Hardware-Aware Performance Evaluation for the Co-Design of Image Sensors and Vision Algorithms

C. Villegas-Pachón, R. Carmona-Galán, J. Fernández-Berni and Á. Rodríguez-Vázquez
Institute of Microelectronics of Seville (IMSE-CNM),
CSIC-University of Seville, Spain
Email: rcarmona@imse-cnm.csic.es

***Abstract*— The top-down approach to system design allows obtaining separate specifications for each subsystem. In the case of vision systems, this means propagating system-level specifications down to particular specifications for e. g. the image sensor, the image processor, etc. This permits to adopt different design strategies for each one of them, as long as they meet their own specifications. This approach can lead to over-design, which is not always affordable. Conversely, if higher-level specifications are too tight, they can lead to impossible specifications at the lower levels. This is certainly the case for embedded vision systems in which high-performance needs to be paired with a very restricted power budget. In order to explore alternative architectures, we need tools that allow for simultaneous optimization of different blocks. However, the link between low-level non-idealities and high-level performance is missing. CAD tools for the design and verification of analog and mixed-signal integrated circuits are not well suited for the simulation of higher-level functionalities. Our approach is to extract relevant data from circuit-level simulation and to build an OpenCV model to be employed in the design of the algorithm. The utility of this approach is illustrated by the evaluation of the effect of column-wise and pixel-wise FPN at the sensor on the performance of Viola-Jones face detection.**

Keywords—CMOS Image Sensors; HW/SW co-design; OpenCV; Embedded Vision Systems

I. INTRODUCTION

CAD tools for the design of analog and mixed-signal integrated circuits are certainly limited for the simulation of system-level functionalities. Of course, higher-level processing tasks cannot and should not be handled at transistor-level accuracy —especially if one does not have supercomputing facilities [1]. Because of this, higher-level models and architectural descriptions have been introduced for behavioral simulation [2]. This connects the work of system designers with that of implementation designers. However, in the specific field of embedded vision systems, the gap between application engineers and physical implementation designers has not been filled yet. Several attempts have been reported to secure the flow between vision application development software and integrated circuit design tools [3] [4]. Yet application engineers and system designers have different worldviews. In order to make them compatible, a top-down approach is typically employed. Therefore, application engineers generate system-level specifications that sooner or later propagate down the implementation hierarchy [5]. Each subsystem is designed to meet its own specifications, derived from system-level specs. For the conventional model in computer vision, in which image capture and processing are completely separated tasks, this does not represent a major difficulty. Chip designers will work for a particular set of specifications, i. e. spatial and temporal resolution, power consumption, etc. At the other end, computer scientists will take care of the algorithm once they receive their pictures fitting to the prescribed specifications. The result of this mindset is an architecture that is theoretically universal, although may not be capable of solving every problem. In the first place, this approach can lead to over-design. In the second, technological limits can be reached and specifications may end up being unrealizable. In the case of embedded vision systems, where high-performance and power-efficiency need to be combined, this approach can easily lead to no results.

An alternative approach is to explore the interdependences between elements at the different levels of the hierarchy, in order to find optimal combinations. For this to be implemented, the optimization loop must incorporate a detailed, yet manageable, description of the system [6]. That includes parameters describing the algorithm performance and, at the same time, an accurate account of the implementation non-idealities. Let us emphasize that, in embedded vision systems like smart cameras, computational efficiency is generally provided by the appropriate partition of algorithm tasks, the parallelization of heavy loads, the use of distributed resources and the incorporation of close-to-sensor processing and memory elements [7]. Sometimes these actions will require the design of specific circuit blocks and ad-hoc image sensing strategies. All of this needs to be worked out at transistor level, but at the same time, their effect in the overall performance of the algorithm needs to be quickly and accurately evaluated in order to guide the design flow [8]. Our proposal is to make use of the flexibility and versatility of an environment like OpenCV to incorporate hardware non-idealities to the evaluation of the algorithm performance. One of the major attractions of this approach is that computer vision experts will be able to consider deviations caused by the physical implementation when designing and fine-tuning their vision algorithms without having to develop any expertise in chip design and IC CAD tools. As an example, we have modeled a

This work has been funded by the Spanish Government through projects TEC2015-66878-C3-1 MINECO (European Region Development Fund, ERDF/FEDER), IPT-2011-1625-430000 MINECO, IPC- 20111009 CDTI (ERDF/FEDER), Junta de Andalucía through project TIC 2338-2013 CEICE, and the Office of Naval Research (USA) N000141410355.

3T-APS image sensor, incorporating deviations in the parameters of critical transistors, following the EMVA 1288 standard [9]. The utility of this approach is then illustrated by the evaluation of the effect of column-wise and pixel-wise fixed-pattern-noise (FPN) on the performance of Viola-Jones face detection [10].

II. Modeling of Physical Implementation Errors

The EMVA standard No. 1288 [9] has been defined to characterize image sensors and camera chips. It is based on a linear mode (Fig. 1) in which photons (n_p) are absorbed and converted to electrons (n_e) according to the quantum efficiency (QE). These electrons, together with those that are product of noise and other device non-idealities (n_d), are translated into a voltage (y) by means of a conversion gain (CG). Finally, the output voltage is converted to a digital number (y_{DN}) by means of an ADC, that introduces a quantization noise (σ_q).

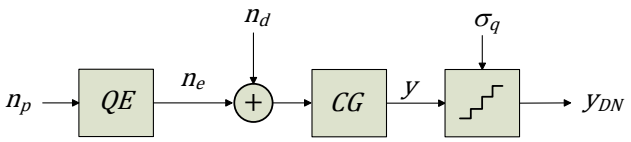


Fig. 1. EMVA standard 1288 image sensor model

The average value of the pixel´s output voltage is obtained from the sum of the average photogenerated electrons and the average number of electrons generated by other means, what is called dark signal:

$$\bar{y} = CG(\overline{n_e} + \overline{n_d}) \tag{1}$$

Because of temporal and spatial noise contributions, the variance of the output voltage is given by:

$$\sigma_y^2 = CG^2(\sigma_e^2 + \sigma_d^2) + \sigma_q^2 \tag{2}$$

where signal dependent contributions are included in σ_e, like photon shot noise, and those contributions related to the reset transistor, the readout circuitry and dark current noise, are grouped into σ_d. Besides, σ_q stands for the quantization noise.

In the case of a 3T active pixel sensor [11] (Fig. 2), the conversion gain corresponds to the ratio between the elementary charge and the sensing capacitance:

$$CG = q/C_{\text{pix}} \tag{3}$$

The KTC noise coming from the thermal noise of the reset transistor, M_{RST}, is one of the main sources of temporal noise. Expressed in number of electrons, its contribution amounts to:

$${\sigma_{\text{KTC}}}^2 = k_B T C_{\text{pix}}/q^2 \tag{4}$$

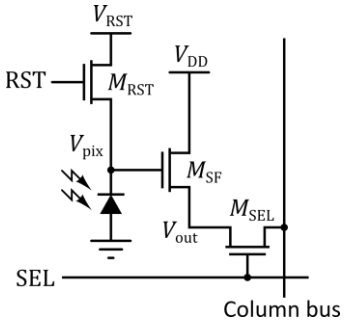


Fig. 2. 3T Active pixel sensor

The thermal noise of the readout transistor, M_{SF}, is generated after conversion from electrons to voltage, therefore, expressed in electrons is:

$${\sigma_{\text{SF}}}^2 = (1/CG^2)(k_B T/C_{\text{col}}) \tag{5}$$

Another important contribution is the fixed-pattern noise (FPN). In the model already described, FPN introduces additional components to both σ_e and σ_d. These contributions constitute the so-called photoresponse non-uniformity (PNRU) and dark-signal non-uniformity (DSNU), respectively. In order to illustrate the effects of non-idealities in the system performance, and without loss of generality, we are going to consider two different contributions to DSNU. They are related with the operation of the source follower, M_{SF}, which in a first approximation provides an output voltage given by:

$$V_{\text{out}} = V_{\text{pix}} - V_{T_{\text{SF}}} - \sqrt{I_B/\beta_{\text{SF}}} \tag{6}$$

in this simplified model, the spatial variations of the transistor threshold voltage ($V_{T_{\text{SF}}}$) introduce an offset in the output voltage that is different for each pixel. In terms of noise contributions, it can be incorporated to the model as:

$${\sigma_{\Delta V_{T_{\text{SF}}}}}^2 = (1/CG^2)\left(A_{V_T}^2/W_{\text{SF}}L_{\text{SF}}\right) \tag{7}$$

following Pelgrom's mismatch model [12]. Also, variations of the column bus current (I_B) introduce an additional offset. Its contribution in electrons to σ_d is:

$${\sigma_{\Delta I_Q}}^2 = \frac{I_B}{4\beta_{\text{SF}}CG^2}\left(\frac{A_\beta^2}{W_B L_B} + 4\frac{\beta_B}{I_B}\frac{A_{V_T}^2}{W_B L_B}\right) \tag{8}$$

This model is certainly incomplete. The dependence of the saturation current of M_{SF} on V_{out} —not contemplated in Eq. (6)— and the variations on its transconductance parameter and its substrate effect constant end up in a gain error — PRNU— that translates into a contribution to σ_e in the model. In any case, the already mentioned terms suffice to illustrate the procedures.

III. Behavioral Simulation in OpenCV

The Open Source Computer Vision Libray [13]—commonly known as OpenCV— is a BSD-licensed library for computer vision and machine learning. It contains more than 2500 optimized algorithms for object detection, object tracking, stereo vision, etc. It is one of the most popular tools in the computer vision industry; we have therefore incorporated the already described sensor model into OpenCV. For example, in order to perform object detection we will make use of the already pre-trained classifiers. Their data are stored in XML files at the `opencv/data/haarcascades/` folder. We only need to implement an image capture block (Fig. 3) that retrieves images from the dataset and processes them according to the sensor data, also stored in a XML file. After that, the preprocessed image enters the object detection routines.

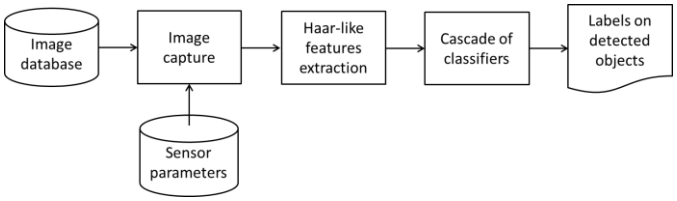


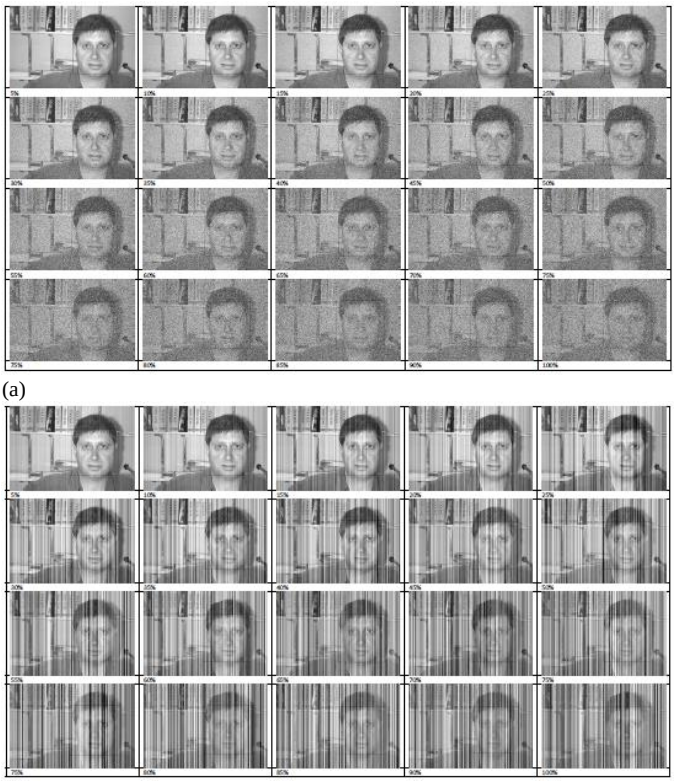
Fig. 3. Simulation of hardware non-idealities in OpenCV dataflow

Of course, the object detection cascade can be re-trained to adapt to the peculiarities of the defined sensor. However, in our experiments we have employed the already available weights for face detection.

IV. Case study: Influence of FPN

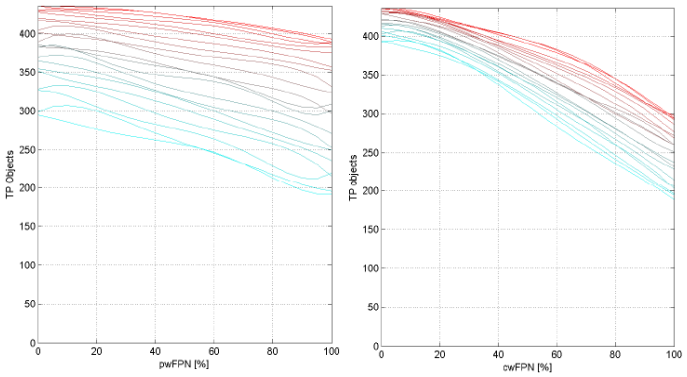
As an illustration of the possibilities of incorporating hardware non-idealities to the evaluation of the algorithm performance, we have chosen to test the influence of pixel-wise (pw) and column-wise FPN on the precision and recall rates of the Viola-Jones face detector included in the OpenCV library. Precision is the fraction of true positive detections from all objects detected —including false positives— and recall is the fraction of true positive detections from all the relevant objects in the dataset, were they detected or not. The images employed to test the algorithm is the Caltech Frontal Face Dataset [14], containing 450 896×592-pixel images belonging to 27 different people. Fig. 4 displays an example of an image of the dataset affected by growing values of both pixel-wise and column-wise FPN.

The consequences of applying different values for the mismatch in the threshold of the pixels' source follower, which results in a pw-FPN, and the column' bias current, that results in a cw-FPN, can be seen in Figs. 5 to 9. In all these graphs, pw-FPN is varied for a fixed value of the cw-FPN in the (a) plot, and the other way around in the (b) plot. Within each graph, the darker/reddish lines correspond to the smallest values of the alternative parameter —which ranges from 0% to 100% in steps of 5%— while the lighter/bluish to the largest.



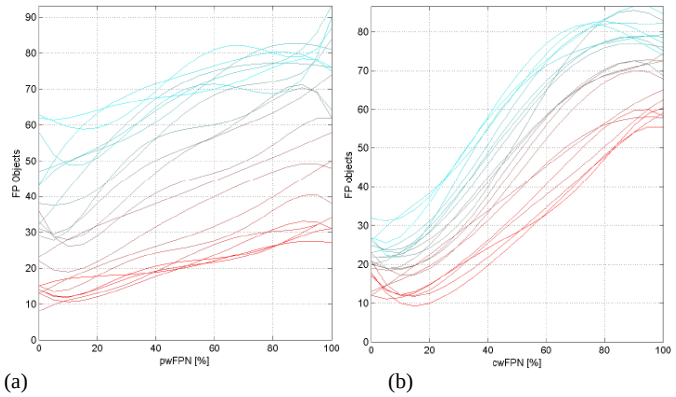
(a) (b)

Fig. 4. Output images for growing values of (a) pw-FPN and (b) cw-FPN



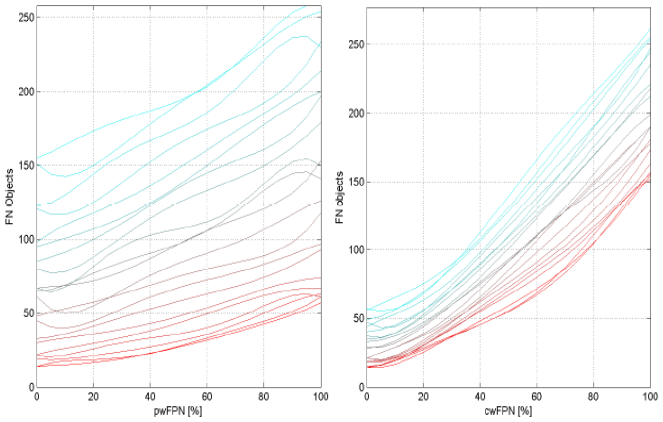
(a) (b)

Fig. 5. True positive detections as a function of (a) pw- FPN and (b) cw-FPN



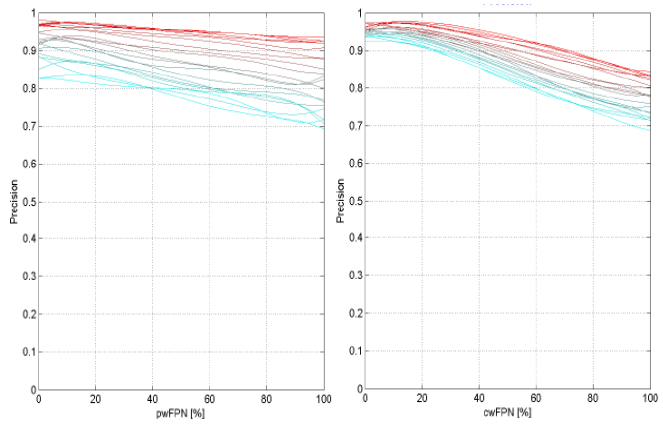
(a) (b)

Fig. 6. False positive detections vs. (a) pw- FPN and (b) cw-FPN



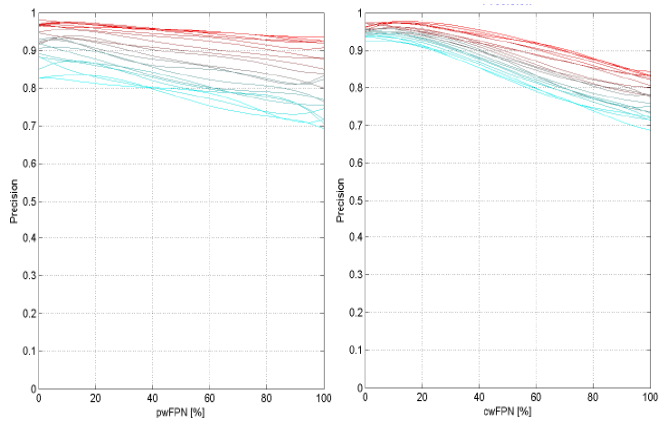
(a) (b)

Fig. 7. False negative detections vs. (a) pw- FPN and (b) cw-FPN



(a) (b)

Fig. 8. Precision vs. (a) pw- FPN and (b) cw-FPN



(a) (b)

Fig. 9. Recall vs. (a) pw- FPN and (b) cw-FPN

It can be seen that for a particular value of cw-FPN, by increasing the value of pw-FPN —what can be seen in (a) plots—, the degradation of different performance indexes like the number of true positive detections, false positives, false negatives, precision and recall; is much weaker than the variation obtained by fixing pw-FPN and increasing cw-FPN —which is depicted in (b) plots. It can be concluded that cw-FPN has a stronger deteriorating effect in the object detection algorithm than pw-FPN has. Therefore, the major circuit design efforts should be put in eliminating cw-FPN. Or equivalently, from the point of view of the algorithm, a new cascade of classifiers could be trained to cope with the artifacts caused by a large cw-FPN. Either way, the influence of hardware non-idealities are evidenced with this methodology, allowing comprehensive optimization and co-design of image sensor hardware and algorithm parameters.

V. Conclusions

The design of an embedded vision system cannot always be accomplished by a top-down approach. Optimization involving design parameters at different levels may be required. It is possible to work at algorithm levels and still incorporate a low-level description of the image sensor non-idealities in order to evaluate their influence on the vision algorithm performance. A feasible way to do it is to incorporate the necessary models into OpenCV library. The major advantage is that it is well-known by the vision application developer community. Co-design of vision hardware and software is in this way possible. As an example, we have displayed the effect of pw- and cw-FPN on precision and recall of the Viola-Jones algorithm.

References

[1] R. A. Saleh, K. A. Gallivan, et al. “Parallel circuit simulation on supercomputers”. *Proceedings of the IEEE*, Vol. 77, No. 12, pp. 1915-1931, Dec. 1989.

[2] K. Kundert, and O. Zinke, *The Designer’s Guide to Verilog-Ams.* Springer, 2004.

[3] M. Willems, “Developing Embedded Vision Systems”. *Synopsys Insight Newsletter*, No. 3, 2013.

[4] C. Rowen, “Designing and Selecting Instruction Sets for Vision”. *Embedded Vision Summit*, San Jose, CA (USA), May 2015.

[5] D. D. Gajski, “System-level synthesis: From specification to transaction level models”. *Int. Conf. on Communications, Circuits and Systems, (ICCCAS 2009)*, pp. 1134-1138, July 2009.

[6] P. Reichel, C. Hoppe, J. Döge and N. Peter, “Simulation environment for a vision-system-on-chip with integrated processing”. *Proc. of the 9th Int. Conf. on Distributed Smart Cameras* (ICDSC’15), pp. 20-25, Seville, Spain, September 2015.

[7] F. Berry, R. Kleihorst, R. Carmona-Galán (eds.) *Special Issue on Smart Camera Architecture*, *Journal of Systems Architecture*, Vol. 59, No. 10, Part A, pp. 817-920, November 2013.

[8] J. Fernández-Berni et al. “Bottom-up performance analysis of focal-plane mixed-signal hardware for Viola-Jones early vision tasks”. *Int. J. of Circuit Theory and Apps.* Vol. 43, No. 8, pp. 1063-1079, Aug. 2015.

[9] EMVA, *Standard for Characterization of Image Sensors and Cameras*, European Machine Vision Association, No. 1288, Release 3.0. November 29, 2010

[10] P. Viola, M. Jones, “Rapid object detection using a boosted cascade of simple features”. *Proc.of the IEEE CS Conf. on Computer Vision and Pattern Recognition (CVPR 2001)*, Vol. 1, pp. I.511-I.518, 2001

[11] J. Nakamura, *Image sensors and signal processing for digital still cameras*. CRC Press, Taylor & Francis Group, Boca Raton, FL, 2006.

[12] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors", *IEEE J. Solid-State Circuits*, Vol. 24, No. 5, pp. 1433-1440, Oct. 1989.

[13] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, Octubre 2008.

[14] M. Weber, *Frontal face dataset*. California Institute of Technology, http://www.vision.caltech.edu/Image_Datasets/faces/faces.tar