Stochastic dichotomic models for digital video sequences

Cécile COT

LMENS - 99 - 5
Stochastic dichotomic models for digital video sequences

Cécile COT

LMENS - 99 - 5

January 1999
Stochastic dichotomic models for digital video sequences

Abstract. The aim of this paper is to introduce a method for restoring a sequence of time varying images. Our approach is based on the description of images by contour lines in a Bayesian context. The use of contour lines for image restoration has already been discussed in Catoni [9]. Here we adapt the methodology to our problem and consider the contour surfaces described by contour lines when the time dimension is added. We use a generalization of the noisy Ising model on the level sets representation of grey-level sequences of images. In order to allow boundary smoothing, we choose appropriately the connectivity of the model, so that we can make the Hamiltonian proportional to a satisfactory approximation of the Euclidean length of the boundaries. The noise model reflects the distance between the observation and the reconstructed sequence. We use a Gaussian noise model translated back to the level sets representation, to take into account the pixel contrast. We propose a generalization of the dichotomic Markov field model introduced in [9].

Bayesian methods of restoration are numerically intensive. We discuss the advantages of the contour lines dichotomic conditioning restoration compared to the usual global conditioning method of restoration. In order to speed-up the restoration, we implement the hierarchical multi-resolution algorithm ([13]) for the dichotomic model of reconstruction of contour lines. A range of practical examples illustrate this approach.

1. Introduction.

Restoration of images brings an important enhancement of visual information and is used in a wide variety of applications: It is often associated with segmentation or serves as pre-processing for motion analysis (to help to determine the images instantaneous characteristics).

The problem addressed here is the restoration of sequences of time varying images, by noise elimination and motion shaking regularization. The sequences under-question are (or supposed to be) corrupted with an i.i.d. additive Gaussian noise $N_3(0, \sigma^2)$ in the spatial and temporal direction.

Our approach is based on the contour lines description of images, in a Bayesian statistical framework (following a previous work of Catoni ([9]) for fixed images). Let $y$ denote an observed sequence of $T$ time-varying images discretized on a finite lattice $S \subset \mathbb{Z}^2$. The pixels will take values in a finite set of grey levels of the form $\Lambda = \{0, \ldots, 2^n - 1\}$. One usually considers that $y$ belongs to the probability space $E = \Lambda^{S \times \{1, \ldots, T\}}$. Under the contour lines model, we choose to modelize $y$ by its level
sets $L(y) = (L_\lambda(y))_{\lambda \in \Lambda}$ where

$$L_\lambda(y) = \{(s,t) \in S \times \{1, \ldots, T\}, y_{s,t} \geq \lambda\}, \ \forall \lambda \in \Lambda \setminus \{0\},$$

so that we would rather embed $E$ into $\Omega = (\mathcal{P}(S \times \{1, \ldots, T\}))^{2^n-1}$.

Under the usual global conditioning model, one takes into account all the inclusion relations between level sets (it is hidden in the construction), so that one considers the reduced probability space $\tilde{\Omega}$ such that any level set satisfies the inclusion relations, i.e. such that

$$\lambda_2 \leq \lambda_1 \Rightarrow L_{\lambda_1} \subset L_{\lambda_2}.$$ 

Note that one can identify $E$ with $\tilde{\Omega}$ in a canonical way.

We need to define a posterior distribution on the product space $\Omega$. To stick to the Bayesian theory, we choose a posterior distribution on $\Omega$ of the Gibbs form

$$\forall L \in \Omega, P(L \mid L(y)) = \frac{1}{Z_{\tau,y}} \exp \left( -\frac{1}{\tau} (H(L) + N(L, L(y))) \right).$$

Then we choose $L$ according to the posterior maximum likelihood estimator

$$\hat{L} = \arg \min_{L \in \Omega} H(L) + N(L, L(y)).$$

The estimator is computed using stochastic optimization methods.

The energy model is appropriate to our context and to the nature of the noise. Restricted to $\tilde{\Omega}$, this is a spatio-temporal energy function defined on the contour lines of sequences. The prior term $H(L)$ plays the role of a selective "smoothing" function, which controls the surface of the boundaries. We will use a generalization of the noisy Ising model on the level sets representation of images. The goodness-of-fit term $N(L, L(y))$ plays the role of a "scale" function, which fixes the minimal size of the kept details. It measures the distance between the observation and the reconstructed sequence. It has the dimension of a volume. Under the usual model, in order to deal with i.i.d. Gaussian noise elimination, one usually uses the $\chi^2$ statistics of the noise

$$\forall x, y \in E, \tilde{N}(x, y) = \beta \sum_{t=1}^{T} \sum_{s \in S} (x_{s,t} - y_{s,t})^2,$$

where $\beta > 0$ is a real noise parameter.

We will use an equivalent function on the description of images by their level sets. This function will enable to take into account the pixel contrasts, so that significant boundaries should be kept.

In the case of the reconstruction of piecewise constant images, a Bayesian energy function usually contains two terms, one of the dimension of a perimeter and one of the dimension of an area. It is natural and widely used. The justification may be found in Mumford and Shah ([25] [24]). The term that has the dimension of an area measures the variance of the noise. It is an integral term with respect to Lebesgue
measure on $S$. The term that has the dimension of a perimeter is an integral term with respect to the Hausdorff measure of the set of boundaries, that is the "length" along the boundaries. For instance, the natural model of energy used for a Bayesian image restoration-segmentation would be

$$U(x, K) = \beta \int_{S \setminus K} (x - y)^2 ds + l(K),$$

where $K$ denotes the boundary set of segmentation, $l(K)$ the Hausdorff measure of $K$, and $x$ is constant on each connected component of $S \setminus K$.

This kind of functional is also proposed in other approaches for approximating $y$ by "smoothing" (see Cohen, DeVore, Petrushev and Hu [14]). A justification that $BV(Q)$ (the set of functions of bounded variation on $Q$) is an appropriate approximation space can be found in [2]. Consider the deterministic gradient descent algorithm $\frac{du}{dt} = -\nabla E(u) = -\int |Du(t, x)| dx$, where $u(t, x)$ is the smoothed version of $y$ depending on the "scale parameter" $t$, and $Du$ is its gradient. The evolution has the desirable property to be invariant by monotonous transformations of level sets. Indeed, each contour line follows a "mean curvature motion" independently of the other contour lines. In particular, the connected components of the level sets remain connected.

In the non-linear approximation theory, the observation $y$ is supposed to belong to $L^2(Q)$, for $Q = [0, 1]^d$. The technique consists in finding minimizers or "near minimizers" on appropriate spaces like $BV(Q)$ for the function

$$K(y, \theta) = \inf_{x \in BV(Q)} \left\{ ||y - x||^2_{L^2(Q)} + \theta |x|_{BV(Q)} \right\},$$

where $\theta > 0$ is a real fixed parameter.

A "near minimizer" is a function $x_\theta$ such that

$$||y - x_\theta||^2_{L^2(Q)} + \theta |x_\theta|_{BV(Q)} \leq C K(y, \theta),$$

for some constant $C > 0$ independent of $\theta$.

For $\theta = \frac{1}{N^d}$, $N \geq 1$, constructions of a "near minimizer" are proposed on certain approximation spaces $\Sigma_N$, whose elements are linear combinations of at most $N$ piecewise constant functions. In the case of a wavelet thresholding, the approximation space is simply the set of all linear combinations $\sum c(y) \lambda$, with at most $N$ terms and $(\lambda_\lambda)_{\lambda \in \Lambda}$ is the Haar system of $L^2(Q)$. Other approximation spaces are proposed such as the set of all linear combinations of at most $N$ characteristic functions of dyadic "rings" that form a partition of $Q$. In those last two cases, a "near minimizer" is built by a quadtree algorithm, that builds an adaptative partition of $Q$ into dyadic cubes or "rings" by successive splittings.

Our method proposes another way to approximate $BV$ by discretizing the space.
We implement the dichotomic conditioning model of reconstruction of contour lines, described in [9], and discuss its advantages compared to the classical global conditioning model.

The dichotomic conditioning model consists in grouping the restoration of several level sets

\[ \mathcal{L}_k = \left\{ I_\lambda, \forall \lambda \in 2^{n-k-1} \mathbb{N} \cap \lambda \right\}, \text{ for } k = 0 \text{ to } k = n - 1 \]

taking into account the inclusion relation, instead of treating them separately.

Consider the projection maps \( \Pi_k = \Pi_{\mathcal{L}_k} \) of \( \Omega \) on \( \mathcal{L}_k \), and consider the subsets \( \tilde{\Omega}_k = \Pi_k^{-1} \circ \Pi_k(\tilde{\Omega}) \). It is easy to check that

\[ \tilde{\Omega} = \tilde{\Omega}_{n-1} \subset \tilde{\Omega}_{n-2} \subset \ldots \subset \tilde{\Omega}_0 = \Omega. \]

The dichotomic conditioning model consists in building the Markov chain \((\mathcal{L}_0, \ldots, \mathcal{L}_{n-1})\) on \( \Omega \) with initial distribution \( P(\mathcal{L}_0 \mid I(y)) \) and with \( k^{th} \) transition

\[ P(\mathcal{L}_k \mid I(y), \mathcal{L}_{k-1}, \tilde{\Omega}_k). \]

The usual model consists in simulating directly the distribution \( P(\mathcal{L}_{n-1} \mid I(y), \tilde{\Omega}) \).

Speed-up techniques for Monte-Carlo algorithms are an important subject of investigation ([3],[5],[4],[10],[11],[13],[26],[27]). The techniques rely on parallelization, localization of algorithms and reduction of their state space.

We propose here a "real world" illustration of the stochastic optimization algorithms studied in [13]. We will compare two speed-up candidates to the standard Metropolis algorithm implemented for the dichotomic model of reconstruction of contour lines.

The first one is the multi-resolution algorithm, which uses independent multiple local Metropolis trials. As it is the case in a product state space with independent components, localization of the stochastic optimization method on windows provides acceleration.

The second candidate to acceleration is the multi-resolution algorithm with hierarchical trials. It uses two scales of resolution to describe an image. A hierarchical trial is a Metropolis pre-processing at coarse scale followed by a short Metropolis post-processing at fine (standard) scale. We use a specific sub-sampling operation that is adapted to the dichotomic model of reconstruction of contour lines.

The structure of the paper is as follows. In section 2, we focus on the energy model used for a sequence of time varying binary images. The generalization to grey-level movies immediately follows. Section 3 presents a generalization of the dichotomic Markov field model introduced in [9] to the restoration of a grey-level movie. In section 4, we describe the classical multi-resolution dynamics running on overlapping cubic shape windows, then the multi-resolution dynamics running on contour line partitions. Finally, we describe the multi-resolution dynamics using hierarchical trials. In section 5, we illustrate the performances of those algorithms on sequences.
of images. We discuss the contributions of the acceleration techniques and compare their performances.

This paper was part of my Ph.D. dissertation under the supervision of O. Catoni. I am pleased to acknowledge him for useful discussions, advices and careful reading. I also acknowledge NASA and the NSSDC as the supplier of the last data (a video sequence of pathfinder on Mars).

Many other images and video sequences are available via ftp://nssdc.gsfc.nasa.gov/pub/photo_gallery/.

2. Penalized Hamiltonian on a grey level movie.

2.1. Smoothing a sequence of binary images. Let us focus on a sequence of time varying binary images. This will be a technical step towards the restoration of sequences of grey level images.

We choose to modelize the observation $y : S \times \{1, \ldots, T\} \to \{-1, +1\}$ where $S \subset \mathbb{Z}^2$ is a finite lattice of sites and where $T$ is a finite number of images of the sequence, by a noisy Ising model. We will write

$$y = \left(y_{i,t}, \quad i \in S, \quad t \in \{1, \ldots, T\}, \quad y_{i,t} \in \{-1, +1\}\right).$$

We build the posterior distribution of the original sequence of binary images $X$ in the Bayesian framework. It is supposed to be

$$P(x \mid y) = \frac{1}{Z_{y,\tau}} \exp \left(-\frac{1}{\tau} \left(H(x) + N(x, y)\right)\right),$$

where $\tau$ denotes the temperature parameter and where $Z_{y,\tau}$ is a normalizing constant which is impossible to compute, and where

$$H(x) = -\sum_{i=1}^{T} \left( \sum_{(i<j) \in S^2} \xi_{ij} x_{i,t} x_{j,t} + \sum_{(i \leq j) \in S^2} \alpha_{ij} x_{i,t} x_{j,t-1} \right),$$

with $\xi_{ij}$ some spatial links between pixels of an image, and $\alpha_{ij}$ some temporal links between pixels of consecutive images, and where

$$N(x, y) = -\beta \sum_{i=1}^{T} \sum_{i \in S} x_{i,t} y_{i,t},$$

with $\beta > 0$ some noise coefficient.

The goodness-of-fit $N(x, y)$ measures the distortion between the observation $y$ and the original sequence $x$. It is proportional to the 3 dimensional area delimited by the pixels of $x$ and $y$ with different signs $\{(i, t) \in S \times \{1, \ldots, T\}, x_{i,t} \neq y_{i,t}\}$. 
To restore the observation $y$, one may choose the MAP estimator 
(Maximum A Posteriori)

$$\hat{x} = \arg \max_{x \in E} P(x \mid y) = \arg \min_{x \in E} H(x) + N(x, y),$$

or the MPM estimator (Maximum a Posteriori Marginals)

$$\hat{x}_{i,t} = \arg \max_{x_{i,t} \in \{-1, +1\}} P_{i,t}(x_{i,t} \mid y).$$

One usually simulates the posterior distribution or its marginals with Monte Carlo 
optimization algorithms until equilibrium (their invariant distribution being loga-
rithmically equivalent at large $\tau$ to $P(\cdot \mid y)$). The aim is to smooth boundaries by 
minimizing the Hamiltonian under the constraint of the goodness-of-fit term.
The temperature parameter $\tau$ determines what neighbourhood of the minima of the 
posterior energy $H(x) + N(x, y)$ will be explored during the Monte Carlo simulation.

By choosing appropriately the spatial and temporal links $\xi_{i,j}$ and $\alpha_{i,j}$, we can make 
the Hamiltonian be proportional to a good discrete approximation of the Euclidean 
length of the boundary of 3 dimensional blobs. The noise coefficient $\beta$ also takes a 
special interest as it represents the relative weight between the goodness-of-fit and 
the Hamiltonian. It may be chosen such that small blobs shrink at low temperature.

We choose

$$\xi_{i,j} = \begin{cases} 
1 & \text{if } |i - j| = 1 \\
\frac{1}{\sqrt{2}} & \text{if } |i - j| = \sqrt{2} \\
0 & \text{otherwise,}
\end{cases}$$

so that for any $x = (x_i)_{i \in S} \in \{-1; +1\}^S$ the quantity $\sum_{i<j} \xi_{i,j} x_i x_j$ is equal (up to an 
additive constant) to a discrete approximation of the Euclidean boundary length of 
the 2 dimensional blob corresponding to $x$ (i.e. the blob of positive pixels identified 
to the set $\{i \in S, x_i = +1\}$).
The constant $\xi_{i,j} = \frac{1}{\sqrt{2}}$ for $|i - j| = \sqrt{2}$ is chosen so that the computation of the 
length is exact for edges with orientations $0, \frac{\pi}{4}, \frac{\pi}{2}$ and $\frac{3\pi}{4}$.
For instance, let $x = (x_i)_{i \in S} \in \{-1; +1\}^S$ be some 2 dimensional square blob of size 
a in an uniform background, and let $R_{\frac{\pi}{4}} x$ be the discretized square blob version of 
x rotated with a $\frac{\pi}{4}$ angle. By computing the number of pairs $(i < j) \in S^2$ with
\(|i - j| = 1\) or \(|i - j| = \sqrt{2}\) with opposite signed pixels appearing in \(R_{\frac{a}{2}}x\), we have

\[
- \sum_{(i < j) \in S^2, |i - j| = 1} R_{\frac{a}{2}}x_i R_{\frac{a}{2}}x_j - \xi \sum_{(i < j) \in S^2, |i - j| = \sqrt{2}} R_{\frac{a}{2}}x_i R_{\frac{a}{2}}x_j = 2 \left(4\left[\frac{a}{\sqrt{2}}\right](2 + \xi) + 4\left[\frac{a}{\sqrt{2}}\right] - 1\right)\xi - 4\right) - C
\]

with \(C = \frac{1}{2}\{(i < j) \in S^2, |i - j| = 1\} + \frac{1}{2}\{(i < j) \in S^2, |i - j| = \sqrt{2}\} \). On the other hand,

\[
- \sum_{(i < j) \in S^2, |i - j| = 1} x_i x_j - \xi \sum_{(i < j) \in S^2, |i - j| = \sqrt{2}} x_i x_j = 2 \left(4a(1 + 2\xi) - 4\xi\right) - C.
\]

The only possibility that equality holds \(\forall \alpha > 0\) is to tune \(\xi = \frac{1}{\sqrt{2}}\).

The noise coefficient \(\beta > 0\) is tuned so that steady spatial blobs of diameter lower than \(d_0\) easily shrink at low temperature, for some fixed parameter \(d_0\). Following this idea, let \(y\) be some 2-dimensional \(d_0 \times d_0\) square blob over a uniform background, and let \(x\) be the background image. We choose \(\beta\) such that

\[
- \sum_{(i < j) \in S^2} \xi_{i,j} x_i x_j - \beta \sum_{i \in S} x_i y_i \leq - \sum_{(i < j) \in S^2} \xi_{i,j} y_i y_j - \beta \sum_{i \in S} y_i^2.
\]

Computing the number of pairs \((i < j) \in S^2\) with \(|i - j| = 1\) or \(|i - j| = \sqrt{2}\) with opposite signed pixels appearing in this expression, this is equivalent to

\[
\beta \leq \frac{4(1 + \sqrt{2})}{d_0}.
\]

Thus, we choose

\[
\beta = \frac{4(1 + \sqrt{2})}{d_0}.
\]

The temporal links \(\alpha_{i,j}\) are adjusted in order that any spatial blob standing in a few number of images of the sequence tends to vanish. To simplify the problem, let us start with

\[
\alpha_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ \alpha & \text{if } i = j, \end{cases}
\]

for some parameter \(\alpha > 0\).

Let us consider the restoration of static sequences. Let \(b_c\) denotes the critical time under which spatial blobs should disappear. Let \(\Delta t\) denotes the time interval between two consecutive images. Thus, shapes that last the critical time appear in \(\frac{b_c}{\Delta t}\) consecutive images.
Let $y$ be some static sequence with a $a \times a$ square blob over a uniform background appearing in less than $\frac{b_c}{\Delta t}$ consecutive images, and let $x$ be the background sequence. We choose $\alpha$ such that

$$H(x) + N(x, y) \leq H(y) + N(y, y), \quad \forall a > 0$$

or equivalently, such that

$$\beta a^2 \frac{b_c}{\Delta t} \leq 4a \frac{b_c}{\Delta t}(1 + \sqrt{2}) + 2a^2 \alpha, \quad \forall a > 0$$

or,

$$\alpha \geq \frac{\beta b_c}{2 \Delta t} - \frac{2 b_c}{a \Delta t}(1 + \sqrt{2}), \quad \forall a > 0.$$ 

Thus, we choose

$$\alpha = \frac{\beta b_c}{2 \Delta t}.$$ 

**Remark 2.1.** The temporal link is a decreasing function of $\Delta t$, as it should be.

Let us generalize this energy model to grey level movies.

### 2.2. Smoothing a sequence of grey level images.

For a $2^n$ grey level movie ($n = 8$ if one consider 256 grey levels), let $\Lambda = \{0, \ldots, 2^n - 1\}$ denote the set of grey levels and modelize the observation by $y : S \times \{1, \ldots, T\} \to \Lambda$. Let us write

$$y = (y_{i,t}, \quad i \in S, \quad t \in \{1, \ldots, T\}, \quad y_{i,t} \in \Lambda).$$

We will also represent it by its level sets $L(y) = (L_\lambda(y))_{\lambda \in \Lambda \setminus \{0\}}$, where

$$L_\lambda(y) = \{(i, t) \in S \times \{1, \ldots, T\}, \quad y_{i,t} \geq \lambda\}, \quad \forall \lambda \in \Lambda \setminus \{0\}.$$

We will identify $y$ to $L(y)$, therefore it may be considered that $y$ takes its values in the set $\bar{\Omega} = \left(\mathcal{P}(S \times \{1, \ldots, T\})\right)^{2^n - 1}$. Our approach will be to define first a model on the product space $\Omega$, then to restrict it to $\bar{\Omega} = \{(L_\lambda)_{\lambda \in \Lambda}, \quad L_{\lambda+1} \subset L_\lambda\}$ where all the inclusion relations between level sets are taken into account. It enables to define the global conditioning model (the usual model). Then we will define the dichotomic conditioning model.

Note that we can identify $\bar{\Omega}$ with $E = \Lambda^{S \times \{1, \ldots, T\}}$ up to an isomorphism. Thus, for all $L \in \bar{\Omega}$, $L$ is the set of grey levels of a unique sequence of images $x \in E$. We will often identify $L = L(x)$ with $x$.

Let us define a product probability measure $P$ on the product space $\Omega$: let $\chi$ denote
the characteristic function with values in \{-1, +1\}. For any \( L = (L_\lambda)_{\lambda} \in \Omega \) let

\[
H(L) = -\sum_{\lambda=1}^{2^n-1} \sum_{i=1}^{T} \left( \sum_{(i<j) \in S^2} \xi_{ij} \chi(L_\lambda)_{i,t} \chi(L_\lambda)_{j,t} \right) + \alpha \sum_{i \in S} \chi(L_\lambda)_{i,t} \chi(L_\lambda)_{i,t-1},
\]

where the spatial links are

\[
\xi_{i,j} = \begin{cases} 
  1 & \text{if } |i - j| = 1 \\
  \frac{1}{\sqrt{2}} & \text{if } |i - j| = \sqrt{2} \\
  0 & \text{otherwise},
\end{cases}
\]

and where the temporal link \( \lambda = \frac{1}{\sqrt{2}} \) will be tuned below.

We define the posterior distribution on \( \Omega \) as

\[
P(L \mid y) = \frac{1}{Z_{y,\tau}} \exp \left( -\frac{1}{\tau} \left( H(L) + N(L, y) \right) \right),
\]

where the noise model will be discuss afterwards.

Under the usual global conditioning model, we consider \( Q_{\text{global}} = P(\cdot \mid y, \hat{\Omega}) \) defined on \( \hat{\Omega} \).

**Remark 2.2.** Observe that for any \( x \in E \),

\[
(1) \quad \sum_{\lambda=1}^{2^n-1} \chi(x_{i,t} \geq \lambda) \chi(y_{i,t} \geq \lambda) = 2^n - 1 - 2|x_{i,t} - y_{i,t}|.
\]

**Remark 2.3.** Consider the goodness-of-fit function defined on \( \hat{\Omega} \times \hat{\Omega} \) by

\[
\tilde{N}(x, y) = -\beta \sum_{\lambda=1}^{2^n-1} \sum_{t=1}^{T} \sum_{i \in S} \chi(x_{i,t} \geq \lambda) \chi(y_{i,t} \geq \lambda),
\]

for some noise coefficient \( \beta > 0 \).

Then, thanks to (3.1), it is equal up to an additive constant to the goodness-of-fit defined on \( E \times E \) by

\[
\tilde{N}(x, y) = 2\beta \sum_{t=1}^{T} \sum_{i \in S} |x_{i,t} - y_{i,t}|.
\]

This is clearly not a \( \chi^2 \) statistic for the noise. That is why, in the following, we will modify the role of \( \beta \), making \( \beta \) depend on \( \lambda, i, t \) and \( y \).

Let

\[
\beta(\lambda, y, i, t) = \beta \left( |\lambda - y_{i,t}| \right) - \frac{1}{2} \chi(y_{i,t} \geq \lambda) \).
\]
for some noise coefficient $\beta > 0$. We will consider in the following the goodness-of-fit function defined on $\Omega \times \tilde{\Omega}$ by

$$N(L, L(y)) = -\sum_{\lambda=1}^{2^n-1} \sum_{t=1}^T \sum_{i \in S} \beta(\lambda, y, i, t) \chi(L_{\lambda}), t \chi(L_{\lambda}(y)), t.$$ 

**Remark 2.4.** The term $|\lambda - y_{i, t}|$ appearing in $\beta(\lambda, y, i, t)$ is a contrast detector term, which allows us to decide whether a detail is sharp enough to be kept.

**Remark 2.5.** Observe that for any $L \in \tilde{\Omega}$, there exists a unique $x \in E$, such that $L = L(x)$ (that is $\forall \lambda, L_{\lambda} = \{(i, t) \in S \times \{1, \ldots, T\} : x_{i, t} \geq \lambda\}$) and

$$\sum_{\lambda=1}^{2^n-1} \beta(\lambda, y, i, t) \chi(L_{\lambda}), t \chi(L_{\lambda}(y)), t$$

$$= -\beta(x_{i, t} - y_{i, t})^2 + \frac{\beta}{2} (y_{i, t} - 1)^2 + \frac{\beta}{2} (2^n - y_{i, t})^2 + \beta.$$

Thanks to (3.2), the goodness-of-fit function $N$ restricted to $\tilde{\Omega} \times \tilde{\Omega}$ is equal up to an additive constant to the i.i.d. Gaussian noise model on $E \times E$

$$\tilde{N}(x, y) = \beta \sum_{t=1}^T \sum_{i \in S} (x_{i, t} - y_{i, t})^2.$$

This good-of-fit function is widely used and corresponds to an i.i.d. Gaussian noise. Furthermore, it deals well with sharp discontinuities (that is thin patterns floating on a contrasted background). The contrast detector term serves to find the significant boundaries.

The noise coefficient $\beta$ is adjusted in order that any spatial square blob of diameter inferior to $d_0$ and of maximal grey level variation $\lambda_0$ tends to vanish at low temperature.

Keeping the notations of the previous paragraph, let $y$ be a uniform 2 dimensional square blob of diameter $d_0$ over a uniform background with grey level variation $\lambda_0$, and let $x$ be the background image. We choose $\beta$ so that,

$$-\sum_{\lambda=1}^{2^n-1} \sum_{(i < j) \in S^2} \xi_{ij} \chi(x_i \geq \lambda) \chi(x_j \geq \lambda) + \sum_{i \in S} (x_i - y_i)^2$$

$$\leq -\sum_{\lambda=1}^{2^n-1} \sum_{(i < j) \in S^2} \xi_{ij} \chi(y_i \geq \lambda) \chi(y_j \geq \lambda).$$
Or equivalently, using (3.1), so that

\[ \beta \sum_{i \in \mathcal{S}} (x_i - y_i)^2 \leq \sum_{(i<j) \in S^2} \xi_{i,j} |y_i - y_j|. \]

Thus, we choose

\[ \beta = \frac{4(1 + \sqrt{2})}{d_0 \lambda_0}. \]

We choose the temporal link \( \alpha \), in order to ensure appropriate interactions between images. The argument is that any shape with grey variation larger than \( \lambda_{\text{min}} \) standing more than the critical time \( b_c \) should be preserved at low temperature. Let for instance \( y \) be a sequence with a spatial uniform \( a \times a \) square blob over a uniform background appearing in \( \frac{b_c}{\Delta t} \) consecutive images, and let \( x \) be the background sequence, if \( \lambda \) is the grey variation in \( y \), we choose \( \alpha \) such that

\[ H(x) + N(x, y) \geq H(y) + N(y, y), \quad \text{for } a \text{ large enough}, \forall \lambda > \lambda_{\text{min}}. \]

Using (1), this inequality becomes

\[ \beta \sum_{i=1}^{T} \sum_{i \in \mathcal{S}} (x_{i,t} - y_{i,t})^2 \geq \sum_{i=1}^{T} \left( \sum_{(i<j) \in S^2} \xi_{i,j} |y_{i,t} - y_{j,t}| + \alpha \sum_{i \in \mathcal{S}} |y_{i,t} - y_{i,t-1}| \right). \]

or equivalently

\[ \beta \frac{b_c}{\Delta t} a^2 \lambda^2 \geq 4a(1 + \sqrt{2}) \frac{b_c}{\Delta t} \lambda + \alpha 2a^2 \lambda, \quad \text{for } a \text{ large enough}, \forall \lambda > \lambda_{\text{min}}. \]

Thus we choose

\[ \alpha = \frac{\beta}{2} \lambda_{\text{min}} \frac{b_c}{\Delta t}. \]
Finally, we find a simple and useful expression for the posterior probability restricted to $\Omega = E$,

$$
P(x \mid y, \tilde{\Omega}) = \frac{1}{Z_{y,\tau}} \exp \left\{ -\frac{1}{\tau} \sum_{t=1}^{T} \left( \sum_{(i < j) \in S_0, |i - j| = 1} |x_{i,t} - x_{j,t}| \right. 
+ \sum_{(i < j) \in S_0, |i - j| = \sqrt{2}} \frac{1}{\sqrt{2}} |x_{i,t} - x_{j,t}| 
+ \frac{2 \lambda_{\text{min}} (1 + \sqrt{2}) b_c}{d_0 \lambda_0} \sum_{i \in S} |x_{i,t} - x_{i,t-1}| 
+ \frac{4 (1 + \sqrt{2})}{d_0 \lambda_0} \sum_{i \in S} (y_{i,t} - x_{i,t})^2 \right) \right\},
$$

where we recall that $d_0$ is the diameter and $\lambda_0$ is the grey variation of the critical spatial blob, and where $\frac{\Delta t}{\Delta t}$ denotes the critical number of images after which any shape of grey variation $\lambda_{\text{min}}$ is preserved at low temperature. Those parameters must be tuned within the application ($d_0$ should be tuned very thin when there are sharp noisy peaks in the movie).

The use of contour lines is totally hidden in the expression of the posterior probability.

3. **Dichotomic mono-resolution restoration of contour lines.**

The dichotomic restoration process is slightly different. In this second approach, as we restore step by step larger and larger sets of level sets, we are taking into account a partial inclusion relation between the level sets that have already been restored and those that are being restored. But we do not take into account the inclusion relation concerning the level sets that will be restored after. The difference between the usual global restoration process and the contour lines dichotomic one only stands in whether the inclusion relation (between level sets) is used. In practice, it sticks to the binary representation of numbers and almost reduces the complexity of the restoration of a grey level picture to the complexity of restoration of a binary picture.

The dichotomic procedure consists in grouping the restoration of several level sets

$$
\mathcal{L}_k = \{ L_{\lambda}, \forall \lambda \in 2^{n-1-k} \mathbb{N} \cap \Lambda \}
$$

(so that the resolution (in grey levels) becomes more and more refined).

We start with the restoration of the medium grey level set $\mathcal{L}_0 = \{ L_{2^{n-1}} \}$. During the second step, the level sets which correspond to the contour lines $2^{n-2}$ and
$2^{n-2} + 2^{n-1}$ are restored taking into account the inclusion relations between them and $L_{2^{n-1}}$. During the $k^{th}$ step, $2^{k-1}$ level sets are restored at the same time: those which correspond to the contour lines $2^{n-k} + l2^{n-k+1}$ for $l = 0$ to $l = 2^{k-1} - 1$. During the $n^{th}$ step, all the remaining level sets will be restored, those of $L_{n-1} \setminus \bar{L}_{n-2}$, by taking into account the inclusion relations between them and the already restored ones.

Construction of the dichotomic process: As all we are concerned about is the inclusion relations between level sets, we reduce the probability space $\Omega = \left( \mathcal{P}(S \times \{1, \ldots, T\}) \right)^{2^{n-1}}$ to the subset $\bar{\Omega}$ such that any level sets satisfy the inclusion relations i.e. $\Omega = \{(L_{\lambda})_{\lambda \in \Lambda}; L_{\lambda+1} \subset L_{\lambda}\}$.

We consider also $\bar{\Omega}_k$ where we impose only the inclusion relation on $L_k$,

$$\bar{\Omega}_k = \{(L_{\lambda})_{\lambda \in \Lambda} \in \Omega; L_{\lambda+2^{n-k}-k} \subset L_{\lambda}, \lambda \in L_k\}.$$  

It is easy to check that $\bar{\Omega} = \bar{\Omega}_{n-1} \subset \bar{\Omega}_{n-2} \subset \cdots \subset \bar{\Omega}_0 = \Omega$.

For any probability measure $Q$ defined on $\Omega$, the finite chain $(\bar{L}_0, \ldots, \bar{L}_{n-1})$ defined on $\Omega$ with initial distribution $Q$ and with $k^{th}$ transition $Q(L_k \mid \bar{L}_{k-1})$ is a Markov chain such that $L_{n-1} \in \bar{\Omega}$ and

$$Q(L_{n-1}) = Q(L_0)Q(L_1 \mid L_0) \cdots Q(L_{n-1} \mid L_{n-2}).$$

In order to apply this construction to our dichotomic process, we define a probability measure $Q$ that sticks to the Bayesian theory. We construct $Q$ from $P(L \mid y) = \frac{1}{Z_{y,T}} \exp \left(-\frac{1}{T} (H(L) + N(I, y))\right)$, $L \in \Omega$, putting

$$Q(L_0(x)) = Q\left(\left. x \right| 2^{n-1} \right) = P(L_{2^{n-1}}(x) \mid y)$$

and

$$Q(L_{k-1}(x) \mid L_{k-2}(x)) = Q\left(\left. x \right| 2^{n-k} \right), \quad \left. x \right| 2^{n-k+1} = P(L_{k-1}(x) \mid y, L_k(x), \bar{\Omega}_{k-1}),$$

for $k = 2$ to $k = N$.

More precisely,

$$Q\left(\left. x \right| 2^{n-1} \right) = \frac{1}{Z_{y,T}} \exp \left(\frac{1}{T} \sum_{i=1}^{T} \left\{ \sum_{i \neq j \in S^2} \xi_{i,j} \chi(x_{i,t} \geq 2^{n-1}) \chi(x_{j,t} \geq 2^{n-1}) \right. \right.$$

$$+ \left. \alpha \sum_{i \in S} \chi(x_{i,t} \geq 2^{n-1}) \chi(x_{i,t-1} \geq 2^{n-1}) \right.$$  

$$+ \left. \beta \sum_{i \in S} \left( y_{i,t} - 2^{n-1} - \frac{1}{2} \right) \chi(x_{i,t} \geq 2^{n-1}) \right)$$
and

\[
Q\left(\frac{x}{2^{n-k}} \mid \frac{x}{2^{n-k+1}}\right) = \frac{1}{Z_{y,\left[\frac{x}{2^{n-k+1}}\right],\tau}} \exp\left(\frac{1}{\tau} \sum_{\lambda \in \{2^{n-k} + 2^{n-k+1}\} \cap \Lambda} \sum_{t=1}^{T} \right. \\
\left. \left\{ \sum_{(i<j) \in S^2} \xi_{i,j} \chi(x_{i,t} \geq \lambda) \chi(x_{j,t} \geq \lambda) + \alpha \sum_{i \in S} \chi(x_{i,t} \geq \lambda) \chi(x_{i,t-1} \geq \lambda) + \beta \sum_{i \in S} \left(y_{i,t} - \frac{1}{2}\right) \chi(x_{i,t} \geq \lambda) \right\} \right),
\]

with

\[
Z_{y,\left[\frac{x}{2^{n-k+1}}\right],\tau} = \sum_{h \in I(x,k)} \exp\left(\frac{1}{\tau} \sum_{\lambda \in \{2^{n-k} + 2^{n-k+1}\} \cap \Lambda} \sum_{t=1}^{T} \right. \\
\left. \left\{ \sum_{(i<j) \in S^2} \xi_{i,j} \chi(h_{i,t} \geq \lambda) \chi(h_{j,t} \geq \lambda) + \alpha \sum_{i \in S} \chi(h_{i,t} \geq \lambda) \chi(h_{i,t-1} \geq \lambda) + \beta \sum_{i \in S} \left(y_{i,t} - \frac{1}{2}\right) \chi(h_{i,t} \geq \lambda) \right\} \right),
\]

where

\[
I(x,k) = \left\{ h \in \left(\Lambda \cap 2^{n-k} \mathbb{N}\right)^{S \times \{1, \ldots, T\}} : \left| \frac{h}{2^{n-k+1}} \right| = \left\lfloor \frac{x}{2^{n-k+1}} \right\rfloor \right\}.
\]

If we compute the expression of the conditional distribution of one site \((i, t) \in S \times \{1, \ldots, T\}\) knowing the others, we can simplify the numerator and the denominator. The summation over \(\lambda\) reduces to one term, namely

\[
\lambda = 2^{n-k} + 2^{n-k+1} \left| \frac{x_{i,t}}{2^{n-k+1}} \right|.
\]
Hence, if we let $\lambda_{i,t} = 2^{n-k} + 2^{n-k+1} \left[ \frac{x_{i,t}}{2^{n-k+1}} \right]$, 

$$Q(\left[ \frac{x_{i,t}}{2^{n-k}} \right] | y_{i,t}, \left[ \frac{x_{j,s}}{2^{n-k+1}} \right] ; \left( j, s \right) \neq (i, t) ) = \frac{1}{Z} \exp \left( \frac{1}{\tau} \chi(x_{i,t} \geq \lambda_{i,t}) \right) \left\{ \sum_{j \in S} \xi_{i,j} \chi(x_{j,t} \geq \lambda_{i,t}) + \alpha \chi(x_{i,t-1} \geq \lambda_{i,t}) + \alpha \chi(x_{i,t+1} \geq \lambda_{i,t}) + \beta \left( y_{i,t} - \lambda_{i,t} - \frac{1}{2} \right) \right\}. $$

Manipulating level sets directly is very advantageous. It leads to quicker computer programs. The dichotomic procedure requires less memory allocation because it generates a bit-flip updating dynamics. The whole computation reduces to manipulations on bits: comparisons of bits, proposal movement on bits ...

For a 256 grey levels, we need 8 bits/pixel for the observation, 8 bits/pixel for the current sequence, and 1 bit/pixel for the simulation.

This is well reflected in the following C implementation of the test $\{ x_{j,t} \geq 2^{n-k} + 2^{n-k+1} \left[ \frac{x_{j,t}}{2^{n-k+1}} \right] \}$:

```c
: unsigned char MASK1, MASK2;

MASK1 = 1 << (n - k);
MASK2 = (~ 0) << (n - k + 1);

if (x_{j,t} >= (x_{i,t} & MASK2) | MASK1) {
    ...
}
```

Testing $\{ x_{i,t} \geq 2^{n-k} + 2^{n-k+1} \left[ \frac{x_{i,t}}{2^{n-k+1}} \right] \}$, is even simpler as it is a test only on the bit number $n - k$:

```c
: if ((x_{i,t} & MASK1) != 0) {
    ...
}
```

Moreover, the tabulation of the exponentials is easy in the dichotomic model. No more than 4 values of $n_1$ are needed for $\exp(\frac{1}{\tau}n_1)$. No more than 4 values of $n_2$ are needed to tabulate $\exp(\frac{1}{\tau} \sqrt{2} n_2)$. No more than 3 values of $n_3$ are needed for $\exp(\frac{1}{\tau} \alpha n_3)$. Finally, 2 values of $n_4$ and 256 values of $n_5$ are needed to tabulate $\exp(\frac{1}{\tau} \beta n_4(n_5 - 0.5))$.

At the end of the $k$th dichotomic step, we have restored $\left[ \frac{x_{i,t}}{2^{n-k+1}} \right]$, that is the $k$th left most bits of $x_{i,t}$, for all $(i, t)$. Therefore at the beginning of the $(k + 1)$th dichotomic step, there remain to initialize the $(k + 1)$th bits that is to choose between the two
possible values $\left\lfloor \frac{n}{2^{n-k}} \right\rfloor = 2 \left\lfloor \frac{n}{2^{n-k+1}} \right\rfloor$ or $\left\lfloor \frac{n}{2^{n-k}} \right\rfloor = 2 \left\lfloor \frac{n}{2^{n-k+1}} \right\rfloor + 1$. Between those two values, we choose the one that is closest to the observation $\left\lfloor \frac{n}{2^{n-k}} \right\rfloor$. We sketch here the C programming test for updating the pixels between the $k$th and the $(k+1)$th dichotomic step:

```c
unsigned char MASK3, MASK4;

MASK3 = (~ 0) << (n - k);
MASK1 = ~ MASK3;

for ((i, t) ∈ S × {1, . . . , T}) “scanning the lattice” {
    if ((x_i, t & MASK3) < (y_i, t & MASK3)) {
        x_i,t = MASK4;
    } else {
        if ((x_i, t & MASK3) > (y_i, t & MASK3)) {
            x_i,t = MASK3;
        } else {
            x_i,t = y_i,t;
        } 
    }
}
```

4. Dichotomic multi-resolution restoration of contour lines with hierarchical trials.

4.1. Usual multi-resolution algorithmic model. The model. We perform here the contour lines dichotomic restoration in the context of a multi-resolution dynamics. We consider a collection of windows overlapping each other that covers the lattice $S \times \{1, . . . , T\} \subset \mathbb{Z}^3$. We describe the possible choices of window size and locations. The size and locations of windows can be related to the choice of the critical blob and the critical number of images $\frac{\beta_c}{\Delta t}$. By analogy with rigorous results for the noisy Ising model, we will choose a collection of overlapping windows such that any blob of size $d_0 \times d_0$ is contained in at least one window and such that the same property holds for any shape standing less than the critical time. If the ”critical” blob is inscribed in a square of size $d_0 \times d_0$, then we build the collection of windows as follows: first choose the same 2D-grid on each image of the sequence, with unit paving of size $d_0 \times d_0$. A window will be any three dimensional block of size $2d_0 \times 2d_0 \times 2\frac{\beta_c}{\Delta t}$ centered at an edge of the grids. There are $\frac{d_0 \times d_0}{\sqrt{\pi}} \times \frac{T}{\beta_c}$ of them, as much as edges. The such defined collection of windows covers the whole movie: any shape of diameter less than $d_0$ and standing less than the critical time is inscribed in at least one window. In practice, to decrease the overlap between windows, we have kept only
one window over two.

The multi-resolution dynamics. The multi-resolution dynamics uses independent multiple local optimization attempts. Let \((S_r)^R_{r=1}\) denotes the collection of optimization windows built above. Due to the construction, they overlap each other and cover the 3 dimensional finite lattice. Let \((1, \ldots, R)\) denotes the scanning order chosen for the windows. It may be a progressive scanning from top-left to down-right or a Besag like parallel scanning.

Put \(E = \bigcup_{r=1}^{R} S_r\).

Consider the dichotomic step \(k\). Let \((\lambda_{i,t})_{i,t}\) denote the set of intensity levels of configuration \(x\) to be restored during the \(k\)th step. By definition,

\[
\lambda_{i,t} = 2^{n-k} + 2^{n-k+1}\left\lfloor \frac{x_{i,t}}{2^{n-k+1}} \right\rfloor.
\]

Let \(y = (y_{i,t})_{i,t}\) denote the observed sequence.

The energy model associated with the reconstruction in \(S_r\) of the contour lines \(\lambda_i = 2^{n-k} + 12^{n-k+1}\) for \(i \in [0; 2^{k-1}]\) is

\[
U_k(x) = -\sum_{t=1}^{T} \sum_{i \in S} \chi(x_{i,t} \geq \lambda_{i,t}) \left\{ \sum_{j \in S} \xi_{i,j} \left[ \frac{1}{2} \mathbb{I}_{\lambda_j = \lambda_{i,t}} + \mathbb{I}_{\lambda_j \neq \lambda_{i,t}} \right] \chi(x_{j,t} \geq \lambda_{i,t}) + \alpha \left( \frac{1}{2} \mathbb{I}_{\lambda_{i,t-1} = \lambda_{i,t}} \right) \chi(x_{i,t-1} \geq \lambda_{i,t}) \right. \\
+ \alpha \left( \frac{1}{2} \mathbb{I}_{\lambda_{i,t+1} = \lambda_{i,t}} \right) \chi(x_{i,t+1} \geq \lambda_{i,t}) + \beta \left( y_{i,t} - \lambda_{i,t} - \frac{1}{2} \right) \left\}.
\]

Note that for any \(r\), there exists \(\tilde{U}_{k,r}\) and \(R\), such that for any \(x \in E\),

\[
U_k(x) = \tilde{U}_{k,r} \left( \left\lfloor \frac{x}{2^{n-k}} \right\rfloor_{S_r} + R \right) + R \left( \left\lfloor \frac{x}{2^{n-k}} \right\rfloor_{S_r} \right).
\]

where \(V\) is the set of the 11 first neighbours of the origin.

The multiple local attempts on \(S_r\) will be some independent Metropolis algorithms. To define them, let us take an outside window configuration \(\overline{x_0}^{S_r}\) and fix a number of attempts \(L\). Each attempt is a Metropolis dynamics running on \((\Lambda \cap 2^{n-k}\mathbb{N})^{S_r} \times \{\overline{x_0}^{S_r}\}\) with temperature \(L\tau\). The transition kernel at site \((i, t)\) is

\[
p^{i,t}_{r,L\tau}(x, x') = \begin{cases} 
q^{i,t}_{r,L\tau}(x, x') e^{-\frac{1}{L\tau} (U_k(x') - U_k(x))} & \text{if } x \neq x' \\
1 - \sum_{z \neq x, \overline{x_r} = x_0} p^{i,t}_{r,L\tau}(x, x') & \text{if } x = x',
\end{cases}
\]
for some perturbation kernel $q_{i,t}^{i,t}$ irreducible with symmetrical support. 

And the transition kernel of the Metropolis dynamics running on $\left(\mathcal{X} \cap 2^{n-k}\mathbb{N}\right)^{S_r} \times \{x_0^{S_r}\}$ is 

$$p_{r, L} = \prod_{(i, t) \in S_r} p_{r, L}^{i,t}.$$ 

This non commutative product is performed in some order defining the way window $S_r$ is scanned. 

In our applications, we choose a bit-flip dynamics. Only one pixel at a time is modified. And the modification occurs at bit number $n - k$ (for the $k$th dichotomic step). 

The perturbation kernel at site $(i, t)$ satisfies therefore 

$$q_{r}^{i,t}(x, x') > 0 \implies \bar{x}^{(i,t)} = \bar{x'}^{(i,t)} \text{ and } \lambda^t = \lambda'^t.$$ 

Note that the difference of energies between two configurations $x$ and $x'$ in $\left(\mathcal{X} \cap 2^{n-k}\mathbb{N}\right)^{S_r} \times \{x_0^{S_r}\}$ which differ only from one site $(i, t) \in S_r$ and on the value of bit $n - k$ simply reduces to 

$$U_k(x') - U_k(x) = \left(\chi(x_{i,t} \geq \lambda_{i,t}) - \chi(x'_{i,t} \geq \lambda_{i,t})\right)$$ 

$$\times \left\{ \sum_{j \in S} \xi_{i,j} \chi(x_{j,t} \geq \lambda_{i,t}) + \alpha \chi(x_{i,t-1} \geq \lambda_{i,t}) + \alpha \chi(x_{i,t+1} \geq \lambda_{i,t}) + \beta \left(y_{i,t} - \lambda_{i,t} - \frac{1}{2}\right) \right\}.$$ 

**Remark 4.1.** Usually, 

$$q_{r}^{i,t}(x, x') = \|\bar{x}^{(i,t)} = \bar{x'}^{(i,t)}, \left|\frac{x_{i,t}}{2^{n-k+1}}\right| = \left|\frac{x'_{i,t}}{2^{n-k+1}}\right|, x \neq x'\|,$$

which corresponds to a deterministic flip of the $(n - k)$th bit at site $(i, t)$. The movement at each step is described as follows: let $x$ denote the state of the dynamics at step $n$. In order to perform step $n + 1$, choose $x'$ with probability $q_{r}^{i,t}(x, x')$, then accept $x'$ as the state of the $n + 1$ step with probability 

$$\exp \left(-\frac{1}{\tau}(U_k(x') - U_k(x))_+ \right).$$ 

The computation of the energy difference described above is easy. 

Consider $L$ independent time homogeneous Markov chains 

$$((X_{n,l}^{i})_{n \in \mathbb{N}}, (\mathcal{X}^{S_r} \times \{x_0^{S_r}\}, \mathcal{P}_r) \text{ for } l = 1, \ldots, L \text{ with transition matrix } p_{r, L, r}.$$ 

Let 

$$\hat{X}^{i} = X_{M,l}^{i} = \text{arg min}\{U_k(X_{M,l}^{i}); l = 1, \ldots, L\},$$
the best solution from the \( L \) Metropolis attempts stopped at iteration \( M \). The number of iterations shall be tuned later on. Then, put

\[
\nu_r^*(x, x') = P_r(X_{M,l}^r = x' \mid X_{M,l}^r = x; l = 1, \ldots, l = L).
\]

We are ready now to define the multi-resolution dynamics on \( E \) as the non commutative matrices product

\[
q_r(x, x') = \left( \prod_{r=1}^R \nu_r^*(x, x') \right).
\]

Let \( H_3^r \) denote the maximal third critical depth of the classical Metropolis dynamics \( p_r(. \mid x^{S_r}) \).

**proposition 4.1.** For any \( \varepsilon > 0 \), if \( \tau \) is low enough, then

\[
M = \left[ \exp \frac{1}{L \tau} \left( \max_r H_3^r + \varepsilon \right) \right]
\]

iterations of the Markov chains \((X_{n,l}^r)_{n \in \mathbb{N}}\) are sufficient to reach equilibrium (the asymptotic behaviour). Thus, for any \( l \), any \( r \), any \( x, x' \) in \( \Lambda_{S_r}^l \times x_{S_r}^l \),

\[
d^{-1} e^{-\frac{1}{\tau}(U_k(x') - \min_{z \in x_{S_r}^l} U_k(z))} \leq \nu_r^*(x, x') \leq de^{-\frac{1}{\tau}(U_k(x') - \min_{z \in x_{S_r}^l} U_k(z))},
\]

for some constant \( d > 0 \) independent of \( \tau \) and \( U_k \), and the invariant measure \( \mu_r \) of the multi-resolution dynamics \( q_r \) satisfies, for some constant \( a > 0 \),

\[
a^{-1} e^{-\frac{1}{\tau}(U_k(x) - \min U_k)} \leq \mu_r(x) \leq ae^{-\frac{1}{\tau}(U_k(x) - \min U_k)}.
\]

**Proof:** We refer to [13] for a detailed proof.

**Implementation of the dynamics.** This dynamics needs three memory spaces of "window size" to store the original sequence, the current sequence and a buffer sequence which contains the current best state. It needs also two memory spaces of size "a real" to store the current energy value and the current best one. We briefly sketch the main loop and the buffer management:

- Dichotomic step k:
  for \((r = 1 \text{ to } r = R)\) "scanning every 3D-window" {
    \( U_{buf} = +\infty; \)
    for \((l = 1 \text{ to } l = L)\) {
      Metropolis restoration of \( x_{S_r} \), with the observation \( y_{S_r} \), (energy update during the restoration);
      if \((U_k(x) < U_{buf})\) {
        Refresh the buffer \( b_{S_r}; \)
        \( U_{buf} = U_k(x); \)
      }
  }
4.2. Partitioning of windows into conditionally independent components.
Let us put \( \Lambda(k) = \left\{ 2^{n-k} + 2^{n-k+1} l, l \in \mathbb{N} \right\} \cap \left[ 0; 2^n \right], \) for any \( k = 1, \ldots, n \) and let us consider the \( k \)th dichotomic step. Conditionally on the already restored level set \( \mathcal{L}_{k-2}(x) \), the following partition divides the lattice into independent components \( \{ C_{i}(x); \lambda \in \Lambda(k) \} \), where for any \( \lambda \in \Lambda(k) \),
\[
C_{i}(x) = L_{i-2n-k}(x) \setminus L_{i+2n-k}(x).
\]
We group there the pixels whose "already restored bits" are the same.
Therefore, we partition each window \( \mathcal{S}_{r} \) dynamically by grouping the pixels whose already restored bits are the same. We index each component \( \mathcal{S}_{r, \lambda}(x) \) of the partition by the intensity level \( \lambda \) corresponding to the next bit to be restored:
\[
\mathcal{S}_{r, \lambda}(x) = \mathcal{S}_{r} \cap C_{i}(x) = \left\{ (i, t) \in \mathcal{S}_{r}, \lambda_{i,t} = 2^{n-k} + 2^{n-k+1} \left\lfloor \frac{x_{i,t}}{2^{n-k+1}} \right\rfloor = \lambda \right\}.
\]
Conditionally on the random field \( \left( \left\lfloor \frac{x_{i,t}}{2^{n-k+1}} \right\rfloor \right)_{(i,t) \in \mathcal{S}_{r}} \) of "already restored bits", the partition \( \left( \mathcal{S}_{r, \lambda}(x) \right)_{\lambda \in \Lambda(k)} \) splits \( \left( x_{i,t} \right)_{(i,t) \in \mathcal{S}_{r}} \) into independent components.
On each independent component \( C_{i}(x) \), we consider the energy function defined by
\[
U_{k, \lambda}(x) = -\sum_{i=1}^{T} \sum_{t \in \mathcal{S}_{i}; \lambda_{i,t} = \lambda} \chi(x_{i,t} \geq \lambda) \left\{ \sum_{j \in \mathcal{S}_{i}} \left( \frac{1}{2} \mathbf{1}_{\lambda_{j,t} = \lambda} + \mathbf{1}_{\lambda_{j,t} \neq \lambda} \right) \xi_{i,j} \chi(x_{j,t} \geq \lambda) \right\}
\]
\[
+ \alpha \left( \frac{1}{2} \mathbf{1}_{\lambda_{i,t-1} = \lambda} + \mathbf{1}_{\lambda_{i,t-1} \neq \lambda} \right) \chi(x_{i,t-1} \geq \lambda)
\]
\[
+ \alpha \left( \frac{1}{2} \mathbf{1}_{\lambda_{i,t+1} = \lambda} + \mathbf{1}_{\lambda_{i,t+1} \neq \lambda} \right) \chi(x_{i,t+1} \geq \lambda) + \beta \left( y_{i,t} - \lambda - \frac{1}{2} \right) \}
\]
Therefore,
\[
P\left( \left[ \frac{x}{2^{n-k}} \right] \right| y, \left[ \frac{x}{2^{n-k+1}} \right], \Omega_{k-1} \right) = \bigotimes_{\lambda \in \Lambda(k)} P\left( \left[ \frac{x}{2^{n-k}} \right] \right| y, \left[ \frac{x}{2^{n-k+1}} \right], \Omega_{k-1} \right)
\]
\[
= \frac{1}{Z_{\tau}} \exp \left( -\tau \sum_{\lambda \in \Lambda(k)} U_{k, \lambda}(x) \right),
\]
and we have \( U_{k}(x) = \sum_{\lambda \in \Lambda(k)} U_{k, \lambda}(x) \).
Note that, for any \( \lambda \), any \( r \) and any \( x \), there exists \( \bar{U}_{k,\lambda} \) and \( R_{S_{r, \lambda}(x)} \) such that
\[
U_{k, \lambda}(x) = \bar{U}_{k,\lambda}(x_{\mathcal{S}_{r, \lambda}(x)} + \nu) + R_{S_{r, \lambda}(x)}(x_{\mathcal{S}_{r, \lambda}(x)}).
\]
During the dichotomic step $k$, conditionally on the "already restored" bits, the multiple independent attempts are performed on the domains $S_{r,\lambda}(x)$ for $r = 1$ to $r = R$ and $\lambda \in \Lambda(k)$. More precisely, we perform a deterministic scan of the windows. On each window $S_r$, a test on each pixel determines which domain $S_{r,\lambda}(x)$ for $\lambda \in \Lambda(k)$ is being restored.

**Description of implementation.** For convenience, define the global energy function $U$ as an array of reals with double precision of size $2^n$. Each element of the array will represent a current energy value $U(\lambda)$. Init the array by computing all the energies $U(\lambda)$ of the original sequence $x_0$, for $\lambda \in [1; 2^n]$.

During dichotomic step $k$, only the elements numbered $2^n - k + 2^{n-k+1}l$ with $l \in [0;2^{k-1}]$ will be modified. For instance, the computation on a difference of energies between two configurations $x$ and $x'$ which differ only at site $(i, t)$ and on bit $n - k$ induces a modification on $U_{\lambda_{i,t}}$ for $\lambda_{i,t} = 2^{n-k} + 2^{n-k+1}l \frac{x_{t + 2^{-k+1}}}{x_{t + 2^{-k+1}}}$. In order to implement the multi-resolution dynamics, we need to define a second array of size $2^n$ to store the current best energy values of each domain $\lambda$. We briefly sketch the main loop of the algorithm:

: Dichotomic step $k$:  
  for $(r = 1$ to $r = R)$ "scanning the 3D-windows" {  
    $\forall \lambda \in \Lambda(k)$, $U_{\text{buff}}[\lambda] = +\infty$;  
    for $(l = 1$ to $l = L)$ {  
      Metropolis restoration of $x_{Sr}$ with the observation $y_{Sr}$,  
      (update the energies $U[\lambda], \lambda \in \Lambda(k)$);  
      for ($\lambda \in \Lambda(k)$) {  
        if ($U[\lambda] < U_{\text{buff}}[\lambda]$) {  
          Refresh the buffer $b_{Sr,\lambda}$ with the better  
          configuration reached in $S_{r,\lambda}$;  
          $U_{\text{buff}}[\lambda] = U[\lambda]$;  
        }  
      }  
    }  
    Copy the buffer $b_{Sr}$ in $x_{Sr}$:  
    $\forall \lambda \in \Lambda(k)$, $U[\lambda] = U_{\text{buff}}[\lambda]$;  
  }

4.3. Multi-resolution model with hierarchical trials. *The hierarchical model.* On each window $S_r$, we consider two levels of resolution: $S_r$ will still denote the standard grid, and $S'_r$ will denote a coarser grid included in $S_r$.

Let $a$ be the spatial compression coefficient from $S_r$ to $S'_r$ and $\delta$ be the temporal compression coefficient. Namely, let us put $a = 1/p'$ and $\delta = T/T'$, where $S_r =$
$l \times l \times T$ and $S'_r = l' \times l' \times T'$.

Let $\Pi : S_r \rightarrow S'_r$ be the canonical projection from the fine sites to the coarse ones. Any coarse site $s \in S'_r$ corresponds to a cube of fine sites $\Pi^{-1}(s) \subseteq S_r$, a "cube of resolution" of area $a^2 \times \delta$.

Let us stress the fact that any of the classical sub-sampling operations such as linear-filtering, or median-filtering, working on the values of grey levels seems to be inappropriate to the dichotomic procedure. Indeed, we use a median-filtering that works on the bit values.

Consider the $k$th dichotomic step. The sub-sampling operation $\mathcal{S} : \Lambda^{S_r} \rightarrow \Lambda^{S'_r}$ consists, for every bits from $n - 1$ to $n - k$, in an election by the majority of 1 or 0. The description of the main loop by which a coarse pixel is built from a "cube of resolution" of fine pixels follows:

```plaintext
: MASK = 1 << n - 1;
count = 0;
m = a^2 \delta / 2;

Scanning every sites $s$ of the "resolution cube" $\Pi$ {
    if ($x_s \geq MASK$) {
        count = count + 1;
    }
}
if (count $\geq m$) {
    result = MASK;
} else {
    result = 0;
}

MASK1j = 1 << n - 1;
for ($j = 2$ to $j = k$) {
    count = 0;
    MASKjj = 1 << n - j;
    MASK1j = MASK1j $\mid$ MASKjj;
    Scanning every sites $s$ of the "resolution cube" $\Pi$ {
        if ($x_s \geq (result \& MASK1j) \mid MASKjj$) {
            count = count + 1;
        }
    }
    if (count $\geq m$) {
        result = result $\mid$ MASKjj;
    } else {
        result = result $\&$ ($\sim$ MASKjj);
    } 
}
```
return (result);

The sub-sampling operator $S : \Lambda^{S_r} \rightarrow \Lambda^{S_r}$ defined this way is a surjective mapping. The decompression operation, inversely, is just the canonical injection $I : \Lambda^{S_r} \rightarrow \Lambda^{S_r}$, $(I(x'))_s = x'_{\Pi(s)}$.

The multi-resolution dynamics using hierarchical attempts. Let $k$ be fixed. The multiple independent local attempts of the multi-resolution dynamics will hold on two levels of resolution. A hierarchical attempt consists in some pre-processing on $S'_r$ and some post-processing on $S_r$.

To define a hierarchical attempt, let $\overline{x}_0^{S_r}$ be the outside window configuration and $L$ the number of attempts performed, and let $S'_r = \{(i, t) \mid (i, t) + \forall \in S'_r\}$ denote the interior of $S'_r$. We choose the family $S_r$ such that

$$\bigcup_r \Pi^{-1}(S'_r) = S.$$  

The pre-processing of a hierarchical attempt is some Metropolis dynamics at temperature $L\tau$ defined on $\left(\Lambda \cap 2^{n-k} \mathbb{N}\right)^{S'_r} \times x_0^{\pi \setminus S'_r}$ by $p_{r,L\tau} = \prod_{(i,t) \in S'_r} p_{i,t}^{r,L\tau}$, where

$$p_{i,t}^{r,L\tau}(x, x') = \begin{cases} q_{i,t}^{r,L\tau}(x, x') e^{-\frac{1}{L\tau} (U_k(I(x')) - U_k(I(x)))} & \text{if } x \neq x' \\ 1 - \sum_{z \neq x, z \neq x'} p_{i,t}^{r,L\tau}(x, x') & \text{if } x = x', \end{cases}$$

for some perturbation kernel $q_{i,t}^{r,L\tau}$ irreducible with symmetrical support. In our applications, it is a deterministic bit-flip dynamic.

The computation on the energy difference between two coarse configurations $x$ and $x'$ which differ only from one coarse site $(i, t) \in S'_r$ and on the value of bit $n-k$ reduces simply

$$U_k(I(x')) - U_k(I(x)) = \left(\chi(x_{i,t} \geq \lambda_{i,t}) - \chi(x'_{i,t} \geq \lambda_{i,t})\right)$$

$$\times \left\{ \sum_{j:(j,t) \in S'_r, |j-i|=1} \delta(a(1 + \sqrt{2}) - \sqrt{2}) \chi(x_{j,t} \geq \lambda_{i,t}) \right.$$  

$$+ \sum_{j:(j,t) \in S'_r, |j-i|=\sqrt{2}} \delta \frac{1}{\sqrt{2}} \chi(x_{j,t} \geq \lambda_{i,t}) + \alpha a^2 \chi(x_{i,t-1} \geq \lambda_{i,t})$$

$$+ \alpha a^2 \chi(x_{i,t+1} \geq \lambda_{i,t}) + \beta \delta a^2 (y_{i,t} - \lambda_{i,t} - \frac{1}{2})\right\},$$
where we let for any $(i,t) \in S'_{r}$, $y_{i,t} = \frac{1}{a^2 \delta} \sum_{s \in \Pi^{-1}(i,t)} y_{s}$. The post-processing is the Metropolis dynamics on $S_{r}$ already defined in the previous sub-section.

**Updating the energy during the sub-sampling operation.** Consider the sub-sampling projection operator $\mathcal{S} : \Lambda^{S'}_{r} \rightarrow \Lambda^{S'_{r}}$, described at the beginning of this section.

Let $x \in \Lambda^{S'}_{r}$, and note $X_{y_{i,t} = 1} a^2 / x_{S} / B_{n} Z_{r} / (i,t)$.

We have, $U_{k}(X) - U_{k}(x) = \sum_{(i,t) \in S_{r}} \Delta U_{r,i,t}$ with

$$
\Delta U_{r,i,t} = \left( \chi(x_{i,t} \geq \lambda_{i,t}) - \chi(X_{i,t} \geq \lambda_{i,t}) \right) \times \left\{ \sum_{j \in (j,t) \in S_{r}, i < j} \xi_{i,j} \chi(x_{j,t} \geq \lambda_{i,t}) 
+ \sum_{j : (j,t) \in S_{r}, j < i} \xi_{i,j} \chi(X_{j,t} \geq \lambda_{i,t}) + \alpha \chi(X_{i,t-1} \geq \lambda_{i,t}) 
+ \alpha \chi(x_{i,t+1} \geq \lambda_{i,t}) + \beta \left( y_{i,t} - \lambda_{i,t} - \frac{1}{2} \right) \right\},
$$

as $\lambda_{i,t} = 2^{n-k} + 2^{n-k+1} \left\lfloor \frac{x_{i,t}}{2^{n-k+1}} \right\rfloor = 2^{n-k} + 2^{n-k+1} \left\lfloor X_{i,t} \right\rfloor$.

The computation of $\Delta U_{r,i,t}$ may be done during the scanning of pixels of the window $S_{r}$.

**Implementation of the dynamics.** This dynamics needs two buffer memory spaces of size ”a coarse window”. The first one will contain the original sequence sub-sampled with a medium-filter. The second one will contain the current coarse sequence. It needs also one more array of size $2^{n}$ to store the energy values $U_{\lambda}$ for $\lambda \in \{1, \ldots, 2^{n} - 1\}$ reached after each pre-processing attempt. We briefly sketch the main loop:

: Dichotomic step $k$:

for (r = 1 to r = R) ”scanning the 3D-windows” { 
    Sub-sample $x_{S_{r}}$. to $x_{S'_{r}}$ on $S_{r}$.
    (update the energies $U[\lambda]$, $\forall \lambda \in \Lambda(k)$ during the sub-sampling operation).
    Sub-sample $y_{S_{r}}$. to $y_{S'_{r}}$.
    $\forall \lambda \in \Lambda(k)$, $U_{\text{buff}}[\lambda] = +\infty$.
    for (l = 1 to l = L) { 
        Pre-processing restoration of $x_{S'_{r}}$, update the energies $U[\lambda]$, $\lambda \in \Lambda(k)$.
        Decompression of $x_{S'_{r}}$. into $x_{S_{r}}$, update the energies $U[\lambda]$, $\lambda \in \Lambda(k)$.
        Post-processing restoration of $x_{S_{r}}$, update the energies $U[\lambda]$, $\lambda \in \Lambda(k)$.
        for ($\lambda \in \Lambda(k)$) { 
            if ($U[\lambda] < U_{\text{buff}}[\lambda]$) { 
                Refresh the buffer $b_{S_{r},\lambda}$ with the better configuration reached in $S_{r,\lambda}$.
                $U_{\text{buff}}[\lambda] = U[\lambda]$.
            }
        }
    }
}
A series of synthetic and real sequences of images are shown and discussed here to illustrate our approach.

The first sequence is made from a geometrical image $256 \times 256$. It is static in time and contains 6 images. The original geometrical image contains two connected level sets, and gets two unique grey levels denoted $\mu_1 < \mu_2 \in [0; 2^8]$. The difference of grey levels $\mu_2 - \mu_1$ has been chosen so that smoothing is a hard task. We took $\mu_1 = 110$ and $\mu_2 = 122$. The sequence has been corrupted with an i.i.d. additive Gaussian noise $\mathcal{N}_3(0, 40^2)$.

The second and third sequences are time-varying ones. Their original image contains many level sets. They have been corrupted with some i.i.d. additive Gaussian noise.

The windows size and locations: In order to show the robustness of the choice of the size of windows of the multi-resolution model, we considered $3 \times 21 \times 21$ cubic windows in all sequences. As for their locations on the space-lattice, we chose the paving illustrated in figure 1.

![Figure 1. Pattern from the paving of the space-lattice by overlapping windows.](image)

The tuning of parameters of the hierarchical model: There is $\alpha$ the space-lattice compression parameter, and $\delta$ the time compression one. We chose $\alpha = 3$ and $\delta = 1$ for all sequences.

The tuning of parameters of the energy model: There is $\alpha$ the temporal regularization parameter, and $\beta$ the noise parameter.

As for their tuning, we followed the theoretical indications of section 2, that is $\beta$ must be chosen according to the details that one want to eliminate.
For the first sequence of size $6 \times 256 \times 256$, we chose $\beta$ to keep details of diameter 10 and maximal grey variation 40. We tuned $\beta = 0.025$.

For the second sequence $4 \times 337 \times 268$, as the original image has lots of sharp details, we considered that the critical spatial blob has maximal diameter 2 and maximal grey variation 10, so that we tuned $\beta = 0.4$.

As for the third sequence $9 \times 252 \times 253$, we considered that spatial drops of maximal size $d_0 \times d_0$ and maximal grey-variation $\lambda_0$ such that $d_0 \lambda_0 = 120$ should disappear. Thus we tuned $\beta = 0.08$.

Note that, for the temporal regularization parameter $\alpha$, the larger it is, the more uniform in time the solutions should be. We tuned $\alpha = 0.6$ for all sequences.

*The influence of each parameter:* The first sequence, being static is not suitable to study the influence of $\alpha$. The noise parameter $\beta$ is very sensitive. The larger it is, the more difficult it is to eliminate the noise. We studied more precisely its influence on the first sequence (table 1). Let $\hat{\sigma}_{diff}$ denote the standard deviation of the difference between the observation and the restored sequence, let also put $\hat{\sigma}_1$, $\hat{\mu}_1$, and $\hat{\sigma}_2$, $\hat{\mu}_2$ for the estimated standard deviations and means on both homogeneous areas of the restored sequence.

<table>
<thead>
<tr>
<th>$d_0 \lambda_0$</th>
<th>$\beta$</th>
<th>$\hat{\sigma}_{diff}$</th>
<th>$\hat{\mu}_1$</th>
<th>$\hat{\sigma}_1$</th>
<th>$\hat{\mu}_2$</th>
<th>$\hat{\sigma}_2$</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>960</td>
<td>0.01</td>
<td>40.08</td>
<td>111.21</td>
<td>0.89</td>
<td>121.57</td>
<td>0.90</td>
<td>image structure unkept</td>
</tr>
<tr>
<td>400</td>
<td>0.025</td>
<td>40.03</td>
<td>110.60</td>
<td>1.19</td>
<td>122.13</td>
<td>0.84</td>
<td>best result</td>
</tr>
<tr>
<td>240</td>
<td>0.04</td>
<td>40.00</td>
<td>111.12</td>
<td>1.28</td>
<td>122.34</td>
<td>1.10</td>
<td>$20 \times 20$ artefacts</td>
</tr>
<tr>
<td>120</td>
<td>0.08</td>
<td>39.64</td>
<td>110.77</td>
<td>4.39</td>
<td>122.21</td>
<td>4.22</td>
<td>Gaussian noise not vanished</td>
</tr>
</tbody>
</table>

*Table 1.* The tuning of the noise parameter $\beta$, made with “multi-resolution dichotomic” algorithm.

As for the temperature parameter, we tuned $1/\tau = 1.5$ for all sequences (to show its robustness). For a lower $\tau$, the energy barriers become too high, and the algorithms are close to stochastic gradient descents. They reach a local non global minimum of the energy function. The restored sequence appears smoothed with many spatial blobs with low variations of intensities that manifestly do not come from the structure of the original image. For a higher $\tau$, it becomes difficult to get any algorithm to stabilize in a reasonable time.

*Comparison of the performances of the algorithms:* We implemented the ”dichotomic” algorithm and the ”multi-resolution dichotomic” one. The latter uses multiple local hierarchical attempts. We keep the best solutions on ”contour lines” domains.
Two criteria help to determine the stabilization of the algorithms: If no visible improvement is observed while increasing the number of sweeps, and if the energy/time function used for each dichotomic step does not go down in the mean any more, then we will consider that stabilization is reached. For the first sequence, of size $6 \times 256 \times 256$, stabilization of the "multi-resolution dichotomic" algorithm is reached after 2 local hierarchical attempts with 10 sweeps per bit for pre-processing and 10 sweep per bit for post-processing. Only one movie-scan per dichotomic step is performed. The restoration takes about 520 seconds of CPU times (when implemented in C on a Pentium 133 Mega Hertz). For the same sequence, stabilization of the "dichotomic" algorithm is reached after 200 sweeps per bit (that takes about 1430 seconds of CPU times). It is up to 3 times slower.

To compare the restored sequence of both algorithms, we stopped them after an equivalent number of 40 visits of the pixels per dichotomic step (i.e. the time the "multi-resolution dichotomic" algorithm stabilizes). As for the quality of the restored sequence by the "multi-resolution dichotomic" dynamics and by the "dichotomic" dynamics, we get the following statistical results (refer to tables 2 and 3):

<table>
<thead>
<tr>
<th>iteration per bit for pre-process</th>
<th>iteration per bit for post-process</th>
<th>$\hat{\sigma}_{diff}$</th>
<th>$\hat{\mu}_1$</th>
<th>$\hat{\sigma}_1$</th>
<th>$\hat{\mu}_2$</th>
<th>$\hat{\sigma}_2$</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>40.03</td>
<td>110.60</td>
<td>1.19</td>
<td>122.13</td>
<td>0.84</td>
<td>Stabilization reached</td>
</tr>
</tbody>
</table>

**Table 2.** Quality results for "multi-resolution dichotomic" algorithm.

<table>
<thead>
<tr>
<th>iteration per bit</th>
<th>$\hat{\sigma}_{diff}$</th>
<th>$\hat{\mu}_1$</th>
<th>$\hat{\sigma}_1$</th>
<th>$\hat{\mu}_2$</th>
<th>$\hat{\sigma}_2$</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>39.89</td>
<td>110.99</td>
<td>1.22</td>
<td>121.85</td>
<td>1.19</td>
<td>Stabilization not reached</td>
</tr>
<tr>
<td>200</td>
<td>39.9</td>
<td>110.83</td>
<td>1.03</td>
<td>121.86</td>
<td>0.85</td>
<td>Stabilization reached</td>
</tr>
</tbody>
</table>

**Table 3.** Quality results for "dichotomic" algorithm.

Figures (2;3;4) illustrate the quality of the restorations on this static and geometrical sequence. Figure (2) is the original image with no change of the contrast. In figure (4), we modified the contrast of the images just for the representation here, for a better eye-appreciation: the grey-intensities are kept linear between 40 and 140 but with values in $\{0; 255\}$, and they are constant elsewhere.
Figures (5;6;7;8) illustrate the quality of the restorations on the second sequence. The sequence has been corrupted with an additive Gaussian noise $N_3(0, \sigma^2)$ with $\sigma = 10$. The peak signal-to-noise ratio $\max x_s/\sigma$ is around 25. An eye-appreciation would not detect any difference. But the time for stabilization is not the same: for this $4 \times 337 \times 268$ sequence, 40 sweeps per bit where needed by the "dichotomic" algorithm (that are 168 seconds), and by the second algorithm, 2 attempts with 20 sweeps per bit for pre-processing and 1 sweep per bit for post-processing (that are 85 cumulative seconds). As post-processing takes most of the time compared to pre-processing, it is important to decrease it as far as possible.

Figure (9) illustrates the quality of the restorations on the third sequence of size $9 \times 252 \times 253$. The sequence has been corrupted with an additive Gaussian noise $N_3(0, 50^2)$. The peak signal-to-noise ratio is around 5. The "dichotomic" algorithm used 200 sweep per bit until equilibrium (that are around 1900 seconds). The "multi-resolution dichotomic" algorithm used 2 localized hierarchical attempts with 50 sweeps per bit for pre-processing and 1 sweep per bit for post-processing (that are around 400 cumulative seconds).

In conclusion, one can say that the model of additive Gaussian noise used here is a natural model in many situations. It is usefull for a wider range of applications than the i.i.d. noise model in which a certain percent of pixels are corrupted, the other ones being unchanged that was used in [9]. The dichotomic approach seems to be well adapted to reconstruct images with several grey levels. The first reason is that it leads to a bit after bit reconstruction of pixels, the second reason is that it leads to a very significant acceleration compared with the classical global method as it is check in [9]. The method of multi-resolution and hierarchy can be adapted to the dichotomic model, and bring further additional acceleration. The experimental results illustrates this fact. This article is a companion paper to the theoretical study of speed-up techniques for Metropolis algorithms on a lattice ([13]). The latter proposes a mathematical study of the acceleration rates provided by the multi-resolution and the hierarchical dynamics.
Figure 2. Geometrical image size $256 \times 256$, two grey levels $\mu_1 = 110$ and $\mu_2 = 122$.

Figure 3. Noisy image size $256 \times 256$. 
Figure 4. Geometrical image size $256 \times 256$ (upper left), image corrupted by i.i.d. additive Gaussian noise $\mathcal{N}(0, 40^2)$ (upper right), Solution of the "dichotomic" algorithm after 40 sweeps per bit (lower left), Solution of the "dichotomic multi-resolution" algorithm with 2 local hierarchical attempts of 10 sweeps per bit for pre-processing and 10 sweeps per bit for post-processing (lower right). All the images of this figure are represented here with an modified linear grey level histogram in the range 40-140.
Figure 5. Original image size $337 \times 268$ (up), image corrupted by i.i.d. additive Gaussian noise $N(0, 10^2)$, peak signal to noise ratio around 25 (down).
Figure 6. Solution obtained by the "dichotomic" algorithm after 40 sweeps per bit (up). Solution of the "dichotomic multi-resolution" algorithm with 2 local hierarchical attempts with 10 sweeps per bit for pre-processing and 1 sweep per bit for post-processing (down).
Figure 7. Zoom on noisy image.
Figure 8. Zoom on "dichotomic" Solution (left), zoom on "dichotomic multi-resolution" solution (right).
Figure 9. Original image size $252 \times 253$ (upper left), image corrupted by i.i.d. additive Gaussian noise $N(0,50^2)$, peak signal to noise ratio around 5 (upper right), Solution of the "dichotomic" algorithm after 200 sweeps per bit (lower left), Solution of the "dichotomic multi-resolution" algorithm with 2 local hierarchical attempts with 50 sweeps per bit for pre-processing and 1 sweep per bit for post-processing (lower right).
References