IMAGE DECONVOLUTION USING A STOCHASTIC DIFFERENTIAL EQUATION APPROACH*

X. Descombes[1], M. Lebellego[1,2]
[1] *Ariana Research Group, INRIA/I3S, 2004, route des lucioles, BP 93, 06902, Sophia Antipolis Cedex, France*

E. Zhizhina[2]
[2] *Dobrushin Laboratory, IITP of RAS, Bol'shoi Karetnyi per., 19, 127994, GSP-4, Moscow, Russia*

Keywords: Image deconvolution, Stochastic Differential Equation, Langevin dynamics, Euler approximation.

Abstract: We consider the problem of image deconvolution. We foccus on a Bayesian approach which consists of maximizing an energy obtained by a Markov Random Field modeling. MRFs are classically optimized by a MCMC sampler embeded into a simulated annealing scheme. In a previous work, we have shown that, in the context of image denoising, a diffusion process can outperform the MCMC approach in term of computational time. Herein, we extend this approach to the case of deconvolution. We first study the case where the kernel is known. Then, we address the myopic and blind deconvolutions.

1 INTRODUCTION AND METHOD

Image restoration is a wellknown ill-posed problem which has motivated many works. A first complete review of image restoration approaches was given (Andrews and Hunt, 1977). Since then, numerous approaches have been proposed, among them variational and stochastic approaches play a leading role. However, there is still no completly satisfactory solution, especially in case of blind deconvolution, for which the kernel is unknown. As an ill-posed problem, image restoration is adapted to Bayesian approaches which embed a prior model, which constrains the solution. Therefore, models based on Markov Random Fields (MRFs), preserving the discontinuities while restoring the data have been proposed (Geman and Reynolds, 1992). More sophisticated Markov models have been recently proposed, such as in (Molina et al., 2000; Mignotte, 2006). Classically, MRFs, for image restoration, are optimized using a Gibbs sampler embedded in a simulated annealing scheme (Geman and Geman, 1984). In this paper, we consider a classical MRF modelling but explore a new optimization scheme based on a stochastic differential equation approach. In a previous work, we have shown that this new scheme outperforms the Gibbs sampler in term of computational time in the case of image denoising (Descombes and Zhizhina, 2004). We extend this work to the deconvolution problem.

*The authors would like to thank EGIDE for partial financial support within the ECONET project 10203YK.

1.1 The Stochastic Approach for Image Deconvolution

We consider a degraded image Y. The degradation, including noise and blurring can be modelled by the following equation :

$$Y = K * X + n \quad (1)$$

Where X is the original image without noise, that we want to reconstruct, n is a Gaussian additive noise and K a convolution kernel.

Then the different steps of this approach are the following :

- We define an energy function associated to a configuration X
- We construct a diffusion process based on this energy thanks to the Langevin operator
- We derive from this process a discretized process for computer simulations.

Descombes X., Lebellego M. and Zhizhina E. (2007).
IMAGE DECONVOLUTION USING A STOCHASTIC DIFFERENTIAL EQUATION APPROACH.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications*, pages 157-164
DOI: 10.5220/0002064701570164
Copyright © SciTePress

- We finally define an estimator to optimize the model and find a solution

1.2 Energy Function

To the configurations X, we associate an energy function, which is interpreted as the Hamiltonian of a Gibbs field.

The energy is defined by the sum of an interaction term, modelling some prior knowledge and a data driven term:

$$H(X,Y) = \Phi_1(X) + \Phi_2(X,Y) \tag{2}$$

with

$$\Phi_1(X) = \beta \cdot \sum_{(i,j)\in\Lambda, j\in V(i)} U(X_i - X_j) \tag{3}$$

and

$$\Phi_2(X,Y) = \lambda \cdot \sum_i ((K * X)(i) - Y_i)^2 \tag{4}$$

Where Λ is the set of all the pixels in the image. And $V(i)$ is the neighborhood of the pixel i. λ and β are two parameters of the model. β controls the smoothness and λ the weight of the data.

The function U is the following :

$$U(X_i - X_j) = -\frac{1}{1 + \frac{(X_i - X_j)^2}{d^2}} \tag{5}$$

U is a ϕ-function, as proposed in (Geman and Reynolds, 1992), d is a parameter. The bigger d, the smoother the image.

1.3 The Langevin Equation

We then want to construct a diffusion process :

$$X(t) = \{X_i(t) \in [0,512], i \in \Lambda\}$$

on the configuration space $E = [0,512]^{|\Lambda|}$, which is a stationary Markov process with the Gibbs measure associated with the above Hamiltonian $H(X,Y)$:

$$d\mu_\sigma = \frac{e^{-\frac{2}{\sigma^2} \cdot H(X,Y)}}{Z_\sigma} \cdot d\mu_\sigma \tag{6}$$

To construct this process, we consider the functional Hilbert space $\mathcal{L}^2(E, d\mu_\sigma)$

Let's us consider the operator L_f defined on the function space E by the following equation :

$$\int L_\sigma f \cdot d\mu_\sigma = 0 \tag{7}$$

where $d\mu_\sigma$ is the Gibbs measure. It is a generator of the stationary process with the invariant measure μ_σ. It is a generator of a Langevin dynamics. There is not a unique solution of the equation (6), but the following generator is one of them :

$$\begin{aligned} L_\sigma f &= \frac{1}{2} \cdot \sigma^2 \cdot \Delta f - \nabla H \cdot \nabla f \\ &= \frac{1}{2} \cdot \sigma^2 \cdot \sum_{i\in\Lambda} \frac{\partial^2 f}{\partial x_i^2} - \sum_{i\in\Lambda} \frac{\partial H}{\partial x_i} \cdot \frac{\partial f}{\partial x_i} \end{aligned} \tag{8}$$

This generator is a generator of a diffusion process. We now have an operator defined on the functional space. From this process on the functional space, it is possible to reconstruct the process on the configuration space.

Using the relation between the two processes, we get the following stochastic equation, describing the evolution of the configuration :

$$dX(t) = \underbrace{\sigma \cdot dW(t)}_{I} - \underbrace{\nabla_X H(X,Y) \cdot dt}_{II} \tag{9}$$

The second term of this equation (II), is a deterministic term, depending on the gradient of the energy function. The first term of this equation (I), is a diffusive term, $W = \{W(t), t \geq 0\}$ being a m-dimensional Wiener process. So this equation can be interprated as a gradient descent with a random part, σ being the temperature of the scheme.

Since the stochastic equation describes the stationary process, the realization of $X(t)$ at time t will be a typical configuration of the Gibbs measure $d\mu_\sigma$.

1.4 The Euler Approximation

In section 1.3, we have constructed a continuous process. But we need to discretize it to perform computer simulations. Therefore, we consider an approximation of the process by a discrete time Markov process.

To discretize the process, we use the Euler approximation (Kloeden and Platen, 1992). We consider a time discretization of the interval $[0,t]$: $\tau(\delta) = \{\tau_n, n = 0..n_t\}$ by time steps $\delta_n = \tau_{n+1} - \tau_n$

The approximation process $Z(n) = \{Z_i(n), i \in \Lambda\}$, $n = 0..n_t$ has the same initial state $X(0)$ as the process $X(t)$, and can be constructed by the following iterative scheme :

$$\begin{cases} Z_i(0) = X_i(0) \\ Z_i(n+1) = Z_i(n) + a_i(Z(n),Y)\cdot\delta_n + \\ \qquad\qquad \sigma\cdot(W(\tau_{n+1}) - W(\tau_n)) \end{cases} \quad (10)$$

Where $a_i(Z_n,Y) = -\nabla_i H(Z(n),Y)$ and $\sigma\cdot dW$ is a Wiener process. In practice, $W(\tau_{n+1}) - W(\tau_n)$ can be simulated by sampling a centered normal law $\mathcal{N}(0,\delta_n)$ with a variance equal to δ_n.

1.5 The Estimator

Finally, we define an estimator which optimizes the Hamiltonian. We use here the Maximum A Posteriori (MAP) criterion. The MAP criterion consists in minimizing the energy H :

$$\hat{X} = argmin_X H(X,Y)$$

So we are looking for a configuration X giving the global minimum of the Hamiltonian. To estimate $\hat{X}$, we apply a simulated annealing scheme where the temperature parameter σ decreases during iterations.We also make decrease the time discretization parameter of the equation (10) : δ_n .

In theory to avoid local minima, the decreasing scheme of the parameter σ have to be logarithmic. In practice, for some computational reasons , we consider an exponential decreasing scheme for both parameters : $e^{\alpha\cdot t}$ but with α close to zero. For the tests, we have considered σ decreasing from 1 to 0.01 and between 1000 and 3000.

1.6 Results with a Known Kernel

We first consider that the convolution kernel K is known. That means we know exactly the blur of the picture, which is of course a strong constraint. The second step will be to consider that we don't know this kernel.

The simulations for the different algorithms have been done on two different 128x128 images : a synthetic image consisting of several uniform areas (see figure 1) and Lena picture (see figure 2), which were blurred by a 7x7 Kernel, and on which we have added a centered Gaussian noise (with different standard deviations s). Here, we assume that we know exactly the convolution kernel. Results on the synthetic image for $s = 0,3,10$ are shown on figure 1 for the synthetic image and on figure 2 for Lena picture.

For high level of noise ($s = 10$), we have to consider a stronger prior (high value for parameter β) which leads to an edge delocalization.

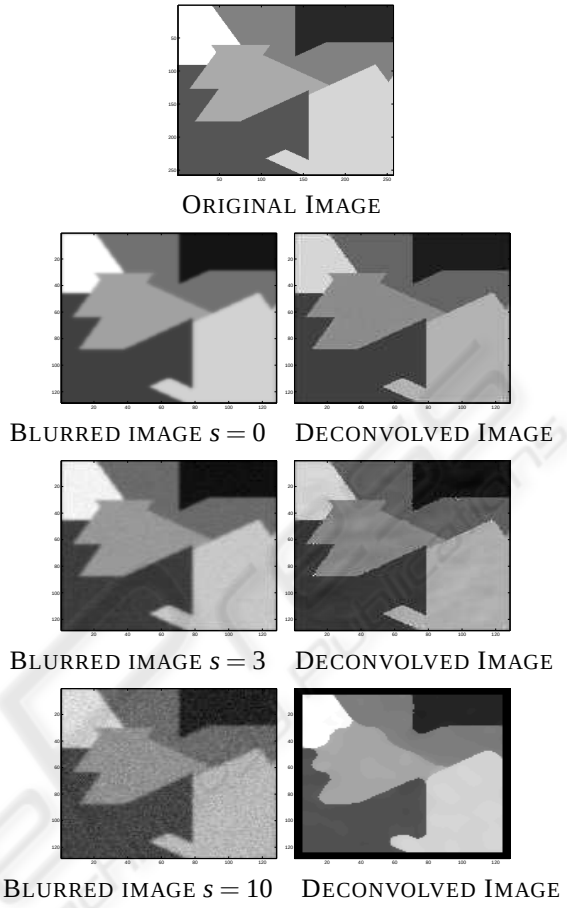

Figure 1: Deconvolution for $s = 0,3,10$.

In practice, we rarely have any information about the kernel. So let's now consider the case where the kernel is unknown. We then have to estimate its coefficients or to introduce a parametric model for the kernel.

2 BLIND DECONVOLUTION

2.1 A Stochastic Scheme for K

In this second scheme, we have two unknowns to update at each iteration: the current image X and the kernel K.
The stochastic scheme for X is :

$$X_i(n+1) = X_i(n) + a_i(X(n),K(n),Y)\cdot\delta_{n_1} + \sigma_1\cdot (W(\tau_{n+1}) - W(\tau_n))$$

Where $a_i(X(n),K,Y) = -\nabla_{X_i} H(X(n),K,Y)$

And we now introduce a stochastic scheme for K :

ORIGINAL IMAGE

BLURRED IMAGE

RECOVERED IMAGE

Figure 2: Deconvolution for $s = 1$.

$$\begin{aligned} K_i(n+1) &= K_i(n) + b_i(X(n),K(n),Y)\cdot\delta_{n_2} \\ &+ \sigma_2\cdot(W(\tau_{n+1}) - W(\tau_n)) \end{aligned} \quad (11)$$

Where $b_i(X,K,Y) = -\nabla_{K_i} H(X,K,Y)$

Prior Knowledge on K

Here we do not want to introduce a strong prior knowledge on K, because we assume we don't know anything on K. But still, we want it to be a convolution kernel, so it has to be normalized. But this prior knowledge doesn't play any role in the energy function. We don't introduce it in the prior term but just normalize the kernel at the end of each iteration.
We thus suppose, if we denote $K = (k_s)$, that $\sum_s k_s$=1

K Dynamics

The energy function is : $H(X,K,Y) = \lambda_1\cdot\phi_1(X) + \lambda_2\cdot\phi_2(X,K,Y)$
ϕ_1 is the prior knowledge term, ϕ_2 is the data-attached term. And :

$$\begin{aligned} \phi_2(X,K,Y) &= \sum_{i,j}(K * X_{i,j} - Y_{i,j})^2 \\ &= \sum_{i,j}\left(\sum_{u=(v,w)} k_v\cdot X_{i+v-N,j+w-N} - Y_{i,j}\right)^2 \end{aligned} \quad (12)$$

Where :

$$\begin{cases} u = v\cdot dimK + w, w < dimK \\ N = \frac{dimK-1}{2} \end{cases}$$

So the derivative of the energy w.r.t. k_s is :

$$\frac{\partial\phi_2}{\partial k_s} = 2\cdot\sum_{i,j} X_{i+a-N,j+b-N}(K * X(i,j) - Y_{i,j})$$

Where :

$$\begin{cases} s = a\cdot dimK + b, b < dimK \\ N = \frac{dimK-1}{2} \end{cases}$$

And finally

$$\begin{aligned} b_i(X,K,Y) &= -\lambda_2\cdot\frac{\partial\phi_2}{\partial k_i} \\ &= -2\cdot\lambda_2\cdot\sum_{i,j} X_{i+a-N,j+b-N}(K * X(i,j) - Y_{i,j}) \end{aligned} \quad (13)$$

Scheme Parameters
Let's come back to the new stochastic scheme:

$$K_i(n+1) = K_i(n) + b_i(X(n),K(n),Y)\cdot\delta_{n_2} + \sigma_2\cdot(W(\tau_{n+1}) - W(\tau_n))$$

We can suppose that this second scheme has no connection with the first one: the variations of X_i and k_s at each iteration are not of the same order. So we have different coefficients δ and σ for the two schemes.

Again we choosed a normal law for the probabilistic part. Here k_s belongs to $[0,1]$. So the more intuitive is to take again a normal law centered on 0. So that we have a high probability not to move far.

Update of X and K

We can then wonder about the order of updating of the two unknowns. We have two possibilities :

1. At each iteration I of the scheme, we update X and then K
2. We make N iterations for X and then M for K and then we come back to X with the new K and so on $\cdots$

Each one has drawbacks :
In the first one, we recalculate the Kernel after each iteration of X. But at each iteration, there are only slight changes on X. And so the recalculation of the kernel at each steps does not take into account enough changes. In the second one, let's suppose we have a poor estimate of the kernel in the current configuraiton. Then, making a big number of iterations on X degrades it strongly, and then the next calculation on the kernel will be even worse. So each new update degrades more the picture. In such a case, to converge to the MAP would require a very low decreasing scheme for the parameters δ and σ. In practice we choose the second scheme but with moderate values for N and M.

As we have now introduced a new scheme and new calculations of the derivatives, the computational time required to obtain the convergence will be highly increased. However, here we have a possibility to minimize the time of calculation: instead of calculating the parametric kernel with information from the entire image, a little window in the image is used to find out the information to evaluate the Kernel. This is possible because we assumed that the blur is uniform on the image. This window can be chosen randomly in the image or can be fixed at the beginning of the algorithm. But we have to be careful in the choice of this window, because if we pick up an homogeneous part of the image, without any information on edges, this window will not contain any information on the blur. So the evaluation of the kernel will not be accurate.

2.2 Results

The introduction of the convolution product into the derivative of the of ϕ_2 has brought a first difficulty.

The problem with this derivative $\left(\frac{\partial \phi_2}{\partial k_s} = 2 \cdot \sum_{i,j} X_{i+a-N,j+b-N}(K * X(i,j) - Y_{i,j})\right)$ is that the value is very big. In fact, it is a sum on all the pixels of the image and each term has a high value, negative or positive.

Consequently we had to ponderate this term so that we don't have too big steps between the former coefficient of K and the new one. But if we introduce a very small coefficient λ_2 it will also affect the stochastic scheme for X. And as the derivative of ϕ_2 is not that high w.r.t X_i, the attach to the data will not be strong enough. We introduced a coefficient which ponderates this derivative but only in the stochastic scheme concerning K. This coefficient is quite large and depends on the size of the image. The bigger the image, the bigger this coefficient.

Figure 3: Blurred image (top), Recovered after 4000 iterations (bottom).

The initialization have been made to the identity Kernel. During the first iterations, the algorithm tends to the right solution. For example, the result on figure 3 is obtained with 4000 iterations on the blurred Lena Picture. However, if we run the algorithm up to the convergence, we obtained an uniform kernel. We therefore have to constrain the problem by adding a prior on the kernel to avoid this trivial solution.

3 MYOPIC DECONVOLUTION

As seen in the previous section , the blind deconvolution is quite complex and the convergence is not obtained. We now introduce a prior knowledge on the convolution kernel K. In this section, we develop an algorithm dealing with a parametric model of the kernel (which represents a Gaussian blur in our case).

3.1 Gaussian Kernel

Modelling As we supposed we have a Gaussian Blur, the kernel can be written as :

$$K_{i,j} = \frac{1}{Z} \cdot e^{-\left(k \times \left((i-c)^2+(j-c)^2\right)\right)} \quad (14)$$

Where :

$$\begin{cases} c = \frac{dimH-1}{2} \text{ is the coordinate of the center} \\ \frac{1}{Z} \text{ is a normalization term} \\ k \text{ is twice the inverse of the variance} \end{cases}$$

Here the prior knowledge, is not introduced in the energy function, like for the image, but in the fact that we now only consider a single parameter k and that we impose a Gaussian shape for the kernel.

This model has the following advantage: we have an unique unknown, $k = \frac{1}{\sigma^2}$, to estimate instead of the 49 unknowns in the blind deconvolution.

We have to calculate the new derivative of the energy function. The energy has not changed: $H(X,Y,K) = \lambda_1 \cdot \varphi_1(X) + \lambda_2 \cdot \phi_2(X,K,Y)$ where now:

$$\begin{aligned} \phi_2(X,K,Y) &= \sum_{i,j}(K * X_{i,j} - Y_{i,j})^2 \quad (15) \\ &= \sum_{i,j}\left(\sum_{(a,b)} k_{a,b} \cdot X_{i+a-N,j+b-N} - Y_{i,j}\right)^2 \\ &= \sum_{i,j}\left(\sum_{(a,b)} \frac{1}{Z} \cdot e^{-k\{(a-N)^2+(b-N)^2\}} \right. \\ &\qquad \left. \cdot X_{i+a-N,j+b-N} - Y_{i,j}\right)^2 \end{aligned}$$

And where : $N = \frac{dimK-1}{2}$. Let us then write the derivative of ϕ_2 :

$$\begin{aligned} \frac{\partial \phi_2}{\partial k_s} &= 2 \cdot \sum_{i,j}\left[\left(\sum_{a,b}\left(\frac{1}{Z} \cdot e^{-kA} \cdot X_{i+a-N,j+b-N} - Y_{i,j}\right)\right)\right. \\ &\quad \left. \times \left(\sum_{a,b} \frac{-1}{Z} \cdot A \cdot e^{-kA} \cdot X_{i+a-N,j+b-N} - Y_{i,j}\right)\right] \quad (16) \end{aligned}$$

where $A = (a-N)^2 + (b-N)^2$.
which can be written as:

$$\begin{aligned} \frac{\partial \phi_2}{\partial k_s} &= 2 \cdot \sum_{i,j}\left[\left(K * X(i,j) - Y_{i,j}\right) \cdot \right. \quad (17) \\ &\quad \left. \left(\sum_{a,b}\left(\frac{-1}{Z} \cdot A \cdot e^{-kA} \cdot X_{i+a-N,j+b-N} - Y_{i,j}\right)\right)\right] \end{aligned}$$

Here we suppose that the factor $\frac{1}{Z}$ does not depend on k. This is a first approximation because in fact: $\frac{1}{Z} = \sqrt{\frac{k}{2 \cdot \pi}}$

Results The considered kernel is the following:

$$K = \begin{vmatrix} 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0.02 & 0.01 & 0 & 0 \\ 0 & 0.01 & 0.05 & 0.1 & 0.05 & 0.01 & 0 \\ 0.01 & 0.02 & 0.1 & 0.2 & 0.1 & 0.02 & 0.01 \\ 0 & 0.01 & 0.05 & 0.1 & 0.05 & 0.01 & 0 \\ 0 & 0 & 0.01 & 0.02 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \end{vmatrix}$$

It is not exactly a gaussian Kernel. The following matrix is the gaussian matrix obtained for $k = 1.6$:

$$K = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0.009 & 0.016 & 0.009 & 0.001 & 0 \\ 0 & 0.009 & 0.057 & 0.106 & 0.057 & 0.009 & 0 \\ 0 & 0.016 & 0.106 & 0.198 & 0.0106 & 0.016 & 0 \\ 0 & 0.009 & 0.057 & 0.106 & 0.057 & 0.009 & 0 \\ 0 & 0.001 & 0.009 & 0.016 & 0.009 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

To face again the problem of huge derivative of ϕ_2 , as seen above, we have introduced a dividing factor, but also we constrain k to belong to the interval $[0.1;3]$.

Results of the simulations: Despite of these measures, we still have problem with the ϕ_2 derivative:

Through this Gaussian form, we have introduced a sum of exponential terms. This sum is very 'reactive' because of the exponential. The behavior of the algoritm is the following:

- If k is too big, then the exponential is very small, and then the sum tends to 0 and the derivative of ϕ_2 tends to 0 also. And if we first consider a simple gradient descent :$K_i(n+1) = K_i(n) + b_i(X(n),K(n),Y) \cdot \delta_n$, then $bi = 0$ and K does not move.
- If now k is too small, then the sum is too big and then bi is very big and we have a big jump for the derivative. And we face to the precedent case where k is too big.

The problem with this derivative is that the scale value is very large. So we first have to introduce a dividing factor before the derivative, to prevent infinite values for the variation of k. But then the derivative comes down. So if we keep this coefficient, it will not move. We can make it change at each iteration, but again, how to make it change ? A simple scheme like for the temperature descent is not good because the

derivative suddenly goes down and it is quite unpredictable.
So here in this modelling, we have only one parameter to estimate, but we have had another difficulty with the introduction of the exponential.

In the next step, we try to avoid the difficulty of the calculation of the derivative of the ϕ_2 function, since it is our main problem in this section. So next, we consider a simple Metropolis scheme for the calculation of the Kernel, which does not involve the calculation of this derivative.

3.2 A Metropolis Scheme for the Kernel Estimation

3.2.1 Algorithm

Here we propose to test a Metropolis scheme for the kernel so that we skip the problem of the derivative of the ϕ_2 function. We may assume that the minimization of the energy w.r.t K is faster than the optimization w.r.t X. So considering this fact, we may assume that here a Metropolis scheme for Kcan give good results and should not make the algorithm lost its advantage of speed because we only have one variable to estimate and besides the search space for K is not that large.

The algorithm is the following:

For each iteration IT :

- We randomly modify the configuration of the current kernel K to obtain a new state K' belonging to our search space.
- We calculate the energy associated to this new state, which is $H(X,K')$
- We compare $H(X,K')$ and $H(X,K)$: if $H(X,K') < H(X,K)$ then our new current kernel is K' else if $H(X,K') > H(X,K)$, then we use a Boltzmann acceptance criteria to decide whether we accept this new state K' or not. The acceptance probability depends on the temperature T of the system : $p = e^{\frac{H(X,K)-H(X,K')}{T}}$
- We finally decrease the temperature T of the system

Here we propose to test the following scheme:

1. Langevin scheme for X
2. Metropolis scheme for K and the Gaussian form for K defined previously

3.2.2 Results

Herein, we use the following Gaussian form of section 3.3.1 for the kernel K : $K_{i,j} = \frac{1}{Z} \cdot e^{-k\cdot((i-c)^2+(j-c)^2)}$

Our search space for k is the interval $[0,4]$ with a precision of 2 decimals.

So the following simulations have been done:

The data is the blurred image without noise. We start the simulation with pure noise. Then :

- *PhaseI* : pre-treatment . We run $300 < N < 1000$ iterations for the X scheme
- *PhaseII* : we run n times the following cycle :
  1. $m = 1500$ iterations for K
  2. $p = 200$ iterations for the X scheme

The initialization of the kernel is very important. We cannot start with the identity kernel because this is a trivial solution of our optimization problem for the energy : $\phi_2(X,K,Y) = \sum_i (H * X(i) - Y_i)^2$

Y is the data image and X is the current image. If we initialize with the identity kernel, then after the pre-treatment (*phaseI*), the current image X, is denoised, but blurred. So at the beginning of the *phaseII*, X is close to the data Y with less noise. But the important fact is that the blur in these two images is the same. So that optimizing the energy w.r.t. k has the identity kernel as a trivial solution since the above sum can be minimized with it.
How initialize the convolution Kernel knowing that there is a stability with the identity Kernel ?

In practice, we have initialized it very close from the identity. By that way, the small difference to the identity Kernel let us avoid the previous problem. And also the proximity to the identity does not degradate the image as a strong convolution Kernel would have done.

We finally obtained interesting results with this algorithm. Results on the synthetic image with the same parameters of noise and blur as for the tests in section 3.1 are shown on figure 5.
The edges are recovered well. The result is as good as the image obtained in section 3.1 with the known kernel.

The result image is satisfying. We have recovered the edges without introducing artifacts. The result is as good as the one we obtained in section 3.1 (known kernel).

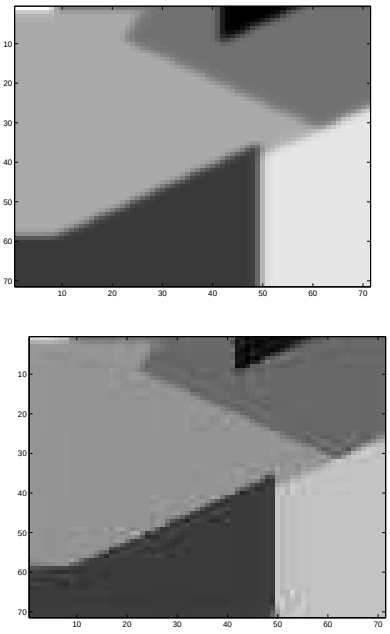

Figure 4: Blurred image top and Recovered image (bottom).

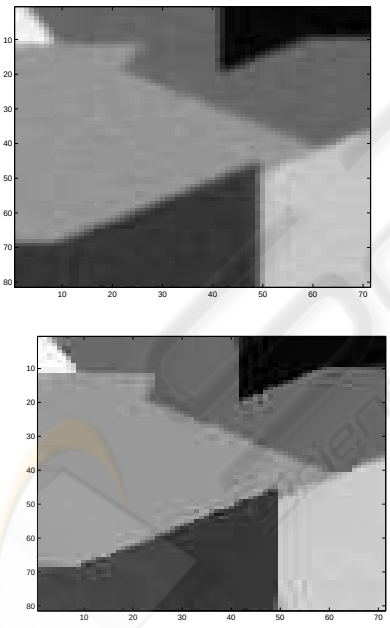

Figure 5: Blurred and noisy ($\sigma = 3$) image (top) and Result (bottom).

4 CONCLUSION

We have shown that embedding the deconvolution problem into a stochastic framework allows to build solutions which avoid the local minima of the functionnal, which is defined by a MRF modeling in our case. The stochastic differential equation framework appears to be a good alternative to the classical Gibbs sampler approach. When the kernel is known, we have obtained satisfactory solutions. However, faster algorithms such as inverse filtering lead to similar results and require less computational time (Andrews and Hunt, 1977). In the case of blind deconvolution, we have shown that the problem is not enough constraint to exibit a unique solution. Therefore, some prior on the kernel has to be considered. When using a parametric model for the kernel, we obtained satisfactory results by comnsidering a mixutre between the proposed Langevin dynamics and the Metropolis algorithm. This case of myopic deconvolution is the main motivation for using the proposed approach. Future work will consist of parameter estimation.

REFERENCES

Andrews, H. and Hunt, B. (1977). *Digital image restoration*. Prentice Hall, Englewood Cliffs, NJ.

Descombes, X. and Zhizhina, E. (2004). Applications of gibbs fields methods to image processing problems. *Problems of Information Transmission*, 40(3):279–295.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayeian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Geman, S. and Reynolds, G. (1992). Constrained restoration and recovery of discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(3):367–383.

Kloeden, P. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer Verlag, Berlin, Heidelberg.

Mignotte, M. (2006). A segmentation-based regularization term for image restoration. *IEEE Trans. on Image Processing*, 15(7):1973–1984.

Molina, R., Katsaggelos, A., Mateos, J., Hermoso, A., and Segall, A. (2000). Restoration of severely blurred high range images using stochastic and deterministic relaxation algorithms in compound Gauss-Markov random fields. *Pattern Recognition*, 33(4):555–571.