ELSEVIER

Image and Vision Computing 14 (1996) 285–295

Bayesian image classification using Markov random fields

Marc Berthod, Zoltan Kato, Shan Yu*, Josiane Zerubia

*INRIA, BP-93, 06902 Sophia Antipolis Cedex, France**

Received 10 May 1995; revised 13 October 1995

Abstract

In this paper, we present three optimisation techniques, Deterministic Pseudo-Annealing (DPA), Game Strategy Approach (GSA), and Modified Metropolis Dynamics (MMD), in order to carry out image classification using a Markov random field model. For the first approach (DPA), the a posteriori probability of a tentative labelling is generalised to a continuous labelling. The merit function thus defined has the same maxima under constraints yielding probability vectors. Changing these constraints convexifies the merit function. The algorithm solves this unambiguous maximisation problem, and then tracks down the solution while the original constraints are restored yielding a good, even if suboptimal, solution to the original labelling assignment problem. In the second method (GSA), the maximisation problem of the a posteriori probability of the labelling is solved by an optimisation algorithm based on game theory. A non-cooperative *n*-person game with pure strategies is designed such that the set of Nash equilibrium points of the game is identical to the set of local maxima of the a posteriori probability of the labelling. The algorithm converges to a Nash equilibrium. The third method (MMD) is a modified version of the Metropolis algorithm: at each iteration the new state is chosen randomly, but the decision to accept it is purely deterministic. This is also a suboptimal technique but it is much faster than stochastic relaxation. These three methods have been implemented on a Connection Machine CM2. Experimental results are compared to those obtained by the Metropolis algorithm, the Gibbs sampler and ICM (Iterated Conditional Mode).

Keywords: Bayesian image classification; Markov random fields; Optimisation

1. Introduction

Markov random fields (MRFs) have become more and more popular during the last few years in image processing. A good reason for this is that such a modelisation is the one which requires the least a priori information on the world model. In fact, the simplest statistical model for an image consists of the probabilities of classes, or grey levels, for isolated pixels. The knowledge of the dependencies between nearby pixels is much more powerful, and imposes few constraints. In a way, it is difficult to conceive of a more general model, even if it is not easy to determine the values of the parameters which specify a MRF. Another good reason is of course the Hammersley–Clifford theorem, reported, for example, in Ref. [1], which considerably easied, as stressed in Ref. [2], the determination of these parameters by allowing specification of the model either by conditional or joint probabilities.

Many standard image processing problems, such as image classification, can thus be expressed quite naturally as combinatorial optimisation ones. Direct optimisation is not tractable even in the smallest cases. Many heuristics have been proposed to solve them: Iterated Conditional Modes [3,4], Graduated Non-Convexity (GNC) [5,6], Mean Field Annealing [7,8]. Simulated Annealing [1,9,10], Dynamic Programming [11], etc.

We present here three different optimisation techniques. The first approach, which we propose to call Deterministic Pseudo Annealing (DPA), is related to relaxation labelling, a quite popular framework for a variety of computer vision problems [12–14]. The basic idea is to introduce weighted labellings which assign a weighted combination of labels to any object (or site) to be labeled, and then to build a merit function of all the weighted labels in such a way that this merit function takes the values of the probability of a global assignment of labels (up to a monotonic transform) for any weighted labelling, which assigns the value 1 to one label and 0 to the others at any site. Besides, these values are the only extrema of this function, under suitable constraints. DPA consists of changing the constraints so as to

* Email: ⟨name⟩@sophia.inria.fr

SSDI 0262-8856(95)01072-6

convexify this function, find its unique global maximum, and then track down the solution, by a continuation method, until the original constraints are restored, and a discrete labelling can be obtained. The proposed algorithm is new, and departs significantly from GNC and MFA, though it has been developed in the same spirit.

The second method, called the Game Strategy Approach (GSA), uses a game theory approach to maximise the a posteriori probability of the labelling. Game theory has been developing since the 1930s. Recently, it has been applied to computer vision problems [15–18]. Here we are interested in a special branch of the theory, known as non-cooperative n-person game theory [19]. The basic idea is to consider the pixels as the players and the labels as the strategies. The maximisation of the payoff function of the game corresponds to the maximisation of the a posteriori probability of the labelling.

The third approach, Modified Metropolis Dynamics (MMD), is a modified version of the Metropolis algorithm [10]: for each iteration, a global state is chosen at random (with a uniform distribution), and for each site, the decision on accepting the new state is deterministic.

The goal of this paper is to evaluate the performance of these three methods. They have been implemented on a Connection Machine CM2. Their performances for image classification are compared to ICM [3,4], a well-known fast, deterministic relaxation scheme, the Metropolis algorithm [10], and Gibbs sampler [1]. The last two are classical fully-stochastic relaxation techniques.

The paper is organized as follows. Firstly, we present the image model using the Markov random field formulation. Secondly, we describe each of the three algorithms. Lastly, we compare the performance of these algorithms, as well as that of ICM, the Metropolis algorithm and the Gibbs Sampler, through their application to image classification.

2. Probabilistic modelisation

In this article, we are interested in the following general problem: we are given a set of units (or sites) $\mathcal{S} = \{S_i,\ 1 \leq i \leq N\}$, and a set of possible labels $\Lambda = \{1, 2, \dots, M\}$. Each unit can take any label from 1 to M. We are also given an MRF on these units, defined by a graph G (where the vertices represent the units, and the edges represent the label constraints of the neighbouring units), and the so-called 'clique potentials.' Let c denote a clique of G, and $\mathcal{C}$ the set of all cliques of G, and $\mathcal{C}_i = \{c : S_i \in c\}$. The number of sites in the clique is its degree: $\deg(c)$, and $\deg(G) = \max_{c \in \mathcal{C}} \deg(c)$.

A global discrete labelling L assigns one label L_i $(1 \leq L_i \leq M)$ to each site S_i in $\mathcal{S}$. The restriction of L to the sites of a given clique c is denoted by L_c. The definition of the MRF is completed by the knowledge of the clique potentials V_{cL} (shorthand for V_{cL_c}) for every c in $\mathcal{C}$ and every L in $\mathcal{L}$, where $\mathcal{L}$ is the set of the M^N discrete labellings (recall that M is the number of possible labels, which is assumed to be the same for any site for simplicity, and N is the number of sites).

The nice result of Hammersley–Clifford is that the probability of a given labelling L may be computed quite easily (assuming deg $(\mathcal{C})$ is small, at most 2 or 3) by:

$$P(L) = \frac{\prod_{c \in \mathcal{C}} \exp(-V_{cL})}{Z}, \tag{1}$$

where Z, the partition function, is a normalising factor such that:

$$\sum_{L \in \mathcal{L}} P(L) = 1.$$

We assume here that the sufficient positivity condition $P(L) > 0$ is met.

The basic problem, for most applications, is to find the labelling L which maximises $P(L)$, knowing that exhaustive search of all the labellings is strictly intractable.

Before explaining the proposed methods, it is necessary to give more detail about how Bayesian modelling behaves with Markov random fields. First, it is important to notice that for most applications, the information available stems from two different sources: a priori knowledge about the restrictions that are imposed on the simultaneous labelling of connected neighbour units; and observations on these units for a given occurrence of the problem.

The first source is generic, and is typically referred to as the 'world model.' For example, discrete relaxation relies on the knowledge of allowed couples, or n-tuples of labels between neighbouring sites. This type of knowledge may be more detailed, and reflect statistical dependencies between the labels of neighbouring sites, thus defining a Markov random field. For example, when dealing with images, the knowledge of the likelihood of configurations of nearby pixels may take the form of an MRF with cliques of order 1 to 2 (4-connectivity), or order 1 to 4 (8-connectivity). The other source of information consists of the observations. Combining these two sources of information may be achieved in different ways; Bayesian modelling, whether strictly applied or not, comes in very naturally at this stage. Let us assume, for simplicity, that the observations consist of the grey levels (or any other scalar or vector quantities) of the pixels in an image: y_i is thus the grey level for site S_i, and $Y = (y_1, \dots y_N)^t$ here represents the observed image. A very general problem is to find, given for example a first-order MRF on these pixels (i.e. knowing the statistics of couples of labels with 4-connectivity), the labelling L which maximises $P(L/Y)$. Bayes' theorem tells us that $P(L/Y) = P(Y/L)P(L)/P(Y)$. Actually, $P(Y)$ does not depend on the labelling L, and plays exactly the same role as Z in Eq. (1), as we are not concerned with testing the likelihood of the MRF. We now have to assume that we are able to model the noise process. Standard assumptions, which roughly amount to white invariant noise, are that:

$$P(Y/L) = \prod_{i=1}^{N} P(y_i/L) = \prod_{i=1}^{N} P(y_i/L_i). \qquad (2)$$

The term $P(L)$ is taken care of by the MRF modelling the a priori world model, as in Eq. (1). It is then easy to see that the a posteriori probability, which we are trying to maximise, is given by:

$$P(L/Y) \propto \prod_{i=1}^{N} P(y_i/L_i) \prod_{c\in\mathscr{C}} \exp(-V_{cL}). \qquad (3)$$

It is obvious from this expression that the a posteriori probability also derives from a Markov random field, with cliques of order 1 and 2 (and not only 2 as for the a priori probability). The energies of cliques of order 1 directly reflect the probabilistic modelling of labels without context, which would be used for classifying or labelling the pixels independently. This equivalence was used in Refs. [20,21] for initialising a continuous labelling before relaxation. It is easy to prove that it is always possible, by suitable shifts on the clique potentials, to keep only the potentials of maximal cliques. The procedure to do so directly derives from the proof of the Hammersley–Clifford theorem given in Ref. [22]. The problem at hand is thus strictly equivalent to maximising:

$$f(L) = \sum_{c\in\mathscr{C}} W_{cL}, \qquad (4)$$

where $W_{cL} = -V_{cL}$ and L is the corresponding labelling. As we shall see, this transformation may not be really necessary, but it will simplify some results. The following property is more interesting: shifting every clique potential of a given clique by the same quantity is equivalent to scaling all the $P(L)$'s by a given factor, or equivalently to changing the normalisation factor Z of Eq. (1). Thus, the maximisation problem is not changed. For example, it will be possible to shift all the W_i's so that they all become non-negative, or even positive, without changing the solution to the problem.

3. Deterministic pseudo-annealing

Several approaches have been proposed to find at least a reasonably good labelling. One of the best known is probably simulated annealing [1]. But other, more 'algorithmic' approaches are worth mentioning: Iterated Conditional Modes (ICM) [3] or Dynamic Programming [11], for example. Even former work on relaxation labelling, already mentioned, was (knowingly or not) a way to tackle this problem. We start from one such example [21]. The key point here is to cast this discrete, combinatorial, optimisation problem into a more comfortable maximisation problem on a compact subset of $\mathscr{R}^N$. Let us define a real function $f(X)$ ($X \in \mathscr{R}^{NM}$) as follows:

$$f(X) = \sum_{c\in\mathscr{C}} \sum_{l_c\in L_c} W_{cl_c} \prod_{j=1}^{\deg(c)} x_{c_j,l_{c_j}}, \qquad (5)$$

where c_j denotes the jth site of clique c, and l_{c_j} the label assigned to this site by l_c. It is clear from Eq. (5) that f is a polynomial in the $x_{i,k}$'s; the maximum degree of f is the maximum degree of the cliques. If we assume for simplicity that all the cliques have the same degree d (this is true with 4-neighbourhoods on images, after suitable shifts on the coefficients), then f is a homogeneous polynomial of degree d. This is by no means necessary in what follows, but will alleviate the notations.

Moreover, it is clear that f is linear with respect to any $x_{i,k}$ (where i refers to a site, and k to a label to be attached to the site). Let us now restrict X to $\mathscr{P}_{NM}$, a specific compact subset of $\mathscr{R}_{NM}$ defined by the following constraints:

$$\forall i,k : x_{i,k} \geq 0, \qquad (6)$$

$$\forall i : \sum_{k=1}^{M} x_{i,k} = 1. \qquad (7)$$

These constraints simply mean that x is a probabilistic labelling. It admits many maxima on $\mathscr{P}_{NM}$, but the absolute maximum X^* is on the border:

$$\forall i, \exists k : x^*_{ik} = 1, \quad l \neq k \Rightarrow x^*_{il} = 0. \qquad (8)$$

It directly yields a solution to our problem. The difficulty is, of course, that f is not concave but convex with a tremendous number of such maxima, and that any standard gradient technique will usually lead to a local maximum (all located on the border of the domain), and not to the absolute maximum. It is thus vital to find a good starting point before applying such a technique.

The basic idea in DPA is to temporarily change the subset on which f is maximised so that f becomes concave, to maximise f, and to track this maximum while slowly changing the constraints until the original ones are restored so that a discrete labelling can be deduced.

First, when $c = 2$ (cliques of order 2), f is a quadratic form, and can always be written as $f = X^tAX$, where A is an $NM * NM$ symmetric matrix. Besides, after suitable shift, A has non-negative entries. After Perron-Frobenius [23], A has a unique real non-negative eigenvector, which is strictly positive, the corresponding eigenvalue being positive and equal to the spectral radius; besides any other eigenvalue has a smaller modulus. Actually, this is a generic case, which admits degeneracies, but they can be dealt with easily and are not considered here (see Ref. [20] for more details). This eigenvector maximises f under constraints different from the preceding ones:

$$\forall i,k : x_{i,k} \geq 0, \tag{9}$$

$$\forall i : \sum_{k=1}^{M} x_{i,k}^2 = 1. \tag{10}$$

We call $Q^{NM,d}$ the compact subset of $\mathscr{R}^{NM}$ so defined.

It must be well understood that these constraints have been chosen only to make f concave, which makes it easy to optimise. On the other hand, the $x_{i,k}$'s can no longer be interpreted as a probabilistic labelling. The vector X can be obtained very efficiently by using (for example) the iterative power method: start from any X^0, and apply

$$X^{n+1} = \frac{AX^n}{\|AX^n\|_{L^2}} \propto AX^n. \tag{11}$$

A fundamental point is the following: if f is a polynomial with non-negative coefficients and maximum degree d, then f has a unique maximum on $Q^{NM,d}$ (with, again, possible degeneracies as mentioned when $d = 2$). A complete proof is given in Ref. [24].

The iterative power method can be readily extended, becoming: select $X = X^0$, and apply

$$X^{n+1} \propto (\nabla f(X^n))^{1/(d-1)}, \quad \|X^{n+1}\|_{L^d} = 1. \tag{12}$$

This simply means that, at each iteration, we select the pseudo-sphere of degree d the point where the normal is parallel to the gradient of f. Obviously, the only stable point is singular, and thus is the maximum we are looking for. We have only proved experimentally that the algorithm does converge very quickly to this maximum.

This procedure, already suggested in Ref. [21], yields a maximum which, as in the case $d = 2$, is inside $Q^{NM,d}$ (degeneracies apart), and thus does not yield a discrete labelling. The second key point is now to decrease d down to 1. More precisely, we define an iterative procedure as follows:

```
• set $\beta = d$, select some $X$;
• *while* ($\beta > 1$) *do*:
    find $X^*$ which maximises $f$ on $Q^{NM,\beta}$, starting from $X$,
    decrease $\beta$ by some quantity,
    project $X^*$ on the new $Q^{NM,\beta}$ giving $X$;
  *od*.
• for each $S_i$, select the label with value 1.
```

This iterative decrease of β can be compared up to a point to a cooling schedule, or better to a Graduated Non-Convexity strategy [5].

The last step (projection) is necessary, as changing β changes $Q^{NM,\beta}$. Actually, the normalisation performed at the first iteration of the maximisation process, for any β, takes care of that. On the other hand, the process defined by Eq. (11) cannot be applied when $\beta = 1$. Maximisation in that case has been thoroughly studied in Ref. [12]. In practice, it is simpler to stop at some β slightly larger than 1 (e.g. 1.1), as experiments confirm that for these values, the vector X^* almost satisfies the constraints in Eq. (8), and thus selecting the best label is trivial.

It is also important to notice that, though shifting the coefficients does not change the discrete problem nor the maximisation problem on $\mathscr{P}^{NM}$, it changes it on $Q^{NM,d}$, and thus there is no guarantee that the same solution is reached. Nor is it guaranteed that the procedure converges toward the global optimum; actually, it is not difficult to build simple counter-examples on toy problems. Experiments nevertheless show that, on real problems, a very good solution is reached, and that the speed with which β is decreased is not crucial: typically 5–10 steps are enough to go from 2 to 1.

4. Game strategy approach

Here we are interested in using game theory to find the maximum of $P(L/Y)$ defined in Eq. (3), or equivalently, to find the maximum of $f(L)$ defined in Eq. (4). The basic idea is to consider image sites as players, and labels as strategies of the players. Our cost function $f(L)$ is a global measure of energy potentials. This implies that the corresponding game should be designed in such a way that the total payoff of the team of players is maximised. In other words, the game should be a cooperative one: all the players have to take part in a coalition in order to find the global maximum of $f(L)$. However, as this problem is NP-hard, we will thus not be able to use the optimisation methods in this framework. Rather, we will take a non-cooperative game approach due to its simplicity. This method was first proposed in Ref. [25].

In a non-cooperative n-person game, there is a set of players $I = \{R_i, 1 \leq i \leq N\}$. Each player R_i has a set of pure strategies T_i (a mixed strategy is a probability distribution on the set of a player's pure strategies). The process of the game consists of each player R_i choosing independently his strategy $t_i \in T_i$ to maximise its own payoff $H_i(t)$. Thus, a play (a situation) $t = (t_1, \ldots, t_N)$ is obtained. It is assumed that each player knows all possible strategies and the payoff under each possible situation.

The solutions to such a game are the Nash equilibria (Nash points) [26]. A play $t^* = (t_1^*, \ldots, t_N^*)$ is a Nash equilibrium if none of the players can improve his expected payoff by unilaterally changing his strategy. In terms of the payoff functions, t^* satisfies the following relations:

$$\forall i : H_i(t^*) = \max_{t_i \in T_i} H_i(t^* \| t_i),$$

where $t^* \| t_i$ denotes the play obtained from replacing t_i^* by t_i. It is known [26] that Nash points always exist in n-person games with pure or mixed strategies.

We define the payoff functions in such a way that each player's payoff function depends on its own strategy and those of its neighbours, and that the total payoff takes into account our merit function $f(L)$:

$$H_i(L) = \sum_{c \in C_i} W_{cL_c}, \tag{13}$$

where $L = (L_1, \ldots, L_N)$. In general, $f(L) \neq \sum_{i=1}^{N} H_i(L)$, except when the potentials of higher order cliques sum to zero. However, we have proved [25] that the Nash equilibria exist for the game defined above, and that the set of Nash equilibria of the game is identical to the set of local maxima of the function $f(L)$. If one considers only cliques of orders 1 and 2, the game obtained, according to Miller and Zucker [17], is equivalent to the relaxation labelling formulated by variational inequalities [13].

Let $L^{(k)} = (L_1^{(k)}, \ldots, L_N^{(k)})$ denote the labelling at the k-th iteration, K be an integer representing the maximum number of iterations, $\alpha \in (0, 1)$ be a real number representing the probability of acceptance of a new label. The relaxation algorithm is as follows:

```
1. Initialize $L^{(0)} = (L_1^{(0)}, \ldots, L_N^{(0)})$, set $k = 0$.
2. At iteration $k \geq 0$, for each site $S_i$, *do*:
  2.1. Choose a label $L_i' \neq L_i^{(k)}$ such that
       $H_i(L^{(k)} \| L_i') = \max_{L_i \in \Lambda - \{L_i^{(k)}\}} H_i(L^{(k)} \| L_i)$;
  2.2. If $H_i(L^{(k)} \| L_i') \leq H_i(L^{(k)})$, then $L_i^{(k+1)} = L_i^{(k)}$;
       otherwise, accept $L_i'$ with probability $\alpha$;
  2.3. Let $L^{(k+1)} = (L_1^{(k+1)}, \ldots, L_N^{(k+1)})$.
  *od*.
3. If $L^{(k+1)}$ is a Nash point, or if $k \geq K$, then stop; otherwise, $k = k + 1$ and go to step 2.
```

In this algorithm, the players choose their strategies at discrete times $k = 0, 1, 2, \ldots$. At any time $k \geq 1$, each player has one-step delayed information on the strategies of its neighbours, and each player decides independently its new strategy in order to maximise its expected payoff. Therefore, the algorithm is intrinsically parallelisable in both SIMD and MIMD machines, and requires only local interprocessor communications.

The label updating scheme is randomised whenever $0 < \alpha < 1$. Such a randomisation not only guarantees the convergence of $L^{(k)}$ to a Nash equilibrium point when k tends to ∞ [25], but also makes the final labelling less dependent on the initialisation. When $\alpha = 1$, the algorithm may not converge. Simple counter-examples can be constructed where the labels oscillate. However, a deterministic version of the algorithm (with $\alpha = 1$) can be designed by excluding simultaneous updating of neighbouring objects. For example, the labels of objects can be updated sequentially. In this case one obtains a relaxation scheme similar to ICM [3].

5. Modified metropolis dynamics

We present another pseudo-stochastic method which optimises the same energy as in DPA and GSA. The definition of the local energies is given in section 6 for image classification (see Eqs. (19) and (20)). The proposed algorithm is a modified version of Metropolis Dynamics [10]: the choice of the new label state is made randomly using a uniform distribution; the rule to accept a new state is deterministic.

To guarantee the convergence of the parallel algorithm, we partition the entire image into disjoint regions $\mathcal{R}_n$ such that pixels belonging to the same region are conditionally independent of the pixels in all the other regions:

$$\mathcal{S} = \bigcup_n \mathcal{R}_n \quad \text{and} \quad \mathcal{R}_n \cap \mathcal{R}_m = \emptyset \ (n \neq m). \tag{14}$$

The parallel algorithm is as follows

1. Pick randomly an initial configuration $L^{(0)} = (L_1^{(0)}, \ldots, L_N^{(0)})$, with iteration $k = 0$ and temperature $T = T(0)$.
2. Pick a global state L' using a uniform distribution such that: $1 \leq L_i' \leq M$ and $L_i' \neq L_i^k$, $1 \leq i \leq N$.
3. For each site S_i, the local energy $\mathcal{E}_i(L')$, where $L' = (L_1^k, \ldots, L_{i-1}^k,\ L_i',\ L_{i+1}^k, \ldots, L_N^k)$ is computed in parallel using Eq. (20), presented in section 6.2.
4. Let $\Delta\mathcal{E}_i = \mathcal{E}_i(L') - \mathcal{E}_i(L^k)$. A new label state at site S_i is accepted according to the following rule:

$$L_i^{k+1} = \begin{cases} L_i' & \text{if } \Delta\mathcal{E}_i \leq 0 \text{ or } \left[\Delta\mathcal{E}_i > 0 \text{ and } \alpha \leq \exp\left(-\dfrac{-\Delta\mathcal{E}_i}{T}\right)\right] \\ L_i^k & \text{otherwise,} \end{cases}$$

   where $\alpha \in (0, 1)$ is a constant threshold; chosen at the beginning of the algorithm.
5. Decrease the temperature $T = T(k + 1)$ and go to step 2 until the number of modified sites is less than a threshold.

There is no explicit formula to get the value of α. For image classification, the more noise in the image, the smaller the value of α will be. α also regulates the speed of the algorithm as well as its degree of randomisation. See Ref. [27] for more details.

6. Performance comparisons

6.1. Implementation on a Connection Machine CM2

In this section, we briefly describe the architecture of the Connection Machine CM2. A more detailed description can be found in Ref. [28]. The Connection Machine is a single instruction multiple data (SIMD) parallel computer with 8 K to 64 K processors. Each processor is a 1-bit serial processor, with 32 K bytes of local memory and an 8 MHz clock. The Connection Machine is

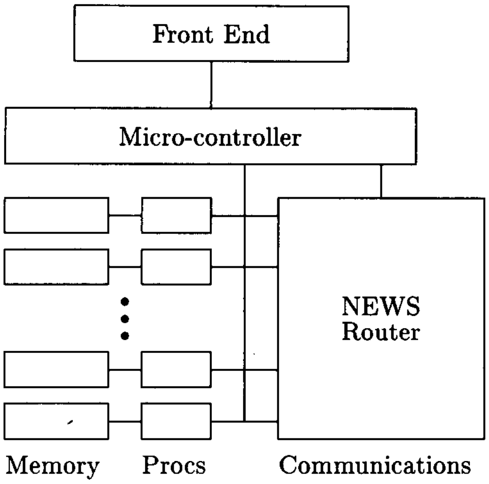


Fig. 1. CM-2 architecture.

accessed via a front-end computer which sends macroinstructions to a microcontroller. All processors are cadenced by the microcontroller to receive the same nano-instruction at a given time from it. Physically, the architecture is organised as follows (Fig. 1):

- The CM2 chip contains 16 1-bit processors.
- A Section is the basic unit of replication. It is composed of 2 CM2 chips, the local memory of the 32 processors and the floating point unit.
- The interprocessor communication architecture is composed of two distinct networks:
  - A nearest-neighbour network, the NEWS Network (North-East-West-South), interconnects processors in groups of four.
  - A more complex network, called the Router Network, is used to provide general communication between any pair of processors. Each group of 16 processors is connected to the same router, and each router is connected to 12 other routers forming a 12-dimensional hypercube.

For a given application, the user can dynamically define a particular geometry for the set of physical processors that has been attached.

The processor resource can be virtualised (VP Ratio) when the number of data elements to be processed is greater than the number of physical processors. In such a case, several data elements are processed on a single physical processor. Note that when the VP Ratio increases, the efficiency of the machine also increases, because the instruction is only loaded once.

Table 1
Results on the 'chess board' image with two classes

Algorithm	VPR	No. of Iter	Total time (s)	Time per It. (s)	Energy
ICM	2	8	0.078	0.009	52011.35
Metropolis	2	316	7.13	0.023	49447.60
Gibbs	2	322	9.38	0.029	49442.34
MMD	2	357	4.09	0.011	49459.60
GSA	2	20	0.20	0.010	50097.54
DPA	2	164	2.82	0.017	49458.02

Table 2
Results on the 'triangle' image with four classes

Algorithm	VPR	No. of Iter	Total time (s)	Time per It. (s)	Energy
ICM	2	9	0.146	0.016	49209.07
Metropolis	2	202	7.31	0.036	44208.56
Gibbs	2	342	14.21	0.042	44190.63
MMD	2	292	7.41	0.025	44198.31
GSA	2	31	0.71	0.023	45451.43
DPA	2	34	1.13	0.033	44237.36

For the three algorithms described in this paper, we use data parallelism on a square grid (one pixel per virtual processor) and the fast local communications (NEWS). Such an architecture is well suited for computer vision as it is expressed in Ref. [29].

6.2. Model for image classification

Using the same notation as in section 2, we want to maximise $P(L/Y)$ as defined in Eq. (3). We suppose that $P(y_i/L_i)$ is Gaussian:

$$P(y_i/L_i) = \frac{1}{\sqrt{2\pi}\sigma_{L_i}} \exp\left(\frac{-(y_i - \mu_{L_i})^2}{2\sigma_{L_i}^2}\right), \tag{15}$$

where μ_{L_i} $(1 \leq L_i \leq M)$ is the mean and σ_{L_i} is the standard deviation of class L_i. It is obvious from Eq. (3) that this expression with a factor $1/T$ corresponds to the energy potential of cliques of order 1. The second term of Eq. (3) is Markovian:

$$P(L) = \exp\left(-\frac{U(L)}{T}\right) = \exp\left(-\frac{1}{T}\sum_{c\in\mathscr{C}} V_{cL}\right)$$
$$= \exp\left(-\frac{1}{T}\sum_{\{S_i,S_j\}\in\mathscr{C}} \beta\gamma(L_{S_i}, L_{S_j})\right), \tag{16}$$

$$\gamma(L_{S_i}, L_{S_j}) = \begin{cases} -1 & \text{if } L_{S_i} = L_{S_j}, \\ +1 & \text{if } L_{S_i} \neq L_{S_j}, \end{cases} \tag{17}$$

Table 3
Results on the 'noise' image with three classes

Algorithm	VPR	No. of Iter	Total time (s)	Time per It. (s)	Energy
ICM	2	8	0.302	0.037	−5552.06
Metropolis	2	287	37.33	0.130	−6896.59
Gibbs	2	301	35.76	0.118	−6903.68
MMD	2	118	10.15	0.086	−6216.50
GSA	2	17	1.24	0.073	−6080.02
DPA	8	15	1.33	0.089	−6685.52

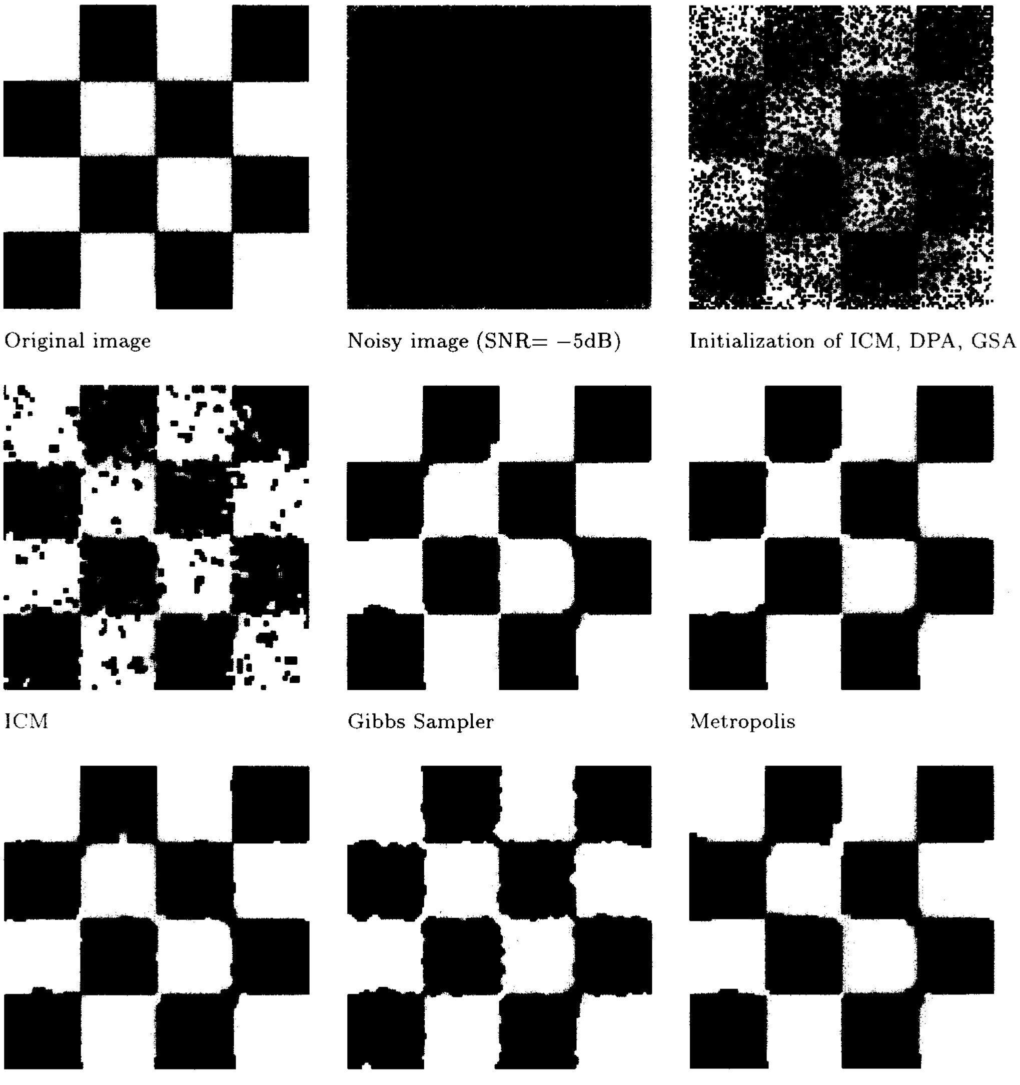
where β is a model parameter controlling the homogeneity of regions. We get the estimation of L, denoted by $\hat{L}$, as follows:

$$\begin{aligned}\hat{L} &= \arg\max_{L\in\mathscr{L}}\left(\frac{1}{T}\ln P(Y/L) + \ln P(L)\right)\\ &= \arg\max_{L\in\mathscr{L}}\left(\sum_{i=1}^{N} -\frac{1}{T}\left(\ln\sqrt{2\pi}\sigma_{L_i} + \frac{(y_i-\mu_{L_i})^2}{2\sigma_{L_i}^2}\right)\right.\\ &\quad\left. -\frac{1}{T}\sum_{\{S_i,S_j\}\in\mathscr{C}}\beta\gamma(L_{S_i},L_{S_j})\right)\\ &= \arg\min_{L\in\mathscr{L}}\left(\sum_{i=1}^{N}\frac{1}{T}\left(\ln\sqrt{2\pi}\sigma_{L_i} + \frac{(y_i-\mu_{L_i})^2}{2\sigma_{L_i}^2}\right)\right.\\ &\quad\left. +\frac{1}{T}\sum_{\{S_i,S_j\}\in\mathscr{C}}\beta\gamma(L_{S_i},L_{S_j})\right). \qquad (18)\end{aligned}$$

Using the above equation, it is easy to define the global energy $\mathscr{E}_{\text{glob}}(L)$ and the local energy $\mathscr{E}_i(L)$ at site S_i of labelling L:

$$\begin{aligned}\mathscr{E}_{\text{glob}}(L) &= \frac{1}{T}\left(\sum_{i=1}^{N}\left(\ln\sqrt{2\pi}\sigma_{L_i} + \frac{(y_i-\mu_{L_i})^2}{2\sigma_{L_i}^2}\right)\right.\\ &\quad\left. + \sum_{\{S_i,S_j\}\in\mathscr{C}}\beta\gamma(L_{S_i},L_{S_j})\right), \qquad (19)\end{aligned}$$

$$\begin{aligned}\mathscr{E}_i(L) &= \frac{1}{T}\left(\ln\sqrt{2\pi}\sigma_{L_i} + \frac{(y_i-\mu_{L_i})^2}{2\sigma_{L_i}^2}\right.\\ &\quad\left. + \sum_{\{S_i,S_j\}\in\mathscr{C}_i}\beta\gamma(L_{S_i},L_{S_j})\right). \qquad (20)\end{aligned}$$



Original image

Noisy image (SNR= −5dB)

Initialization of ICM, DPA, GSA

ICM

Gibbs Sampler

Metropolis

DPA

GSA

MMD

Fig. 2. Results on the ‘chess board’ image with two classes.

6.3. Experimental results

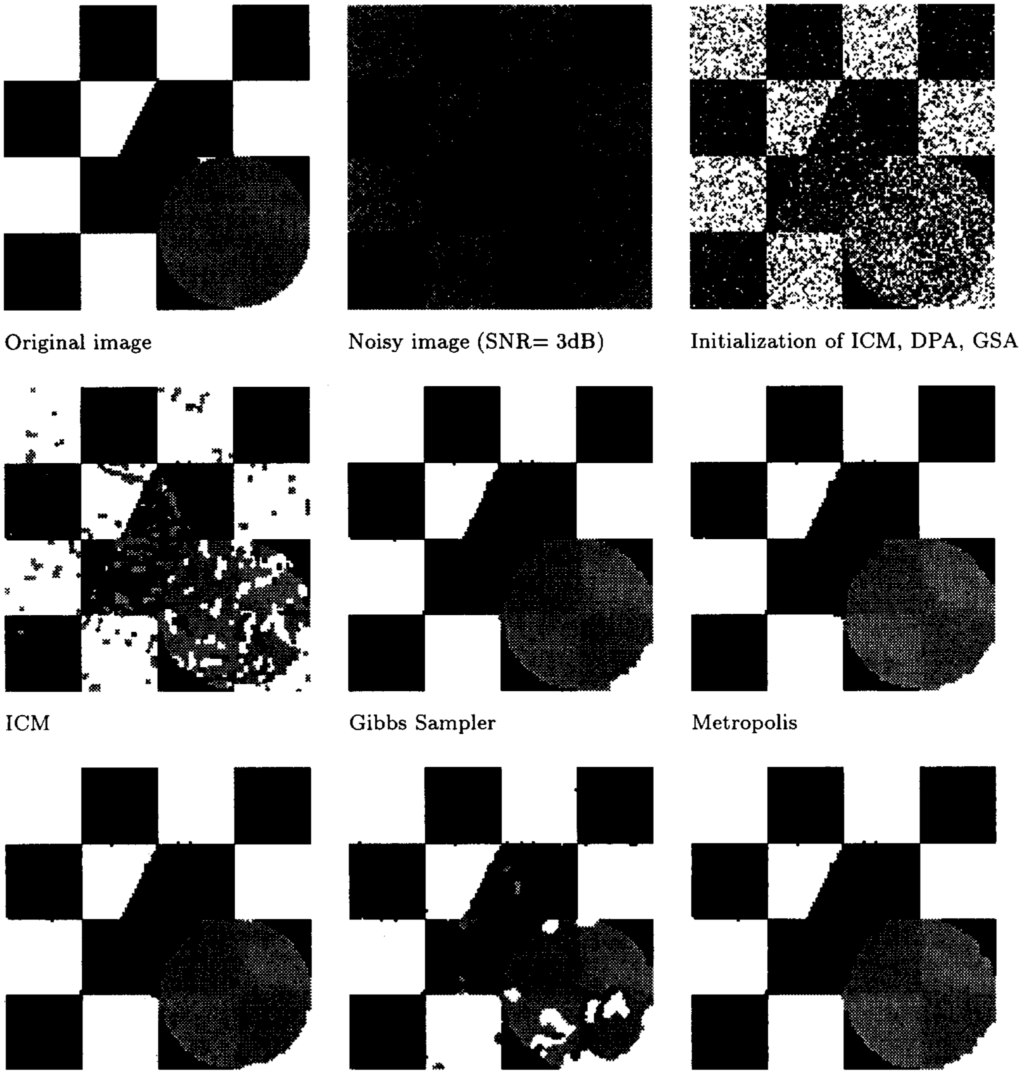
The goal of this experiment is to evaluate the performance of the three algorithms (DPA, GSA and MMD) proposed by the authors. We compare these algorithms with three well-known MRF based methods: ICM [3,4], a fast, deterministic relaxation scheme; Metropolis Dynamics [10]; and Gibbs Sampler [1]. The last two are classical fully-stochastic relaxation techniques. The performances are evaluated in two respects for each algorithm: the reached global minimum of the energy function and the computer time required. We remark that in all cases, the execution is stopped when the energy change ΔU is less than 0.1% of the current value of U.

We display experimental results on the following images:

1. 'chess board' image with 2 classes, pixel size: 128 × 128 (Fig. 2),
2. 'triangle' image with 4 classes, pixel size: 128 × 128 (Fig. 3),
3. 'noise' image with 3 classes, pixel size: 256 × 256 (Fig. 4),
4. a SPOT image with 4 classes, pixel size: 256 × 256 (Fig. 5).

Tables 1–4 give, for each image and for each algorithm, the Virtual Processor Ratio (VPR), the number of iterations, the computer time, and the reached minima of the energies. Note that only the relative values of the energies matter. The following parameters are used for the experimentation:

- *Initialisation of the labels*: Random values are assigned to the labels for the initialisation of the Gibbs, Metropolis and MMD algorithms. For the ICM, DPA and GSA techniques, the initial labels are obtained using only the Gaussian term in Eq. (18) (for ICM, this


Original image

Noisy image (SNR= 3dB)

Initialization of ICM, DPA, GSA

ICM

Gibbs Sampler

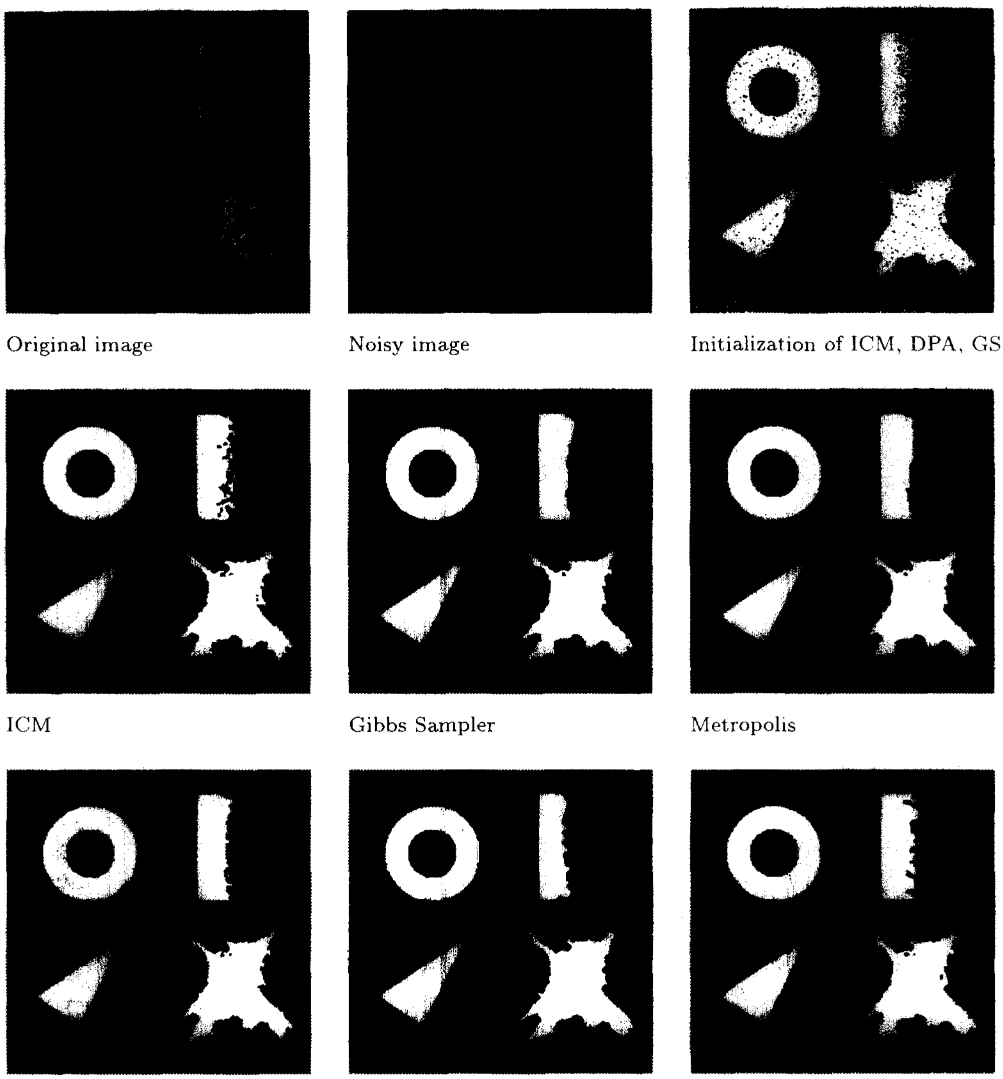
Metropolis

DPA

GSA

MMD

Fig. 3. Results on the 'triangle' image with four classes.



Original image

Noisy image

Initialization of ICM, DPA, GSA

ICM

Gibbs Sampler

Metropolis

DPA

GSA

MMD

Fig. 4. Results on the 'noise' image with three classes.

means the 'Maximum Likelihood' estimate of the labels). ICM is very sensitive to the initial conditions, and better results perhaps could have been obtained with another initialisation method. Nevertheless, the ICM, DPA and GSA algorithms have been initialised with the same data for the experiment.

- *Temperature*: The initial temperature for those algorithms using annealing (i.e. Gibbs Sampler, Metropolis, MMD) is $T_0 = 4$, and the decreasing schedule is given by $T_{k+1} = 0.95\ T_k$. For other algorithms (i.e. ICM, DPA, GSA), $T = 1$.
- *Mean and standard deviation of each class*: They are computed using a supervised learning method. Their values are listed in Table 5.
- *Choice of β*: β controls the homogeneity of regions. The greater the value of β is, the more we emphasise the homogeneity of regions. The values of β used for different images are shown in Table 6.

Table 4
Results on the 'SPOT' image with four classes

Algorithm	VPR	No. of Iter	Total time (s)	Time per It. (s)	Energy
ICM	8	8	0.381	0.048	−40647.96
Metropolis	8	323	42.37	0.131	−58037.59
Gibbs	8	335	46.73	0.139	−58237.32
MMD	8	125	10.94	0.087	−56156.53
GSA	8	22	1.85	0.084	−56191.61
DPA	8	15	1.78	0.119	−52751.71

Table 5
Model parameters for the four images

Image	μ_1	σ_1^2	μ_2	σ_2^2	μ_3	σ_3^2	μ_4	σ_4^2
chess board	119.2	659.5	149.4	691.4	–	–	–	–
triangle	93.2	560.6	116.1	588.2	139.0	547.6	162.7	495.3
noise	99.7	94.2	127.5	99.0	159.7	100.1	–	–
SPOT	30.3	8.2	37.4	4.6	61.3	128.1	98.2	127.1

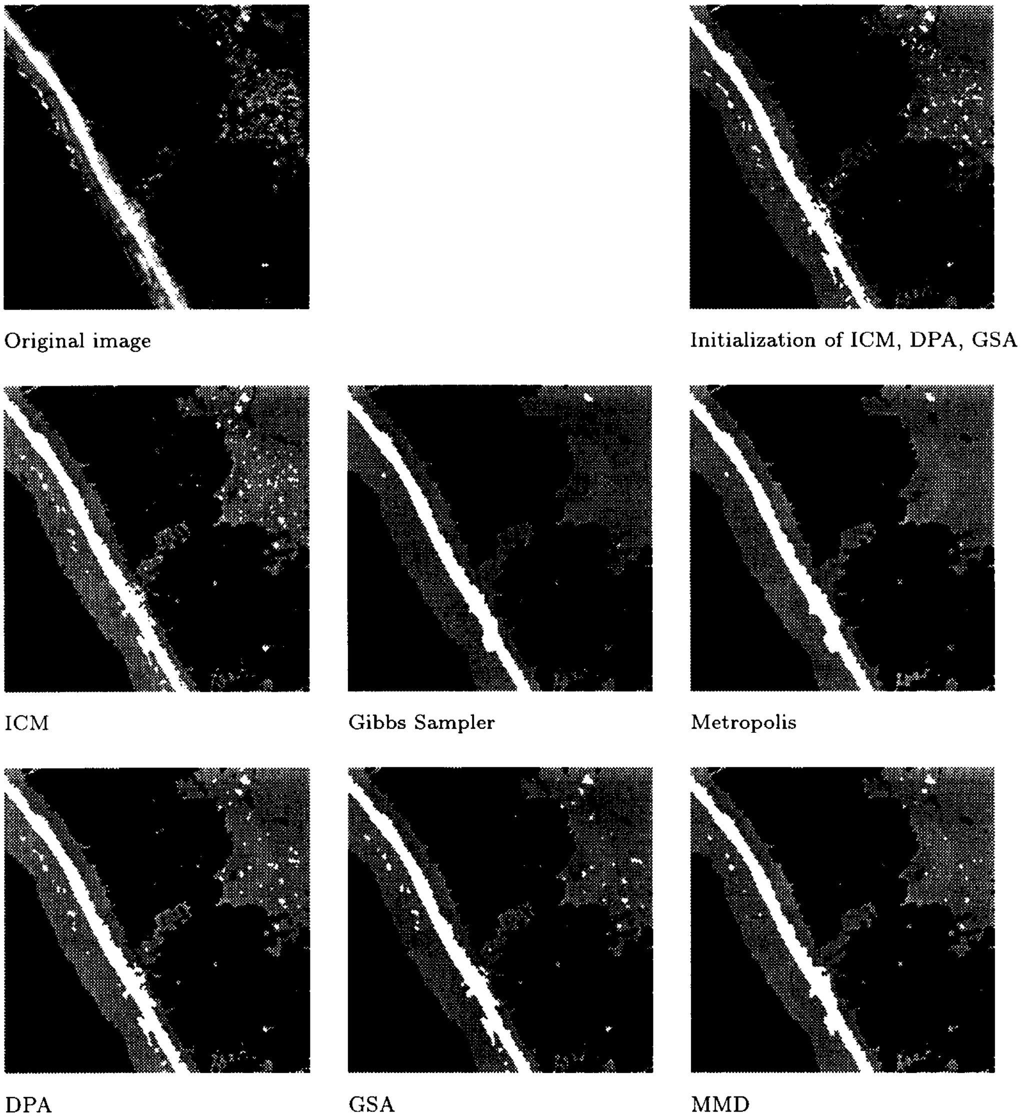


Fig. 5. Results on the 'SPOT' image (four classes).

- *Choice of α for MMD*: α regulates the speed of the algorithm of MMD as well as its degree of randomisation. The values of α for different images are also shown in Table 6.

We remark that both α and β are chosen by trial and error.

As shown by Tables 1–4 and Figs. 2–5, stochastic schemes are better regarding the achieved minimum, thus the classification error; deterministic algorithms are better regarding the computer time. ICM is the fastest, but the reached minimum is much higher than for the other methods (as mentioned earlier, another initialisation might lead to a better result, but a more elaborate initialisation usually increases the computer time). DPA, GSA and MMD seem to be good compromises between quality and execution time. Sometimes the results obtained by these algorithms are very close to the ones of stochastic methods. On the other hand, they are much less dependent on the initialisation than ICM, as shown by the experiment results and also in the theoretical aspects. Note that, in Fig. 4, we are lucky that the maximum likelihood initialisation is very close to the true solution. But it is not generally the case. ICM improves the results in other figures, but not really in Fig. 4 because, on this particular image, the initial conditions are really good.

Table 6
The β value, and the α value for MMD

Image	β	α (for MMD)
chess board	0.9	0.3
triangle	1.0	0.3
noise	2.0	0.7
SPOT	2.0	0.7

It should be noticed that the three proposed algorithms do about the same when averaged across the different test images. It is impossible to say that one approach is better than the others. The choice of using which one of these methods is a question of taste.

Finally, we remark that the random elements in GSA and in MMD are different from each other, and also different from that in simulated annealing schemes. In GSA, candidate labels are chosen in a deterministic way. Only the acceptance of the candidate is randomised. In MMD, the choice of the new label state is done randomly, and the rule to accept a new state is deterministic. In simulated annealing, both the selection and the acceptance of candidates are randomly decided.

7. Conclusion

In this paper, we have described three deterministic relaxation methods (DPA, GSA and MMD) which are suboptimal but are much faster than simulated annealing techniques. Furthermore, the proposed algorithms give good results for image classification and compare favourably with the classical relaxation methods. They represent a good trade-off for image processing problems. It should be noticed that the three algorithms have about the same performance when averaged across the different test images.

Until now, we have used them in a supervised way, but an interesting problem would be to transform them into fully data-driven algorithms. This should be done using parameter estimation techniques such as Maximum Likelihood, Pseudo-Maximum Likelihood or Iterative Conditional Estimation, for instance. Finally, the three proposed methods could be adapted to multi-scale or hierarchical MRF models.

Acknowledgements

This work has been partially funded by CNES (French Space Agency). The data shown in Fig. 3 was provided by GdR TdSI (Working Group in Image and Signal Processing) of CNRS.

References

[1] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, IEEE Trans. Patt. Analysis and Mach. Intel., 6 (1984) 721-741.

[2] R. Azencott, Markov fields and image analysis, Proc. AFCET, Antibes, France, 1987.

[3] J. Besag, On the statistical analysis of dirty pictures, J. Roy. Statis. Soc. B., 68 (1986) 259-302.

[4] F.C. Jeng and J.M. Woods, Compound Gauss-Markov Random Fields for image estimation, IEEE Trans. Acoust., Speech and Signal Proc., 39 (1991) 638-697.

[5] A. Blake and A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, MA, 1987.

[6] A. Rangarajan and R. Chellappa, Generalised graduated nonconvexity algorithm for maximum a posteriori image estimation, Proc. ICPR, June 1990, pp. 127-133.

[7] D. Geiger and F. Girosi, Parallel and deterministic algorithms for MRFs: surface reconstruction and integration, Proc. ECCV90, Antibes, France, 1990, pp. 89-98.

[8] J. Zerubia and R. Chellappa, Mean field approximation using compound Gauss-Markov random field for edge detection and image restoration, Proc. ICASSP, Albuquerque, NM, 1990.

[9] P.V. Laarhoven and E. Aarts, Simulated Annealing: Theory and applications, Reidel, Dordrecht, Holland, 1987.

[10] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Physics, 21 (1953) 1087-1092.

[11] H. Derin, H. Elliott, R. Cristi and D. Geman, Bayes smoothing algorithms for segmentation of binary images modeled by markov random fields, IEEE Trans. Patt. Analysis and Mach. Intel., 6 (1984).

[12] O. Faugeras and M. Berthod, Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach, IEEE Trans. Patt. Analysis and Mach. Intel., 4 (1981) 412-423.

[13] R. Hummel and S. Zucker, On the foundations of relaxation labeling processes, IEEE Trans. Patt. Analysis and Mach. Intel., 5(3) (1983) 267-287.

[14] R.H.A. Rosenfeld and S.W. Zucker, Scene labeling by relaxation operation, IEEE Trans. Systems, Man and Cybernetics, 6(6) (1984) 420-433.

[15] H.I. Bozma and J.S. Duncan, Integration of vision modules: a game-theoretic approach, Proc. Int. Conf. Computer Vision and Pattern Recognition, Maui, HI, 1991, pp. 501-507.

[16] H.I. Bozma and J.S. Duncan, Modular system for image analysis using a game-theoretic framework, Image & Vision Computing, 10(6) (1992) 431-443.

[17] D.A. Miller and S.W. Zucker, Copositive-plus Lemke algorithm solves polymatrix games, Oper. Res. Letters, 10 (1991) 285-290.

[18] D.A. Miller and S.W. Zucker, Efficient simplex-like methods for equilibria of nonsymmetric analog networks, Neural Computation, 4 (1992) 167-190.

[19] T. Basar and G.J. Olsder, Dynamic Noncooperative Game Theory, Academic Press, New York, 1982.

[20] M. Berthod, L'amélioration d'étiquetage: une approche pour l'utilisation du contexte en reconnaissance des formes. Thèse d'état, Université de Paris VI, December 1980.

[21] M. Berthod, Definition of a consistent labeling as a global extremum. Proc. ICPR6, Munich, Germany 1982, pp. 339-341.

[22] J.E. Besag, Spatial interaction and the statistical analysis of lattice systems (with discussion), J. Roy. Statis. Soc. B, 36 (1974) 192-236.

[23] A. Berman and R. Plemmons, Non-negative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.

[24] L. Baratchart and M. Berthod, Optimization of positive generalized polynomials under l^p constraints. Technical report, INRIA, France, 1995.

[25] S. Yu and M. Berthod, A game strategy approach to relaxation labeling, Computer Vision and Image Understanding, 61(1) (January 1995) 32-37.

[26] J.F. Nash, Equilibrium points in n-person games, Proc. Nat. Acad. Sci. USA, 36 (1950) 48-49.

[27] Z. Kato, Modélisation Markovienne en classification d'image mise en oeuvre d'algorithmes de relaxation, Rapport de stage de DEA, Université de Nice, 1991.

[28] W.D. Hillis, The Connection Machine, MIT Press, Cambridge, MA, 1985.

[29] J. Little, G. Belloch and T. Cass, Algorithmic techniques for computer vision on a fine-grain parallel machine, IEEE Trans. Patt. Analysis and Mach. Intel. 11 (1989) 244-257.