

**Efficient Parallel Non-Linear Multigrid Relaxation
Algorithms for Low-level Vision Applications**

E. Mémin, F. Heitz and F. Charot

N° 2184

Avril 1994

PROGRAMME 4

Robotique,

image

et vision


R
**apport
de recherche**



Efficient Parallel Non-Linear Multigrid Relaxation Algorithms for Low-level Vision Applications

E. Mémin, F. Heitz and F. Charot

Programme 4 — Robotique, image et vision
Projet TEMIS

Rapport de recherche n° 2184 — Avril 1994 — 39 pages

Abstract: Many tasks in computer vision and image analysis have recently been expressed as the minimization of global energy functions describing the local interactions between the observed data and the images features to be extracted. For the minimization of these global (often non-linear) energy functions, a variety of stochastic or deterministic non-linear relaxation algorithms have been described in the literature. The major drawback of relaxation algorithms remains the amount of computation required to update the image. For real world applications the computation time quickly becomes prohibitive on workstations. Several efficient approaches have been proposed to alleviate this computational burden. Among them, multigrid techniques [7, 23, 43]. have shown to significantly improve the convergence rate of linear and non-linear relaxation schemes. It is also well known that the computations involved by these algorithms are regular and local, and lead naturally to massive data parallelism, which is well suited for parallel processing on array processor architectures. Standard parallel implementations of relaxation algorithms are based on the simultaneous updating of the different sites of the image. This kind of data parallelism unfortunately does not exploit the large computing resources of large massively parallel processor arrays when the image grid to be controlled is small. This is particularly true for multigrid relaxation algorithms, in which coarse and fine grids have to be handled sequentially.

In this paper, we present a new algorithmic framework which enables making a full use of the large potential of data parallelism available on 2D processor arrays for the implementation of non-linear multigrid relaxation methods. The approach combines two different levels of parallelism: parallel updating of the image sites and concurrent explorations of the configuration space of the problem. This approach is very efficient and leads to fast convergence towards quasi-optimal solutions. The method is demonstrated on two different low-level vision applications: restoration of noisy images and optical flow computation.

Key-words: global optimization, image processing, parallel relaxation, 2D SIMD array architectures

(Résumé : tsvp)

Extended version of paper presented at CVPR94, submitted to the Journal of Parallel and Distributed Computing.

Algorithmes parallèles de relaxation multigrille pour des applications de vision bas-niveau

Résumé : Dans les domaines de l'analyse d'images et de la vision un nombre important de tâches s'expriment sous la forme de la minimisation de fonctions d'énergie globale. Ces fonctions d'énergie décrivent les interactions locales entre les primitives que l'on souhaite extraire et un ensemble d'observations. Cette minimisation – non-linéaire dans la plupart des cas et portant sur un nombre très élevé de variables – est usuellement menée par des algorithmes itératifs dit de *relaxation*.

Malgré l'élaboration de techniques de relaxation multirésolution qui permettent d'accroître la vitesse de convergence de ces schémas itératifs, l'utilisation d'une telle algorithmique dans des applications réelles reste problématique. Le volume de calculs sous-tendu par ces méthodes est en effet extrêmement important et ne peut être réalisé efficacement sur des calculateurs séquentiels conventionnels. Fort heureusement, les calculs induits par ces algorithmes sont *réguliers* et *locaux*. Ces propriétés présagent d'un parallélisme de données intensif pouvant être mis en œuvre de manière efficace sur des architectures SIMD massivement parallèles.

Les techniques classiques de parallélisation des algorithmes de relaxation consistent à remettre à jour simultanément des sous-ensembles de sites de la grille image. Ce parallélisme de données se caractérise par une sous-utilisation du réseau de processeurs lorsque la taille de l'image à traiter est réduite. Ceci est particulièrement vrai dans le cas des techniques de relaxation multigrille où les grilles associées aux différentes résolutions doivent être réactualisées de façon séquentielle.

Nous proposons dans cet article de nouveaux schémas de relaxation parallèles qui, tout en s'appuyant sur des techniques algorithmiques multigrilles, permettent l'utilisation totale du large potentiel de parallélisme de données octroyé par les architectures SIMD massivement parallèles. Ces approches, mêlant plusieurs niveaux de parallélismes (réactualisation simultanée des sites et parcours de solutions concurrentes dans l'espace des configurations du problème) se révèlent particulièrement efficaces comparées aux techniques de relaxation classiques. La généralité de ces approches est démontrée sur deux problèmes de vision bas-niveau : la restauration d'images bruitées et la mesure du mouvement apparent.

Mots-clé : optimisation globale, vision bas-niveau, relaxation parallèle, parallélisme de données, architectures SIMD 2D.

1 Introduction

Many tasks in computer vision and image analysis have recently been expressed as global optimization problems [6, 12, 16]. The general issue is to find the global minimum of an objective (also called energy) function which describes the interaction between the different variables modeling the image features in a given problem [16]. Two kinds of variables are generally considered: *observation variables* which correspond to the representation of the observed data and *hidden variables* (or labels) which are the representations to be extracted from the original images. Energy functions generally involve two components, one of which expresses the interaction between the hidden labels and the observations and the other which encodes constraints on the desired solution [16]. To keep the problem tractable, the energy is often decomposed as a sum of local interaction functions defined on a neighborhood [16]. Standard regularization approaches [41] as well as Markov Random Field-based image analysis lead to the minimization of such global energy functions [16] (Markov Random Field (MRF) models define a global probability distribution that is maximized when an energy function is minimized [18]). The choice of these energy functions is either heuristic or may be guided by a statistical modeling of the interaction between the variables.

Defining global energy functions is a powerful tool for specifying non-linear interactions between different image features (luminance, edges, region labels, etc.). They help to combine and organize spatial and temporal information by introducing strong generic knowledge about the features to be estimated. For instance, global energy functions have been successfully introduced in image restoration [4, 10, 18], edge detection [16], image segmentation [16], stereovision [2], computed tomography, surface reconstruction [12], visual motion analysis [8, 9, 21, 29] and scene interpretation [35].

However, minimizing a global energy function is often an intricate problem: the number of possible label configurations is generally *very large* and the global energy function may exhibit many local minima. Computationally demanding stochastic relaxation algorithms are generally necessary to compute optimal solutions. Less cpu intensive deterministic relaxation algorithms such as ICM [4], HCF (Highest Confidence First [12]) or GNC (Graduated Non Convexity [6]) can often be used instead, when a good initial guess is available. Deterministic approaches converge to configurations corresponding to local minima of the global energy function. They may be combined with multigrid methods to improve the convergence rate of iterative relaxation schemes [7, 23, 43].

The major drawback of relaxation algorithms is the amount of computation required to update the image. For real world applications the computation time quickly becomes prohibitive on workstations. On the other hand, in low-level vision,

the global energy functions usually adopted decompose into local interaction functions which only involve a pixel and its neighbors [18, 41]. This local decomposition may be related to the local statistical properties of Markov Random Field models [14, 18]. As a consequence, the computations involved by these algorithms are regular and local, and thus independent variables in the image may be updated at once. This leads naturally to massive data parallelism, which is well suited for parallel processing on array processor architectures.

Standard parallel implementations of relaxation algorithms are thus based on the simultaneous updating of subsets of non-neighboring sites in the image. These subsets are obtained by considering a coloring¹ of the graph associated to the neighborhood system defined on the image grid (this partition of the image sites is often called a *codage* [4]). Most implementations of non-linear relaxation schemes on 2D SIMD arrays make use of this paradigm (see for instance [26, 36] on the DAP 510 or [2, 5, 27, 40, 45] on the CM2). Implementations on linear SIMD arrays, based on the same principle [13, 34, 33] have also been described, in order to design specialized architectures. Alternate approaches involve the development of highly specialized linear or non-linear (electrical) analog networks [25, 32, 41], Hopfield neural networks [31, 38, 46] or Boltzmann machines [24], for mapping the underlying global optimization problems.

Nevertheless, standard data parallelism does not exploit the large computing resources of the now available large massively parallel processor arrays *when the image grid to be controlled is small* [37]. This is particularly true for multigrid relaxation algorithms, in which coarse and fine grids have to be handled sequentially.

In this paper, we present new algorithmic techniques which enable to make a full use of the large potential of data parallelism available on 2D processor arrays for the implementation of non-linear multigrid relaxation methods. The approach combines two different levels of parallelism: parallel updating of image sites and concurrent explorations of the configuration space of the variables modeling the image features. This approach turns out to be very efficient and leads to fast convergence towards quasi-optimal solutions. The generality and the efficiency of this technique is demonstrated on two different low-level applications: optical flow computation and restoration of noisy images.

The remainder of this paper is organized as follows. In Section 2, the general class of non linear relaxation techniques considered in this paper is described. Representative examples of stochastic, deterministic and multigrid relaxation techniques are presented. In Section 3, we present a parallel algorithmic framework which achieves full efficiency on 2D array processors for non-linear multigrid relaxation schemes. Benchmarks on synthetic as well as real-world images are reported in Section 4 in the case of motion estimation and image restoration problems.

¹The coloring of a graph is the assignment of a color to each site such that two neighboring sites are not of the same color

2 Non-linear relaxation algorithms and global optimization

2.1 Global optimization in low-level vision

The optimization of a global energy function is a powerful approach to extract relevant representations (labels) from an image. The method proceeds as follows:

- One or more specialized modules extract features (spatio-temporal gradients, edges...) from the images, that will be used as observations in the optimization process.
- Observations are combined using local photometric and structural relations with generic a priori knowledge on the label configurations (regularization scheme) in order to derive estimates of the unknown labels [12, 16].

The problem is stated as the minimization of a global energy function describing the interactions between the different (observed and hidden) variables.

Let $o = \{o_s, s \in S\}$ designate the observed data defined on a rectangular lattice S . Let $e = \{e_s, s \in S\}$ denote the unobserved (hidden) label field, defined on the same lattice S^2 . Let Λ be the (discrete) state space of variable e_s and Ω the (finite) set of all possible label configurations e . Let $\mathcal{G} = \{\mathcal{G}_s, s \in S\}$ define a neighborhood system on S .

In the following we assume that the interactions between observed data o and hidden labels e can be defined in terms of an energy function of the form

$$U(e, o) = U_1(e, o) + U_2(e), \quad (1)$$

$$\begin{aligned} \text{with } U_1(e, o) &\triangleq \sum_{s \in S} V_1(e_s, o_s), \text{ and} \\ U_2(e) &\triangleq \sum_{c \in \mathcal{C}} V_c(e). \end{aligned} \quad (2)$$

The optimal labeling is, by definition, the minimum of this energy function. The first component $U_1(e, o)$ of the energy expresses (pointwise) interaction between the hidden labels e and the observations o , the second term $U_2(e)$ encodes constraints on the desired solution. The set of cliques associated to the neighborhood system \mathcal{G} , is denoted by \mathcal{C} . Cliques c are subsets of sites which are mutual neighbors. The local interaction function V_c depends only on the variables of clique c and expresses the local interactions between these different variables. The form of the local interaction functions V_c is problem dependent. Local interaction models have several relevant advantages: they are easy to specify and they yield tractable iterative computational schemes in which the label configurations can be updated locally [4, 12, 18].

In the following, we consider a standard 8-neighborhood system for \mathcal{G} . The cliques $c \in \mathcal{C}$ associated to the 8-neighborhood are shown in *Fig. 1* and contain at most four sites. This energy class comprises most usual energy functions used in computer

²Different lattices for E and O can also be adopted

vision [2, 16, 21] (the extension of the approach presented in this paper to larger neighborhoods is straightforward, although not only a notational matter).

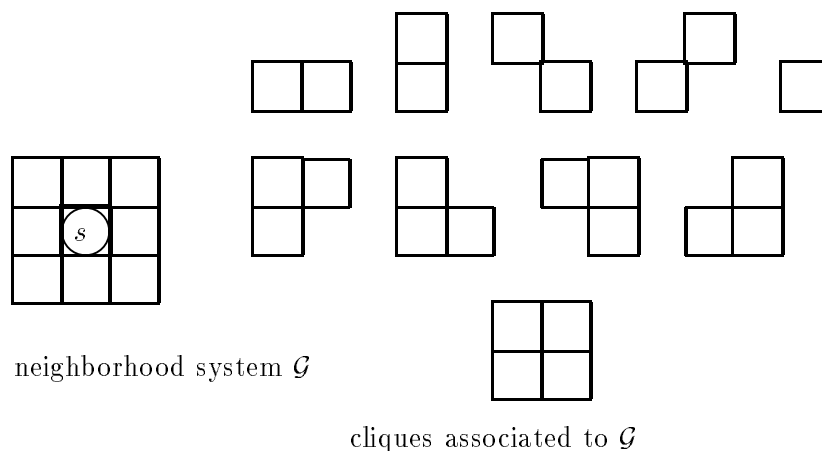


Figure 1: *8-neighborhood structure and associated cliques*

For the minimization of these global (possibly non-linear) energy functions, a variety of stochastic or deterministic relaxation algorithms as well as multigrid techniques have been described in the literature. These different classes of algorithms are described in the following section.

2.2 Non-linear relaxation algorithms

2.2.1 Stochastic relaxation

Since the seminal work of Kirkpatrick *et al.* [28] on simulated annealing, stochastic relaxation has been intensively studied and used in large scale optimization problems [1, 18]. Stochastic relaxation algorithms theoretically guarantee convergence towards the global minimum of highly non-linear and non-convex objective functions. The final solution theoretically does not depend on the initial state of the system. This class of algorithm thus provides robust solutions when no prior knowledge on the optimal solution (or no relevant initial guess) is available. These attractive features however are generally obtained at a great computational expense. Several hundred iterations (*i.e.* full scans of the image) are indeed generally necessary to reach convergence.

Stochastic relaxation may be seen as the combination of two procedures: a sampling process and an annealing schedule. The sampling process generates a sequence of states according to a Gibbs distribution (*Eqn.* (3)) at temperature T . At each temperature the system is allowed to reach a statistical *equilibrium*. The probability of being in a state e with energy $U(e, o)$ at temperature T is given by the following

Gibbs (or Boltzmann) distribution:

$$Pr\{E = e | O = o\} = \frac{1}{Z(T)} \cdot \exp\left(-\frac{U(e, o)}{T}\right), \quad (3)$$

where $Z(T)$ is a normalization constant – known as the partition function – depending on temperature T . The annealing schedule is defined by a controlled decrease of temperature T . If the temperature is decreased adequately (i.e. slowly enough) one can show that the sampling process converges towards the fundamental states of the system i.e. the minimum energy states (see [1, 18] for convergence theorems).

A typical example of a stochastic relaxation algorithm is described in *Fig. 2*. The algorithm presented in *Fig. 2* is based on a sampling process called the “Gibbs sampler”, proposed by Geman and Geman in [18] for the optimization of MRF-related global energy functions. Its dynamics differ somewhat from the standard Metropolis algorithm [1] in that it directly uses the local characteristics of Gibbs distribution (*Eqn. (3)*) rather than a random choice based on the energy difference between two states of the system. The updating of a given site s is local and only involves site s and its neighbors (*Fig. 2*). This particular algorithm has been widely

```

function SR( $\hat{e}, U(o, e)$ ) result  $\hat{e} \in \Omega$  ;

 $\hat{e}$            : initial configuration ;
 $e(k)$         : current configuration at iteration k ;
 $T_0$          : initial temperature value ;
 $\mathcal{AS}(T)$     : annealing schedule ;

 $k = 0$  ;
 $e(k) = \hat{e}$  ;
repeat until convergence criterion satisfied
{
  choose a site s (randomly or according to a specified order) ;
  assign to site s label  $e_s$  according to the conditional distribution:
   $Pr\{E_s = e_s | E_r = e_r, r \in \nu_s ; o\}$ 
    =  $\frac{1}{Z_s(T_k)} \exp[-\frac{1}{T_k} \{V_1(e_s, o_s) + \sum_{c \in \mathcal{C}|s \in c} V_c(e)\}]$ 
   $T_k = \mathcal{AS}(T_{k-1})$  ;
}
 $\hat{e} = e(k)$  ;

```

Figure 2: A stochastic relaxation algorithm based on the Gibbs sampler

used in image processing and computer vision applications [10, 11, 18]. It is adopted here as a representative example of this class of global optimization techniques.

2.2.2 Deterministic relaxation

Stochastic optimization algorithms are time consuming, especially in low-level vision applications in which the configuration spaces are often very large. Most of the recent

papers [8, 9, 11, 12, 21] resort to deterministic schemes which are more appealing, as far as computation time is concerned. Deterministic relaxation converges to a local minimum of the energy function, but this loss of optimality may be compensated for by an appropriate initial guess. When a relevant initial guess is not available, the solution at convergence may however be far from the optimum and lead to poor performances [23, 29].

A very popular non-linear deterministic relaxation scheme, known as the “Iterated Conditional Mode” (ICM) algorithm, has been proposed by Besag in 1986 [4]. ICM basically corresponds to non-linear Gauss-Seidel relaxation. It typically converges within about ten iterations (*i.e.* full scans of the image) to a local minimum of the objective function. As already stated, these good performances on the convergence speed have to be moderated, since poor solutions may be reached on highly non-linear models when a suitable initial configuration - close to the optimum - is not available.

The updating scheme in the ICM algorithm is *local* and only involves the labels at the current site s and its neighbors (*Fig. 3*). This algorithm is described in *Fig. 3*. It is used in this paper as a standard example of non-linear deterministic relaxation.

```

function ICM( $\hat{e}, U(o, \epsilon)$ ) result  $\hat{e} \in \Omega$  ;

|  $\hat{e}$       : initial configuration ;
|  $\epsilon(k)$  : current configuration at iteration  $k$  ;
 $k = 0$  ;
 $\epsilon(k) = \hat{e}$  ;
repeat until convergence criterion satisfied
{
  choose a site  $s$  (randomly or according to a specified order) ;
  assign to site  $s$  the label  $e_s$  that minimizes:

   $\Delta U(e_s) = U(\epsilon(k-1), o) - U(\epsilon(k), o)$  where :  $e_s(k) = e_s$ 
                                                     $e_r(k) = e_r(k-1) \forall r \neq s$ 
   $= V_1(e_s, o_s) + \sum_{c \in \mathcal{C} | s \in c} V_c(e)$ 

}
 $\hat{e} = \epsilon(k)$  ;

```

Figure 3: *The ICM relaxation algorithm*

2.2.3 Multigrid relaxation

It is well known that multigrid methods can significantly improve the convergence rate of linear and non-linear iterative relaxation schemes [7, 20, 23, 43]. Multigrid

methods may also be useful when the energy to be minimized has many local minima, as is often the case with non-linear models. It has indeed been conjectured that multigrid analysis may, to a certain extent, *smooth* the energy landscape. Fast deterministic relaxation schemes can then be used at coarse scales to obtain a good initial guess that may be refined at the finer scales. This procedure often yields fast convergence towards good estimates, even if the initial configuration of the system is far from the optimum [7, 23, 27, 29].

Multigrid relaxation techniques have been considered for image analysis models based on partial differential equations [15, 43] as well as on MRF models [2, 29]. Yet, in multigrid implementations of relaxation algorithms devoted to the minimization of global non-linear energy functions, the choice of the energy functions (and the adjustment of their parameters) at different scales remains a key problem. A standard choice is to adopt the same function at all scales, even though the interactions between variables are often resolution-dependent.

In [23], a new multigrid algorithm (called “multiscale relaxation”) applicable to the minimization of global energy functions used in image analysis or computer vision tasks is developed. This algorithm is not equivalent to the standard multigrid approaches proposed in [15, 29, 43]. It is related to a multiscale “constrained” exploration of the set of solutions of the original optimization problem. The global optimization problem is solved within a sequence of particular subspaces of the original space of configurations. These subspaces contain constrained configurations describing the expected solutions at different scales. Each subspace defines a new “coarse” energy function whose parameters are derived from the original (full resolution) objective function. This constrained optimization is implemented using a coarse to fine procedure on a pyramidal structure.

“Multiscale relaxation” typically converges within a few iterations (about ten full scans of the image) towards estimates that are close to the global optimum of the energy function [23]. In comparison, a standard stochastic method (simulated annealing) requires several hundred iterations to converge to similar solutions. It is also shown in [23] that these relaxation schemes yield better results than the standard multiresolution processing used in this field [21, 29]. This attractive relaxation scheme will be considered in the following as a good candidate for highly efficient parallel implementations. It is described in the remainder of this section.

As already explained, the “multiscale relaxation” scheme is based on a “multiscale” exploration of the set of all possible label configurations Ω [23]. Let us assume that the size of the full resolution lattice is $2^m \times 2^m$. First recall that the energy function assigns an energy value to all possible configurations of the labeling $e = \{e_s, s \in S\}$, $e_s \in \Lambda$, $e \in \Omega$.

Instead of considering the minimization of energy function (*Eqn. 2*) directly on the full generally huge label configuration space Ω , let us consider the minimization of the energy function U on a hierarchy of nested subspaces Ω^l which are composed of label configurations constrained to be blockwise constant over cells of size $2^l \times 2^l$ (*Fig. 4.*)

These nested configuration spaces ($\Omega^l \subset \Omega^{l-1} \subset \dots \subset \Omega^1 \subset \Omega^0 = \Omega$) are related to a description of the labeling at different scales. Scale 0 corresponds to the original configuration space Ω , scale l corresponds to configurations which are constant over blocks \mathcal{B}_k^l , $k = 1, \dots, N_l$, partitioning S and of size $2^l \times 2^l$ (Fig. 4)³.

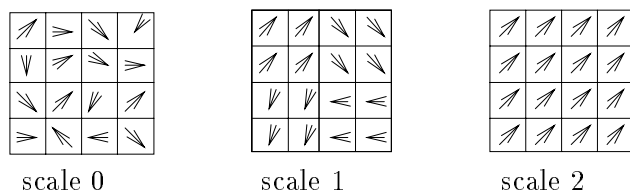


Figure 4: *Multiscale spaces of configurations: example of constrained configurations in $\Omega^0 = \Omega$ (scale 0), Ω^1 (scale 1) and Ω^2 (scale 2). Labels are represented by vectors for easier visual interpretation.*

It is easy to see that at scale l , the number of independent labels is reduced by a factor $2^l \times 2^l$. Hence a configuration $e = \{e_s, s \in S\} \in \Omega^l$ can be represented on a coarse grid S^l of size $2^{m-l} \times 2^{m-l}$ (Fig. 5). The corresponding configuration on the grid S^l will be denoted $e^l = \{e_s^l, s \in S^l\}$ (e_k^l being the common label of all the sites of block $\mathcal{B}_k^l \subset S$). These configurations e^l belong to a “reduced” configuration space Ω^l .

As a consequence, the energy of a labeling $e \in \Omega^l$ may be rewritten as a function of labels defined on S^l . It turns out that at scale l one can derive from the energy function $U(e, o)$ an equivalent coarse energy function $U^l(e^l, o)$ which is a function of the coarse labeling e^l defined on S^l :

$$U^l(e^l, o) = U_1(\Phi^l(e^l), o) + U_2(\Phi^l(e^l)). \quad (4)$$

Here Φ^l is an isomorphism that associates to a configuration e^l defined on the coarse grid S^l , the corresponding full resolution configuration in Ω^l (Fig. 5). The multiscale energy functions U^l are defined for scales $l = 0, \dots, L$.

Now it is easy to verify that if \mathcal{G} is an 8-neighborhood system, the energy function U^l at scale l can be expressed as a sum of local interaction functions associated to the same 8-neighborhood system on the coarse grid S^l . Indeed, let us consider $c \in \mathcal{C}$, an arbitrary clique associated to the fine grid energy (c contains at most 4 mutual neighboring sites, see Fig. 1). The sites in c may be either included in a $2^l \times 2^l$ constant label cell, or may sit astride 2, 3 or 4 different cells (see Fig. 6). As a consequence, the local interaction functions $V_1(e, o)$ and $V_c(e)$ associated to the original fine grid energy function can be rewritten on the coarse grid for cliques containing up to four neighboring sites on grid S^l . Hence U^l can be expressed as a sum of interaction

³Other classes of “coarse” configurations may be defined: bilinear approximations are for instance described in [39].

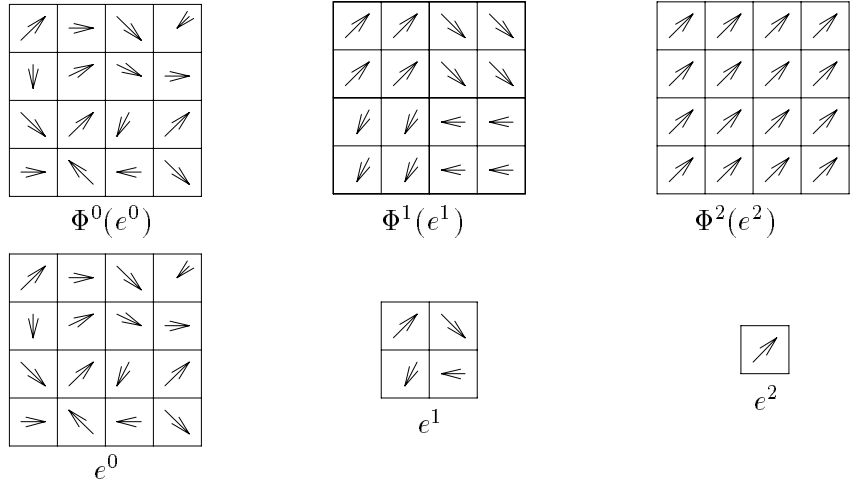


Figure 5: A constrained labeling at scale l can be represented on a coarse grid S^l . Φ^l associates to a coarse configuration e^l the corresponding configuration in Ω^l (defined at full resolution)

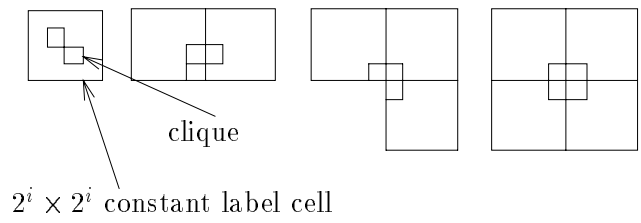


Figure 6: Example of clique locations with respect to cells with constant labels

functions depending on 1, 2, 3 or 4 site cliques associated to an 8-neighborhood system.

To exploit the sequence of multiscale energy previously defined (Eqn. 4), instead of minimizing the original energy function (Eqn. 2) over the full configuration space Ω , we consider the following sequence of optimization problems:

$$\hat{e}^l = \arg \min_{e \in \Omega^l} U(e, o), \quad l = L, \dots, 0. \quad (5)$$

By Eqn. 4, this is equivalent to the minimization of the coarse energy functions:

$$\hat{e}^l = \arg \min_{e^l \in \Gamma^l} U^l(e^l, o), \quad l = L, \dots, 0. \quad (6)$$

These optimization problems are solved using a standard coarse-to-fine multigrid strategy (Fig. 7). Starting from a coarse scale L , the optimization problem is first solved in subset Ω^L (Eqn. 5) by solving the equivalent problem (Eqn. 6). An estimate of \hat{e}^L is obtained by the ICM algorithm described previously. This defines a first (crude) solution to the original problem, and will be refined on the subsequent finer levels.

At level l , let \hat{e}^l designate the estimate obtained after the convergence of the deterministic relaxation. The algorithm at resolution level $l - 1$ is initialized by the configuration $[\Phi^{l-1}]^{-1} \circ \Phi^l(\hat{e}^l)$ which corresponds to an interpolation of e^l on the finer grid S^{l-1} using a simple repetition of the estimated labels (see Fig. 7). Note

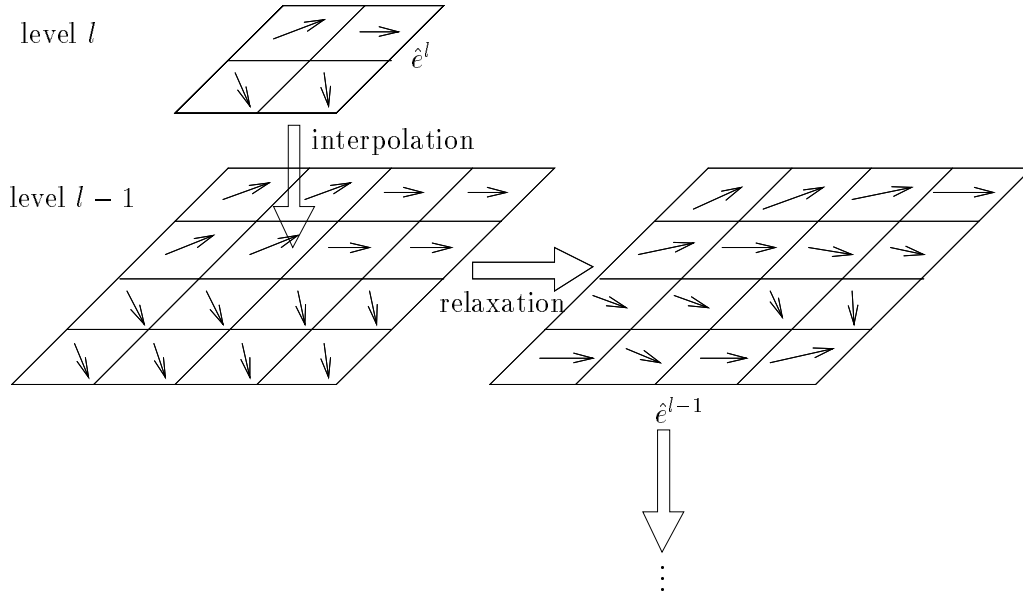


Figure 7: A coarse-to-fine strategy for optimization at scales $i = n, \dots, 0$

that this interpolation defines a consistent prolongation (or interpolation operator) from coarse space Ω^l to space Ω^{l-1} (indeed, \hat{e}^l and its interpolation at level $l-1$ are associated to the same configuration $\Phi^l(\hat{e}^l)$ at full resolution and hence have identical energies).

After interpolation, a new relaxation step is performed until convergence is obtained at scale $l-1$; this procedure is repeated at the finer scales.

We note that whereas the algorithm uses a multigrid representation of labels, only one level (full resolution) is used for the observation field o , as can be seen in Eqn. 4 (Fig. 8). The multiscale algorithm is summarized in Fig. 9.

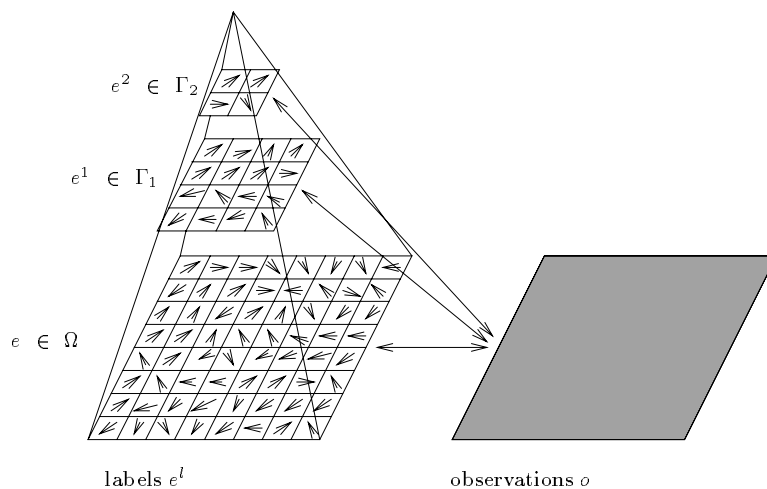


Figure 8: *The multiscale relaxation structure*

This approach has been used in several low-level vision applications, [23, 27] and tested on real world image sequences. The multiscale relaxation scheme leads to solutions that are close to the one obtained by stochastic relaxation. The updating, at each level, in the multiscale relaxation algorithm remains *local* and only involves a pixel and its neighbors. Since the algorithm converges rapidly (typically within 10 full scans of the image) this results in a highly efficient massively parallelizable algorithm.

3 Parallel implementations on SIMD 2D array

3.1 Parallel updating of the image sites

Most of the low-level image processing algorithms have properties of locality – the calculations at each image site depend only on the site itself and on surrounding sites – and regularity – the same computation has to be carried out on each pixel. These properties are synonymous with massive data parallelization.

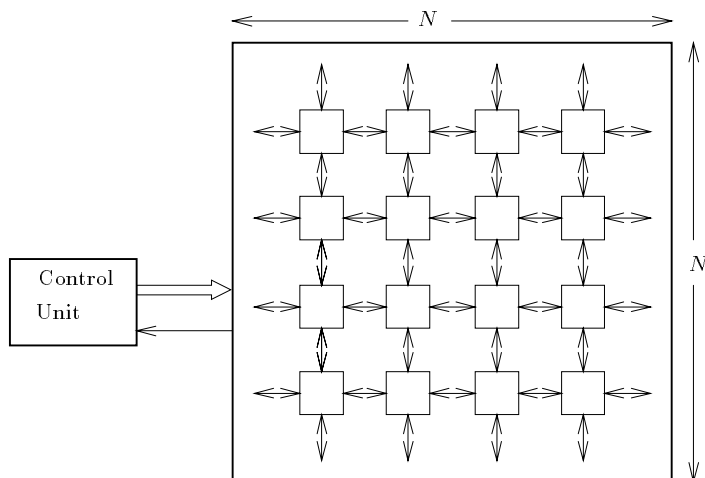


Figure 10: SIMD 2D array architecture

2D array architectures are thus well suited for doing massively parallel synchronous updatings of the state of the image field.

Before describing the different parallel relaxation algorithms which have been studied, we first present the general constraints imposed by the parallel implementation of non-linear relaxation schemes on SIMD machines.

3.1.1 Constraints in the parallel synchronous updating of the image

As stated in the previous section, the relaxation algorithms associated with low-level vision models have several appealing features. To update an image site, one simply replaces its label by a function of the site label itself and of its neighbors' labels: this updating is spatially invariant. The assignment of one pixel to each processing element, in order to update simultaneously the whole image grid, is a very attractive solution. However, the following constraint has to be satisfied: neighboring pixels should not be updated simultaneously to ensure convergence of the relaxation algorithm⁴. To take into account this constraint, the grid is usually partitioned into different sets where pixels belonging to one set may be updated simultaneously. Such a partition is called a "coding" [3]. The number of sets is, of course, neighborhood dependent, as illustrated in Figure 11 where 4 and 8-neighborhood systems are considered. Pixels belonging to the same set have the same label. We can notice that in the case of a 4-neighborhood the image is divided into two sets. Four sets are necessary when an 8-neighborhood is considered.

⁴The theoretical convergence properties of parallel versions of stochastic relaxation algorithms, which violate this constraint, are studied in [44]

1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1
1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1
1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1
1	2	1	2	1	2	1	2
2	1	2	1	2	1	2	1

1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4

(a)
(b)

Figure 11: Coding of the image plane for a 4 or 8-neighborhood structure

3.1.2 Efficient pixel allocation schemes

As explained previously, the standard allocation scheme, (one pixel per processor), used for instance in [30, 36, 45], leads to a lack of efficiency due to the updating constraint. For instance only half of the processor array would work simultaneously in the case of a 4-neighborhood. To keep as many processors as possible busy one can use a block allocation scheme, based on non overlapping blocks containing sites which should not be updated simultaneously.

A block is composed of adjacent pixels having different labels. Its size is equal to the number of sets associated to the coding. Pixels belonging to the same block are updated sequentially and are controlled by the same processor. The size of the blocks is 2x1 for a 4-neighborhood and 2x2 for an 8-neighborhood. This allocation scheme is the basis of our different parallel implementations. It is used in particular for the parallelization of the single resolution stochastic and deterministic relaxation algorithms.

3.2 Standard parallel relaxation

The first parallelization approach we consider here is a standard approach which corresponds to a straightforward implementation of relaxation methods on bidimensional SIMD arrays. The simplest method consists of associating, (at the initial resolution for the single resolution schemes and at each resolution for the multigrid scheme), one processing element per site of the grid [36]. Taking into account the constraints described in the previous section, a more efficient allocation scheme consists of assigning one processor to each block of sites that should not be updated simultaneously. In the multigrid relaxation, the different resolution levels are processed sequentially. Therefore this algorithm may be seen as successive *initialization/relaxation* steps on grids $\{S^l, l = 0, \dots, L\}$ of size $|S^l| = \frac{N^2}{2^{2l}}$, where N^2 is the

size of the full resolution image. If $b_{i,j} \subset S^l$ designates a block of sites that should not be updated simultaneously the number of processors necessary to process grid S^l is:

$$\mathcal{N}_P^l = \frac{N^2}{2^{2l}} \cdot \frac{1}{|b_{i,j}|}. \quad (7)$$

This simple implementation of multigrid methods therefore generally yields a lack of efficiency on large size SIMD architectures. It is indeed generally not possible, in such schemes, to keep all processors active. If one processor is allocated to each site of the successive grids, the efficiency is at most: $\frac{N_l}{P}$ where P is the number of processors and N_l is the size of the grid at resolution l . In the drastic case of a coarse grid of 1×1 the efficiency is lower than $1/P$.

A way to cope with this lack of efficiency is to introduce additional levels of parallelism in the multigrid relaxation methods. Several directions have been explored. They are described in the next sections.

3.3 A need for additional levels of parallelism

As explained previously, standard data parallelism does not exploit the large computing resources of the available large massively parallel processor arrays when the image grid to be controlled is small. This is particularly true for multigrid relaxation algorithms, in which coarse and fine grids have to be handled sequentially.

In this section, we consider different solutions which enable to make a full use of the large potential of data parallelism available on 2D processor arrays for the implementation of non-linear multigrid relaxation methods.

3.3.1 Full parallel multigrid relaxation

Several solutions have been proposed to introduce additional parallelism in multigrid relaxation algorithms. One method consists in designing non-linear multigrid algorithms that are completely parallel over all the resolution levels. In standard multiresolution algorithms the different levels are processed sequentially. The efficiency of these algorithms may be increased by processing all resolution levels in parallel (the idea was suggested – but not developed – by Terzopoulos in [42]). This may also be done as in [27] by extending the spatial neighborhood of a site by introducing hierarchical cliques to connect adjacent resolution levels in the multigrid structure. The extension of the neighborhood enables parallel updating of non-neighboring sites over the whole pyramid. However, this leads to a modification of the underlying model and to a very significant augmentation of the configuration space. As a consequence this technique is very time consuming [27].

Another way to break the sequential nature of multigrid relaxation algorithms is to develop specific parallel inter-level cooperation strategies. In recent studies [1, 19, 22] such relaxation algorithms have been considered. In [22] the parallel relaxation consists in running cooperative relaxation chains at each level of the pyramid. The

interactions between levels are based on the periodical exchange of configurations between chains. This strategy enables exploring the original configuration space at different resolutions in parallel in order to converge faster towards good solutions. These algorithmic solutions are close to the one obtained by a sequential coarse to fine strategy but are better suited to MIMD parallel machines [22].

3.3.2 Concurrent exploration of the configuration space

Another solution for improving the efficiency of parallel multigrid relaxation schemes is to keep the sequential character of the algorithm and to make use of the idle processors at the coarse resolutions. In [37], Narayanan *and al.* suggest a technique called “*replicated data algorithms*” to use idle processors in the parallel implementation of image processing algorithms such as histogramming or convolution on SIMD massive parallel machines. This technique consists of a replication, over the array processor, of the data structure involved in the considered algorithm. The problem is split up in functional blocks which may be processed simultaneously using address autonomy capabilities of general purpose SIMD arrays.

Another solution consists in exploring in parallel the configuration space of the original optimization problem. This principle can be implemented very efficiently with the “multiscale” algorithm described in [23] since this algorithm may itself be interpreted as a constrained optimization on different configuration spaces. We present in the following sections two parallel algorithms that perform concurrent relaxation sweeps at a given resolution: the concurrent subspaces multiscale relaxation and a multi-initialization multiscale relaxation scheme. These two methods combines two different levels of parallelism: parallel updating of the image sites and concurrent explorations of the configuration space of the problem. The cooperation between these two levels of parallelism proves to be very efficient and leads to fast convergence towards quasi-optimal solutions as will be seen on different benchmarks in Section 4. The convergence speed of these algorithms at coarse resolution levels is indeed faster than standard parallel relaxation schemes and hence they provide better initializations for the intermediate and the finest levels.

3.4 Concurrent subspaces multiscale relaxation

Let us recall that the multiscale scheme consists in computing the label field on a sequence of subspaces $\Omega^n \subset \Omega^{n-1} \subset \dots \subset \Omega^0 = \Omega$ which correspond to configurations of increasing resolution. Subspace Ω^l is the subspace of the configurations of Ω which are constant on blocks of size $2^l \times 2^l$. We may notice that there are $2^l \times 2^l$ different ways for choosing a block partition over the original grid. Consequently, at each scale l , we can build $2^l \times 2^l$ different coarse configuration spaces $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$. On space $\Omega^{l,k}$, the label configurations are constrained to be constant on the corresponding block partition $\mathcal{B}_i^{l,k}$, $i = 1, \dots, N_l$. The spaces $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$, are obtained by considering successive shifts of the block partition along the horizontal and vertical directions (see *Fig. 12* for an example at

level 1). Blocks of variable size have to be introduced at the border of the image, in order to fit into the finite image size (*Fig. 12*).

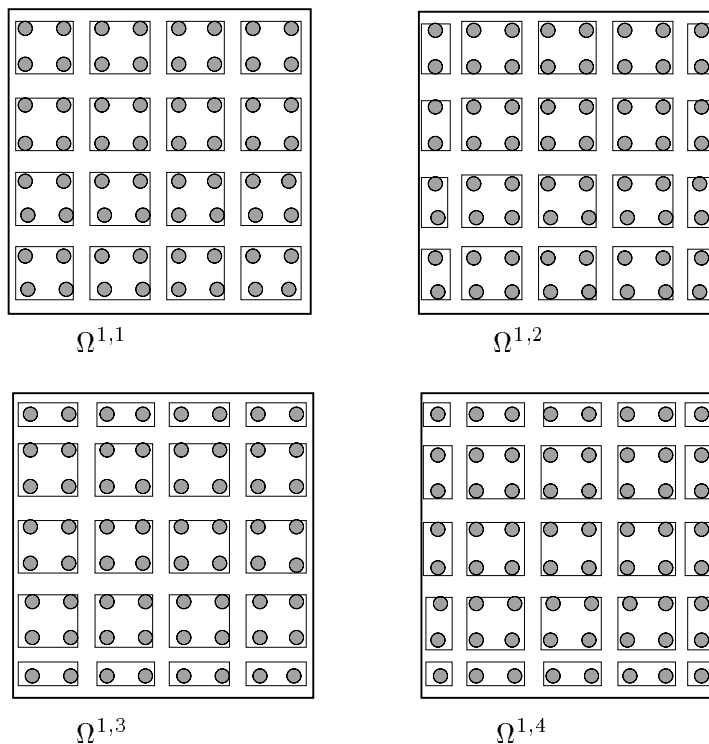


Figure 12: *The four subspaces of configurations at level $l = 1$*

Recalling that in the multiscale scheme the optimization on space Ω^l works to estimate a label field at a reduced scale on a subspace Ω^l , we can define $2^l \times 2^l$ subspaces $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$ of reduced configurations. $\{S^{l,k}, k = 1, \dots, 2^l \times 2^l\}$ is the set of coarse grids supporting the configurations belonging to $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$.

It is now possible to define a relaxation scheme in which, at each scale l , the $2^l \times 2^l$ configuration spaces $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$ are explored in parallel. The different energy functions $\{U^{l,k}, k = 1, \dots, 2^l \times 2^l\}$ are derived, with the same technique as for the multiscale scheme, from the initial energy function on space Ω .

The relaxation algorithm (*Fig. 13*) associated to this structure is a coarse to fine strategy where at each resolution level, $2^l \times 2^l$ concurrent relaxations are run in parallel on subspaces $\Omega^{l,k}$, $k = 1, \dots, 2^l \times 2^l$. The energies of the resulting configurations are then compared. The configuration of lowest energy is selected and projected on all the subspaces $\Omega^{l-1,k}$, $k = 1, \dots, 2^{l-1} \times 2^{l-1}$ associated with the next finer resolution. The projection (P^\perp) carried out is an orthogonal projection according to the euclidian distance which leads to compute the average of the labels on the different blocks (see *Fig. 14* for an example on vector labels).

CSMS Relaxation Algorithm

$\hat{e}^{l,k} \in \mathcal{S}^{l,k}$: initial configuration at resolution l on grid $S^{l,k}$;
 $\hat{e}^{l,k} \in \mathcal{S}^{l,k}$: estimated configuration at resolution l on grid $S^{l,k}$;
 Y^l : minimal energy configuration at resolution l ;
 $P_{\Gamma^{l-1,k}}^\perp(Y^l)$: orthogonal projection of the configuration Y^l on subspace $\mathcal{S}^{l-1,k}$;

Choose a set of initial configurations $\{\hat{e}^{L,1}, \dots, \hat{e}^{L,2^{2L}}\}$;
for $l = L$ (coarse resolution) **to** $l = 0$ (finest resolution)
 { **par**^a
 {
 $\hat{e}^{l,1} = \text{ICM}(\hat{e}^{l,1}, U^{l,1})$;
 \vdots
 $\hat{e}^{l,2^l \times 2^l} = \text{ICM}(\hat{e}^{l,2^l}, U^{l,2^l})$;
 }
 $Y^l = \arg \min_{e \in \{\hat{e}^{l,1}, \dots, \hat{e}^{l,2^l}\}} (U^{l,\cdot}(e^l, o))$, $Y \in \mathcal{S}^{l,k}$;
if $l \neq 0$
par
 {
 $\hat{e}^{l-1,1} = P_{\Gamma^{l-1,1}}^\perp(Y^l)$;
 \vdots
 $\hat{e}^{l-1,2^{l-1} \times 2^{l-1}} = P_{\Gamma^{l-1,2^{l-1} \times 2^{l-1}}}^\perp(Y^l)$;
 }
 }
 }

^a**par** $\{f_1; f_2; \dots; f_n\}$ denotes the parallel execution of functions f_1, f_2, \dots, f_n

Figure 13: The concurrent subspaces multiscale (CSMS) relaxation algorithm

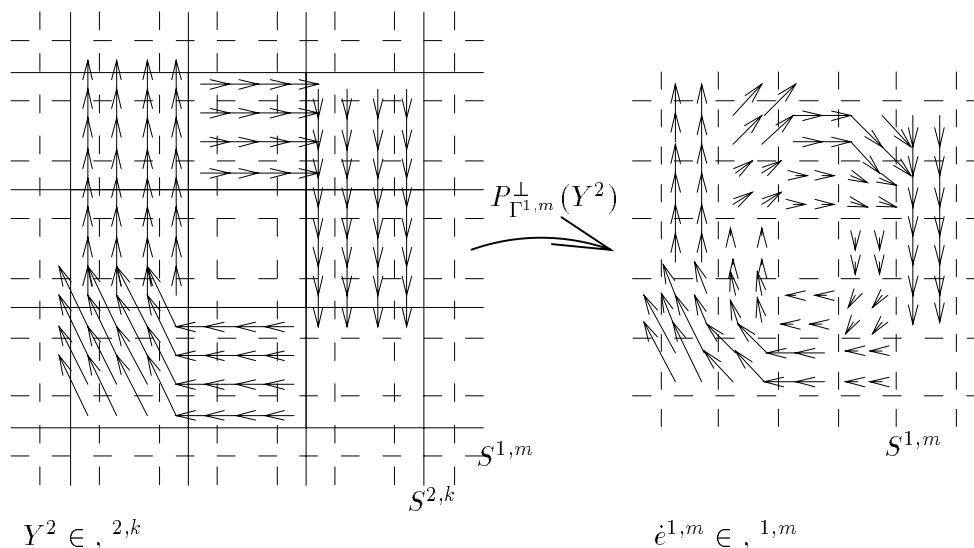


Figure 14: Example of the orthogonal projection of configuration $Y \in S^{2,k}$ onto $S^{1,m}$

At a given resolution level all the relaxations running in parallel are stopped once any of the concurrent relaxations has reached convergence – *i.e.* no longer changes the state of its labels. Other convergence criteria may be used but this one guarantees that the relaxation will be stopped early enough to yield reduced CPU times.

The full parallel multiscale scheme developed here provides an elegant way to combine concurrent relaxation chains on adjacent subspaces. Results concerning the quality of the energy minimization and computing times are given in section 4. Before analyzing these results, we present another algorithm which also leads to full parallelism.

3.5 Multi-initialization multiscale relaxation

It is well known that the ICM relaxation algorithm [4], used at each level in the multiscale scheme, gets easily stuck in local minima of the energy function. As a consequence, the choice of the initial configuration in such deterministic minimization methods is often crucial. It is conjectured, [21, 43] that multiresolutional representations of information (such as in the multiscale scheme) leads to “smoother” energies at coarse resolutions⁵. This does not however guarantee that the energy function has only one minimum. It is therefore appealing to propose different initializations in parallel at the coarsest resolution. This leads to a different parallel relaxation strategy on the pyramidal structure.

⁵A multiresolution representation allows large displacements in the configuration space. It enables therefore, to remove minima associated to local displacements in that space.

MIMS Relaxation Algorithm

$\left\{ \begin{array}{ll} \hat{e}^{l,k} \in \cdot^{l,k} & : \text{initial configuration at resolution } l \text{ on grid } S^{l,k} ; \\ \hat{e}^{l,k} \in \cdot^{l,k} & : \text{estimated configuration at resolution } l \text{ on grid } S^{l,k} ; \\ Y^{l,i} & : \text{minimal energy configuration in the sub-group } G^{l,i} ; \\ [\Phi^{l-1}]^{-1} \circ \Phi^l(Y^{l,i}) & : \text{projection of the configuration } Y^{l,i} \text{ on the subspace } \cdot^{l-1} ; \end{array} \right.$

Choose the set of initial configurations $\{\hat{e}^{L,1}, \dots, \hat{e}^{L,2^L \times 2^L}\}$

for $l = L$ (coarse resolution) **to** $l = 0$ (full resolution)

{ par

{

$\hat{e}^{l,1} = \text{ICM}(\hat{e}^{l,1}, U^l) ;$

\vdots

$\hat{e}^{l,2^{2l}} = \text{ICM}(\hat{e}^{l,2^{2l}}, U^l) ;$

}

$Y^{l,i} = \arg \min_{e^{l,k} \in G^{l,i}} (U^l(\hat{e}^{l,k}, o)), i = 1, \dots, 2^{2l-2} ;$

if $l \neq 0$

par

{

$\hat{e}^{l-1,1} = [\Phi^{l-1}]^{-1} \circ \Phi^l(Y^{l,1}) ;$

\vdots

$\hat{e}^{l-1,2^{l-1} \times 2^{l-1}} = [\Phi^{l-1}]^{-1} \circ \Phi^l(Y^{l,2^{2l-2}}) ;$

}

}

Figure 15: *The multiple initialization multiscale relaxation (MIMS) algorithm*

In the case of multiscale relaxation this strategy can be described as follows. At level l of the multiscale scheme, $2^l \times 2^l$ relaxation chains may be run on the same subspace \mathcal{V}^l with different initial configurations. These relaxation chains are run on $2^l \times 2^l$ replicated grids $S^{l,k}$, $k = 1, \dots, 2^l \times 2^l$, which all support configurations belonging to \mathcal{V}^l . At level $l-1$, $2^{l-1} \times 2^{l-1}$ final configurations must be selected from the $2^l \times 2^l$ resulting configurations at the upper level to initialize the next relaxation chains. In other words, one configuration among four must be selected and projected on resolution level $l-1$. To this end, the $2^l \times 2^l$ configurations generated at level l are clustered in subgroups $G^{l,i}$, $i = 1, \dots, 2^{l-1} \times 2^{l-1}$ of four configurations. The lower energies configurations among all these subgroups are selected and projected on the next finer level.

The different steps of the multi-initialization algorithm (initialization, choice of the best configuration of a cluster, projection) are iterated up to the finest resolution. At full resolution only one chain is running on the whole processor array and one single solution is produced. The *Fig. 15* describes a general summary of the resulting relaxation algorithm.

The projection used here is the same as in the standard parallel multiscale relaxation scheme. As in the previous relaxation scheme the convergence criteria is satisfied if one of the relaxation chain reaches convergence.

4 Experimental results

The efficiency and the generality of the parallel algorithmic framework described in the previous section is illustrated here on two global optimization problems related to two different energy functions and applications: optical flow computation and image restoration.

The new parallel relaxation algorithms are compared to the standard parallel implementation of the considered relaxation algorithms.

4.1 Optical flow estimation

We have first applied the different parallelization methods developed in the previous section to a model of optical flow measurement. We consider a discrete formulation of the optical flow measurement problem which has been used in [29]. This model is known to lead to a complex energy landscape exhibiting numerous local minima [29]. Hence, it is a good benchmark for comparing the performances of global optimization algorithms.

Let $f_t(s)$ denote the observed intensity function at point $s = (x, y)$, $s \in S$, t denotes the time variable. The velocity vector at point s is denoted $\vec{\omega}(s)(u_s, v_s)$, $u_s = \frac{dx}{dt}(s)$, $v_s = \frac{dy}{dt}(s)$ and $\vec{\omega} = \{\vec{\omega}(s), s \in S\}$. Velocities are defined on the same grid S as the pixels and the velocities are discretized according to a discrete state space $\mathcal{W} = (-u_{max} : u_{max}, -v_{max} : v_{max})$ with step size δ . The model is associated with an 8-neighborhood system (*Fig. 1*), where only the two element cliques (doubletons) are considered. The model is specified by the following energy

function:

$$\begin{aligned}
 U(f, \vec{\omega}) &= \sum_{s \in S} \{ f_t(s) - f_{t+dt}(s + \vec{\omega}(s).dt) \}^2 \\
 &+ \alpha^2 \sum_{(s,r) \text{ neighbors}} \| \vec{\omega}(s) - \vec{\omega}(r) \|^2
 \end{aligned} \tag{8}$$

The first term in the energy (known as the “*displaced frame difference*”) expresses the constant brightness assumption for a physical point over time. This term is non-linear and leads to a highly non-convex energy function [23, 29]. The second term can be interpreted as a regularization term which favors smooth solutions. The parameter α controls the relative weight of the two terms. This approach to long-range motion estimation relates to standard methods based on the matching of iconic structures.

The different parallel relaxation schemes described previously have been implemented on a MASPAP MP1 parallel computer. The MASPAP MP1 parallel computer is a massively parallel machine of 32×32 processors interconnected on a two dimensional grid. Each processor is locally connected to its eight neighbors. It is a load/store arithmetic processing element with its own local memory (16 KBYTES of RAM) and sixteen 32 bit registers. The array is controlled by a single flow of instruction (SIMD). The array controller (named ACU) is a processor with its own memory and registers. It has a data memory capacity of 128 KBYTES and 1 MBYTES of RAM.

In our experiments, five different algorithms have been implemented and compared:

1. a standard parallel implementation of a single-resolution deterministic relaxation algorithm (the “Iterated Conditional Mode” (ICM) algorithm described in Section 2.2.2);
2. a standard parallel implementation of a single-resolution stochastic relaxation (SR) algorithm based on the Gibbs sampler (described in Section 2.2.1);
3. a standard parallel implementation of the multiscale relaxation method (MS) as described in Section 3.2;
4. the parallel concurrent subspaces multiscale relaxation algorithm (CSMS) described in Section 3.4;
5. the parallel multi-initialization multiscale relaxation algorithm (MIMS) described in Section 3.5.

The “standard” parallel implementation corresponds here to a parallel updating of the image sites relying on a classical coding technique, as described in Section 3.

The temperature schedule used in the stochastic relaxation algorithm was : $T(j) = T_0.A^j$, with $A = 0.98$ and $T_0 = 1000$, where j designates the number of sweeps on the image. The same parameters were chosen in every case for the finest resolution model. Three resolution levels were considered in the multiscale methods. A discrete label space $\mathcal{W} = (-4 : +4, -4 : +4)$ with a step size of $\delta = 1$

has been adopted at the full resolution.

The parallel relaxation algorithms have been compared on different typical image sequences. A first benchmark composed of 41 short sequences of two 64×64 frames has been obtained by applying a complex motion field (*Fig. 16*) on different real images (*Fig. 17*). The motion field includes a translation, a rotation, a dilation and more complex movements (*Fig. 16*). This benchmark is a difficult example since different moving regions are present simultaneously in the scene.

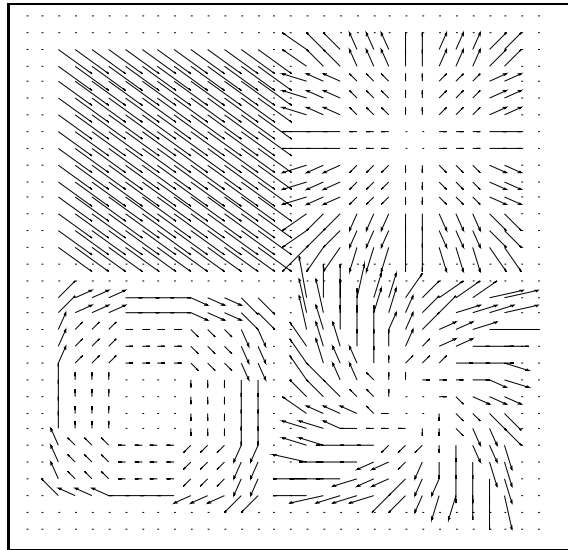


Figure 16: *Synthetic motion field applied on the 41 sequences benchmark*

The second benchmark is a real-world sequence of 48 frames that shows a traffic scene observed from a bridge (*Fig. 18*). These scenes involve several moving cars with different apparent velocities and sizes in the image plane.

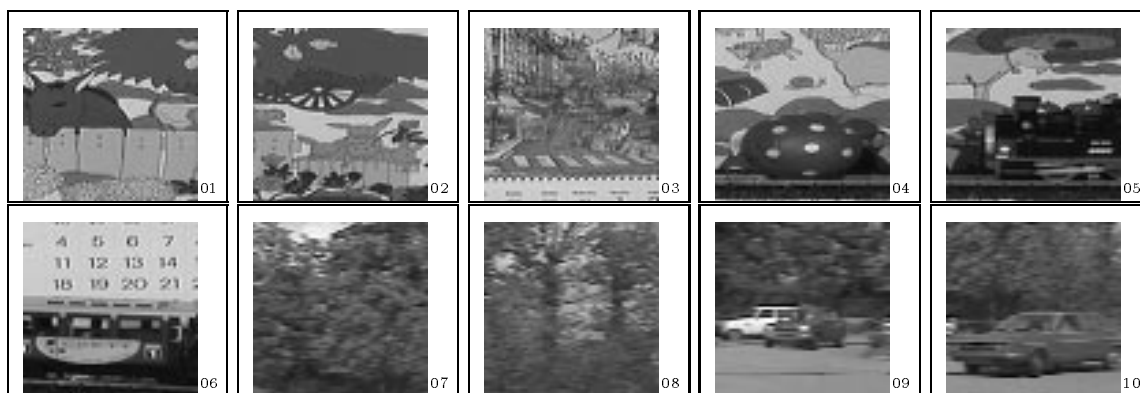


Figure 17: *First images of the 41 short sequences benchmark*



Figure 18: *Non-successive frames of the "Highway" sequence*

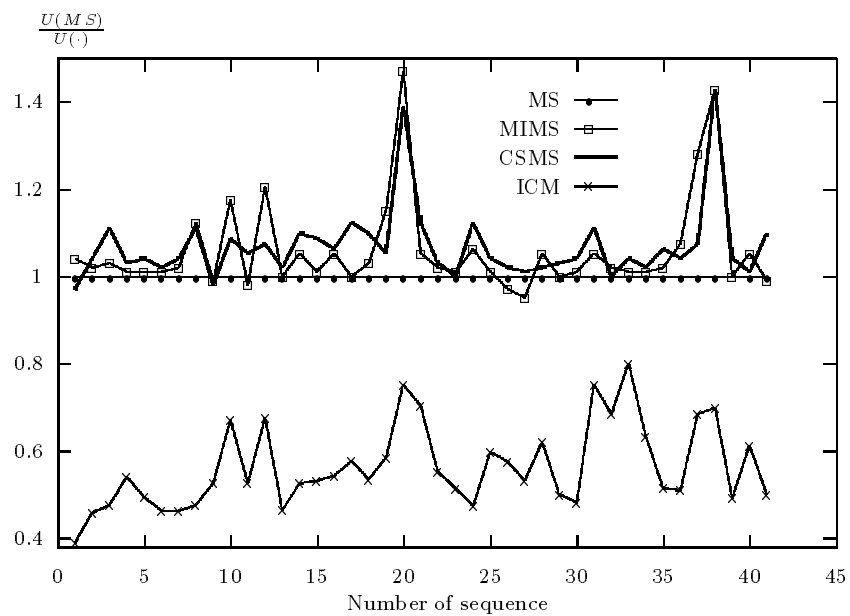


Figure 19: Ratio of the final energy level $U(MS)$ reached by the standard parallel multiscale implementation to the energy levels $U(\cdot)$ reached by the parallel ICM, CSMS and MIMS algorithms - (41 sequences benchmark).

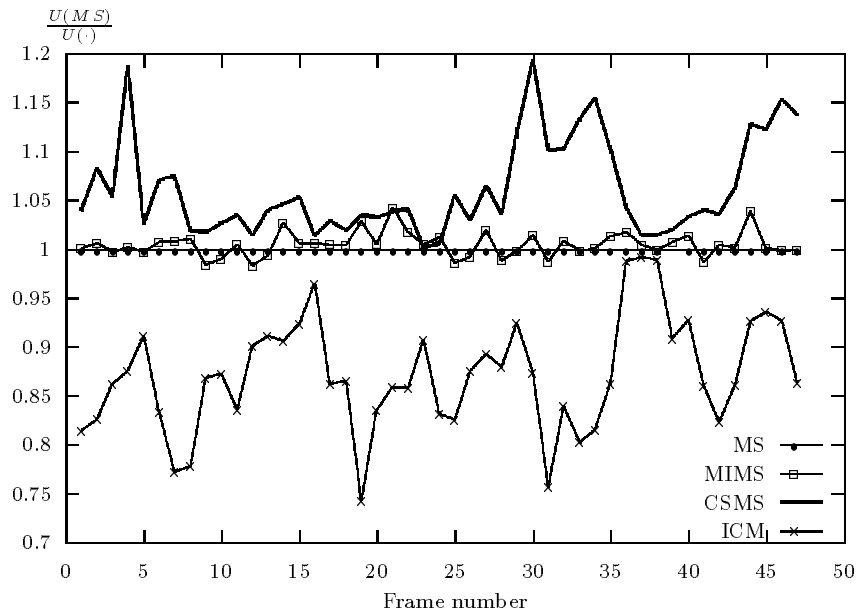


Figure 20: *Ratio of the final energy level $U(\text{MS})$ reached by the standard parallel multiscale implementation to the energy levels $U(\cdot)$ reached by the parallel ICM, CSMS and MI algorithms - (Highway sequence).*

The different parallel algorithms have been run with the same random initialization, in order to evaluate the robustness of the different methods in situations (which are usual in applications) in which no relevant initialization is available.

The global behavior of the different algorithms can be appreciated by considering the energy plots reported in Fig. 20 and 22 for the *Highway* sequence, and Fig. 19 and 21 for the 41 synthetic sequences benchmark. The curves in Fig. 19 and 20 represent the ratio of the energy level reached by the standard parallel multiscale method (MS) to the energy at convergence for the other parallel multiscale schemes (CSMS, MIMS). The plot obtained for the ICM algorithm is also reported here for comparison purposes. A second family of energy plots (Fig. 21 and 22) describes the behavior of the multiscale schemes in comparison with stochastic relaxation (SR) which theoretically should lead to the lowest energy⁶.

⁶This is true if the annealing schedule is slow enough, i.e. logarithmic, which is not the case here.

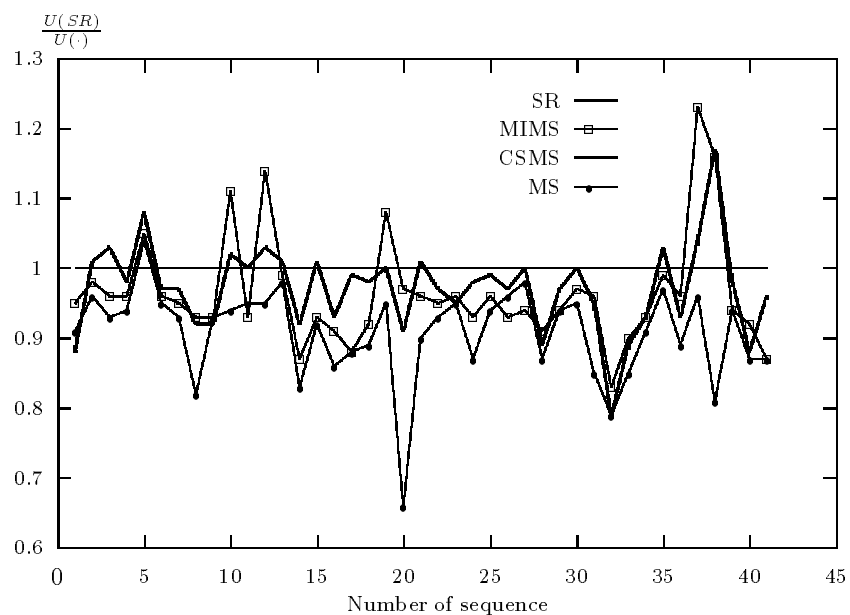


Figure 21: Ratio of the final energy level $U(SR)$ reached by the parallel stochastic relaxation to the energy levels $U(\cdot)$ reached by the parallel multiscale schemes (MS, CSMS and MIMS) - (41 sequences benchmark).

	<i>41 sequences benchmark</i>	<i>Highway sequence</i>
SR	<i>7 mn 27 s</i>	<i>7 mn 24 s</i>
MS	<i>5 s 68</i>	<i>4 s 79</i>
CSMS	<i>4 s 95</i>	<i>4 s 35</i>
MIMS	<i>4 s 79</i>	<i>4 s 29</i>
ICM	<i>4 s 20</i>	<i>4 s 26</i>

Table 1: *Everage CPU time for the different parallel algorithms – optical flow computation (Highway sequence and 41 sequences benchmark)*

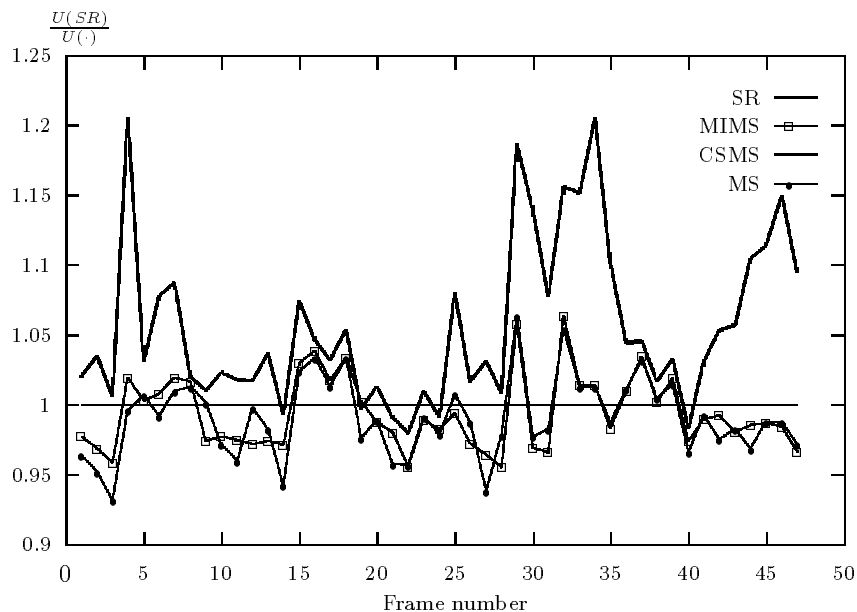


Figure 22: *Ratio of the final energy level $U(\text{SR})$ reached by the parallel stochastic relaxation to the energy levels $U(\cdot)$ reached by the parallel multiscale schemes (MS, CSMS and MIMS) algorithms - (Highway sequence).*

As expected the deterministic ICM relaxation algorithm gets usually stuck in high energy minima. The parallel CSMS and MIMS algorithms improve significantly the results of the standard parallel multiscale implementation (MS). In the case of the 41 sequences benchmark the concurrent subspaces multiscale scheme performs slightly better in average than the multi-initialization relaxation algorithm (*Fig. 19*). The solutions produced by the MIMS and CSMS schemes are very close – in term of energy levels – to those obtained by stochastic relaxation. Let us notice that in some cases the configurations generated by CSMS and MIMS are even better – i.e. of lower energy levels – than those estimated by time consuming stochastic relaxation.

In the case of the *Highway* sequence the standard multiscale and the multi-initialization schemes lead to comparable results (*Fig. 20*), whereas the concurrent

subspaces scheme improves significantly the quality of the computed solutions. The ICM algorithm – which is one of the most used in MRF applications – yields poor configurations corresponding to local minima associated with high energy values. The energy levels reached by MS and MIMS are of the same order as those observed for stochastic relaxation. It is worth noticing here that the concurrent subspaces relaxation scheme (CSMS) leads to even lower energy levels than stochastic relaxation (SR) ! This is relevant since the annealing schedule used for the SR algorithm is not optimal. A slower annealing schedule could of course have been introduced for the stochastic relaxation, but this would have increased the cpu time of this already time consuming method.

The average CPU times⁷ (computed over the whole sequence) are given in table 1. As can be seen, the CSMS and MIMS scheme are faster on an average than the standard multiscale scheme although they have a higher cost per iteration. As expected these two algorithms show a faster convergence at coarse resolutions that is due to the competition between several relaxation chains which enable faster convergence to lower energy solutions. On average, the parallel multiscale relaxation methods lead to a computational saving on the order of 90, with respect to parallel stochastic relaxation. This is noteworthy since the final solutions are very close, both on a qualitative and a quantitative point of view (close energy levels).

Examples of computed optical flow fields are given in *Fig. 23* for the 41 synthetic sequences benchmark. The quality of the different resulting fields readily shows the gain obtained by the parallel exploration of the space of solutions in the MIMS and CSMS method.

4.2 Noisy image restoration

To bring additional experimental evidence, the different parallel relaxation algorithms have been applied to a second standard model, used in low-level vision for noisy image restoration. We consider here a simplified version of a model introduced by Geman and Reynolds in [17].

Let $X = \{x_s, s \in S\}$ denotes the intensity field of the noise-free image. The observed noisy image $Y = \{y_s, s \in S\}$, is such that $Y = X + N$ where N is a 2D white Gaussian noise, independent of X , with zero mean and variance σ^2 . Under this assumption the adequate energy function is easily derived [17]:

$$U(x, y) = \frac{1}{2\sigma^2} \sum_{s \in S} (x_s - y_s)^2 - \sum_{\{s,r\} \in C} \frac{1}{1 + |x_s - x_r|} \quad (9)$$

The first component of the energy function defines a quadratic distance between the the unknown (noise-free) image X and the noisy observations image Y . The second term specifies the *a priori* properties of the desired solution. This term favors

⁷A sequential implementation of the RS, ICM and MS relaxation methods described in [23] on a Sparc ss1 station has led to the following average CPU times : 184 mn 11 s for SR, 7 mn 25 s for ICM and 3 mn 13 s for MS.

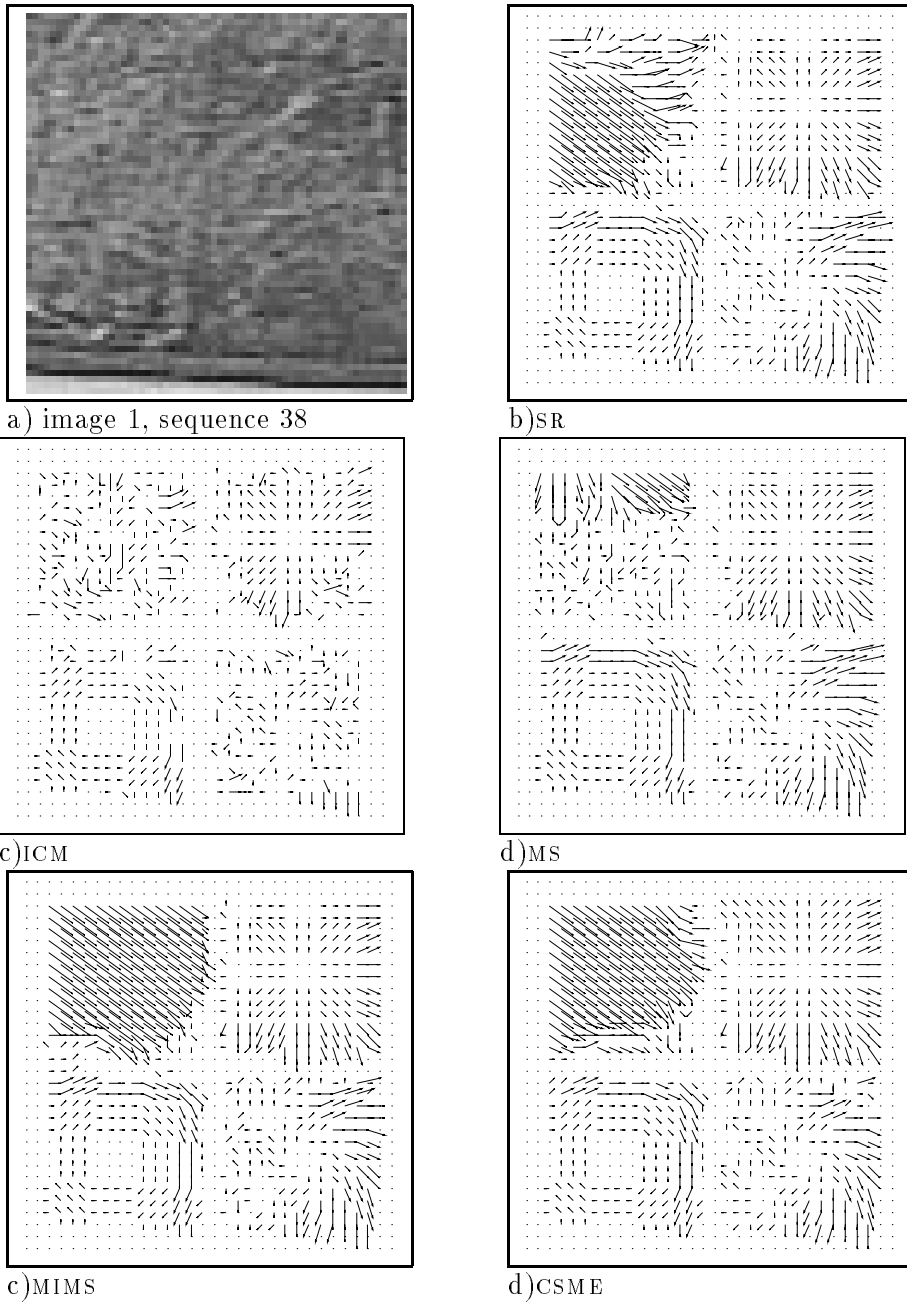


Figure 23: *Computed optical flow fields (sequence 38 of the benchmark)*

piecewise smooth solutions and enables to take into account intensity discontinuities without introducing auxiliary variables to model edges (i.e. line process) [18, 21].

As done previously, the different parallel relaxation algorithms have been run on the MASPAR MP1 parallel computer on a benchmark composed of 30 images of size 64×64 . This benchmark has been obtained by adding a white Gaussian noise to different original images with a SNR of 5 dB (see Fig. 24).

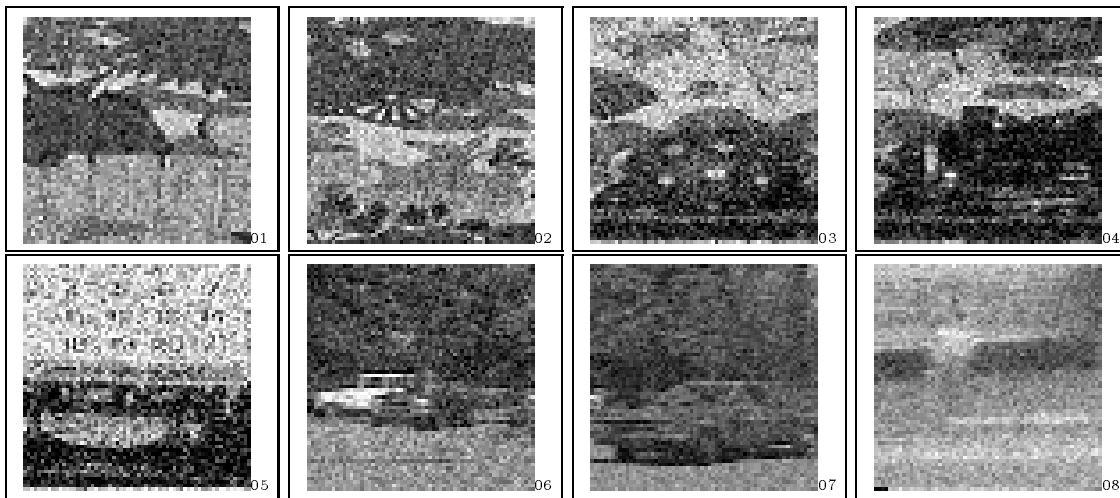


Figure 24: *First image of the noisy images benchmark*

The different parallel relaxation algorithms have been run with the same parameter ($2\sigma^2 = 1000$). The other parameters were the same as in the previous experiment.

Figure 25 shows the ratio of the final energies reached by the parallel multiscale relaxation algorithms (MS, CSMS, MIMS) and the parallel ICM algorithm, to the energy reached by parallel stochastic relaxation SR (notice that the final energies are here negative).

As in the case of the optical flow estimation model, the MIMS and CSMS relaxation algorithms have nearly the same behavior. These two algorithms reach energy levels that are of the same order as those obtained by stochastic relaxation at a reduced CPU cost (see table 2). On average, the gain here is on the order of a factor of 20 for similar (and even better) qualitative and quantitative results. The ICM algorithm leads in all the cases to results of lower quality. An example of restored images obtained by the different relaxation algorithms is presented in *Fig. 26*. The qualitative enhancement is, as can be seen on this figure, closely related to the energy levels reached by the algorithms. High energies correspond to poor solutions (this is the case for the ICM algorithm). The low energies levels reached by stochastic relaxation and parallel multiscale relaxation are related to restorations of better quality.

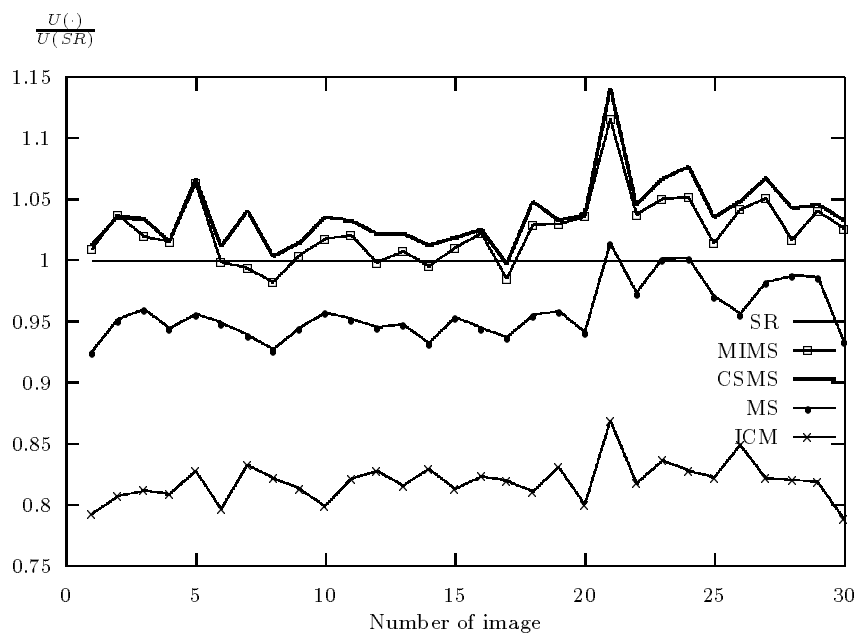


Figure 25: *Ratio of the final energy level $U(\cdot)$ reached by the parallel multiscale schemes (MS, CSMS, MIMS) and the ICM algorithm to the energy reached by stochastic relaxation $U(SR)$. - (Noisy images benchmark).*

SR	MS	ICM	CSMS	MIMS
11 mn 37 s	39 s 70	34 s 08	33 s 39	32 s 60

Table 2: *Average CPU time for the different algorithms – image restoration*

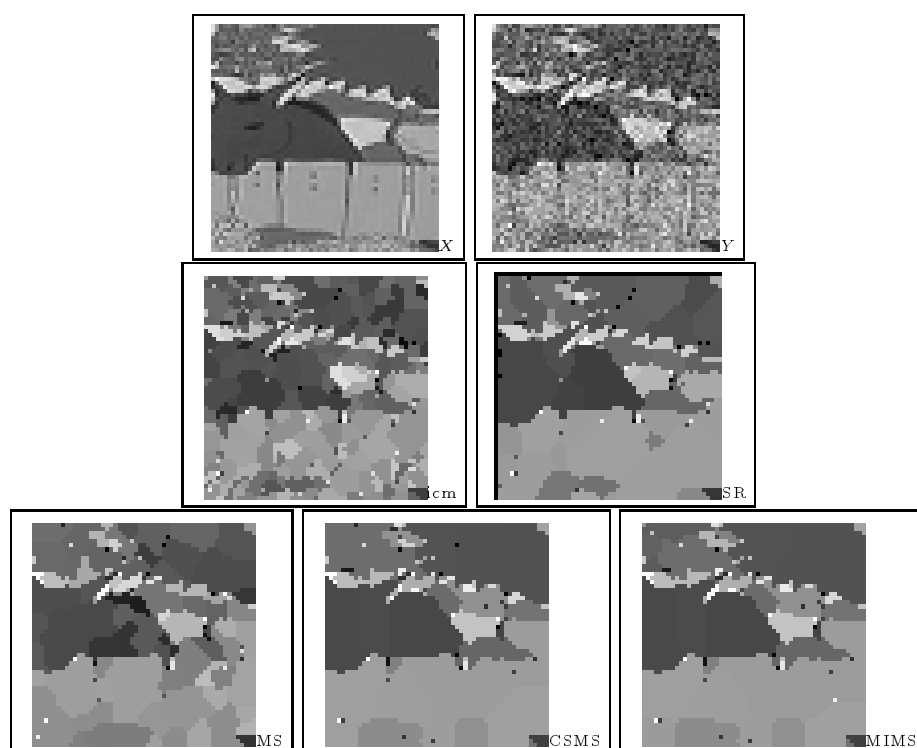


Figure 26: Results produced by the different algorithms on image 1 of the benchmark

5 Conclusion

In this paper, the parallel implementation of several general non-linear relaxation algorithms, widely used in low-level vision applications, has been considered. Relaxation algorithms, based on the modeling of local interactions between variables in the image plane, have locality and regularity properties that make them good candidates for implementation on bi-dimensional SIMD massively parallel processor arrays. Particular attention has been paid here to recently developed non-linear multigrid relaxation algorithms [23] which have shown very attractive convergence properties (fast convergence towards quasi-optimal solutions). These algorithms are however sequential by nature, each grid having to be processed in turn. As a consequence, at coarse scales (corresponding to small image grids), standard data parallelism does not exploit the large computing resources of large massively parallel processor array.

A new algorithmic framework that enables exploiting all the computing resources of large 2D SIMD processor arrays has been proposed. Different levels of parallelism have been considered to avoid idle processors at coarse scales in the multigrid processing. This technique yields a full efficiency of the array and significantly improves the performances (convergence speed and quality of the solution) of the relaxation process. Gains of up to two orders of magnitudes in the convergence speed with respect to parallel stochastic relaxation have been observed with these new parallel multigrid algorithms. The final solutions are close (and sometimes better) in quality to those obtained using time-consuming stochastic relaxation algorithms.

As a conclusion, we have shown that an efficient (full) utilization of 2D processor arrays enables the development of new parallel algorithms that not only provide speed-up but also better and more robust solutions than their sequential counterparts. The contribution of parallelism in image processing should therefore be considered both from the point of view of algorithmic improvement and cpu-time reduction.

Acknowledgments *This work has been partly supported by MRE (French Ministry of Research and Technology), Peugeot-Citroën and Renault in the context of the EUREKA PROMETHEUS project (ProLab2), by the GDR TdSI 134 and by Brittany County Council under contribution to student grant.*

References

- [1] E.H.L. AARTS and P.J.M. van LAARHOVEN. – *Simulated Annealing: Theory and Applications*. – D. Reidel Publishing Company, 1987.
- [2] S. BARNARD. – Stereo matching. – In R. CHELLAPPA and A. JAIN, editors, *Markov Random Fields Theory and Application*, pages 245–271. Academic Press, 1993.
- [3] J. BESAG. – Spatial interaction and the statistical analysis of lattice systems. – *J. Royal Statist. Soc.*, Serie B, Vol. 36: pages 192–236, 1974.
- [4] J. BESAG. – On the statistical analysis of dirty pictures. – *J. Royal Statist. Soc.*, Vol. 48, Serie B, No 3: pages 259–302, 1986.
- [5] M. J. BLACK. – *Robust incremental optical flow*. – PhD thesis, Yale university, Sept. 1992.
- [6] A. BLAKE and A. ZISSERMAN. – *Visual Reconstruction*. – MIT Press, 1987.

-
- [7] C. BOUMAN and B. LIU. – Multiple resolution segmentation of textured images. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No 2: pages 99–113, Feb. 1991.
- [8] P. BOUTHEMY and E. FRANÇOIS. – Motion segmentation and qualitative dynamic scene analysis from an image sequence. – *Int. J. Computer Vision*, Vol. 10, No 2: pages 157–182, 1993.
- [9] P. BOUTHEMY and P. LALANDE. – Recovery of moving object masks in an image sequence using local spatio-temporal contextual information. – *Optical Engineering*, Vol. 32, No 6: pages 1205–1212, June 1993.
- [10] B. CHALMOND. – An iterative Gibbsian technique for reconstruction of m-ary images. – *Pattern Recognition*, Vol. 22, No 6: pages 747–761, 1989.
- [11] R. CHELLAPPA and A.K. JAIN, editors. – *Markov Random Fields : Theory and Applications*. – Academic Press, 1993.
- [12] P.B. CHOU and C.M. BROWN. – The theory and practice of bayesian image modeling. – *Int. J. Comp. Vis.*, Vol. 4: pages 185–210, 1990.
- [13] H. DERIN and C.S. WON. – A parallel image segmentation algorithm using relaxation with varying neighborhoods and its mapping to array processor. – *Computer Vision, Graphics, Image Processing*, Vol 40: pages 54–78, 1987.
- [14] R.C. DUBES and A.K. JAIN. – Random field models in image analysis. – *J. Applied Statistics*, Vol. 16, No 2: pages 131–163, 1989.
- [15] W. ENKELMANN. – Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. – *Comput. Vision, Graphics, Image Processing*, Vol. 43: pages 150–177, 1988.
- [16] D. GEMAN, S. GEMAN, C. GRAFFIGNE, and D. PONG. – Boundary detection by constrained optimization. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, No 7: pages 609–628, July 1990.
- [17] D. GEMAN and G. REYNOLDS. – Constrained restoration and the recovery of discontinuities. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, No 3: pages 367–383, March 1992.
- [18] S. GEMAN and D. GEMAN. – Stochastic relaxation, Gibbs distributions and the bayesian restoration of images. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, No 6: pages 721–741, November 1984.
- [19] C. GRAFFIGNE. – Parallel annealing by periodically interacting multiple searches : an experimental study. – In R. Azencoot, editor, *Simulated Annealing : parallelization techniques*, pages 47–79, New-York, 1992. John Wiley & Sons.
- [20] W. HACKBUSCH, editor. – *Multi-grid methods and applications*. – Springer series in computational mathematics, Berlin, 1985.
- [21] F. HEITZ and P. BOUTHEMY. – Multimodal estimation of discontinuous optical flow using Markov Random Fields. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 15, No 12: pages 1217–1232, December 1993.
- [22] F. HEITZ, E. MEMIN, P. PEREZ, and P. BOUTHEMY. – Parallel relaxation algorithms for the statistical analysis of visual motion. – In *IEEE 7th Workshop on Multidimensional Signal Processing*, Lake Placid, Sept. 1991.
- [23] F. HEITZ, P. PEREZ, and P. BOUTHEMY. – Multiscale minimization of global energy functions in some visual recovery problems. – *CVGIP : Image Understanding*, Vol. 59, No 1: pages 125–134, January 1994.
- [24] G. F. HINTON, T. J. SEJNOWSKY, and D. H. ACKLEY. – Boltzmann machines: constraint Satisfaction networks that learn. – Technical Report CMU-CS-84-19, Carnegie-Mellon University, 1984.
- [25] J. HUTCHINSON, C. KOCH, J. LUO, and C. MEAD. – Computing motion using analog and binary resistive networks. – *Computer*, Vol. 21: pages 52–63, March 1988.

- [26] F. C. JENG, J. W. WOODS, and S. RASTOGI. – Compound Gauss-Markov Random Fields for parallel image processing. – In R. CHELLAPPA and A. JAIN, editors, *Markov Random Fields Theory and Application*, pages 11–38. Academic Press, 1993.
- [27] Z. KATO, M. BERTHOD, and J. ZERUBIA. – Multiscale Markov random field models for parallel image classification. – In *Proc. Int. Conf. Computer Vision*, pages 253–257, Berlin, May 1993.
- [28] S. KIRKPATRICK, C.D. GELATT, and M.P. VECCHI. – Optimization by simulated annealing. – *Science*, Vol. 220: pages 671–680, 1983.
- [29] J. KONRAD and E. DUBOIS. – Bayesian estimation of motion vector fields. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, No 9: pages 910–927, 1992.
- [30] S.Y. KUNG. – *VLSI array processors*. – Prentice Hall, Englewood Cliffs, New Jersey, USA, 1988.
- [31] B. S. MANJUNATH, T. SIMCHONY, and R. CHELLAPPA. – Stochastic and deterministic networks for texture segmentation. – *IEEE trans. on ASSP*, Vol 38, No 6: pages 1039–1049, 1990.
- [32] C. MEAD. – *Analog VLSI and Neural Systems*. – Addison-Wesley, 1989.
- [33] E. MEMIN. – *Parallel algorithms and architectures for Markov random field based image analysis – in french*. – PhD thesis, University of Rennes I, Juin 1993.
- [34] E. MEMIN, F. CHAROT, and F. HEITZ. – Parallel architecture for multiscale Markov Random Field-based image analysis. – In *Computer Architecture for Machine Perception (CAMP'91)*, pages 309–320, Paris, France, Dec. 1991.
- [35] J.W. MODESTINO and J. ZHANG. – A Markov Random Field model-based approach to image interpretation. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, No 6: pages 606–615, June 1992.
- [36] D.W. MURRAY, A. KASHKO, and H. BUXTON. – A parallel approach to the picture restoration algorithm of Geman and Geman on a SIMD architecture. – *Image and Vision Computing*, Vol. 4, No 3: pages 133–142, August 1986.
- [37] P.J. NARAYANAN and L.S. DAVIS. – Replicated data algorithm in image processing. – *CVGIP : Image Understanding*, Vol. 56, No 3: pages 351–365, Nov. 1992.
- [38] N. M. NASRABADI and C. Y. CHOO. – Hopfield network for stereo vision correspondence. – *IEEE trans. on Neural Network*, Vol. 3, No. 1: pages 5–13, 1992.
- [39] P. PÉREZ and F. HEITZ. – Multiscale markov random fields and constrained relaxation in low level image analysis. – In *International Conference on Acoustics, Speech and Signal Processing*, pages 61–64, Vol. 3, San Francisco, Mar. 1992.
- [40] T. POGGIO and D. WEINSHALL. – The MIT vision machine: progress in the integration of vision modules. – In R. CHELLAPPA and A. JAIN, editors, *Markov Random Fields Theory and Application*, pages 447–470. Academic Press, 1993.
- [41] POGGIO, T., TORRE, V., and KOCH, C. – Computational vision and regularization theory. – In M.A. Fischler and O. Firschein, editors, *Readings in Computer Vision : Issues, Problems, Principles and Paradigms*, pages 638–643. Morgan Kaufmann, 1987.
- [42] D. TERZOPOULOS. – *Multiresolution computation of visible surface representations*. – PhD thesis, MIT, January 1984.
- [43] D. TERZOPOULOS. – Image analysis using multigrid relaxation methods. – *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 8, No 2: pages 129–139, March 1986.
- [44] A. TROUVE. – Massive parallelization of simulated annealing: a mathematical study. – In R. AZENCOTT, editor, *Simulated annealing: parallelization techniques*. J. Wiley & Sons, 1992.
- [45] J. ZERUBIA and F. PLOYETTE. – Parallel algorithms for low-level vision on a Connection Machine CM2. – In *Proc. 13th Imacs World Congress on Computation and Applied Mathematics*, pages 763–765, Dublin, July 1991.

- [46] Y. ZHOU, R. CHELLAPPA, A. VAID, and B. K. JENKINS. — Image restoration using a neural network. — *IEEE trans. on ASSP*, Vol 36, No 7: pages 1141–1151, 1988.



Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399