MARKOVIAN-BASED MODELING ON PROGRAMMABLE RETINA

Antoine Manzanera* Françoise Prêteux Thierry M. Bernard

ENSTA/LEI
32 Bd Victor
75015 Paris

INT/UP Artemis
9 Rue Charles Fourier
91011 Evry Cedex

ENSTA/LEI
32 Bd Victor
75015 Paris

***Abstract* - This paper addresses the implementation of a Markovian model on a programmable digital retina. The original contribution is to adapt the coding of the local energy in order to get compact implementations of deterministic (Iterated Conditional Mode), and stochastic (Metropolis Dynamics) relaxation algorithms for energy minimization. The logical conciseness of the procedures, together with the fine-grained parallelism of the retinal architecture, makes possible the real-time computation of the simulated annealing on a low-power vision system. The proposed model is applied to the segmentation of moving objects over a static background.**

1 Introduction

With the sustained improvement of VLSI technology, real-time artificial vision is becoming possible - little by little - on standard computers. However, with the present emergence of mobile applications, there is now a need for *very low power and compact* vision systems. Waiting some more years for even better technology will not allow to fulfil these strong requirements. Instead, we advocate the use of a *programmable artificial retina* (PAR), an architectural and technological concept presented in Section 2. Considering the case of *motion detection*, we show how the PAR concept allows, today, the realization of compact, long-autonomy remote surveillance systems.

Markovian modeling, the principles of which are summarized in Section 3, is a widely used framework in image analysis. However, Markovian methods are generally considered as heavy from a computational point of view, especially when dealing with stochastic versions, and thus not so well-adapted to low power operation... Nevertheless, the Markovian model proposed in Section 4 leads to a simple, and extremely fast deterministic relaxation. Furthermore, it allows the real-time computation of the *stochastic simulated annealing* (Section 5). Section 6 shows some segmentation results illustrating the relevance of the approach. Finally the expected performances of the detection system are discussed.

*This work was mainly done at the *Centre Technique d'Arcueil* (DGA/DCE) and the *ARTEMIS Project Unit* (INT Evry), with the financial support of *EADS Aérospatiale-Matra*

2 The programmable retina

The principle of an artificial retina, as part of a vision system, is to perform most of the image processing computations in the focal plane, where the images are sensed, thus dramatically reducing the flow of information toward external processing resources, which may then be devoted to high-level tasks only. The *programmable retina* is an image sensor array in which small digital processing elements are integrated near the photoreceptors, in all pixels. Such a circuit is adapted to real-time constrained image processing, thanks to its high degree of parallelism, as well as embarkable or autonomous systems, thanks to its performances in terms of compactness and low-power [14].

The algorithmic principles of the programmable retina are those of an SIMD machine, with an important originality: the small amount of local memory available at the pixel level because of the integration constraints. The processed objects are binary images (which may represent the grey levels of one single image). The only hard-wired operations on these data are: (1) One-pixel translation of one image. (2) Boolean operations between two binary images (OR, AND, XOR...).

Two medium size circuits of that kind have already been designed, fabricated and successfully operated: one 65×76 array of three-bits pixels in 2μ CMOS process [2], and one 128×128 array of five-bits pixels in 0.8μ [13]. The next generation (probably 0.25μ), is expected to reach better resolution and about 20 bits per pixel. We have shown [10] that in spite of memory limitation, some complex image processing algorithms can be computed on such circuits. While analog circuits are usually considered as more efficient for the computation of spatio-temporal changes, the fully-digital framework of the programmable retina allows complex and versatile modeling for the regularization process. Furthermore, the algorithm proposed in this paper achieves a very fast, or very low-power, motion-based segmentation.

3 Markov random fields

The segmentation issue in image processing consists in attributing a label to every pixel according to a certain predicate. This predicate is generally related to a physical feature of an object (reflectance, motion...) that can be retrieved only partially from the 2D projected information gathered in the image. Thus segmentation is said to be "ill-posed" and this issue is addressed by adding some *a priori* knowledge about the underlying phenomenon.

Within the *Markovian framework*, such a knowledge is expressed as a topological structure specifying probabilistic dependency relations [11], and the solution to the problem is seen as the most probable realization of a random process. Modeling spatial (or spatio-temporal) local dependencies, *Markov random fields* (MRFs) are well-suited to image processing problems. The Hammersley-Clifford theorem [3], states the equivalence between Markov and Gibbs fields. Thus a Markovian model can also be defined by means of an *energy function* such that the most probable realization is the one that minimizes the energy function. Furthermore, samples of the specified random field can be obtained thanks to simulation algorithms by *Markov chains*.

MRFs have been used in image processing for the restoration of dirty pictures [7], for texture segmentation [6], and for motion detection [4]. Our model roughly follows the principles of the models given in the literature dedicated to motion segmentation by Markovian approaches [4], [5], [9]: starting from the differences between successive images of the video sequence (also called observations), the motion-based segmentation is estimated as the most probable realization of a binary Gibbs field (0: still, 1: moving). The next section details the design of the Gibbs model.

4 Energy function design and the deterministic algorithm

The Gibbs field is a random field whose probability is a Gibbs (or Boltzmann) measure. Let $\mathbb{S}$ denote the set of sites, $\mathbb{V}$ the set of possible labels assigned to each site. Here, $\mathbb{E} = \mathbb{V}^{\mathbb{S}}$ is the set of all the possible realizations of a random field X. X is a Gibbs field if there exists a function U (called *energy* function) from $\mathbb{E}$ to $\mathbb{R}$ such that:

$$P(X = x) = \frac{e^{-U(x)}}{Z}.$$

where $Z = \sum_{y \in \mathbb{E}} e^{-U(y)}$ is the normalizing factor. The Gibbs field is associated with a *topology* over $\mathbb{S}$. Indeed, the energy U is expressed as:

$$U(x) = \sum_{s \in \mathbb{S}} U_s(x),$$

where $U_s(x)$ is the local energy on site s, which only depends on the values of x over cliques containing s.

Dealing with binary segmentation, corresponding to bi-valued fields, the set $\mathbb{V}$ is $\{0, 1\}$. In the proposed model, the energy is designed using fixed potentials. This is equivalent to defining a (regular) topology and then attributing a (translation-invariant) weight to each clique. The contribution of every site (pixel) s to the energy of the realized field is the sum, over all the cliques containing s, of $\pm\beta_c$, where the value of β_c depends on the nature of the clique c, and the sign is " $-$ " if the values of the sites inside the clique are the same, " $+$ " otherwise. For example, if we consider only cliques of order 2, $U_s(x)$ is given by:

$$\begin{aligned} U_s(x) &= \sum_{r \in \mathcal{V}(s)} V_x(s, r), \text{ with:} \\ V_x(s, r) &= -\beta_{sr} \text{ if } x(s) = x(r), \\ &= +\beta_{sr} \text{ if } x(s) \neq x(r). \end{aligned}$$

Where $\mathcal{V}(s)$ denotes the set of all neighbors of s in the associated topology, made up of various types of links:

- *spatial* links modeling shape regularity.
- *temporal* links modeling motion consistency.
- one special link modeling *attachment to the input data* and allowing certain discontinuities.

This model can also be interpreted in terms of the Bayesian estimation of the *maximum a posteriori* [4], involving one term of (spatio-temporal) regularization and one data-driven term.

On the programmable retina, the dynamics must be carefully sized because of the small amount of memory. The algorithm whose presentation follows is designed to be the least memory consuming possible. The frame difference is computed in a destructive way, which means that the bits used to code $|I_{t+1} - I_t|$ are those which coded I_t. The difference is thresholded, and the resulting binary image is used as: (1) the initial value of the segmentation in the relaxation process, (2) the value of the observation which represents the data-driven term in the Gibbs model. Thus the computation of the energy involves 4 binary images: (1) the threshold difference Δ_t, (2) the estimated segmentation $\hat{E}_t$, (3) the computed segmentation of the previous frame E_{t-1}, (4) the threshold difference of the next frames Δ_{t+1}.

The energy representation consists in coding at every pixel s, the number $W_s(x)$, defined as follows:

$$\begin{aligned} W_s(x) &= \sum_{r \in \mathcal{V}(s)} W_x(s,r), \text{ with:} \\ W_x(s,r) &= \beta_{sr} \quad \text{if } x(r) = 1, \\ &= 0 \quad\;\; \text{if } x(r) = 0. \end{aligned}$$

Thus, we simply compute the weighted sum of the bits in the neighborhood of s. This number provides the local energy for both values 1 and 0. More precisely, for $e \in \mathbb{V}$, let $u_x(s,e)$ denote the local energy in s associated with label e such that:

$$u_x(s,e) = U_s(x) \Leftrightarrow x(s) = e.$$

If the sum of all weights (that is constant over $\mathbb{S}$) is $S = \sum_{r \in \mathcal{V}(s)} \beta_{sr}$, then the local energy associated with each label is given by:

$$u_x(s,0) = -u_x(s,1) = 2W_s(x) - S$$

If the weights are normalized such that $S = 2^n - 1$, then we get:

$$u_x(s,1) < u_x(s,0) \Leftrightarrow W_s(x) \geq 2^{n-1}.$$

This implies that the most significant bit (MSB) of $W_s(x)$ indicates the label which leads to the (locally) lowest energy. Finally the computation of $W_s(x)$ directly provides the deterministic algorithm called *iterated conditional mode* (ICM), that consists in initializing the label by $\hat{E}_t = \Delta_t$, then replacing iteratively $\hat{E}_t$ by the MSB of $W_s(x)$, until convergence. Figure 1 shows an example of such a computation procedure. Let us recall that the binary operations are the only hard-wired operations on the PAR. This diagram is not a hardware implementation, but represents all the operations (softly) performed per pixel. The size of the diagram (number of binary gates) indicates the computation time.

The deterministic algorithm is based upon the assumption that the initialization is a confident instanciation of the modeled Gibbs field. It may give satisfying results if the thresholded difference is close enough to the desired segmentation, as it converges to the closest local minimun of the energy function. To converge to the global minimum, we have to consider stochastic relaxation procedures.

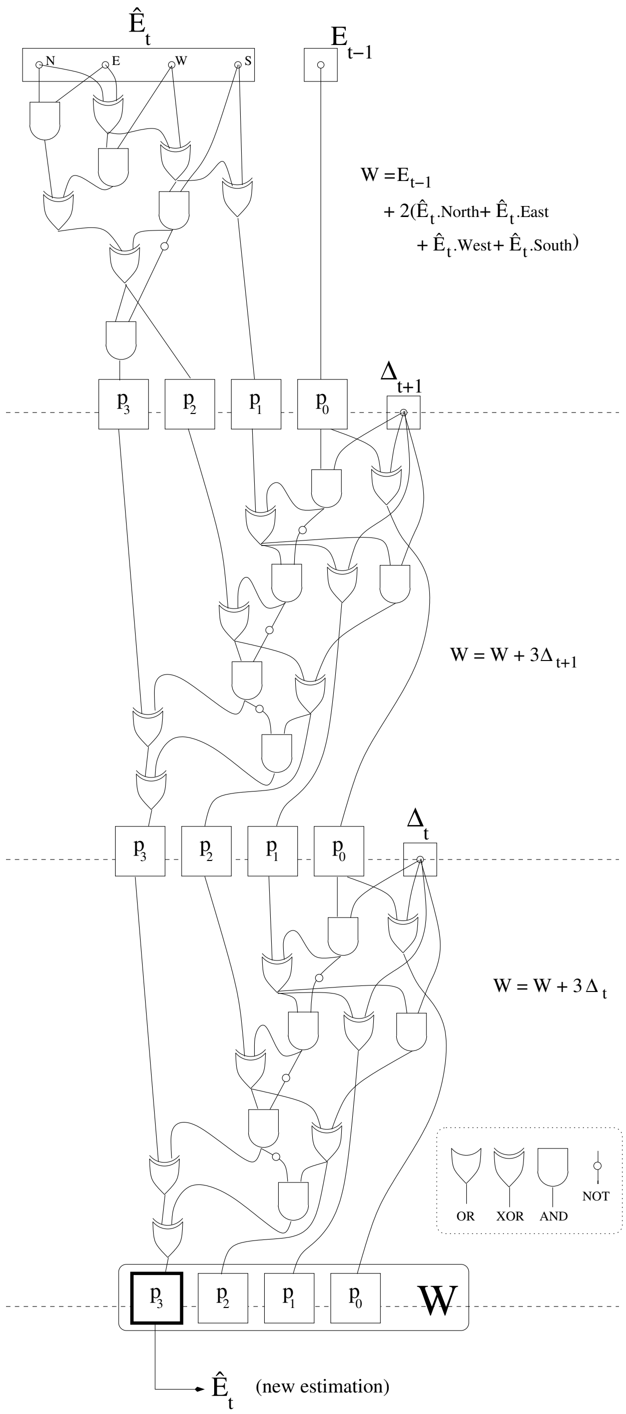

Figure 1: Computation of the energy and deterministic relaxation ICM. In this example, the spatial neighbors are limited to the 4-adjacent pixels. The weights attributed to spatial, past, and future links and to the initialization are respectively 2, 1, 3 and 3. Thus the energy is coded on 4 bits, and the MSB of W provides the label leading to the lowest local energy.

5 Markov chains and the stochastic algorithm

The first step toward the non-deterministic relaxation is to get samples of random fields corresponding to the model we have defined. This is achieved by constructing a Markov chain of random fields whose distribution converges to the defined Gibbs measure. Such a construction can be performed by using a simulation procedure like the Metropolis algorithm [12], which can be described as follows:

Let $X_0 = x_0$ be whatever initial state. For $n \geq 0$, the following procedure is applied:

1. Choose a random site s (even probability on $\mathbb{S}$).
2. Choose a random label e (even probability on $\mathbb{V}$). Let $\tilde{x} \in \mathbb{E}$ such that $\tilde{x}(r) = x_n(r)$ if $r \neq s$ and $\tilde{x}(s) = e$.
3. Compute the difference $\Delta U = U(\tilde{x}) - U(x_n)$.
   If $\Delta U < 0$, then: $x_{n+1} = \tilde{x}$.
   else: $x_{n+1} = \tilde{x}$ with probability $e^{-\Delta U}$
   $x_{n+1} = x_n$ with probability $1 - e^{-\Delta U}$

If this algorithm is provided with a parameter T (temperature) such that ΔU is replaced by $\frac{\Delta U}{T}$, then the sequence of random fields converges in distribution to the Gibbs measure of energy $\frac{U(x)}{T}$. The principle of the *simulated annealing* (SA) [8] is to use such a parameter, and to let it go to zero, in such a way that the corresponding Gibbs measure tends to the uniform probability on the global minima of the energy function U.

On the programmable retina, a ring of three cascaded invertors is used as oscillator (with a frequency of several GHz) to generate a random bit with even distribution at the cost of one elementary instruction.

The first part of the stochastic algorithm is strictly identical to the deterministic one: the number $W_s(x)$ is computed on each site s. Remember that the MSB of $W_s(x)$ indicates the lowest local energy label. Furthermore, if $W_s(x)$ is coded on n bits, we get:

$$|W_s(x) - 2^{n-1}| \simeq \frac{|\Delta U|}{4},$$

where ΔU is the difference of energy obtained by changing the label, used in the Metropolis algorithm. Now note that if $W_s(x)$ is coded on the n bits $\{w_i\}_{0 \leq i \leq n-1}$, $|W_s(x) - 2^{n-1}|$ is coded on the $(n-1)$ bits $\{d_i\}_{0 \leq i \leq n-2}$, such that:

$$\forall i, 0 \leq i \leq n-2, d_i = w_i \oplus \neg w_{n-1},$$

where $\oplus$ and $\neg$ denote respectively the logical XOR and NOT (top of Figure 2).

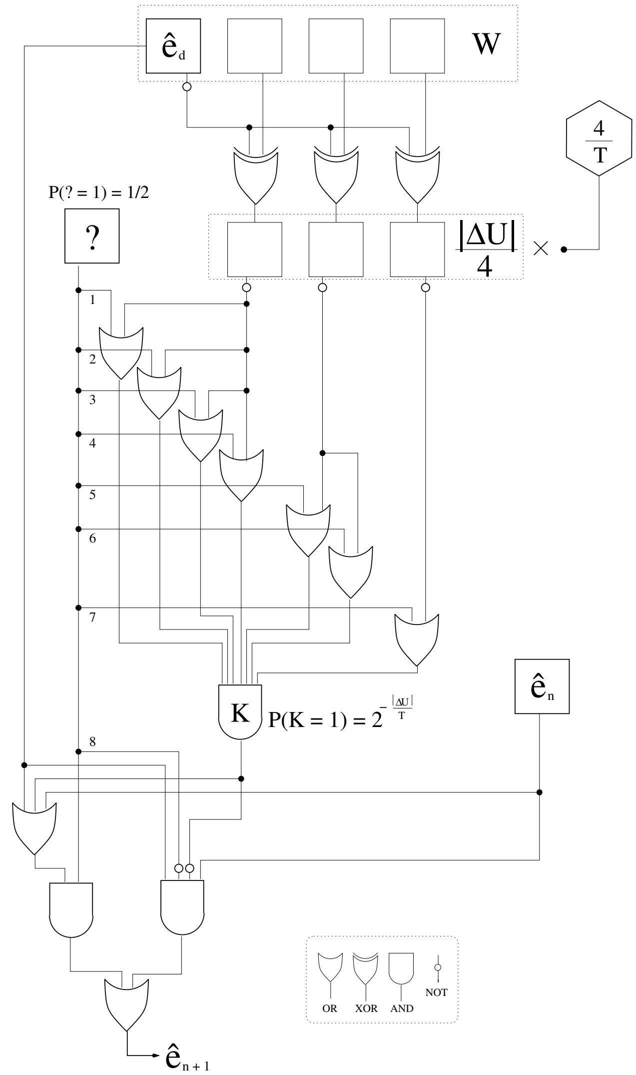

Figure 2: Stochastic relaxation: the Metropolis algorithm. (Energy coded on 4 bits). The input W corresponds to the output of the deterministic part. For clearness purposes, multiple gates have been represented in this diagram, so a gate with p entries represent $p-1$ binary operations. The nodes numbered from 1 to 8 represent different tosses of the random bit.

Then, if a number X is coded on the n bits $\{x_i\}_{0 \le i \le n-1}$ in every pixel, and $\{r_i\}_{1 \le i \le 2^n - 1}$ is a sequence of $(2^n - 1)$ random bits, the binary random variable K, defined as:

$$K = \bigwedge_{i=0}^{n-1} \left(\bigwedge_{j=0}^{2^i - 1} r_{2^i + j} \vee \neg x_i \right),$$

(where $\wedge$ and $\vee$ denote respectively the logical AND and OR) is equal to 1 with probability 2^{-X} (middle of Figure 2).

We now put everything together. Considering a given pixel: let $\hat{e}_n$ be the estimated label at iteration n. $W_s(x)$ is computed, its MSB, denoted by $\hat{e}_d$ is the deterministic label. On its other bits, we compute $\frac{|\Delta U|}{4}$. The external parameter T steps in, as we multiply $\frac{|\Delta U|}{4}$ by $\frac{4}{T}$ (addition/shifting). Then the random bit K is computed, such that $P(K = 1) = 2^{-\frac{|\Delta U|}{T}}$. Independently, a random bit B is generated, such that $P(B = 1) = \frac{1}{2}$. Finally the next estimation $\hat{e}_{n+1}$ is computed by:

$$\hat{e}_{n+1} = ((\hat{e}_d \vee K \vee \hat{e}_n) \wedge B) \vee (\hat{e}_d \wedge \neg B \wedge \neg K \wedge \hat{e}_n).$$

This formula is interpreted as follows: B is the random label, if $B = \hat{e}_d$ then $\Delta U < 0$, and then $\hat{e}_{n+1} = B = \hat{e}_d$. Otherwise, the result depends on K: if $K = 1$ then $\hat{e}_{n+1} = B$, else $\hat{e}_{n+1} = \hat{e}_n$ (bottom of Figure 2). This completes the computation of the simulated annealing by the Metropolis algorithm.

Before showing some results, we have to mention the issue of *synchronous* parallel computing. Indeed, the theoretical bases of the Markovian methods lie on *asynchronous* Monte-Carlo updating of the sites. But the validity of some synchronous techniques have been proved [1]. On the PAR, different updating techniques can be implanted: fully parallel, periodic semi-parallel and stochastic inhibition. For all these methods, however, our experiments in the case of motion segmentation did not provide significantly different results...

6 Results and concluding remarks

Results of motion-based segmentation on a video sequence are shown on Figure 3, on the deterministic and stochastic frameworks.

We now evaluate the proposed algorithm in terms of memory occupation and computation time.

Thanks to the fact that the frame difference is computed destructively and that all the images that take part of the energy computation are binary, the algorithm only needs 4 bits more than the memory used to code the images. The presented results, for example, can be obtained on a programmable retina with 12 bits per pixel.

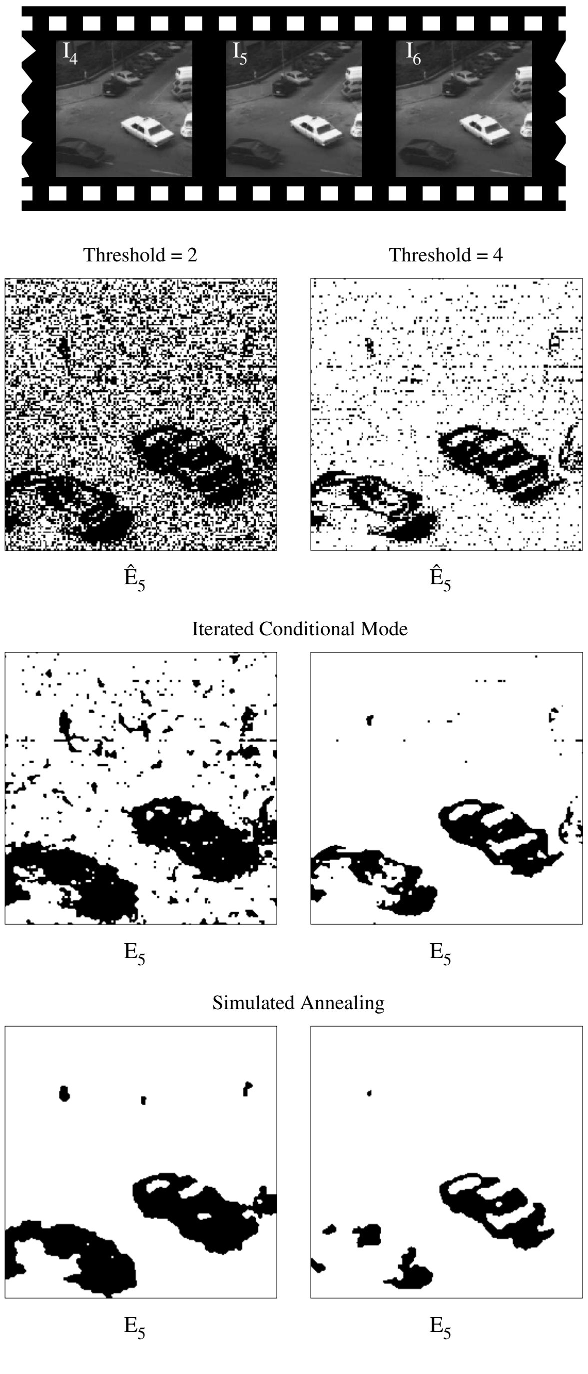

Figure 3: Segmentation results at some point of the Hamburg taxi sequence. Computation uses 12 bits of memory. spatial 8-connectivity, images coded on 8 bits, energy coded on 6 bits. The two columns correspond to two different thresholds. First line: the thresholded frame difference. Second line: ICM, 5 iterations. Third line: SA, temperature decreasing linearly, 50 iterations.

Finally we present the expected time performances. The computation time is proportional to the number of elementary operations performed per pixel. Let us now detail the computational cost of the different part of the algorithm. This cost depends on $S = \sum_{r \in \mathcal{V}(s)} \beta_{sr}$, which corresponds to the maximal value of W_s.

- **Computation of W_s:** in the worst case ($\forall r, \beta_{sr} = 1$), the number of Boolean operations is equal to $2S\log(S)$.
- **Metropolis algorithm:** $S + \log(S)$.
- **Multiplication by $\frac{4}{T}$:** in the worst case (all bits of $\frac{4}{T}$ are equal to 1), the number of Boolean operations is equal to $\frac{(\log(S))^3}{3} + \frac{(\log(S))^2}{2} + \log(S)$. The mean number of Boolean operations is equal to half this value.

For the presented results (Figure 3), the approximate number of operations for one iteration of relaxation is 200 in the deterministic case, and 320 in the stochastic case. The control frequence of the programmable retina depends on the power supply; it typically varies between 1 and 100 MHz. So for a given control frequence, a smaller number of operations allows to treat a greater number of images per second (i.e. to increase the acquisition frequence), and for a given acquisition frequence, a smaller number of operations allows to decrease the control frequence (thus reducing the power consumption). For example, in the algorithms presented Figure 3, at 10 MHz, the result is computed in $5 \times 200 \times 0.1 = 100\mu s$ for the ICM and $50 \times 320 \times 0.1 = 1.6ms$ for the SA.

These results show that the proposed algorithm makes possible high performance detection systems including a programmable retina. More generally, it provides a compact modeling of a Markovian framework allowing regularization by ultra-fast deterministic or real-time stochastic relaxation, that can be used in other contexts. The very small consumption of memory resources and computation time allows to consider more complex models with higher order topologies, as well as multi-label segmentation, that will be dealt in a future work.

References

[1] R. Azencott. *Simulated annealing: parallelization techniques.* John Whiley, 1992.

[2] T.M. Bernard, B.Y. Zavidovique, and F. Devos. A programmable Artificial Retina. *IEEE Journal of Solid-state Circuits*, 28-7:789–798, 1993.

[3] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of Royal Statistic Society*, B-36:192–326, 1974.

[4] P. Bouthémy and P. Lalande. Recovery of moving object in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32-6:1205–1212, 1993.

[5] A. Caplier, C. Dumontier, F. Luthon, and P.Y. Coulon. Algorithme de détection de mouvement par modélisation markovienne. Mise en œuvre sur DSP. *Traitement du signal*, 13-2:175–190, 1996.

[6] G.C. Cross and A.K. Jain. Markov random field texture models. *IEEE Transactions on pattern analysis and machine intelligence*, 5-1:25–39, 1983.

[7] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 6-6:721–741, 1984.

[8] S. Kirkpatrick, C.D. Jr Gellatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–690, 1983.

[9] L. Lacassagne, M. Milgram, and P. Garda. Motion detection, labeling, data association and tracking in real-time on risc computer. In *International Conference on Image Analysis and Processing*, pages 520–525, Venice - Italy, September 1999. IEEE.

[10] A. Manzanera. *Vision artificielle rétinienne.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, November 2000.

[11] J.L. Marroquín. *Probabilistic solution of inverse problems.* PhD thesis, Massachusetts Institute of Technology, September 1985.

[12] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and Teller E. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[13] F. Paillet, D. Mercier, and T.M. Bernard. Second generation programmable artificial retina. In *Proc. IEEE ASIC/SOC Conf.*, pages 304–309, September 1999.

[14] F. Paillet, D.S. Mercier, T.M. Bernard, and E. Senn. Low power issues in a digital programmable artificial retina. In *Proc. IEEE Workshop on Low Power Design*, pages 153–161, Como, Italy, April 1999.