Parallel HDR Tone Mapping and Auto-focus on a Cellular Processor Array Vision Chip

Julien N. P. Martel *, Lorenz K. Müller *, Stephen J. Carey † and Piotr Dudek †
Email: jmartel@ini.ethz.ch lorenz@ini.ethz.ch stephen.carey@manchester.ac.uk p.dudek@manchester.ac.uk
* Institute of Neuroinformatics, University of Zurich / ETH Zurich, Zurich 8057, Switzerland
† School of Electrical & Electronic Engineering, The University of Manchester, Manchester M13 9PL, United-Kingdom

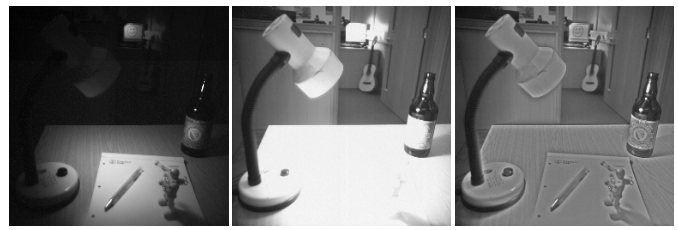
***Abstract*—To improve computational efficiency, it may be advantageous to transfer part of the intelligence lying in the core of a system to its sensors. Vision sensors equipped with small programmable processors at each pixel allow us to follow this principle in so-called near-focal plane processing, which is performed on-chip directly where light is being collected. Such devices need then only to communicate relevant pre-processed visual information to other parts of the system. In this work, we demonstrate how two classical problems, namely high dynamic range imaging and auto-focus, can be solved efficiently using two simple parallel algorithms implemented on such a chip. We illustrate with these two examples that embedding uncomplicated algorithms on-chip, directly where information acquisition takes place can replace more complex dedicated post-processing. Adapting data acquisition by bringing processing at the sensor level allows us to explore solutions that would not be feasible in a conventional sensor-ADC-processor pipeline.**

I. Introduction

In traditional information processing pipelines the sensor, where data is acquired, is generally separate from the processing core, where computation takes place. This leads to a scaling problem, for example in artificial vision: the increase in the resolution of the sensors, as well as the availability of larger communication bandwidth, result in a bottleneck in the core of the processing system. Because of massive amounts of data, it is difficult to design systems that can achieve the low latencies necessary to meet real-world constraints. To alleviate the computational burden that the core of the processing pipelines have to bear, a natural shift in strategy is to equip the sensors with more processing capabilities, and change the way they acquire information.

For a decade this trend to move processing to the periphery, closer to the sensors and to the data-acquisition has led to very successful systems at both the hardware level and software or algorithmic level. First by creating new sensors or modifying the way they acquire information [1]–[3], and second by providing new computational methods [4], [5]. A successful example is compressed sensing [5] in which sensor acquisition and information processing have been jointly addressed to deal with efficient signal compression and reconstruction. Also, by changing the way a system acquires information may help to perform further processing. In this line of thought a variety of vision sensors able to perform some processing directly at the pixel level have been designed and fabricated [6], [7]. These vision chips usually embed a pixel-wise circuitry that performs computation directly on the incoming light-intensity

This work was funded by SNF grant 143947.



(a) (b) (c)

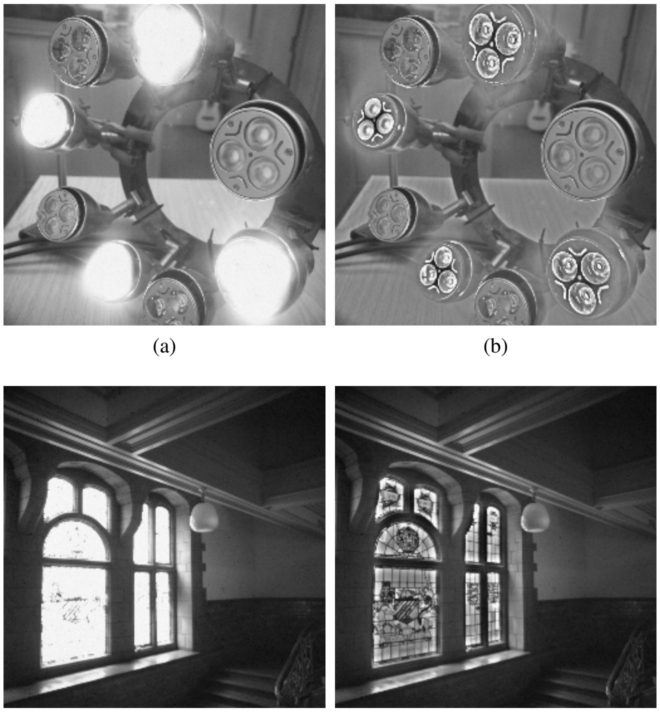
Fig. 1. A scene, exhibiting a wide radiosity range, imaged by a SCAMP-5 vision chip (a) with a low exposure (b) with a high exposure and (c) with our HDR tone mapping adaptive sensing algorithm.

thus implementing what has been coined as near focal-plane processing [8]. The idea of these devices is to benefit from massively parallel computation that can be performed at the pixel-level and thus reduce the amount of data transmitted to the core of a processing pipeline. Some of these devices, known as Cellular Processor Arrays (CPA), consist of an array of programmable processors, in which pixel processors can share information with their neighbours [9]–[14]. A variety of algorithms can be demonstrated on these devices.

Using programmable vision chips [13], [14], we design an algorithm that exploits the parallelism and the near-focal plane processing of these devices to produce on-chip a stream of High Dynamic Range (HDR) frames, used as input of a simple auto-focus algorithm also performed on the programmable vision chip. The device equipped with our algorithm would therefore output in-focus HDR frames. Producing HDR images is usually a post-processing procedure that requires a dedicated algorithm [15]–[17]. Our method demonstrates that it is indeed possible to benefit from the advantages of processing close to the data to guide acquisition and deploy a simple method to solve a non-trivial problem.

In Section II of this work we introduce our adaptive sensing HDR tone mapping algorithm and review its properties. The parallel auto-focus we designed is described in Section III. Because our method is based on contrast detection it benefits from the use of HDR images produced by the first algorithm in order to increase its robustness against contrast variations across the image. Finally we present in Section IV results of the implementation of these two algorithms on actual CPA vision-chips [13], [14].

978-1-4799-5341-7/16/$31.00 ©2016 IEEE



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

Fig. 2. Examples of two scenes imaged with and without our on-chip HDR algorithm: (a) (c) without HDR, (b) (d) with HDR tone mapping running on chip.

II. "HIGH DYNAMIC RANGE" TONE MAPPING

A. Related works in HDR sensing & algorithms

The concept of HDR imaging is to be able to image scenes containing both very bright and dark parts. Although various computational photography techniques have been proposed as post-processing [15]–[17], such as taking images with multiple exposures and combining them, hardware allows the production of HDR images on the device during sensing: altering the way light is sampled. For instance, in [18] a range of 120 dB encoded on 20 bits can be captured thanks to an event-based circuitry, resetting and signaling when saturation of the pixel occurs. The multiple exposure idea is directly translated in [19], [20] in which differently exposed frames are combined on-chip. In other works, an integration time is determined "independently" for each pixel; in [21] several iterations of a CNN are used to compute a map a-priori prescribing pixel-wise integration time, while in [22], dedicated pixel-circuits adjust the integration time in regions of the image. Apart from [18], the output of these algorithms is a tone mapped image: A standard range is used for the encoding (eg. 8 bits) but a non-linear space-variant function maps the photo-current to a pixel value. Thus, conventional formats and display can be used for sharing and visualization.

B. An adaptive-sensing algorithm for HDR tone mapping

Principle: In our case, we assume the output from the vision chip is limited to 256 levels encoded on 8-bits. The internal computations have similarly limited dynamic range. Therefore, in order to produce a tone mapped HDR frame, we need to directly compute the tone mapped values, rather than performing additional steps over a HDR frame, with higher per pixel resolution. Practically, an image region that receives a lot of light should stop integrating photo-current earlier than an image region that receives little light.

This can only be achieved by changing the acquisition of light on-chip since we want to specify the integration time pixel-wise depending on some criteria. In a conventional camera all pixels are exposed for the same amount of time, on our vision chip the challenge was to find a way to emulate pixel-wise independent integration time. However, unlike [21] the "computation" for determining integration time is not performed a-priori but while exposing.

We work on Single Instruction Multiple Data (SIMD) vision chips where the same instruction is dispatched to all the pixels' processors to act on their own data stored in local registers. We denote the light-sensitive register as R_{pix}, the value stored in this register gradually increases, at a rate proportional to the incident light intensity (i.e. it integrates the photocurrent). In addition, we assume we can "mask" a subset of the processors by specifying them not to be active –i.e not to perform the instructions that are dispatched by the central controller– when an activity flag is set. This mechanism is a way to perform a branching condition dependant upon some logical clause evaluated on the value of a register.

What we want is to let the R_{pix} register integrate for a certain time. This time is determined as the time it would take to make every single pixel of the vision-chip be at their brightest level. During this integration time we can periodically –say every δ_t– copy the value of this light-sensitive register to another free register when some condition is met and mask the processor when the condition is not met (no copy will occur). After a fixed time determined by a multiple of δ_t one can readout the tone mapped HDR frame in the register where the copies have been performed.

Require: R_1, R_2, R_{frame}: three register arrays
Require: N, θ_1, θ_2, S_1, S_2: five hyper-parameters
$R_{\text{pix}} \leftarrow 0$ ▷ Reset the pixels
do
  $R_1 \leftarrow R_{\text{pix}} * K(S_1)$ ▷ Avg. neighb. small
  $R_2 \leftarrow R_{\text{pix}} * K(S_2)$ ▷ Avg. neighb. large
  if $R_1 \leq \theta_1$ and $R_2 \leq \theta_2$ **then**
    $R_{\text{frame}} \leftarrow R_{\text{pix}}$
  end if
  $t \leftarrow t + \delta t$
while $t < N \cdot \delta t$
trigger frame readout of R_{frame}

Properties: The condition we chose to perform a copy from R_{pix} to R_{frame} consists of two parts:

First, the integration time should depend on the amount of light collected at the pixel itself –or actually in a small neighbourhood of size S_1 (=1 pix typically). If it is less than some threshold θ_1, the pixel may need more light and copies from R_{pix} may continue to be performed. If the condition fails, then no copy occurs and R_{frame} keeps its value. This single threshold must not determine alone whether the copy should be performed or not. A second condition should take into account a larger neighbourhood of the pixel. A convolution with a Gaussian kernel K parameterized with its standard deviation (its size) S is performed in R_2 to average the light intensity in its neighbourhood. Only if this value is also less than a

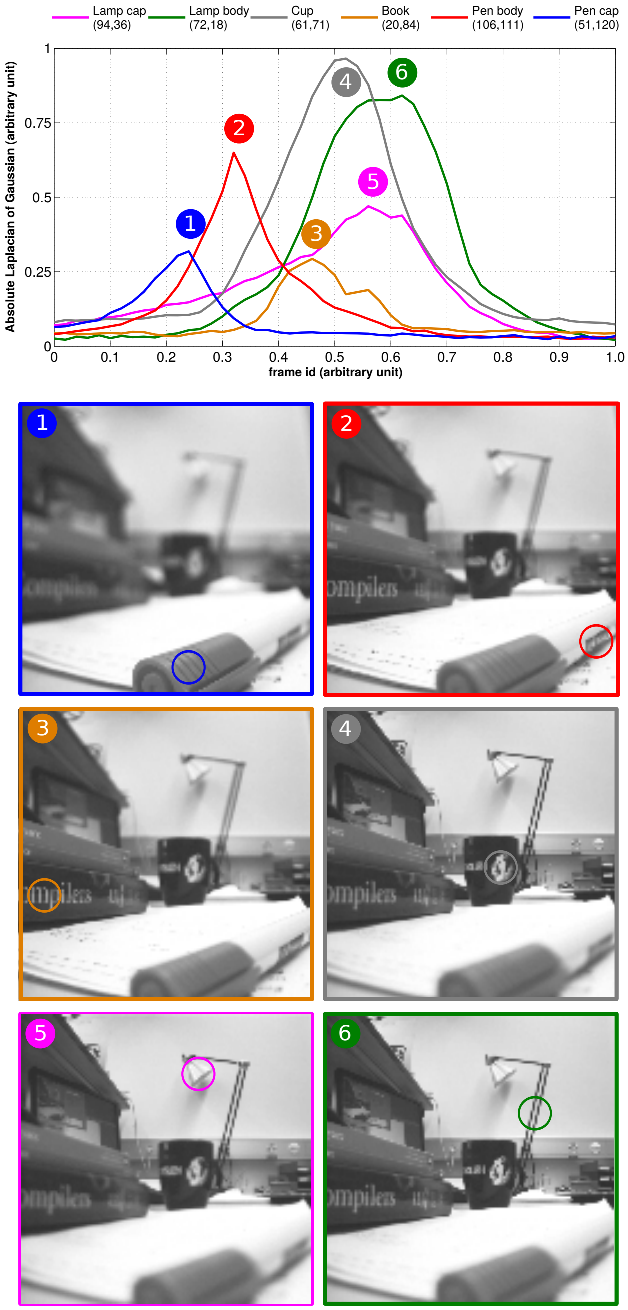


Fig. 3. Auto-focus implemented on SCAMP-3. The plot shows the value of the Laplacian of Gaussian for different image regions (pixel location is given in the legend) vs. the frame id, when sweeping the focus from near to far. Captured images are shown at times corresponding to the maximum value in each of the regions.

threshold θ_2 the pixel should continue to integrate. Note that the maximum integration time is bounded by $N \cdot \delta t$ and that δ_t is not a free parameter but is determined by the time it takes to perform the instructions 3 to 9.

In this algorithm the maximal pixel intensity will be near θ_1 since no further copy happens once the threshold is reached. A similar argument holds for the local average intensity that will be near θ_2. The neighbourhood size S allows to "interpolate" between two extreme images: if S is the image size then a single exposure for all the pixels is chosen and produces a non-hdr image, if $S = 0$ then all the pixels will be near θ_1.

III. A Parallel auto-focus algorithm

A. Auto-focus Systems

The aim of an auto-focus (AF) system is to adjust the distance between the focal-plane of the optic system and the camera sensor so as to bring parts or the entirety of the imaged scene into focus. Two major categories of AF systems exist: active and passive. Active AF are named for the defining characteristic that they emit energy, such as infrared light or ultrasonic waves to probe the scene and compute the distance between the sensor and various points in it. Here we consider a passive AF, more precisely a contrast detection algorithm.

B. Simple, parallel Auto-focus algorithm

To optimally match the capabilities of a CPA vision-chip, we construct an algorithm that relies on a computation performed at each single pixel to determine whether it is in-focus, using only information about the values of that pixel and its direct neighbours.

Principle: Because being out-of-focus blurs together neighbouring pixels, one can find the optimally focused frame for a given pixel, by analysing the average difference of that pixel to its direct neighbours. Our algorithm finds the optimal focus for a given pixel p in a set of frames in a two step procedure: For each frame taken while sweeping the focus compute the average difference of p to its direct neighbours. Find the frame for which this value is maximal; this is the frame with optimal focus. A mathematically equivalent formulation is to compute the Laplacian (i.e. the divergence of the gradient) of the brightness at each pixel. To get a smoother estimation of focus pixel-wise, the Laplacian can be averaged in a local neighbourhood by convolving it with a Gaussian Kernel thus obtaining the Laplacian of Gaussian (LoG).

In practice an AF needs to be able to focus multiple pixels simultaneously. To achieve this we could simply maximize the sum over the LoG for all pixels in the region of interest. This however has the drawback of preferentially putting high contrast edges in focus, rather than low contrast ones; Figure 3 shows an example of varying peak values of the Laplacian for varying contrasts. One method of alleviating this is to acquire the image with the previously described HDR tone mapping algorithm, which performs a global contrast equalization. In order to make the estimation of the focus even more robust to contrast change across the image one can threshold the LoG and maximize the sum over the thresholded LoG.

On chip regions of interest for the autofocus can be selected by turning-on the activity-flags of our CPA only for the pixels belonging to this region.

Finally the AF should be able to guide the focus sweep as it is happening: This is made possible with this approach. As Figure 3 shows, the Laplacian is almost perfectly concave; thus the change of the Laplacian between two subsequent frames is a reliable indicator of whether the focus sweep is moving towards better or worse focus.

IV. Implementation & Results

We implemented both our adaptive-sensing HDR tone mapping and parallel AF algorithm on CPA vision-chips from the SCAMP family [13], [14].

The SCAMP device we consider here is a mixed-signal CPA vision-chip consisting of an array of cells/processing elements (PEs). A single PE includes a light-sensitive register, 6 local registers and a common register to share information with its 4-adjacent neighbours. In addition, the cell's PE contains global I/O and a comparator feeding an activity-flag latch. Analogue switched current techniques are employed to implement the PE: arithmetic operations are then performed without the need of a complex ALU. The device is programmable: instructions are dispatched by an external global controller to all the cells. Each of them performs the given instruction on their local data thus implementing SIMD processing. The state of the activity-flag can enable or disable the operation of the cell preventing it from performing the instruction.

HDR tone mapping: Figure 1, in which we image a scene with a high radiance range, illustrates how the HDR tone mapping can preserve information that would be lost to a standard sensor. As can be seen in the figure, it is impossible to simultaneously expose the top of the table and the rest of the scene, without using a HDR approach.
In Figure 2, we see the effect of the HDR tone mapping applied to two indoor scenes of different illuminance. Also, note that the analogue registers on the SCAMP device store analogue samples, but their values tend to decay over time, and this is not uniform across the array. Hence some noise appears in the image when maintaining an image for too long in a register; this issue typically arises when a long exposure time is needed. The HDR algorithm we proposed and implemented depends on a maximal integration time controlled by the hyper-parameter N. This entirely determines its latency since processing is happening when integrating. This typically ranges between less than a frame per second (FPS) =1s of integration in dark conditions to more than 25 FPS (40ms) in bright conditions. The exposure times for pixels vary across the image between few microseconds (20µs) and fractions of seconds, the range of captured light intensities exceeds $1 : 500\,000$.
The system running this algorithm draws 1.6W of power, only a third of which is drawn by the vision-chip.

Parallel AF: Results of our parallel AF can be seen in Figure 3 where different processors are selected with their activity-flags to focus on different parts of the image.
The algorithm is very lightweight, and has been designed to entirely benefit from the SIMD mode of the CPA. We demonstrated it running at 25 frames per second, but it can be easily executed at higher-frame rates, conditioned on enough incoming light. When running the algorithm using the HDR tone mapping as an input, the latency of the system is mostly dependent on the latter which drives the integration time.

V. Conclusion

We proposed a HDR tone mapping adaptive-sensing algorithm and a parallel AF algorithm relying on contrast detection. We implemented both algorithms on SCAMP CPA vision-chips and demonstrated that they worked in real-world conditions. These two examples illustrate how non-trivial problems can be successfully tackled with relatively simple algorithms when an implementation at the sensor level is considered. Algorithms that would not be feasible in a traditional image capture and processing system, due to excessive data bandwidth requirements between the sensor and processor, are enabled by the near-focal plane processing principles. These intelligent sensors, equipped with adequate algorithms, allow us to effectively address imaging problems that would otherwise require greater effort.

References

[1] R. F. Lyon, "The optical mouse, and an architectural methodology for smart digital sensors," in *VLSI Systems and Computations*, vol. 1, no. 81, 1981, pp. 1–19.

[2] M. Mahowald, "An analogue VLSI chip for stereocorrespondence," in *Proc. of the IEEE Internat. Symp. on Circuits and Systems, ISCAS'94*, vol. 6, 1994, pp. 347–350.

[3] P. Lichtsteiner *et al.*, "A 128×128 120dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[4] J. Martel *et al.*, "Toward joint approximate inference of visual quantities on cellular processor arrays," in *Proc. of the IEEE Internat. Symp. on Circuits and Systems, ISCAS'15*, 2015, pp. 2061–2064.

[5] E. Candès *et al.*, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2006.

[6] N. Cottini *et al.*, "A 33uW 64x64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 850–863, 2013.

[7] J. Fernandez-Berni *et al.*, "FLIP-Q: A QCIF resolution focal-plane array for low-power image processing," *IEEE J. Solid-State Circuits*, vol. 46, no. 3, pp. 669–680, 2011.

[8] A. Zarandy, Ed., *Focal-plane Sensor-Processing Chips*. Springer, 2011.

[9] W. Zhang *et al.*, "A programmable vision chip based on multiple levels of parallel processors," *IEEE J. Solid-State Circuits*, vol. 46, no. 9, pp. 2132–2147, 2011.

[10] W. Miao *et al.*, "A programmable SIMD vision chip for real-time vision applications," *IEEE J. Solid-State Circuits*, vol. 43, no. 6, pp. 1470–1479, 2008.

[11] M. Ishikawa *et al.*, "A CMOS vision chip with SIMD processing element array for 1ms image processing," in *Digest of Technical Papers, IEEE Internat. Solid-State Circuits Conf., ISSCC'99*, 1999, pp. 206–207.

[12] M. Laiho *et al.*, *Focal-plane Sensor-Processing Chips*. Springer, 2011, ch. MIPA4k: Mixed-Mode Cellular Processor Array, pp. 45–71.

[13] P. Dudek and S. J. Carey, "General-purpose 128×128 SIMD processor array with integrated image sensor," *Electronics Letters*, vol. 42, no. 12, pp. 678–679, 2006.

[14] S. J. Carey *et al.*, "A 100'000 fps vision sensor with embedded 535 GOPS/W 256×256 SIMD processor array," in *Proc. of the VLSI Circuits Symp.'13*, 2013, pp. C182–C183.

[15] P. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. of the 24th annual Conf. on Computer graphics and interactive techniques, SIGGRAPH'97*, 1997, pp. 369–378.

[16] R. Fattal *et al.*, "Gradient domain high dynamic range compression," in *Proc. of the 29th annual Conf. on Computer graphics and interactive techniques, SIGGRAPH'02*, 2002, pp. 249–256.

[17] J. DiCarlo and B. Wandell, "Rendering high dynamic range images," in *Proc. of the SPIE Electronic Imaging Conf.*, vol. 3965, 2000, pp. 392–401.

[18] J. Lenero-Bardallo *et al.*, "A high dynamic range image sensor with linear response based on asynchronous event detection," in *Proc. of the European Conf. on Circuit Theory and Applications, ECCTD'15*, 2015.

[19] P. Dudek, "Adaptive sensing and image processing with a general-purpose pixel-parallel sensor/processor array integrated circuit," in *Proc. of the the Internat. Workshop on Computer Architecture for Machine Perception and Sensing*, 2006.

[20] M. Mase *et al.*, "A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12 bit column-parallel cyclic A/D converters," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2787–2795, 2005.

[21] R. Wagner *et al.*, "High dynamic range perception with spatially variant exposure," in *Proc. of the IEEE Internat. Workshop on Cellular Neural Networks and their Applications, CNNA'04*, 2004.

[22] J. Fernández-Berni *et al.*, "High dynamic range adaptation for ROI tracking based on reconfigurable concurrent dual sensing," in *IET Electronics Letters*, vol. 50, no. 24, pp. 1832–1834.