



## Image Toolbox for CMOS Image Sensors Fast Simulation

By David Navarro, Zhenfu Feng & Ian O'Connor

*Université de Lyon*

*Abstract* - This paper presents a new toolbox, dedicated to image sensors designers. It permits fast analog simulations with a novel parametric simulation method. Image sensor matrixes are composed of millions of pixels. Such architectures are long to simulate, if not impossible; and input/ output data management is complex. A graphical toolbox has been developed in Cadence Analog Design Environment (ADE) to overcome these problems. That toolbox has been completely written in Cadence SKILL language. It permits to manipulate images as input, launch parametric analysis on a single pixel, and interpolate results for all pixels in the matrix. It then automatically generates images at output, in order to check the quality of microelectronic design. Low level aspects can also be analyzed at system (image) level.

*Keywords* : image sensor, microelectronic design, APS, simulation, modeling.

*GJCST-F Classification*: 1.4.8



IMAGE TOOLBOX FOR CMOS IMAGE SENSORS FAST SIMULATION

*Strictly as per the compliance and regulations of:*



RESEARCH | DIVERSITY | ETHICS

# Image Toolbox for CMOS Image Sensors Fast Simulation

David Navarro <sup>α</sup>, Zhenfu Feng <sup>σ</sup> & Ian O'Connor <sup>ρ</sup>

**Abstract** - This paper presents a new toolbox, dedicated to image sensors designers. It permits fast analog simulations with a novel parametric simulation method. Image sensor matrixes are composed of millions of pixels. Such architectures are long to simulate, if not impossible; and input/output data management is complex. A graphical toolbox has been developed in Cadence Analog Design Environment (ADE) to overcome these problems. That toolbox has been completely written in Cadence SKILL language. It permits to manipulate images as input, launch parametric analysis on a single pixel, and interpolate results for all pixels in the matrix. It then automatically generates images at output, in order to check the quality of microelectronic design. Low level aspects can also be analyzed at system (image) level.

**Keywords** : image sensor, microelectronic design, APS, simulation, modeling.

## I. INTRODUCTION

Image sensors in standard CMOS technology are now a well established alternative to the CCD image sensors technology. Indeed, process maturation, integration possibilities and low power consumption make CMOS image sensor widespread. Moreover, recent 3D technologies focused researchers and industry on new image sensor architectures and 3D floorplanning [1] [2].

CMOS image sensors are electronic systems that are composed of analog blocks (pixel matrix, noise reduction block (CDS: Correlated Double Sampling), column amplifier), digital blocks (controller and decoders), and mixed blocks (ADC: Analog to Digital Converter) [3]. Fig. 1 details a basic CMOS image sensor floorplan.

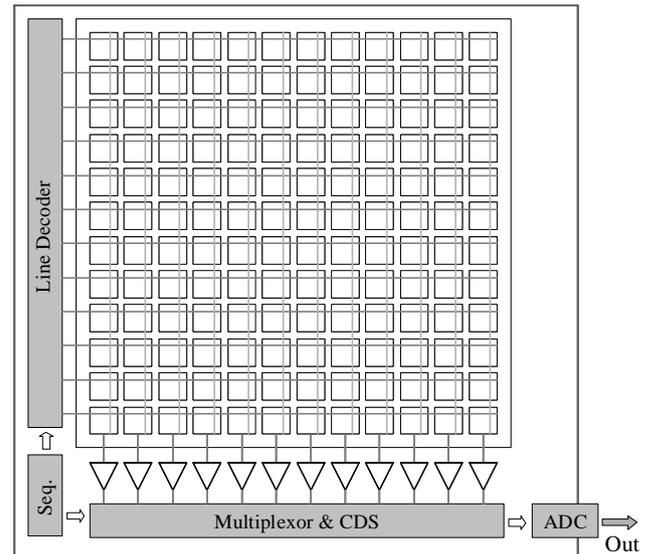


Figure 1 : Classical CMOS image sensor floorplan

## II. PROBLEMS IN IMAGE SENSOR SIMULATIONS AND STATE OF ART

A first major problem in these matrix structures is density: it is difficult to make system-level analysis because of the multiplicity of inputs and outputs. Indeed, in a classical pixel, a 3-transistors active pixel, also called 3-T APS [3], shown in Fig.2, three inputs are necessary. A "reset" signal –for initialization-, a "select" signal –for reading- and luminosity are required. The two digital control signals, "reset" and "select", have precise timings. These signals are common for each line, and have to respect an interlaced sequence. In a M x N matrix, M lines have to be considered to generate these signals. The major problem comes in fact from the third input: M x N luminosity inputs have to be considered.

Author <sup>α</sup> : Université de Lyon; Institut des Nanotechnologies de Lyon INL-UMR5270, CNRS, Ecole Centrale de Lyon, Ecully, F-69134, France. E-mail : david.navarrohere@eclyon.fr

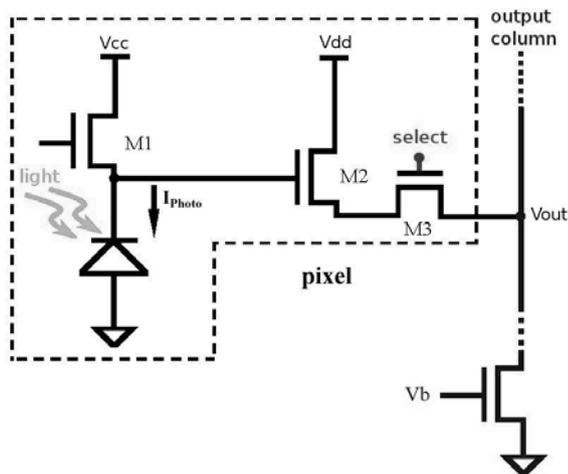


Figure 2 : Classical CMOS 3-T Active Pixel Sensor (APS)

Luminosity is often set as a constant value, anyway classical pixels integrate current over an exposure time (also called integration time). Depending on the photodiode model, the luminosity is set with a current source that emulates the photocurrent if a simple equivalent schematic is drawn, or luminosity can be considered if model is of upper level (for example in Verilog-A or VHDL-AMS language [4]). Of course, it is mandatory to simulate different values on pixel to realistically simulate the sensor characteristics. The problem is how to set easily thousands or millions of design variables. The same problem exists at sensor output: it is difficult to manage and analyze millions of analog values. In classical CMOS image sensors, outputs are voltages, sometimes ADC output. The second major problem is simulation time of image sensors, as Fig. 3 shows.

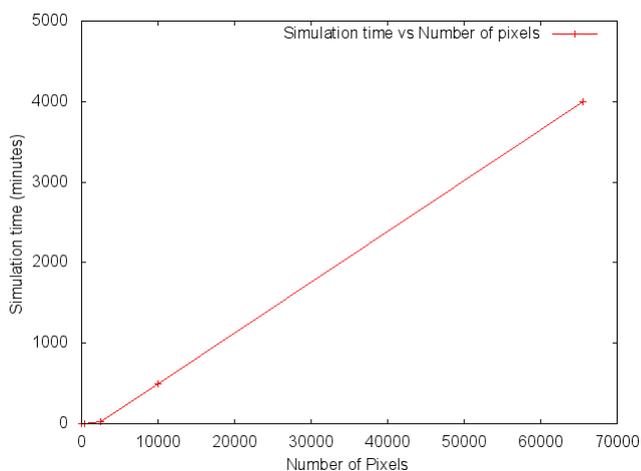


Figure 3 : Simulation time of images sensors with SPICE classical simulations

As a consequence, designers rarely simulate full imagers at a time. Blocks are validated separately, and small matrixes are used to validate the behavior.

Testchips (ASICs) are also required to properly characterize pixels characteristics. This toolbox is proposed to help designers running "image sensor – level" simulations. Available image sensor simulators are TCAD and focus on physical and FDTD simulations [5] [6]. On the other hand, image processing toolbox exist [7]. ECAD image sensor simulators are also missing.

Some high-level models –for example VHDL-AMS or MATLAB- have been developed [5], but it appears that the gap between a real analog structure and a high-level model make it uneasy to use within an optimization work. Experience showed that analog designers trust better in a level 53 SPICE model than in a third-party high-level model. If modeling and simulation tools are not integrated in classical framework, they cause another drawback: the use of several design and simulation platforms. Computer power calculation has rapidly increased past ten years, and computer clusterization is now a widespread solution, so simulation time can be lowered. Meanwhile, this paper shows that only small matrixes can be simulated with a classical approach. Moreover, designers cannot do without post-layout simulations that are easily accessible in the classical micro-electronic design flow.

To answer the above mentioned problems, we propose in this paper a graphical toolbox that is dedicated to electronic designers, as it is integrated in Cadence Analog Design Environment (ADE). Novelty is to propose an easy way to manage many (millions) parameters at input and output, and to fasten simulation with a new parametric approach.

### III. INPUT AND OUTPUT IMAGE PROCESS

In order to handle all the input and output signals, a high-level mechanism has been set. It permits to read an image as input, and generates an image as output. As Fig. 4 shows, several steps are required.

We consider an image that has the same resolution as the image sensor. In that way, a pixel in the input image will be converted into luminosity or a photocurrent value. Then, that input value is set to the pixel input. A first skill processing function converts image values into lux, watts or amperes (according to the photodiode model), and then a second one assigns each value to each pixel. Design variables in ADE are used to make that mapping. Before assignment, an automatic creation of millions of design variables is done. Names are also fixed by software, for example lum\_0\_0 for the first pixel, lum\_3\_200 for 201th pixel of fourth line. Then, the simulation that was configured in ADE is run.

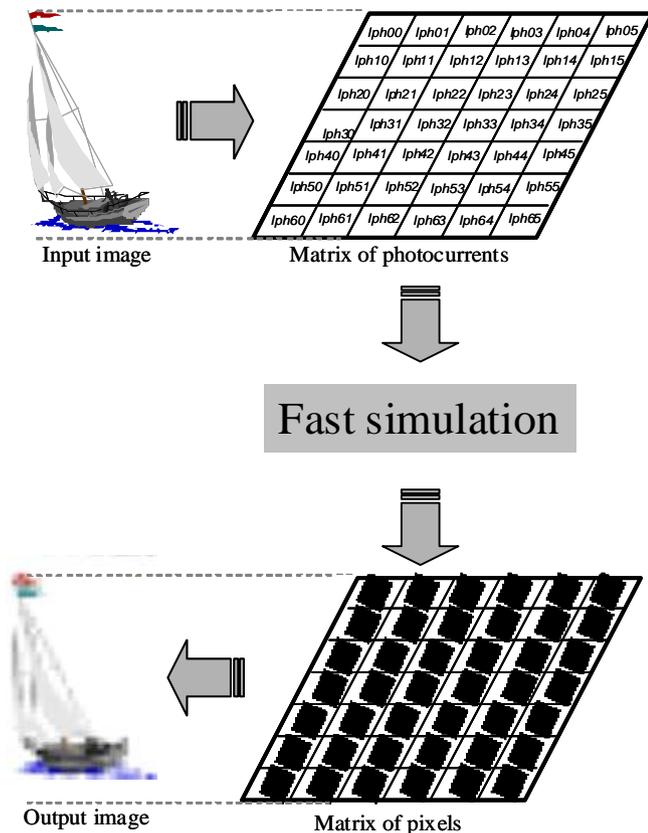


Figure 4 : Input and output image flow

As a result, pixel output voltage can be a raw voltage value or a simulated double sampled one, as illustrated in Fig. 6. In case of automatic double measurement, two simulations are run. The original netlist is modified in order to turn on the select transistor, so  $V_{rst}$  can be read at reset state. It is effectively sampled  $1\mu s$  before falling edge of reset signal. Considering classical timings in such circuits, pixel output voltage will have a stable and final value. This timing is a parameter that can be changed through a setup menu. Then, the user-defined simulation is run, and  $V_{os}$  signal is sampled at the middle time of select window.

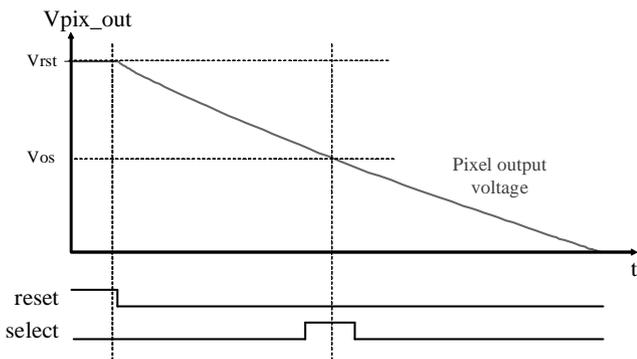


Figure 5 : Automatic timing to read voltages values at pixel output: on user request, simulator can output  $V_{os}$  or  $(V_{rst} - V_{os})$  based data

At end of simulation, Cadence Spectre output files are processed in order to display results. It is mandatory to detail a classical output simulation to best explain the internal result processing. The output voltage, computed with one of the two previously explained manners, is converted in grey level. Three solutions can also be configured: raw, absolute or relative coding.

- Raw coding is a basic reading of  $V_{os}$  output signal voltage and supply voltage  $V_{cc}$  is set as maximal digital code (255 in 8-bit or 1023 in 10-bit). Equation 1 gives a calculation example for 8-bit resolution. This solution can be used as a debug mode for designer, in order to check the raw simulation output compared to classical (manual) simulation.

$$\text{Raw grey-value} = 255 \times \left( 1 - \frac{V_{os}}{V_{cc}} \right) \quad (1)$$

- Absolute coding consists in setting the maximal pixel output ( $V_{rst}$ ) as maximal digital code. Saturation, ie 0v, is set as the minimal value. Equation 2 gives the equivalent calculation for 8-bit resolution. This calculation gives an equivalent result as a classical CDS block would do.

$$\text{Absolute coding grey-value} = 255 \times \left( 1 - \frac{V_{os}}{V_{rst}} \right) \quad (2)$$

- Relative coding consists in finding maximal and minimal pixel output voltages (respectively  $V_{max}$  and  $V_{min}$ ) in the matrix, and to set them to maximal and minimal codes. Equation 3 gives the equivalent calculation for 8-bit resolution. It is what an image signal processor would do to optimize dynamic range.

$$\text{Relative coding grey-value} = 255 \times \left( 1 - \frac{V_{os}}{V_{max} - V_{min}} \right) \quad (3)$$

These three solutions permit to optimize hardware and software in image sensor: hardware blocks such as CDS and ADC, and software for image signal processor at output. Hardware block that can be hierarchically optimized are at pixel, matrix, matrix and CDS, and matrix and CDS and ADC levels.

#### IV. FAST SIMULATION METHOD AND USER INTERFACE

Classical simulations are long, and permit only small matrixes analysis. It is to notice that very complex equations will be simulated to calculate voltages that will enter an analog to digital converter. Output will also have a discrete step, as output is composed of 256 grey levels. Instead of downsizing result at output (infinite values of voltages into 256 finite grey level values), we discretize input.

We also run a set of 256 equations that composes all the possible inputs on a single pixel, whose line and column capacitances have been considered to render the matrix real values. All the possible outputs are also known. This permits to create a look up table, as shown in Fig. 6.

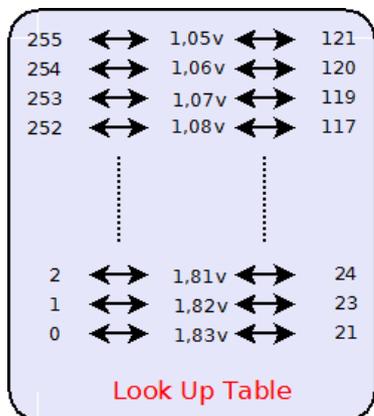


Figure 6 : Look up table that associates input grey levels to output pixel voltage and then output grey levels

After this 256 simulations step (for look up table establishment), each of the MxN inputs are associated to an output during a mapping step.

The graphical user interface is showed in Fig. 7. It is opened via an "imager" menu in ADE.

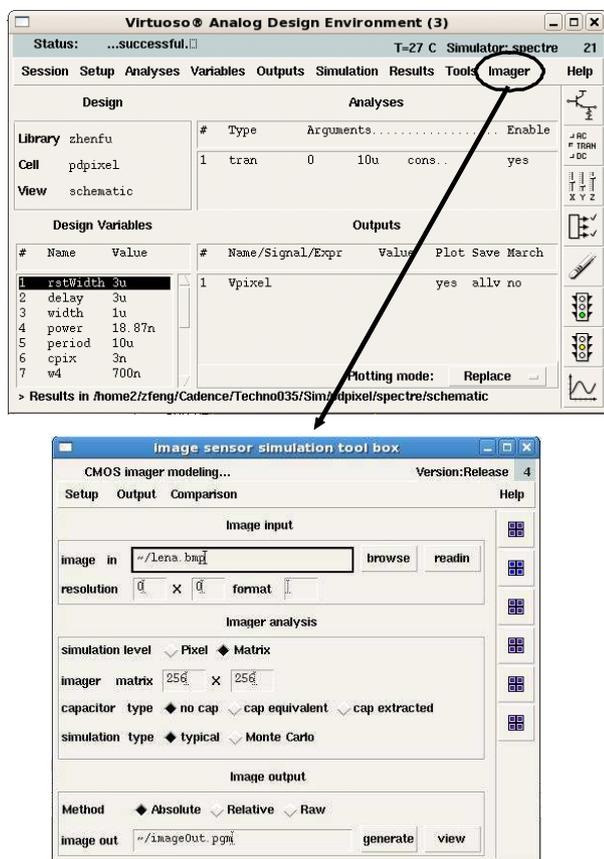


Figure 7 : Fast simulator user interface

User can select the input image (in uncompressed BMP or PGM formats). Clicking the read-in button imports image and converts color or grey pixels values into light information (light power or photocurrent according to the photodiode model that is used). This conversion is done by considering silicon sensitivity to light.

As analysis setup is launched from a single pixel schematic, it is possible to select a single pixel simulation (sim-Pixel), or a matrix simulation (sim-Matrix). For a single pixel simulation, coordinates of pixel within the matrix (pixel at location 100, 100 in Fig. 7) are selected. For a m x n matrix simulation, the same pixel is simulated m x n times with respective m x n stimuli. As a matter of fact, output node of the pixel (vout) as to be defined. Vout is set as "pixel" value in Fig. 7. It is to notice that many pixel structures can be simulated, since simulation and toolbox can run on any schematic. Moreover, this toolbox could be used for any matrix simulation, like memories.

## V. RESULTS

As test example, we have considered an input image, a classical 3T pixel, and several configurations. As we can observe, from an input image, Fig. 8, it is possible to automatically set the matrix data (light) input that is applied on electronic structure, to simulate electronic circuit with Cadence Spectre, and then to monitor output images according to the chosen reading mode configuration (Fig. 9 to Fig. 11). Raw or absolute coding will be used to characterize low level characteristics of pixels matrix, or to study a system composed of matrix and Correlated Double Sampling block, eventually with Analog to Digital Converter (ADC).

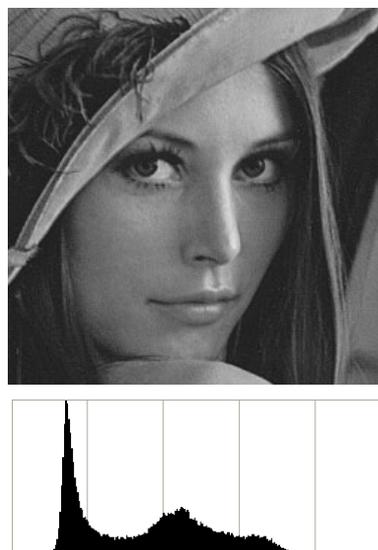


Figure 8 : Input image and histogram

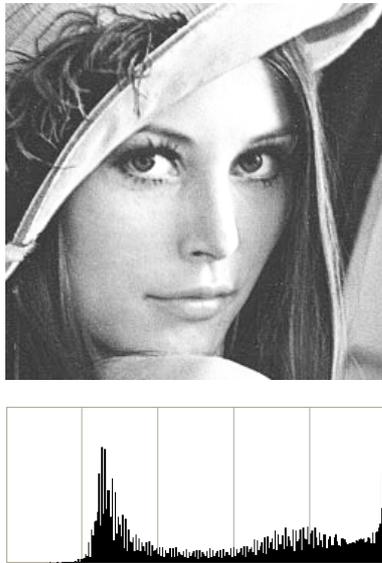


Figure 9 : Absolute coding output image and histogram,  $W_{\text{follower}} = 1 \mu\text{m}$

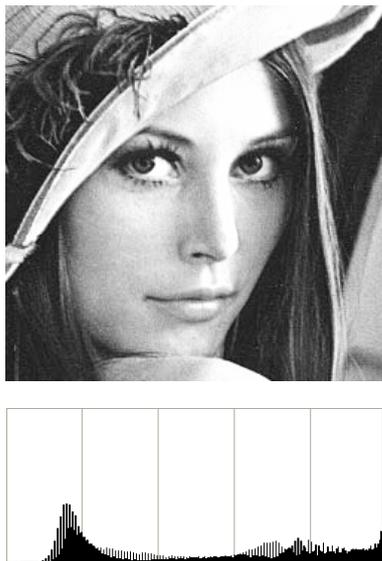


Figure 10 : Relative coding output image and histogram,  $W_{\text{follower}} = 1 \mu\text{m}$

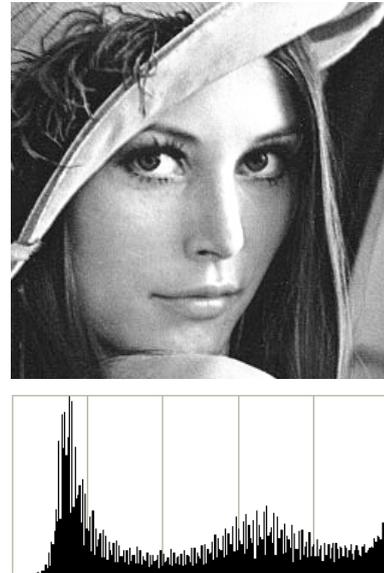


Figure 11 : Relative coding output image and histogram,  $W_{\text{follower}} = 15 \mu\text{m}$

As double sampling is done in CDS block, it is indeed useless to simulate an image sensor that comprises such a block with the relative coding option. Relative coding is useful to characterize a standalone pixels matrix, or specific smart image sensors [8] that don't embed CDS. Following blocks parameters, such as amplifiers, CDS and ADC, can also be studied and dimensioned. We can observe that contrast and mean values differ from input to output images, and it is more visible on histograms. Histograms, that present grey values versus number of pixels, clearly show that transfer function of sensor is not linear. Moreover, Fig. 11 shows the transistor size impact on output image quality. In this testcase, a bigger  $W$  size gives a closer histogram so a tradeoff sensor size / quality can be led. Histogram values of Fig. 11 are shifted to the right compared to Fig. 10, this sensor suffers also a nonlinearity. Moreover, it has saturation (too light values at right of histogram). Electronic and architecture impacts on image sensor quality can also be clearly studied.

Simulation time is detailed in Table 1, gain is significant. Classical simulation is not possible on an Intel Xeon processor 2.6GHz with 4GB of RAM for a 512x512 matrix, because of memory overflow. It is to note that fast simulation time is almost constant, because look up table establishment (i.e 256 simulations) is important and look up table reading (mapping) is negligible.

Matrix size	8x8	12x12	60x60	256X256	512X512
Classical simulation (min)	3.7	7.5	179.23	3465	error
Fast simulation (min)	11.57	12.01	12.21	12.3	12.31

Table 1 : Simulation time for classical simulation and fast simulation

In terms of accuracy, table 2 gives difference between a 20x20 real matrix simulation and the one-pixel-based fast simulation.

	Pixel voltage output	error
Classical simulation	779,92527 mV	0,00043 %
Fast simulation	779,92185 mV	

Table 2 : Accuracy of fast simulation method

## VI. CONCLUSION

This paper presents a new toolbox for CMOS image sensor simulations in Cadence Analog Design Environment. It is written in SKILL language, and it permits to input automatically an input image, fitting each image pixel on each pixel of the electronic sensor under study. Light stimuli is calculated and applied on each. Output images permit to check the quality of analog design in the pixel. Low level aspects of the sensor can be analyzed at system (image) level. This toolbox is composed of a new fast simulator that permits several mega-pixels simulation within a few minutes. Its principle is to study a single pixel in a parametrical way, and to project result on the matrix.

## REFERENCES RÉFÉRENCES REFERENCIAS

1. Knickerbocker, J.U. et al. "3-D Silicon Integration and Silicon Packaging Technology Using Silicon Through-Vias", IEEE Journal of Solid-State Circuits, vol. 41, no. 8, pp.1718-1725, August 2006.
2. Topol et. al., "Three-dimensional integrated circuits", IBM Journal Research and Development,. Vol. 50 no. 4/5, pp. 491-506, July/September 2006.
3. E.R. Fossum, "CMOS Image Sensors: Electronic Camera-On-A-Chip", IEEE Trans. on Electronic Devices, Vol 44, N° 10, October 1997.
4. D. Navarro, D. Ramat, F. Mieyeville, I. O'Connor, F. Gaffiot, L. Carrel, "VHDL & VHDL-AMS modeling and simulation of a CMOS imager IP", Forum on specification & Design Languages, Lausanne, Switzerland, September 2005.
5. Silvaco, "CMOS Image Sensor Simulation - ATLAS", [www.silvaco.com/content/kbase/CIS\\_april2010.pdf](http://www.silvaco.com/content/kbase/CIS_april2010.pdf)
6. CrossLight Inc, "3D Simulation of CMOS Image Sensor", [www.crosslight.com/applications/crosslight\\_3DCIS3.pdf](http://www.crosslight.com/applications/crosslight_3DCIS3.pdf)

7. Matworks, "MATLAB Image Processing Toolbox", <http://www.mathworks.com/products/image>
8. J. Kramer, R. Sarpeshkar, C. Koch, "Pulse-Based Analog Velocity Sensors", IEEE Trans. On Circuits and Systems II : Analog and Digital Signal Processing, Vol 44, pp 86-101, 1997.