RECURSIVE RECONSTRUCTION OF PIECEWISE CONSTANT SIGNALS BY MINIMIZATION OF AN ENERGY FUNCTION

ANASS BELCAID∗, MOHAMMED DOUMI
AND ABDELKADER FASSI FIHRI
Ecole Nationale Supérieure d’Arts et Métiers
Meknès, Morocco

(Communicated by Naoki Saito)

Abstract. The problem of denoising piecewise constant signals while preserving their jumps is a challenging problem that arises in many scientific areas. Several denoising algorithms exist such as total variation, convex relaxation, Markov random fields models, etc. The DPS algorithm is a combinatorial algorithm that excels the classical GNC in term of speed and SNR resistance. However, its running time slows down considerably for large signals. The main reason for this bottleneck is the size and the number of linear systems that need to be solved. We develop a recursive implementation of the DPS algorithm that uses the conditional independence, created by a confirmed discontinuity between two parts, to separate the reconstruction process of each part. Additionally, we propose an accelerated Cholesky solver which reduces the computational cost and memory usage. We evaluate the new implementation on a set of synthetic and real world examples to compare the quality of our solver. The results show a significant speed up, especially with a higher number of discontinuities.

1. Introduction. In this paper, we focus on recovering Piecewise constant (PWC) signals from noisy measurements. A PWC signal is characterized by a set of flat regions, called plateaus or levels, with a finite number of instantaneous discontinuities also called jumps, shifts, steps or singularities according to the discipline. This class of signals arises in several fields, including geophysics [16], bioinformatics [2], biology [14] and digital imagery [3]. PWC signals offer two important pieces of information: the values of the plateaus and the boundaries located in each discontinuity. Thus, a denoising technique must not only recover the value of each plateau, in the true signal, but also detect and preserve its discontinuities. So the main problem could be tackled as a joint restoration-segmentation inverse problem, which generally leads to better results than executing each task independently [24].

We focus on the classical degradation model where the observed discrete signal \( y \) is related to the underlying discrete signal \( x \) by:

\[
y = x + b, \quad x, y, b \in \mathbb{R}^{n+1}
\]
The ground truth signal is considered to be PWC without any limitation on the plateaus values, \( x_i \in \mathbb{R} \), and the number of discontinuities is unknown. Additionally, \( x \) is corrupted by a zero-mean additive white Gaussian noise \( b \) with a standard deviation \( \sigma \) (Figure 1).

Usually, the inverse problem in Eq. (1) is tackled by minimizing the following energy function:

\[
E(x) = \sum_{i=0}^{n+1} (x_i - y_i)^2 + \lambda^2 \sum_{i=0}^{n-1} \varphi(x_{i+1} - x_i)
\]

The first term in the equation is called the data fidelity term which ensures closeness between the signal \( x \) and the observation \( y \). In a statistical setting, the choice of an \( l_2 \) data fidelity term implicitly corresponds to the assumption that the noise is white Gaussian. The second term is called the smoothness term and it is added since the information provided by the model in Eq. (1) is not sufficient to compute an adequate solution. The smoothness term is expressed as a sum of Potential function (PF)’s over neighboring points [4]. The PF is chosen to force the solution to exhibit a set of a priori features like the presence of discontinuities in our case. Finally, the parameter \( \lambda^2 \) (whose choice is a difficult problem by itself [12]) is set to control the trade-off between the fidelity term and the prior constraints.

The minimization of the energy function in Eq. (2) is equivalent to the Maximum A Posteriori (MAP) in the Markov random fields (MRF) framework. The global minimizer \( x^* \) of Eq. (2) maximizes the conditional probability \( p(x|y) \) given the observable data \( y \). The data fidelity term represents the likelihood in the MRF framework and the second term serves as the prior distribution \( p(x) \). Numerous PF’s have been tested in the literature [5] for the denoising task. This is achieved by forcing a smoothness prior among interacting sites. But an everywhere smoothness constraint, represented by a PF \( V(r) = r^2 \), is not adequate to preserve large differences located in the discontinuities. Therefore, the need of an edge-preserving PF, that allows the recovery of discontinuities in the original signal while applying smoothness elsewhere. A solution was proposed by (D.Geman and S.Geman) in their line process model [18] or the weak string constraint by (A.Blake and A.Zisserman) [1]. The weak string model uses a truncated quadratic interaction function \( V(r) = \min(\alpha, r^2) \) to switch off the smoothness prior when the gradient magnitude exceeds a given threshold. Above this threshold, the quadratic smoothness is replaced by a constant penalty indicating the presence of a discontinuity. The weak string model uses a deterministic approach to represent the signal. The same energy was obtained by (D.Geman and D.Geman) [18] using the MRF framework and an additional boolean variable called Line Process (LP). The switch off between the quadratic and constant penalty is explicitly represented by the boolean variable indicating the presence or absence of a discontinuity. So given a parameter \( h \) called the sensitivity that represents the minimal value of a jump, a discontinuity is detected if the difference \( |x_{s+1} - x_s| \) exceeds \( h \).

\[
l_i = \begin{cases} 
1 & \text{if } |x_i - x_{i+1}| > h \\
0 & \text{otherwise}
\end{cases}
\]

As was proven in [6], the sensitivity \( h \) is related to the truncated constant \( \alpha \) by \( h = \sqrt{\frac{2\alpha}{\lambda}} \). The energy function for the LP model becomes a function of two
variables: the estimated signal $x$, and the LP $l$:

$$E(x, l) = \sum_{i=0}^{n} (x_i - y_i)^2 + \sum_{i=1}^{n} \lambda^2 (x_i - x_{i-1})^2 (1 - l_i) + \sum_{i=1}^{n} \alpha l_i$$

The parameters $\alpha$ and $\lambda$ are empirical constants that must be fixed a priori. Therefore, the energy function (4) must be minimized for several parameters in order to find an adequate solution. Minimizing the energy function in Eq. (4) leads to a general, nonconvex and mixed variables optimization problem [19].

$$\text{(P)} \begin{cases} \min E(x, l) \\ x \in \mathbb{R}^{n+1} \\ l \in \{0, 1\}^n \end{cases}$$

The Problem (5) is mixed since it involves a continuous variable $x$ and an integer binary variable representing the LP $l$. The nonconvexity comes from the use of the nonconvex truncated quadratic PF. Although a considerable effort was made to develop convex edge-preserving PF [7, 9], nonconvex PF offers better performance for signals with extreme discontinuities [22].

Finding a global minimizer of $E$ is an NP-hard problem [8] due to the use of an edge-preserving PF. The energy function is nonconvex with a large number of local minima, one for each configuration $l$ of the LP. Also, the energy is not smooth at the global minimizer which makes local minimization techniques impractical.

Several global optimization techniques were proposed to compute an approximation of the global minimizer. These methods are either based on stochastic approach or continuous deterministic relaxation. In the statistical approach, we mention the Iterated Conditional Modes (ICM) [4] and the Simulated Annealing (SA) [18, 17]. The results of these algorithms are extremely inefficient since ICM is sensible to the initial guess and the cooling schedule of SA is slow. In the relaxation approach, we mention (Blake and Zisserman) Graduated non-convexity (GNC) algorithm [6] which is a deterministic annealing method for approaching a global minimum with an unconstrained continuous variable. The GNC algorithm excels SA in both speed and quality of the solution [6]. For a detailed account of the history, we refer the reader to [21, 27].
The combinatorial Discontinuity Position Sweep (DPS) algorithm offers a better alternative to GNC in terms of running time and restoration quality [15] for short and medium duration signals. Unfortunately, DPS becomes very slow for long duration signals, as it has to solve an important number of the linear systems with size \( n \), where \( n \) is the length of the signal.

In this paper, we present a recursive scheme to implement the DPS algorithm that drastically reduces its running time. The recursive scheme is based on the conditional independence, between two parts of the signal, that emerges after the confirmation of a jump position. A direct effect of this splitting approach is the shrinkage of the linear systems. Furthermore, we propose an accelerated Cholesky solver that avoids recomputing the factorization of each system, but instead, deduces it from a unique factorization corresponding to the initial system.

The remaining of the article is organized as follows. In section 2, we present the key ideas behind the classical DPS algorithm that transforms the Pb. (5) into a purely combinatorial one. This could be considered as a dual approach to GNC’s which keeps the continuous variable \( x \) and eliminates the LP. For a fixed LP \( l \), finding the optimal solution over \( x \) is an easy task since it only involves solving a tridiagonal linear system. DPS must then search among all possible configurations of the LP space \( \mathcal{L} \). Since the number of LP’s is exponential, \( \text{card}(\mathcal{L}) = 2^n \), a brute force search is unpractical. DPS overcomes this difficulty by structuring all the configurations of \( \mathcal{L} \) into the nodes of a hypercube and integrates an efficient search strategy that prunes an important part of \( \mathcal{L} \). This strategy is based on a unique decomposition of the hypercube into a set of disjoint levels.

2. The DPS algorithm. The key formulation of the DPS algorithm is to transform the mixed problem (5) into a pure combinatorial one. This could be considered as a dual approach to GNC’s which keeps the continuous variable \( x \) and eliminates the LP. For a fixed LP \( l \), finding the optimal solution over \( x \) is an easy task since it only involves solving a tridiagonal linear system. DPS must then search among all possible configurations of the LP space \( \mathcal{L} \). Since the number of LP’s is exponential, \( \text{card}(\mathcal{L}) = 2^n \), a brute force search is unpractical. DPS overcomes this difficulty by structuring all the configurations of \( \mathcal{L} \) into the nodes of a hypercube and integrates an efficient search strategy that prunes an important part of \( \mathcal{L} \). This strategy is based on a unique decomposition of the hypercube into a set of disjoint levels.

2.1. Hypercube representation of the LP. The first idea of DPS is to represent all the configurations of the LP in \( \mathcal{L} = \{0, 1\}^n \) as the nodes of a hypercube or \( n \)-cube. This graph has \( 2^n \) nodes, which we encode as \( n \) bits words using Gray codes [20]. The resulting encoding ensures that two nodes are neighbors if and only if they differ only by one digit.

Each encoding of a node \( l \) in the hypercube \( \mathcal{L} \) could be represented by a set \( I \subset \{1, 2, \ldots, n\} \) that regroups the confirmed discontinuities. We then use the notation \( l = e^I \) such as:

\[
(6)\quad l = e^I_k = \begin{cases} 
1 & \text{if } k \in I \\
0 & \text{otherwise}
\end{cases}
\]

We associate to each set \( I \), an energy function \( E_I(x) = E(x, e^I) \). With this notation, the mixed Pb. (5) could be rewritten as:

\[
(7)\quad \min_{x \in \mathbb{R}^{n+1}} \min_{e^I \in \mathcal{L}} E_I(x)
\]

the energy function in (7) is quadratic and strictly convex, therefore, it admits a unique global minimum. The global minimum \( x^*_I \) is then characterized by the
Recursive Reconstruction of PWC signals

stationarity condition:

\( \nabla E_I(x_I^*) = 0 \)

In order to compute the global minimum \( x_I^* \) of \( E_I \), we transform (8) into a linear system:

\( A(I)x_I^* = y \)

where the matrix \( A(I) \) is tridiagonal, symmetric and definite-positive.

By Eq. (9) we established that for a fixed LP \( l \) indexed by \( I \) computing its associate energy is reduced to solving a linear system. The solution is cheaply computed since the main matrix presents all the advantages in term of symmetry, positivity, and sparsity. As a result, the mixed problem Pb. (5) could be transformed into the following pure combinatorial problem:

\[
(P_2) \begin{cases} 
\min \left( E(A(I)^{-1}y, e^I) \right) \\
e^I \in \mathcal{L}
\end{cases}
\]

2.2. Sweep strategy. Solving (10) is theoretically simple [23], nevertheless, a direct approach, that enumerates all the configurations, is not practical. Since the cardinal of \( \mathcal{L} \) is exponential \( \text{card} (\mathcal{L}) = 2^n \). DPS structures the nodes of \( \mathcal{L} \) in terms of levels and proposes a sweep enumerative search strategy that reduces the number of visited nodes.

In order to guide our search, we use the notion of level in a hypercube. A level \( \mathcal{L}_k \) of the hypercube contains all the configurations that have exactly \( k \) discontinuities. As a result, the hypercube \( \mathcal{L} \) is decomposed into the following disjoint levels (Figure 2a).

\( \mathcal{L} = \bigcup_{k=0}^{n} \mathcal{L}_k \)

where \( \mathcal{L}_k = \{ e^I \in \{0, 1\}^n / \text{card}(I) = k \} \) are the \( (n + 1) \) levels of the hypercube.

We construct a sequence of optimization problems \( (P_k)_{k \in [0,n]} \) based on (11) to limit the search on a given level.

\[
(P_k) \begin{cases} 
\min \ E(A(I)^{-1}y, E^I) \\
e^I \in \mathcal{L}_k
\end{cases} \quad \text{with } k \in \{0 \ldots n\}
\]

In order to compute the solution of \( (P_k) \), we must solve \( \text{card}(\mathcal{L}_k) = \binom{n}{k} \) tridiagonal linear systems. DPS prunes the search space of \( (P_k) \) by applying a cutting enumerative search strategy that limits the domain of \( (P_k) \) to the neighbors of \( e^{I_{k-1}} \). Where \( e^{I_{k-1}} \) is the optimal solution of \( (P_{k-1}) \). In consequence, a confirmed discontinuity will always be kept in the upcoming levels.

The set of neighbors of \( e^{I_{k-1}} \) will be denoted as follows:

\( \mathcal{N}_k(e^{I_{k-1}}) = \bigcup_{i \in I_{k-1}} \{ e^{I_{k-1}\cup\{i\}} \} \subset \mathcal{L}_k \) where \( I_{k-1}^c = \{1 \ldots n\} \setminus I_{k-1} \)

The pruning strategy will lower the number of visited nodes from \( \text{card}(\mathcal{L}_k) = \binom{n}{k} \) to the \( n - k - 1 \) neighbors of \( e^{I_{k-1}} \).

We thereby converted the independent sequence of problems \( (P_k) \) into a cheaper recursive sequence of problems \( (Q_k) \). The definition of \( (Q_k) \) is given by:

\[
(Q_k) \begin{cases} 
\min \ E(A(J)^{-1}y, e^J) \\
e^J \in \mathcal{N}_k(e^{I_{k-1}})
\end{cases}
\]
Figure 2. DPS search space and its pruning process. (Figure 2a) shows the search space (an Hypercube) structured in levels where each level regroups the set of LP with equal number of 1 bits. (Figure 2b) illustrates DPS pruning process which only considers the neighbors of the previous optimal solution. As an example, in the second level, we already found the optimal LP (0100) in the first level. By this information, DPS will only consider its neighbors (red and blue nodes) and will prune the rest (empty nodes). Additionally, DPS will stop the search if the energy did not decrease from a level to another. In the example, we stop the search at the second level (red dashed line) since the energy increased from the first level to the second one.

The sequence of problems is solved by first exploring the level $L_0$, which consists of seeking a solution with no discontinuities. Then at each level, we keep the found discontinuities, and we search if the introduction of a new discontinuity could decrease the energy function, if not the rest of the levels are pruned and the optimal solution lies within the actual level. The resulted technique is called Discontinuity Position Sweep (DPS) and is illustrated in (Figure 2b).

3. The recursive and accelerated DPS algorithm. The DPS algorithm has a $O(n^2)$ time complexity and $O(n)$ space complexity which makes it very fast for small and medium size signals. However, its execution time increases substantially for large signals like the case of images. The main reason for this bottleneck is due to the number and the size of the solved linear systems. As a result, the applicability of DPS for large data is reduced tremendously.

We propose a recursive implementation called Recursive Discontinuity Position Sweep (RDPS) that reduces the size of the solved systems by disconnecting the nodes of the hypercube $L$. This decomposition is based on the conditional independence created by a confirmed discontinuity. This means, discovering a discontinuity at random position $p$, will allow us to compute the parts $x_{[1,p]}$ and $x_{[p,n]}$ independently. In consequence, the hypercube will shrink as we discover new discontinuities.
Algorithm 1: The DPS algorithm

Data: $\lambda$, $h$, $y$

Initialization $k = 0$, $I_k = \emptyset$, $\alpha = \frac{h^2 \lambda}{2}$.

$x_{I_k} = A(I_k)^{-1} y$

$E_{I_k} = E(x_{I_k}, e^{L_k})$.

repeat

$k = k + 1$

$(x^*, e^{L^*}) = \arg\min_{e^{L} \in N_{\alpha}(e^{L_{k-1}})} E(A(L)^{-1} y, e^L)$

Compute $E_{I_k} = E(x^*, e^{L^*})$

if $(E_{I_k} < E_{I_{k-1}})$ then

$I_k = L^*$

$x_{I_k} = x^*$

end

until $(E_{I_k} > E_{I_{k-1}} \text{ ou } k > n)$

Result: $x^*$ the restored signal

We also propose an accelerated Cholesky solver that avoids the computation of each factorization but instead extracts them from a cached matrix $C$. This is possible since all the solved matrices share a common structure and differ only by their size. We denote $A_\lambda$ the set of all the solved matrices, we prove that, if we know the Cholesky factorization of a matrix $A \in A_\lambda$, we could easily compute the Cholesky factorization of any matrix in $A_\lambda$ with lower dimension than $A$. Therefore, the accelerated algorithm will only compute the Cholesky factorization $C$, of the first matrix $A(\emptyset)$, and then will extract all the other factorizations from $C$.

3.1. The recursive implementation of DPS. The DPS algorithm uses a breadth first search in the levels of the hypercube with a cutting strategy that only considers the neighbors of the optimal solution in the previous level (14). Thus, a discontinuity $i_k$, introduced in the level $k$, will be present in all the optimal solutions of the superior levels.

We make use of this property to stop the search process as soon as we encounter the first discontinuity. The search in the upcoming levels is then replaced by a new two independent sub-problems, where each sub-problem is concerned only with restoring a part of the signal. The independence of the two sub-problems is proved in the next proposition:

Proposition 1. If we introduce a discontinuity $i_k$, recovering the signal $x$ could be replaced by recovering $x_{[1,i_k]}$ and $x_{[i_{k+1}, n]}$ independently.

Proof. Restoring the signal with only one discontinuity $i_k$, is equivalent to solving the linear system $A(I)x = y$, where $I = \{i_k\}$.

The matrix $A$ is symmetric tridiagonal and has the following form:

\begin{equation}
A(I) = \begin{pmatrix}
d_0 & -b_0 & & \\
-b_0 & d_1 & -b_1 & \\
& -b_{n-1} & d_{n-1} & -b_n \\
& & -b_n & d_n
\end{pmatrix}
\end{equation}
where $b_i = \lambda^2(1 - e^T_i)$ and $d_i = 1 + b_i + b_{i-1}$ except for $d_0 = 1 + b_0$ and $d_n = 1 + b_n$.

Since $e^T_{i_k} = 1$, the terms $A(I)_{i_k,i_k+1}$ and $A(I)_{i_k-1,i_k}$ vanishes, and $A(I)$ becomes a diagonal block matrix. As as result, we could compute $x$ as:

\begin{equation}
A(I)x = y \iff \begin{pmatrix} A_{1,i_k} & 0 \\ 0 & A_{i_k+1,n} \end{pmatrix} x = y \iff \begin{cases} A_{1,i_k}x_{1,i_k} = y_{1,i_k} \\ A_{i_k+1,n}x_{i_k+1,n} = y_{i_k+1,n} \end{cases}
\end{equation}

Which proves that we can compute $x_{1,i_k}$ and $x_{i_k+1,n}$ independently.

This fundamental property will allow us to reformulate the problem (5) in a sequence of recursive sub-problems $(R_y)$. Since each problem is decomposed in two sub-problems, the decomposition process follows the classical binary tree structure for a recursive formulation.

We define the problem $(R_y)$ as:

\begin{equation}
(R_y) \left\{ \begin{array}{l}
\min E_y(x,l) \\
l \in \mathcal{L}_y^{m}
\end{array} \right.
\end{equation}

Where $\mathcal{L}_y^{m} = \{ e \in \{0,1\}^m \mid \sum_j e_j = 1 \text{ or } \sum_j e_j = 0 \}$ is the LP subset with dimension $m$ containing at most one discontinuity. Which means that we restrict the search only on the first two levels.

The initial problem (10) is then equivalent to solving $(R_y)$ which considers the entire signal. For an instance $(R_{y(l,s)})$ of the sequence $(R_y)$, we compute the optimum $l^*$ and if $l^* \in \mathcal{L}_0$ we stop the recurrence since the signal $y_{l^*,s}$ is constant. In the other hand, if $l^* \in \mathcal{L}_1$, we confirm that the signal $y_{l^*,s}$ contains a unique discontinuity in the position $i_k$, such as $l_{i_k}^* = 1$. The problem $R_{y(l,s)}$ is then divided into the two sub-problems $(R_{y(l,i_k)})$ and $(R_{y(l,i_k+1,s)})$.

This recursive process defines a binary tree $T = (R, E)$ where each node is an optimization problem concerned with the recovery of a part of the signal, and the discontinuities denote separations conditioned on an accepted discontinuity $i_k = 1$.

**Figure 3.** (a): The decomposition binary tree, for restoring a signal with three discontinuities $I = \{i_1, i_2, i_3\}$, (b) the recursive process where the final solution is obtained by regrouping the leaves of the tree.
We propose a simple brute-force algorithm to solve an instance of the sequence $(R_y)$. The algorithm considers all the nodes of the levels $L_0$ and $L_1$, and returns the node with the minimal energy. The algorithm (2) presents an efficient test to detect the presence of a discontinuity in a given part of the signal. The recursive implementation of RDPS will depend heavily on this test to differentiate a base case of the recurrence (a constant signal) with a decomposed one that needs further search for new discontinuities. As mentioned above, the decomposition of the signals follows a binary tree structure, and the optimal solution will be gathered by regrouping all the constant signals in the leaves of the tree.

This recursive approach rapidly reduces the size of the solved systems, and only the first search considers full-size matrices. This is illustrated in (Figure 4), which plots the number of these systems as a function of their sizes. The figure shows that the majority of these systems are of small size due to the shrinking effect of introducing a new discontinuity, and the number of full-size matrices becomes negligible.

### Algorithm 2: Brute force Solver for the problem $(R_y)$.

| Data: | Noisy signal $y$ |
|---|---|
| $n ← |y|$ |
| $E_0 = E_y(A^{-1}(∅)y, e^0)$ |

/* Compute the set of energies in the level $L_1$ */

\[
\text{for } i_k ← 1, \ldots, n \text{ do}
\]

\[
A(\{i_k\})[1,i_k][x[1,i_k]] = y[1,i_k]
\]

\[
A(\{i_k\})[i_k+1,n][x[i_k+1,n]] = y[i_k+1,n]
\]

\[
E_{i_k} = E_{λ,α}(x,e(i_k))
\]

/* Compute the line process with the minimal energy. */

\[
e^* = 0
\]

if $\min_k E_{i_k} < E_0$ then

/* Getting the position of the discontinuity. */

\[
\begin{align*}
&i_k ← \text{argmin}_k E_k \\
&e^*_k ← 1
\end{align*}
\]

Result: $e^*$ the line process with the minimal energy.

3.2. **Accelerated DPS implementation.** The brute force search (algorithm 2) must check the energy value in each LP in the first two levels of the hypercube. This requires solving an important number of systems in the form (18). Conversely, we remark that those matrices share a common structure, and differ only by their size. This common structure is defined by the family:

\[
A_λ = \{A(n) ∈ M_n \mid n ∈ N\}
\]
**Function** ReconstructionRDPS(y,x)

**Data:** Noisy signal y

**Result:** Reconstructed Piecewise constant signal x

\( n \leftarrow |y| \)

/* Compute the line process with the minimal energy. */

\( i^* = \arg\min_{i_k \in \{1,...,n\}} E_{\lambda,\alpha}(A^{-1}(\{i_k\})y, e^{i_k}) \)

/* Verify if the energy decreased. */

if \( E_{\lambda,\alpha}(A^{-1}(\{i^*\})y, e^{i^*}) < E_{\lambda,\alpha}(A^{-1}(\emptyset)y, e^{\emptyset}) \) then

/* Call the subroutine on the left part. */

\( y_l \leftarrow (y_k)_{1 \leq k \leq i^*} \) and \( x_l \leftarrow (x_k)_{1 \leq k \leq i^*} \)

ReconstructionRDPS(y_l, x_l)

/* Call on the right part. */

\( y_r \leftarrow (y_k)_{i^* < k \leq n} \) and \( x_r \leftarrow (x_k)_{i^* < k \leq n} \)

ReconstructionRDPS(y_r, x_r)

/* Concatenate the two parts. */

\( x \leftarrow [x_l, x_r] \)

end

else /* Signal is constant. */

\( x = A^{-1}(\emptyset)y \)

end

**Figure 4.** Number of solved matrices as a function of their size. The example restores a signal of size 1024 with 8 regularly distributed discontinuities.

where \( \mathcal{M}_n \) is the set of square matrices of size \( n \) and the expression of \( A(n) \) is given as:

\[
A(n) = \begin{pmatrix}
1 + \lambda^2 & -\lambda^2 & -\lambda^2 \\
-\lambda^2 & 1 + 2\lambda^2 & -\lambda^2 \\
& \ddots & \ddots \\
& & -\lambda^2 & 1 + \lambda^2
\end{pmatrix}
\]
The accelerated solver will extract the Cholesky decomposition of any matrix of the family $A_\lambda$ from the initial decomposition of $A(\emptyset) = CC^t$. Since any matrix of the family $A_\lambda$, of size $m$, will have the same coefficients as $A(\emptyset)$ up to the $m-1$-th row, the values of its Cholesky decomposition remains the same as $C$, and only the elements of the last row need some adjustments. This will reduce the resolution of each system to a simple linear forward/backward substitution algorithm.

**Proposition 2.** let $M = A(m)$ and $N = A(n)$ denote two matrices from $A_\lambda$ such as $m < n$. Given the Cholesky decomposition of $N = CC^t$, we could extract the Cholesky decomposition of $M = BB^t$ from $C$.

**Proof.** The coefficients of the matrices $M$ and $N$ are the same for all the rows and columns strictly inferior to $m$. So we rewrite $M$ using the following decomposition by block:

\[
M = \begin{pmatrix}
M_{[1,m-1]} & 0 \\
\vdots & \ddots \\
0 & \ldots & -\lambda^2 & 1 + \lambda^2 \\
\end{pmatrix} = \begin{pmatrix}
B_{[1,m-1]} & 0 \\
\vdots & \ddots \\
0 & \ldots & b_{m-1,m} & b_{mm} \\
\end{pmatrix} \begin{pmatrix}
0 \\
\vdots \\
0 \\
\end{pmatrix}
\]

(21)

which leads to:

\[
\begin{cases}
M_{[1,m-1]} = B_{[1,m-1]}B_{[1,m-1]}^t \\
-\lambda^2 = b_{m-1,m-1}b_{m,m-1} \\
1 + \lambda^2 = b_{m,m-1}^2 + b_{m,m}^2
\end{cases}
\]

Also, we rewrite $N$ as a block matrix:

\[
N = \begin{pmatrix}
N_{[1,m-1]} & N_{12} \\
N_{21} & N_{22}
\end{pmatrix} = \begin{pmatrix}
C_{[1,m-1]} & C_{12} \\
C_{21} & C_{22}
\end{pmatrix} \begin{pmatrix}
C_{[1,m-1]}^t & C_{12}^t \\
C_{21}^t & C_{22}^t
\end{pmatrix}
\]

This will implies that $N_{[1,m-1]} = C_{[1,m-1]}C_{[1,m-1]}^t$. And since $N_{[1,m]}$ and $M_{[1,m]}$ are equal, we could compute $B$ as follows:

\[
B_{[1,m-1]} = C_{[1,m-1]}
\]

(22)

\[
b_{m,m-1} = -\frac{\lambda^2}{b_{m-1,m-1}}
\]

(23)

\[
b_{m,m} = \sqrt{1 + \lambda^2 - b_{m,m-1}^2}
\]

(24)

The equation in (22) is derived from the uniqueness of the Cholesky factorization since the matrices $M$ and $N$ have positive diagonal elements. Also, the division in (23) is always possible because $C_{m-1,m-1}$ is not null otherwise it would imply that $N_{m-1,m-2}$ is null, or $N_{m-1,m-2} = -\lambda^2$. 
The proof shows the possibility of extracting the Cholesky coefficients of any matrix in $A_\lambda$ from the decomposition of the initial matrix $C$. The accelerated solver will only compute $C$ and uses (23,24) to deduce the remaining decompositions.

3.3. Complexity comparison. In this section, we study the complexity of DPS and its alternative implementation to illustrate the reduction gained by RDPS. So, we consider a signal $y$ of size $n$ with $p$ discontinuities. We denote $T(n,p)$ the number of basic operations to restore this signal, and we compute $T(n,p)$ for each implementation.

Firstly, the classical implementation of DPS will search all the configurations on the first $p+1$ levels. For each level $L_k$, it will solve $n-k$ linear system of size $n$ except for the initial level $L_0$, where only one system is needed. The solver must compute the Cholesky decomposition and perform a two-pass forward/backward substitution. So each system will take $4n$ division, $4(n-1)$ multiplication and $4(n-1)$ substitution or addition.

Thus, the number of basic operations needed by DPS is given by:

$$T(n,p) = 12n + \sum_{k=1}^{p+1} 12(n-p)n \approx 12(p+2)n^2 = O(n^2)$$

Secondly, for RDPS, all the work is realized inside the test (algorithm 2) to check the existence of a discontinuity in a part of the signal $y_{[i,j]}$ with size $m$. In order to confirm this, the test must solve the problem $R(y_{[i,j]})$ which considers all the nodes of the set $L_{01}^m$. Thus, the test must solve $m+1$ linear system of size $m$, but with the use of the accelerated solver, each system will only consume $6n$ operations instead of $12n$.

Then, we denote $S(n)$ the number of operations to solve the problem $R(y_{[1,n]})$:

$$S(n) = 6n + \sum_{i=1}^{n-1} (6i + 6(n-i)) = 6n^2 = O(n^2).$$

Given the cost of each problem ($R_y$), we could write $T(n,p)$ with a recursive formula based on the reconstruction scheme dictated by the decomposition binary tree $(R,E)$.

$$\begin{align*}
T(n,p) &= S(n) + T(n_1,p_1) + T(n_2,p_2) \\
n &= n_1 + n_2 \\
p &= p_1 + p_2 - 1
\end{align*}$$

The overall expression of $T(n,p)$ can’t be developed any further since it depends on the position of the discontinuities which is a random variable. On the other hand, we could consider the best case for RDPS that consist of a signal with uniformly distributed discontinuities where the important jumps reside in the middle of each block. In this case, we could develop the expression of $T(n,p)$ and we get the following explicit formula:

$$\begin{align*}
T(n,p) &= S(n) + 2T(n, \frac{p-1}{2}) \\
T(n,0) &= S(n)
\end{align*}$$
In this case we get:

\[
T(n,p) = 6 \sum_{k=0}^{l} 2^k \left( \frac{n}{2^k} \right)^2 \quad \text{with} \quad l = \log_2(p) + 1
\]

(29)

\[
= 6n^2 \sum_{k=0}^{l} \frac{1}{2^k}
\]

(30)

\[
= 12n^2 \left( 1 - \left( \frac{1}{2} \right)^{l+1} \right)
\]

(31)

\[
= 12n^2 \left( 1 - \frac{1}{4^p} \right)
\]

(32)

This represents a considerable reduction in arithmetic operations compared to DPS. Furthermore, the reduction becomes relevant with a higher number of discontinuities.

4. Numerical results. In this section, we compare the restoration quality of the RDPS with two modern algorithms. The first one is the Energy based step-finder (EBS) [25] which is based on graph cut [8] and the second algorithm takes a variational approach and minimizes the well-known \(l_2\)-Potts functional [25]. We also illustrate the boost, achieved by the recursive implementation, on a set of synthetic signals with different sizes and number of discontinuities, the two main factors slowing DPS. The results show significant gains especially with a higher number of discontinuities.

4.1. Reconstruction of synthetic data. The algorithm RDPS was implemented in C++ and tested on an Apple machine with an i7 intel processor at 2.2 GHz. The reconstruction of a signal of size \(2 \times 10^3\) took 68ms. One of the main features of DPS is that its time is independent of the signal to noise ratio.

In the first example, we apply the RDPS algorithm to determine the steps in the trajectories of molecular motors which allows to study the motion of individual enzymes. The initial signal is a synthetic signal from [25] which was simulated based on experimental data of RNA polymerase II using published rates [11]. Each point in the data is a noisy measure of the molecular motor position in base pair (bp) unit system at given time \(t\) in seconds. The simulated data was used to assess the accuracy of an algorithm called Energy based step-finder presented in [25]. EBS applies a two-step reconstruction process to detect the steps in high bandwidth signals. The first step receives the full noisy data and tries to attenuate the noise by solving a total variation regularized least-squares problem [10]. The proposed solver for this step offers a linear complexity but the resulting signal \(x^*\) has several false discontinuities. In the second step, EBS filters those false discontinuities by compressing the signal \(x^*\) to tuples of amplitudes and their lengths. Then, EBS applies a combinatorial clustering algorithm based on graph cuts to connect the tuples between false discontinuities and only assign steps to the true ones. Figure (5) outlines the reconstruction of the simulated data by EBS and our algorithm RDPS. As can be seen from the two reconstructions, RDPS only reports a unique false discontinuity at \(t \approx 1.7\), whereas EBS detected several false steps. Also, we computed the Mean square error (MSE) of each reconstruction, RDPS has a lower (MSE = 0.012) compared to EBS error (MSE = 0.091).

In the second example, we compare DPS to a recent algorithm taking a different modelization approach [13]. The presented algorithm formulates the reconstruction
Figure 5. Restoration of a synthetic simulated data of the movement of a molecular motor. The initial data contains $10^4$ data points regularly distributed in $t \in [0, 2s]$. Left (5a) we remark that DPS exactly recovers all parts except for the hard jump at $t \approx 1.7$. In the right (5b), the restored signal by EBS algorithm with several missed jumps.

problem by a variational approach which leads to the minimization of the well known $l_2$-Potts functional:

$$P_\lambda(x) = \lambda ||\nabla(x)||_0 + ||y - x||_2$$

(33)

We use the classical example illustrated in [13] and published in (pottslab). The signal in (Figure 6a) has 256 data points $x_i \in [0, 1]$, and is corrupted by a Gaussian white noise with a standard deviation $\sigma = 0.2$. Both reconstructions detect all the discontinuities with comparable MSE errors. Indeed, DPS reports an error of $5.36 \times 10^{-04}$ (Figure 6b) compared to $5.28 \times 10^{-04}$ for the $l_2$-Potts solver (Figure 6c).

4.2. Benchmarking results. We study empirically the MSE error of the DPS approximate solution as a function of the hyper-parameters. We consider a case study signal with 4 discontinuities with a minimal jump $h_m = 1$. For each choice of $\lambda \in [0.1, 50]$ and $h \in [0.01, 1.2]$, we compute the averaged MSE over 10 realizations of noise. The results are depicted in figure (7). We could emphasis three different regions in the plot: the orange one represents the overfitted solutions with a set of false discontinuities that are either due to lower values of $h$ or higher values for $\lambda$. The yellow region corresponds to convenient values of the hyper-parameters since the model detects all the true discontinuities. Finally, the red region matches the underfitted solutions where the model fails to detect the true discontinuities due to higher values of the sensitivity $h$, typically $h > h_m$.

The DPS algorithm presents many features such as its robustness to the noise and the independence of its time to the noise standard deviation $\sigma$. This is not the case for other algorithms, which either suffer in restoration quality since they cannot differentiate the plateaus from the noise, or take more time for convergence like the GNC algorithm since the number of iterations of its iterative solver is an increasing function of $\sigma$. Figure 8 shows the robustness of DPS and GNC for restoring highly
Figure 6. Restoration quality of DPS and \( l_2 \)-Potts for a signal with size 256 and values \( x_i \in [0, 1] \) corrupted by a white noise with standard deviation \( \sigma = 0.2 \). Figure (6a) represents the initial signal, (6b) shows the reconstructed signal with DPS with no false discontinuities, but with some errors in the height of some plateaus. Finally, the figure (6c) plots the optimal solution for the \( l_2 \)-Potts solver which also detect all the discontinuities.

Figure 7. Mean Square Error as a function of \( \lambda \) and \( h \).

corrupted signals. Both algorithms minimize the same energy (4) from weak string model [1].

Also, we did compare DPS and RDPS in a set of simulated signals with different sizes and number of discontinuities. The results of the experiments are presented in (Figure 9), (10a) and (10b), and confirm the important reduction rate by RDPS.

Conclusion. In this paper, we presented a recursive and accelerated implementation of the DPS algorithm for MRF reconstruction of PWC signals corrupted with gaussian noise. The recursive approach is possible, thanks to the conditional
Figure 8. Mean square error as a function of the noise standard deviation $\sigma$. We remark the curve of DPS increases slowly compared to the other algorithms.

Figure 9. The figure plots the difference $T_b - T_r$, in seconds, between the classical implementation time $T_b$ and the recursive implementation $T_r$. The difference varies depending on two factors: the signal size $n$ and the number of discontinuities $k$. From this figure it can be seen that the reduction is important especially with a higher number of discontinuities.

independence created by the confirmation of a discontinuity. This allows a rapid diminution of the signal size, the main factor slowing DPS. Also, we presented an accelerated Cholesky solver, specially conceived for the family of matrices solved by DPS. The accelerated solver only computes a unique factorisation and deduces all the upcoming factorisations. Numerical results confirmed an important speed up by the new implementation. This speed up is more relevant with a higher number of discontinuities.

One direction for future research is to reduce the workload on the first level by using an unsupervised classification algorithm. The classifier could be applied as a
Recursive Reconstruction of PWC signals

Figure 10. The figure presents the running time in a logarithmic scale as a function of the signal size with fixed number of discontinuities $k = 15$ (a) and as a function of the number of discontinuities with fixed size $n = 10^3$ (b). As shown in the both figures, the recursive implementation offers a lower time and the deviation, from the classical DPS, becomes more compelling with higher number of discontinuities.

preprocessing step, to discard a set of the hypercube. The sweep strategy would then only be applied to the remaining parts.

REFERENCES

[1] A. Blake, Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1989), 2–12.
[2] A. J. Aguirre, C. Brennan, G. Bailey, R. Sinha, B. Feng, C. Leo, Y. Zhang, J. Zhang, J. D. Gans, N. Bardeesy, C. Cauwels, C. Cordon-Cardo, M. S. Redston, R. A. DePinho and L. Chin, High-resolution characterization of the pancreatic adenocarcinoma genome, *Proceedings of the National Academy of Sciences*, 101 (2004), 9067–9072.
[3] D. M. G. Anderson, Z. Ablonczy, Y. Koutalos, J. Spraggins, R. K. Crouch, R. M. Caprioli and K. L. Schey, High Resolution MALDI Imaging Mass Spectrometry of Retinal Tissue Lipids, *Journal of The American Society for Mass Spectrometry*, 25 (2014), 1394–1403.
[4] J. Besag, Statistical analysis of dirty pictures, *Journal of Applied Statistics*, 20 (1993), 63–87.
[5] M. J. Black and A. Rangarajan, On the unification of line processes outlier rejection, and robust statistics with applications in early vision, *International Journal of Computer Vision*, 19 (1996), 57–91.
[6] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MA, USA, 1987.
[7] C. Bouman and K. Sauer. A generalized Gaussian image model for edge-preserving MAP estimation, *IEEE Transactions on Image Processing*, 2 (1993), 296–310.
[8] Y. Boykov, O. Veksler and R. Zabih, Fast approximate energy minimization via graph cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 (2001), 1222–1239.
[9] P. Charbonnier, L. Blanc-Feraud, G. Aubert and M. Barlaud, Deterministic edge-preserving regularization in computed imaging, *IEEE Transactions on Image Processing*, 6 (1997), 298–311.
[10] L. Condat, A direct algorithm for 1-D total variation denoising, *IEEE Signal Processing Letters*, 20 (2013), 1054–1057.
[11] M. Dangkulwanich, T. Ishibashi, S. Liu, M. L. Kireeva, L. Lubkowska, M. Kashlev and C. J. Bustamante, Complete dissection of transcription elongation reveals slow translocation of rna polymerase ii in a linear ratchet mechanism, *Biophysical Journal* 2 (2014), 485a–486a.
[12] C.-A. Deledalle, S. Vaiter, G. Peyré, J. Fadili and C. Dossal, Unbiased risk estimation for sparse analysis regularization, In *Image Processing (ICIP)*, 2012 19th IEEE International Conference on, IEEE, (2012), 3053–3056.

[13] L. Demaret, M. Storath and A. Weinmann, Reconstruction of piecewise constant signals by minimization of the $l1$-potts functional, *arXiv:1207.4642* (2012).

[14] P. Djuric, J.-K. F. J.-K. Fwu, S. Jovanovic and K. Lynn, On the processing of piecewise-constant signals by hierarchical models with application to single ion channel currents, *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 5 (1996), 2762–2765.

[15] M. Douimi and H. Cherifi, Cutting enumerative algorithm for the minimizing of energy function, *GRETSI, Trait. Signal*, 15 (1998), 67–78.

[16] C. G. Farquharson, Constructing piecewise-constant models in multidimensional minimum-structure inversions, *Geophysics*, 73 (2007), K1–K9.

[17] D. Geiger and F. Girosi, Parallel and deterministic algorithms from mrf:s: Surface reconstruction and integration, *Computer Vision–ECCV*, 90 (1990), 89–98.

[18] S. Geman and D. Geman, Stochastic relaxation, gibbs distributions and the bayesian restoration of images, *IEEE Trans. on PAMI*, 6 (1984), 721–741.

[19] R. HORST and P. M. Pardalos, *Handbook of Global Optimization*, vol. 2 of Nonconvex optimization and its applications, Kluwer Academic Publishers, Dordrecht; Boston, 1995.

[20] B. Jackson, B. Stevens and G. Hurlbert, *Research problems on Gray codes and universal cycles*, *Discrete Mathematics*, 309 (2009), 5341–5348.

[21] M. A. Little, N. S. Jones and N. S. Jones, Generalized methods and solvers for noise removal from piecewise constant signals. II. New methods, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467 (2011), 3115–3140.

[22] M. Nikolova, Minimization of cost-functions with non-smooth data-fidelity terms to clean impulsive noise, In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, 391–406.

[23] E. Pappalardo, B. A. Ozkok and F. M. Pardalos, Combinatorial optimization algorithms, In *Handbook of Combinatorial Optimization*, Springer New York, 2013, 559–593.

[24] R. Ramlau and W. Ring, A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data, *Journal of Computational Physics*, 221 (2007), 539–557.

[25] J. Rosskopf, K. Paul-Yuan, M. B. Plenio and J. Michaelis, Energy-based scheme for reconstruction of piecewise constant signals observed in the movement of molecular machines, *Physical Review E*, 94 (2016).

[26] A. Weinmann, M. Storath and L. Demaret, The $L^1$-potts functional for robust jump-sparse reconstruction, *SIAM Journal on Numerical Analysis*, 53 (2015), 644–673.

[27] G. Winkler, O. Wittich, V. Liesbcher and A. Kempe, Don’t shed tears over breaks, *Jahresber. Deutsch. Math.-Verein.*, 107 (2005), 57–87.

Received May 2017; 1st revision October 2017; 2nd revision January 2018.

*E-mail address*: a.belcaid@edu.umi.ac.ma
*E-mail address*: m.douimi@umi.ac.ma
*E-mail address*: a.fassifihri@ensam.umi.ac.ma