A Cellular Analog Network for MRF-based Video Motion Detection
Franck Luthon, Daniela Dragomirescu

To cite this version:
Franck Luthon, Daniela Dragomirescu. A Cellular Analog Network for MRF-based Video Motion Detection. IEEE Transactions on Circuits and Systems Part 1 Fundamental Theory and Applications, Institute of Electrical and Electronics Engineers (IEEE), 1999, 46 (2), pp.281-293. <hal-00408662>

HAL Id: hal-00408662
https://hal.archives-ouvertes.fr/hal-00408662
Submitted on 31 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Cellular Analog Network for MRF-based Video Motion Detection
Franck Luthon and Daniela Dragonirescu, Student Member, IEEE

Abstract—The implementation of a visual motion detection algorithm on an analog network is presented. The algorithm is based on Markov random field (MRF) modeling. Robust motion detection is achieved by using a spatiotemporal neighborhood for modeling pixel interactions. Not only are the moving edges detected, but also the inner part of moving regions. Moreover, the motion echo is eliminated. For hardware implementation, the algorithm is mapped onto a resistive network incorporating, at each pixel-node, light acquisition and on-chip processing. Elementary Markov cells are made of resistors and non-linear current sources driven by observations, derived from luminance information acquired by photoreceptors. Electrical simulations of the cell are reported. Experimental results with synthetic and real-world image sequences exhibit the performance of the network. For circuit design, the switched-current technique is used. The circuit is intended for submicron complementary metal-oxide-semiconductor (CMOS) technology.

Keywords—Image sequences, Markov random fields (MRF's), motion detection, motion chip, smart sensor.

I. INTRODUCTION

THE design of smart sensors for visual motion analysis is of interest for the following applications: telesurveillance, videoconferencing, videophone, compression, and computer vision. Motion analysis in image sequences mainly consists of four stages [1]: detection of temporal changes; estimation of optical flow or velocity; segmentation of different moving regions; and interpretation of the scene content. Here, we focus on robust motion detection. By robust motion detection we mean a process in between detection and segmentation of moving areas, i.e., obtaining of a binary field accurately representing the projection (or mask) of moving objects in the image plane. Assuming a static camera and little variation of the scene lighting, motion information is derived from temporal differences in the intensity function I. To recover complete masks of moving objects, this basic information must be processed in order to reduce the influence of acquisition noise, to cancel the echo of motion, and to fill in the interior of masks (Fig. 1).

For that purpose, a regularization technique is used, based on Markov random field (MRF) modeling and energy minimization via the iterated conditional modes (ICM) relaxation algorithm. But computation cost is high and software implementation is incompatible with real-time processing.

Hardware implementation on special-purpose vision circuits (ASIC's) gives the best computation and communication efficiency. The main advantages of analog ASIC's are processing speed, low power consumption, small hardware size, and high pixel density, while the drawbacks are limited computational precision and flexibility, development times and financial cost. Two approaches exist for the design of smart sensors [2]: computational direction and biological models. Here, a computational approach is adopted. Interestingly, however, starting from a mathematical viewpoint (i.e., statistical regularization based on MRF) and mapping the algorithm onto an analog network results in a circuit which resembles, in many respects, a biologically inspired cellular neural nonlinear network (CNN) [3].

A brief survey of the literature reveals some interesting points. First, most approaches for motion-chip design are biologically inspired. They use either gradient based schemes or correlation schemes to compute global image motion [4], [5], [6]. Second, from the application viewpoint, a clear distinction should be made between motion detection, estimation, or segmentation and between static and mobile camera. Moreover, robustness with respect to noise and to the number, size, and speed of mobile objects is of major concern. To our knowledge, no chip exists that recovers the complete masks of moving regions. Most circuits either detect moving edges only (motion transitions) or estimate the global velocity in the image plane. Many so-called motion chips are based on the detection on excited cells of positive and negative pulses in temporal derivatives. They actually compute basic temporal differences [7]. A few circuits report sparse or dense optical flow, but no segmentation is available (cf. optical flow computation chip with biological approach [8]). The first instance of a robust
optical flow computation chip is presented in [2]. A motion
detection-and-estimation algorithm that is an amalgamation
of biologically and computationally based models is
implemented on a smart sensor. Third, many authors put
emphasis on the effectiveness of adopting a current-mode
approach (i.e., representing signals as currents instead of
voltages) for the design of building blocks in analog very
large scale integration (VLSI) neural systems, although
they do not agree about the best MOS transistor operat-
ing mode: the subthreshold region (translinear circuits)
[6], linear region [9], or saturation region.

The context in which we place our work is restricted to
robust motion detection (i.e., recovery of moving masks
without velocity estimation) in the case of a static camera.
The algorithm is based on a computational (nonbiological)
method, and an analog implementation is presented
which uses a current-mode approach. The paper is
organized as follows. Section II presents the mathematical
model used for robust motion detection and the electrical
analogy found for mapping the algorithm onto an analog
network. Section III deals with circuit architecture. In
section IV, the circuit design for VLSI implementation is
detailed. Electrical simulations exhibit the cell behavior. In
section V, the network is simulated and motion detection
results with both synthetic and real-world image sequences
are reported.

II. MRF FRAMEWORK WITH ELECTRICAL MODELING

A. Observations, Labels, and Neighborhood Structure

Motion detection aims at labeling each pixel of a video
sequence according to moving and static areas in con-
secutive images. Usual assumptions are a static camera
and almost constant illumination of the scene. Thus,
the raw observation \( o_s \) at pixel (or site) \( s = (i,j,t) \)
is given by temporal changes\(^1\) in the intensity function
\( o_s = |I(i,j,t) - I(i,j,t-1)| \) where \((i,j)\) are the discrete
spatial coordinates of pixel \( s \) and \( t \) is the discrete time instant.
The motion label \( l_s \) to be given to site \( s \) may take
one of the two values

\[
l_s = \begin{cases} 
  \text{\# "1",} & \text{if } s \text{ is moving area,} \\
  \text{\# "0",} & \text{if } s \text{ is static background.}
\end{cases}
\]

(1)

Let \( l = \{l_s, s \in S\} \) (respectively \( o = \{o_s, s \in S\} \) be one
realization of label field \( L \) (respectively observation field \( O \))
on image \( S \) at time\(^2\) \( t \). The random field \( L \) is assumed to
have MRF properties, i.e., a Gibbs distribution and
a neighborhood structure. The spatiotemporal neighborhood
\( \eta \) of site \( s \) is given in Fig. 2.

According to [10], maximizing the a posteriori probability
of label field realization \( l \), given observation field
realization \( o \), is equivalent to minimizing an energy or cost
function \( U \) consisting of two terms \( U = U_m(l) + U_o(o,l) \).

\(^1\) The absolute value is taken in order to be insensitive to the sign
of contrast between mobile and static regions.

\(^2\) Whenever needed, a temporal subscript will be explicitly added to
the notations (e.g., \( o_{t+1} \) for observation field \( o \) at time \( t+1 \)).
motion detection, \( \Psi \) may be defined as 3

\[
\Psi(l_s) = \begin{cases} 
0 & \text{if } l_s = l_p = b \text{ (background)} \\
\alpha_1 > 0 & \text{if } l_s = l_p = m \text{ (motion)} \\
\alpha_2 >> \alpha_1 & \text{if } l_s \neq l_p \text{ (transition)}
\end{cases}
\]

where \( \alpha_1 \) and \( \alpha_2 \) are positive parameters that may be predefined or estimated online [11].

C. Software Energy Minimization via ICM Relaxation

The algorithm's flow-chart is given in Fig. 3. Because of the spatiotemporal neighborhood chosen for motion detection (Fig. 2), three label fields are required for relaxation to start: past, present and future. Therefore, the algorithm implies a one image delay since the future image at time \( t+1 \) is required to get the label field corresponding to time \( t \).

The minimum energy may be computed using either stochastic relaxation algorithms, like simulated annealing [10], or deterministic algorithms, such as iterated conditional modes (ICM) [12]. Here, the iterative method ICM is used because of its lower computation cost. But it is suboptimal in the sense that it may get stuck in the first encountered local minimum of the energy function. Therefore, label field initialization is crucial in order to be sure to converge to the global minimum. At time \( t \) past label field \( l_{t-1} \) is already available as the result of previous relaxation. However, present and future label fields must be initialized. This is done by binarization of observation fields \( o_1, o_2 \) (simple comparison to a threshold \( \theta_m \) or likelihood test methods), yielding initial field estimates \( \hat{P}_t, \hat{P}_{t+1} \). Then, ICM relaxation runs over field \( L \) and field \( L \) at time \( t \) is scanned pixel by pixel. At each site, a local decision is taken. The label given to a site is the one inducing the lower energy within its neighborhood. This relaxation iterates until convergence is achieved, based on a relative energy decrease criterion (typically \( \Delta U / U < 0.01 \% \)). The processing rate is typically of 1 image/s on a Sparc-10 workstation for images of size 128 \( \times \) 128.

Instead of software implementation via ICM for finding the global minimum solution, hardware implementation on an analog VLSI network is an efficient alternative in achieving real-time processing. The basic principle consists of allowing an electrical network to find the minimum energy configuration by relaxing to its state of minimal power dissipation.

D. Electrical Analog

The idea of mapping the algorithm on an analog network stems from a close study of the equation underlying the state of minimal energy. Let us use the following explicit notations, corresponding to current pixel \( s = (i,j,t) \): \( \alpha_2 = \alpha, l_{ij} = l_t, p_{ij} = l_p \) for past label and \( f_{ij} = l_f \) for future label. Then, denoting \( K = 1/2\sigma^2 \), the total energy \( U \) may be rewritten as

\[
U = \sum_{i,j} \beta_s(l_{ij} - l_{i+1,j})^2 + \beta_s(l_{ij} - l_{i-1,j})^2 + \beta_s(l_{ij} - l_{i,j+1})^2 + \beta_s(l_{ij} - l_{i,j-1})^2 + \beta_f(l_{ij} - f_{ij})^2 + \beta_p(l_{ij} - p_{ij})^2 + K [ \alpha_{ij} - \Psi(l_{ij}) ]^2 .
\]

For an analog implementation, we must suppose that labels \( l_{ij} \) may take continuous values \( \in [b, m] \) instead of binary values in the set \{b, m\}. The minimum of \( U \) with respect to all \( l_{ij} \) corresponds to null partial derivatives

\[
\nabla l_{ij} U = 0 \Leftrightarrow \beta_s \nabla^2 l_{ij} + \beta_f (l_{ij} - f_{ij}) + \beta_p (l_{ij} - p_{ij}) - K \frac{\partial \Psi}{\partial l_{ij}} (\alpha_{ij} - \Psi(l_{ij})) = 0.
\]

6To recover the final binary labels relevant for motion detection, a simple comparison to a threshold \( \theta_{def} \) applied to these continuous labels is required.
represent conductances. The last term in (6) is a current source depending on observation $a_{ij}$ and on attachment function $\Psi$. Its role is similar to the attachment energy $U_a(a,l)$ (i.e., observation fit). Moreover, it should be easily implemented on an analog circuit (which means that a linear function of $l_{ij}$ is required for $\Psi$). Bearing these constraints in mind, and inspired by (4), there are many possible candidates for $\Psi$. The simplest form would be $\Psi(l_{ij}) = A l_{ij}$ with $A$ constant. However, to properly cancel motion echo, the following expression is proposed: $\Psi(l_{ij}) = A (l_{ij} - p_{ij})$, where $A_{ij} = \alpha_2(p_{ij} - f_{ij}) - \alpha_1$ with $\alpha_2 > \alpha_1 > 0$. Replacing $\Psi$ and its derivative in (6), and introducing a leakage capacitance specific of the circuit dynamics yields, for all $ij$,

$$C \frac{d l_{ij}}{dt} = \beta_s v^2 l_{ij} + \beta_f (l_{ij} - f_{ij}) + \beta_a (l_{ij} - p_{ij}) - KA_{ij} a_{ij}$$

(7)

with $\beta_a = \beta_s + KA_{ij}^2$. At electrical equilibrium, the left member of (7) tends to zero, corresponding to the minimum energy solution. Thus, this linear system of differential equations may be solved via an analog resistive network as implemented below.

III. CIRCUIT ARCHITECTURE

A. Elementary Markov Cell

The cell shown in Fig. 4 contains six resistors (four spatial resistors $R_s$, two temporal resistors $R_f, R_p$) and two voltage-controlled current sources $J_1$ and $J_2$. The four resistors $R_s$ are connected to the four nearest neighboring cells. Past and future neighbors are actually not physical but, rather, are virtual neighbors in the sense that they do not correspond to actual electrical cells as spatial neighbors do. They only represent side information entering the physical nodes. Theoretically, resistor $R_p$, at least, should be adjustable since it depends on $\sigma^2$ and on $ij$ past and future labels. In practical implementation, however, one can choose a fixed value since the temporal dependency of $R_p$ can be compensated with a proper choice of $\alpha_2$ and $\alpha_1$. Moreover, generator $J_2$ is connected to $V_{cc}/2$ in order to be able to generate current in both directions, depending on the sign of $(f_{ij} - p_{ij})$, while $J_1$ is connected to $V_{cc}$ since it always injects current into the node ($a_{ij}$ is always $\geq 0$).

The elementary cell of Fig. 4 is in charge of a local electrical processing at node $ij$. After all the required command voltages $(a_{ij}, f_{ij}, p_{ij})$ are set up onto the cell, one lets the cell relax. When there is no motion information ($a_{ij} = 0$), current sources are inactive and electrical potential $l_{ij}$ is only influenced by its neighbors (spatiotemporal smoothing due to resistors). When motion information is present ($a_{ij} \neq 0$), current sources are active and a current is injected into (respectively, extracted from) node $ij$, depending on the sign of $(f_{ij} - p_{ij})$, i.e., on the type of transition (static-to-mobile, mobile-to-static, etc.)

Current sources $J_1, J_2$ ensure the good fit to observed data. $J_1$ is responsible for recovering the inner part of moving masks, while $J_2$ handles transition areas (front and echo). Let $\Delta J$ be the sum of currents delivered by the two sources. Four typical cases may occur, corresponding, respectively, to background, interior, echo, and front areas (Fig. 1).

1. **Static-to-static pixel**: $a_{ij} = 0$ and $p_{ij} = f_{ij} = 0$. Then current sources do not work: $\Delta J = 0$.

2. **Mobile-to-mobile pixel**: $a_{ij}$ small and $p_{ij} = 1$ - With adequate input threshold ($\theta_{in}$ low): $f_{ij} = 1$. Then generator $J_1$ only is active: $\Delta J = G_1 a_{ij} > 0$. A small current is injected into node $ij$, increasing slightly its electrical potential. This allows to fill in the inner part of moving masks.

- With inadequate input threshold ($\theta_{in}$ too high): $f_{ij} = 0$. Both generators are active: $\Delta J = (G_1 - G_2)a_{ij} < 0$ since $G_2 > G_1$. A small current is extracted from node $ij$, decreasing slightly its electrical potential. However, since past neighbor is labeled as mobile and observation is small, this influence is not critical and does not break mask homogeneity.

3. **Mobile-to-static transition**: $a_{ij}$ large and $p_{ij} = 1, f_{ij} = 0$. $\Delta J = (G_1 - G_2)a_{ij} \ll 0$. A large current is pumped out of the node, decreasing drastically its electrical potential. This allows to cancel the echo area.

4. **Static-to-mobile transition**: $a_{ij}$ large and $p_{ij} = 0, f_{ij} = 1$. $\Delta J = (G_1 + G_2)a_{ij} \gg 0$. A large current is pumped into the node, increasing drastically its electrical potential. This allows to enhance the moving front area.

When steady state is reached after relaxation, the electrical potential $l_{ij}$ is output to a comparator, giving a binary value corresponding to the final motion label at pixel $ij$. This Markov cell has the desired behavior for recovering the

$7(f_{ij} - p_{ij})$ may take only three values: $0, +1$ or $-1$. Indeed, $p_{ij}$ and $f_{ij}$ take binary values in the set $\{0 \equiv \text{GND} \quad 1 \equiv V_{cc}\}$ since they are obtained after thresholding electrical potentials. Hence, we have $(f_{ij} - p_{ij}) = \text{sgn}(f_{ij} - p_{ij})$.

$8$We suppose that motion detection was correctly performed at the previous instant (i.e., past label $p_{ij}$ has the right value).
complete motion masks, and it eliminates isolated points due to acquisition noise, thanks to the resistive smoothing. This clearly outperforms a simple frame difference.

B. Complete Cell

The elementary Markov cell needs some circuitry around it in order to work properly when integrated in a focal plane network. The complete cell (Fig. 5) must include a photoreceptor plus the required circuits to compute \( o_{ij}, f_{ij} \), and \( p_{ij} \) to give the desired binary output, i.e., static or mobile labeling, and must be interconnected to other cells and to line and column shift registers (or address decoders) for outputting results on a video monitor.

![Figure 5](image)

Fig. 5. Complete cell with acquisition (photoreceptor), preprocessing (computation of \( o_{ij}, f_{ij} \)), processing (Markov cell relaxation), and post-processing (binarization, and memory output latch to get \( p_{ij} \)). The preprocessing stage with the track and hold (T/H) circuit is depicted inside the white box.

Light intensity at pixel \( ij \) is acquired continuously by a photoresistor. An analog T/H circuit samples the input current at a rate corresponding to standard video rate, e.g., 30 images/s (clock \( \text{clk}_0 \)). Observation and future label are computed by elementary circuits (delay memories, absolute value of difference, and comparator). This constitutes the preprocessing stage.

The processing stage (relaxation) is embedded in the Markov cell itself (dark box in the middle of Fig. 5). Once command voltages are settled, clock \( \text{clk}_r \) triggers the network relaxation, allowing all Markov cells to relax in parallel to their state of minimal power dissipation.

The postprocessing stage consists of a comparator and a memory output latch. When steady state is reached after network relaxation, clock \( \text{clk}_s \) triggers output binarization and storage of final label at the same rate as input sampling. A simple threshold \( \theta_{\text{out}} \), applied on the electrical potential, gives the desired binary label. The binary output is also stored in memory to give \( p_{ij} \) for the next clock trigger \( \text{clk}_r \). The timing diagram is given in Fig. 6.

![Figure 6](image)

Fig. 6. Timing diagram (\( \Delta t = 33\,\text{ms} \)). \( \text{acq}, \text{pp}, \text{proc}, \) and \( \text{post} \) stand respectively for acquisition, preprocessing, processing, and postprocessing.

C. Network Architecture

The network architecture is given in Fig. 7. In addi-
IV. VLSI IMPLEMENTATION

A. Switched-Current Mode for Preprocessing Stage

For hardware implementation, one may choose among three approaches. The most usual way is to handle information as voltages (classical technology using operational amplifiers (op-amps)). Another choice is the technique of switched capacitors, where information is represented as electrical charges. This usually requires linear precision capacitors fabricated with additional process step over standard MOS technology.

An attractive alternative for smart-pixel CNN implementation is to work in current-mode [13]. The technique of switched currents (SI) is well suited for the preprocessing stage. The SI-concept was introduced by Hughes et al. [14] for the design of analog systems in standard MOS technology for sampled-data signal processing. Here, the sampling is temporal, corresponding to standard video rate of 30 images/s, i.e., Δt = 33ms between two consecutive images. The main advantages of SI technique are low-voltage operation, high speed, no need for precise capacitors or op-amps.

The basic building block of the SI technique is the dynamic current mirror, which is controlled by a switching transistor driven by a clock signal Clk. When Clk is set, the circuit acts as a classical inverting current amplifier. When Clk is reset, the circuit has a memory effect. The output current keeps the value it had at the switching instant. This circuit thus plays the role of a basic current T/H circuit, i.e., an analog current memory.

A keystone circuit for the SI technique is the analog delay memory (z^(-1)). The basic dynamic current mirror cannot be used as such to implement a delay memory, mainly because of the switch clock feedthrough effect between signals and clocks. In order to reduce the influence of clock feedthrough, a current cancellation mirror is used, based on the idea of Song et al. [15]. Many other solutions for improved dynamic current mirrors are reported in the literature, e.g., [16]. This remains an active field of research.

The circuit in charge of computing observation α(t+1) is given in Fig. 8. Its relative complexity (20 transistors) is due to a design effort made for reducing the dead zone arising for low output currents. SPICE analysis shows the behavior of this circuit (Fig. 9). The insensitivity to small currents is favorable to the processing since it acts as a non-linear filter applied on observations by lowering the influence of too small currents that are not significant of actual intensity changes induced by motion.

Since the observation computed by the AVD circuit is represented as a current, the input thresholding circuit (θ_m) required for computing f_{ij} works in current mode. Thus, a current comparator is needed for the preprocessing stage. One can use circuits such as the one proposed in [17], although a simpler circuit was used here.

Fig. 10 reports SPICE simulation of the whole preprocessing stage in the case of a sinusoidal variation of input current. Electrical signals, corresponding to observation and future label, are of sufficient quality for our purpose despite the peaks observable on other signals involved.
in preprocessing stage.

A voltage-controlled current inverter \((\text{sgn})\) is required to get a signed value of \(\alpha_{ij}\); the sign \(0, \pm 1\) depending on \((F_{ij} - p_{ij})\). It might be implemented either with current mirrors or op-amps (design not detailed here).

Concerning the photoreceptor, either a phototransistor or a photodiode may be used [13]. An important issue is light adaptation [5].

**B. Postprocessing Circuitry**

The building blocks required for the design of the postprocessing stage work in classical voltage-mode, and have already been addressed in the literature. An output thresholding circuit is required to compare the steady-state electrical potential \(I_{ij}\) with a fixed reference voltage \(\theta_{\text{out}}\) for binarization after relaxation. This output comparator working in voltage mode can be implemented with a classical transconductance amplifier followed by a comparator (op-amp). An output latch (\(z^{-1}\)) is also required to get \(p_{ij}\). For outputting binary labels one may choose among various solutions. Either shift-registers (made of flip flops), as proposed here, or classical memory addresedecoding (requiring only one transistor at each cell output, which is less expensive in terms of area) may be used.

**C. Markov Cell Design and Simulation**

Voltage-controlled current sources \(J_1\) and \(J_2\) may be realized with a single MOS transistor that consumes little area.

Markov cell resistors (especially \(R_p\)) should be adjustable in order to be able to tune model parameters \(\beta_i\) according to acquisition and lighting conditions. Moreover, floating resistors are needed, and they should be as symmetric as possible to get an isotropic behavior. A simple solution consists in using the channel of a MOS transistor working in the linear region (Fig. 11). The aspect ratio \(W/L\) of spatial resistors \(R_e\) in each cell, which spare nine transistors per cell. Electrical simulations (HDL-A simulator)\(^9\) for three different polarizations of the resistor circuit are given in Fig. 12, when \(V_{DS}\) is varied in the range \([-0.5V; 0.5V]\). One obtains nonlinear resistors. Interestingly enough, however, is the fact that this nonlinearity is favorable for motion detection, as explained below.

Electrical simulation of the adjustable MOS resistor for three different polarizations \(V_{pol}: 2.18V (\times); -2.28V (+); 2.38V (\bullet)\). Source voltage applied to \(MN_{res}\) is constant \((V_{G} = 5.5V)\) while \(V_{DS}\) is varied in the range \([0.4V; 0.7V]\).

Fig. 12. (a) Drain current in transistor \(MN_{res}\) as a function of \(V_D\). (b) Resistance variation as a function of \(V_D\).

Visual comparison simulation obtained for a cell made of resistors designed with MOS transistors. They are in good agreement with the theoretical behavior depicted in Fig. 13(c). The nonlinear behavior of a MOS transistor is illustrated in Fig. 12, where the drain current \(I_D\) is plotted against the gate voltage \(V_G\).

\[ I_D = K \left( \frac{V_G}{V_T} - 1 \right)^2 \]

\[ V_T = V_{DS} \left( \frac{	ext{width}}{	ext{length}} \right) \]

Fig. 11. Adjustable and floating MOS resistor. Circuit schematic. Transistors \(MP_1, MN_1,\) and \(MN_2\) constitute the polarization circuit. \(MN_{res}\) is the floating adjustable resistor.

\(^9\)HDL-A and Eldo are modeling and simulation tools, developed by ANACAD Electrical Engineering Software, for analog and mixed-signal simulation with VHDL-based language and SPICE syntax.

\(^{10}\)Note that for static-to-static transition (curve 0), the simulation reports a slightly increasing potential, whereas it should stay constant. This is not a bad behavior of the cell, but only results from the fact that \(V_{DS}\) is actually decreasing for simulation purposes. In a real situation, however, a static pixel gives a null observation \(V_{DS} = 0\). Hence, \(I_{ij}\) would actually remain constant.
Fig. 13. Electrical simulation of Markov cell. Curve symbols correspond to the four typical cases static-to-static (○); mobile-to-static (+); mobile-to-mobile (●); and static-to-mobile (×). Observation voltage \( V_{\text{obs}} \equiv o_{ij} \) is varied linearly in the range [0–1V] for simulation purposes. (a) Current generated by \( J_2 \) as a function of \( V_{\text{obs}} \). (b) Current generated by \( J_1 \). (c) Electrical potential \( V_{\text{tot}} \equiv \sum_{ij} \) for a cell made of ideal MOS resistors (HDL-A simulator). (d) Idem, but for a cell made of real MOS resistors (HDL-A simulator).

Electrical simulation of Markov cell. Curve symbols correspond to the four typical cases static-to-static (○); mobile-to-static (+); mobile-to-mobile (●); and static-to-mobile (×). Observation voltage \( V_{\text{obs}} \equiv o_{ij} \) is varied linearly in the range [0–1V] for simulation purposes. (a) Current generated by \( J_2 \) as a function of \( V_{\text{obs}} \). (b) Current generated by \( J_1 \). (c) Electrical potential \( V_{\text{tot}} \equiv \sum_{ij} \) for a cell made of ideal MOS resistors (HDL-A simulator). (d) Idem, but for a cell made of real MOS resistors (HDL-A simulator).

Electrical simulation of Markov cell. Curve symbols correspond to the four typical cases static-to-static (○); mobile-to-static (+); mobile-to-mobile (●); and static-to-mobile (×). Observation voltage \( V_{\text{obs}} \equiv o_{ij} \) is varied linearly in the range [0–1V] for simulation purposes. (a) Current generated by \( J_2 \) as a function of \( V_{\text{obs}} \). (b) Current generated by \( J_1 \). (c) Electrical potential \( V_{\text{tot}} \equiv \sum_{ij} \) for a cell made of ideal MOS resistors (HDL-A simulator). (d) Idem, but for a cell made of real MOS resistors (HDL-A simulator).

the chip.

The power dissipation for a network of size 64 × 64 is evaluated to be about 150 mW. Resistor values are in the range 100 kΩ to 1 MΩ. The order of magnitude for currents is about 1 μA with \( V_{\text{bus}} = 1V \). Relaxation duration is typically less than 1 μs, so that real-time motion detection is achievable.

V. EXPERIMENTAL RESULTS

Electrical simulations of the network were performed with Eldo and HDL-A. Since the complete chip with photoreceptors is not yet fabricated, gray-level images stored in computer memory are used. Pixels are 8-bit coded. Command voltages \( (o_{ij}, f_{ij}) \) are precomputed externally and entered into the Markov cell description file. Hence, only the electrical behavior of network-interconnected Markov cells is simulated. Concerning parameter setting, threshold values \( \theta_{\text{in}}, \theta_{\text{out}} \) given in the captions, refer to the 256 gray-level dynamic range of images. They are adjusted depending on the sequence (lighting conditions, noise, etc.). For a proper working of the network, the condition \( R_f \leq R_p < R_c \) is required. \( R_c \) and \( R_f \) values are constant for all sequences. Typical values are \( R_c = 400 \, \text{kΩ}, R_f = 200 \, \text{kΩ}, R_p = 220 \, \text{kΩ} \). \( G_1 \) and \( G_2 \) are the transconductances of current sources \( J_1 \) and \( J_2 \). Their order of magnitude is \( \sim 10^{-6} \) Siemens. An important feature is the scalability of all parameters, that can be chosen according to the maximum power dissipation acceptable in the network.

A synthetic sequence of eight images is shown on the upper row of Fig. 15. Image size, and hence network size, is 16 × 16. The sequence contains two dark squares moving on a noisy background. The squares have a nonuniform intensity (linear variation of gray level). The second row represents observation fields where transition areas (front and echo) are visible. In the inner part of moving objects...
(sliding areas) observations are small but not zero (light gray barely visible). The coarse estimate of future label fields (binary maps) is given in third row. Because of a too high input threshold \( \theta_{in} \), only transition areas are retained and inner parts are eliminated. They are restored, however, by the relaxation process, as shown in the fourth row, representing the moving masks obtained after thresholding steady-state electrical potentials of the relaxed network. The masks are perfectly extracted after the fourth image\(^{12}\).

The echo is canceled, and the inner part of the masks is filled in. The lower row shows the three-dimensional maps of electrical potentials. Nodes having a high electrical potential correspond to mobile pixels, while nodes at a potential near the ground voltage correspond to static pixels.

Five images of a real-world sequence are shown in the upper row of Fig. 16. This street sequence contains a white car moving in a street as seen from the top of a building. Other cars in the background are parked. This sequence was acquired with a three-CCD (charge coupled device) camera (only the 8-bit green component is processed). Image size, and hence network size, is 64 x 64. The second row shows the observation fields, where echo is clearly visible. The third row gives the future label fields, with enlarged masks because of the echo phenomenon. The bottom row of Fig. 16 gives the moving masks detected by the network after relaxation. Although the car is moving rapidly (passing the field of view within eight images) and is poorly-textured (uniform white roof and hood), it is fairly well detected.

The circuit is insensitive to contrast: dark or light moving objects are equally well detected. Other simulations not reported here prove that the network also detects slow or fast moving objects, small objects, or objects in noisy sequences.

VI. Discussion

This article reports the design of a smart sensor for robust motion detection. Starting from a computational method, the analog implementation leads to what we call a cellular analog network (CAN), rather than CNN, in order to emphasize the fact that it is not biologically inspired. Still, the resulting circuit reveals analogy with biological neural networks. The proposed chip is a full vision system, coupling sensor and computation, that recovers accurately the complete masks of moving objects in the case of a static camera. The processing proved to be robust in various situations, such as contrast, noise, speed, number, and size of moving objects.

The counterpart of this smart low-level processing is a fill factor that is not yet satisfactory. Therefore, much work has still to be done, such as further investigation for the design of all building blocks in the circuit, study of submicronic technological aspects (interconnections, power consumption, surface, delay on clock-lines, etc.), and precise evaluation and validation of the circuit (influence of all parameters). Interesting is the fact that, similar to biological systems, accuracy is not a critical issue. We found that analog implementation even outperforms computer simulations, both for processing speed and detection quality.

Acknowledgment

The authors wish to thank V. G. Popescu and S. Popescu for their contribution to this work and J. Hérault for fruitful discussions.
REFERENCES


