A Hierarchical Markov Random Field Model and Multi-Temperature Annealing for Parallel Image Classification

Zoltan KATO
Marc BERTHOD
Josiane ZERUBIA

N° 1938—version 2
Août 1993, version revisée Octobre 1994

PROGRAMME 4

Robotique, image et vision

Rapport de recherche

1993
Champs de Markov hiérarchiques et Recuit Multi-Température. Application à la classification d’image par algorithmes parallèles.

Zoltan KATO, Marc BERTHOD and Josiane ZERUBIA

INRIA - 2004 Route des Lucioles - BP 93
06902 Sophia Antipolis Cedex - FRANCE
Tel (33) 93 65 78 37 - Fax (33) 93 65 76 43
email: kato@sophia.inria.fr, berthod@sophia.inria.fr,
zerubia@sophia.inria.fr

Résumé

Dans ce rapport, nous nous intéressons à la classification d’image par algorithmes de relaxation multi-échelle mis en œuvre de façon massivement parallèle. Les techniques multi-grille sont bien connues pour améliorer nettement les taux de convergence ainsi que la qualité des résultats des techniques itératives de relaxation. Tout d’abord, nous présentons un modèle multi-échelle classique qui consiste à travailler sur une pyramide des étiquettes mais à conserver tout le champ d’observation. Le calcul des fonctions de potentiel aux grilles grossières est obtenu très simplement. L’optimisation est d’abord réalisée à une échelle grossière grâce à une algorithme parallèle de relaxation, puis le niveau plus fin suivant est initialisé par la projection du résultat obtenu à l’échelle plus grossière. Dans un deuxième temps, nous proposons un modèle Markovien hiérarchique construit à partir du modèle précédent. Nous introduisons des nouvelles interactions entre les niveaux voisins de la pyramide. Ceci permet de travailler avec des cliques dont les sites sont assez éloignés à un coût raisonnable. Ce modèle conduit à un algorithme de relaxation utilisant un nouveau type de recuit: le Recuit Multi-Température. Il s’agit d’associer de hautes températures aux niveaux les plus grossiers, étant ainsi moins sensibles aux minima locaux. Nous avons prouvé la convergence de cet algorithme vers un optimum global en généralisant le théorème de Geman et Geman.

Mots Clés
champs de Markov, multi-échelle, modèle hiérarchique, algorithmes de relaxation, classification d’image supervisée.

Remerciements

Ce travail a reçu un soutien financier partiel du CNES, de l’AFIRST et du GdR TdSI (DRED/CNRS).
A Hierarchical Markov Random Field Model and Multi-Temperature Annealing for Parallel Image Classification

Zoltan KATO, Marc BERTHOD and Josiane ZERUBIA
INRIA - 2004 Route des Lucioles - BP 93
06902 Sophia Antipolis Cedex - FRANCE
Tel (33) 93 65 78 37 - Fax (33) 93 65 76 43
e-mail: kato@sophia.inria.fr, berthod@sophia.inria.fr, zerubia@sophia.inria.fr

Abstract

In this report, we are interested in massively parallel multiscale relaxation algorithms applied to image classification. It is well known that multigrid methods can improve significantly the convergence rate and the quality of the final results of iterative relaxation techniques. First, we present a classical multiscale model which consists of a label pyramid and a whole observation field. The potential functions of coarser grids are derived by simple computations. The optimization problem is first solved at the higher scale by a parallel relaxation algorithm, then the next lower scale is initialized by a projection of the result. Second, we propose a hierarchical Markov Random Field model based on this classical model. We introduce new interactions between neighbor levels in the pyramid. It can also be seen as a way to incorporate cliques with far apart sites for a reasonable price. This model results in a relaxation algorithm with a new annealing scheme: The Multi-Temperature Annealing (MTA) scheme, which consists of associating higher temperatures to higher levels, in order to be less sensitive to local minima at coarser grids. The convergence to the global optimum is proved by a generalisation of the annealing theorem of Geman and Geman.

Key Words

Markov Random Fields, multiscale, hierarchical model, relaxation algorithms, supervised image classification.

Acknowledgments

This work has been partially funded by CNES (French Aerospace Agency), AFIRST (French-Israeli Collaboration) and GdR TdSI (DRED/CNRS).
# Contents

1 Introduction .............................................. 1

2 Markov Random Fields .................................... 2
   2.1 Neighborhood Systems ................................. 2
   2.2 Gibbs Distribution and MRF’s ....................... 3
   2.3 A General Markov Image Model ...................... 4
   2.4 Relaxation Algorithms ............................... 5

3 The Classical Multiscale Model ......................... 6
   3.1 General Description .................................. 7
   3.2 Relaxation Scheme .................................... 9
   3.3 A Special Case ...................................... 10

4 The Hierarchical Model .................................. 12
   4.1 General Description ................................. 12
   4.2 A Special Case ...................................... 14

5 Multi-Temperature Annealing ............................ 14
   5.1 Parallel Relaxation Scheme ......................... 14
   5.2 Complexity ........................................ 16
   5.3 Convergence Study .................................. 17
      5.3.1 Homogeneous Annealing ......................... 18
      5.3.2 Inhomogeneous Annealing ....................... 19
      5.3.3 Multi-Temperature Annealing .................... 21

6 Application to Supervised Image Classification ......... 22

7 Experimental Results .................................. 23
   7.1 Comparison of the Schedules ....................... 24
   7.2 Comparison of the Models ......................... 24

8 Conclusion ............................................. 26

A Proof of The Multi-Temperature-Annealing Theorem .... 28
   A.1 Notation ............................................. 28
   A.2 Proof of the Theorem ................................ 29
List of Figures

1. First order neighborhood-system with cliques ........................................ 2
2. Updating sets in the case of a first order MRF ........................................ 5
3. Parallelized Markov chains (\(T_0 > T_1 > T_2 > T_3 > T_4\)) ..................... 5
4. The isomorphism \(\Phi^i\) between \(B^i\) and \(S^i\) ...................................... 7
5. The multiscale relaxation scheme ............................................................. 9
6. The two subsets of \(C\) in the case of a first order neighborhood system, a: \(C^i_k\); b: \(C^i_{k,i}\) ................................................................. 10
7. The functions \(\Psi\) and \(\Psi^{-1}\) ............................................................... 12
8. The neighborhood system \(\tilde{G}\) and the cliques \(\tilde{C}_1\tilde{C}_2\) and \(\tilde{C}_3\) .................. 12
9. Relaxation scheme on the pyramid. The levels connected by pointers are updated at the same time ................................................................. 15
10. Memory complexity of the hierarchical model ......................................... 16
11. Communication scheme of the hierarchical model ..................................... 16
12. Energy decrease with the MTA schedule ............................................... 24
13. Energy decrease with the inhomogeneous annealing schedule ..................... 24
14. Results of the Gibbs sampler on a synthetic image with inhomogeneous and MTA schedules ................................................................. 25
15. Histogram of the SPOT image with 6 classes .......................................... 25
16. Results on a synthetic image with 2 classes ........................................... 36
17. Results on a synthetic image with 4 classes ........................................... 37
18. Results on a synthetic image with 16 classes ......................................... 38
19. Results on a SPOT image with 4 classes ................................................. 39
20. Results on an indoor scene with 4 classes ............................................... 40
21. Results on a medical image with 4 classes ............................................. 41
22. Original SPOT image with 6 classes ...................................................... 42
23. Ground truth data ..................................................................................... 43
24. Results of the ICM algorithm. Comparison with ground truth data. ............. 44
25. Results of the Gibbs Sampler. Comparison with ground truth data .............. 45

Zoltan KATO, Marc BERTHOD, Josiane ZERUBIA
List of Tables

1. Results on a noisy synthetic image with 2 classes ........................................ 47
2. Results on a noisy synthetic image with 4 classes ........................................ 47
3. Results on a noisy synthetic image with 16 classes ...................................... 47
4. Results on a SPOT image with 4 classes ...................................................... 48
5. Results on the SPOT image with 6 classes .................................................... 48
6. Original energy function \( U \) ................................................................. 54
7. Modified energy function \( U' \) ................................................................. 54
Notations

\( \mathcal{S} = \{s_1, s_2, \ldots, s_N\} \) .......................... Set of sites or pixels
\( \mathcal{G} = \{\mathcal{G}_s \mid s \in \mathcal{S}\} \) .......................... Neighborhood system over \( \mathcal{S} \)
\( \mathcal{G}_s \) ................................. Neighborhood of \( s \)
\( C \subseteq \mathcal{S} \) ................................. A clique
\( \mathcal{C} \) ................................. Set of cliques
\( C_s \) ................................. Set of cliques containing \( s \)
deg(\( \mathcal{C} \)) ................................. Degree of the cliques
\( L \) ................................. Lattice defined on \( \mathcal{S} \) \((s = (i, j))\)
\( \mathcal{G}^n \) ................................. Homogeneous neighborhood system of order \( n \)
\( \mathcal{X} = \{X_s : s \in \mathcal{S}\} \) .......................... MRF over \( \mathcal{S} \)
\( \Lambda = \{0, 1, \ldots, L \leftrightarrow 1\} \) .......................... Common state space for \( X_s \)
\( \Omega \) ................................. Set of all possible configurations
\( \omega = (\omega_{s_1}, \ldots, \omega_{s_N}) : \omega_{s_i} \in \Lambda, 1 \leq i \leq N \) .......................... Any configuration of \( \mathcal{X} \)
\( \pi(\omega) \) ................................. Gibbs distribution on \( \Omega \)
\( Z \) ................................. Partition function
\( T \) ................................. Temperature
\( U(\omega) \) ................................. Energy function
\( V_C(\omega) \) ................................. Potential of clique \( C \)
\( \mathcal{F} = \{f_s : s \in \mathcal{S}\} \) .......................... Observed data
\( \{n_k, k = 1, 2, \ldots\} \) .......................... Updating order of the sites
\( W = w^n \) ................................. Width of the lattice \( L \)
\( H = h^m \) ................................. Height of the lattice \( L \)
\( M = \inf(n, m) \) .......................... The highest level of the pyramid (we have \( M + 1 \) levels)
\( \mathcal{B}^i = \{b^i_1, \ldots, b^i_{N_i}\} \) .......................... \( i^{th} \) scale
\( b^i_k \) ................................. \( k^{th} \) block at the \( i^{th} \) scale
\( N_i = N/(wh)^i \) .......................... Number of blocks at the \( i^{th} \) scale
\( \omega^i_j \in \Lambda \) .......................... Common label of the block \( b^i_j \)
\( \Omega_i \) ................................. Configuration space at the \( i^{th} \) scale
\( C^i_j \) ................................. A clique of order \( j \) at scale \( i \)
\( \mathcal{C}^i \) ................................. Set of cliques at scale \( i \)
\( \mathcal{D}_{C^i_j} \) ................................. Set of cliques included in the clique \( C^i_j \) at scale \( i \)
Notations

\( A^i_j \) \hspace{1cm} \text{Set of cliques included in any clique of order} \ j \ \text{at scale} \ i

\( V_{C_j}^i \) \hspace{1cm} \text{Potential of the clique} \ C_j \ \text{at scale} \ i

\( S^i \) \hspace{1cm} \text{Grid at the} \ i^{th} \ \text{level of the pyramid}

\( \Xi_i = \{ \xi^i_s : s \in S^i, \xi^i_s \in \Lambda \} \) \hspace{1cm} \text{Configuration space of the} \ i^{th} \ \text{level of the pyramid}

\( \Phi^i \) \hspace{1cm} \text{Isomorphism between} \ S^i \ \text{and} \ B^i

\( U^i(\xi^i) \) \hspace{1cm} \text{Energy function at level} \ i

\( V_{\xi^i}^i(\xi^i) \) \hspace{1cm} \text{Clique-potentials at level} \ i

\( \bar{S} = \{ \bar{s}_1, \ldots, \bar{s}_N \} \) \hspace{1cm} \text{Set of sites of the pyramid}

\( \bar{N} \) \hspace{1cm} \text{Number of sites of the pyramid}

\( \bar{\Omega} \) \hspace{1cm} \text{Configuration space of the pyramid}

\( \bar{\omega} \) \hspace{1cm} \text{A configuration of the pyramid}

\( \Psi \) \hspace{1cm} \text{Projection-function between two neighbor levels}

\( \bar{g} \) \hspace{1cm} \text{Neighborhood system on the pyramid}

\( g_i \) \hspace{1cm} \text{Neighborhood system at level} \ i

\( \bar{C} \) \hspace{1cm} \text{Set of cliques over the pyramid}

\( C^* \) \hspace{1cm} \text{Set of the cliques located between two neighbor levels}

\( \bar{\bar{X}} \) \hspace{1cm} \text{MRF over the pyramid}

\( \bar{U}(\bar{\omega}) \) \hspace{1cm} \text{Energy function on the pyramid}

\( \bar{V}_{C}(\omega) \) \hspace{1cm} \text{Clique-potentials on the pyramid}

\( U^*(\bar{\omega}) \) \hspace{1cm} \text{Energy over the cliques} \ C^*

\( P_{\omega,\eta}(k \Leftrightarrow 1, k) \) \hspace{1cm} \text{Probability of the} \ k^{th} \ \text{transition} \ \omega \to \eta

\( X(k) \ (k = 1, 2, \ldots) \) \hspace{1cm} \text{Markov chain generated by the Simulated Annealing algorithm}

\( P_{\omega,\eta}(T) \) \hspace{1cm} \text{Transition matrix}

\( G_{\omega,\eta}(T) \) \hspace{1cm} \text{Generation matrix}

\( A_{\omega,\eta}(T) \) \hspace{1cm} \text{Acceptance matrix}

\( \Omega_{\text{opt}} \) \hspace{1cm} \text{Set of globally optimal configurations}

\( T(k, C) \) \hspace{1cm} \text{Temperature function depending on the iteration} \ k \ \text{and on the clique} \ C

\( \pi_{T(k, C)}(\omega) \) \hspace{1cm} \text{Gibbs distribution with temperature} \ T(k, C)

\( \otimes \) \hspace{1cm} \text{The operation} \ \sum_{C \in \bar{C}} \frac{V_{C}(\omega)}{T(k, C)}

\( \pi_0 \) \hspace{1cm} \text{Uniform distribution on} \ \Omega_{\text{opt}}

\( U^{\text{sup}} \) \hspace{1cm} \text{Maximum value of the energy function} \ U(\omega)

\( U^{\text{inf}} \) \hspace{1cm} \text{Minimum value of the energy function} \ U(\omega)
Δ .......................... Difference between the maximum and minimum of $U(\omega)$
$T_k^{\inf}$ .......................... Minimum of $T(k, C)$ at the $k^{th}$ iteration
$\mu_\lambda$ .......................... Mean value of class $\lambda \in \Lambda$
$\sigma_\lambda$ .......................... Deviation of class $\lambda \in \Lambda$
$\beta$ .......................... Second order clique-potential at the finest level
$\gamma$ .......................... Second order clique-potential between neighbor levels
$P(k, \omega|l, \eta)$ .............. The same as the transition probability $P(X(k) = \omega|X(l) = \eta)$
$P(k, \omega|l, \mu)$ .......................... The same as $\sum_\eta P(X(k) = \omega|X(l) = \eta)\mu(\eta)$
$P(k, t|l, \mu)$ .......................... The “$\sim$” means any configuration here
$\|\mu \Leftrightarrow \nu\|$ .......................... The $L^1$ norm of two distributions on $\Omega$
$T_k^{\inf}$ .......................... Minimum of $T(k, C)$ at the $k^{th}$ iteration
1 Introduction

Markov Random Fields (MRF) have become more and more popular during the last few years in image processing [11,10,12,14,30]. A good reason for that is that such a modelization is the one which requires the less a priori information on the world model. On the other hand, the local behavior of MRF permits to develop highly parallel algorithms in the resolution of the combinatorial optimization problem associated with such a model.

In this report, we are interested in massively parallel multiscale relaxation algorithms applied to image classification [8,9,16,26,28]. It is well known that multigrid methods can improve significantly the convergence rate and the quality of the final results of iterative relaxation techniques.

There are many approaches in multigrid image segmentation. F. Marques et al. [26] propose a hierarchical Compound Gauss-Markov Random Field model with a label pyramid and an observation pyramid. Bouman [8,9] proposes a multiscale MRF model where each scale is causally dependent on the coarser grid field above it. This model yields to a non-iterative segmentation algorithm and direct methods of parameter estimation. The basis of our approach is a consistent multiscale MRF model proposed by F. Heitz et al. in [16,28] for motion analysis. This model consists of a label pyramid and a whole observation field. The original energy function can be decomposed as a sum of potential functions which are defined on neighbor blocks and only depend on the labels associated with these blocks and on the observation field. Using this decomposition, the parameters of coarser grids can be computed very easily. This model results in a multigrid relaxation scheme which replaces the original optimization problem by a sequence of more tractable problems. Using a top down strategy in the label pyramid, the optimization problem is first solved at a higher level; then the lower grid is initialized with the previous result by a simple projection. This algorithm is very efficient in the case of deterministic relaxation (for instance ICM [4,18]) which gets stuck in a local minimum near the starting configuration. In the case of stochastic relaxation (for instance Simulated Annealing [13,24,27]) which are far less dependent on the initial configuration, the results are only slightly better but the method is still interesting with respect to computer time especially on a sequential machine. After a brief introduction to the theory of Markov Random Fields (Section 2), we give a general description of this model and the relaxation scheme associated with it in the Section 3.

Then, we propose a new hierarchical MRF model defined on the whole label pyramid (Section 4). In this model, we have introduced a new interaction scheme between neighboring levels in the pyramid yielding a better communication between the grids. It can also be seen as a way to incorporate cliques with far apart sites for a reasonable price. This model gives a relaxation algorithm with a new annealing scheme which can be run in parallel on the entire pyramid. The basic idea of this annealing scheme which we propose to call Multi-Temperature Annealing (MTA) is the following: to the higher levels, we associate higher temperatures which enable the algorithm to be less sensitive to local minima. However at a finer resolution, the relaxation is performed at a lower temperature (at the bottom level, it is closed to 0). The complete convergence study of the relaxation algorithm in the case of a homogeneous and inhomogeneous Multi-Temperature Annealing schedule can be found in Section 5. In the multi-temperature case, our annealing theorem is a generalization of the well known theorem of Geman and Geman [13] and the proof can be found in Appendix A.
In Section 6 we apply these models to supervised image classification. Using a first order MRF model to take into account the context and a Gaussian representation of the classes, we define the energy function for the monogrid, multiscale and hierarchical models.

Finally, experiments are shown in Section 7 with the Gibbs sampler [13] and the Iterated Conditional Mode [4,18] using the three models for each algorithm (monogrid, multiscale and hierarchical). These methods have been implemented in parallel on a Connection Machine CM200 [17].

2 Markov Random Fields

First, we briefly give an introduction to the theory of Markov Random Fields (MRF) [19] and then we describe a general image model used in the following sections. Finally, we recall a few classical relaxation algorithms used for the optimization of the cost function of the model.

2.1 Neighborhood Systems

Let $S = \{s_1, s_2, \ldots, s_N\}$ be a set of sites.

**Definition 2.1 (Neighborhood system)** $\mathcal{G} = \{G_s \mid s \in S\}$ is a neighborhood system for $S$ if

1. $s \not\in G_s$,
2. $s \in G_r \iff r \in G_s$

**Definition 2.2 (Clique)** A subset $C \subseteq S$ is a clique if every pair of distinct sites in $C$ are neighbors. $\mathcal{C}$ denotes the set of cliques and $\deg(C) = \max_{C \in \mathcal{C}} |C|$.

The most commonly used neighborhood systems are the homogeneous systems. In this case, we consider $S$ as a lattice $\mathcal{L}$ and define these neighborhoods as

$$\mathcal{G}^n = \{G^n_{i,j} \mid (i, j) \in \mathcal{L}\},$$

$$G^n_{i,j} = \{(k, l) \in \mathcal{L} : (k \leftrightarrow i)^2 + (l \leftrightarrow j)^2 \leq n\}.$$

Obviously, sites near the boundary have fewer neighbors than interior ones. Furthermore, $\mathcal{G}^0 \equiv S$ and for all $n \geq 0 : \mathcal{G}^n \subset \mathcal{G}^{n+1}$. Figure 1 shows a first-order neighborhood corresponding to $n = 1$. The cliques are $\{(i, j)\}, \{(i, j), (i, j+1)\}, \{(i, j), (i+1, j)\}$. 

![Figure 1: First order neighborhood system with cliques](Image)
2.2 Gibbs Distribution and MRF’s

Let $X = \{ X_s : s \in S \}$ denote any family of random variables so that $\forall s \in S : X_s \in \Lambda$ where $\Lambda = \{ 0, 1, \ldots, L \leftrightarrow 1 \}$ is a common state space. Let $\Omega = \{ \omega = (\omega_1, \ldots, \omega_N) : \omega_i \in \Lambda, 1 \leq i \leq N \}$ be the set of all possible configurations.

**Definition 2.3 (Markov Random Field)** $X$ is a Markov Random Field (MRF) with respect to $G$ if

1. for all $\omega \in \Omega$: $P(X = \omega) > 0$,
2. for every $s \in S$ and $\omega \in \Omega$:
   $$P(X_s = \omega_s \mid X_r = \omega_r, r \neq s) = P(X_s = \omega_s \mid X_r = \omega_r, r \in G).$$

The functions in 2. are called the local characteristics of the MRF and the probability distribution $P(X = \omega)$ of any process satisfying 1. is uniquely determined by these conditional probabilities. However, it is extremely difficult to determine these characteristics in practice.

**Definition 2.4 (Gibbs distribution)** A Gibbs distribution relative to the neighborhood system $G$ is a probability measure $\pi$ on $\Omega$ with the following representation:

$$\pi(\omega) = \frac{1}{Z} \exp \left( \frac{-U(\omega)}{T} \right),$$

where $Z$ is the normalizing constant or partition function:

$$Z = \sum_\omega \exp \left( \frac{-U(\omega)}{T} \right).$$

$T$ is a constant called the temperature and the energy function $U$ is of the form

$$U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega).$$

Each $V_C$ is a function defined on $\Omega$ depending only on those elements $\omega_s$ of $\omega$ for which $s \in C$. Such a function is called a potential.

One of the most important theorems is probably the Hammersley-Clifford theorem [1] which points out the relation between MRF and Gibbs distribution:

**Theorem 2.1 (Hammersley-Clifford)** $X$ is a MRF with respect to the neighborhood system $G$ if and only if $\pi(\omega) = P(X = \omega)$ is a Gibbs distribution with respect to $G$.

The main benefit of this equivalence is that it provides us a simple way to specify MRF’s namely specifying potentials instead of local characteristics (see definition 2.3) which is usually very difficult.
2.3 A General Markov Image Model

We now look at the image labeling model. Image labeling is a general framework to solve low level vision tasks such as image classification, edge detection, etc. To each pixel of the image, we assign a label. The meaning of the labels depends on the task that we want to solve. For image classification, a label means a class; for edge detection, it means the presence or the direction of an edge; etc. Thus, we have the following general problem:

We are given a set of pixels (an image) $S = \{s_1, s_2, \ldots, s_N\}$ with some neighborhood system $\mathcal{G} = \{G_s : s \in S\}$ and $\mathcal{F} = \{f_s : s \in S\}$ a set of image data (or observations). Each of these pixels may take a label from $\Lambda = \{0, 1, \ldots, L \Leftrightarrow 1\}$. The configuration space $\Omega$ is the set of all global discrete labeling $\omega = (\omega_1, \ldots, \omega_N), \omega_s \in \Lambda$. We assume that $\mathcal{X}$ is a MRF relative to $\mathcal{G}$ with a corresponding energy function $U_2$ and potentials $\{V_C\}$:

$$P(\mathcal{X} = \omega) = \frac{1}{Z} \exp \left( \frac{-U_2(\omega)}{T} \right)$$

$$U_2(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega)$$

Now, we will construct a Bayesian estimator to find the optimal labeling that is the labeling which maximizes the posterior distribution $P(\mathcal{X} = \omega | \mathcal{F})$ of the label field:

$$P(\mathcal{X} = \omega | \mathcal{F}) = \frac{P(\mathcal{F} | \mathcal{X} = \omega) P(\mathcal{X} = \omega)}{P(\mathcal{F})}$$

Since $P(\mathcal{F})$ is constant, the MAP estimator of the label field is given by:

$$\max_{\omega \in \Omega} P(\mathcal{X} = \omega | \mathcal{F}) = \max_{\omega \in \Omega} P(\mathcal{F} | \mathcal{X} = \omega) P(\mathcal{X} = \omega).$$

If we assume that the observed image $\mathcal{F}$ is affected at site $s$ only by the pixel $s$ itself (i.e., the image is not blurred) one can prove that $P(\mathcal{F} | \mathcal{X} = \omega)$ is a Gibbs distribution over $\mathcal{G}^0 \equiv \mathcal{S}$ with an energy function $U_1$ and potentials $V_{f_s}$ (a blurred image model is studied for example in [13]). Thus, the posterior distribution is also a MRF over $\mathcal{G}$ with the following energy function:

$$U(\omega) = U_1(\omega) + U_2(\omega)$$

$$U_1(\omega) = \sum_{s \in \mathcal{S}} V_{f_s}(\omega_s)$$

$$U_2(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega_C)$$

Using this function, the MAP estimator is given by:

$$\hat{\omega} = \arg \max_{\omega \in \Omega} P(\mathcal{X} = \omega | \mathcal{F}) = \arg \max_{\omega \in \Omega} \frac{1}{Z} \exp \left( \frac{-U(\omega)}{T} \right)$$

$$= \arg \min_{\omega \in \Omega} U(\omega),$$
2.4 Relaxation Algorithms

It is usually an extremely hard computational problem to find even a near-optimal solution of Equation (8). Standard optimization algorithms do not work in this case since \( U \) is generally a non-convex function. Several approaches have been proposed to solve this task such as Simulated Annealing (SA) \[13\] \cite{24,27} Graduated Non-Convexity (GNC) \[7\] Iterated Conditional Mode (ICM) \[4, \cite{18}\] Mean Field Annealing (MFA) \[6, \cite{12, 30}\] Game Strategy Annealing (GSA) \[25\] Deterministic Pseudo Annealing (DPA) \[22, \cite{3}\] Modified Metropolis Dynamics (MMD) \[22, \cite{23}\]... Multigrid schemes have also been proved to be very efficient for energy minimization \[8, \cite{9, 16, 28}\].

Herein, we review two well known methods. The first one is the Iterated Conditional Mode (ICM) proposed by Besag in \[4\]. This algorithm realizes a deterministic search for the MAP estimate given in Equation (8). An advantage of the method is that the convergence is much faster than that of the stochastic methods. However, the achieved optimum is only a local one located near the initial configuration. Thus, a good initialization is vital for this method. The sequential ICM algorithm is described as follows: Let \( \{n_k, k = 1, 2, \ldots \} \) be the sequence in which the sites are visited for updating and set the temperature parameter \( T \) to 1 throughout the algorithm.

**Algorithm 2.1 (ICM)**

1. Choose a good initial configuration \( \omega^0 \), set \( k = 0 \) and let \( \epsilon > 0 \) be a threshold.
2. \( \omega^{k+1} = \eta \) if \( \eta = \omega^k \big|_{n_k} = \lambda \) for any \( \lambda \in \Lambda \) and \( \eta = \arg \min_{\lambda \in \Lambda} U(\omega^k \big|_{n_k} = \lambda) \).
3. \( k = k + 1 \) and go to 2 until \( |U(\omega^k) \leftrightarrow U(\omega^{k-1})| < \epsilon \).

The second algorithm is the Gibbs Sampler introduced by D.Geman and S.Geman in \[13\]. This method converges to the global optimum (see proof in \[13\]) but the convergence is very slow.
Let again \( \{ n_k \} \) be the sequence of visiting the sites. The temperature parameter is lowered here after each complete sweep of \( S \).

**Algorithm 2.2 (Gibbs Sampler)**

1. Set \( k = 0 \), assign an initial configuration \( \omega^0 \) and let \( \epsilon > 0 \) be a threshold.
2. \( \omega^{k+1} = \eta \) with probability \( \pi_T(\eta) = \exp(-U(\eta)/T) \) if \( \eta = \omega^k \mid_{n_k} = \lambda \) for any \( \lambda \in \Lambda \).
3. \( k = k + 1 \) and \( T \) is lowered if a full sweep is achieved. Go to 2 until \( |U(\omega^k) - U(\omega^{k-1})| < \epsilon \).

The sequence \( \{ n_k, k = 1, 2, \ldots \} \) is usually a simple raster scanning. A more sophisticated scanning is for example the Highest Confidence First (HCF) [11], where the sites with large likelihood ratios of one label over the others are visited earlier.

Let us consider now the parallelization [2]. We use a coding technique similar to the one reported in [4] which requires no modification of the theoretical background of the methods. Considering a SIMD (Single Instruction Multiple Data) machine, we update at the same time the sites which are conditionally independent. More precisely, we partition the set \( S \) into disjoint regions (coding regions or updating sets) such that sites belonging to the same region are conditionally independent given the data of all the other regions. In this way, we can update the sites of a whole region at each iteration instead of one site. The Figure 2 shows a coding in the case of a first order MRF.

Another scheme has been proposed by Graffigne in [15]. In this case, each processor generates a Markov chain at different temperature (see Figure 3). After a certain number of iterations, the processors at higher temperatures transfer their configuration to the lower processors. This configuration is accepted only if it is better (i.e., of lower energy) than the current one. This algorithm can be implemented both on a MIMD (Multiple Instruction Multiple Data) or a SIMD machine assuming in the latter case that we have the same relaxation algorithm for each chain. The only problem with this scheme is that it is not equivalent to its sequential version and the convergence of such a method is not proved.

We use the idea of performing relaxation at different temperatures in our Multi-Temperature Annealing schedule (see Section 5) but in our case, the convergence has been proved (see Appendix A).

### 3 The Classical Multiscale Model

This multiscale model has been proposed by F. Heitz et al. in [16] for motion analysis using a second order neighborhood system (see Definition 2.1). Herein, we give a more general description of this model then we present the energy-minimization scheme and finally we study it in the case of a first order neighborhood system (see Figure 1) which is the most commonly used in image classification problems.
3.1 General Description

Let us suppose again that $\mathcal{S} = \{s_1, \ldots, s_N\}$ is a $W \times H$ lattice so that:

$$\mathcal{S} \equiv \mathcal{L} = \{(i, j) : 1 \leq i \leq W \text{ and } 1 \leq j \leq H\},$$

and $1 \ W = w^n \Gamma H = h^m$. Furthermore, we have some neighborhood system $\mathcal{G}$ on these sites. Let $\mathcal{X}$ be a MRF over $\mathcal{G}$ with an energy function $U$ and potentials $\{V_C\}_{C \in \mathcal{C}}$. The following procedure will generate the multigrid MRF corresponding to $\mathcal{X}$:

1. Let $\mathcal{B}^0 \equiv \mathcal{S}$ and $\Omega_0 \equiv \Omega$.

2. For all $1 \leq i \leq M$ ($M = \inf(n, m)\Gamma$), $\mathcal{S}$ is divided into blocks of size $w_i \times h_i$. These blocks will form the scale $\mathcal{B}^i = \{b_i^1, \ldots, b_i^{N_i}\}$ ($N_i = N/(wh)^i$).

The labels assigned to the sites of a block are supposed to be the same over the whole block. The common label of the block $b_i^k$ is denoted by $\omega_i^k \in \Lambda$. This constraint yields a configuration space $\Omega_i$ which is a subset of the original set $\Omega$. Obviously, for all $0 \leq i \leq M$: $\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 \equiv \Omega$.

Now let us consider the neighborhood system at scale $i$. It is clear that $b_i^j$ and $b_i^l$ are neighbors if and only if there exist two neighbors $s \in \mathcal{S}$ and $r \in \mathcal{S}$ such that $s \in b_i^j$ and $r \in b_i^l$. This yields the same cliques as in $\mathcal{C}$. The cliques can be defined in the following way: Let

\[\Phi_i: \mathcal{B}^i \rightarrow \mathcal{S}^i\]

\[\Phi_0: \mathcal{B}^0 \rightarrow \mathcal{S}^0\]

\[\Phi^2: \mathcal{B}^2 \rightarrow \mathcal{S}^2, \quad \Phi^1: \mathcal{B}^1 \rightarrow \mathcal{S}^1\]

\[\Phi^0: \mathcal{B}^0 \rightarrow \mathcal{S}^0\]

**Figure 4**: The isomorphism $\Phi^i$ between $\mathcal{B}^i$ and $\mathcal{S}^i$.

\[\Phi_0: \mathcal{B}^0 \rightarrow \mathcal{S}^0\]

\[\Phi^2: \mathcal{B}^2 \rightarrow \mathcal{S}^2, \quad \Phi^1: \mathcal{B}^1 \rightarrow \mathcal{S}^1\]

\[\Phi^0: \mathcal{B}^0 \rightarrow \mathcal{S}^0\]

---

1. This assumption introduces some restrictions on $\mathcal{L}$ but this is not crucial in practice since we work mostly on images where both $W$ and $H$ is a power of 2.
\[ d = \text{deg}(\mathcal{C}). \] For all \( 1 \leq j \leq d \Gamma \) the set of \( j \) blocks \( C_j^i \) at scale \( i \) is a clique of order \( j \) if there exists a clique \( C \in \mathcal{C} \) (that is a clique at the finest scale) such that:

1. \( C \subseteq \bigcup_{b_j^i \in C_j^i} b_j^i \)
2. \( \forall b_j^i \in C_j^i : C \cap b_j^i \not= \emptyset. \)

The set of cliques at scale \( i \) is denoted by \( \mathcal{C}^i \) \((C^0 \equiv \mathcal{C})\). The set of all cliques satisfying 1 and 2 for a given \( C_j^i \) is denoted by \( \mathcal{D}_{C_j^i} \subseteq \mathcal{C}. \)

Let us partition the original set \( \mathcal{C} \) into the following disjoint subsets: For all \( 1 \leq j \leq d \Gamma \) let \( \mathcal{A}_j^i \) be the set of cliques \( C \in \mathcal{C} \) for which there exists a clique \( C_j^i \) (that is a clique of order \( j \) at the scale \( i \)) satisfying 1 and 2. Then it turns out from the definition of \( \mathcal{D}_{C_j^i} \) and \( \mathcal{A}_j^i \) that

\[ \mathcal{A}_j^i = \bigcup_{C_j^i \in \mathcal{C}^i} \mathcal{D}_{C_j^i}. \] (10)

Using this partition the energy function \( U \) can be decomposed in the following way:

\[ U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega) = \sum_{C \in \mathcal{A}_j^i} V_C(\omega) + \cdots + \sum_{C \in \mathcal{A}_d^i} V_C(\omega) \] (11)

\[ = \sum_{C_j^i \in \mathcal{C}_j^i} \sum_{C \in \mathcal{D}_{C_j^i}} V_C(\omega) + \cdots + \sum_{C_j^i \in \mathcal{C}_j^i} \sum_{C \in \mathcal{D}_{C_j^i}} V_C(\omega) \] (12)

The main benefit of this decomposition is that the potentials at coarser scales can be derived by simple computation from the potentials at the finest scale. If we note the potential corresponding to a clique \( C_j^i \) of order \( j \) at the scale \( i \) by \( V_{C_j^i}^{B^i} \) we have the following family of potentials at scale \( B^i \):

\[ V_{C_j^i}^{B^i}(\omega) = \sum_{C \in \mathcal{D}_{C_j^i}} V_C(\omega) \] (13)

If we examine our model we see that there is some redundancy at coarser scales; we have the same label over the sites of a block. It seems then natural to associate a unique site to each block. These sites have the common state of the corresponding block and they form a coarser grid \( S^i \) isomorphic to the corresponding scale \( B^i \). The coarser configuration space \( \Xi_i = \{ \xi_s^i : s \in S^i, \xi_s^i \in A \} \) is isomorphic to \( \Omega_i \). Obviously \( \Gamma \Xi_0 \equiv \Omega_0 \equiv \Omega \). The isomorphism \( \Phi^i \) from \( S^i \) in \( B^i \) is just a projection of the coarser label field to the fine grid \( S^0 \equiv S \):

\[ \Phi^i : \Xi_i \leftrightarrow \Omega_i \]
\[ \xi^i \leftrightarrow \omega = \Phi^i(\xi^i). \] (14)

\( \Phi^i \) keeps the same neighborhood structure on \( S^i \) as on \( B^i \) and the cliques on \( S^i \) inherit the potentials from the cliques defined on \( B^i \). These grids form a pyramid where level \( i \) contains the grid \( S^i \). The energy function of level \( i \) \((i = 0, \ldots, M)\) is of the form:

\[ U^i(\xi^i) = \sum_{C \in \xi^i} V_C^i(\xi^i) \quad i = 0, \ldots, M \] (15)

where \( V_C^i(\xi^i) = V_C^{B^i}(\Phi^i(\xi^i)) \). (16)
3.2 Relaxation Scheme

Basically this is the same procedure as in the monogrid case. The only difference is that we have more functions to minimize which are less complex than the original one. The algorithm is the following (see Figure 5): instead of minimizing the original energy function $U$ we tackle the sequence of problems $U^i (M \geq i \geq 0)$ using a top-down strategy in the pyramid. First we solve the problem at a higher level $i$ using a parallel relaxation scheme then the level $i \leftrightarrow 1$ is initialized by $(\Phi^i)^{-1} \circ \Phi^i(\hat{\xi}^i)$ which is just a projection of $\hat{\xi}^i$ on the finer grid $S^{i-1}$ ($\hat{\xi}^i$ is the solution at level $i$).

The advantages of this algorithm are clear: each $\hat{\xi}^i$ gives a more or less good estimate of the final result. The estimate is better as $i$ goes down to 0. On the other hand for the higher values of $i$ the corresponding problem is simpler since the state space has only a few elements.

This scheme is particularly well adapted to the deterministic relaxation methods which are more sensitive to the initial configuration than the stochastic ones. In our experiments (see section 7) the final result is improved compared to the monogrid version of the same algorithm. However for the stochastic ones the final result is only slightly improved since these methods are independent of the initial configuration.

Another important measure of the efficiency is the speed of convergence. On a sequential
machine the proposed scheme exhibits fast convergence properties. However, on a SIMD machine the speed depends mainly on the Virtual Processor Ratio (VPR = number of the virtual processors per physical processor). This means that the monogrid scheme may be faster on such a machine considering the (very simple) parallelization described above. Faster because the multiscale scheme demands usually more iterations (the relaxation algorithm must converge at each level and there is a minimal number of iterations necessary for the convergence). In our experiments the monogrid scheme was always faster than this one on a Connection Machine CM200 (see Section 7).

We note that in [16] another parallelization scheme has been proposed which consists of generating configurations in parallel using different temperatures at different levels with periodic interactions between them. The interaction introduces a transfer at every $n$ iterations of a small block of labels to the next finer level. The block is accepted if the energy of the new block is lower (deterministic rule). We also implemented a finer version of this scheme. In our approach each site at each iteration transfers its state to the next lower level. At the lower scale this information is taken into account as the state of an additional neighbor site. The transition is then governed by the Gibbs Sampler or any other method taking into account this external information (probabilistic rule).

The problem with both algorithms is that to our knowledge the convergence of such an algorithm has not been proved. Looking for a better parallelization scheme with a theoretical background may be a future work.

### 3.3 A Special Case

In the following we will focus on a MRF with a first order neighborhood-system (see Figure 1) where the energy function is given by:

$$U(\omega, F) = U_1(\omega, F) + U_2(\omega)$$

where $U_1$ (resp. $U_2$) denotes the energy of the first order (resp. second order) cliques. The notation $U_i(\omega, F)$ means that the first order potentials depend not only on the actual labeling but also on the given observations.

We follow the procedure described in the Section 3.1 to generate a multigrid MRF model. Let $B^i = \{b_1^i, \ldots, b_{N^i}^i\}$ denote the set of blocks and $\Omega^i$; the configuration-space at scale $i$ ($\Omega^i \subset$...
Now, we can define our pyramid (see Figure 4) where level $i$ contains the coarse grid $S^i$ which is isomorphic to the scale $B^i$. The coarse grid has a reduced configuration space $\Xi_i = \Lambda^{N_i}$.

The model on the grids $S^i (i = 0, \ldots, M)$ defines a set of consistent multiscale MRF models $\Gamma$ whose energy functions are derived from Equations (21) and (22)

$$U^i(\xi, \mathcal{F}) = U^i_{1}(\xi, \mathcal{F}) + U^i_{2}(\xi)$$

with $U^i_{1}(\xi, \mathcal{F}) = \sum_{k \in S^i} (V^i_{1}(\omega^i_k, \mathcal{F}) + V^i_{2}(\omega^i_k)) = \sum_{k \in S^i} V^i_{1}(\xi^i_k, \mathcal{F})$ (24)

and $U^i_{2}(\xi) = \sum_{\{k,l\text{neighbors}\}} V^i_{2}(\omega^i_k, \omega^i_l) = \sum_{C^i \in \mathcal{C}^i} V^i_{2}(\xi^i_C)$ (25)

where $C^i$ is a second order clique corresponding to the definition in Equation (18) and $\mathcal{C}^i$ is the set of cliques on the grid $S^i$.

$$\Omega_{i-1} \subset \cdots \subset \Omega_0 = \Omega$$. The label associated with block $b^i_k$ is denoted by $\omega^i_k$. We can define the same neighborhood structure on $B^i$ as on $S$:

$$b^i_k \text{ and } b^i_l \text{ are neighbors} \iff \exists C \in \mathcal{C} \mid C \cap b^i_k \neq \emptyset \text{ and } C \cap b^i_l \neq \emptyset$$

(18)

Now let us partition the original set $\mathcal{C}$ into two disjoint subsets $\{\mathcal{C}^i\}$ and $\{\mathcal{C}^i_{k,l}\}$:

1. cliques which are included in $b^i_k$ (see Figure 6/a.):

$$\mathcal{C}^i_k = \{C \in \mathcal{C} \mid C \subset b^i_k\}$$

(19)

2. cliques which sit astride two neighboring blocks $\{b^i_k, b^i_l\}$ (see Figure 6/b.):

$$\mathcal{C}^i_{k,l} = \{C \in \mathcal{C} \mid C \subset (b^i_k \cup b^i_l) \text{ and } C \cap b^i_k \neq \emptyset \text{ and } C \cap b^i_l \neq \emptyset\}$$

(20)

It is obvious from this partition that our energy function (see Equation (17)) can be decomposed as:

$$U_1(\omega, \mathcal{F}) = \sum_{s \in S} V_1(\omega_s, f_s) = \sum_{b^i_k \in B^i} \sum_{s \in b^i_k} V_1(\omega_s, f_s) = \sum_{b^i_k \in B^i} V^i_{B^i}(\omega^i_k, \mathcal{F})$$

and

$$U_2(\omega) = \sum_{C \in \mathcal{C}} V_2(\omega_C) = \sum_{b^i_k \in B^i} \sum_{C \in \mathcal{C}^i_k} V_2(\omega_C) = \sum_{b^i_k \in B^i} V^i_{B^i}(\omega^i_k)$$

(21)

$$= \sum_{b^i_k \in B^i} V^i_{B^i}(\omega^i_k) + \sum_{\{k,l\text{neighbors}\}} V^i_{B^i}(\omega^i_k, \omega^i_l)$$

(22)

Now we can define our pyramid (see Figure 4) where level $i$ contains the coarse grid $S^i$ which is isomorphic to the scale $B^i$. The coarse grid has a reduced configuration space $\Xi_i = \Lambda^{N_i}$.

3.3 A Special Case

\begin{align*}
\Omega_{i-1} \subset \cdots \subset \Omega_0 = \Omega.
\end{align*}
4 The Hierarchical Model

In this section, we propose a new hierarchical MRF model. The basic idea is to find a better way of communication between the levels than the initialization used for the multiscale model. Our approach consists in introducing new interactions between two neighbor grids\(^2\) in the pyramid. This scheme permits also the parallelization of the relaxation algorithm on the whole pyramid.

First, we give a general description of the model, then we study a special case with a first order neighborhood system.

4.1 General Description

We consider here the label pyramid and the whole observation field defined in the previous section. Let \(\mathcal{S} = \{\bar{s}_1, \ldots, \bar{s}_N\}\) denote the sites of this pyramid. Obviously,

\[
\mathcal{S} = \bigcup_{i=0}^{M} S^i
\]

\[
\bar{N} = \sum_{i=0}^{M} N_i.
\]

\(^2\)One can imagine interactions between more than two levels but these schemes are too complicated for practical use.
\( \Omega \) denotes the configuration-space of the pyramid:

\[
\Omega = \Xi^0 \times \Xi^1 \times \ldots \times \Xi^M = \{ \bar{\omega} \mid \bar{\omega} = (\xi_0, \xi_1, \ldots, \xi_M) \}
\] (27)

Let us define the following function \( \Psi \) between two neighbor levels which assigns to a site of any level the corresponding block of sites at the level below it (that is its descendants). \( \Psi^{-1} \) assigns its ancestor to a site (see Figure 7):

\[
\Psi : \quad S^i \leftrightarrow S^{i-1} \\
\Psi(\bar{s}) = \{ \bar{\tau} \mid \bar{s} \in S^i \Rightarrow \bar{\tau} \in S^{i-1} \text{ and } b^i_{\bar{\tau}} \subseteq b^i_{\bar{s}} \}
\] (28)

Now we can define on these sites the following neighborhood-system (see Figure 8):

\[
\tilde{G} = \left( \bigcup_{i=0}^{M} G_i \right) \cup \{ \Psi^{-1}(\bar{s}) \cup \Psi(\bar{s}) \mid \bar{s} \in \bar{S} \}
\] (29)

where \( G_i \) is the neighborhood structure of the \( i^{th} \) level and we have the following cliques:

\[
\mathcal{C} = \left( \bigcup_{i=0}^{M} \mathcal{C}^i \right) \cup \mathcal{C}^* (30)
\]

where \( \mathcal{C}^* \) denotes the new cliques sitting astride two neighbor grids. We can easily estimate the degree of the new cliques since it depends on the block size: Each site interacts with its ancestor (there is one) and its descendants (there are \( wh \)) thus:

\[
\deg(\mathcal{C}^*) = \max_{\mathcal{C}^* \in \mathcal{C}^*} | \mathcal{C}^* | = wh + 2
\] (31) and

\[
\deg(\mathcal{C}) = \deg(\mathcal{C}) + \deg(\mathcal{C}^*) \Leftrightarrow 1.
\] (32)

Furthermore let \( \mathcal{X} \) be a MRF over \( \mathcal{G} \) with energy function \( \bar{U} \) and potentials \( \{ \bar{V}_C \}_{C \in \mathcal{C}^*} \). The energy function is of the following form:

\[
\bar{U}(\bar{\omega}) = \sum_{\bar{C} \in \mathcal{C}} \bar{V}_C(\bar{\omega}) = \sum_{i=0}^{M} \sum_{\bar{C} \in \mathcal{C}^i} \bar{V}^i_C(\bar{\omega}) + \sum_{\bar{C} \in \mathcal{C}^*} \bar{V}_C(\bar{\omega}) = \sum_{i=0}^{M} \sum_{\bar{C} \in \mathcal{C}^i} \bar{V}^i_C(\bar{\xi}^i) + \sum_{\bar{C} \in \mathcal{C}^*} \bar{V}_C(\bar{\omega}) = \sum_{i=0}^{M} U^i(\bar{\xi}^i) + U^*(\bar{\omega})
\] (33)

It turns out from the above equation that the energy function consists of two terms. The first one corresponds to the sum of the energy functions of the grids defined in the previous section and the second one \( (U^*(\bar{\omega})) \) is the energy over the new cliques located between neighbor grids.
4.2 A Special Case

In this section, we study the model in the case of a first order neighborhood system. We will consider herein only first and second order cliques. Clique-potentials for the other cliques are supposed to be 0. The cliques can be partitioned into three disjoint subsets $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ corresponding to first order cliques, second order cliques which are on the same level, and second order cliques which sit astride two neighboring levels (see Figure 8). Using this partition, we can derive the following energy function:

\begin{align}
\bar{U}(\bar{\omega}, F) &= \bar{U}_1(\bar{\omega}, F) + \bar{U}_2(\bar{\omega}) \\
\bar{U}_1(\bar{\omega}, F) &= \sum_{\pi \in \mathcal{S}} \bar{V}_1(\bar{\omega}, F) \\
&= \sum_{i=0}^{M} \sum_{\xi \in \mathcal{S}^i} V^i_1(\xi^i, F) = \sum_{i=0}^{M} U^i_1(\xi^i, F) \\
\bar{U}_2(\bar{\omega}) &= \sum_{C \in \mathcal{C}_2} V^i_2(\bar{\omega}_C) + \sum_{C \in \mathcal{C}_3} V^i_2(\bar{\omega}_C) \\
&= \sum_{i=0}^{M} \sum_{C \in \mathcal{C}_2^i} V^i_2(\xi^i_C) + \sum_{C \in \mathcal{C}_3^i} V^i_2(\xi^i_C) \\
&= \sum_{i=0}^{M} U^i_2(\xi^i_C) + \sum_{C \in \mathcal{C}_3} V^i_2(\bar{\omega}_C) 
\end{align}

In the next section, we propose a new annealing scheme for the efficient minimization of the energy function of the hierarchical model.

5 Multi-Temperature Annealing

5.1 Parallel Relaxation Scheme

Now let us see how the energy $U(\bar{\omega})$ is minimized. If we use a deterministic relaxation method where the temperature parameter is kept constant during the iterations (for example ICM [4]), then the original formulation of the algorithm does not change. The only difference is that we work on a pyramid and not on a rectangular shape as in the monogrid case. We can easily parallelize this algorithm using the coding technique described by Besag in [4]: we partition the pyramid $\mathcal{S}$ into disjoint updating sets so that pixels which belong to the same set are conditionally independent given the data of all the other sets. This enables us to update different levels at the same time (see Figure 9).

Let us consider in the followings a relaxation algorithm where the temperature change during the iterations. The temperature-change is controlled by a function $\Gamma$ the so-called annealing schedule. Such a method is for example the Simulated Annealing (Gibbs Sampler [13], Metropolis algorithm [24][27]) or some deterministic schemes such as Modified Metropolis Dynamics [21][23]. For these algorithms, we introduce a new annealing schedule: the Multi-Temperature Annealing
The idea is to associate different temperatures to different levels in the pyramid. For the cliques sitting between two levels we use either the temperature of the lower level or the higher level (but once chosen we always keep the same level throughout the algorithm). For the parallelization we use the same coding technique as in the previous case.

We have three ways of annealing. The first two are well known [24]; they require no modification of the original algorithm except that we work on a pyramid instead of a rectangular shape. The third one is a new annealing schedule which is the most efficient with the hierarchical model:

1. **Homogeneous annealing**: We assign to each level of the pyramid the same initially high temperature. The relaxation is performed with this fixed temperature until an equilibrium is reached (i.e., until the change of the energy function associated with the model is less than a threshold). The temperature is then lowered. The algorithm is stopped at a low temperature closed to 0. This algorithm can be described by a sequence of homogeneous Markov chains which are generated at a fixed temperature. The temperature will be decreased in between subsequent Markov chains.

2. **Inhomogeneous annealing**: The same initially high temperature is assigned to each level however the temperature is now lowered after each transition. In this case the algorithm
Multi-Temperature Annealing (MTA): To the higher levels, we associate higher temperatures which enable the algorithm to be less sensitive to local minima. However at a finer resolution, the relaxation is performed at a lower temperature (at the bottom level, it is closed to 0).

In all cases, the final configuration of the finest level is taken as the solution of the problem.

5.2 Complexity

In this section, we study the complexity of the optimization of the hierarchical model in terms of the required memory (or number of processors in the parallel implementation) and the required communication compared to the monogrid model.

Memory/processor: We refer to the notations of the Section 3; let us suppose that our image is of the size $W \times H$. Following the procedure described in Section 3.1, we generate a pyramid containing $M + 1$ levels. Without loss of generality, we can assume that $W/w \leq H/h$ where $w \times h$ is the block size. Both $w$ and $h$ are greater than or equal to two. The hierarchical model requires a maximum of $(1 + 1/w)WH$ processors (cf. Equation (37)). Since all levels must be stored at the same time, the memory (or processors) required for the storage of these levels (see Figure 10) considering a rectangular shape is given by:

$$WH + \frac{WH}{w} + \frac{WH}{(wh)^2} + \cdots + \frac{WH}{(wh)^M} = WH \sum_{i=0}^{M} \frac{1}{(wh)^i} < \left(1 + \frac{1}{w}\right)WH$$  (37)
Communication: Considering only the first and second order cliques (mostly used in practice see Figure 11) it is clear that we have \((wh + 1)\) more communications per processors. Each site interacts with its ancestor (there is one) and its descendants (there are \(wh\)).

It turns out that the new model demands more processors and more computer time. However as we can see later experiments show that the new interaction is a better way to communicate between the grids yielding faster convergence (with respect to the number of iterations) for the stochastic relaxation algorithms and giving estimates which are closer to the global optimum for deterministic as well as for stochastic relaxation schemes.

5.3 Convergence Study

The mathematical model of the Simulated Annealing (SA) was extensively studied by Aarts and van Laarhoven in [24]. We briefly describe this model considering the Metropolis algorithm (for the Gibbs sampler see [13]):

The SA generates a sequence of configurations which constitutes a Markov chain. \(P_{\omega,\eta}(k \leftrightarrow 1, k)\) is the probability that the configuration obtained after \(k\) transition is \(\eta\) given the previous configuration is \(\omega\) (both \(\eta\) and \(\omega \in \Omega\)). Furthermore let \(X(k)\) denotes the state reached after the \(k^{th}\) transition. The probability of this event is given by:

\[
P(X(k) = \omega) = \sum_{\zeta} P(X(k \Rightarrow 1) = \zeta) P_{\omega,\eta}(k \Rightarrow 1, k) \quad k = 1, 2, \ldots
\]  

If the transition probability \(P_{\omega,\eta}(k \leftrightarrow 1, k)\) does not depend on \(k\) the corresponding chain is homogeneous otherwise it is inhomogeneous. The transition probabilities depend also on the temperature parameter \(T\). Thus if \(T\) is kept constant the chain will be homogeneous and the transition matrix \(P = P(T)\) can be written as:

\[
P_{\omega,\eta}(T) = \begin{cases} G_{\omega,\eta}(T)A_{\omega,\eta}(T) & \forall \eta \neq \omega \\ 1 \Leftrightarrow \sum_{\zeta} G_{\omega,\zeta}(T)A_{\omega,\zeta}(T) & \eta = \omega \end{cases}
\]  

Where \(G_{\omega,\eta}(T)\) is the generation probability of generating \(\eta\) from \(\omega\) and \(A_{\omega,\eta}(T)\) is the acceptance probability of configuration \(\eta\) once it has been generated from \(\omega\). It is clear from the definition in Equation (39) that \(P(T)\) is a stochastic matrix:

\[
\forall \omega : \sum_{\zeta} P_{\omega,\zeta}(T) = 1
\]  

In the original formulation of the algorithm \(G_{\omega,\eta}(T)\) is given by the uniform distribution on the configurations \(\eta\) which differs from \(\omega\) only for one component. \(A(T)\) is given by the Metropolis criterion:

\[
A_{\omega,\eta}(T) = \min(1, \exp(\langle U(\eta) \leftrightarrow U(\omega) \rangle / T))
\]  

where \(U(\omega)\) is the cost or energy function.

The SA algorithm obtains a global minimum iff after \(K\) transitions (\(K\) sufficiently large) the following holds:

\[
P(X(K) \in \Omega_{opt}) = 1
\]
where $\Omega_{\text{opt}}$ is the set of globally minimal configurations. In the following sections we give the conditions on the matrices $A(T)$ and $G(T)$ described by Aarts and van Laarhoven in [24] to ensure the convergence:

$$\lim_{T \to 0} \lim_{K \to \infty} P(X(K) \in \Omega_{\text{opt}}) = 1$$  \hfill (43)

### 5.3.1 Homogeneous Annealing

In this case we associate the same temperature $T$ to each level of the pyramid. Then configurations are generated at this fixed temperature until equilibrium is reached. What does it mean from the point of view of the generated sequence of states? They constitute an homogeneous Markov chain with transition matrix given by Equation (39). After the equilibrium is reached the temperature is lowered and configurations are generated at this temperature etc...

Thus the algorithm is described by a sequence of homogeneous Markov chains. Each chain is generated at a fixed value of $T$ and $T$ is decreased in between subsequent chains. Therefore the SA algorithm on the pyramid is identical to its monogrid version and we can apply the same proof [24]:

First conditions are given on the existence of the stationary distribution of the chain then further conditions are imposed to assure the convergence of the stationary distribution.

**Theorem 5.1 (Feller)** The stationary distribution $q$ of a finite homogeneous Markov chain exists if the Markov chain is irreducible and aperiodic. Furthermore, the vector $q$ is uniquely determined by the following equations:

$$\forall \omega : q_\omega > 0, \sum_\omega q_\omega = 1,$$

$$\forall \omega : q_\omega = \sum_\eta q_\eta P_{\eta, \omega}.$$  \hfill (45)  

**Definition 5.1 (Irreducibility)** A Markov chain is irreducible iff for all pairs of configurations $(\omega, \eta)$ there is a positive probability of reaching $\eta$ from $\omega$ in a finite number of transitions:

$$\forall \omega, \eta \exists n : 1 \leq n < \infty \land (P^n)_{\omega, \eta} > 0.$$  \hfill (46)  

**Definition 5.2 (Aperiodicity)** A Markov chain is aperiodic iff for all configurations $\omega \in \Omega$, the greatest common divisor of all integers $n \geq 1$, such that $(P^n)_{\omega, \omega} > 0$ is equal to 1.

Since $\forall \omega, \eta, T > 0 : A_{\omega, \eta}(T) > 0$ (see Equation (41)) and $\forall \omega, \eta (\omega \neq \eta), T > 0 : G_{\omega, \eta}(T) > 0$ ($G(T)$ is uniform) irreducibility is satisfied. To establish aperiodicity the following theorem is used:

**Theorem 5.2** An irreducible Markov chain is aperiodic if the following condition is satisfied:

$$\forall T > 0 : \exists \omega_T \in \Omega : P_{\omega_T, \omega_T}(T) > 0.$$  \hfill (47)
Thus, for aperiodicity it is sufficient to assume that
\[ \forall T > 0 : \exists \omega, \eta \in \Omega : A_{\omega, \eta} < 1 \tag{48} \]
which is always satisfied by setting \( \Gamma = 0 \).

Now, further conditions are imposed to ensure the convergence of the stationary distribution (for more details, see [24]):

1. \( \forall \omega, \eta \in \Omega : G_{\eta, \omega} = G_{\omega, \eta} \tag{49} \)
2. \( \forall \omega, \eta, \kappa \in \Omega : U(\omega) \leq U(\eta) \leq U(\kappa) \Rightarrow A_{\omega, \eta}(T) = A_{\omega, \eta}(T)A_{\eta, \kappa}(T) \tag{50} \)
3. \( \forall \omega, \eta \in \Omega : U(\omega) \leq U(\eta) \Rightarrow A_{\omega, \eta}(T) = 1 \tag{51} \)
4. \( \forall \omega, \eta \in \Omega, T > 0 : U(\omega) < U(\eta) \Rightarrow A_{\omega, \eta}(T) < 1 \tag{52} \)
5. \( \forall \omega, \eta \in \Omega : U(\omega) < U(\eta) \Rightarrow \lim_{T \to \infty} A_{\omega, \eta}(T) = 0 \tag{53} \)

It is clear that the condition 1. is satisfied in our case since \( G(T) \) is uniform. Its diagonal is 0 and the other elements are equals. The conditions 2. ⇔ 5. are implied by the definition of \( A(T) \) in Equation (41). These conditions are sufficient but not necessary to ensure the convergence. In this case, the stationary distribution is given by:
\[ q_{\omega}(T) = \frac{\exp\left(\Psi(\omega) \Leftrightarrow U_{\text{opt}}(T)\right)}{\sum_{\eta \in \Omega} \exp\left(\Psi(\omega) \Leftrightarrow U_{\text{opt}}(T)\right)} \tag{54} \]

### 5.3.2 Inhomogeneous Annealing

In this case, we associate the same temperature \( T \) to each level. Then configurations are generated such that after each transition \( T \) is lowered. The obtained sequence of configurations constitutes an inhomogeneous Markov chain whose transition matrix \( P(k \Leftrightarrow 1, k) \) is defined by:
\[
P_{\omega, \eta}(k \Leftrightarrow 1, k) = \begin{cases} 
G_{\omega, \eta}(T_k)A_{\omega, \eta}(T_k) & \forall \eta \neq \omega \\
1 \Leftrightarrow \sum_{\zeta \neq \omega} G_{\omega, \zeta}(T_k)A_{\omega, \zeta}(T_k) & \eta = \omega 
\end{cases} \tag{55}
\]
As previously, the annealing scheme on the pyramid is analogous to the monogrid case; thus the same proof [24] is applied.

Since \( T \) is changed in between subsequent transitions, the conditions of the convergence not only relate to the matrices \( G(T_k) \) and \( A(T_k) \) but also impose restrictions on the way of changing the control parameter \( T \). The following assumption are made:

1. \( \lim_{k \to \infty} T_k = 0 \tag{56} \)
2. \( T_k \geq T_{k+1}, \ k = 0, 1, \ldots \tag{57} \)

**Definition 5.3 (Weak ergodicity)** An inhomogeneous Markov chain is weakly ergodic if for all \( m \geq 1, \omega, \eta, \zeta \in \Omega: \)
\[ \lim_{k \to \infty} (P_{\omega, \zeta}(m, k) \Leftrightarrow P_{\eta, \zeta}(m, k)) = 0 \tag{58} \]
where the transition matrix \( P(m, k) \) is defined by:
\[ P_{\omega, \eta}(m, k) = P(X(k) = \eta | X(m) = \omega). \tag{59} \]
\textbf{Definition 5.4 (Strong ergodicity)} An inhomogeneous Markov chain is strongly ergodic if there exists a vector $\pi$, satisfying:

$$\sum_{\omega} \pi_{\omega} = 1, \forall \omega : \pi_{\omega} > 0,$$  \hfill (60)

such that for all $m \geq 1, \omega, \eta \in \Omega$:

$$\lim_{k \to \infty} P_{\omega,\eta}(m, k) = \pi_{\eta}.$$  \hfill (61)

\textbf{Theorem 5.3 (Seneta)} An inhomogeneous Markov chain is weakly ergodic iff there is a strictly increasing sequence of positive numbers $k_l$ such that

$$\sum_{i=1}^{\infty} (1 \Leftrightarrow \tau_1(P(k_i, K_{i+1}))) = \infty;$$  \hfill (62)

where $\tau_1(P)$, the coefficient of ergodicity of $P$, is defined as

$$\tau_1(P) = 1 \Leftrightarrow \min_{\omega,\eta} \sum_{\zeta} \min(P_{\omega,\zeta}, P_{\eta,\zeta}).$$  \hfill (63)

\textbf{Theorem 5.4 (Isaacson and Madsen)} An inhomogeneous Markov chain is strongly ergodic if it is weakly ergodic and if for all $k$ there exists a vector $\pi(k)$ such that $\pi(k)$ is an eigenvector with eigenvalue 1 of $P(k \Leftrightarrow 1, k)$, $\sum_{\omega} | \pi_{\omega}(k) | = 1$ and

$$\sum_{k=1}^{\infty} \sum_{\omega \in \Omega} | \pi_{\omega}(k) \Leftrightarrow \pi_{\omega}(k+1) | < \infty.$$  \hfill (64)

Moreover, if $\pi = \lim_{k \to \infty} P_{\omega,\eta}(m, k)$, then $\pi$ is the vector in Definition 5.4 satisfying Equation (61).

Under the assumptions of the previous section on $A(T)$ and $G(T)$, there exists an eigenvector $q(T_k)$ of $P(k \Leftrightarrow 1, k)$ for each $k \geq 0$, namely the stationary distribution of the homogeneous Markov chain. Using Theorem 5.4 with $\pi(k) = q(T_k)$, strong ergodicity can be proved by showing that the following conditions are satisfied:

1. The Markov chain is weakly ergodic
2. The $q(T_k)$ satisfy Equation (64).

This was proved by Geman and Geman in [13]. They show that if

$$T_k \geq \frac{N \Delta}{\log k},$$  \hfill (65)

$$\Delta = \max_{\omega \in \Omega} U(\omega) \Leftrightarrow \min_{\omega \in \Omega} U(\omega),$$  \hfill (66)

then Equation (62) is satisfied and thus weak ergodicity is obtained.
5.3 Convergence Study

5.3.3 Multi-Temperature Annealing

The main purpose and study of this section is a new Multi-Temperature Annealing (MTA) schedule. We associate different temperatures to different levels of the pyramid such that the temperature depends not only on the iteration but also on the clique. In this case, the configurations are generated at different temperatures at different sites. The temperature is then lowered after each transition according to the MTA schedule (see Theorem 5.5). More generally, we have the following problem:

Let \( S = \{s_1, \ldots, s_N\} \) be a set of sites \( G \) some neighborhood system with cliques \( C \) and \( X \) a MRF over these sites with energy function \( U \). We define an annealing scheme where the temperature \( T \) depends on the iteration \( k \) and on the cliques \( C \). Let \( \ominus \) denotes the following operation:

\[
P(X = \omega) = \pi_{T(k,C)}(\omega) = \exp(\ominus U(\omega) \ominus T(k,C))
\]

where \( U(\omega) \ominus T(k,C) = \sum_{C \in C} V_C(\omega) \overline{T(k,C)} \).

Let us suppose that the sites are visited for updating in the order \( \{n_1, n_2, \ldots \} \subset S \). The resulting stochastic process is denoted by \( \{X(k), k = 0, 1, 2, \ldots \} \) where \( X(0) \) is the initial configuration. \( X(k) \) is an inhomogeneous Markov chain with transition matrix:

\[
P_{\omega,\eta}(k \leftrightarrow 1, k) = \begin{cases} 
G_{\omega,\eta}(T(k,C))A_{\omega,\eta}(T(k,C)) & \forall \eta \neq \omega \\
1 \ominus \sum_{\zeta \neq \omega} G_{\omega,\zeta}(T(k,C))A_{\omega,\zeta}(T(k,C)) & \eta = \omega 
\end{cases}
\]

Considering the Gibbs sampler, the generation matrix \( G_{\omega,\eta}(T(k,C)) \) and acceptance matrix \( A_{\omega,\eta}(T(k,C)) \) is given by:

\[
G_{\omega,\eta}(T(k,C)) = \begin{cases} 
1, & \text{if } \eta = \omega \mid \omega_k = \lambda \text{ for some } \lambda \in \Lambda \\
0, & \text{otherwise}
\end{cases}
\]

\[
A_{\omega,\eta}(T(k,C)) = \pi_{T(k,C)}(\eta)
\]

The transition matrix at time \( k \) is then of the following form:

\[
P_{\omega,\eta}(k) = G_{\omega,\eta}(k)A_{\omega,\eta}(T(k,C)) = \begin{cases} 
\pi_{T(k,C)}(\eta), & \text{if } \eta = \omega \mid \omega_k = \lambda \text{ for some } \lambda \in \Lambda \\
0, & \text{otherwise}
\end{cases}
\]

Let \( \Omega_{\text{opt}} \) be the set of globally optimal configurations:

\[
\Omega_{\text{opt}} = \{\omega \in \Omega : U(\omega) = \min_{\eta} U(\eta)\}
\]

Let \( \pi_0 \) be the uniform distribution on \( \Omega_{\text{opt}} \) and define:

\[
U^{\text{sup}} = \max_{\omega} U(\omega), \quad U^{\text{inf}} = \min_{\omega} U(\omega), \quad \Delta = U^{\text{sup}} \ominus U^{\text{inf}}.
\]

The following theorem gives an annealing schedule basically the same as in [13]. However, the temperature here is a function of \( k \) and \( C \in C \).

\[\]
Theorem 5.5 (Multi-Temperature Annealing) Assume that there exists an integer \( \kappa \geq N \) such that for every \( k = 0, 1, 2, \ldots \), \( S \subseteq \{ n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa} \} \). Let \( T(k,C) \) be any decreasing sequence of temperatures for which

1. There exists \( T^{inj}_k \) such that for all \( C \in \mathcal{C} \): \( T^{inj}_k \leq T(k,C) \).
2. \( \lim_{k \to \infty} T(k,C) = 0 \)
3. For all \( k \geq k_0 \), for some integer \( k_0 \geq 2 \): \( T^{inj}_k \geq N \Delta / \log k \).

Then for any starting configuration \( \eta \in \Omega \) and for every \( \omega \in \Omega \):

\[
\lim_{k \to \infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega). \quad (75)
\]

The first condition states the existence of a lower bound for the temperature function. The second condition is that the temperature decreases as the system evolves. The third condition assures that each “local” temperature decreases in an appropriate way, i.e., not too quickly. The proof of this theorem can be found in Appendix A.

6 Application to Supervised Image Classification

We have applied the proposed models to supervised image classification. In this section, we give the equations of the MRF models (monogrid, multiscale, hierarchical) used for this problem.

First, we give the equations for the monogrid model [21][23]. Let us suppose that the observations consist of the grey-levels. A very general problem is to find the labeling \( \hat{\omega} \) which maximizes \( P(\omega \mid \mathcal{F}) \). Bayes theorem tells us that \( P(\omega \mid \mathcal{F}) = \frac{1}{Z} P(\mathcal{F} \mid \omega) P(\omega) \). Actually \( P(\mathcal{F}) \) does not depend on the labeling \( \omega \) and we have the assumption that \( P(\mathcal{F} \mid \omega) = \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s) \). It is then easy to see that \( \mathcal{F} \) under some independence assumption [5] the global labeling which we are trying to find is given by:

\[
\hat{\omega} = \max_{\omega \in \Omega} \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s) \prod_{C \in \mathcal{C}} \exp\left( -V_C(\omega_C) \right). \quad (76)
\]

It is obvious from this expression that the a posteriori probability also derives from a MRF. The energies of cliques of order 1 directly reflect the probabilistic modeling of labels without context which would be used for labeling the pixels independently. Let us assume that \( P(f_s \mid \omega_s) \) is Gaussian; the class \( \lambda \in \Lambda \) is represented by its mean value \( \mu_\lambda \) and its deviation \( \sigma_\lambda \). We get the following energy function (see Equation (17)):

\[
U_1(\omega, \mathcal{F}) = \sum_{s \in \mathcal{S}} \left( \log(\sqrt{2\pi\sigma_{\omega_s}}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) \quad (77)
\]
and

\[
U_2(\omega) = \sum_{C \in \mathcal{C}} V_2(\omega_C) \quad (78)
\]
where

\[
V_2(\omega_C) = V_{s,r}(\omega_s, \omega_r) = \begin{cases} \varepsilon \beta & \text{if } \omega_s = \omega_r \\ +\beta & \text{if } \omega_s \neq \omega_r \end{cases} \quad (79)
\]
We can easily extend these equations to the multiscale model using Equations (21) and (22). For simplicity, the block size is supposed to be \( n \times n \) (that is \( w = h = n \)). Then we get [19Γ20]:

\[
U^i_j (\xi^i, \mathcal{F}) = \sum_{x \in S^i} V^i_1 (\xi^i_x, \mathcal{F})
\]

where \( V^i_1(\xi^i_x, \mathcal{F}) = \sum_{x \in B^i} V_1(\omega_x, f_x) + \sum_{C \in C^i} V_2(\omega_C) \)

\[
= \sum_{x \in B^i} \left( \log(\sqrt{2\pi\sigma_{\omega_x}}) + \frac{(f_x - \mu_{\omega_x})^2}{2\sigma_{\omega_x}^2} \right) \Leftrightarrow p^j^i \beta
\]

and \( U^i_2(\xi^i) = \sum_{C^i = \{r^i, s^i\} \in \mathcal{C}^i} V^i_2(\xi^i_C) \)

where \( V^i_2(\xi^i_{C^i}) = \sum_{\{r^i, s^i\} \in \mathcal{D}^i} V_2(\omega_r, \omega_s) = \begin{cases} \Leftrightarrow q^{ij} \beta & \text{if } \omega_r = \omega_s \\ +q^{ij} \beta & \text{if } \omega_r \neq \omega_s \end{cases} \) (81)

The values of \( p^i \) and \( q^i \) depend on the chosen block size and the neighborhood structure. \( p^i \) is the number of cliques included in the same block at scale \( B^i \) and \( q^i \) is the number of cliques between two neighboring blocks at scale \( B^i \). Considering blocks of \( n \times n \) and a first order neighborhood system we get:

\[
p^i = 2n^i (n^i \Leftrightarrow 1)
\]

\[
q^i = n^i
\]

In the case of the hierarchical model we derive [19Γ20] the following equations (using Equations (35) and (36)):

\[
U_1(\omega, \mathcal{F}) = \sum_{i=0}^{M} \sum_{x \in S^i} V^i_1(\xi^i_x, \mathcal{F})
\]

and \( \tilde{U}_2(\omega) = \sum_{i=0}^{M} \sum_{C^i \in \mathcal{C}^i} V^i_2(\xi^i_C) + \sum_{C \in \mathcal{C}} \tilde{V}_2(\omega_C) \) (85)

where \( \tilde{V}_2(\omega_C) = \tilde{V}_{\{r^i, s^i\}}(\omega_r, \omega_s) = \begin{cases} \Leftrightarrow \gamma & \text{if } \omega_r = \omega_s \\ +\gamma & \text{if } \omega_r \neq \omega_s \end{cases} \) (86)

7 Experimental Results

We compare the Gibbs sampler [13] and Iterated Conditional Mode [4Γ18] using three models for each algorithm (original multiscale and hierarchical). We have also compared the inhomogeneous and MTA schedules. All tests have been conducted on a Connection Machine CM200 [17] with 8K processors. In the tables (see Appendix C) we give for each model and for each method the number of levels in the pyramid (for the monogrid model this is 1), the Virtual Processor Ratio (VPR) [17], the initial temperature (for the hierarchical model this is not the same at each level using the new MTA schedule!), the number of iterations, the computing time, the error of the classification (= the number of misclassified pixels) and the parameter \( \beta \) (see Equations (79)), (80), (81)) and \( \gamma \) (see Equation (86)).
7.1 Comparison of the Schedules

In Figure 14 we compare the inhomogeneous and MTA schedules on a noisy synthetic image using the Gibbs sampler. In both cases the parameters were strictly the same: the only difference is the applied schedule: the pyramid contains 4 levels yielding a VPR equals to 4. The initial temperature were respectively 4 (at the highest level) and 2 and 1 (at the lowest level). The potential $\beta$ equals to 0.7 and $\gamma$ equals to 0.1. In Figure 13 (resp. 12) we show the global energy (computed at a fixed temperature) versus the number of iterations of the inhomogeneous (resp. MTA) schedule. Both reach practically the same minimum (53415.4 for the inhomogeneous and 53421.4 for the MTA) However the inhomogeneous schedule requires 238 iterations (796.8 sec. CPU time) but the MTA schedule requires only 100 iterations (340.6 sec. CPU time) for the convergence.

7.2 Comparison of the Models

First we have tested the models on noisy synthetic images of size $128 \times 128$. The first one is a checkerboard image (see Figure 16 Table 1) with 2 classes and a SNR equals to $\approx 5 dB$. Note that the multiscale model gives better results than the monogrid one especially with the ICM algorithm. This type of image is well adapted for the multiscale model because the image has a rectangular structure similar to the model itself. The hierarchical model gives the best results with both algorithms but the computing time is quite greater than for the multiscale or monogrid model. The reason is that in the hierarchical case the whole pyramid is stored at the
same time yielding a greater VPR ratio. On the other hand we cannot use the fast “NEWS” communication scheme [17] as in the other cases.

In the second image we have added different geometrical forms (circle and triangle) to the checkerboard image (see Figure 17 and Table 2). In this case we studied the geometrical sensitivity of the models. The Gibbs sampler gives nearly the same result in all cases. However the ICM is more sensitive. The multiscale model gives better result than the monogrid one but the result is not fine in the triangle and the circle. These forms have a different structure than the block structure of the model, the initialisation was wrong in these regions and the ICM was not able to correct these errors. In the hierarchical case instead of the initialisation we have a real time communication between the levels which is able to give results closed to the ones obtained with the Gibbs sampler. Of course this model requires more computing time than the other ones.

The third image is a checkerboard image with 16 classes (see Figure 18 and Table 3). For ICM there is a significant improvement but for the Gibbs Sampler we observe only a slight improvement for the multiscale and hierarchical cases.

In Figures 19 and 21 we show some real images of size $256 \times 256$: a SPOT image with 4 classes (see Figure 19 and Table 4), an indoor scene with 4 classes (see Figure 20) and a medical image with 3 classes (see Figure 21).

Finally we present a SPOT image of size $512 \times 512$ (see Figure 22) with ground truth data (see Figure 23). In the following table we give the mean ($\mu$) and the deviation ($\sigma^2$) for each class (we have 6 classes):

\[ \text{Table 4: Results for the} \ SPOT \ \text{image} \]

<table>
<thead>
<tr>
<th>Class</th>
<th>$\mu$</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
As we can see, the classes 2 and 5 have nearly the same parameters, it is difficult to distinguish between them. Figure 15 shows the histogram of the original image. We can clearly distinguish three peak (about at 64, 80 and 120). Figure 24 (resp. Figure 25) shows the results obtained with the ICM (resp. Gibbs Sampler). For these results we give a map drawn by an expert (ground truth data, see Figure 23). The classes 1, 5, correspond to the regions $B_3, B_2, B_1, a_2, h$ and $9_2$ on the map. For the hierarchical model, a slight improvement can be noticed for the results of the Gibbs sampler, however, for the ICM, the improvement is more significant. In Table 5 we give the parameters and the computing time for each model and each method.

### 8 Conclusion

In this report we have presented a classical multiscale model and proposed a new hierarchical MRF model based on the classical one. We have introduce a new interaction scheme between two neighbor grids in the label pyramid. The new connections allow to propagate local interactions more efficiently yielding faster convergence (w.r.t. the number of iterations) in many cases and giving estimates closer to the optimum for deterministic as well as for stochastic relaxation techniques. On the other hand these interactions make the model more complex which demands computationally more expensive algorithms.

We have proposed also a new annealing scheme, the Multi-Temperature Annealing, which is especially well adapted to the minimization of the energy function of the hierarchical model. This algorithm can be run in parallel on the entire pyramid and usually decreases the computing time compared to the classical schemes. A generalization of the annealing theorem of Geman and Geman [13] has been proposed which gives a theoretical background for the convergence of this method towards global optimum.

Finally, the hierarchical model as well as the theoretical study given in this report are quite general. They have been adapted for supervised image classification but they can be used for other low level vision tasks such as edge detection, image restoration, data fusion, motion etc. . . . We are currently working on the parameter estimation of these models for unsupervised image classification.
Appendix
A Proof of The Multi-Temperature-Annealing Theorem

We follow the proof of the annealing theorem given by Geman and Geman in [13]. Essentially we can apply the same proof. Only a slight modification is needed.

A.1 Notation

We recall a few notations used previously in this report: $\mathcal{S} = \{s_1, \ldots, s_N\}$ denotes the set of sites. $\Gamma = \{0, 1, \ldots, L \in \mathbb{N}\}$ is a common state space and $\omega, \eta, \eta' \ldots \in \Omega$ denote configurations of the label field $\Gamma$ where $\Omega = \Lambda^N$ is finite. The sites are updated in the order $\{n_1, n_2, \ldots\} \subset \mathcal{S}$. The generated configurations constitute an inhomogeneous Markov chain $\{X(k), k = 0, 1, 2, \ldots\}$ where $X(0)$ is the initial configuration. The transition $X(k \leftrightarrow 1) \rightarrow X(k)$ is controlled by the Gibbs distribution $\pi_{T(k,C)}$ according to the transition matrix at time $k$:

$$P_{\omega,\eta}(k) = G_{\omega,\eta}(k)A_{\omega,\eta}(T(k,C)) = \begin{cases} \pi_{T(k,C)}(\eta), & \text{if } \eta = \omega|_{x_{i_k}=\lambda} \text{ for some } \lambda \in \Lambda \\ 0, & \text{otherwise} \end{cases}$$

(87)

where

$$\pi_{T(k,C)}(\omega) = \exp(-U(\omega) \odot T(k,C))$$

(88)

with $U(\omega) \odot T(k,C) = \sum_{C \in C} \frac{V_C(\omega)}{T(k,C)}$.

We use also the following definitions:

$$U^{\sup} = \max_{\omega} U(\omega),$$

$$U^{\inf} = \min_{\omega} U(\omega),$$

$$\Delta = U^{\sup} \Leftrightarrow U^{\inf}.\quad (90)$$

Given any starting distribution $\mu_0$ the distribution of $X(k)$ is given by the vector $\mu_0 \times \prod_{i=1}^{k} P(k)$:

$$P_{\mu_0}(X(k) = \omega) = \left(\mu_0 \times \prod_{i=1}^{k} P(k)\right)_{\omega}$$

(91)

$$= \sum_{\eta} P(X(k) = \omega|X(0) = \eta)\mu_0(\eta)$$

(92)

We use the following notation for transitions: $\forall l < k$ and $\omega, \eta \in \Omega$:

$$P(k, \omega|l, \eta) = P(X(k) = \omega|X(l) = \eta),$$

and for any distribution $\mu$ on $\Omega$:

$$P(k, \omega|l, \mu) = \sum_{\eta} P(X(k) = \omega|X(l) = \eta)\mu(\eta).$$

Sometimes we use this notation as $P(k, \cdot|l, \mu)\Gamma$ where “.” means any configuration from $\Gamma$. Finally let $\|\mu \Leftrightarrow \nu\|$ denotes the following distance between two distributions on $\Omega$:

$$\|\mu \Leftrightarrow \nu\| = \sum_{\omega} |\mu(\omega) \Leftrightarrow \nu(\omega)|.$$

It is clear that $\lim_{n \to \infty} \mu_n = \mu$ in distribution (i.e. $\forall \omega : \mu_n(\omega) \to \mu(\omega)$) if and only if $\|\mu_n \Leftrightarrow \mu\| \to 0$. 

Zoltan KATO, Marc BERTHOD, Josiane ZERUBIA
A.2 Proof of the Theorem

First we state two lemmas which imply Theorem 5.5:

Lemma A.1 For every \( k_0 = 0, 1, 2, \ldots \):

\[
\lim_{k \to \infty} \sup_{\omega, \eta, \eta'} |P(X(k) = \omega | X(k_0) = \eta') - P(X(k) = \omega | X(k_0) = \eta'')| = 0.
\]

Proof of Lemma A.1:
Fix \( k_0 = 0, 1, 2, \ldots \) and define \( K_i = k_0 + l \Gamma i = 0, 1, 2, \ldots \) where \( \kappa \) is the number of transitions necessary for a full sweep of \( S \) (for every \( k = 0, 1, 2, \ldots \), \( S \subseteq \{ n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa} \} \)). Let \( \delta(k) \) be the smallest probability among the local characteristics (see Definition 2.3):

\[
\delta(k) = \inf_{\substack{i \leq i \leq N \\omega \in \Omega \\text{and define}}} \pi_T(k, C) (X_{s_i} = \omega_{s_i} | X_{s_j} = \omega_{s_j}, j \neq i).
\]

A lower bound for \( \delta(k) \) is given by:

\[
\delta(k) \geq \frac{\exp(\xi(T_{k,C}) \cap T_{k,C})}{L \exp(\xi(T_{k,C}) \cap T_{k,C})} \geq \frac{\exp(\Delta \cap T_{k,C})}{L} \geq \frac{1}{L} \exp(\Delta / T_{k,C}).
\]

Now fix \( l \) and define \( m_i \) as the time of the last replacement of site \( s_i \) before \( K_i + 1 \) (that is before the \( l \)-th full sweep):

\[
\forall 1 \leq i \leq N: m_i = \sup \{ k: k \leq K_i, n_k = s_i \}.
\]

Without loss of generality we can assume that \( m_1 > m_2 \ldots > m_N \) (otherwise relabel the sites). Then:

\[
P(X(K_i) = \omega | X(K_{i-1}) = \omega') = P(X_{s_1}(m_1) = \omega_{s_1}, X_{s_2}(m_2) = \omega_{s_2}, \ldots, X_{s_N}(m_N) = \omega_{s_N} | X(K_{i-1}) = \omega')
\]

\[
= \prod_{i=1}^{N} P(X_{s_i}(m_i) = \omega_{s_i} | X_{s_{i+1}}(m_{i+1}) = \omega_{s_{i+1}}, \ldots, X_{s_N}(m_N) = \omega_{s_N}, X(K_{i-1}) = \omega')
\]

\[
\geq \prod_{i=1}^{N} \delta(m_i) \geq L^{-N} \prod_{i=1}^{N} \exp(\Delta / T_{m_i}^{n_i}) \geq L^{-N} \exp \left( \frac{\Delta N}{T_{k_0 + l \kappa}^{n_i}} \right)
\]

(93)

since \( m_i \leq K_i = k_0 + l \kappa, i = 1, 2, \ldots, N \) and \( T_{k_0}^{n_i} \) is decreasing. If \( k_0 + l \kappa \) is sufficiently large \( \Gamma \), Equation (93) can be continued as:

\[
P(X(K_i) = \omega | X(K_{i-1}) = \omega') \geq L^{-N}(k_0 + l \kappa)^{-1}.
\]

For a sufficiently small constant \( \Gamma \) \( 0 < \Gamma \leq 1 \) we can assume that

\[
\inf_{\omega, \omega'} P(X(K_i) = \omega | X(K_{i-1}) = \omega') \geq \frac{\Gamma L^{-N}}{k_0 + l \kappa}
\]

(94)

for every \( k_0 = 0, 1, 2, \ldots \) and \( l = 1, 2, \ldots \). Keeping in mind that \( K_i \) depends on \( k_0 \).
Consider now the limit given in lemma A.1 and for each \( k > k_0 \) define \( K^{sup}(k) = \sup\{ l : K_l < k \} \) (the last sweep before the \( k^{th} \) transition) so that \( \lim_{k \to \infty} K^{sup}(k) = \infty \). Fix \( k > K_1 \):

\[
\sup_{\omega, \eta, \eta'} P(X(k) = \omega | X(0) = \eta') \iff P(X(k) = \omega | X(0) = \eta'')
\]

\[
= \sup_{\omega} \left( \sup_{\eta} P(X(k) = \omega | X(0) = \eta) \iff \inf_{\eta} P(X(k) = \omega | X(0) = \eta) \right)
\]

\[
= \sup_{\omega} \left( \sup_{\eta} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta) \right)
\]

\[
\iff \inf_{\eta} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta) \right) \equiv \sup_{\omega} Q(k, \omega).
\]

Furthermore, for each \( \omega \in \Omega \):

\[
\sup_{\eta} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') P(X(K_1) = \omega' | X(0) = \eta)
\]

\[
\leq \sup_{\mu} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega'),
\]

where \( \mu \) is any probability measure on \( \Omega \). Using Equation (94), we get:

\[
\mu(\omega') \geq \frac{\Gamma L^{-N}}{k_0 + l_K}.
\]

Suppose that \( P(X(k) = \omega | X(K_1) = \omega') \) is maximized at \( \omega' = \omega^{sup} \) and minimized at \( \omega' = \omega^{inf} \). Then we get:

\[
\sup_{\mu} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega') \leq \left( 1 \iff (L^N \iff 1) \right) \frac{\Gamma L^{-N}}{k_0 + l_K} P(X(k) = \omega | X(K_1) = \omega^{sup})
\]

\[
+ \frac{\Gamma L^{-N}}{k_0 + l_K} \sum_{\omega' \neq \omega^{sup}} P(X(k) = \omega | X(K_1) = \omega')
\]

and in a similar way:

\[
\inf_{\mu} \sum_{\omega'} P(X(k) = \omega | X(K_1) = \omega') \mu(\omega') \geq \left( 1 \iff (L^N \iff 1) \right) \frac{\Gamma L^{-N}}{k_0 + l_K} P(X(k) = \omega | X(K_1) = \omega^{inf})
\]

\[
+ \frac{\Gamma L^{-N}}{k_0 + l_K} \sum_{\omega' \neq \omega^{inf}} P(X(k) = \omega | X(K_1) = \omega')
\]
Then it is clear that
\[ Q(k, \omega) \leq \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \right) \left( P(X(k) = \omega | X(K_1) = \omega^{\sup}) \Leftrightarrow P(X(k) = \omega | X(K_1) = \omega^{inf}) \right), \]

hence:
\[ \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(0) = \eta') \Leftrightarrow P(X(k) = \omega | X(0) = \eta'')| \leq \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \right) \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(K_1) = \eta') \Leftrightarrow P(X(k) = \omega | X(K_1) = \eta'')| \]
\[ \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \right) \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(K_2) = \eta') \Leftrightarrow P(X(k) = \omega | X(K_2) = \eta'')| \right) \]

Proceeding this way we have the following bound:
\[ \leq \prod_{k=1}^{K^{\sup}(k)} \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(K^{\sup}(k)) = \eta') \Leftrightarrow P(X(k) = \omega | X(K^{\sup}(k)) = \eta'')| \right) \]

and finally since the the possible maximal value of the supremum is 1:
\[ \sup_{\omega, \eta', \eta''} |P(X(k) = \omega | X(0) = \eta') \Leftrightarrow P(X(k) = \omega | X(0) = \eta'')| \leq \prod_{k=1}^{K^{\sup}(k)} \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \right). \]

It is then sufficient to show that
\[ \lim_{m \to \infty} \prod_{k=1}^{m} \left( 1 \Leftrightarrow \frac{\Gamma}{k_0 + l_k} \right) = 0. \]

which is a well known consequence of the divergence of the series
\[ \sum_i (k_0 + l_k)^{-1} \]
for all \( k_0 \) and \( \kappa \). This completes the proof of Lemma A.1.

\textbf{Q.E.D.}

\textbf{Lemma A.2}
\[ \lim_{k_0 \to \infty} \sup_{k \geq k_0} \| P(k, \cdot | k_0, \pi_0) \Leftrightarrow \pi_0 \| = 0. \]

\textbf{Proof of Lemma A.2:}
In the following let \( P_{k_0, k}(\cdot) \) stand for \( P(k, \cdot | k_0, \pi_0) \) so that for any \( k \geq k_0 > 0 \):
\[ P_{k_0, k}(\omega) = \sum_\eta P(X(k) = \omega | X(k_0) = \eta) \pi_0(\eta). \]

First we show that for any \( k > k_0 \geq 0 \):
\[ \| P_{k_0, k} \Leftrightarrow \pi_T(k, C) \| \leq \| P_{k_0, k-1} \Leftrightarrow \pi_T(k, C) \|. \quad (95) \]
Finally, we can prove that

\[ \| P_{k_0, k} \Leftrightarrow \pi_{T(k, C)} \| = \]

\[ \sum_{(\omega_1, \ldots, \omega_N)} \left| \pi_{T(k, C)}(X_{s_1} = \omega_s, s \neq s_1) \right| \left| \pi_{T(k, C)}(X_s = \omega_s) \right| \]

\[ = \sum_{(\omega_2, \ldots, \omega_N)} \left( \sum_{\omega_1 \in \Lambda} \pi_{T(k, C)}(X_{s_1} = \omega_s, s \neq s_1) \right) \left| P_{k_0, k-1}(X_s = \omega_s, s \neq s_1) \right| \]

\[ = \sum_{(\omega_2, \ldots, \omega_N)} \left( \sum_{\omega_1 \in \Lambda} \pi_{T(k, C)}(X_{s_1} = \omega_s, s \in S) \right) \left| P_{k_0, k-1}(X_s = \omega_s, s \in S) \right| \]

\[ \leq \sum_{(\omega_2, \ldots, \omega_N)} \left( \sum_{\omega_1 \in \Lambda} \pi_{T(k, C)}(X_{s_1} = \omega_s, s \in S) \right) \left| P_{k_0, k-1}(X_s = \omega_s, s \in S) \right| \]

\[ = \| P_{k_0, k-1} \Leftrightarrow \pi_{T(k, C)} \|. \]

Secondly, we prove that \( \pi_{T(k, C)} \) converges to \( \pi_0 \) (the uniform distribution on \( \Omega_{opt} \)):

\[ \lim_{k \to \infty} \| \pi_0 \Leftrightarrow \pi_{T(k, C)} \| = 0. \]

To see this, let \( | \Omega_{opt} | \) be the number of globally optimal configurations. Then

\[ \lim_{k \to \infty} \pi_{T(k, C)}(\omega) = \lim_{k \to \infty} \frac{\exp(\Leftrightarrow T(k, C))}{\sum_{\omega' \in \Omega_{opt}} \exp(\Leftrightarrow U(\omega') \cap T(k, C))} = \frac{\exp(\Leftrightarrow U(\omega) \Leftrightarrow T(k, C))}{\sum_{\omega' \in \Omega_{opt}} \exp(\Leftrightarrow U(\omega') \Leftrightarrow T(k, C))} \]

\[ = \begin{cases} 0 & \omega \notin \Omega_{opt} \\ \frac{1}{| \Omega_{opt} |} & \omega \in \Omega_{opt} \end{cases} \]

Finally, we can prove that

\[ \sum_{k=1}^{\infty} \| \pi_{T(k, C)} \Leftrightarrow \pi_{T(k+1, C)} \| < \infty \]

since

\[ \sum_{k=1}^{\infty} \| \pi_{T(k, C)} \Leftrightarrow \pi_{T(k+1, C)} \| = \sum_{\omega} \sum_{k=1}^{\infty} \left| \pi_{T(k, C)}(\omega) \Leftrightarrow \pi_{T(k+1, C)}(\omega) \right| \]

and since

\[ \forall \omega : \pi_{T(k, C)}(\omega) \Leftrightarrow \pi_0(\omega), \]

it is enough to show that \( \pi_T(\omega) \) is monotonous for every \( \omega \). However, it is clear from Equation (96) that

\[ Zoltan KATO, Marc BERTHOD, Josiane ZERUBIA \]
• if $\omega \notin \Omega_{opt}$ then $\pi_T(\omega)$ is strictly increasing for $0 < T \leq \epsilon$ for some sufficiently small $\epsilon \Gamma$

• if $\omega \in \Omega_{opt}$ then $\pi_T(\omega)$ is strictly decreasing for all $T > 0$.

Fix $k > k_0 \geq 0$. From Equations (95) and (97) we obtain:

$$\|P_{k_0,k} \Leftrightarrow \pi_0\| \leq \|P_{k_0,k} \Leftrightarrow \pi_T(k,C)\| + \|\pi_T(k,C) \Leftrightarrow \pi_0\| \leq \|P_{k_0,k-1} \Leftrightarrow \pi_T(k-1,C)\| + \|\pi_T(k-1,C) \Leftrightarrow \pi_0\|$$

$$\leq \|P_{k_0,k-2} \Leftrightarrow \pi_T(k-2,C)\| + \|\pi_T(k-2,C) \Leftrightarrow \pi_T(k-1,C)\| + \|\pi_T(k-1,C) \Leftrightarrow \pi_T(k,C)\| + \|\pi_T(k,C) \Leftrightarrow \pi_0\|$$

$$\leq \cdots \leq \|P_{k_0,k_0} \Leftrightarrow \pi_T(k_0,C)\| + \sum_{i=k_0}^{k-1} \|\pi_T(i,C) \Leftrightarrow \pi_T(i+1,C)\| + \|\pi_T(k,C) \Leftrightarrow \pi_0\|.$$  

On the other hand

$$P_{k_0,k_0} = \pi_0$$

and

$$\lim_{k \to \infty} \|\pi_T(k,C) \Leftrightarrow \pi_0\| = 0.$$  

Then we have

$$\lim_{k \to \infty} \sup_{k_0 \geq k_0} \|P_{k_0,k} \Leftrightarrow \pi_0\| \leq \lim_{k \to \infty} \sup_{k_0 \geq k_0} \sum_{i=k_0}^{k-1} \|\pi_T(i,C) \Leftrightarrow \pi_T(i+1,C)\|$$

$$= \lim_{k \to \infty} \sum_{i=k_0}^{\infty} \|\pi_T(i,C) \Leftrightarrow \pi_T(i+1,C)\| = 0.$$  

The last term is 0 by (97) which completes the proof of Lemma A.1. Q.E.D.

**Theorem 5.5 (Multi-Temperature Annealing)** Assume that there exists an integer $\kappa \geq N$ such that for every $k = 0, 1, 2, \ldots$ \( S \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\} \). Let $T(k,C)$ be any decreasing sequence of temperatures for which

1. There exists $T_{k}^{\inf}$ such that for all $C \in C$: $T_{k}^{\inf} \leq T(k,C)$.

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

3. For all $k \geq k_0$, for some integer $k_0 \geq 2$: $T_{k}^{\inf} \geq N \Delta/ \log k$.

Then for any starting configuration $\eta \in \Omega$ and for every $\omega \in \Omega$:

$$\lim_{k \to \infty} P(X(k) = \omega | X(0) = \eta) = \pi_0(\omega).$$  (98)
Proof of the theorem:  
Using the above mentioned lemmas, we can easily prove the annealing theorem:

\[ \lim_{k \to \infty} \| P(X(k) = \cdot | X(0) = \eta) \otimes \pi_0 \| = \lim_{k_0 \to \infty} \lim_{k \to \infty} \| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) \otimes \pi_0 \| \]

\[ \leq \lim_{k_0 \to \infty} \lim_{k \to \infty} \| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) \otimes P(k, \cdot | k_0, \pi_0) \| \]

\[ + \lim_{k_0 \to \infty} \lim_{k \to \infty} \| P(k, \cdot | k_0, \pi_0) \otimes \pi_0 \| . \]

The last term is 0 by Lemma A.2. Moreover, \( P(k_0, \cdot | 0, \eta) \) and \( \pi_0 \) have total mass 1; thus:

\[ \| \sum_{\eta'} P(k, \cdot | k_0, \eta') P(k_0, \eta' | 0, \eta) \otimes P(k, \cdot | k_0, \pi_0) \| \]

\[ = \sum_{\omega} \sup_{\eta''} \left\| \sum_{\eta'} \left( P(k, \omega | k_0, \eta') \Rightarrow P(k, \omega | k_0, \eta'') \right) \left( P(k_0, \eta' | 0, \eta) \Rightarrow \pi_0 (\eta') \right) \right\| \]

\[ \leq 2 \sum_{\omega} \sup_{\eta'', \eta'} \left\| P(k, \omega | k_0, \eta') \Rightarrow P(k, \omega | k_0, \eta'') \right\|. \]

Finally:

\[ \lim_{k \to \infty} \| P(X(k) = \cdot | X(0) = \eta) \otimes \pi_0 \| \]

\[ \leq 2 \sum_{\omega} \lim_{k_0 \to \infty} \lim_{k \to \infty} \sup_{\eta', \eta''} \left\| P(k, \omega | k_0, \eta') \Rightarrow P(k, \omega | k_0, \eta'') \right\| = 0 \]

The last term is 0 by Lemma A.1 which completes the proof of the annealing theorem. Q.E.D.
B Images
Figure 16: Results on a synthetic image with 2 classes.
Figure 17: Results on a synthetic image with 4 classes.
Figure 18: Results on a synthetic image with 16 classes.
Figure 19: Results on a SPOT image with 4 classes.
Figure 20: Results on an indoor scene with \( \frac{1}{4} \) classes.
**Figure 21:** Results on a medical image with 4 classes.
Figure 22: Original SPOT image with 6 classes.
Figure 23: Ground truth data.
Figure 24: Results of the ICM algorithm. Comparison with ground truth data.
Figure 25: Results of the Gibbs Sampler. Comparison with ground truth data.
Tables
<table>
<thead>
<tr>
<th>original</th>
<th>levels</th>
<th>VPR</th>
<th>$T_0$</th>
<th>iter.</th>
<th>total time</th>
<th>time/iter.</th>
<th>error</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>62</td>
<td>1.91 sec.</td>
<td>0.03 sec.</td>
<td>260 (1.59%)</td>
<td>0.9</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>0.77 sec.</td>
<td>0.009 sec.</td>
<td>1547 (9.44%)</td>
<td>0.9</td>
<td>—</td>
</tr>
<tr>
<td>multiscale</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>1.2</td>
<td>4</td>
<td>136</td>
<td>3.25 sec.</td>
<td>0.02 sec.</td>
<td>236 (1.44%)</td>
<td>0.7</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>1.2</td>
<td>1</td>
<td>18</td>
<td>0.14 sec.</td>
<td>0.008 sec.</td>
<td>465 (2.83%)</td>
<td>0.7</td>
<td>—</td>
</tr>
<tr>
<td>hierarchical</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>4</td>
<td>4,3,2,1</td>
<td>23</td>
<td>50.1 sec.</td>
<td>2.18 sec.</td>
<td>115 (0.7%)</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>16.6 sec.</td>
<td>1.5 sec.</td>
<td>300 (1.83%)</td>
<td>0.7</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Table 1:** Results on a noisy synthetic image with 2 classes

<table>
<thead>
<tr>
<th>original</th>
<th>levels</th>
<th>VPR</th>
<th>$T_0$</th>
<th>iter.</th>
<th>total time</th>
<th>time/iter.</th>
<th>error</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>68</td>
<td>3.01 sec.</td>
<td>0.04 sec.</td>
<td>183 (1.12%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>0.15 sec.</td>
<td>0.02 sec.</td>
<td>2948 (17.99%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>multiscale</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>1.2</td>
<td>4</td>
<td>101</td>
<td>3.85 sec.</td>
<td>0.04 sec.</td>
<td>136 (1.07%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>1.2</td>
<td>1</td>
<td>17</td>
<td>0.22 sec.</td>
<td>0.01 sec.</td>
<td>1657 (10.11%)</td>
<td>0.9</td>
<td>—</td>
</tr>
<tr>
<td>hierarchical</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>4</td>
<td>4,3,2,1</td>
<td>41</td>
<td>141.97 sec.</td>
<td>3.46 sec.</td>
<td>191 (1.16%)</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>11</td>
<td>30.17 sec.</td>
<td>2.74 sec.</td>
<td>293 (1.78%)</td>
<td>0.8</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 2:** Results on a noisy synthetic image with 4 classes

<table>
<thead>
<tr>
<th>original</th>
<th>levels</th>
<th>VPR</th>
<th>$T_0$</th>
<th>iter.</th>
<th>total time</th>
<th>time/iter.</th>
<th>error</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>201</td>
<td>22.96 sec.</td>
<td>0.12 sec.</td>
<td>340 (2.08%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>0.55 sec.</td>
<td>0.05 sec.</td>
<td>8721 (53.22%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>multiscale</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>1.2</td>
<td>4</td>
<td>337</td>
<td>32.9 sec.</td>
<td>0.1 sec.</td>
<td>331 (2.02%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>1.2</td>
<td>1</td>
<td>15</td>
<td>0.68 sec.</td>
<td>0.05 sec.</td>
<td>5198 (31.72%)</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>hierarchical</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>4</td>
<td>4,3,2,1</td>
<td>107</td>
<td>1169.46 sec.</td>
<td>10.93 sec.</td>
<td>316 (1.93%)</td>
<td>1.0</td>
<td>0.2</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>162.87 sec.</td>
<td>10.18 sec.</td>
<td>795 (4.85%)</td>
<td>1.0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table 3:** Results on a noisy synthetic image with 16 classes
### Table 4: Results on a SPOT image with 4 classes

<table>
<thead>
<tr>
<th></th>
<th>levels</th>
<th>VPR</th>
<th>$T_0$</th>
<th>iter.</th>
<th>total time</th>
<th>time/iter.</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>64</td>
<td>9.06 sec.</td>
<td>0.14 sec.</td>
<td>2.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>7</td>
<td>0.33 sec.</td>
<td>0.047 sec.</td>
<td>2.0</td>
<td>—</td>
</tr>
<tr>
<td>multiscale</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>1-8</td>
<td>4</td>
<td>106</td>
<td>10.22 sec.</td>
<td>0.09 sec.</td>
<td>2.0</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>1-8</td>
<td>1</td>
<td>37</td>
<td>1.14 sec.</td>
<td>0.03 sec.</td>
<td>1.0</td>
<td>—</td>
</tr>
<tr>
<td>hierarchical</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>4</td>
<td>16</td>
<td>4,3,2,1</td>
<td>29</td>
<td>353.54 sec.</td>
<td>12.19 sec.</td>
<td>2.0</td>
<td>0.4</td>
</tr>
<tr>
<td>ICM</td>
<td>4</td>
<td>16</td>
<td>1</td>
<td>6</td>
<td>58.59 sec.</td>
<td>9.76 sec.</td>
<td>0.5</td>
<td>0.6</td>
</tr>
</tbody>
</table>

### Table 5: Results on the SPOT image with 6 classes

<table>
<thead>
<tr>
<th></th>
<th>levels</th>
<th>VPR</th>
<th>$T_0$</th>
<th>iter.</th>
<th>total time</th>
<th>time/iter.</th>
<th>$\beta$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>32</td>
<td>4</td>
<td>234</td>
<td>168.18 sec.</td>
<td>0.69 sec.</td>
<td>1.5</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>1</td>
<td>32</td>
<td>1</td>
<td>8</td>
<td>2.03 sec.</td>
<td>0.25 sec.</td>
<td>1.5</td>
<td>—</td>
</tr>
<tr>
<td>multiscale</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>5</td>
<td>1-32</td>
<td>4</td>
<td>580</td>
<td>180.17 sec.</td>
<td>0.31 sec.</td>
<td>1.5</td>
<td>—</td>
</tr>
<tr>
<td>ICM</td>
<td>5</td>
<td>1-32</td>
<td>1</td>
<td>36</td>
<td>5.15 sec.</td>
<td>0.14 sec.</td>
<td>0.3</td>
<td>—</td>
</tr>
<tr>
<td>hierarchical</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Gibbs</td>
<td>5</td>
<td>64</td>
<td>4,3,2,1</td>
<td>154</td>
<td>9629.33 sec.</td>
<td>62.53 sec.</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>ICM</td>
<td>5</td>
<td>64</td>
<td>1</td>
<td>16</td>
<td>915.99 sec.</td>
<td>57.25 sec.</td>
<td>1.0</td>
<td>0.2</td>
</tr>
</tbody>
</table>
References


In Theorem 5.5 (pages 22 and 33) there is a problem: The conditions of the theorem permits to choose temperatures such that the original optimization criteria changes. The following example illustrates this problem:

Example 1. Consider the cost function:

\[ U(X) = (X_1 - 9)^2 + (X_1 - X_2)^2 + X_2^2 \]  

( xcix) 

where \( X = (X_1, X_2) \) is a MRF with cliques \( \{X_1\}, \{X_2\} \) and \( \{X_1, X_2\} \). The state space is simply \( \{0, \ldots, 9\} \).

Using the Theorem 5.5, these cliques may have the following temperature functions:

\[ T(k, \{X_1\}) = \frac{300000\Delta}{\log k}, \quad T(k, \{X_2\}) = \frac{3\Delta}{\log k} \quad \text{and} \quad T(k, \{X_1, X_2\}) = \frac{3\Delta}{\log k^2} \]

yielding the following energy function:

\[ U_{MTA}(X, k) = (10^{-6}(X_1 - 9)^2 + (X_1 - X_2)^2 + X_2^2) \frac{\log k}{3\Delta} \]

which is nothing else but a conventional annealing with the modified energy function:

\[ U'(X) = 10^{-6}(X_1 - 9)^2 + (X_1 - X_2)^2 + X_2^2 \]

However, the minimum of \( U(X) \) is \( X = (6, 3) \) whereas the minimum of \( U'(X) \) is \( X = (0, 0) \).

This mistake is due to the decomposition of \( U(\omega) \otimes T(k, C) \) defined in Equation (68) (page 21). Let \( \omega' \in \Omega_{opt} \) be a globally optimal configuration. Thus \( U(\omega') \otimes U^{inj} \) equals to 0 \( (U^{inj} \) has been defined in Equation (74) in page 21). In the case of a classical annealing dividing by a constant temperature does not change this relation (obviously \( \forall k : (U(\omega') \otimes U^{inj}) / T(k) \) is still 0). But it is not necessarily true that \( (U(\omega') \otimes U^{inj}) \otimes T(k, C) \) is also 0! Because choosing sufficiently small temperatures for the cliques where \( \omega'_C \) is locally not optimal (i.e. strengthening the non-optimal cliques) and choosing sufficiently high temperatures for the cliques where \( \omega'_C \) is locally optimal (i.e. weakening the optimal cliques) we obtain \( (U(\omega') \otimes U^{inj}) \otimes T(k, C) > 0 \) meaning that \( \omega' \) is no longer globally optimal.

Before modifying the theorem let us introduce some useful notations. First let us examine the decomposition over the cliques of \( U(\omega) \otimes U(\eta) \) for arbitrary \( \omega \) and \( \eta \Gamma \omega \neq \eta \):

\[ U(\omega) \otimes U(\eta) = \sum_{C \in \mathcal{C}} (V_C(\omega) \otimes V_C(\eta)) \]

Indeed there may be negative and positive members in the decomposition. According to this fact we get:

\[ \sum_{C \in \mathcal{C}} (V_C(\omega) \otimes V_C(\eta)) = \sum_{C \in \mathcal{C}} (V_C(\omega) \otimes V_C(\eta)) + \sum_{C \in \mathcal{C} \setminus \{C\}} (V_C(\omega) \otimes V_C(\eta)) + \sum_{C \in \mathcal{C} \setminus \{C\}} (V_C(\omega) \otimes V_C(\eta)) \]

We would like to thank to the second reviewer of the paper we have submitted to CVGIP — Graphical Models and Image Processing for pointing out a lack in the theorem. We present here his counterexample.
Now let us examine $\Delta$ defined in Equation (74) (page 21):

$$\Delta = U^{\sup} \leftrightarrow U^{\inf}$$

If we want to decompose $\Delta$ as defined above we have to choose some configuration $\omega'$ with a maximum energy (i.e. $U(\omega') = U^{\sup}$) and another configuration $\omega''$ with a minimum energy (i.e. $U(\omega'') = U^{\inf}$). Obviously there may be more than one decomposition depending on the number of globally optimal configurations $|\Omega^{\text{opt}}|$ and the number of configurations with maximal global energy $|\Omega^{\sup}|$. Thus the decomposition of $\Delta$ for a given $(\omega', \omega'')$ is of the following form:

$$\Delta = \Sigma^- (\omega', \omega'') + \Sigma^+ (\omega', \omega'')$$

Furthermore let us define $\Sigma^+_\Delta$ as:

$$\Sigma^+_\Delta = \min_{\omega' \in \Omega^{\sup}, \omega'' \in \Omega^{\text{opt}}} \Sigma^+ (\omega', \omega'').$$

Obviously $\Delta \leq \Sigma^+_\Delta$.

Theorem 5.5 is of the following form:

**Theorem 5.5 (Multi-Temperature Annealing)** Assume that there exists an integer $\kappa \geq N$ such that for every $k = 0, 1, 2, \ldots, S \subseteq \{n_{k+1}, n_{k+2}, \ldots, n_{k+\kappa}\}$. For all $C \in \mathcal{C}$, let $T(k, C)$ be any sequence of temperatures decreasing in $k$ for which

1. $\lim_{k \to \infty} T(k, C) = 0$.
   Let us denote respectively by $T^{\inf}_k$ and $T^{\sup}_k$ the maximum and minimum of the temperature function at $k$ ($\forall C \in \mathcal{C}; T^{\inf}_k \leq T(k, C) \leq T^{\sup}_k$).

2. For all $k \geq k_0$, for some integer $k_0 \geq 2$: $T^{\inf}_k \geq N \Sigma^+_\Delta / \log k$.

3. If $\Sigma^- (\omega, \omega') \neq 0$ for some $\omega \in \Omega \setminus \Omega^{\text{opt}}$, $\omega' \in \Omega^{\text{opt}}$ then a further condition must be imposed:
   For all $k$: $\frac{U^{\sup}}{T^{\inf}_k} \leq R$ with
   $$R = \min_{\omega \in \Omega \setminus \Omega^{\text{opt}}, \omega' \in \Omega^{\text{opt}} \atop \Sigma^- (\omega, \omega') \neq 0} \frac{U(\omega) \leftrightarrow U^{\inf}}{\Sigma^+ (\omega', \omega')}$$

Then for any starting configuration $\eta \in \Omega$ and for every $\omega \in \Omega$:

$$\lim_{k \to \infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega).$$

**Remarks:**

1. In practice we cannot determine $R$ and $\Sigma^+_\Delta$ as we cannot compute $\Delta$ neither.
Considering $\Sigma^+_\Delta$ in condition 2\(\Gamma\) we have the same problem as in the case of a classical annealing (the only difference is that in a classical annealing we have $\Delta$ instead of $\Sigma^+_\Delta$). Consequently the same solutions may be used (an exponential schedule with a sufficiently high initial temperature).

The factor $R$ is more interesting. We propose herein two possibilities which can be used for practical implementations of the method: Either we choose a sufficiently small interval $[T_0^{\text{inf}}, T_0^{\text{sup}}]$ and suppose that it satisfies the condition 3 (we have used this technique in the simulations described in the report) or we use a more strict but easily verifiable condition instead of condition 3, namely:

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

What happens if $\Sigma^- (\omega, \omega')$ is zero for all $\omega$ and $\omega'$ in condition 3 and thus $R$ is not defined? This is the best case because it means that all globally optimal configurations are also locally optimal. That is we have no restriction on the interval $[T_k^{\text{inf}}, T_k^{\text{sup}}]$ thus any local temperature schedule satisfying conditions 1–2 is good.

We have done a computer simulation with the counterexample using the modified theorem:

---

**Example 2.** Since the energy function in Equation (xcix) is simple, we can compute it for each possible configuration (cf. Table 6). We obtain $R = 0.042553$, $\Delta = 216$ and $\Sigma^+_\Delta = 216$. Now, let us define a temperature schedule according to the modified theorem:

$$T(k, \{X_1\}) = \frac{1 \cdot 216 \cdot N}{\log k}, \quad T(k, \{X_2\}) = \frac{1.04 \cdot 216 \cdot N}{\log k} \quad \text{and} \quad T(k, \{X_1, X_2\}) = \frac{1.02 \cdot 216 \cdot N}{\log k}.$$ 

The modified energy function is of the following form:

$$U_{MTA}(X, k) = \left( (X_1 - 9)^2 + \frac{(X_1 - X_2)^2}{1.02} + \frac{X_2^2}{1.04} \right) \frac{\log k}{216 \cdot N},$$

which is equivalent to the following energy function using classical annealing:

$$U'(X) = (X_1 - 9)^2 + \frac{(X_1 - X_2)^2}{1.02} + \frac{X_2^2}{1.04}.$$ 

The values of $U'$ have also been computed as shown in Table 7. Clearly, we obtain the same minimum as for the original function $U$, that is $X = (6, 3)$.

Now let us go back to the proof of the theorem. The following modifications were needed:

1. The lower bound for $\delta(k)$ given at the beginning of the proof of Lemma A.1 (page 29) is changed:

$$\delta(k) \geq \frac{\exp(\varepsilon \Delta \sigma \cap T(k, C))}{L \exp(\varepsilon \Delta \sigma \cap T(k, C))} = \frac{\exp(\varepsilon \Delta \sigma \cap T(k, C))}{L},$$

as before but since $\Delta \leq \Sigma^+_\Delta$ and $\Sigma^+_\Delta > 0$:

$$\geq \frac{1}{L} \exp(\varepsilon \Sigma^+_\Delta \cap T(k, C)) \geq \frac{1}{L} \exp(\varepsilon \Sigma^+_\Delta / T_k^{\text{inf}}).$$
Table 6: Original energy function $U$.

$$
\begin{array}{cccccccccc}
X_1 \setminus X_2 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 81 & 83 & 89 & 99 & 113 & 131 & 153 & 179 & 209 & 243 \\
1 & 68 & 65 & 69 & 77 & 89 & 105 & 125 & 149 & 177 & 209 \\
2 & 55 & 51 & 53 & 59 & 69 & 83 & 101 & 123 & 149 & 179 \\
3 & 41 & 41 & 41 & 41 & 41 & 41 & 41 & 41 & 41 & 41 \\
4 & 33 & 33 & 33 & 33 & 33 & 33 & 33 & 33 & 33 & 33 \\
6 & 17 & 17 & 17 & 17 & 17 & 17 & 17 & 17 & 17 & 17 \\
7 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 \\
8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
$$

Table 7: Modified energy function $U'$.

$$
\begin{array}{cccccccccc}
X_1 \setminus X_2 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 81 & 82 & 88 & 98 & 112 & 120 & 130 & 146 & 160 & 176 \\
1 & 64 & 64 & 68 & 76 & 88 & 103 & 123 & 146 & 173 & 204 \\
2 & 52 & 52 & 52 & 58 & 68 & 81 & 99 & 120 & 145 & 174 \\
3 & 44 & 44 & 44 & 44 & 44 & 44 & 44 & 44 & 44 & 44 \\
4 & 36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 & 36 \\
5 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\
6 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 & 20 \\
7 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\
8 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
$$

2. Consequently, the lower bound at the end of Equation (93) (page 29) has the following form:

$$
L^{-N} \exp \left( \frac{-N \sum_{C}^{+}}{T_{\text{inf}}^{+} T_{k+i\epsilon}} \right)
$$

3. After Equation (96) (page 32) the following paragraph has to be added: Equation (96) is true if $(U(\omega) \Leftrightarrow U^{\text{inf}}) \cap T(k, C) \geq 0$. Let us rewrite this inequality as

$$
\sum_{C \in \mathcal{C}} V_C(\omega) \Leftrightarrow V_C(\omega') = T(k, C) \geq 0
$$

where $\omega'$ is any globally optimal configuration (i.e., $\omega' \in \Omega_{\text{opt}}$). While $V_C(\omega) \Leftrightarrow V_C(\omega')$ may be negative $\Gamma U(\omega) \Leftrightarrow U^{\text{inf}}$ is always positive or zero (since $U^{\text{inf}}$ is the energy minimum). We denote by $\Sigma(\omega)$ the energy difference in Equation (ci) without the temperature. Obviously, it is non-negative:

$$
\Sigma(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega) \Leftrightarrow V_C(\omega') = U(\omega) \Leftrightarrow U^{\text{inf}} \geq 0
$$

Then let us decompose $\Sigma(\omega)$ according to Equation (c):

$$
\Sigma(\omega) = \Sigma^{+}(\omega, \omega') + \Sigma^{-}(\omega, \omega').
$$
From which: \[ \Sigma^+ (\omega, \omega') = \Sigma (\omega) \Leftrightarrow \Sigma^- (\omega, \omega'). \]

Now let us consider Equation (ci):

\[ \sum_{C \in \mathcal{C}} \frac{V_C(\omega) \Leftrightarrow V_C(\omega')}{T(k, C)} = \Sigma^- (\omega, \omega') \otimes T(k, C') + \Sigma^+ (\omega, \omega') \otimes T(k, C) \]

\[ \geq \Sigma^- (\omega, \omega') / T_k^{\mathrm{inj}} + \Sigma^+ (\omega, \omega') / T_k^{\mathrm{sup}} = \frac{\Sigma^- (\omega, \omega') \cdot T_k^{\mathrm{sup}} + \Sigma^+ (\omega, \omega') \cdot T_k^{\mathrm{inj}}}{T_k^{\mathrm{inj}} T_k^{\mathrm{sup}}} \geq 0 \]

Furthermore:

\[ \Sigma^- (\omega, \omega') \cdot T_k^{\mathrm{sup}} + \Sigma^+ (\omega, \omega') \cdot T_k^{\mathrm{inj}} = \Sigma^- (\omega, \omega') \cdot T_k^{\mathrm{sup}} + (\Sigma (\omega) \Leftrightarrow \Sigma^- (\omega, \omega')) T_k^{\mathrm{inj}} \]

Therefore:

\[ \Sigma^- (\omega, \omega') (T_k^{\mathrm{sup}} \Leftrightarrow T_k^{\mathrm{inj}}) \Leftrightarrow \Sigma (\omega) \cdot T_k^{\mathrm{inj}} \geq 0 \]

Dividing by \( \Sigma^- (\omega, \omega') \) which is negative we get:

\[ T_k^{\mathrm{sup}} \Leftrightarrow T_k^{\mathrm{inj}} \leq \frac{\Sigma (\omega)}{\Sigma^- (\omega, \omega')} T_k^{\mathrm{inj}} \]

Which is true due to condition 3 of the theorem.