

**Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.**



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Review article

Multilabel partition moves for MRF optimization[☆]A. Shabou^{a,*}, J. Darbon^{b,1}, F. Tupin^a^a Institut TELECOM, TELECOM ParisTech, CNRS LTCI, France^b CMLA, ENS Cachan, CNRS, PRES UniverSud, France

ARTICLE INFO

Article history:

Received 13 May 2010

Received in revised form 2 September 2011

Accepted 19 October 2012

Keywords:

Markov random fields

Graph-cut

Approximate optimization

Image restoration

Multichannel InSAR phase unwrapping

ABSTRACT

This paper presents new graph-cut based optimization algorithms for image processing problems. Popular graph-cut based algorithms give approximate solutions and are based on the concept of *partition move*. The main contribution of this work consists in proposing novel partition moves called *multilabel* moves to minimize Markov random field (MRF) energies with convex prior and any likelihood energy functions. These moves improve the optimum quality of the state-of-the-art approximate minimization algorithms while controlling the memory need of the algorithm at the same time. Thus, the two challenging problems, improving local optimum quality and reducing required memory for graph construction are handled with our approach. These new performances are illustrated on some image processing experiments, such as image restoration and InSAR phase unwrapping.

© 2012 Elsevier B.V. All rights reserved.

Contents

1. Introduction	14
2. Partition move and graph-cut based optimization algorithms	15
2.1. Move space	15
2.2. Large and binary partition moves (LBPM)	16
2.3. Exact optimization move	16
2.4. Large and multilabel partition moves (LMPM)	16
3. LMPM optimization	16
3.1. LMPM algorithm	16
3.2. Graph construction	18
3.3. Discussions	20
4. Experimental results	21
4.1. Image restoration	22
4.2. InSAR multichannel phase unwrapping	24
5. Conclusion	30
References	30

1. Introduction

Many image processing and low level computer vision problems can be formulated as the minimization of energy (an objective function). General-purpose optimization techniques, such as simulated

annealing [15], perform optimization with an exponential time and might be slow in practice. Therefore, fast and accurate energy minimization algorithms are needed. A family of fast approaches has been proposed based on the *graph-cut* technique. The latter consists in building a graph where the s,t-minimum-cut yields an optimal labeling of the considered energy.

In this paper we shall consider energies derived from a Markov random field (MRF) with pairwise interactions. It is assumed that images are defined on a lattice denoted by \mathcal{P} . The value of the image x at each pixel p is referred to as x_p and takes a value in a finite set of labels \mathcal{L} . The lattice is endowed with a neighborhood system and pairwise interactions are considered. Two sites p and q that are in interaction

[☆] This paper has been recommended for acceptance by Nikos Paragios.

* Corresponding author. Tel.: +33 6 19338102.

E-mail addresses: aymen.shabou@telecom-paristech.fr (A. Shabou), darbon@cmla.ens-cachan.fr (J. Darbon), florence.tupin@telecom-paristech.fr (F. Tupin).¹ Part of this work has been done while J. Darbon was with the UCLA Mathematics Department, and was supported by the Office of Naval Research through grant N000140710810.

with each other are denoted by (p, q) . The set of all considered pairwise interactions is referred to as \mathcal{N} . A first order Markovian energy is therefore defined as the following:

$$E(x) = \sum_{p \in \mathcal{P}} E^p(x_p) + \sum_{(p,q) \in \mathcal{N}} E^{p,q}(x_p, x_q).$$

The terms E^p encode the data likelihood, measuring the distance of the reconstruction to the observed data, while the terms $E^{p,q}$ correspond to the prior we have on the interactions. In this paper, only priors that are a function of the difference between labels are considered, i.e., $E^{p,q}(x_p, x_q) = E^{p,q}(x_p - x_q)$, (see [9] for instance for the general case).

Since exact optimization is generally an NP-hard problem for many energy functions encountered in image processing and computer vision, some authors have considered specific classes of energy functions [18,9–11] for which the optimization is tractable. However they require a large amount of memory which makes them difficult to use for practical applications. Another class of approaches relies on approximate energy optimization algorithms [6,20,28,21]. The main challenge consists in proposing fast algorithms with low memory consumption while computing a good optimum (ideally the global one). Indeed, recall that the huge memory requirement is the main issue for exact optimization methods, whereas the possibility of being trapped in a bad local minimum is the main drawback of approximate approaches.

Among the classes of approximate algorithms, the most popular are most probably the α -expansion and $\alpha\beta$ -swap move making algorithms [6]. These latter are based on the concept of *partition move* as originally proposed in [27]. They perform binary large partition moves iteratively, changing the current configuration of the solution until its convergence toward a local optimum of the energy function. These approximate algorithms have been successfully applied for solving several image processing and computer vision tasks [26,4,12]. In many cases, the computed solution is a global optimum or a local optimum very close to a global one. Nevertheless, they are restricted to some classes of energy functions (MRF energies with semi-metric or metric potentials). A more recent approach is proposed in [21] and proposes an alternative and more general view of state-of-the-art techniques like the α -expansion algorithm, and generates solutions with guaranteed optimality properties for a much wider class of problems (even for MRFs with non-metric potentials). We also refer the reader to [23] for other efficient moves to approximately minimize general MRF energies. However, there still remain some problems such as those encountered in radar image processing where the associated energy may present many local minima and requires more accurate minimization algorithms [13].

The contribution of this paper consists in proposing new approximate minimization algorithms that yield a better energy minimum than the state-of-the-art algorithms do. These algorithms are iterative and based on the concept of partition move and especially on a novel move that we call the *multilabel* move. Their main property is the trade-off between the memory need and the quality of the optimum in a versatile way.

We shall note that the recent work [29] proposes a similar concept of multilabel moves for minimizing MRF energies with truncated convex priors. Proposed algorithms use the so-called *range move* based strategies [28], that are particular moves in the multilabel move space. Indeed, a subset of labels related to the subset of states proposed to a given variable has to be a sub-range of labels, i.e., a subset of consecutive labels. Moreover, same size of subsets of labels for all variables has been considered to build the multilabel graph.

In other recent work [7], authors extend the work [28] and propose a new strategy for graph construction, called *graphical restriction*, to cope with the subsets of non sub-range labels that might change from 1 pixel to another. Using a recursive and iterative scheme to select subsets of

labels, a multilabel swap move based algorithm is introduced that is able to reach approximate minima of submodular MRF energies in a shorter amount of time, compared to the range moves.

In our current work, a more generalized multilabel graph construction is proposed that can handle various multilabel move making strategies and perform approximate optimization of submodular multilabel MRF energies with a trade-off between computational requirements of the algorithm and optimum quality. Different multilabel move strategies are also explored in depth to solve more properly some image processing problems.

The remainder of this paper is as follows. In Section 2, the proposed multilabel move is introduced as well as a classification of move making based optimization approaches. In Section 3, the proposed large and multilabel approximate optimization algorithm is described. Technical details about the new multilabel constructed graph are also provided. Experimental results for natural and radar image processing are presented in Section 4. Some conclusions are drawn in Section 5.

2. Partition move and graph-cut based optimization algorithms

A partition move based optimization algorithm is a way to make the estimated solution changing during an optimization scheme of an energy function. A standard algorithm that reflects this concept is the iterated conditional modes (ICM) [3]. The latter performs the optimization by assigning iteratively to every pixel a label in the discrete set \mathcal{L} that locally optimizes the energy function. During the optimization process, the solution changes its configuration iteratively until its convergence to a local optimum. However, the ICM algorithm has proven to be non-effective in the case of non-convex energy functions. New move making optimization algorithms have been proposed for this purpose based on the graph-cut technique [6]. The latter was first applied to minimize pseudo-boolean energy functions in [17] and was initially introduced in computer vision to solve the problem of binary image restoration with ferromagnetic Ising-based models in [16]. Then, it was applied to minimize energy functions defined on multilabel variables [6,18,28,9]. Nevertheless, using the graph-cut based optimization technique is restricted to some classes of energy functions due to the *submodularity* condition needed to construct the graph and computing the s-t minimum cut/maximum flow in a polynomial time.

In the following, we describe in more detail the partition move approach. First, some definitions are given for the rest of the paper. Then, approximate algorithms relying on large and binary partition moves are described. Next, exact optimization algorithms are described concisely. Eventually, a new partition move is introduced and a classification of all possible move making optimization algorithms is proposed.

2.1. Move space

Let us first fix some vocabulary, similarly to the one given in [27]. A *move* is a pair of labeling $(x, x') \in \mathcal{X} \times \mathcal{X}$, where \mathcal{X} is the set of all possible labeling. The set of moves $\mathcal{M} \subset \mathcal{X} \times \mathcal{X}$ is called the move space. If $(x, x') \in \mathcal{M}$, then we will say that x' is a labeling obtained by one move from x .

We now give the definition of local or global optimality with respect to a move as the following:

Definition 2.1. Local/global minimum

A labeling x is a local minimum with respect to a move space \mathcal{M} , if for each move $(x, x') \in \mathcal{M}$, we have $E(x) \leq E(x')$.

Note that in case of $\mathcal{M} = \mathcal{X} \times \mathcal{X}$, a local minimum with respect to \mathcal{M} is also a global minimum. We now describe possible move spaces.

2.2. Large and binary partition moves (LBPM)

Algorithms of this part are based on large and binary partition moves. A move from x to x' is said to be *large* if many pixels change their configurations simultaneously. Such a large move leads to bigger changes affecting the current configuration compared to non-large move based optimization algorithms, such as the ICM algorithm. This makes the algorithm more robust to local minima and faster in convergence. These moves have been proved to be effective through the two popular algorithms: α -expansion and $\alpha\beta$ -swap presented in [6].

The α -expansion move consists in proposing to all sites of \mathcal{P} either to take a new label $\alpha \in \mathcal{L}$, or to keep their current labeling in such way that the energy $E(x)$ decreases, i.e., $E(x') < E(x)$, where x' is obtained from x by an α -expansion move.

The $\alpha\beta$ -swap move consists in proposing to a set of sites of x having labels in the subset $\{\alpha, \beta\}$, called active sites, to exchange their labels in such a way that the energy $E(x)$ decreases.

These two moves are called binary moves since only two states are proposed to each site within an iteration of the optimization algorithm. Algorithms that perform local minimization using these two moves work by starting from an initial labeling and then make a series of changes through optimal binary moves. This process is iterated until no moves can further decrease the energy. They are based on the graph-cut technique and described through specific steps called *cycles* and *iterations* [6]. In a cycle, the minimization algorithm, based on the binary expansion move (respectively binary swap move), performs an iteration for every label $\alpha \in \mathcal{L}$ (respectively every pair of labels $(\alpha, \beta) \in \mathcal{L}^2$), in a certain order (fixed or random). At each iteration, an exact pseudo-boolean optimization is performed thanks to the minimum cut. A cycle is said to be successful if a labeling that gives a decrease of the energy is found for at least one iteration. The algorithm stops after the first unsuccessful cycle, i.e., no further improvement is possible.

In [27], the author proves the convergence of these algorithms to local minima in a finite number of cycles and shows that few number of cycles are needed to converge in practice. However, since the submodularity property of the energy function is required to perform exact pseudo-boolean minimization via graph-cuts with a polynomial time [16,20], only a family of energy functions can be minimized based on the proposed large and binary moves. It has been shown in [6] that the $\alpha\beta$ -swap move works for the *semi-metric* prior function while the α -expansion move works only for the *metric* prior (i.e., semi-metric that satisfies the triangle inequality).

Finding the optimal partition move is a polynomial time consuming task since the size of the constructed graph is linear with respect to the number of sites and polynomial algorithms exist for the maximum flow computation on this graph (in particular, the very efficient maximum flow approach proposed in [5] to the community working on computer vision problems).

2.3. Exact optimization move

Another family of optimization algorithms [18,9,24] has been proposed to perform exact optimization of some energy functions in a pseudo-polynomial time [1]. These algorithms can be seen as partition move algorithms in the space of moves $\mathcal{M} = \mathcal{X} \times \mathcal{X}$, i.e., all sites in \mathcal{P} can take any label in \mathcal{L} .

A first approach has been proposed by Ishikawa in [18]. The main idea is to construct a graph in such way that any cut on this graph corresponds to a configuration of x . Thus an s-t minimum cut corresponds to an optimal configuration, i.e., a configuration that has minimal energy. This graph is very large since for every site in \mathcal{P} , $|\mathcal{L}|$ nodes related to all labels and edges connecting them to those of the neighboring sites in \mathcal{P} are necessary. Edges have specific weights making correspondence between a cut on the graph and a configuration of x . It is

shown in [18] that the convexity of the prior is required for the applicability of the approach.

The second approach was proposed in [11,9]. This approach relies on mapping the problem of finding an optimal solution of the Markovian energy, where variables take values in the discrete set \mathcal{L} , into a combinatorial one which involves only binary variables. This mapping is done by considering the level-sets of the labels instead of the label values themselves. Using the exact binary optimization, as stated before, a global optimum of the original energy is reconstructed by preserving the monotone property in energies defined on the level-sets of x . The size of the graph constructed to perform this optimization approach is the same as for the graph proposed by Ishikawa, since for every level in \mathcal{L} , nodes related to all sites in \mathcal{P} are created. However, edges are different from those of Ishikawa's graph, since all the nodes in each level are now connected to the source and to the sink (to perform binary optimizations) and additional edges are used to preserve the monotone property. We note that computing minimum s-t cut in this graph might be faster than computing it in the Ishikawa's graph [11].

To extend the exact MRF energy minimization task to higher order cliques, we refer to the work [24] where authors proceed by transforming special classes of multi-label higher order functions to submodular second order boolean functions, which can be minimized exactly using the min-cut/max-flow technique.

2.4. Large and multilabel partition moves (LMPM)

We now introduce a new move called *multilabel* move. Since the binary move exploits a limited number of two labels among those in \mathcal{L} , proposing more labels to a pixel in a single move may yield to a better energy minimization result. The idea we explore next is the use of large subsets of labels in \mathcal{L} to perform multilabel moves. Particular cases of this new move give the previously described algorithms:

- if the label set is a binary set, it corresponds to a binary large move, as described in Section 2.2,
- if the label set is the set of all labels, it corresponds to a global optimization move, as described in Section 2.3.

A possible classification of the move making optimization algorithms involving the class of multilabel moves is proposed in Fig. 1. This classification is relative to the breadth of the move and the size of the used label set. In this paper, we study the interest of the intermediate size of the label sets. First, we describe the graph construction. Then we analyze some experiments.

3. LMPM optimization

We first describe the algorithm to perform an LMPM move before describing the associated graph construction. Then we discuss the properties of this approach.

3.1. LMPM algorithm

The LMPM algorithm, depicted in Fig. 2, is a general form of a graph-cut and partition move based optimization algorithm. At each iteration, an optimal move is accepted if it strictly decreases the energy function value. In the case of a multilabel move, the set of labels is denoted by $\mathcal{L}_m \subset \mathcal{L}$. A cycle must visit all the labels in \mathcal{L} and finished after a prescribed number of iterations. More precisely, let us denote by $\mathcal{L}_m^{(i)}$ the label set with size m chosen at the i th iteration. We have $\bigcup_i \mathcal{L}_m^{(i)} = \mathcal{L}$. A cycle is said to be successful, if the energy decreases exactly for at least one iteration. Otherwise, the algorithm has already converged to a local minimum and no further improvements are possible. We also note that for each move (expansion, swap, etc.), the

(Fig. 2). However, the results can be entirely different due to the specific choice of label sets and changing configurations related to each move.

In this work, three issues need to be addressed to perform an LMPM based optimization:

1. Given the label set \mathcal{L}_m , how is the graph built to get the optimal partition move \hat{x} from x in a polynomial (or pseudo-polynomial) time using the minimum-cut/maximum flow technique?
2. Which class of energy functions can be optimized based on this approach?
3. How larger label sets (i.e., applying multilabel moves) in the minimization scheme could improve the minimum quality?

Regarding the problem of label set choice, we discuss this issue from an experimental point of view.

3.2. Graph construction

In this section, the construction of the graph used to compute the optimal move in one iteration of the LMPM algorithm is presented. The constructed graph is a layered graph similarly to the one proposed by Ishikawa [18]. However the capacities on the edges need to be redefined to take into account different subsets of labels for each pixel, and the current labeling at each move. Thus, the proposed construction is a generalization of the layered Ishikawa graph to any multilabel graph for the move making algorithms.

Let us note by $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ a directed graph with non-negative capacity edges, where m is the size of the label set \mathcal{L}_m , \mathcal{V}_m is the set of vertices and \mathcal{E}_m is the set of edges. For the maximum-flow computation, two special vertices, the source s and the sink t are added to the set of vertices.

As stated before, in one iteration i , a label set $\mathcal{L}_m^{(i)}$ is generated. In the case of β_m -swap, this set is identical for all pixels. However, for the α_m -expansion move and the δ_m -jump, the set takes into account the current labels assigned to the pixels in each iteration. Thus, a label set for each pixel p has to be defined, at the iteration i , denoted by $\mathcal{L}_m^{(i)}(p)$. This label set represents the current label of the pixel (label

generated through the iteration $i - 1$) and the new proposed labels $\mathcal{L}_m^{(i)}$, i.e., $\mathcal{L}_m(p) = \mathcal{L}_m^{(i)} \cup \{x_p\}$. This label set is denoted by $\mathcal{L}_m(p)$. For the rest of this paper we consider that a linear order is chosen on this set (ascending for example). Finally, let us denote by k_p the size of $\mathcal{L}_m(p) = \{l_j^p ; j = 1..k_p\}$.

The main idea of the method is to construct a graph such that a cut in the graph corresponds to a configuration of the MRF given the label sets $\{\mathcal{L}_m(p) ; p \in \mathcal{P}\}$. The total cost of the cut is equal to the total energy of the configuration up to a constant. Thus, the minimum cut of the graph yields the MRF configuration with minimal energy. If this energy value is less than the current one, we obtain an optimal move from the current configuration.

We now give details of the graph construction. Let us first define the set of vertices \mathcal{V}_m by

$$\mathcal{V}_m = \{\mathcal{V}_m(p) | p \in \mathcal{P}\}, \quad (1)$$

with,

$$\mathcal{V}_m(p) = \{v_{p,i} | i = 1..k_p\}. \quad (2)$$

For each pixel p , we create k_p nodes that correspond to the label set $\mathcal{L}_m(p)$.

Let us now define the set of edges \mathcal{E}_m by

$$\mathcal{E}_m = \mathcal{E}_m^D \cup \mathcal{E}_m^C \cup \mathcal{E}_m^P, \quad (3)$$

where \mathcal{E}_m^D is a subset of so called data edges, \mathcal{E}_m^C is a subset of constraint edges and \mathcal{E}_m^P is a subset of penalty edges.

Data edges \mathcal{E}_m^D implement the likelihood energy terms E^P . They are defined, as in [18], as the following

$$\mathcal{E}_m^D = \bigcup_{p \in \mathcal{P}} \mathcal{E}_m^D(p), \quad (4)$$

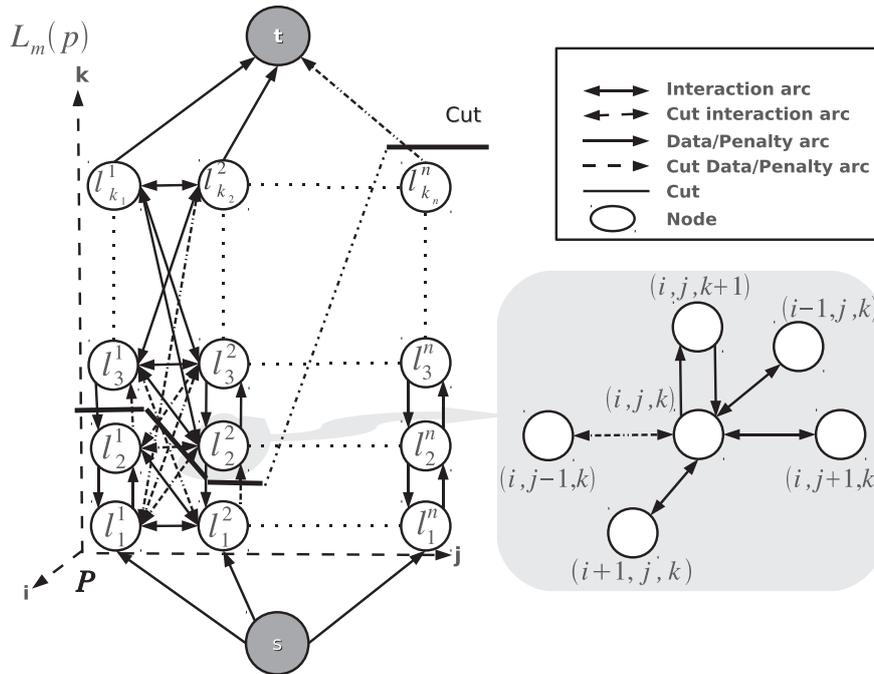


Fig. 3. Graph construction for an optimal multilabel expansion move. On the left, a part of the graph \mathcal{G}_m defined on 3 pixels. A cut is depicted in bold and arcs in the cut are dotted whereas continuous ones are not. A part of the graph is highlighted on the right. The three arc families are distinguished: data edges (vertical edges oriented to the top), constraint edges (vertical edges oriented to the bottom) and penalty edges connecting all label nodes of two 4-connectivity neighboring pixels.

with,

$$\mathcal{E}_m^D(p) = \left\{ (s, v_{p,1}) \right\} \cup \left\{ (v_{p,i}, v_{p,i+1}) \mid i = 1..k_p - 1 \right\} \cup \left\{ (v_{p,k_p}, t) \right\}, \quad (5)$$

and the capacities of these edges are set to

$$\begin{cases} c(s, v_{p,1}) = +\infty, \\ c(v_{p,i}, v_{p,i+1}) = E^P(x_p = l_i^p); i = 1..k_p - 1, \\ c(v_{p,k_p}, t) = E^P(x_p = l_{k_p}^p). \end{cases} \quad (6)$$

To obtain the correspondence between a cut in the graph \mathcal{G}_m and the data terms of the energy function, constraint arcs are added to guarantee that only one data arc per pixel is in the cut. These arcs are defined following [18] by

$$\mathcal{E}_m^C = \bigcup_{p \in \mathcal{P}} \mathcal{E}_m^C(p), \quad (7)$$

with,

$$\mathcal{E}_m^C(p) = \left\{ (v_{p,i+1}, v_{p,i}) \mid i = 1..k_p - 1 \right\}, \quad (8)$$

and the capacity of each constraint arc is set to infinity, i.e. $c(v_{p,i+1}, v_{p,i}) = +\infty$; $\forall i = 1..k_p$.

For the pairwise terms, the penalty edges are defined by

$$\mathcal{E}_m^P = \left\{ (v_{p,i}, v_{q,j}) \mid p, q \in \mathcal{P}; p \sim q; i = 1..k_p; j = 1..k_q \right\}. \quad (9)$$

We shall see in the following that the cut for assigning the label l_i^p for a pixel p and the label l_j^q for the pixel q involves a limited number of arcs.

More precisely, there are firstly the two data arcs $(v_{p,i}, v_{p,i+1})$ and $(v_{q,j}, v_{q,j+1})$ that correspond to the two data energy terms related to the pixels p and q . Then, various other edges, from \mathcal{E}_m^P , are in the cut: those which are going from p to q , i.e., $(v_{p,a}, v_{q,b})$ where $a \leq i$ and $b > j$, and those which are going from q to p , i.e., $(v_{q,b}, v_{p,a})$ where $a > i$ and $b \leq j$. Thus, the total cost of the cut penalty edges for the two neighboring pixels is given by

$$f(l_i^p, l_j^q) = \sum_{a=1}^i \sum_{b=j+1}^{k_q} c(v_{p,a}, v_{q,b}) + \sum_{a=i+1}^{k_p} \sum_{b=1}^j c(v_{q,b}, v_{p,a}). \quad (10)$$

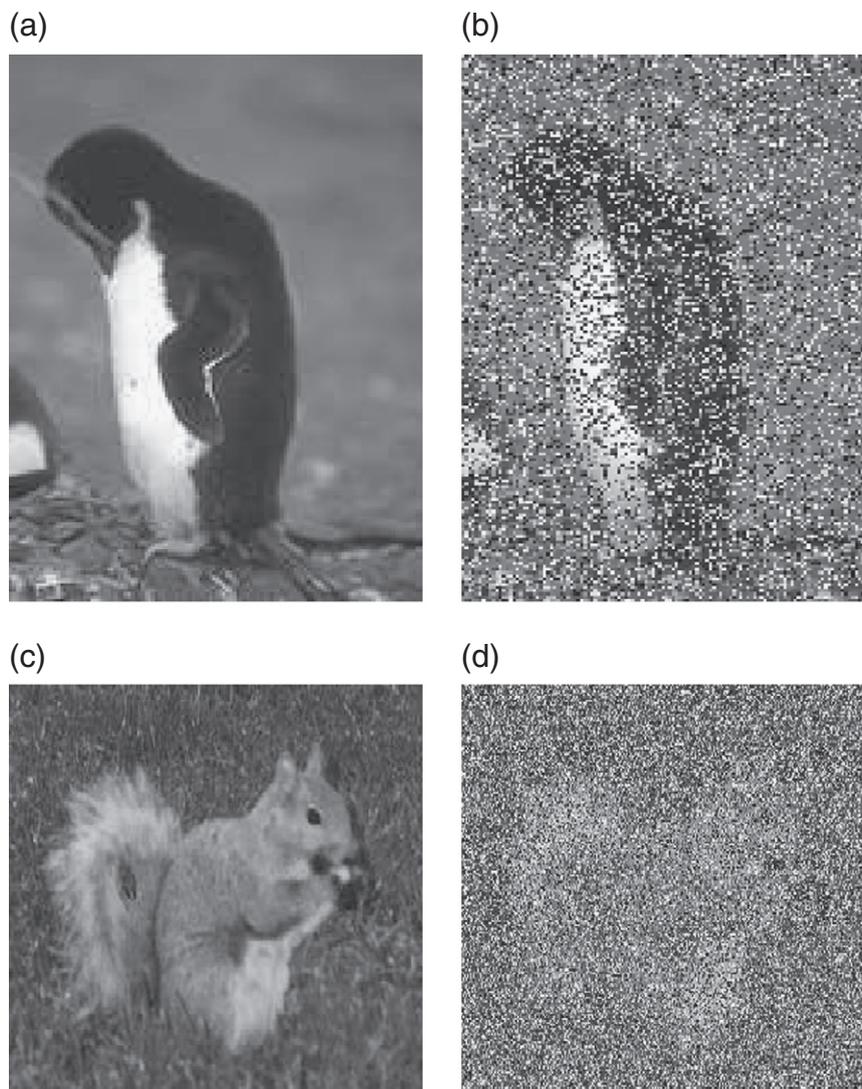


Fig. 4. The original images and the corresponding noisy ones: (a) the true image of Penguin, and (b) the corresponding noisy image with 50% of pixels corrupted by the impulsive noise ($P=0.5$). (c) The true image of Squirrel, (d) the corresponding noisy image with 70% of pixels corrupted by the impulsive noise ($P=0.7$).

To get the correspondences between a cut in the multilabel graph and the total energy cost, penalty edges are redefined as the following

$$c(v_{p,a}, v_{q,b}) = \frac{1}{2} (g(l_a^p - l_{b-1}^q) + g(l_{a-1}^p - l_b^q) - g(l_{a-1}^p - l_{b-1}^q) - g(l_a^p - l_b^q)), \quad (11)$$

with $g = E^{p,q}$ and $a > 1, b > 1$. Auxiliary edges are required to obtain the equivalence between the value of the cut and the convex function g up to a constant. These edges have capacities denoted by $D^{p,q}(j)$ for the node $v_{p,j}$ and $D^{p,q}(i)$ for the node $v_{p,i}$. The total cost of the cut penalty edges can be expressed as follows:

$$f(l_i, l_j) = g(l_i^p - l_j^q) - D^{p,q}(j) - D^{p,q}(i) + Cst, \quad (12)$$

with,

$$\begin{cases} D^{p,q}(i) &= \frac{1}{2} (g(l_i^p - l_1^q) + g(l_i^p - l_{k_q}^q)), \\ D^{p,q}(j) &= \frac{1}{2} (g(l_j^q - l_1^p) + g(l_j^q - l_{k_p}^p)), \\ Cst &= \frac{1}{2} (g(l_1^p - l_{k_q}^q) + g(l_1^q - l_{k_p}^p)). \end{cases} \quad (13)$$

In our construction, the auxiliary edges are merged with data edges since their capacities depend only on one variable.

It has been proven in [18,9] that g needs to be a convex function to have a non-negative capacity $c(v_{p,a}, v_{q,b})$. The same demonstration holds in our case.

Fig. 3 shows an example to summarize our graph construction.

Given the graph G_m , a configuration x of all pixels $p \in \mathcal{P}$ and label sets $\{\mathcal{L}_m(p); p \in \mathcal{P}\}$, there is a one to one correspondence between cuts of finite capacities on this graph and a labeling that lives in \mathcal{L}_m . Besides, the cost of one cut is equal to the energy of the resulting configuration up to a constant. Moreover, the convexity of the prior energy function is a necessary and sufficient condition for the non-negativeness of all edge capacities. Therefore, a minimum cut on the graph is computed in pseudo-polynomial time.

3.3. Discussions

In this section we discuss the main properties and contributions of this algorithm in terms of optimum quality.

The first trivial property is the following: if the same multilabel move (an expansion move for instance) is done with different label sets \mathcal{L}_m and $\mathcal{L}_{m'}$ such that $\mathcal{L}_m \subset \mathcal{L}_{m'}$, then the energy of the resulting labeling is minimum for the move with the largest label set, since the set of optima reached by the move with less labels is included in the set of the other move optima. This property highlights the importance of a multilabel move with a larger label set when one step of the minimization algorithm is performed.

Thus, if the above inclusions hold at every step of the minimization process, then a multilabel large move guarantees a better energy minimum when increasing the size m of subsets \mathcal{L}_m . Nevertheless there are no general theoretical results on the optimum quality of the LMPM. It's assumed that in the worst case, for each expansion, swap and jump moves, the optimum given by an LMPM is as good as the one given by an LBPM, since the latter is a special case when $m = 2$. This assumption will also be verified in the [Experimental results](#) section.

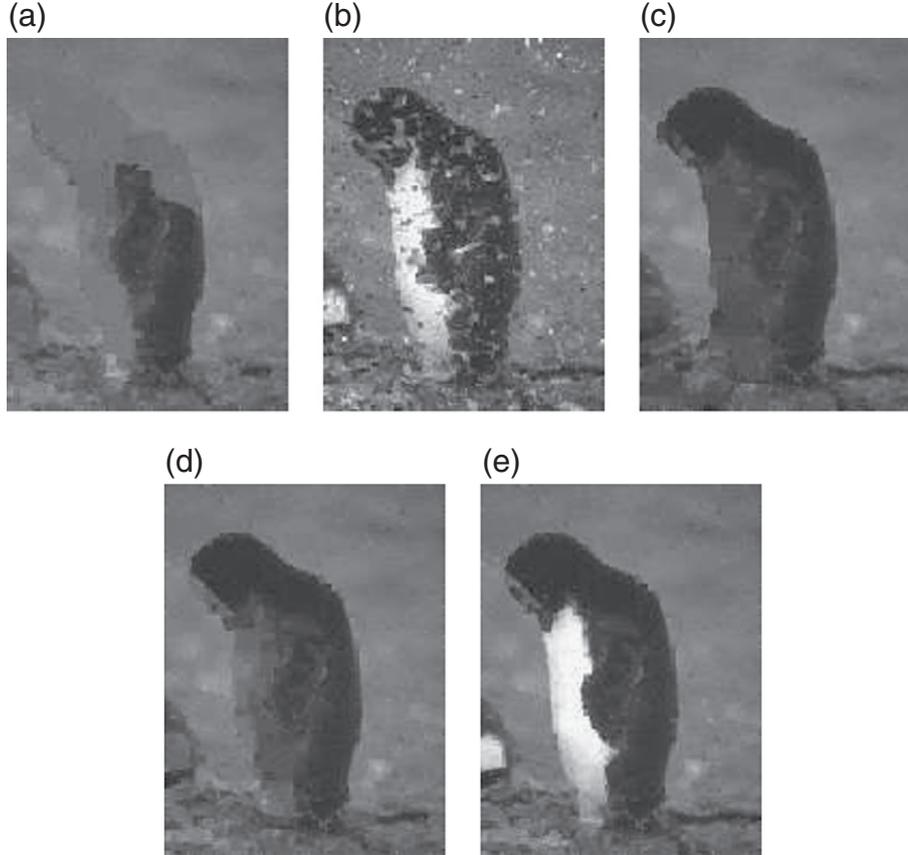


Fig. 5. Restoration of the noisy image Fig. 4(b) using several state-of-the-art optimization algorithms: (a) α -expansion with random labels' scanning [6], (b) $\alpha\beta$ -swap [6], (c) FastPD [22], (d) multilabel expansion move of [29] with sub-ranges of labels of size 32, and (e) exact optimizer of Ishikawa [18].

Table 1

Computational results of the experiment in Fig. 5: time is expressed with (min:s) and graph size with the number of allocated nodes.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E} (\times 10^5)$
5(a)	5121	1.0738	2:01	0.21
5(b)	32,896	1.1942	0:11	0.21
5(c)	18	1.0356	1:17	0.21
5(d)	40	1.0220	5:11	6.72
5(e)	1	0.9954	4:02	55.91

4. Experimental results

In this section, different experiments are presented to study the optimum quality of LMPM algorithms. First, the restoration of images corrupted by a strong uniform noise is presented. Even if this rate of noise seems to be non-realistic for real image processing problems, it provides a good way to compare the performances of approximate optimization algorithms. Nevertheless, usually in the remote sensing SAR image processing tasks, a similar amount of noise characterizes the real data. Therefore, in the second part of the experiments, we perform the MRF optimization in the case of more complex energy functions related to a SAR problem. That is the phase unwrapping (PU) in the interferometric synthetic aperture radar (InSAR) imaging, where the goal is to unwrap and at the same time denoise the interferometric phase using the multichannel data. Such a reconstruction is of great interest in the field of Geoscience and remote sensing since it generates digital elevation models (DEM) of the Earth's surface [2].

For all data, first we conduct comparisons between results obtained with the proposed LMPM based optimization algorithms and some state-of-the-art approaches. We will show that LMPM algorithms perform well on difficult problems, proposing a trade-off between algorithm complexity and optimum quality. They outperform the LBPM based algorithms (such as the α -expansion, $\alpha\beta$ -swap [6], and δ -jump [4,12]), the fast primal-dual based algorithm (FastPD) [21,22], and the multilabel moves proposed in [28,29]. For the latter, we show only the optimal solution obtained with a parameter m (size of subsets of labels) that gives a good result with our LMPM optimizer (trade-off between memory and optimum quality). Exact solutions given by the algorithms of [18,9] are also provided to note how close are the optima reached by the LMPM algorithms to the global one, while using a much less memory.

Second, the properties discussed in Section 3.3 will be verified empirically on the different data. Thus, the different optima reached with several sizes of subsets of labels are depicted, and numerical results (computational complexities, energy minimum) are provided to evaluate the algorithm's performances.

We note that for the experiments on image restoration from the impulsive noise, only the expansion and swap based LMPM algorithms are tested, since they are more appropriate for this class of labeling problem, where we usually quantize the label set in a discrete one \mathcal{L} . Whereas for the InSAR phase unwrapping problem, a preliminary quantization of the continuous label set could introduce a loss of physical information, that is needed to well reconstruct the DEM. Thereby, the jump based LMPM algorithm will be used, allowing a better labeling on this type of data.

The computational results we present have been obtained on a machine with a 2 Ghz dual core CPU and 2 Go of RAM. The min-cut/

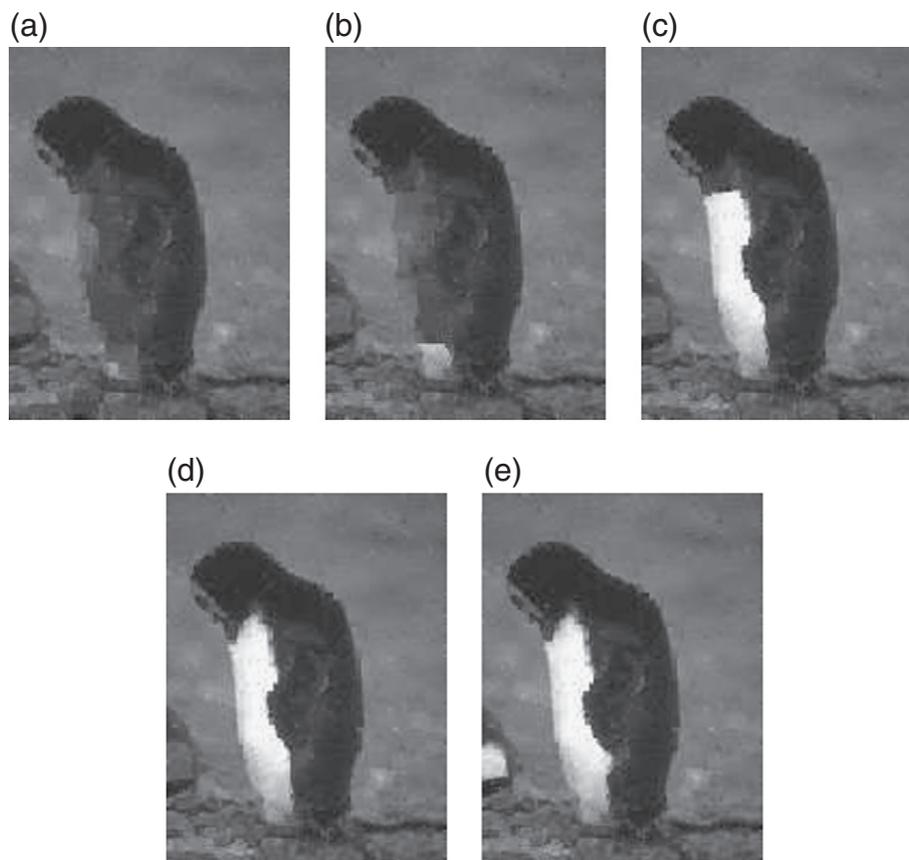


Fig. 6. Restoration of the noisy image Fig. 4(b) using the proposed multilabel expansion move with subsets of random labels for different values of $m \in \{8, 16, 32, 64, 128\}$. (a) α_8 -expansion, (b) α_{16} -expansion, (c) α_{32} -expansion, (d) α_{64} -expansion, and (e) α_{128} -expansion.

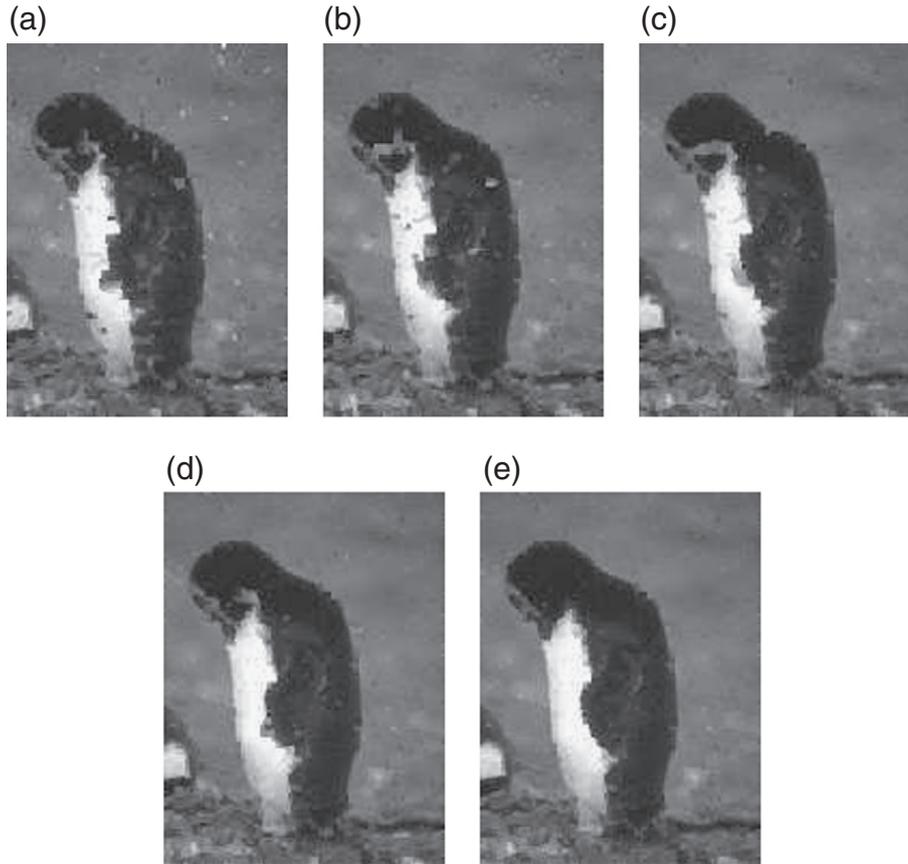


Fig. 7. Restoration of the noisy image Fig. 4(b) using the proposed multilabel swap move with subsets of random labels for different values of $m \in \{8, 16, 32, 64, 128\}$. (a) β_8 -swap, (b) β_{16} -swap, (c) β_{32} -swap, (d) β_{64} -swap, and (e) β_{128} -swap.

max-flow algorithm we use on our graphs is the one proposed in [5], with the C++ implementation provided by the authors.²

4.1. Image restoration

Images used in this experiment are corrupted by impulsive noise with parameter P . A pixel keeps its original value with a probability $(1 - P)$ or it takes a new one uniformly in \mathcal{L} . This kind of noise is known to be more destructive than other noises like the additive one for instance. In fact, if a pixel is corrupted by this noise, the information that it carries is totally lost.

The associated data likelihood energy function for any site p is given by

$$E^p(x_p) = \begin{cases} -\log\left((1-P) + \frac{P}{L}\right) & \text{if } y_p = x_p, \\ -\log\left(\frac{P}{L}\right) & \text{else,} \end{cases} \quad (14)$$

where y denotes the observed noisy image, and L is the cardinal of the set \mathcal{L} . For this experiment, we consider $\mathcal{L} = \{0, 1, \dots, 254, 255\}$.

Fig. 4(b) and (d) illustrates noisy images with different rates of impulsive noise $P=0.5$ and $P=0.7$ respectively. Their corresponding original images are depicted in Fig. 4(a) and (c) respectively.

We consider a discrete total variation (DTV) prior energy for the MRF with the grid of 4-nearest neighbors as the neighborhood system [11]. Thus, the prior energy is given by

$$E^{p,q}(x_p, x_q) = \beta \sum_{(p,q) \in \mathcal{N}} |x_p - x_q|, \quad (15)$$

where β corresponds to a regularization coefficient. This parameter is adjusted in the experiments to obtain the best visual result.

Given this energy model, we can perform restoration experiments using both the α_m -expansion and the β_m -swap move making algorithms.

It is important to note that for the experiments random permutations on the label set \mathcal{L} are applied. Indeed, the results obtained by the standard α -expansion or $\alpha\beta$ -swap for example are much worse if the new labels are proposed following the natural linear order. More precisely, the sets \mathcal{L}_m are generated as follows: we consider any random permutation of the set \mathcal{L} . First, we select the first m elements and then re-order them. This defines \mathcal{L}_m that is used to perform the optimal move. Then, the next m elements of the permuted set \mathcal{L} are selected and the process remains the same. Once all the labels of

Table 2
Computational results of the experiment in Fig. 6.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E}_m (\times 10^5)$
6(a)	448	1.0224	4:59	1.75
6(b)	112	1.0174	3:34	3.51
6(c)	40	0.9984	4:11	6.98
6(d)	20	0.9969	6:42	14
6(e)	6	0.9959	7:09	28

² <http://www.cs.cornell.edu/People/vnk/software.html>.

Table 3
Computational results of the experiment in Fig. 7.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E}_m (\times 10^5)$
7(a)	658	1.0465	0:13	1.75
7(b)	328	1.0295	0:20	3.51
7(c)	164	1.0097	0:55	6.98
7(d)	82	1.0018	3:27	14
7(e)	41	0.9972	6:03	28

\mathcal{L} have been visited, a new permutation on the label set \mathcal{L} is performed and the process starts again until the convergence of the algorithm.

We have also experimentally verified that applying a random permutation on labels gives better results in terms of optimum quality than not performing any permutation. Regarding the initialization, we have verified empirically that for the expansion moves it is better to start from an initial uniform configuration; whereas for the swap moves, starting from the noisy observed image leads to a faster convergence toward a better local minimum of the energy. The latter could be explained by the restriction of the swap move to only active sites. Thus starting from the noisy image would induce a deep decrease in the energy, since many pixels would be involved in the moves at the first steps of the minimization.

In Fig. 5, the restoration results obtained with some state-of-the-art MRF optimization based algorithms (α -expansion Fig. 5(a), $\alpha\beta$ -swap Fig. 5(b), FastPD Fig. 5(c), multilabel expansion move of [28] with sub-ranges of labels of size 32 Fig. 5(d) and the exact optimum of Ishikawa's algorithm Fig. 5(e)) are depicted with the corresponding computational numerical results highlighted in Table 1.

The approximate optimal solutions, compared to the exact solution given by the Ishikawa algorithm (Fig. 5(e)), show poor results, i.e., bad local optima of the considered non-convex energy are reached. Despite their low computational time, these algorithms could not be considered to solve this labeling problem since the obtained solutions are still noisy and far from the optimal one.

To avoid being trapped in poor local optima, while using low memory for graph construction, the use of the approximate LMPM

Table 4
Computational results of the experiment in Fig. 8: time is expressed with (min:s) and graph size with the number of allocated nodes.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E} (\times 10^5)$
8(a)	5120	4.3408	7:35	6.02
8(b)	32,896	4.5395	0:34	6.02
8(c)	35	4.4007	10:51	6.02
8(d)	40	4.2084	13:18	96.31
8(e)	1	4.1576	15:36	773.47

optimization algorithms would be necessary. In Fig. 6 results obtained with the proposed LMPM α_m -expansion algorithm, starting from a uniform null image, with several values of the parameter m , are presented. While, in Fig. 7 results obtained with the proposed LMPM β_m -swap algorithm, starting from the noisy image, with several values of the parameter m , are shown.

We see that we obtain better results than the previous algorithms as m increases. Furthermore, the use of the subsets of random labels requiring our graph construction improves the optimum quality reached by the state-of-the-art multilabel move making algorithms if we compare for instance our result obtained at $m = 32$ (Fig. 6(c)) to the one given by the multilabel expansion move in Fig. 5(d). Table 2 (resp. Table 3) shows the number of steps, the value of the energy minimum, the time and the number of graph nodes needed to compute a minimizer for several m with a multilabel expansion move (resp. swap move).

It is important to note that, although the LMPM algorithm with 32 labels (for each step) converges toward a local optimum of the energy, the obtained solutions in Figs. 6(c) and 7(c) are very close to the global one in Fig. 5(e) in terms of the reached energy's minimum and reconstruction qualities. Besides the amount of memory required to compute this result is much less than for the global solution. Indeed the global minimal energy is $E_{\text{global}} = 0.9954e + 06$ while the approximate solution obtained using the α_{32} -expansion algorithm yields an energy of $E_{\text{approx}} = 0.9984e + 06$. This corresponds to a relative error $\frac{E_{\text{approx}} - E_{\text{global}}}{E_{\text{global}}} = 0.003$. Furthermore, the algorithm for the exact minimization ($m = 256$) requires a huge amount of memory, while

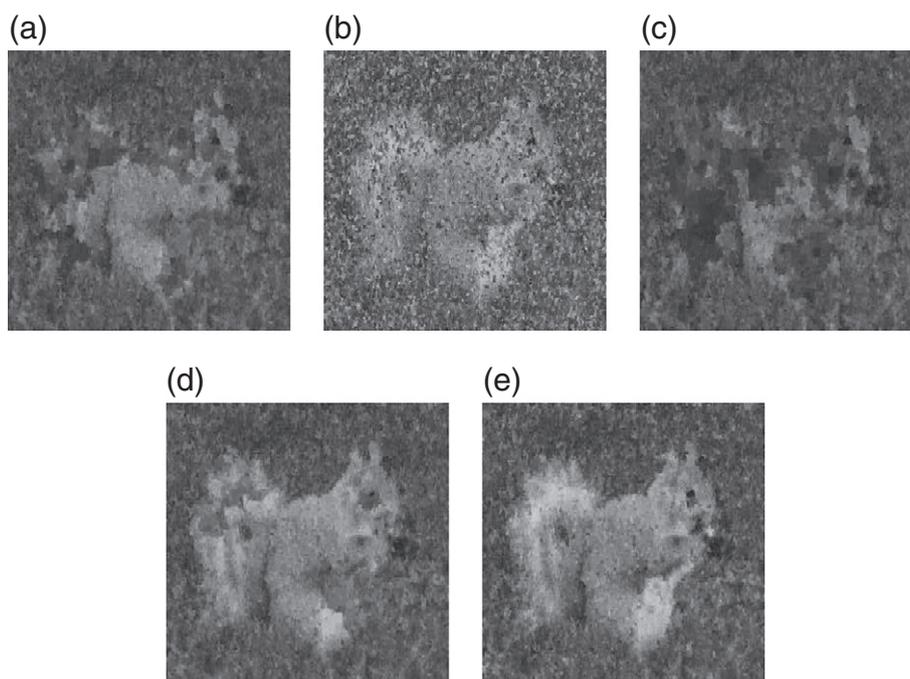


Fig. 8. Restoration of the noisy image Fig. 4(d) using several state-of-the-art optimization algorithms: (a) α -expansion with random labels' scanning, (b) $\alpha\beta$ -swap [6], (c) FastPD [22], (d) multilabel expansion move of [29] with sub-ranges of labels of size 32, and (e) exact optimizer of Ishikawa [18].

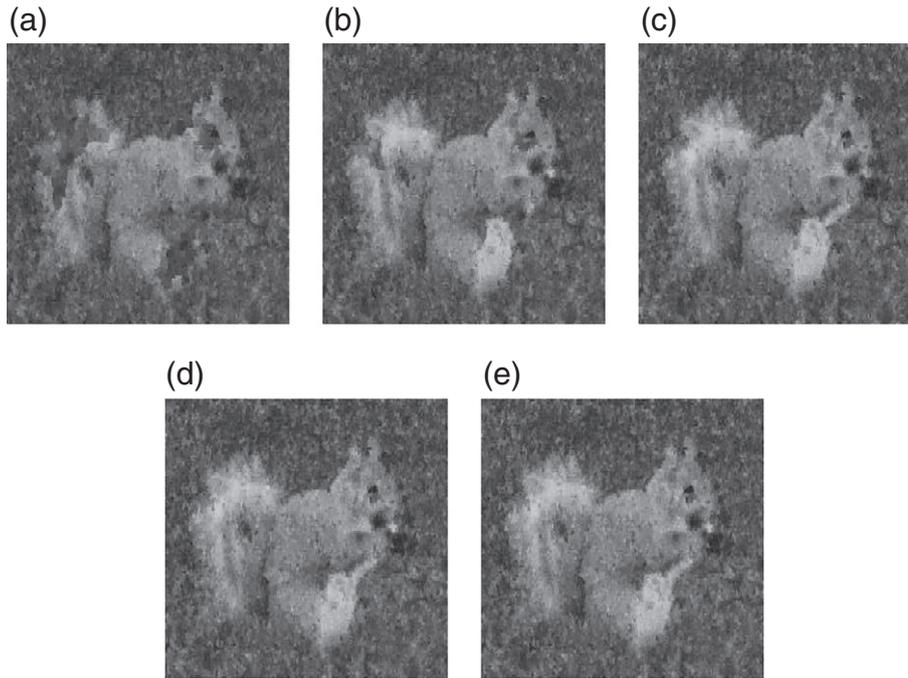


Fig. 9. Restoration of the noisy image Fig. 4(d) using the proposed multilabel expansion move with subsets of random labels for different values of $m \in \{8, 16, 32, 64, 128\}$. (a) α_8 -expansion, (b) α_{16} -expansion, (c) α_{32} -expansion, (d) α_{64} -expansion, and (e) α_{128} -expansion.

the LMPM algorithm with 32 labels requires much less memory (about 1/8th).

The same experiments are performed to restore the noisy image in Fig. 4(d), showing the robustness of the proposed approach even in the case of textured images, where the denoising from the high rate impulsive noise would be a difficult task.

Results of the state-of-the-art optimization algorithms are depicted in Fig. 8 with their corresponding numerical results in Table 4, and those obtained with the proposed α_m -expansion (resp. β_m -swap) with different values of the parameter m are highlighted in Fig. 9 (resp. 10) and Table 6 (resp. 6).

Fig. 11 shows similar quantities to those depicted in Fig. 5 when performing the denoising experiment with several random permutations on labels. More precisely, we have performed 10 random permutations. Averages of the obtained values are depicted in the plots.

We note that the time needed to compute an optimal move increases with larger label sets, since the size of the graph increases with the number of labels. It is important to point out that improving time computation of a partition move based optimization algorithm is not the purpose of this work and the experiments are chosen to prove the ability of the proposed algorithms to reach good energy minima while a low memory is used (Table 5). However, in practice, a few number m of labels is needed by an LMPM move to converge toward a good local optimum of the energy, leading to a low computational time, the same for the exact optimization algorithm. Thereby, this computational time of the proposed LMPM algorithms does not penalize their use for solving image processing problems. Indeed, let

us see the last experiment performed on the noisy squirrel image with the α_m -expansion move, for instance. Optimizing exactly with 256 labels takes a time similar as optimizing with LMPM with $m = 32$. The latter yields an image which is very close to the global minimum with respect to the energy, while requiring much less memory (about 1/8th) than the exact optimizer. The same conclusion holds for the β_m -swap move.

4.2. InSAR multichannel phase unwrapping

In this section, the problem of unwrapping the interferometric SAR phase while denoising it at the same time is considered.

InSAR systems allow the generation of a digital elevation model (DEM) of the Earth's surface. The approach relies on considering the known relation between the interferometric SAR phase and the height of the ground [2]. The main problem is the phase unwrapping operation since the measured interferometric phase is only known in the principal interval $[-\pi, \pi[$ (wrapped phase). In order to restore the relation between interferometric phase and ground height (which is necessary to generate the DEM) it is necessary to unwrap the phase (i.e., to know the phase in its absolute value). The unwrapping operation is not an easy task. In particular, if the so called *Itoh* condition [19] (the absolute value of the phase difference between the neighboring pixels is less than π) is not satisfied, the phase unwrapping operation becomes an ill-posed problem. The *Itoh* condition is easily violated in real InSAR data due to both the presence of low coherence areas and the presence of high discontinuities.

Table 5
Computational results of the experiment in Fig. 9.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E}_m (\times 10^5)$
9(a)	640	4.2518	12:22	24.08
9(b)	128	4.1786	14:05	48.16
9(c)	40	4.1623	14:37	96.31
9(d)	24	4.1600	28:04	192.61
9(e)	14	4.1597	53:19	385.23

Table 6
Computational results of the experiment in Fig. 10.

Figure	Steps to converge	$E_{\min} (\times 10^6)$	Time (min:s)	$ \mathcal{E}_m (\times 10^05)$
10(a)	656	4.3488	0:42	1.75
10(b)	328	4.3172	0:59	3.51
10(c)	164	4.2690	2:16	6.98
10(d)	82	4.2247	8:30	14
10(e)	87	4.1776	15:53	28

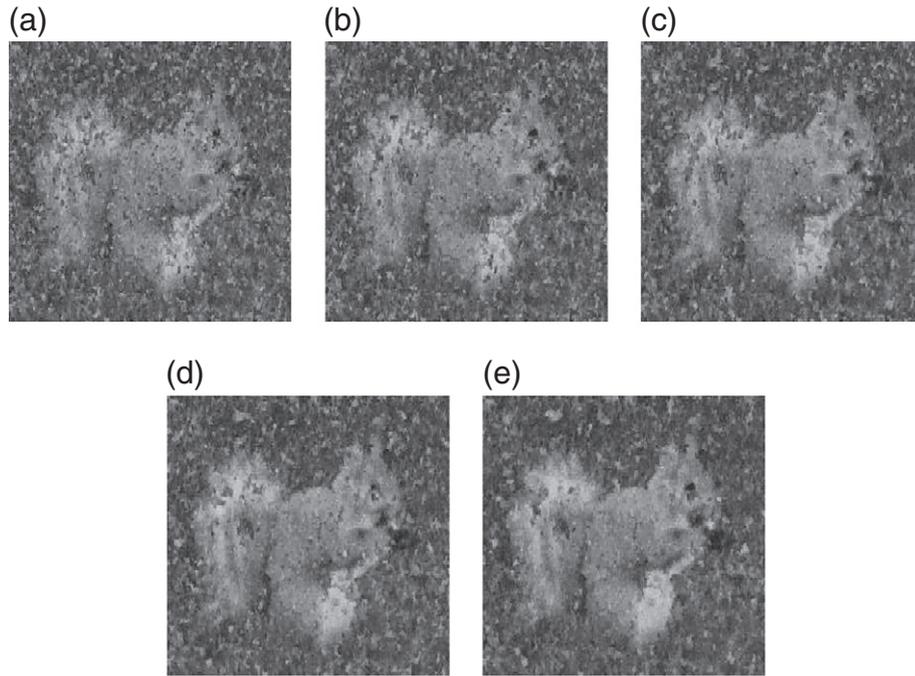


Fig. 10. Restoration of the noisy image Fig. 4(d) using the proposed multilabel swap move with subsets of random labels for different values of $m \in \{8, 16, 32, 64, 128\}$. (a) β_8 -swap, (b) β_{16} -swap, (c) β_{32} -swap, (d) β_{64} -swap, and (e) β_{128} -swap.

To solve the phase unwrapping problem in the non-*Itoh* condition, an efficient and robust method is the multichannel phase unwrapping (MCPU) [14]. Multichannel InSAR techniques exploit the availability of different and independent interferograms referred to the same

scene and obtained using different channels (baselines or frequencies). In order to combine these different available channels, a statistical approach is used. In particular, to recover the unwrapped phase and then generate the DEM, the solution is formulated as a maximum

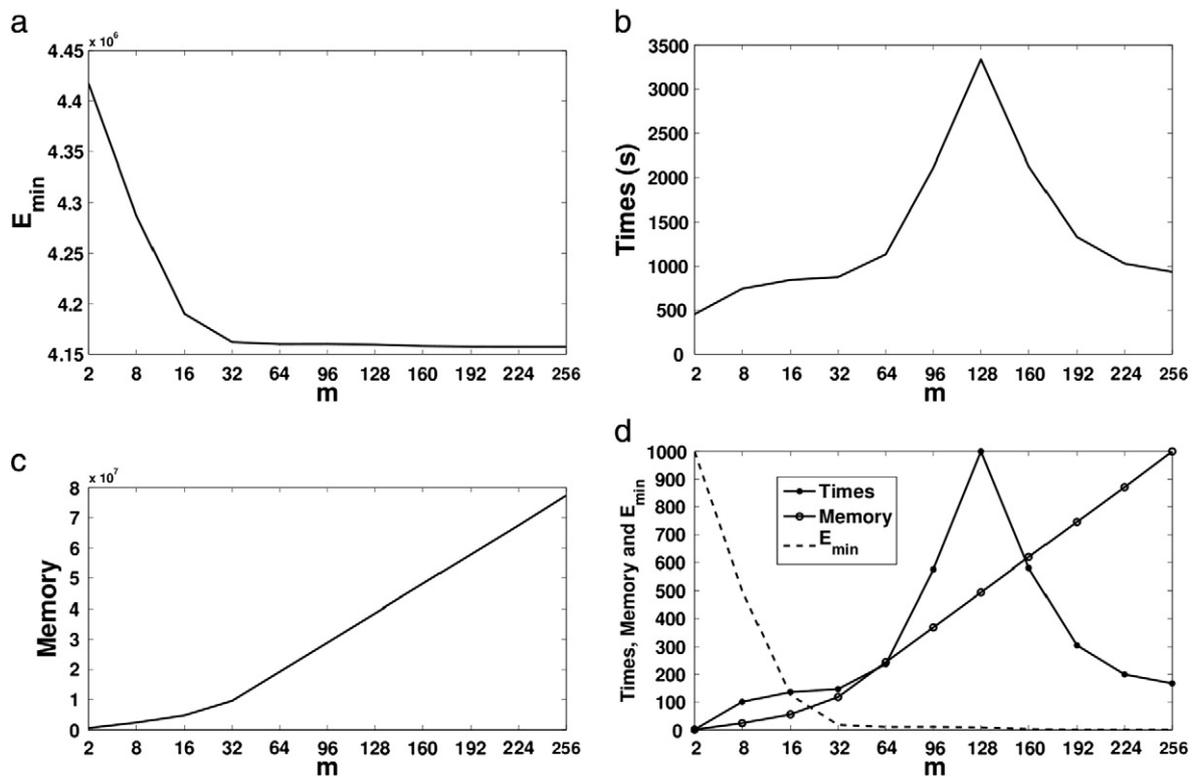


Fig. 11. Plots of times, memory consuming and energy minima obtained by α_m -expansion optimization algorithm performed with an increasing size of label sets \mathcal{L}_m to denoise the squirrel image. (a) Local minima of the energy, (b) time computation (s), (c) number of nodes needed for graph construction, (d) superimposition of the three plots (a), (b), and (c) (note that in (d), all values are normalized in [0,1000] to superimpose the three curves in the same grid).

a posteriori estimation. Thus, the MAP MCPU problem can be seen as an energy minimization problem, where the multichannel data likelihood energy function for any site p is given by [14]

$$E^p(x_p) = \sum_{c=1}^M -\log \left(\frac{1 - \gamma_{p,c}^2}{2\pi(1 - \gamma_{p,c}^2 \cos(y_{p,c} - x_p)^2)} \right) \times \left(1 + \frac{\gamma_{p,c} \cos(y_{p,c} - x_p) \arccos(\gamma_{p,c} \cos(y_{p,c} - x_p))}{\sqrt{1 - \gamma_{p,c}^2 \cos(y_{p,c} - x_p)^2}} \right), \quad (16)$$

where y denotes the observed profile called interferogram (i.e. wrapped and noisy phase), x denotes the regularized phase, M is the total number of the independent interferogram channels and $\gamma_{p,c}$ is the coherence coefficient that depends on pixel p and on channel c .

For the a priori energy function, the discrete total variation (DTV) model can be chosen [13]. As the total variation prior is well adapted when coping with strong discontinuities, it can be used in the case of InSAR applications and it particularly well fits urban scenarios. As we can note, the total MRF energy function is a multi-modal function, thus a highly non-convex one, requiring efficient and robust optimization algorithms to yield satisfying optima.

Experiments shown in this part are performed on simulated data. Two profiles will be considered exhibiting various elevation structures. To generate independent noisy and wrapped phase data (interferograms), four frequencies {5, 6.33, 7.66, 9} GHz are used and an interferometric noise with a constant coherence of $\{\gamma_{p,c} = 0.7; \forall p, c\}$ is added. Fig. 12(c) and (f) shows the 5 GHz noisy interferograms generated from the original elevation profiles of Fig. 12(a) and (d)

Table 7
Computational results of the experiment in Fig. 13.

Figure	Steps to converge	$E_{\min} (\times 10^4)$	Time (min:s)	$ \varepsilon (\times 10^5)$
13(a)	1800	2.9143	0:32	0.16
13(b)	90	2.9997	0:10	0.16
13(c)	6	2.8109	0:12	0.16
13(d)	1	2.6548	0:59	32.77

respectively (Fig. 12(b) and (e) shows the 2D views of the considered profiles respectively). It is important to note that the profile is ambiguous for all the working frequencies. Indeed, there are phase jumps greater than π which violate the *Itoh* condition.

The first synthetic profile is of a size (128×128 pixels) and with a maximum height of 200 m. While the second one seems to be more natural in presenting similar properties to the real InSAR data with a large size (800×300) and a continuous label set with a maximum height of 1000 m.

Similar to the previous experiments, the performances of the proposed LMPM algorithm are presented and compared to the state-of-the-art optimization algorithms.

The phase unwrapping results are presented for different sizes of label sets within the multilabel jump move. This kind of move is the most appropriate to this labeling problem since no quantization of the true label set $[l_{\min}, l_{\max}]$ is needed, where l_{\min} and l_{\max} are respectively the approximate minimum and maximum heights of the reconstructed profile. Starting from an initialization, in each iteration, a set of jump quantities are proposed to the pixels. Different possible strategies to define these quantities are tested. We have verified experimentally that a dichotomy one is more efficient and leads to a good solution within a defined precision. More precisely, the sets \mathcal{L}_m are generated as follows. First, we consider the set of m jump

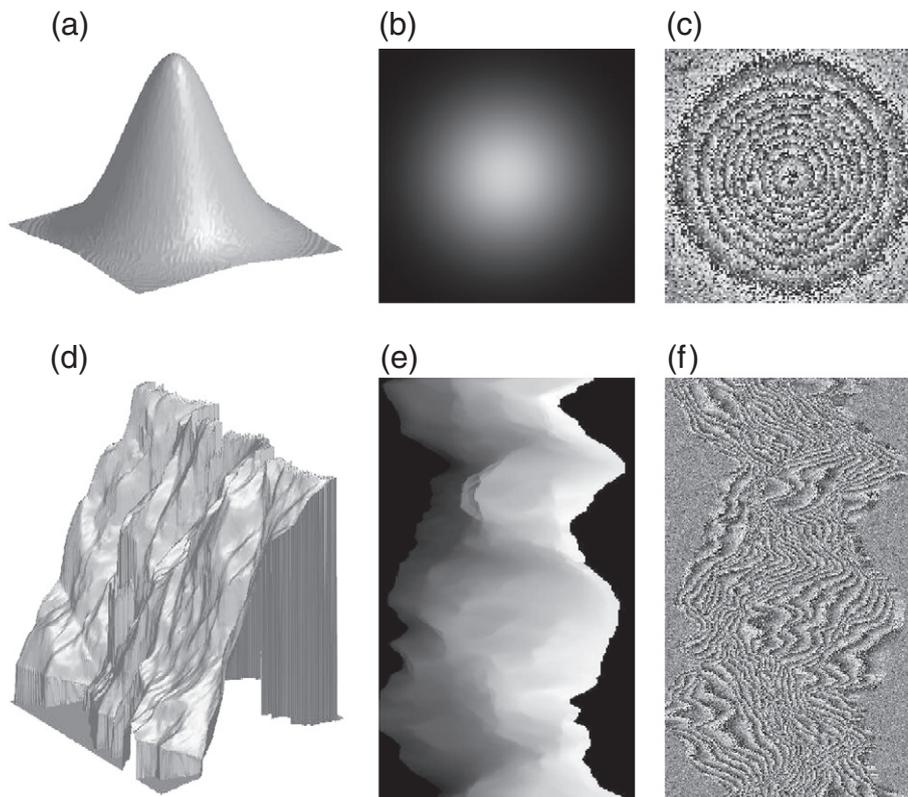


Fig. 12. Synthetic InSAR data: (a) original Gaussian profile (3D view), (b) the corresponding 2D view, and (c) the 5 GHz noisy interferogram. (d) Original mountain profile (3D view), (e) the corresponding 2D view, and (f) the 5 GHz noisy interferogram.

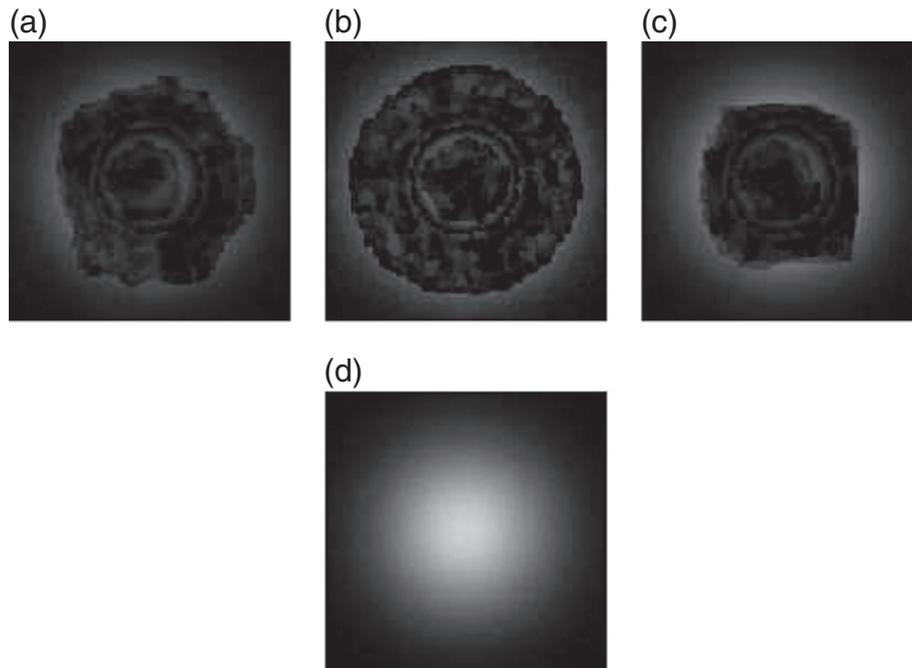


Fig. 13. Phase unwrapping of Fig. 12(c) using several state-of-the-art optimization algorithms: (a) binary jump move of [4], (b) dichotomous binary jump move of [12], (c) fastPD of [22], and (d) exact optimizer of Ishikawa [18].

quantities partitioning $[l_{\min}, l_{\max}]$ in a dichotomous way, i.e., $\mathcal{E}_m = \{-\delta_{\max}, -\frac{\delta_{\max}}{2}, -\frac{\delta_{\max}}{4}, \dots, -\frac{\delta_{\max}}{a_m}, +\frac{\delta_{\max}}{a_m}, \dots, +\frac{\delta_{\max}}{2}, +\delta_{\max}\}$, where δ_{\max} is initially defined depending on l_{\min} and l_{\max} , and $a_m = \frac{1}{2} \log_2(m)$. Then, iteratively, δ_{\max} is reduced and the same process of the label set selection is repeated again until the convergence of the algorithm with a fixed precision on labels. In the next experiment for instance, l_{\min} and l_{\max} are set respectively to 0 m and 200 m, so δ_{\max} is initially set to 200 m and the highest precision on labels is set to 1 m.

The jump move has been already used for the InSAR phase unwrapping and regularization problem in several works [4,12] (Table 7).

Therefore, we compare our results using the LMPM jump moves to the ones obtained by these two works. In [4], a simple binary jump move is considered to unwrap the phase, where partitions increase and decrease iteratively their current labeling with a constant quantity, called jump-step, until its convergence to a local minimum. We shall note that experiments highlighted in the related work are conducted on profiles not ambiguous enough to evaluate the performances of approximate optimization algorithms. Whereas, all the profiles we are using in the current work are ambiguous in a statistical point of view, with similar artifacts to those of the real InSAR data: aliasing, high noise rate and sharp

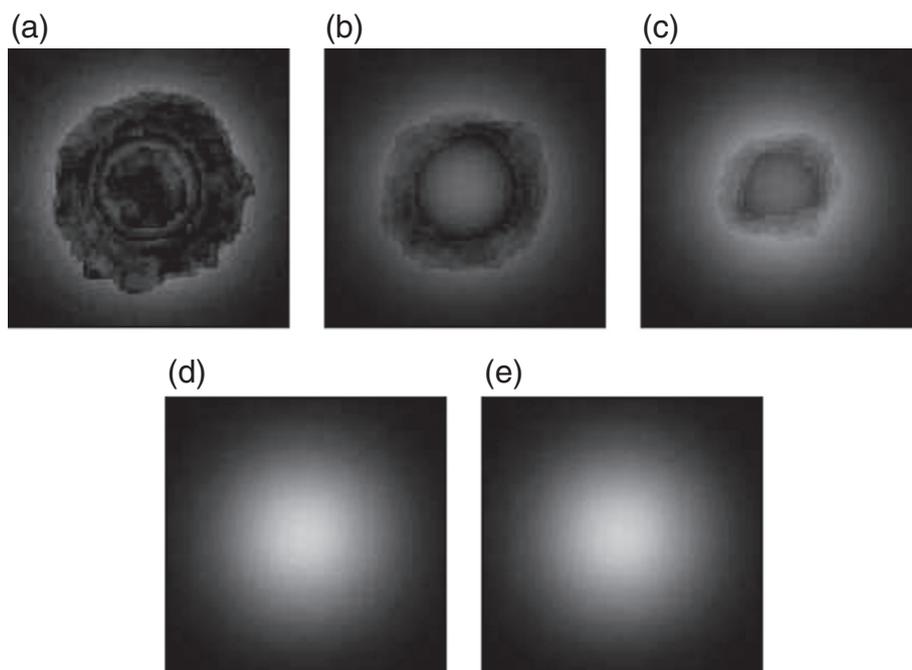


Fig. 14. Phase unwrapping of Fig. 12(c) using the proposed multilabel jump move with subsets of labels for different values of $m \in \{2, 8, 16, 32, 64\}$. (a) δ_2 -jump, (b) δ_8 -jump, (c) δ_{16} -jump, (d) δ_{32} -jump, and (e) δ_{64} -jump.

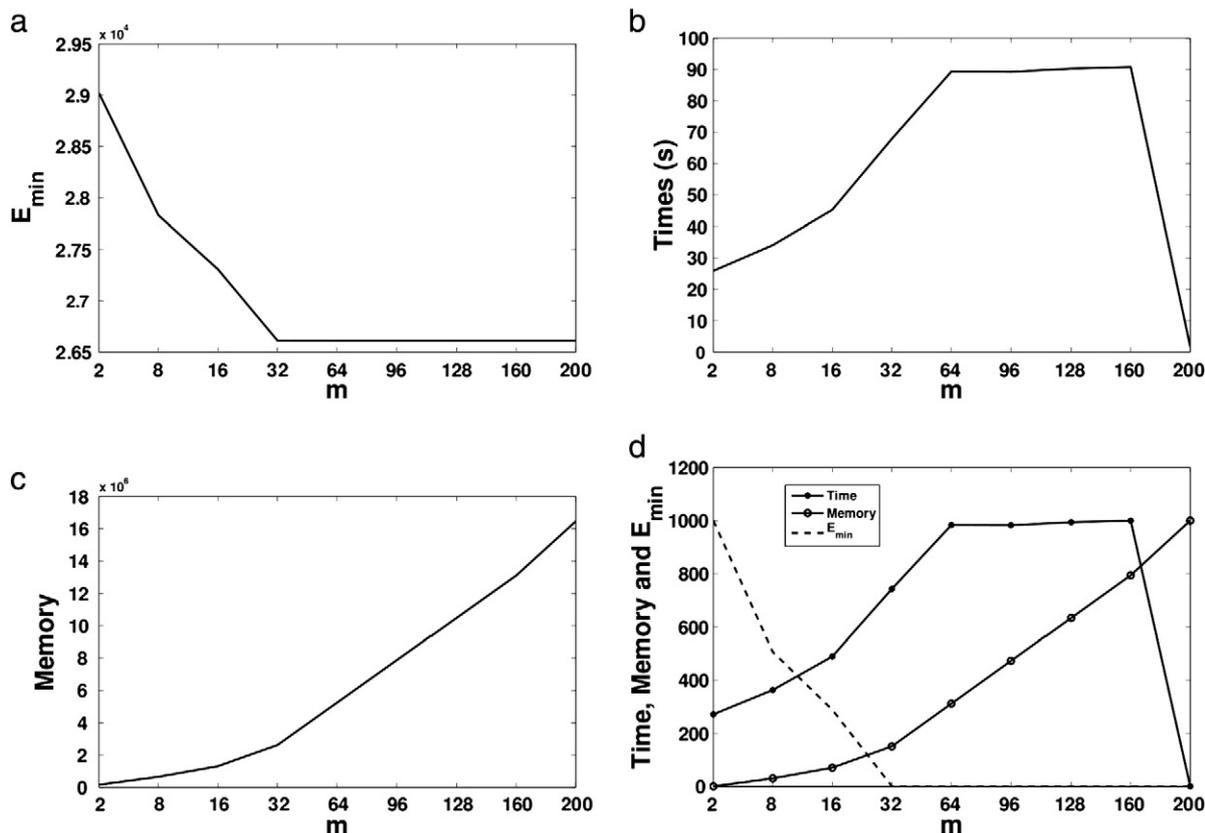


Fig. 15. Plots of times, memory consuming and energy minima obtained by δ_m -jump optimization algorithm performed with an increasing size of label sets to unwrap the Gaussian profile. (a) Local minima of the energy, (b) time computation (s), (c) number of nodes needed for graph construction, and (d) superimposition of the three plots (a), (b), and (c) (note that in (d), all values are normalized in [0,1000] to superimpose the three curves in the same grid).

discontinuities. In [12], a faster version of the binary jump move has been proposed using a dichotomy scheme for the proposed jump-steps in order to regularize the InSAR phase. Thus, the problem of the unwrapping has not been addressed. The latter requires a more robust optimization algorithm to recover the $2k\pi$ multiples of the absolute unwrapped phase.

We also provide a global solution of the unwrapping problem and an approximate one using FastPD [22] within a quantized set of labels in order to show how closer are the optima reached by LMPM moves to a global one compared to the solutions given by the state-of-the-art approximate optimization algorithms.

For the first profile, reconstructed height images using the approximate optimization algorithms of [4,12,22] and the exact optimizer of [18] on the quantized label set $\mathcal{L} = \{0, 1, 2, \dots, 200\}$ are presented in Fig. 13(a), (b), (c) and (d) respectively. As we clearly observe, the optima reached by the approximate optimizers are very poor local optima and far from a global one. Compared to results obtained in the case of image restoration from an impulsive noise, we can note the difficulty that the state-of-the-art approximate MRF optimization algorithms encounter in dealing with the non-convex energies of the phase unwrapping problem.

Table 8
Computational results of the experiment in Fig. 14.

Figure	Steps to converge	$E_{\min} (\times 10^4)$	Time (min:s)	$ \mathcal{E}_m (\times 10^5)$
14(a)	400	2.9021	0:26	0.16
14(b)	220	2.7831	0:36	1.31
14(c)	125	2.7307	0.46	2.62
14(d)	62	2.6613	1:07	5.24
14(e)	30	2.6612	1:30	10.28

The results of the LMPM algorithm with δ_m -jump moves are depicted in Fig. 14 for m which is set to 2, 8, 16, 32 and 64. The results show the contribution of our algorithm to better minimize the energy function than the binary moves in Fig. 13(a) and (b) and the approximate optimizer of [22]. Note also that in this experiment, a local optimum with an energy very close to the global one is reached while using only the 1/7th of memory needed by an exact minimization algorithm (32 labels instead of 200 labels).

Similar to the previous experiments, the multilabel properties stated in Section 3.3 are verified based on the experimental simulations through plots of energy values as a function of size m of the label set \mathcal{L}_m (Fig. 15).

We conduct the same experiment on the second profile to show the robustness and the efficiency of the proposed LMPM jump algorithms to recover a good estimate of the true profile even in presence of complex structures, high noise rate and large scale data (Table 8). We shall indicate that for the current data, performing exact optimization would be computationally prohibitive both for memory and time execution because of the huge graph that has to be built (Table 9). A possible solution to overcome such difficulties could be by reducing the size of the label set, or by partitioning the data into several blocs where optimizations are done in each one. Then, a global solution is obtained by collecting the subproblem solutions. However, we believe

Table 9
Computational results of the experiment in Fig. 16.

Figure	Steps to converge	$E_{\min} (\times 10^5)$	Time (min:s)	$ \mathcal{E} (\times 10^5)$
16(a)	201	4.49	8:00	3.2
16(b)	67	4.3591	3:51	3.2
16(c)	26	3.4781	1:08	3.2

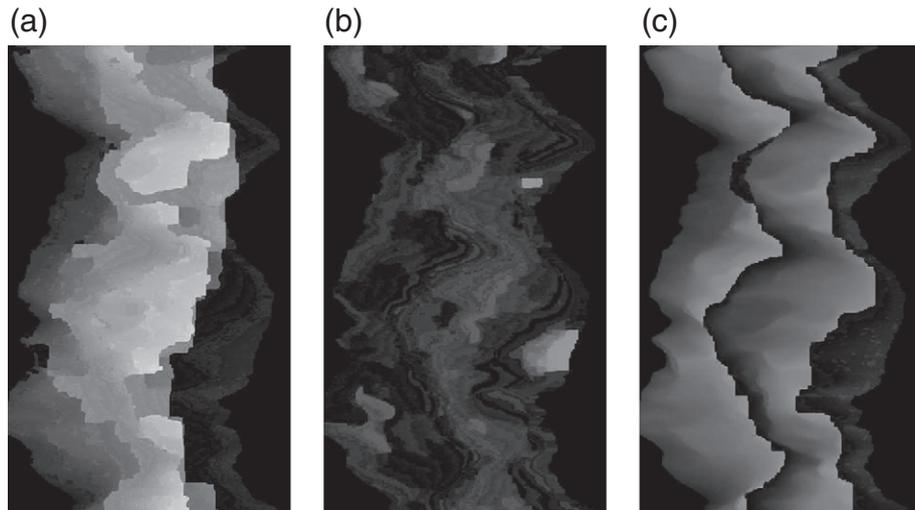


Fig. 16. Phase unwrapping of Fig. 12(f) using several state-of-the-art optimization algorithms: (a) binary jump move of [4], (b) dichotomous binary jump move of [12], and (c) FastPd [22].

that such approaches are not preferred for SAR applications, since on one hand they would not benefit from the global contextual information coming from the whole data, and on the other hand they would

lead to losing some physical information that is necessary for the reconstructed scene analysis. In the light of these requirements, multilabel moves for energy minimization would be an elegant solution

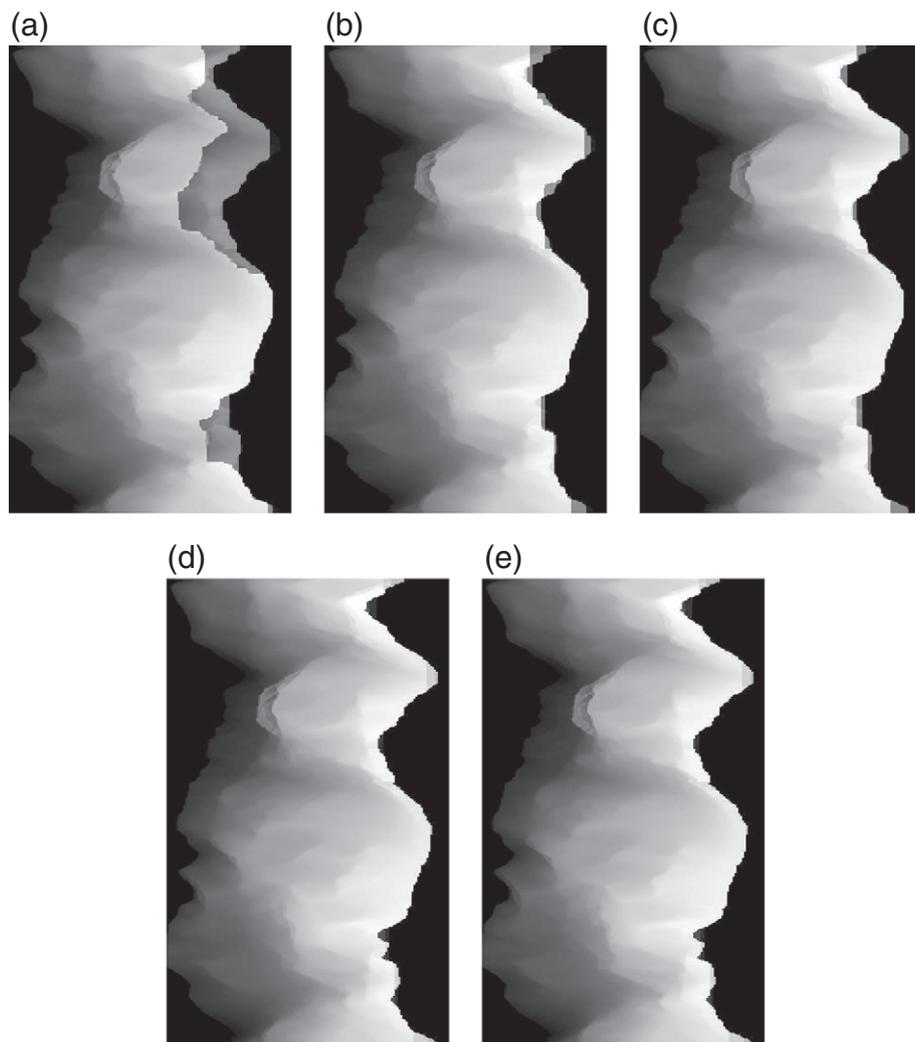


Fig. 17. Phase unwrapping of Fig. 12(f) using the proposed multilabel jump move with subsets of labels for different values of $m \in \{2, 8, 16, 32, 64\}$. (a) δ_2 -jump, (b) δ_8 -jump, (c) δ_{16} -jump, (d) δ_{32} -jump, and (e) δ_{64} -jump.

Table 10

Computational results of the experiment in Fig. 17.

Figure	Steps to converge	$E_{\min} (\times 10^5)$	Time (min:s)	$ \mathcal{E}_m (\times 10^5)$
17(a)	562	2.9519	10:53	6.4
17(b)	111	2.9238	14:31	12.8
17(c)	58	2.9229	15:07	25.6
17(d)	30	2.9228	16:46	51.2
17(e)	16	2.9225	18:30	102.4

and clearly outperform the state-of-the-art approximate optimization algorithms.

Two important conclusions could be driven from the last experiment. First, we clearly see that performing the binary jump move with multilabel graphs (see Fig. 17(a)), i.e., making the optimal move from the three possible states: increasing or decreasing or keeping the current labeling, gives much more better results than the iterative schemes proposed in [4,12] (see Fig. 16(a) and (b)). Thus, we believe that for the less ambiguous area to reconstruct, the proposed δ_2 -jump move making algorithm would guarantee better results, or even a convergence toward a global optimum of the energy, in a short amount of time, even for a large scale data (Table 10). The second conclusion is that the requirement of a huge memory for graph allocation in order to converge toward a global optimum of the energy is well managed using the proposed multilabel moves; since local optima that are very close to a global one could be from now on reached while using only some labels within the set of all possible labels. This fact is of a great interest in many real applications, for the satellite and medical image processing for instance, due to the data size and continuous label sets. Multilabel move making algorithms provide a robust and efficient method that offers a trade-off between complexities (in time and memory) and the solution's quality.

5. Conclusion

We have proposed in this work new graph-cut based optimization algorithms for approximate optimization of pairwise Markovian energies with convex prior. The algorithms are based on the concept of iterative partition moves that converge to a local or global optimum depending on the number of chosen labels during the optimization process. This choice leads to a new move that we called multilabel move which is able to minimize complex Markovian energy functions more accurately, i.e., converging to better optima of energies, while less memory is needed. This trade-off is necessary in many image processing and computer vision problems when working with high dimension data with complex noise affecting them. The results presented in this work show that better solutions are obtained using this approach compared to those using binary or multi-range moves and with less memory than those of the exact optimization methods. A future work is dedicated to the fusion of these proposed deterministic optimization algorithms with a stochastic approach in order to guarantee a convergence to a better optimum while using more restricted label set sizes (i.e., less memory).

References

- [1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
- [2] R. Bamler, P. Hartl, Synthetic aperture radar interferometry, *Inverse Probl.* 14 (1998) R1–R54.
- [3] J. Besag, On the statistical analysis of dirty pictures, *J. R. Stat. Soc. B* 48 (1986) 259–302.
- [4] J.M. Bioucas-Dias, G. Valadao, Phase unwrapping via graph cuts, *IEEE Trans. Image Process.* 16 (3) (2007) 698–709.
- [5] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (9) (2004) 1124–1137.
- [6] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (11) (2001) 1222–1239.
- [7] P. Carr, R. Hartley, Solving multilabel graph cut problems with multilabel swap, in: *Digital Image Computing: Techniques and Applications*, 2009, pp. 532–539.
- [8] J. Darbon, Composants logiciels et algorithmes de minimisation exacte d'énergies dédiés au traitement des images. Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications de Paris, 2005.
- [9] J. Darbon, Global optimization for first order Markov random fields with submodular priors, *Discret. Appl. Math.* 157 (16) (2009) 3412–3423.
- [10] J. Darbon, M. Sigelle, Image restoration with discrete constrained total variation part i: fast and exact optimization, *J. Math. Imaging Vis.* 26 (3) (2006) 261–276.
- [11] J. Darbon, M. Sigelle, Image restoration with discrete constrained total variation part ii: levelable functions, convex priors and non-convex cases, *J. Math. Imaging Vis.* 26 (3) (2006) 277–291.
- [12] L. Denis, F. Tupin, J. Darbon, M. Sigelle, SAR image regularization with fast approximate discrete minimization, *IEEE Trans. Image Process.* 18 (7) (2009) 1588–1600.
- [13] G. Ferraioli, A. Shabou, F. Tupin, V. Pascasio, Multichannel phase unwrapping with graph-cuts, *IEEE Geosci. Remote. Sens. Lett.* 6 (3) (2009) 562–566.
- [14] G. Ferraiuolo, V. Pascasio, G. Schirinzi, Maximum a posteriori estimation of height profiles in InSAR imaging, *IEEE Geosci. Remote. Sens. Lett.* 1 (2004) 66–70.
- [15] D. Geman, S. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* 6 (1984) 721–741.
- [16] D.M. Greig, B.T. Porteous, A.H. Seheult, Exact maximum a posteriori estimation for binary images, *J. R. Stat. Soc. B* 51 (2) (1989) 271–279.
- [17] P.L. Hammer, Some network flow problems solved with pseudo-boolean programming, *Oper. Res.* 13 (1965) 388–399.
- [18] H. Ishikawa, Exact optimization for Markov random fields with convex priors, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (10) (2003) 1333–1336.
- [19] K. Itho, Analysis of the phase unwrapping problem, *Appl. Opt.* 2 (14) (1982).
- [20] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2) (2004) 147–159.
- [21] N. Komodakis, G. Tziritas, Approximate labeling via graph cuts based on linear programming, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (8) (2007) 1436–1453.
- [22] N. Komodakis, G. Tziritas, N. Paragios, Performance vs computational efficiency for optimizing single and dynamic MRFs: setting the state of the art with primal-dual strategies, *Comput. Vis. Image Underst.* 112 (1) (2008) 14–29.
- [23] V. Lempitsky, C. Rother, S. Roth, A. Blake, Fusion moves for Markov random field optimization, *IEEE Trans. Pattern Anal. Mach. Intell.* (2009) 1392–1405.
- [24] S. Ramalingam, P. Kohli, K. Alahari, P. Torr, Exact inference in multi-label CRFs with higher order cliques, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [25] A. Shabou, F. Tupin, J. Darbon, A graph-cut based algorithm for approximate MRF optimization, in: *IEEE International Conference on Image Processing*, 2009, pp. 2413–2416.
- [26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for Markov random fields with smoothness-based priors, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (6) (2008) 1068–1080.
- [27] O. Veksler, Efficient graph-based energy minimization methods in computer vision. Ph.D. thesis, Cornell University, 1999.
- [28] O. Veksler, Graph cut based optimization for MRFs with truncated convex priors, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [29] O. Veksler, Multi-label moves for MRFs with truncated convex priors, in: *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009, pp. 1–13.