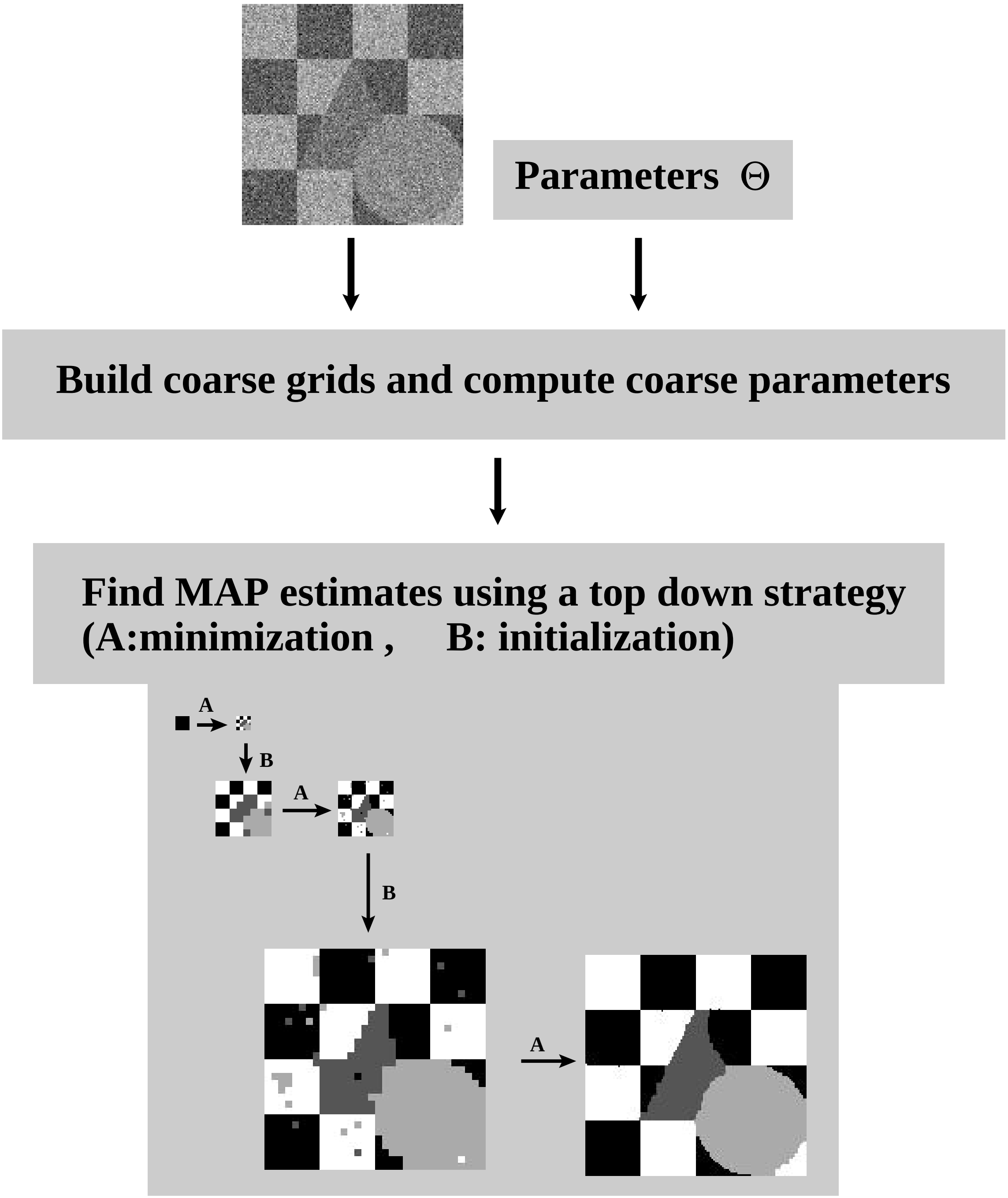
Drawbacks:

– The multiscale scheme demands usually more iterations than the monogrid version

– On a SIMD machine (CM200 for ex.), the multiscale scheme may be slower than the monogrid one (the rapidity depends only on the Virtual Processor Ratio)

– For the stochastic relaxation methods, the final result is only slightly improved since they are independent on the initial configuration.

Supervised Multiscale Segmentation

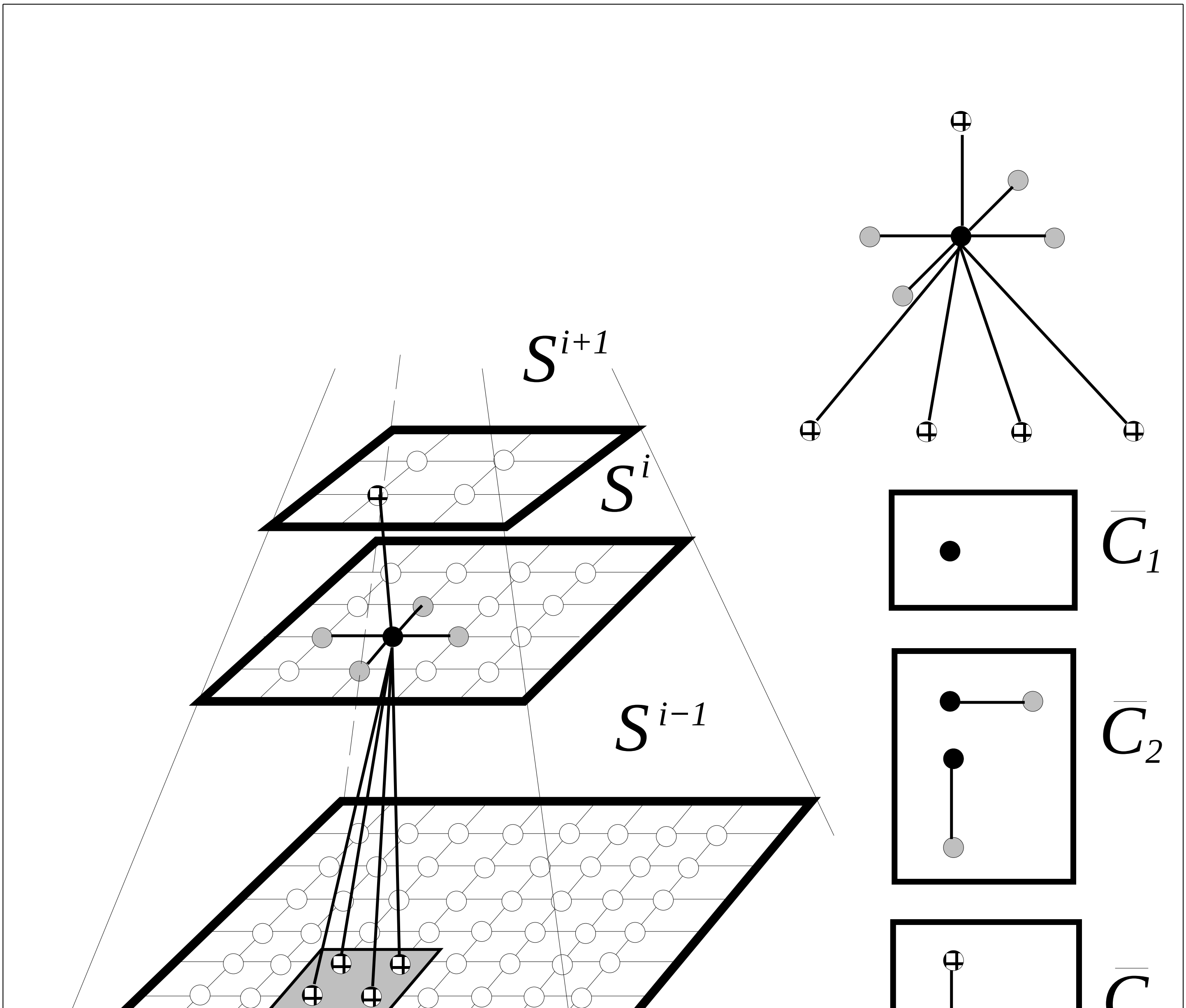


Parameters Θ

Build coarse grids and compute coarse parameters

Find MAP estimates using a top down strategy
(A:minimization , B: initialization)

Hierarchical Model



New neighborhood system

Keeping the interactions at each level, we introduce a new interaction between two neighbor grids:

- Each site interacts with its ancestor and its descendants.

- We consider only the first and second order cliques.

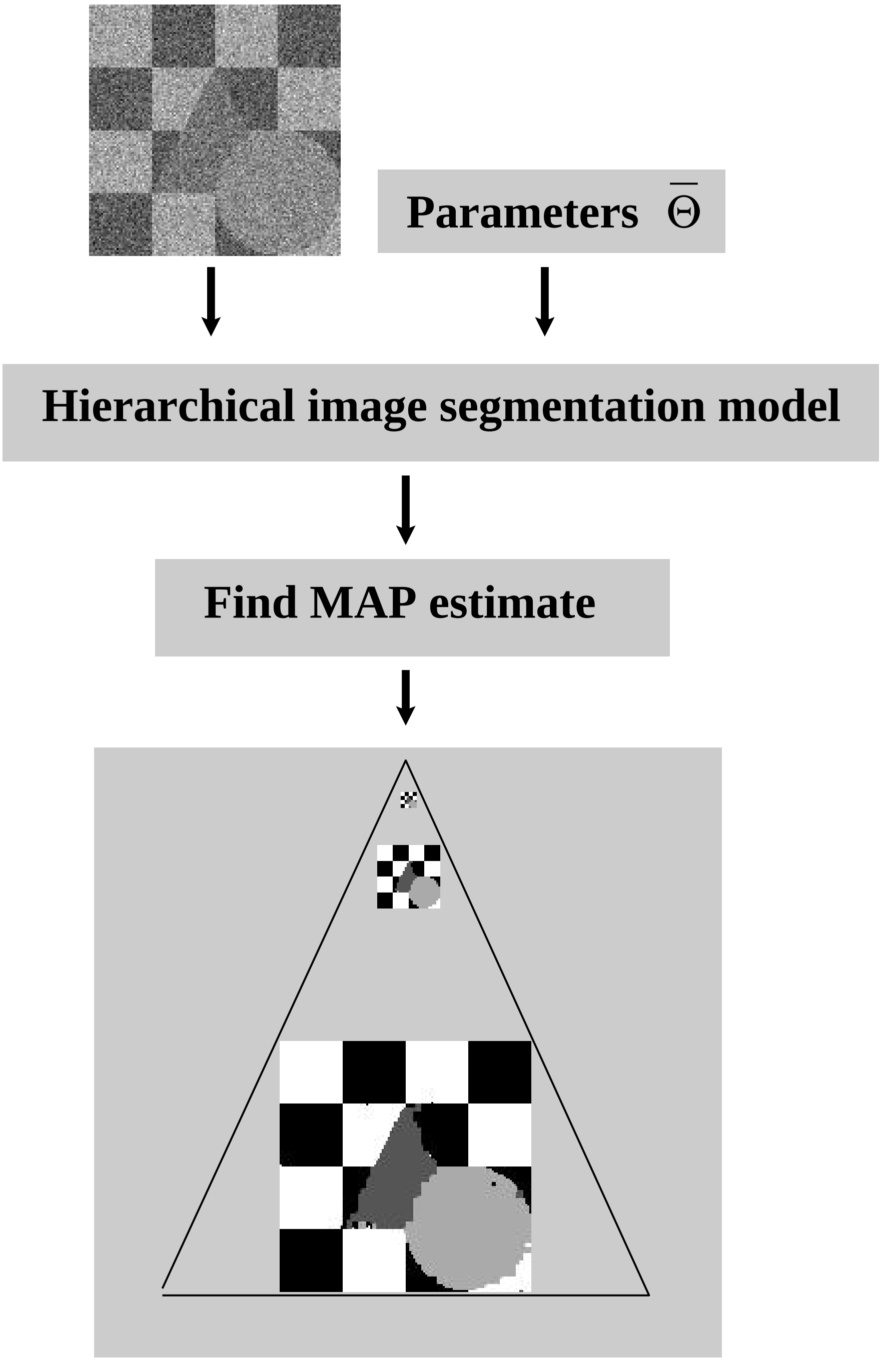
Energy function on the pyramid

- For the cliques which are located on the same level, the potential is not changed.

- For the cliques which site astride two neighbor levels, we will favour similair classes:

$$V_{\{s,r\}}(\omega_s,\omega_r) = \begin{cases} -\gamma & \text{if } \omega_s = \omega_r \\ +\gamma & \text{if } \omega_s \neq \omega_r \end{cases} \quad \gamma > 0$$

Supervised Hierarchical Segmentation

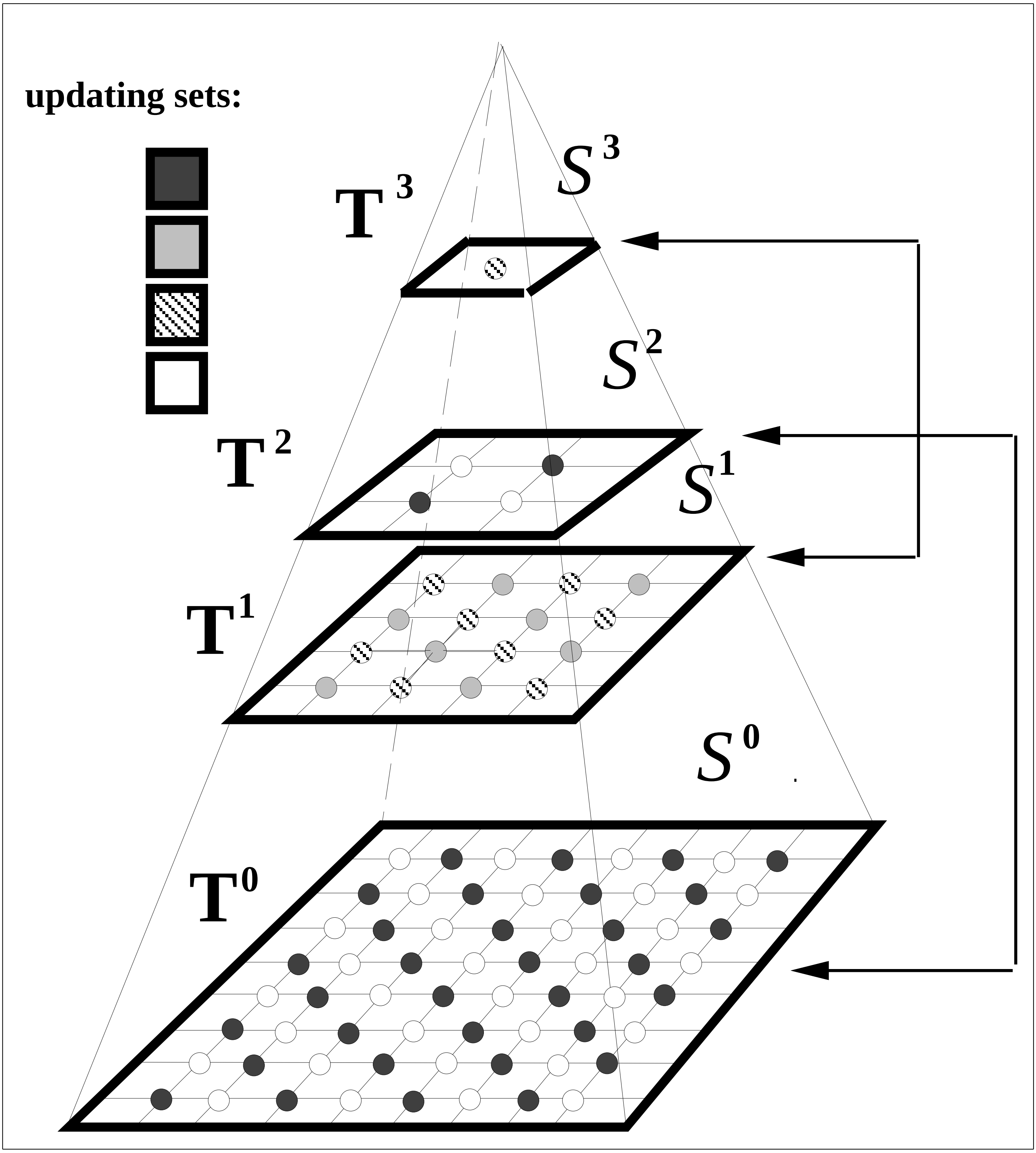


Parameters $\bar{\Theta}$

Hierarchical image segmentation model

Find MAP estimate

Multi-Temperature Annealing



$$T^3 > T^2 > T^1 > T^0$$

Why Shall We Use Multi-Temperature Annealing?

- MTA converges about **two times faster** than classical annealing schemes.

- Convergence is proved towards **global minima**.

Behaviour of MTA:

- **At high temperature:** The energy landscape will be explored with large moves.

- **At intermediate temperatures:** The selected energy-valley will be investigated.

- **At temperature close to** 0**:** Finding the minimum of the energy-valley.

Convergence Conditions

The convergence is proved in the most general case where each clique has its own temperature schedule $T(k,C)$, decreasing in k. **Convergence** towards **global minima** is guaranteed if:

1. For all clique C:

$$\lim_{k\to\infty} T(k,C) = 0$$

2. For all k: $T_k^{inf} \geq \frac{K}{\ln(k)}$
   $(\forall C \colon T_k^{inf} \leq T(k,C) \leq T_k^{sup})$.

3. For all k:

$$\frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} \leq R$$

where K and R are constants depending on the energy function.

How to Implement MTA to Satisfy these Conditions?

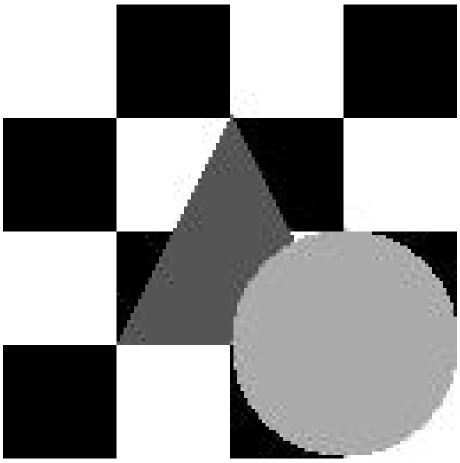
- **Conditions 1 and 2:** The same implementation may be used as in the case of classical annealing: An **exponential schedule** with a sufficiently **high initial temperature**.

- **Condition 3:** Since R cannot be computed in practice, we propose two possible solutions:
  - **Ad hoc way:** Choose a sufficiently small interval $[T_0^{inf}, T_0^{sup}]$.
  - **Using a more strict** but easily verifiable **condition** instead of the 3^d condition:

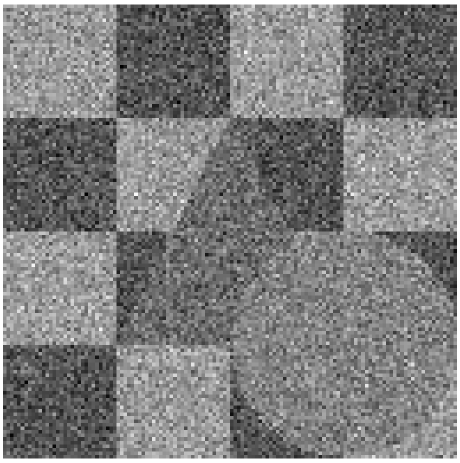
$$\lim_{k\to\infty} \frac{T_k^{sup} - T_k^{inf}}{T_k^{inf}} = 0$$

MTA is a **quite general algorithm**, it does not suppose a pyramidal structure. Optimization of the energy function of hierarchical models is only a possible application.

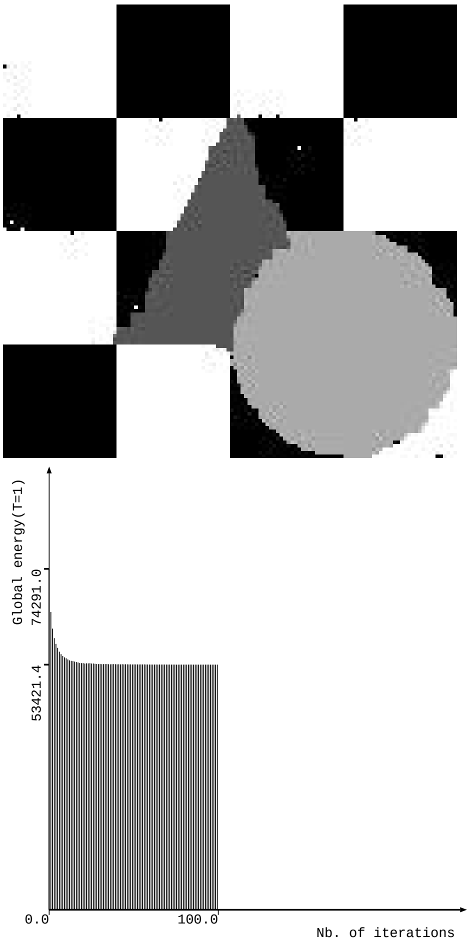
Results on a synthetic image with 4 classes



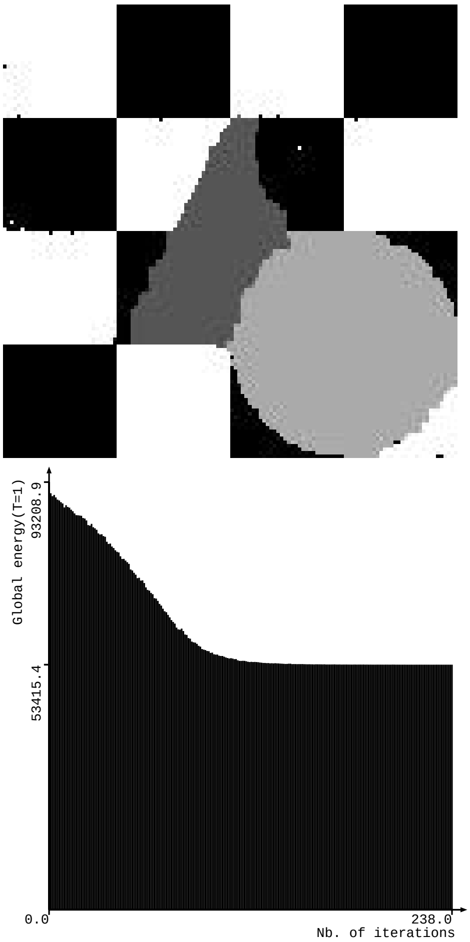
Original image



Noisy image (SNR= $3dB$)

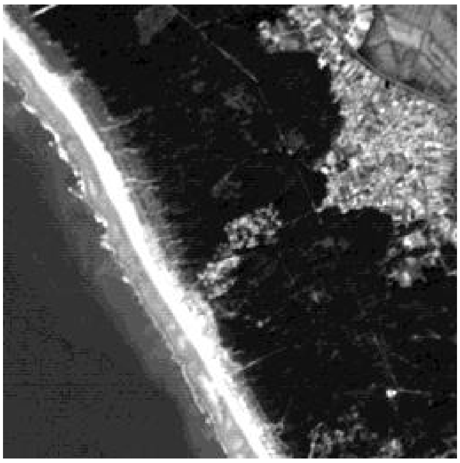


MTA (100 iterations)



Inhomogeneous (238 iterations)

Results on a satellite image with 4 classes

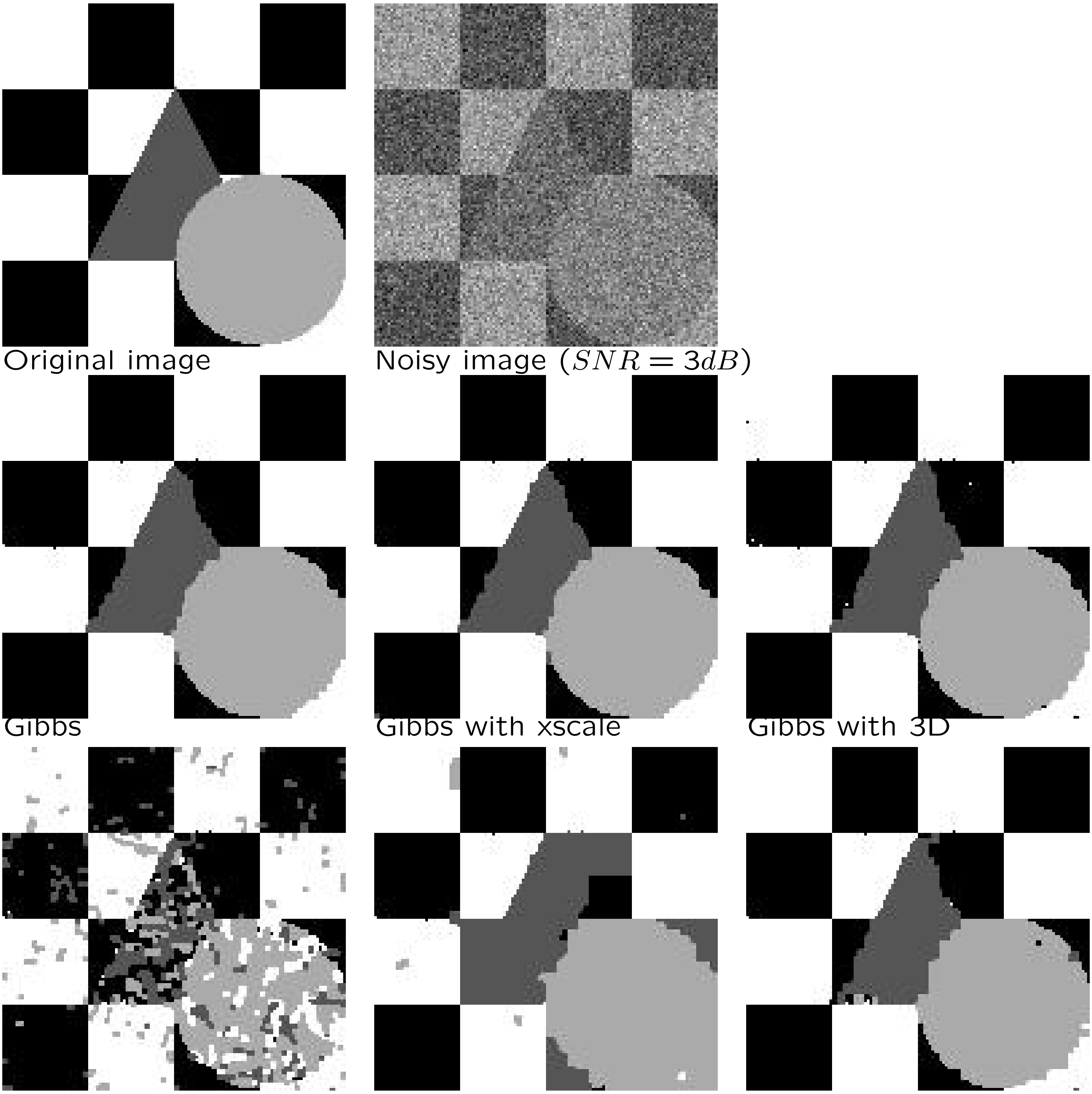


Original image

MTA (44 iterations)
151 sec. CPU time
Final Energy: -12166

Inhomogeneous (118 iterations)
404 sec. CPU time
Final Energy: -12156

Results on a synthetic image with 4 classes



Original image

Noisy image ($SNR = 3dB$)

Gibbs

Gibbs with xscale

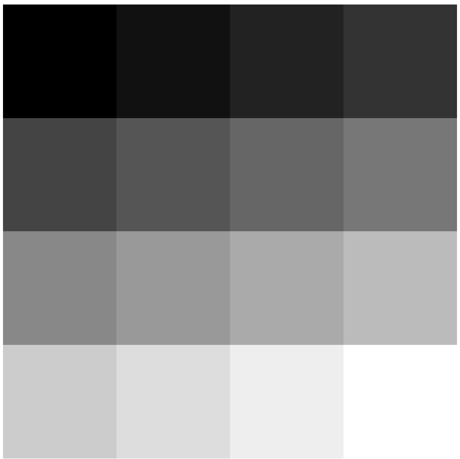
Gibbs with 3D

ICM

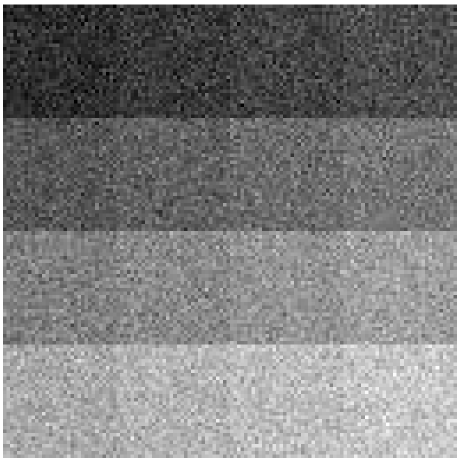
ICM with xscale

ICM with 3D

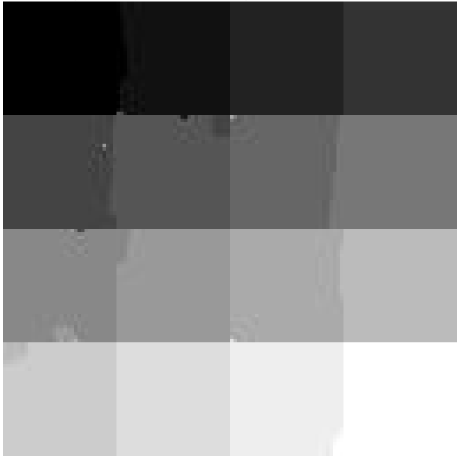
Results on a synthetic image with 16 classes



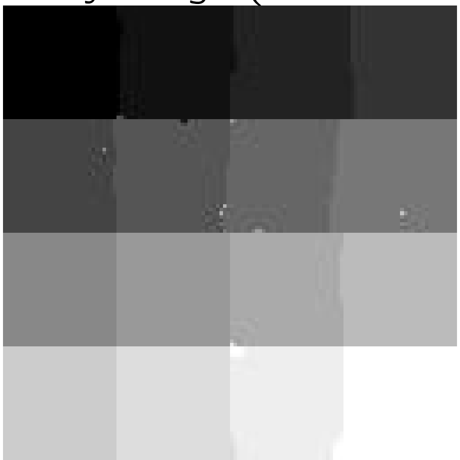
Original image



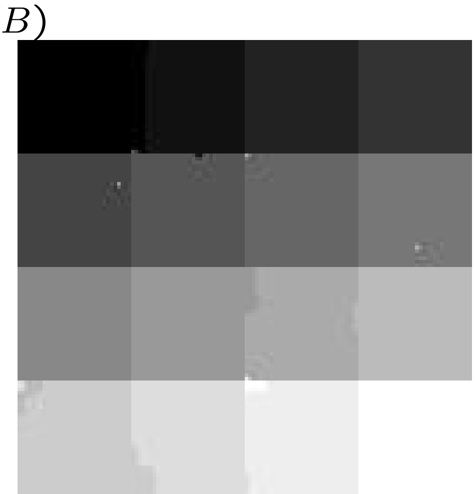
Noisy image ($SNR = 10dB$)



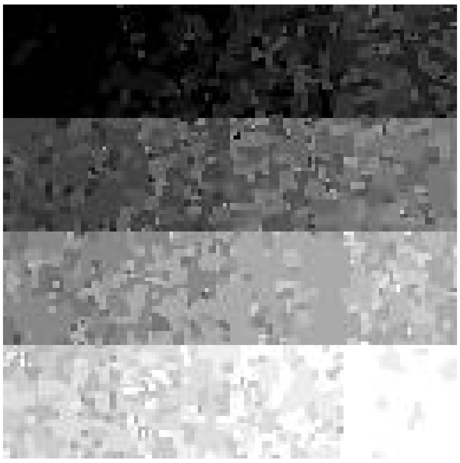
Gibbs



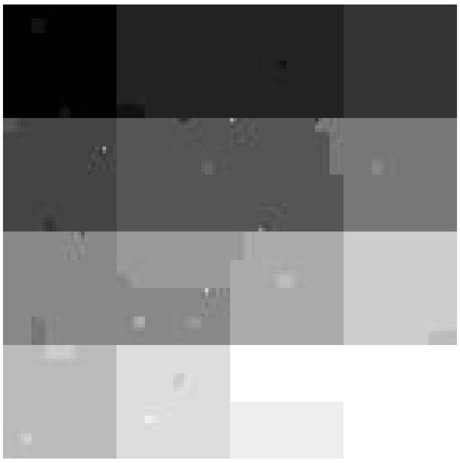
Gibbs with xscale



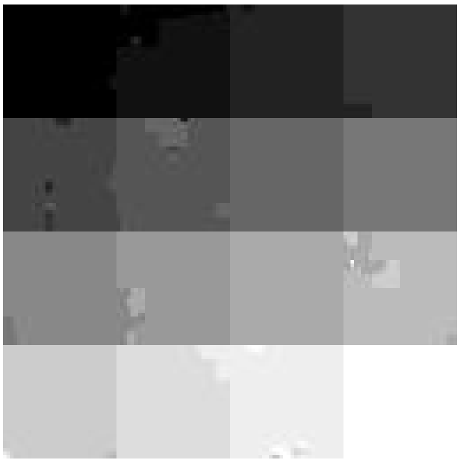
Gibbs with 3D



ICM

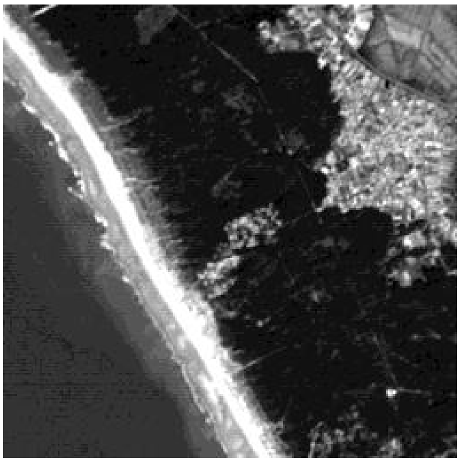


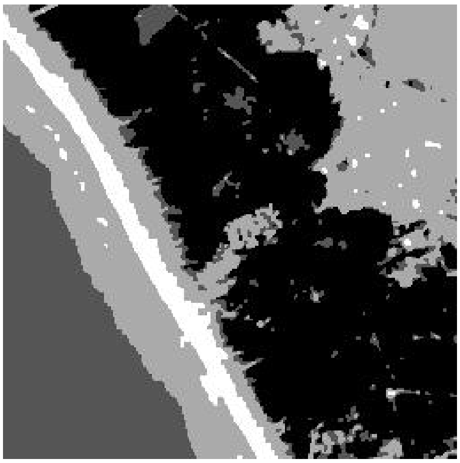
ICM with xscale

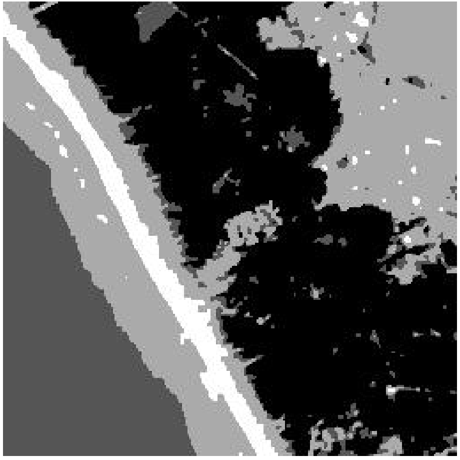


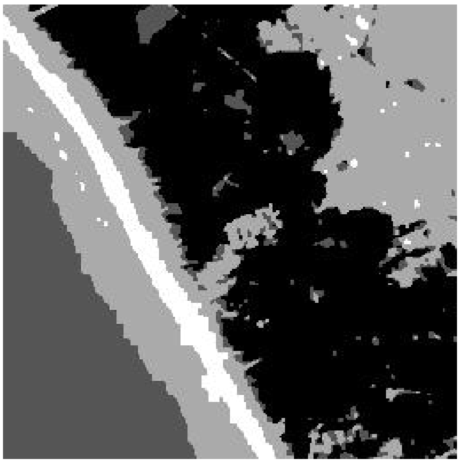
ICM with 3D

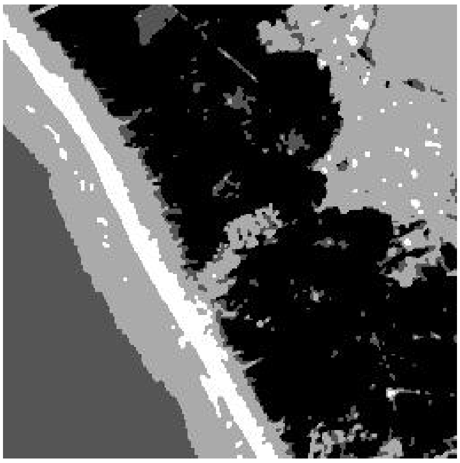
Results on a SPOT image with 4 classes

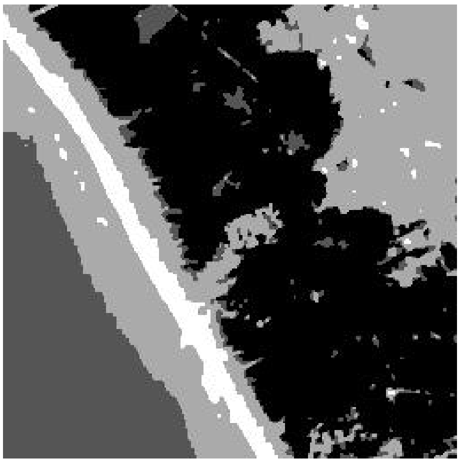

Original image

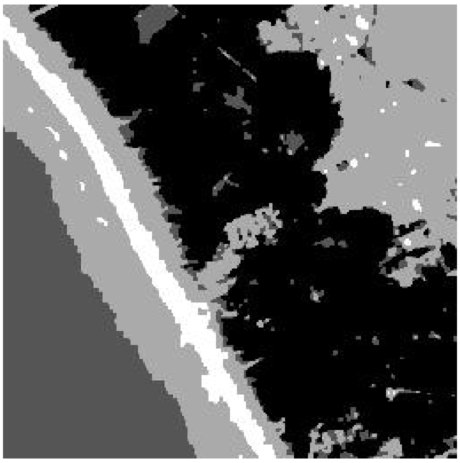

Gibbs


Gibbs with xscale


Gibbs with 3D


ICM

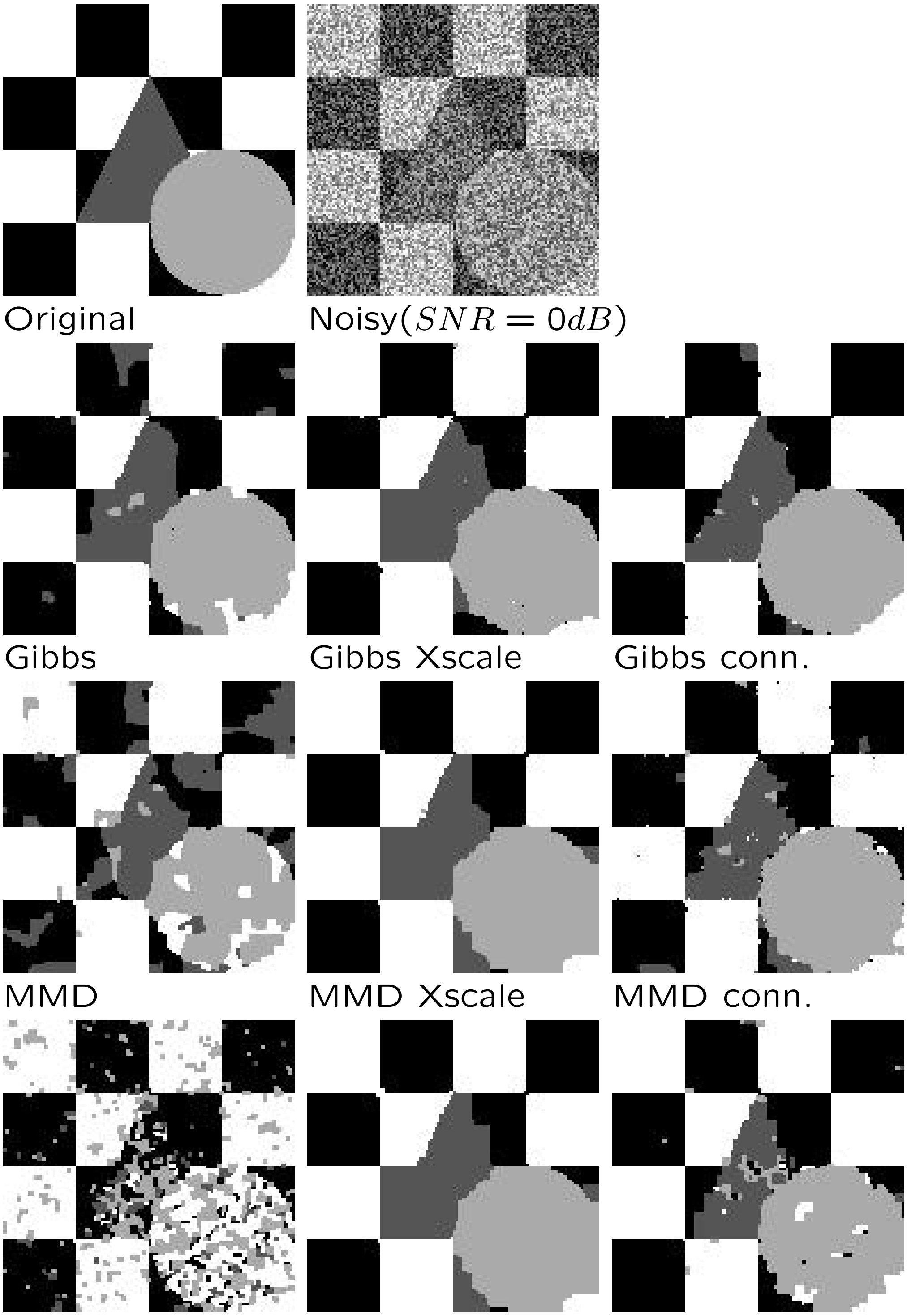

ICM with xscale


ICM with 3D

Computer Time on the CM with 8K processors

original	lev.	VPR	T_0	it.	total(s)	t/it.	β	γ
Gibbs	1	2	4	60	82.16	1.37	0.5	—
ICM	1	2	1	5	0.23	0.046	0.5	—
Xscale	—	—	—	—	—	—	—	—
Gibbs	4	1,2	4	60	63.58	1.06	0.5	—
ICM	4	1,2	1	17	0.79	0.06	0.5	—
conn.	—	—	—	—	—	—	—	—
Gibbs	4	4	4,3,2,1	18	169.37	9.41	0.5	0.2
ICM	4	4	1	5	21.52	4.3	0.5	0.2

Results on a noisy synthetic image



Original

Noisy($SNR = 0dB$)

Gibbs

Gibbs Xscale

Gibbs conn.

MMD

MMD Xscale

MMD conn.

ICM

ICM Xscale

ICM conn.

Computer Time on the CM with 8K processors

orig.	lev.	VPR	T_0	it.	total(s)	t/it.	error	β	γ
Gibbs	1	2	4	53	70.03	1.32	949	0.7	—
ICM	1	2	1	8	0.37	0.05	4686	0.7	—
MMD	1	2	4	83	4.68	0.06	2650	0.7	—
Xscale	—	—	—	—	—	—	—	—	—
Gibbs	4	1,2	4	56	45.84	0.82	611	0.6	—
ICM	4	1,2	1	13	0.65	0.05	863	0.6	—
MMD	4	1,2	4	33	2.0	0.06	851	0.6	—
conn.	—	—	—	—	—	—	—	—	—
Gibbs	4	4	4->1	33	314.38	9.53	358	0.4	0.1
ICM	4	4	1	23	99.6	4.33	759	0.6	0.4
MMD	4	4	4->1	41	125.78	3.07	709	0.3	0.2