All-MOS implementation of RC networks for time-controlled Gaussian spatial filtering

J. Fernández-Berni*, R. Carmona-Galán

Institute of Microelectronics of Seville (IMSE-CNM),
Consejo Superior de Investigaciones Científicas y Universidad de Sevilla,
C/ Américo Vespucio s/n, 41092, Seville, Spain (e-mail: berni@imse-cnm.csic.es).

SUMMARY

This paper addresses the design and VLSI implementation of MOS-based RC networks capable of performing time-controlled Gaussian filtering. In these networks, all the resistors are substituted one by one by a single MOS transistor biased in the ohmic region. The design of this elementary transistor is carefully realized according to the value of the ideal resistor to be emulated. For a prescribed signal range, the MOSFET in triode region delivers an interval of instantaneous resistance values. We demonstrate that, for the elementary 2-node network, establishing the design equation at a particular point within this interval guarantees minimum error. This equation is then corroborated for networks of arbitrary size by analysing them from a stochastic point of view. Following the design methodology proposed, the error committed by a MOS-based grid when compared to its equivalent ideal RC network is, despite the intrinsic nonlinearities of the transistors, below 1% even under mismatch conditions of 10%. In terms of image processing, this error hardly affects the outcome, which is perceptually equivalent to that of the ideal network. These results, extracted from simulation, are verified in a prototype vision chip with QCIF resolution manufactured in the AMS 0.35μm CMOS-OPTO process. This prototype incorporates a focal-plane MOS-based RC network which performs fully-programmable Gaussian filtering.

KEY WORDS: Gaussian filtering, focal-plane processing, RC networks, time-controlled diffusion, VLSI implementation

1. INTRODUCTION

Gaussian filtering is a basic task for early vision. It is used for reducing the noise associated to the image capture without affecting subsequent processing stages. In fact, the image enhancement through the Difference of Gaussians (DoG) is preferred over other image enhancemement methods as it preserves the details of interest within the scene while filtering

*Correspondence to: berni@imse-cnm.csic.es

Contract/grant sponsor: Junta de Andalucía-CICE; Ministerio de Ciencia e Innovación; contract/grant number: 2006-TIC-2352; TEC 2009-11812, co-funded by FEDER

sharp random noise [1]. In this case, the width of the Gaussian filters involved will depend on the noise nature as well as on the scale of the objects to be analysed, not usually known in advance. Scale is indeed a key point when it comes to efficiently process images. Its adequate selection simplifies the analysis of certain features in a scene by removing information at other scales [2]. If several scales are to be analysed, pyramidal representations can be easily built in order to only store the necessary information for each representation according to the spatial frequencies involved [3]. The scale-space representation of a scene is obtained by applying successive Gaussian filters with increasing widths over its original representation [4].

From the processing features above described, it can be seen that the utility of Gaussian filtering reaches its maximum level when the width, or smoothing degree, is under the control of the user. Programmable digital processors [5] support this possibility by means of user-defined convolution kernels. However, this implementation is not very efficient, taking into account the necessary serialization of the raw image data along with the repeated accesses to memory to operate over each and every pixel and its neighborhood. Besides, this energy inefficiency increases with the width of the filter as more neighbors are involved in the computation for each pixel — the kernel size must be at least 6 times the variance of the targeted Gaussian filter [6]. This has led to alternative approaches, most of them making use of the ability of CMOS processes to integrate pure imaging with signal processing circuitry. Thus, they achieve massively parallel focal-plane processing concurrent with the photosensing, delivering the same result as a digital processor but in a much more efficient way.

Resistive grids are the first alternative to be considered [7, 8]. They are passive networks which can perform different spatial filterings by using positive and, if necessary, negative resistors. Their robustness to mismatch makes them specially suitable for VLSI implementation [9,10]. In [11], a single-chip analog implementation of a resistive network for Gaussian filtering is described. Negative resistors are mandatory in order to attain a Gaussian-like convolution kernel, what makes the circuitry bulky due to the negative impedance converters. The variation of the filter width is achieved by two MOSFETs in parallel biased in the triode region whose control of the gate voltages results in a variable resistor. It is precisely the control circuit of this variable resistor what greatly increases the power consumption of the chip. Moreover, only Gaussians with a width variable by a factor of 2 are available. Other possibility of Gaussian filtering in resistive grids is through MOSFETs working in subthreshold regime [12,13]. In this case, the main drawback is the significant influence of leakage currents and mismatch for a fine control of the filtering width [14]. Finally, the filtering performed by a resistive grid can be theoretically emulated by the CNN framework [15]. However, the unavoidable mismatch presents at any VLSI implementation prevents the typical transconductance-based approach from achieving Gaussian filtering with enough accuracy, specially for large widths [16].

In [17], the physical implementation of Gaussian filters with user-defined width is addressed in a totally different way. It introduces a capacitive network which can be considered as a numeric solver of the spatially-discretized diffusion equation. The variance of the filter is determined by a capacitor ratio, fixed by layout design, and a iteration number associated to the implicit time discretization of the network. Four switches, two switching capacitors and one grounded capacitor amount to each node. An error of 1% is delivered when the iteration number is higher than ten. A more recent VLSI implementation of this approach is reported in [18].

This paper proposes to take advantage of the dynamics of a RC network in order to achieve user-defined Gaussian filters. A RC network can be also considered as a solver of the spatially-

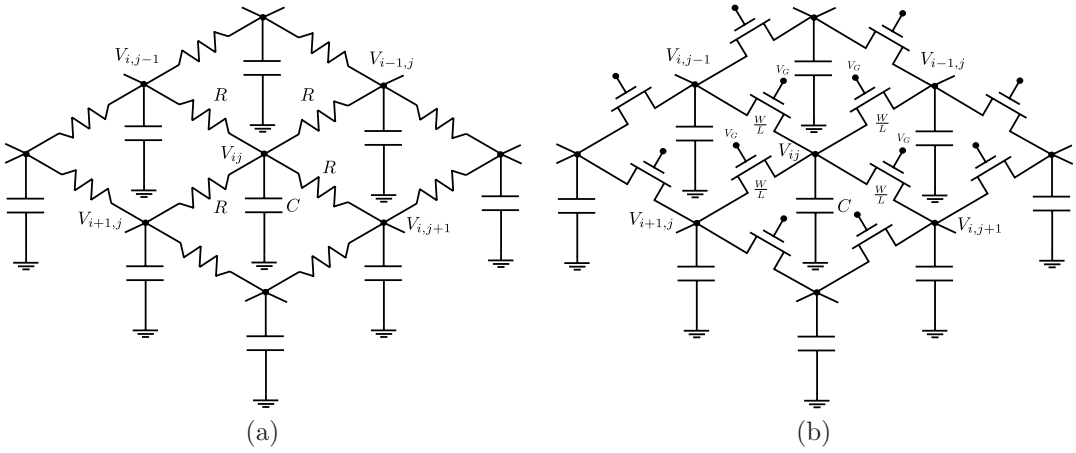


Figure 1. RC network performing linear diffusion (a) and its MOS-based counterpart (b)

discretized diffusion equation, but no time discretization is now realized. This means that any filter width is ideally possible. The main drawback which can be argued against this approach is the considerable area consumption of resistors in CMOS processes. We will demonstrate that this problem can be solved by substituting each resistor by a single MOS transistor biased in the ohmic region. The ratio resistance/area of this elementary MOS is much greater than that of the resistors made with polysilicon or diffusion strips. Besides, the dynamics of the network can be activated or deactivated by controlling the gate voltages of the transistors. The effects of the inevitable nonlinearities introduced by the MOSFETs are alleviated by a careful design of the elementary transistor and a very simple on-chip calibration. An error below 1% is thus achieved, which is translated into perceptually equivalent outputs. Only two transistors, acting as switches with an ON resistance designed ad-hoc, and a grounded capacitor amount to each node.

The paper is organized as follows. Section 2 defines the operation of a RC network as a solver of the spatially-discretized diffusion equation, determining the corresponding filtering function. Section 3 compares a 2-node ideal RC network with its counterpart where the resistor is substituted by a MOSFET. We thus find the design equation for a MOS transistor emulating a targeted resistance with minimum error. Section 4 extends the design equation previously obtained to networks of arbitrary size. Section 5 shows some simulation results for a 64×64 network designed by making use of such a equation. Finally, Section 6 describes an on-chip fully-programmable Gaussian filtering operation which validates the design methodology proposed.

2. GAUSSIAN FILTERING IN A RC NETWORK

The generic RC network analysed throughout this paper is depicted in Fig. 1(a), where the initial voltage at the capacitor of every node represents the value of the corresponding pixel.

$\sigma = 0$ $\sigma = 0.6$ $\sigma = 1$ $\sigma = 1.3$

Figure 2. Gaussian filtering over the Lena image (64×64 px).

Our objective is to design the MOS-based version in Fig. 1(b) in such a way that minimum error is committed. For the ideal RC network, the equation satisfied at each node is:

$$\tau \frac{dV_{ij}}{dt} = -4V_{ij} + V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} \tag{1}$$

where $\tau = RC$. Eq. (1) represents the spatially-discretized diffusion equation and its solution is formally the scale-space representation of 2-D discrete signals [19]. Indeed, what happens in the unforced RC network is that the initial charge of the capacitors actually diffuses, with a pace determined by τ. Applying the DFT to Eq. (1) we obtain:

$$\tau \frac{d\hat{V}_{uv}}{dt} = -4\hat{V}_{uv} + e^{\frac{2\pi i u}{M}}\hat{V}_{uv} + e^{\frac{-2\pi i u}{M}}\hat{V}_{uv} + e^{\frac{2\pi i v}{N}}\hat{V}_{uv} + e^{\frac{-2\pi i v}{N}}\hat{V}_{uv} \tag{2}$$

where we have considered an array whose size is $M \times N$ pixels. Notice that the dynamics of those nodes located just at the edge of the array is not affected by a complete 4-connected neighborhood but by a reduced 2- or 3-connected one. It is equivalent to consider mirroring boundary conditions at every time instant for the edges of the array. That is, Eq. (1) can be also used for the nodes at the boundaries. But bear in mind that the nodes falling outside the array correspond to dummy nodes which do not affect the dynamics of the network as their value always equals that of the boundary node under consideration at every time instant. Eq. (2) can be rewritten as:

$$\tau \frac{d\hat{V}_{uv}}{dt} = -4\left[\sin^2\left(\frac{\pi u}{M}\right) + \sin^2\left(\frac{\pi v}{N}\right)\right]\hat{V}_{uv} \tag{3}$$

and solving now in the time domain we obtain:

$$\hat{H}_{uv}(t) = \frac{\hat{V}_{uv}(t)}{\hat{V}_{uv}(0)} = e^{-\frac{4t}{\tau}\left[\sin^2\left(\frac{\pi u}{M}\right)+\sin^2\left(\frac{\pi v}{N}\right)\right]} \tag{4}$$

where $\hat{V}_{uv}(0)$ represents the DFT of the image defined by the initial voltages at the capacitors and $\hat{V}_{uv}(t)$ is the DFT of the image defined by those same node voltages after a certain time interval t since the network started to evolve at time instant $t = 0$. Thus Eq. (4) describes the filtering process undergone by the initial image as the network evolves. It corresponds to the spatially-discretized version of the ideal Gaussian filtering with spatial width $\sigma = \sqrt{2t/\tau}$ performed by a continous-plane diffusion process [3]. Therefore, for a certain τ, each time

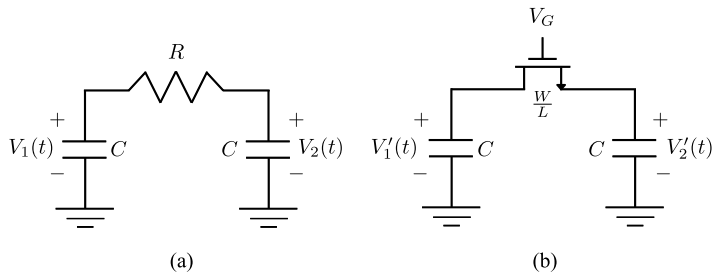


Figure 3. 2-node ideal RC network (a) and its MOS-based implementation (b)

instant of the dynamics of the network will be equivalent to a different Gaussian filter. As an example, consider the Fig. 2, where we have applied Gaussian filters with $\sigma = 0$, $\sigma = 0.6$, $\sigma = 1$ and $\sigma = 1.3$ over the picture of 'Lena'.

3. ANALYSIS OF A 2-NODE NETWORK

Let us compare the 2-node grids in Fig. 3 as a first approximation to the design of MOS resistors for RC networks. For purposes of clarity, we will confine the analysis to n-channel MOS transistors, but it applies equally well for p-type transistors. The gate voltage V_G is fixed and we will assume, without loss of generality, that the initial conditions of the capacitors fulfill $V_{10} > V_{20}$, being $V_{10} = V'_{10}$ and $V_{20} = V'_{20}$. We will also assume that the transistor is biased in the triode region for any voltage at the drain and source terminals, that will range from V_{min} to V_{max}. The evolution of the circuit in Fig. 3(a) is described by this set of ODEs:

$$\left\{ \begin{array}{rcl} C\frac{dV_1}{dt} & = & -\frac{V_1(t)-V_2(t)}{R} \\ C\frac{dV_2}{dt} & = & \frac{V_1(t)-V_2(t)}{R} \end{array} \right. \tag{5}$$

while the behaviour of the circuit in Fig. 3(b) is described by:

$$\left\{ \begin{array}{rcl} C\frac{dV_1'}{dt} & = & -\frac{V_1'(t)-V_2'(t)}{R_M(t)} \\ C\frac{dV_2'}{dt} & = & \frac{V_1'(t)-V_2'(t)}{R_M(t)} \end{array} \right. \tag{6}$$

being:

$$R_M(t) = \frac{1}{k_n S_n \left\{V_C - [V_1'(t) + V_2'(t)]\right\}} \tag{7}$$

where $k_n = \mu_n C'_{ox}/2$, $V_C = 2\left(V_G - V_{T_n}\right)$ and $S_n = W/L$. Several key aspects must be clarified at this point about Eq. (7). Firstly, it corresponds to the instantaneous resistance of the transistor derived from the classical first-order approximation for the drain current of a NMOS biased in the triode region. This is really a coarse approximation for the real behaviour of the transistors. However, it will permit to draw conclusions while keeping the equations reasonably manageable. These conclusions are confirmed not only by simulation, where the

models include a great deal of second-order effects, but also in the physical implementation realized. It means that the mentioned first-order approximation summarizes the essential features of the transistors for a trustful design of MOS-based RC networks. Secondly, due to charge conservation, Eq. (7) can be expressed as:

$$R_M = \frac{1}{k_n S_n [V_C - (V'_{10} + V'_{20})]} \tag{8}$$

which means that, neglecting second-order effects, the resistance of the transistor depends on the sum of the initial conditions and does not vary along the corresponding diffusion. In other words, Eq. (8) is telling us that, if we choose a certain set of initial conditions within the prescribed signal range whose sum coincides and make $R_M = R$ for this sum, the dynamics for any initial conditions within this set will be perfectly emulated by the MOS network. On the contrary, there will always be an error for any other initial conditions outside the set as the resistance of the transistor during the diffusion will never match R. The question arising here is therefore: what is the value of the sum of initial conditions which, fulfilling $R_M = R$, minimizes the maximum error committed for any other possible value of this sum? That is, defining $V_S = V'_{10} + V'_{20} = V_{10} + V_{20}$, what is its optimum value $V_{S_{opt}}$ for which making:

$$\frac{1}{k_n S_n (V_C - V_{S_{opt}})} = R \tag{9}$$

the maximum error committed by the MOS network for any other possible value of V_S is minimum? The design equation of the MOSFET is immediately derived from Eq. (9):

$$S_{n_{opt}} = \frac{1}{k_n R (V_C - V_{S_{opt}})} \tag{10}$$

where the value of $V_{S_{opt}}$ must be within the interval $[2V_{min}, 2V_{max}]$ according to the signal range previously established. Note that this design equation demands to know the exact value of k_n, which can present significant variations across the design space delimited by the corners of the process. We will see in Section 6 how to solve this problem.

In order to determine $V_{S_{opt}}$, notice firstly that the charge extracted from one capacitor will end up in the other at both the ideal network and the MOS-based network. We can thus define the error in the corresponding node voltages as:

$$\begin{cases} V'_1(t) = V_1(t) + \epsilon(t) \\ V'_2(t) = V_2(t) - \epsilon(t) \end{cases} \tag{11}$$

or, equivalently:

$$\epsilon(t) = \frac{V'_1(t) - V'_2(t)}{2} - \frac{V_1(t) - V_2(t)}{2} \tag{12}$$

Since our initial assumptions were $V_{10} = V'_{10}$ and $V_{20} = V'_{20}$, we have that $\epsilon(0) = 0$. Also, the stationary state, reached when $t \to \infty$, renders $\epsilon(\infty) = 0$, as $V_1(\infty) = V_2(\infty)$ and $V'_1(\infty) = V'_2(\infty)$. Therefore, there must be at least one point in time, let us call it t_{ext}, in

which the error reaches an extreme value, either positive or negative. That is, since the time derivative of the error can be expressed as[†]:

$$\tau \frac{d\epsilon}{dt} = \left[V_1'(t) - V_2'(t)\right] \frac{V_{10} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} - 2\epsilon(t) \tag{13}$$

with $\tau = RC$, it must cancel in t_{ext}, resulting in an extreme error of:

$$\epsilon_{ext} = \frac{1}{2}\left[V_1'(t_{ext}) - V_2'(t_{ext})\right] \frac{V_{10} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} \tag{14}$$

where V_C is a constant, $V_{S_{opt}}$ is a design parameter and $V_1'(t_{ext})$ and $V_2'(t_{ext})$ are variables which can be referred to the initial conditions by solving Eq. (6), obtaining:

$$\epsilon_{ext} = \frac{1}{2}(V_{10} - V_{20}) \frac{V_{10} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} e^{\frac{-2t_{ext}}{R_M C}} \tag{15}$$

Finally, t_{ext} can be found by solving Eq. (5) and Eq. (6) and making use of Eq. (11):

$$t_{ext} = \frac{\tau}{2} \frac{\ln(r)}{r - 1} \tag{16}$$

where:

$$r = \frac{R}{R_M} = \frac{V_C - V_{10} - V_{20}}{V_C - V_{S_{opt}}} \tag{17}$$

Substituting Eq. (16) in Eq. (15), we have the following expression:

$$\epsilon_{ext} = \frac{1}{2}(V_{10} - V_{20}) \frac{V_{10} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} r^{\frac{r}{1-r}} \tag{18}$$

from which the first important conclusion can be extracted. The extreme error is independent of S_n and k_n. That is to say, once S_n is defined by Eq. (10) for a value of k_n known, the error committed by the MOS network does not depend on these parameters and therefore is not affected by their mismatch. This robustness to mismatch is confirmed both by simulation and in the physical implementation presented in this paper.

Unfortunately, to obtain the exact analytical expression for the extremes of ϵ_{ext} from Eq. (18) is not possible. This in turn implies that the exact analytical expression for $V_{S_{opt}}$ can not be found either. However, a good approximation for them is still possible. Let us take a look to Eq. (17). The value of r depends on a quotient where V_C is the dominant term at both the numerator and the denominator for deep triode biasing. Let us therefore assume that $\partial r/\partial V_{10} \simeq 0$ and $\partial r/\partial V_{20} \simeq 0$. Under these conditions, it can be demonstrated that only a critical point, more precisely a saddle point, can be found at $(V_{S_{opt}}/2, V_{S_{opt}}/2)$. Therefore, we can only talk of absolute maxima or minima which will be at the boundaries of the domain considered for (V_{10}, V_{20}). It can be demonstrated[‡] that the absolute extremes are located at

[†]See APPENDIX A
[‡]See APPENDIX B

the points $(V_{10}, V_{20}) = (V_{max}, V_{S_{opt}}/2)$ and $(V_{10}, V_{20}) = (V_{S_{opt}}/2, V_{min})$ being, respectively, their values:

$$\begin{cases} \epsilon_{ext}|_{max} &= \frac{1}{8}\frac{\left(2V_{max}-V_{S_{opt}}\right)^2}{V_C-V_{S_{opt}}}\left(\frac{1}{2}+\frac{1}{2}\frac{V_C-2V_{max}}{V_C-V_{S_{opt}}}\right)^{\frac{2(V_C-V_{max})-V_{S_{opt}}}{2V_{max}-V_{S_{opt}}}} \\ \epsilon_{ext}|_{min} &= -\frac{1}{8}\frac{\left(V_{S_{opt}}-2V_{min}\right)^2}{V_C-V_{S_{opt}}}\left(\frac{1}{2}+\frac{1}{2}\frac{V_C-2V_{min}}{V_C-V_{S_{opt}}}\right)^{-\frac{2(V_C-V_{min})-V_{S_{opt}}}{V_{S_{opt}}-2V_{min}}} \end{cases} \tag{19}$$

The final step is to determine the value of $V_{S_{opt}}$ which minimizes $\max|\epsilon_{ext}|$. To this end, let us again assume deep triode biasing. In this way, the exponential terms in $\epsilon_{ext}|_{max}$ and in $\epsilon_{ext}|_{min}$ can be approximated by 1 when varying $V_{S_{opt}}$ within the range of its possible values $[2V_{min}, 2V_{max}]$. Applying this approximation, it can be seen that to increase or to decrease $V_{S_{opt}}$ has antagonistic effects in the magnitude of $\epsilon_{ext}|_{max}$ and $\epsilon_{ext}|_{min}$, being:

$$V_{S_{opt}} = V_{min} + V_{max} \tag{20}$$

the expression for $V_{S_{opt}}$ which minimizes the magnitude of the error:

$$\min\left(\max|\epsilon_{ext}|\right) = \frac{1}{8}\frac{(V_{max}-V_{min})^2}{V_C - V_{min} - V_{max}} \tag{21}$$

Notice that this minimized error depends inversely on V_C, that is, on $V_G - V_{T_n}$, which was considered fixed at the beginning of the design process. Thus, V_G and V_{T_n} must be chosen in such a way that makes their difference as large as possible.

Finally, by substituting Eq. (20) in Eq. (10), the design equation for minimum error is obtained:

$$S_{n_{opt}} = \frac{1}{k_n R(V_C - V_{min} - V_{max})} \tag{22}$$

This conclusion about $V_{S_{opt}}$ invalidates the groundless intuition that the optimal design could be derived from equaling the midpoint of the interval of possible values of R_M to R. On the contrary, the value of R_M which matches R for optimal design is notably below such a midpoint.

It is time now to numerically corroborate the validity of the assumptions realized and, on the way, to know the magnitude of the error achieved by the optimum design. In Table I some results are showed for typical values of the voltages involved. As above mentioned, the larger V_G the smaller the error. Thus, two usual maximum biasing voltages for the transistor gate in current CMOS technologies are introduced in Table I. Regarding V_{T_n}, two typical related values are used. We do not take into account the possibility of using low- or even zero-threshold transistors currently offered by some manufacturers, what would further reduce the error. Note that the columns labelled as $\max e_{ext}$, $\min e_{ext}$ and $V_{S_{opt_{num}}}$ are numerically calculated from Eq. (18) whereas those ones labelled as e_{max}, e_{min} and $V_{S_{opt}}$ are directly calculated from Eq. (19) and Eq. (20) respectively. Several conclusions can be drawn from a careful analysis of Table I. First of all, the deeper the ohmic biasing the less the error committed by the MOS network and the better the approximations applied, as expected. Secondly, despite the large signal swings considered and therefore the large variation of the instantaneous resistance of

the MOSFET, represented by the wide interval of possible values of r, the optimal design keeps the error moderately small. It is due to the suitability of this design for the diffusion dynamics. Note that in the ideal network, the maximum charge injection occurs when one of the nodes equals V_{max} whereas the other equals V_{min}. This situation can only exist just at the beginning of the diffusion. A significant error committed by the MOS-based grid at this point would mean to noticeably alter the rest of the dynamics. However, such a configuration of the voltages makes their sum equal to $V_{min}+V_{max}$ and therefore the MOS network does not commit any error at all. On the contrary, the error committed by the MOS grid is maximum when the voltages of the nodes involved coincide at V_{min} or V_{max}, being their sum $2V_{min}$ or $2V_{max}$ respectively. But there is no charge injection between the nodes in these cases as their voltages are the same. Therefore, the maximum error is committed when the dynamics is not affected. Finally, remark that no specific value of R has been included in Table I as it is not necessary to compute the error. Once set V_G, V_{T_n} and $[V_{min}, V_{max}]$, $S_{n_{opt}}$ can ideally take, from Eq. (22), any value according to the targeted R without affecting the magnitude of the error.

4. NETWORKS OF ARBITRARY SIZE

Taking into account the mathematical framework deployed to optimize the design of a 2-node MOS network, it is obvious that the extension of the results to networks of arbitrary size can not be addressed in the same way. Our proposal for such extension is a stochastic approach. Let us suppose a $M \times N$ RC grid similar to that of Fig. 1(a) where every initial value of pixel can be modelled by a random variable with an uniform distribution between V_{min} and V_{max}, that is, $\mathbf{V_{ij}}(\mathbf{0}) \sim U(V_{min}, V_{max})$, according to notation in [20]. Such a distribution is depicted in Fig. 4(a). This is a rather reasonable supposition, specially if the grid is intended to process natural images. In this way, if we choose any two neighbor nodes of the network, namely (i, j) and (k, l), the resulting distribution of the sum of both nodes is a triangle like that one represented in Fig. 4(b). It can be seen that the most probable value of the sum is $V_{min}+V_{max}$. Let us see what happens at the end of the processing. Consider again Eq. (4), which defines the filtering carried out by a RC network along time. Notice that the DC component is the only one that is not affected by the filtering, that is, $\hat{H}_{00}(t) = 1\ \forall t$. It means that the average value of the pixels does not change during the processing. Thus, when the diffusion is running, the higher frequencies left are progressively filtered until all of them are eventually removed except the DC component. In other words, the values of the pixels are progressively getting closer until all of them eventually coincides at the mean value, which has never been altered. Let $\bar{\mathbf{V}}$ be the random variable representing the mean value of the pixels of an image. It can be expressed as:

$$\bar{\mathbf{V}} = \frac{\mathbf{V_{1,1}}(\mathbf{0}) + \mathbf{V_{1,2}}(\mathbf{0}) + ... + \mathbf{V_{ij}}(\mathbf{0}) + ... + \mathbf{V_{M,N}}(\mathbf{0})}{MN} \tag{23}$$

from which, taking into account that every pixel $\mathbf{V_{ij}}(\mathbf{0})$ presents an uniform distribution and by applying the central limit theorem, we can conclude that $\bar{\mathbf{V}}$ approaches a normal distribution $N(\mu, \sigma^2)$ as follows:

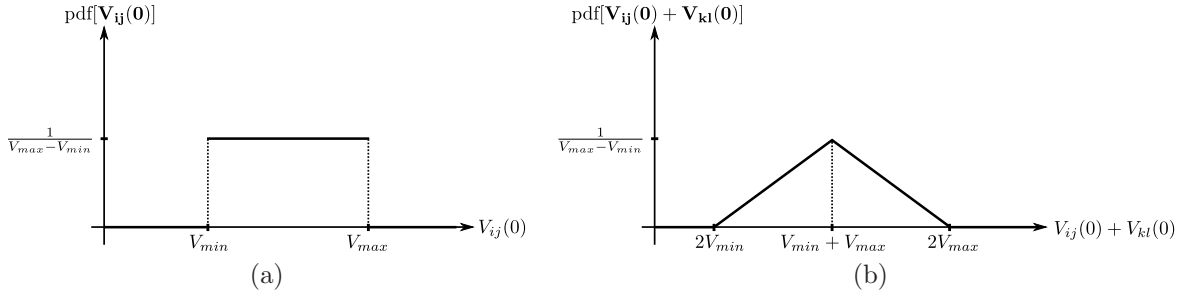


Figure 4. Probability density function for the initial voltage (a) and for the sum of any two initial voltages (b) at the nodes of a RC network.

$$\bar{\mathbf{V}} \sim N\left[\frac{V_{min}+V_{max}}{2}, \frac{(V_{max}-V_{min})^2}{12}\right] \tag{24}$$

and as every pixel will reach the mean value at the end of the diffusion process, $\mathbf{V_{ij}}(\infty)$ presents the same distribution as defined by Eq. (24), depicted in Fig. 5(a). It implies that the most probable value of the sum of any two voltages of the network by the end of the diffusion is again $V_{min} + V_{max}$, as showed in Fig. 5(b). We have seen what happens at the beginning and at the end of the diffusion, but what about the processing itself? We start from uniform distributions for every pixel. When the diffusion is being performed, the probability of a certain pixel to reach the mean value is constantly increasing as the probability of having filtered all the frequencies other than the DC component also increases. Note that, according to Eq. (4), any time interval carrying out diffusion implies necessarily the filtering of frequencies other than the DC component. Thus, the uniform distribution represented in Fig. 4(a) is transformed along the diffusion until eventually becoming that in Fig. 5(a). However, the interest for us in this transformation falls on the fact that, on increasing the probability of a certain pixel to reach the mean value, its most probable value during the diffusion is $(V_{min} + V_{max})/2$ as this is the most probable value of the mean value. And it in turn means that the most probable value of the sum of any two voltages within the ideal network at any time instant during the diffusion is $V_{min} + V_{max}$. From this result, and keeping in mind that the elementary transistor at a MOS-based RC grid emulating the ideal network just described presents an instantaneous resistance:

$$R_{M_{ij,kl}}(t) = \frac{1}{k_n S_n \left\{V_C - \left[V'_{ij}(t) + V'_{kl}(t)\right]\right\}} \tag{25}$$

it can be concluded that this value of resistance should match the resistor of the ideal network for the most probable value of the sum of any two neighbor voltages, that is:

$$\frac{1}{k_n S_n \left[V_C - (V_{min} + V_{max})\right]} = R \tag{26}$$

which directly leads to the design equation obtained for the 2-node case:

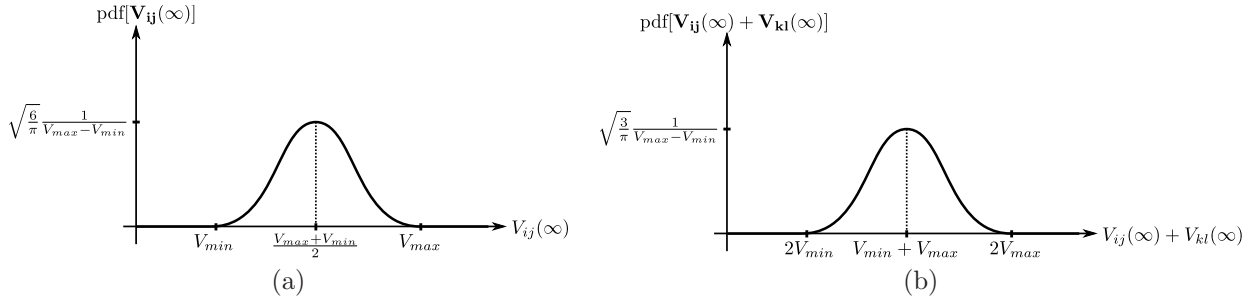


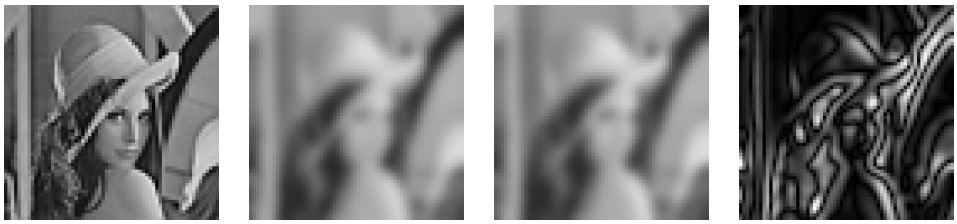
Figure 5. Probability density function for the final voltage (a) and for the sum of any two final voltages (b) at the nodes of a RC network.

$$S_{n_{opt}} = \frac{1}{k_n R(V_C - V_{min} - V_{max})} \tag{27}$$

In this way, we make sure that the instantaneous diffusion performed by each elementary MOS transistor between its drain and source terminals equals that of the corresponding ideal resistor for most of the time instants of the dynamics, introducing thus minimum error from a stochastical point of view. Note that we are implicitly assuming that this error is small enough to apply to the MOS grid the same considerations extracted from the ideal grid regarding the distributions and most probable values of the voltages during the diffusion. The simulation results presented in the next section confirm that this assumption can be made.

5. SIMULATION OF A 64×64 NETWORK

This section addresses the design of a 64×64 MOS-based RC network. The objective is to confirm the validity of the guidelines drawn in the previous Section despite having made use of a coarse approximation for the behavior of the transistors in the triode region. Simulations have been realized using standard 0.35μm CMOS 3.3V process transistor models in HSPICE. The signal range at the nodes is [0V,1.5V], wide enough to evidence the influence of the MOSFET nonlinearities over the Gaussian filtering performed by the grid. V_G is established at 3.3V in order to bias the transistor as deep in the ohmic region as possible. The design specification is to implement a RC network with $\tau = 100$ns by using a resistor $R = 100$kΩ and a capacitor $C = 1$pF. The sizing of the elementary NMOS transistor to achieve this value of R is based on Eq. (27). But this equation does not take into account second-order effects like for example the body effect. It means that S_n does not only depend on the sum $V_{min} + V_{max}$ but also on the values of the voltages at drain and source that render that sum. Thus, for a specific value of S_n, the instantaneous resistance implemented by the transistor can vary $\pm 5\%$ depending on the drain and source voltages applied. In order to take into account these variations, we have selected the minimum possible width $W = 0.4\mu$m. Then, we have swept L until finding that value which makes the average resistance of the transistor for all the possible voltages at the drain and source terminals rendering the optimum sum, i. e. $V_{min} + V_{max}$, equals to 100kΩ.



(a) (b) (c) (d)

Figure 6. (a) Original image, (b) MOS-diffused image at the instant of maximum error, (c) image diffused by the corresponding ideal RC network, (d) absolute error normalized to maximum individual pixel error

The result is $L = 7.54\mu$m[§].

Once designed the elementary transistor, let us suppose that the initial voltage at the capacitors is proportional to the image intensity displayed at Fig. 6(a). The MOS-based RC network runs the diffusion over these initial voltages in parallel with its ideal counterpart in order to be compared. The deviation is measured via the RMSE (Fig. 7(a)) and reaches a maximum soon after the beginning of the diffusion process. The state of the corresponding nodes in both networks at this point, displayed in Figs. 6(b) and 6(c), is perceptually equivalent. The maximum observed RMSE for the complete image is 0.5%, while the maximum individual pixel error is 1.76%. The RMSE remains below 0.6% — equivalent resolution between 6 and 7 bits — even introducing an exaggerated mismatch (10%) in the transistors' $V_{T_{n0}}$ and μ_n (Fig. 7(b)), confirming thus the robustness to mismatch predicted in Section 3.

6. ON-CHIP GAUSSIAN FILTERING

This section describes the programmable focal-plane Gaussian filtering operation performed by a QCIF resolution smart CMOS imager [21] manufactured in the AMS 0.35μm CMOS-OPTO 3.3V process. This CMOS process does not incorporate any special device for image sensors. Indeed, it only differs from the standard AMS 0.35μm process in an anti-reflective coating and an EPI wafer which reduces the dark current. The chip implements a massively parallel focal-plane processing array which can output different kinds of simplified representations from an image sequence at very low energy cost. Focal-plane processing is performed on every single image frame. Only spatial information is employed. Temporal variations between consecutive images in a sequence are not taken into account. The main characteristics of the chip are summarized in Table II. We can compare this prototype with other realizations of programmable bandwidth focal-plane Gaussian filtering, like [11] and in [18]. A figure of merit which contemplates their major features can be computed:

[§]This transistor length lies out of the physical design grid, that fixes the minimum feature size to be 0.05μm. We are using it here as illustrative of the design procedure.

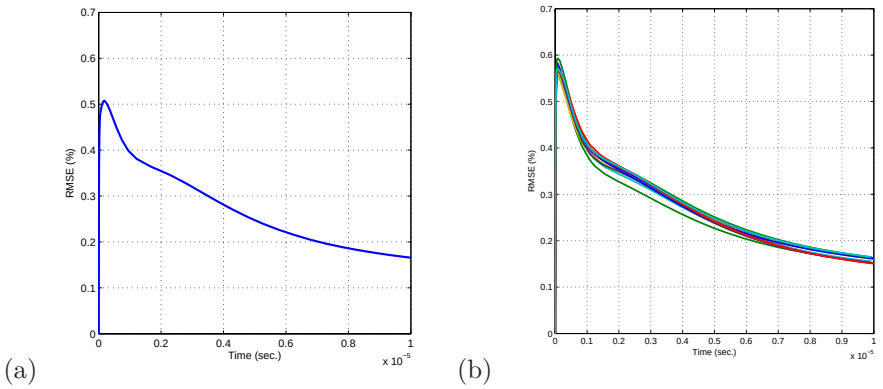


Figure 7. RMSE of the MOS-based grid state vs. ideal RC grid state: (a) w/o mismatch, (b) Monte Carlo with 10% mismatch

FOM = (Power · Area)/(Spatial resolution · Throughput). Thus, for the prototype of this paper, the FOM, measured in pJ·mm^2/px·Sa, results in 84.5 whereas it is 1.49×10^6 for [11], and 95.1 for [18].

The functionality of the array is mostly based on the fully-programmable time-controlled Gaussian filtering carried out by a RC network. In Section 2 we defined the width of such filtering as $\sigma = \sqrt{2t/\tau}$. Since this width depends on the quotient between the time interval which the network is permitted to evolve and the time constant of the network, their values must be correlated in order to achieve a high degree of programmability over the filtering. It in turn establishes several tradeoffs for the design of the corresponding circuitry. On one hand, the larger the value of τ the coarser the necessary time control of the network dynamics to render a certain value of σ, making simpler the circuitry for this control. Besides, larger values of τ need more area for the implementation of the elementary transistor and elementary capacitor of the network, being therefore more robust to mismatch. On the other hand, the area consumed by these components has dramatic consequences for the size of the array as they must be included at each and every elementary cell of the focal plane. From this point of view, a small value of τ results more adequate. The problem is that a finer temporal control of the network dynamics would be mandatory in such a case, even forcing the internal, i.e. on-chip, generation of the pulses which control the evolution of the grid. Otherwise, propagation delays could distort the resulting filtering.

6.1. DIFFUSION DURATION CONTROL

In order to reduce as much as possible the value of τ and therefore the size of the focal-plane processing array, we propose a method for a fine control of t based on an on-chip VCO. The first block of the diffusion duration control circuit is the VCO itself (Fig. 8(a)). It consists of a ring of pseudo-NMOS inverters in which the load current is controlled by 'Vbias_clk', thus modifying the propagation delay of each stage. This circuit provides an internal clock that will be employed to time pulses that add up to the final diffusion duration.

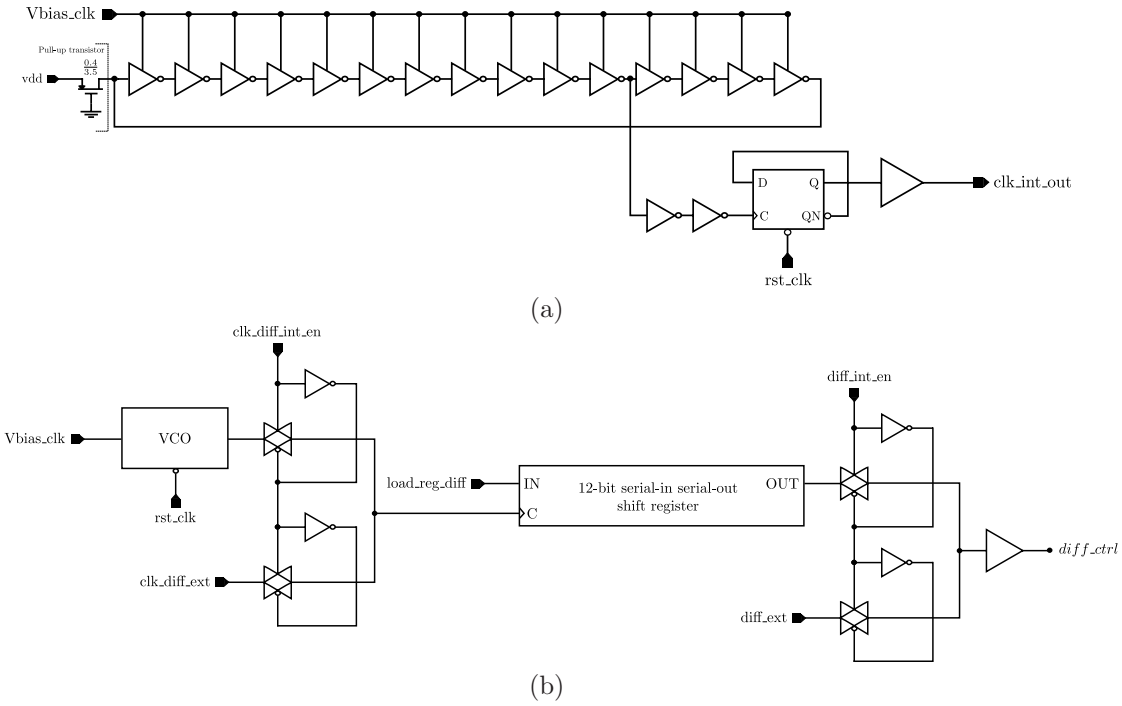


Figure 8. (a) 15-stage inverter ring VCO and (b) diffusion control logic

The main block of the diffusion control is a 12-stage shift register, Fig. 8(b). It will store a chain of 1's indicating how many clock cycles the diffusion process will run. The clock employed for this will be either external or the already described internal VCO[¶]. The output signal, $diff\_ctrl$, is a pulse with the desired duration of the diffusion, t, that is inverted and delivered to the gates of pMOS resistors. Thus, t depends on two parameters, namely: N_1, which is the number of logic '1's within the bit string, and f_{CLK}, the frequency of the clock. In this way, $t = N_1/f_{CLK}$. A minimum step of around $t = 6.66$ns can be achieved.

6.2. MOS-BASED RC NETWORK

Once t_{min} has been set, the design of the RC network can be addressed. Our objective is to implement a value of τ around one order of magnitude greater than t_{min}. This means that Gaussian filters with widths below $\sigma = 1$ must be easily achieved. The design methodology applied for the elementary MOS resistor is exactly the same than in Section 5. The result is depicted in Fig. 9. The MOS-based elementary capacitor has a nominal value of 1pF whereas the elementary MOSFET implements, for typical mean conditions (TM), a resistance of 85kΩ.

[¶]The aim of the internal VCO is to reach a better resolution of the diffusion time than an external clock. For loading the appropriate sequence into the register, an external, and slower, clock is usually preferred.

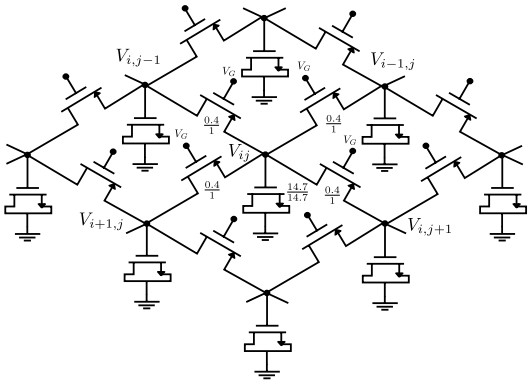


Figure 9. On-chip RC network

A key point which was briefly commented in Section 3 is that $S_{n_{opt}}$ varies, due to its dependence of k_n, across the design space delimited by the corners of the process. Or, equivalently, for a fixed $S_{n_{opt}}$, the resistance implemented by the transistor varies according to the value of k_n. In our case, this value of the resistance ranges from 49kΩ (WP corner) to 148kΩ (WS corner). The problem therefore is to know the exact value of k_n in a specific implementation. We propose a calibration process in order to solve this. This calibration process is carried out by means of two pairs of neighbor pixels whose initial voltage can be externally set. One pair is located at the upper left corner of the array whereas the other one is at the upper right corner. It permits to take the across-die variations into account. In order to obtain the value of τ, one of the two pixels at every pair is set to V_{min} whereas the corresponding neighbor is set to V_{max}. Successive diffusion steps are programmed until reaching the steady state. After each diffusion step, the pixel values are read out in order to be compared off-line to an ideal diffusion over the same initial conditions. This ideal diffusion, where the sum of the drain and source voltages equals at every time instant $V_{min} + V_{max}$, should be perfectly emulated by the MOS-based diffusion according to Eq. (27). Obviously, second-order effects will cause deviations, but a least square fitting of the pixel values read out from the chip with respect to the ideal diffusion will permit to know the average τ, similarly to how the average resistance is obtained in Section 5. The result for the upper left corner is depicted in Fig. 10, where a minimum RMSE of 2.26% is obtained for $\tau = 72.4$ns. In the upper right corner, a minimum RMSE of 0.58% is reached for $\tau = 69.8$ns. These values make perfect sense according to the range of possible values of the MOS resistance above mentioned. The final value of τ considered for the whole array will be the average of the extracted values, that is, $\tau = 71.1$ns.

6.3. EXPERIMENTAL RESULTS

Once τ is calibrated, any on-chip Gaussian filter can be compared to its ideal counterpart obtained by solving the spatially-discretized diffusion equation. Thus, a single image is captured. This image is converted to the digital domain and delivered through the output bus. Ideal Gaussian filters with increasing widths are applied over this image. The same filters are

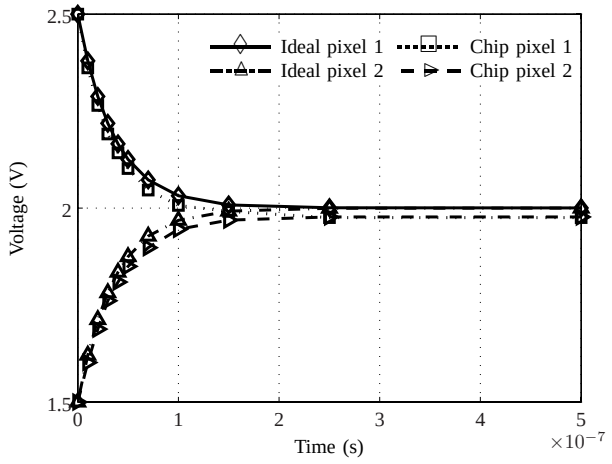


Figure 10. Calibration of τ at the upper left corner

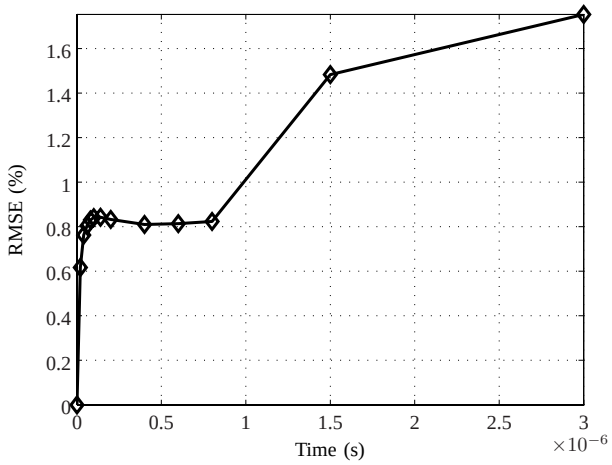


Figure 11. Evolution of the RMSE for on-chip Gaussian filtering with respect to the ideal case

implemented on-chip by programming the adequate time intervals of diffusion. After every on-chip filtering, the resulting image is converted to digital and delivered to the test instruments to be compared to its ideal counterpart generated by MATLAB® (Fig. 11). A total of 12 different filters have been applied over the original captured image. Six of them are represented in Fig. 12 (first row) and compared to the ideal images (second row). The last row contains a pictorial representation of the error, normalized in each case to the highest measured error on individual pixels, which are 0%, 24.99%, 19.39%, 6.17%, 3.58% and 6.68%, respectively. It can be seen how noise eventually becomes the dominant error for the largest values of σ. Keep in mind

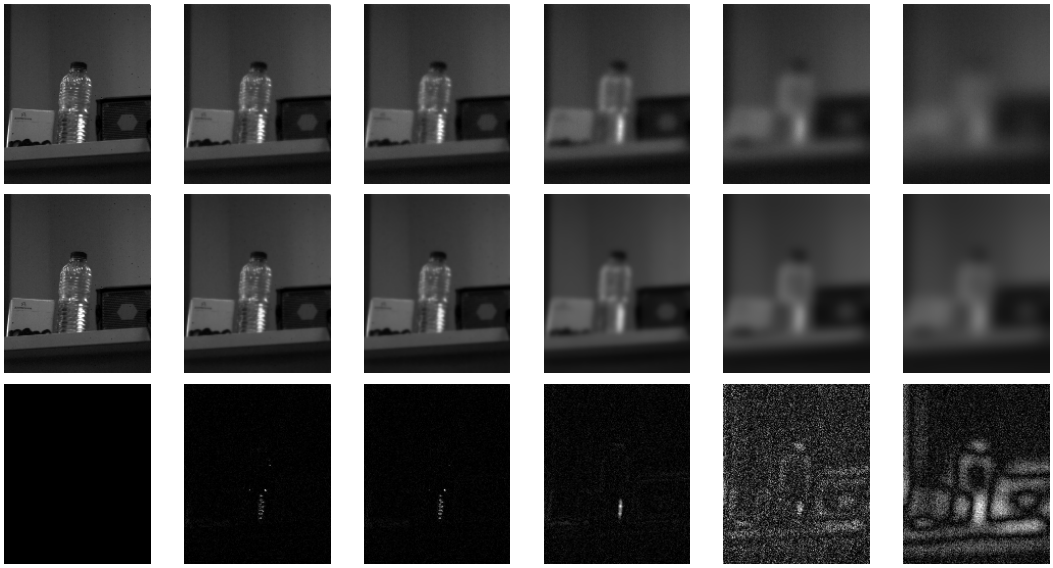


Figure 12. Comparative of Gaussian filtering for different values of σ. The first row corresponds to the images extracted from the chip, the second one corresponds to their ideal counterparts and finally the third one corresponds to their normalized difference.

that this noise is added to each image coming from the chip because of the readout mechanism. On the contrary, the noise present at the initial image is filtered by the ideal Gaussian filters applied off-chip. The key point here is that the error is kept under a reasonable level despite no FPN post-processing is carried out. This fact together with the efficiency of the focal-plane operation is crucial for artificial vision applications under strict power budgets. Furthermore, the accuracy of the processing predicted by simulation is very close to that of the filters with small σ, where noise is not dominant yet. It validates all the steps of the design methodology proposed along this paper, specially the use of the classical first-order approximation for a transistor biased in the triode region. Despite its simplicity, it has proved to be enough for a robust design.

7. CONCLUSIONS

A methodology for the optimal design and VLSI implementation of focal-plane Gaussian filtering has been presented. It is based on the fine control of the diffusion dynamics of a MOS-based RC network. The inclusion of transistors instead of true resistors achieves a much more area-efficient implementation. Besides, the control of their gate voltages permits to stop the diffusion when required, implementing thus filters with programmable width. Together with this programmability, the most remarkable features of the methodology proposed are

the robustness to mismatch and the accuracy of the filtering despite the nonlinearities of the transistors. This makes the VLSI implementation of MOS-based RC networks a very useful and efficient tool for early vision processing.

APPENDIX A

Firstly, time derivative is applied to Eq. (12), obtaining:

$$\frac{d\epsilon}{dt} = \frac{1}{2}\frac{d}{dt}\left[V_1'(t) - V_2'(t)\right] - \frac{1}{2}\frac{d}{dt}\left[V_1(t) - V_2(t)\right] \tag{28}$$

Taking into account now that, from Eq. (6):

$$\frac{d}{dt}\left[V_1'(t) - V_2'(t)\right] = -2\frac{V_1'(t) - V_2'(t)}{R_M C} \tag{29}$$

and from Eq. (5):

$$\frac{d}{dt}\left[V_1(t) - V_2(t)\right] = -2\frac{V_1(t) - V_2(t)}{RC} \tag{30}$$

we can rewrite Eq. (28) as:

$$\frac{d\epsilon}{dt} = \frac{V_1(t) - V_2(t)}{RC} - \frac{V_1'(t) - V_2'(t)}{R_M C} \tag{31}$$

where, defining $\tau = RC$, we have:

$$\tau\frac{d\epsilon}{dt} = V_1(t) - V_2(t) - \frac{R}{R_M}\left[V_1'(t) - V_2'(t)\right] \tag{32}$$

Finally, considering that, according to Eq. (11):

$$V_1(t) - V_2(t) = V_1'(t) - V_2'(t) - 2\epsilon(t) \tag{33}$$

and substituting Eq. (8) and Eq. (9) in Eq. (32), we obtain:

$$\tau\frac{d\epsilon}{dt} = \left[V_1'(t) - V_2'(t)\right]\frac{V_{10} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} - 2\epsilon(t) \tag{34}$$

where we have also applied our initial assumption regarding the initial conditions, that is, $V_{10} = V_{10}'$ and $V_{20} = V_{20}'$.

APPENDIX B

The absolute extremes are at the boundaries of the domain for (V_{10}, V_{20}). The initial assumption $V_{10} > V_{20}$ reduces the boundaries to be searched to only two. The first is that one set by making $V_{10} = V_{max}$, that is, according to Eq. 18:

$$\epsilon_{ext}|_{V_{10}=V_{max}} = \frac{1}{2}(V_{max} - V_{20})\frac{V_{max} + V_{20} - V_{S_{opt}}}{V_C - V_{S_{opt}}} r^{\frac{r}{1-r}} \tag{35}$$

from which calculating the derivative with respect to V_{20} and bearing in mind our assumption that $\partial r/\partial V_{20} \simeq 0$, we have:

$$\frac{d}{dV_{20}}\epsilon_{ext}|_{V_{10}=V_{max}} = \frac{1}{2}\frac{V_{S_{opt}} - 2V_{20}}{V_C - V_{S_{opt}}} r^{\frac{r}{1-r}} \tag{36}$$

and finally, by making this expression equal to 0, the location of the first absolute extreme is obtained, $(V_{10}, V_{20}) = (V_{max}, V_{S_{opt}}/2)$. The other boundary to search absolute extremes is that one set by making $V_{20} = V_{min}$. It means, according to Eq. 18, that:

$$\epsilon_{ext}|_{V_{20}=V_{min}} = \frac{1}{2}(V_{10} - V_{min})\frac{V_{10} + V_{min} - V_{S_{opt}}}{V_C - V_{S_{opt}}} r^{\frac{r}{1-r}} \tag{37}$$

We calculate now the derivative with respect to V_{10}, assuming $\partial r/\partial V_{10} \simeq 0$:

$$\frac{d}{dV_{10}}\epsilon_{ext}|_{V_{20}=V_{min}} = \frac{1}{2}\frac{2V_{10} - V_{S_{opt}}}{V_C - V_{S_{opt}}} r^{\frac{r}{1-r}} \tag{38}$$

and again, by making this expression equal to 0, the second absolute extreme is located at point $(V_{10}, V_{20}) = (V_{S_{opt}}/2, V_{min})$. Finally, by substituting these two points in Eq. 18, the absolute extremes of Eq. 19 are respectively obtained.

REFERENCES

1. Poggio T, Voorhees H, Yuille A. A regularized solution to edge detection. *J. of Complexity* 1988; **4**(2):106–123.
2. Mutch J, Lowe D. Object class recognition and localization using sparse features with limited receptive fields. *Int. J. of Computer Vision* 2008; **80**(1):45–57.
3. Jähne BE. Multiresolutional signal representation. In Chapter 4, *Handbook of Computer Vision and Applications. Volume 2: Signal Processing and Pattern Recognition*, Academic Press, 1999;
4. Babaud J, Witkin AP, Baudin M, Duda RO. Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1986; **8**(1):26–33.
5. Zárandy Á, Rekeczky C. 2D operators on topographic and non-topographic architectures implementation, efficiency analysis, and architecture selection methodology. *International Journal of Circuit Theory and Applications* 2010; n/a. doi: 10.1002/cta.681
6. Sotak GE, Boyer KL. The Laplacian-of-Gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics and Image Processing* 1989; **48**(2):147–189.
7. Mead C. *Analog VLSI and Neural Systems*. Addison-Wesley, 1989;
8. Raffo L, Sabatini SP, Bo GM, Bisio GM. Analog VLSI circuits as physical structures for perception in early visual tasks. *IEEE Transactions on Neural Networks* 1998; **9**(6):1483–1494.
9. Hui KF, Shi BE. Distortion in analog VLSI networks for image filtering. *IEEE Transactions on Circuits and Systems - I* 1999; **46**(10):1161–1171.
10. Shi BE. The effect of mismatch in current- versus voltage-mode resistive grids. *International Journal of Circuit Theory and Applications* 2009; **37**(1):53–65.
11. Kobayashi H, White JL, Abidi AA. An active resistor network for Gaussian filtering of images. *IEEE Journal of Solid-State Circuits* 1991; **26**(5):738–748.
12. Vittoz EA, Arreguit X. Linear networks based on transistors. *Electronic Letters* 1993; **29**(3):297–299.
13. Andreou AG, Boahen KA. A 590,000 transistor 48,000 pixel, contrast sensitive, edge enhancing, CMOS imager-silicon retina. *Proc. Conf. on Advanced Research in VLSI* 1995; 225–240.

14. Lenero-Bardallo JA, Serrano-Gotarredona T, Linares-Barranco B. A mismatch calibrated bipolar spatial contrast AER retina with adjustable contrast threshold. *International Symposium on Circuits and Systems* 2009; 1493–1496.
15. Shi BE, Chua LO. Resistive grid image filtering: input/output analysis via the CNN framework. *IEEE Transactions on Circuits and Systems - I* 1992; **39**(7):531–548.
16. Fernandez-Berni J, Carmona-Galan R. On the implementation of linear diffusion in transconductance-based cellular nonlinear networks. *International Journal of Circuit Theory and Applications* 2009; **37**(4):543–567.
17. Ni Y, Zhu YM, Arian B, Devos F. Yet another analog 2D Gaussian convolver. *IEEE Int. Symp. on Circuits and Systems (ISCAS)* 1993; **1**:192–195.
18. Ni Y. Smart image sensing in CMOS technology. *IEE Proc.-Circuits Devices Syst.* 2005; **152**(5):547–555.
19. Lindeberg T. Discrete scale-space theory and the scale-space primal sketch. *Ph. D. dissertation*, Royal Institute of Technology (Stockholm, Sweden), 1991;
20. Papoulis A, Unnikrishna S. *Probability, Random Variables and Stochastic Processes*, McGraw Hill, 2002;
21. Fernandez-Berni J, Carmona-Galan R. Robust focal-plane analog processing hardware for dynamic texture segmentation. *Proc. IEEE Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA)* 2010; 453–458.

V_G(V)	V_{T_n}(V)	$[V_{min}, V_{max}]$(V)	$\max e_{ext}$(mV)	$\min e_{ext}$(mV)	$V_{S_{opt_{num}}}$(V)	e_{max}(mV)	e_{min}(mV)	$V_{S_{opt}}$(V)	r	Equiv. resol. (bits)
3.3	0.8	[0,1.5]	30.43	-30.43	1.58	33.19	-26.74	1.5	[0.57,1.43]	4.5
3.3	0.8	[0,0.75]	6.17	-6.17	0.77	6.37	-5.83	0.75	[0.82,1.18]	5.8
3.3	0.8	[0.75,1.5]	9.61	-9.61	2.28	10.10	-8.81	2.25	[0.73,1.27]	5.2
1.8	0.5	[0,1]	30.61	-30.61	1.08	34.26	-24.93	1	[0.38,1.62]	3.9
1.8	0.5	[0,0.5]	5.62	-5.62	0.51	5.82	-5.17	0.5	[0.76,1.24]	5.4
1.8	0.5	[0.5,1]	10.79	-10.79	1.53	11.82	-9.40	1.5	[0.55,1.45]	4.4

Table I. Numerical verification of the approximations realized.

Technology	0.35μm CMOS 2P4M
Vendor (Process)	Austria Microsystems (C35OPTO)
Die size (with pads)	7280.8μm $\times$ 5780.8μm
Cell size	34.07μm $\times$ 29.13μm
Fill factor	6.45%
Resolution	QCIF: 176×144 px
Photodiode type	n-well/p-substrate
Power supply	3.3V
Signal range	[1.5V,2.5V]
FPN	0.72%
PRNU (50% signal range)	2.42%
Sensitivity	0.15V/(lux·s)
Power consumption (worst case)	5.6mW@30fps
ADC throughput	0.11MSa/s (9μs/Sa)
Internal clock freq. range	0.5-150MHz

Table II. Summary of prototype chip.