# A QVGA Vision Sensor with Multi-functional Pixels for Focal-Plane Programmable Obfuscation

J. Fernández-Berni, R. Carmona Galán, R. del Río, Á. Rodríguez-Vázquez
Institute of Microelectronics of Seville (IMSE-CNM), CSIC - Universidad de Sevilla
Avda. Américo Vespucio s/n, 41092, Seville, Spain
Contact email: berni@imse-cnm.csic.es

## ABSTRACT

Privacy awareness constitutes a critical aspect for smart camera networks. An ideal flawless protection of sensitive information would boost their application scenarios. However, it is still far from being achieved. Numerous challenges arise at different levels, from hardware security to subjective perception. Generally speaking, it can be stated that the closer to the image sensing device the protection measures take place, the higher the privacy and security attainable. Likewise, the integration of heterogeneous camera components becomes simpler since most of them will not require to consider privacy issues. The ultimate objective would be to incorporate complete protection directly into a smart image sensor in such a way that no sensitive data would be delivered off-chip while still permitting the targeted video analytics. This paper presents a 320×240-px prototype vision sensor embedding processing capabilities useful for accomplishing this objective. It is based on reconfigurable focal-plane sensing-processing that can provide programmable obfuscation. Pixelation of tunable granularity can be applied to multiple image regions in parallel. In addition to this functionality, the sensor exploits reconfigurability to implement other processing primitives, namely block-wise high dynamic range, integral image computation and Gaussian filtering. Its power consumption ranges from 42.6mW for high dynamic range operation to 55.2mW for integral image computation at 30fps. It has been fabricated in a standard 0.18$\mu$m CMOS process.

## Categories and Subject Descriptors

Hardware [**Very large scale integration design**]: Full-custom circuits

## General Terms

Algorithms, Security


*ICDSC'14,* November 04 - 07 2014, Venezia Mestre, Italy

http://dx.doi.org/10.1145/2659021.2659045

## Keywords

Smart Cameras, Privacy, Security, Vision Sensor, Focal-plane Processing, Obfuscation, Pixelation

## 1. INTRODUCTION

Camera networks have been around for some decades now in security and surveillance [1]. A classical picture is the deployment of several installed cameras, often pan-tilt-zoom with embedded video compression and data forwarding to a central spot. Recently many of these cameras are becoming "smart", i.e. video analytics will detect certain events such as the passing of a pedestrian to prompt a warden in order not to have to watch the scene permanently [8, 12]. However, despite this embedded smartness, it is still incredibly difficult for camera networks to enter application scenarios beyond safety and prompt action. Privacy is the primary drawback. People in general do not want their presence and actions to be policed permanently. The issue of privacy is hindering the introduction of smart cameras into retailing analytics, home security or elderly care.

Indeed, most current smart cameras are endowed with enough computational power for the implementation of privacy protection measures [2, 14, 15]. The real limitations come from the significant number of key system components that are part of the implicitly trusted software base: the operating system, the network stack, system libraries etc. It is not possible to provide complete assurance about the potential security and privacy flaws contained in this software [16]. Even widely adopted cryptography libraries are not free of such flaws [13]. A possible hardware-based approach to overcome these limitations is to convey protection as close to the sensor device as possible. The ideal framework would be a front-end vision sensor delivering a data flow stripped off personal/identifiable data. On-chip implementation of privacy awareness would still have to accommodate some degree of reconfigurability in order to balance protection and viability of the video analytics required by particular algorithms. Once protection measures are embedded on-chip at the front-end sensor of each network node, the number of trusted components as well as the impact of potential software flaws are significantly reduced.

Different techniques for privacy protection have been reported in the literature. The most basic form is blanking [3] where sensitive regions are completely removed from the captured images. No behavioral analysis is possible in this case, only the presence and location of a person can be

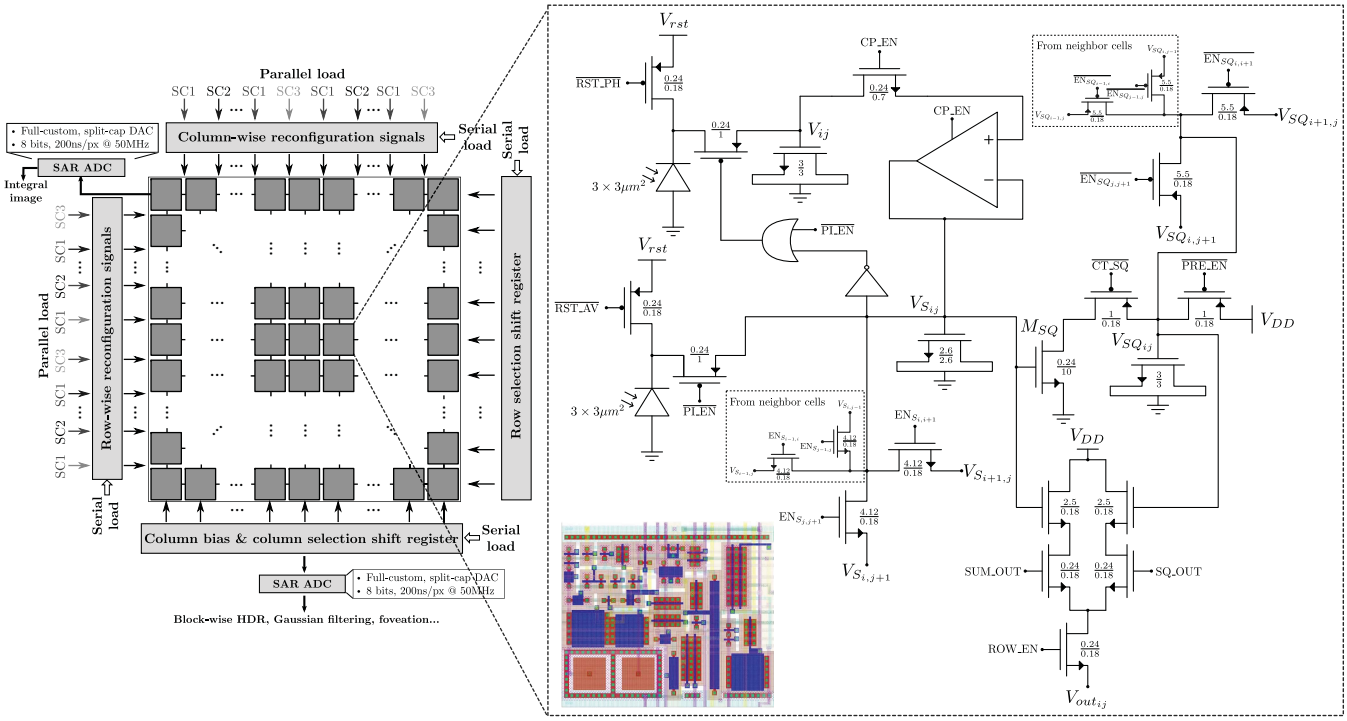


**Figure 1: Chip architecture and schematic and layout of the multi-functional mixed-signal pixel.**

monitored. Other alternatives that do enable such analysis are obfuscation and scrambling [4]. Concerning obfuscation, pixelation of sensitive regions provides the best performance in terms of balance between privacy protection and intelligibility of the surveyed scene when compared to blurring and masking filters [9, 10]. New techniques for obfuscation like warping [11] or cartooning [5] have been recently proposed.

In this paper, we present a full-custom QVGA vision sensor that can be reconfigured to implement programmable pixelation at the focal plane. It is based on focal-plane sensing-processing [18]. An array of 4-connected multi-functional pixels constitutes its operative core. The interaction between these pixels can be arranged block-wise by peripheral circuitry. Different image regions can thus be independently processed, enabling pixelation for multiple regions in parallel. The pixels of the array also include circuitry that exploits reconfigurability to provide additional low-level image processing primitives.

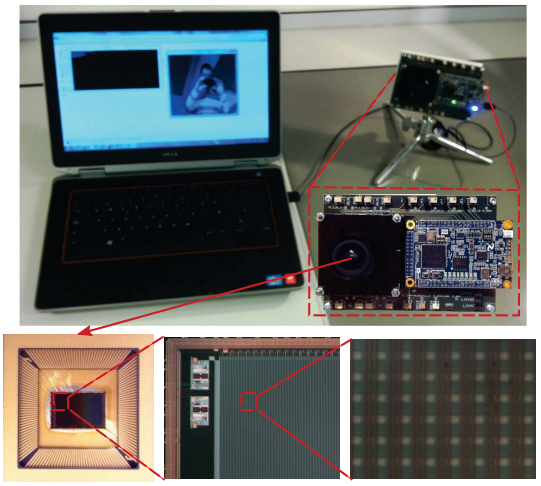

**Figure 2: Snapshot of the test board while capturing images, chip photo and two chip microphotographs.**

## 2. CHIP ARCHITECTURE

The proposed vision sensor is based on the concept of focal-plane sensing-processing, arguably the best architectural approach reported in terms of adaptation to the particular characteristics of early vision. On the one hand, the information to be handled at this processing stage —each and every pixel resulting from the raw readings of the sensors— is massive. On the other hand, the computational flow is very uniform. The same calculations are repeatedly carried out on every pixel. More interestingly, the outcome for each individual pixel does not usually depend on the outcome for the rest. Consequently, while an enormous amount of data must certainly be processed, regular massively parallel operation can still be applied. Focal-plane sensor-processor chips make the most of these characteristics by operating in Single Instruction Multiple Data (SIMD) mode featuring concurrent processing and distributed memory. Focal-plane architectures can also benefit from incorporating analog circuitry just at the point where the analog data feeding the processing chain are sensed. These analog circuits can reach higher performance in terms of speed, area and power consumption than digital circuitry while exploiting the moderate accuracy requirements of early vision tasks [17].

The chip presents the floorplan depicted in Fig. 1. The array of multi-functional pixels can be reconfigured block-wise by peripheral circuitry. The reconfiguration patterns are loaded serially into two shift registers that determine respectively which neighbor columns and rows can interact and which ones stay disconnected. There is also the possibility of loading in parallel up to six different patterns representing six successive image pixelation scales. This is achieved by means of control signals distributed regularly along the horizontal and vertical dimensions of the array. The reconfiguration signals coming from the periphery map into the signals $\mathrm{EN}_{S_{i,i+1}}$, $\mathrm{EN}_{S_{j,j+1}}$, $\overline{\mathrm{EN}_{SQ_{i,i+1}}}$ and $\overline{\mathrm{EN}_{SQ_{j,j+1}}}$ at pixel level. These signals control the activation of MOS switches for charge redistribution between the nMOS capacitors holding the voltages $V_{S_{ij}}$ and $V_{SQ_{ij}}$, respectively. Charge redistribution is the primary processing task that supports all the functionalities of the array, enabling a low-power operation. Concerning A-to-D conversion, there are four 8-bit SAR ADCs. These converters, based on a split-cap DAC, feature tunable conversion range, including rail-to-rail, and a conversion time of 200ns when clocked at 50MHz. Two of them provide integral imaging. The other two convert the pixel voltage $V_{out_{ij}}$ corresponding to the selected output of the source followers associated with $V_{S_{ij}}$ and $V_{SQ_{ij}}$. The column and row selection circuitry is also implemented by peripheral shift registers where a single logic '1' is shifted according to the location of the pixel to be converted.

The vision sensor has been embedded into a test system based on the commercial FPGA-based *DE0-Nano* board from terasIC. The resulting system can be seen in Fig. 2 together with some microphotographs of the chip. The output data flow provided by the sensor is stored in the internal memory of the FPGA for its subsequent serial transmission to a PC through a USB interface. The data rearrangement and image visualization in the PC are implemented by making use of OpenCV functionalities.

## 3. FOCAL-PLANE OBFUSCATION

The on-chip programmable pixelation is achieved by combining focal-plane reconfigurability, charge redistribution and distributed memory. After photointegration, the corresponding pixel values are represented by the voltages $V_{ij}$ distributed across the array. These pixel values can be copied in parallel into the voltages $V_{S_{ij}}$ by enabling the analog buffer included at each elementary cell. This copy process takes about 150ns for the whole array. It is not destructive with respect to the original voltages $V_{ij}$, what is crucial to accomplish focal-plane obfuscation without artifacts, as explained shortly. The next step, once the voltages $V_{S_{ij}}$ are set, consists in establishing the adequate interconnection patterns according to the image regions to be pixelated and the required degree of obfuscation. These patterns, when activated by the corresponding control signal, will enable charge redistribution among the connected capacitors holding $V_{S_{ij}}$, that is, the image copy. A simplified scheme of how the charge redistribution can be reconfigured column-wise and row-wise from the periphery of the proposed focal-plane array is depicted in Fig. 3. Every pair of neighbor columns and rows shares a common interconnection signal that is independent from the interconnection signals for any other pair. When the interconnections of consecutive columns and rows are enabled, averaging—that is, charge redistribution—will

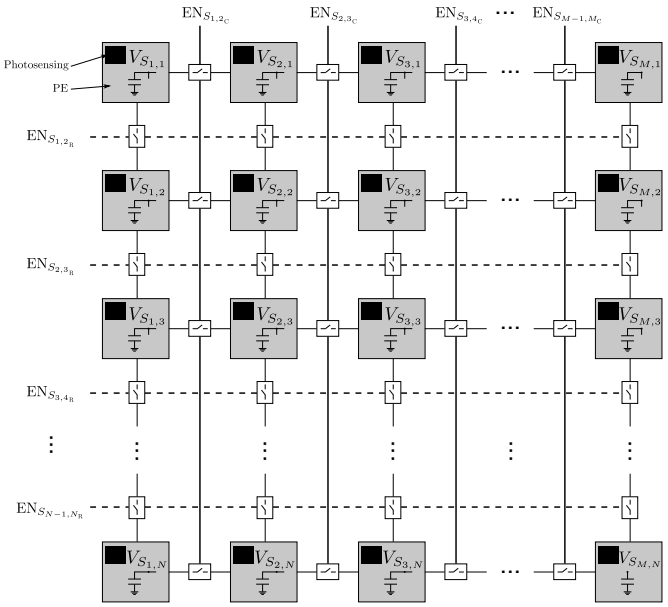


**Figure 3: Simplified scheme of how the charge redistribution can be reconfigured column-wise and row-wise from the periphery of the proposed focal-plane array.**

take place within the resulting block. Otherwise, the pixel values keep unchanged.

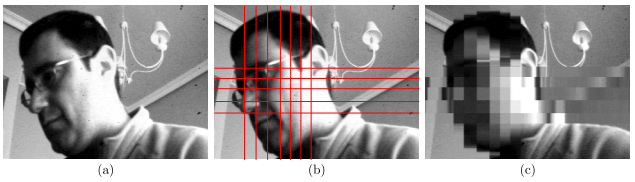


**Figure 4: (a) Original image captured by the chip. (b) Patterns for pixel interconnection. (c) Undesired pixelation artifacts out of the region of interest.**

An example of the focal-plane obfuscation attainable by applying this operation is shown in Fig. 4. The first snapshot represents an image captured by the chip. This image, as well as the rest of images included in this paper, has not undergone any off-chip post-processing at all. The interconnection patterns established for the pixelation of the face in the scene are highlighted in the second picture. Finally, the third image depicts the resulting focal-plane representation. It can be seen that the existence of a single control signal for the interconnection of all the cells along particular neighbor columns and rows generates spurious blocks within which averaging also occurs. The consequent artifacts significantly reduce the amount of useful information contained in the image, distorting its content. However, we can overcome this problem by exploiting the distributed memory inherent to the sensing-processing array. Bear in mind that the image in Fig. 4(c) is the outcome after photointegration, pixel copy, interconnection setting, charge re-

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

**Figure 5: On-chip obfuscation without artifacts: (a) Original image captured by the chip. (b) Pixelation with $4 \times 4$-px blocks. (c) Pixelation with $8 \times 8$-px blocks. (d) Pixelation with $16 \times 16$-px blocks.**

distribution and A-to-D conversion. And this last stage is key to remove the aforementioned artifacts. During A-to-D conversion, we simply need to keep track of those pixels located out of the region of interest and featuring any kind of connection with their neighborhood. For them, we activate the copy of their corresponding original pixel value still stored in the capacitors holding $V_{ij}$ before starting conversion. Otherwise, averaging is allowed. On-chip obfuscation without artifacts can thus be achieved, as shown in Fig. 5. In this case, we set interconnection patterns for progressively coarser parallel pixelation of two different image regions containing faces. The A-to-D conversion stage is adjusted in such a way that the original value of the pixels out of the obfuscated regions is always delivered thanks to the built-in distributed memory.

For this prototype, all the reconfiguration and control of the array must be carried out externally. The FPGA of the *DE0-Nano* board plays this role in the test system. No smartness concerning which particular regions must be obfuscated is embedded into the chip. Our objective is to incorporate such smartness on-chip in the near future. The resulting vision sensor would constitute a key component for the issue of privacy in smart camera networks. The integration of complete protection just at the front-end sensor of each node would imply a solution that cannot be tapped by design, preventing privacy sensitive data from being misused even by legitimate users. In any case, a live demonstration will also be presented. In this demo, the sensor captures images that are sent to a PC from the test board. The Viola-Jones frontal face detector provided by OpenCV is run on these images on the PC. If faces are detected, the coordinates of the corresponding bounding rectangle are sent back to the test board for the vision sensor to reconfigure the image capture in real time. Pixelation of the face regions will take place from that moment on at the focal plane. The degree of pixelation of these regions is adjustable through a button of the test board.

# 4. ADDITIONAL FOCAL-PLANE PROCESSING PRIMITIVES

The exploitation of focal-plane reconfigurability, charge redistribution and distributed memory also enables the realization of additional early vision tasks.

## 4.1 Block-wise HDR

Two photodiodes and two sensing capacitances per pixel are required to implement this low-level operation. Once they have been reset to $V_{rst}$, photointegration starts concurrently in both the pixel capacitance—holding $V_{ij}$—and the averaging capacitance—holding $V_{S_{ij}}$. However, while in the former it is carried out in an isolated way, charge redistribution takes place in parallel in the latter among the averaging capacitances interconnected through the switches controlled from the periphery by $EN_{S_{i,i+1}}$ and $EN_{S_{j,j+1}}$. The pixel photointegration is thus stopped at a certain time instant depending on the input threshold voltage of the inverter connected to $V_{S_{ij}}$. If this threshold voltage is designed to be at the middle point of the signal range, it can be demonstrated [7] that the voltage excursion due to photointegration for each pixel within a certain prescribed block $k$ is given by:

$$\Delta V_{ij_k} = \frac{\Delta V_{ij_{MAX}}}{2} \frac{I_{ph_{ij_k}}}{\bar{I}_{ph_k}} \tag{1}$$

where $\Delta V_{ij_{MAX}} = V_{rst} - V_{min}$ represents the maximum pixel excursion, $I_{ph_{ij}}$ denotes the pixel photogenerated current and $\bar{I}_{ph_k}$ is the block average photocurrent generated during the photointegration period. We can see from Eq. 1 that the maximum pixel illumination to be detected without saturation is double of the average illumination of the block. It is this property, together with the possibility of confining its application to any particular rectangular-shaped image region, what endows our array with the capability of retrieving information, otherwise missed, from HDR scenes.

An example of this primitive is shown in Fig. 6. Global integration time control is applied to the left image. All pixels undergo the same integration time, which is set to 500ms according to the mean illumination of the scene. Details about the lamp are missed due to the extreme deviation with respect to this mean illumination. However, such details can be retrieved by confining the control of the integration period to the region of interest, as can be seen in the right image. In this case, the integration time of the region around the center of the lamp adjusts locally and asynchronously to its mean illumination, stopping the photointegration at around $400\mu$s in that particular area while it continues at the remaining regions. Dynamic ranges up to 102dB have been achieved through this technique.

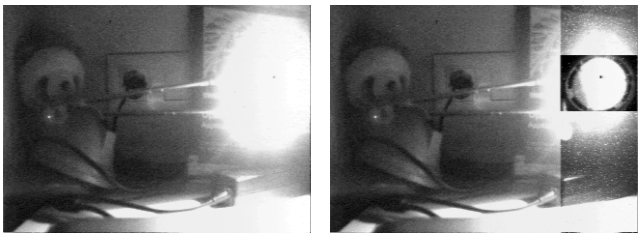


**Figure 6: On-chip block-wise intra-frame integration time control in order to deal with scenes demanding high dynamic ranges.**

## 4.2 Integral image

The so-called integral image is a common intermediate image representation used by well-known vision algorithms, e.g. the Viola-Jones framework for object detection. It is defined as:

$$II(x,y) = \sum_{x'=1}^{x} \sum_{y'=1}^{y} I(x',y') \tag{2}$$

where $I(x,y)$ represents the input image. That is, each pixel composing $II(x,y)$ is given by the sum of all the pixels above and to the left of the corresponding pixel at the input image. In order to deal with the extremely wide signal range required to represent an integral image, charge redistribution plays a key role as the underlying physical operation supporting the computation at the focal plane. Charge redistribution permits to keep the signal swing within the range of individual pixels, no matter how many pixels of the original image are involved in the computation of the current integral image's pixel. The average value obtained for each case simply requires keeping externally track of the position of the pixel being calculated. Thus, we only need to multiply that average value by the number of row and column associated with the corresponding pixel. In other words, the array is capable of computing an averaged version of the integral image mathematically described as:

$$II_{av}(x,y) = \frac{1}{x \cdot y} \sum_{x'=1}^{x} \sum_{y'=1}^{y} I(x',y') \tag{3}$$

This averaged integral image delivered by the chip can be visualized in Fig. 7 along with the integral image that can be directly derived from it. This integral image is compared to the ideal integral image obtained off-chip from the original image captured by the sensor, attaining a RMSE of 1.62%.

The array can also compute an averaged version of the square integral image by precharging the capacitor holding $V_{SQ_{ij}}$ to $V_{DD}$ and exploiting its discharge for a short period of time through the transistor $M_{SQ}$ working in the saturation region. Then, charge redistribution would take place, just as for the integral image. In order to read out and convert every pixel of these integral images, we must simply connect $V_{S_{1,1}}$ and $V_{SQ_{1,1}}$ to respective analog-to-digital converters.

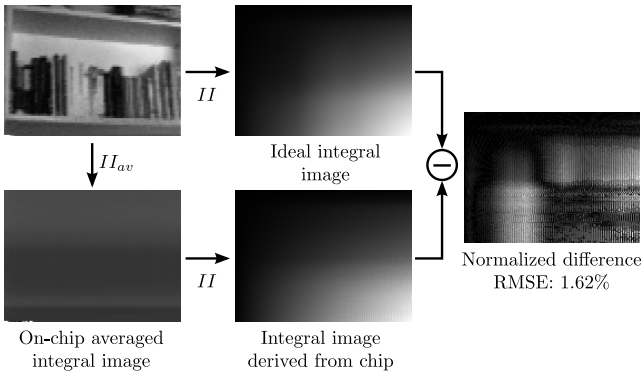


**Figure 7: On-chip integral image computation.**

These voltages will always contain the targeted calculation for each pixel, according to the definition of integral image and the proposed hardware implementation based on charge redistribution.

## 4.3 Gaussian filtering

The combination of charge redistribution and focal-plane reconfigurability enables subsequent reduced kernel filtering by adjusting which pixels merge their values and in which order [6]. Progressive Gaussian filtering can be completely implemented at the focal plane by successively applying the binomial filter mask:

$$\mathbf{G}_b = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

An example of this processing primitive is depicted in Fig. 8 along with the corresponding error measurements.

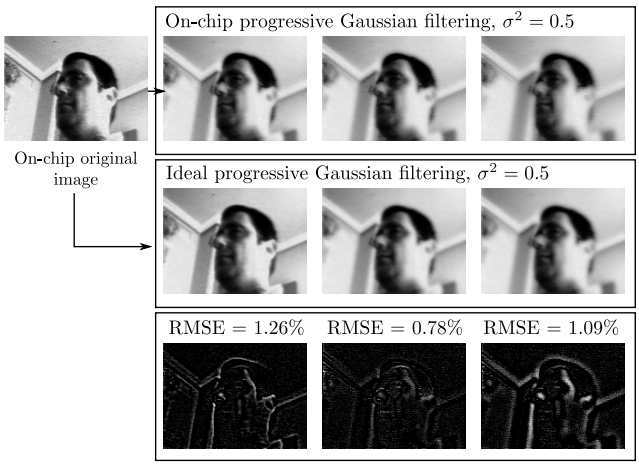


**Figure 8: On-chip Gaussian filtering.**

## 5. CONCLUSIONS

Privacy awareness plays a crucial role when it comes to exploring new application frameworks for smart camera networks. Protection measures implemented on specific hardware close to the imaging device result of great interest in terms of system security. Hackers would not be able to tamper any more with network nodes through potential software flaws. This paper reports a full-custom QVGA vision sensor taking this hardware-based approach for privacy a step further. Different low-level processing primitives are embedded on-chip at the focal plane in addition to raw image capture. Among them, programmable pixelation enables obfuscation of image regions in parallel. The granularity of this operation can be tuned in order to balance privacy and utility of the subsequent video analytics according to the requirements of particular vision algorithms. The ultimate target is to integrate complete protection in a smart image sensor that never delivers sensitive data off-chip.

## 6. ACKNOWLEDGMENTS

This work has been funded by the Spanish Government through projects TEC2012-38921-C02 MINECO (European Region Development Fund, ERDF/FEDER), IPT-2011-1625-430000 MINECO and IPC-20111009 CDTI (ERDF/FEDER), by Junta de Andalucía through project TIC 2338-2013 CEICE, by the Office of Naval Research (USA) through grant N000141410355 and by the Faculty of Engineering of Ghent University through its program for visiting foreign researchers.

## 7. REFERENCES

[1] H. Aghajan and A. Cavallaro, editors. *Multi-Camera Networks: Principles and Applications*. Academic Press, 2009.

[2] A. Chattopadhyay and T. Boult. PrivacyCam: a privacy preserving camera using uCLinux on the Blackfin DSP. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[3] S. Cheung, J. Zhao, and M. Vijay. Efficient object-based video inpainting. In *IEEE Int. Conf. on Image Processing*, pages 705–708, 2006.

[4] D. Dufaux and T. Ebrahimi. Scrambling for privacy protection in video surveillance systems. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(8):1168–1174, 2008.

[5] A. Erdélyi, T. Winkler, and B. Rinner. Serious fun: Cartooning for privacy protection. In *Int. Benchmarking Initiative for Multimedia Evaluation (MediaEval)*, 2013.

[6] J. Fernández-Berni, R. Carmona-Galán, and A. Rodríguez-Vázquez. Image filtering by reduced kernels exploiting kernel structure and focal-plane averaging. In *European Conf. on Circuit Theory and Design (ECCTD)*, pages 229–232, 2011.

[7] J. Fernández-Berni, R. Carmona-Galán, and A. Rodríguez-Vázquez. Reconfigurable focal-plane hardware for block-wise intra-frame HDR imaging. In *Int. Image Sensor Workshop*, pages 289–292, 2013.

[8] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan. Mesheye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Int. Symp. on Information Processing in Sensor Networks, (IPSN)*, pages 360–369, 2007.

[9] P. Korshunov, C. Araimo, F. De Simone, C. Velardo, J. Dugelay, and T. Ebrahimi. Subjective study of privacy filters in video surveillance. In *IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, pages 378–382, 2012.

[10] P. Korshunov, S. Cai, and T. Ebrahimi. Crowdsourcing approach for evaluation of privacy filters in video surveillance. In *ACM Int. Workshop on Crowdsourcing for Multimedia*, pages 35–40, 2012.

[11] P. Korshunov and T. Ebrahimi. Using warping for privacy protection in video surveillance. In *IEEE Int. Conf. on Digital Signal Processing (DSP)*, 2013.

[12] C. Micheloni and G. Foresti. Active tuning of intrinsic camera parameters. *IEEE Trans. on Automation Science and Engineering*, 6(4):577–587, 2009.

[13] OpenSSL Project. Heartbeat overflow issue. `https://www.openssl.org/news/secadv_20140407.txt`, 2014. [Online; accessed 15-April-2014].

[14] D. Serpanos and A. Papalambrou. Security and privacy in distributed smart cameras. *Proceedings of the IEEE*, 96(10):1678–1687, 2008.

[15] T. Winkler and B. Rinner. TrustCAM: Security and privacy-protection for an embedded smart camera based on trusted computing. In *IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 593–600, 2010.

[16] T. Winkler and B. Rinner. Sensor-level security and privacy protection by embedding video content analysis. In *IEEE Int. Conf. on Digital Signal Processing (DSP)*, 2013.

[17] J. Wyatt, C. Keast, M. Seidel, D. Standley, B. Horn, T. Knight, C. Sodini, H. seung Lee, and T. Poggio. Analog VLSI systems for image acquisition and fast early vision. *Int. J. of Computer Vision*, 8(3):217–230, 1992.

[18] A. Zarándy, editor. *Focal-plane Sensor-Processor Chips*. Springer, 2011.