Sensor-level computer vision with pixel processor arrays for agile robots

DOI:
10.1126/scirobotics.abl7755

Document Version
Accepted author manuscript

Citation for published version (APA):
Dudek, P., Richardson, T., Bose, L., Carey, S., Chen, J., Liu, Y., Greatwood, C., & Mayol-Cuevas, W. (2022). Sensor-level computer vision with pixel processor arrays for agile robots. *Science Robotics*, *7*(67), eabl7755. Article eabl7755. https://doi.org/10.1126/scirobotics.abl7755

Published in:
Science Robotics

Sensor-level computer vision with pixel processor arrays for agile robots*

Piotr Dudek[1*], Thomas Richardson[2], Laurie Bose[3], Stephen Carey[1], Jianing Chen[1], Colin Greatwood[2], Yanan Liu[3], Walterio Mayol-Cuevas[3]

[1]Department of Electrical Engineering and Electronics, The University of Manchester; Manchester, United Kingdom; [2]Department of Aerospace Engineering, University of Bristol; Bristol, United Kingdom; [3]Department of Computer Science, University of Bristol; Bristol, United Kingdom; *Corresponding author. Email: pdudek@manchester.ac.uk

Abstract: *Vision processing for control of agile autonomous robots requires low-latency computation, within a limited power and space budget. This is challenging for conventional computing hardware. Parallel processor arrays (PPAs) are a new class of vision sensor devices that exploit advances in semiconductor technology, embedding a processor within each pixel of the image sensor array. Sensed pixel data is processed on the focal plane and only a small amount of relevant information is transmitted out of the vision sensor. This tight integration of sensing, processing, and memory within a massively parallel computing architecture leads to an interesting trade-off between high-performance, low-latency, low-power, low-cost and versatility in a machine vision system. Here, we review the history of image sensing and processing hardware from the perspective of in-pixel computing and outline the key features of a state-of-the-art smart camera system based on a PPA device, through the description of the SCAMP-5 system. We describe several robotic applications for agile ground and aerial vehicles, demonstrating PPA sensing functionalities including high-speed odometry, target tracking, obstacle detection and avoidance. In the conclusions, we provide some insight and perspective on the future development of PPA devices, including their application and benefits within agile, robust, adaptable and lightweight robotics.*

INTRODUCTION

The key to further advancements in autonomous robotics is the ability to move quickly and safely through every day, and sometimes hazardous, environments. Vision is one of the primary sensing modalities through which robots can perceive their surroundings, however, real-time visual information processing is notoriously difficult, especially at the speed required for fast moving robots, and in particular where low weight, low power consumption and cost of the system are of concern. Off-the-shelf sensors and processor hardware are often too slow, too large, or too power-hungry for the task. The reasons for this are primarily related to the way data moves through the system. A conventional vision system is illustrated in Figure 1A. In this system, an image sensor in the camera device acquires visual information, producing video frames that are sent through to the processing hardware. This hardware typically consists of standard microprocessors, i.e. Central Processing Units (CPU), often augmented by Graphics Processing Units (GPU) or other specialised video processing hardware (e.g. Nvidia Tegra , Intel Movidius, etc.) Although there is continuing progress in improving this kind of hardware's speed and efficiency, a fundamental limitation comes from the sensor-processor bottleneck. Massive amounts of visual data are acquired, digitised, and then sent from the camera device to the processor, and then on throughout the processing system. This limits the latency and power dissipation of the system.

To overcome this limitation, we need to move data processing nearer the sensor, as shown in Figure 1B. The role of the *vision sensor* here is not simply to acquire the visual signal, but to digest it, producing meaningful, highly-compressed information, instead of video frames. This could be, for instance, extracted features, keypoint locations, optic flow direction, target location, object identities, or other high-level data extracted from the acquired images and relevant to the application. The resulting data reduction at the sensor level not only speeds up the data transfer, but also reduces the demands on the computing hardware that follows downstream. Clearly, for this to work, the near-sensor processing hardware needs to provide a combination of high computational performance and low power dissipation. The processing circuits are typically parallel digital signal processing units, or more specialised hardware. For example, recent interest in applying deep learning to vision processing has resulted in numerous "AI accelerators", typically employing parallelised multiply-accumulate circuits dedicated to convolutional neural network computation. These can be placed in the vicinity of the sensor. In the most straightforward version, the vision chip can be

constructed by integrating the image sensor array and processing circuitry on the same silicon die (Figure 1C), or through 3D wafer integration technology that allows the stacking of the sensor and processor silicon chips in a single package (1). Keeping sensing and processing in close proximity enables large sensor-processor bandwidth and reduces the power associated with signal communications over large distances. This is clearly advantageous, but the benefits of putting together the separate sensor and processor circuitry on a single device stem simply from the miniaturisation of the conventional visual pipeline, fundamentally, the sensor-processor bottleneck is still there. The present-day microelectronics technology allows us to eliminate it altogether, through re-engineering of the sensor circuitry at the pixel level. The approach surveyed in this article goes beyond co-locating image sensor and processor devices, it puts computation hardware right at the place where the images are acquired, into the image sensor pixels themselves (Figure 1D).

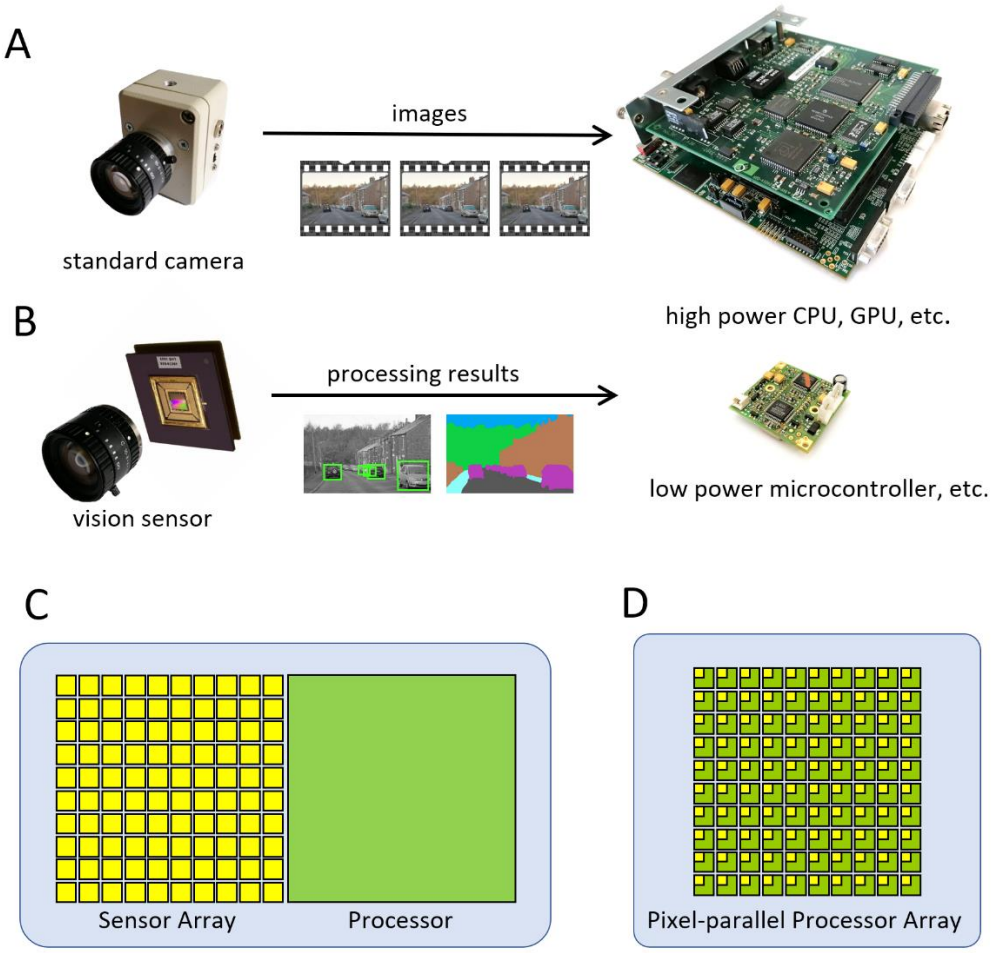

Figure 1. Comparison of approaches to vision systems. (**A**) Conventional vision system consists of a camera, including image sensor device, that sends a continuous stream of images, frame by frame, to a processor system, often implemented using powerful CPU, GPU or specialised hardware. (**B**) A vision sensor based system carries out substantial image processing on the sensor device itself, sending pre-processed high-level information to a processor system that can be implemented using much simpler computational hardware. (**C**) A vision sensor device can be constructed by integrating sensing and processing circuits on a single silicon chip. (**D**) In a pixel-parallel processor array, processing circuits are integrated directly into the pixels of the image sensor.

IMAGE SENSOR TECHNOLOGY EVOLUTION

To appreciate the possibilities of pixel-level processing, it is useful to reflect on the progress made in image sensor technology over the past few decades. It is punctuated by the increasing sophistication of in-pixel circuitry, as illustrated in Figure 2. Solid-state electronic imaging started with CCD (charge-coupled devices) technology, that enabled the construction of semiconductor image sensor devices that converted photons to

electrons, accumulated them in pixels, and then shifted-out the charge packets, "bucket brigade" style, to an external device, which converted the analog charge to a digital signal. The pixel circuit contained no active devices (transistors), but only an arrangement of potential wells defined by semiconductor doping and electrodes. Further advances were possible due to utilisation of CMOS (complementary metal-oxide-semiconductor) fabrication technology, similar to that used for state-of-the-art silicon chips. CMOS image sensors were initially built using a photodiode and one transistor, later integrating 3 or 4 transistors in each pixel, enabling more sophisticated buffering and read-out circuitry at the pixel level (2), as well as integration of noise suppression and analog-to-digital conversion (ADC) on the same silicon device. Leveraging the general progress in silicon fabrication, scaling to ever-decreasing feature sizes, this technology has been continually improving, and has led to the proliferation of image sensors in smartphone cameras, webcams and other consumer devices. Nowadays, in state-of-the-art image sensors, the photosensors and underlying transistor circuitry are vertically integrated, with backside illuminated (BSI) photodiodes on one silicon wafer, connected using copper-copper (Cu-Cu) bonds, to the readout, interface and processing circuits on another silicon wafer (3). The technology allows per-pixel connectivity between the top (photosensitive) and bottom (additional processing) wafers, for instance to achieve a global shutter function (4). Although the main drive for developing this technology is to increase the light-sensitive pixel area whilst improving image resolution and reducing the overall size of the image sensor chip, it offers an intriguing possibility to integrate tens or even hundreds of transistors directly underneath each pixel. It has enabled, for instance, the inclusion of analog-to-digital conversion and local memory in each pixel of the image sensor (5,6). Furthermore, the in-pixel processing does not need to be limited to signal buffering and data conversion. Some of the most interesting recent developments in the field of image sensors are Dynamic Vision Sensors (DVS) (7,8), where the in-pixel circuitry includes detection of temporal intensity changes, and signalling of these changes to the outside world through binary "events". In effect, only the pixels that detect light intensity change transmit data, providing data compression at the sensor level and the ability to achieve low-latency operation, as individual pixel changes can be transmitted without ever reading-out full-frame image frames. Sensor devices based on this idea are becoming commercially available, and have been used in several high-speed robot demonstrations (9-13). The modification of the sensing strategy, beyond conventional frame-based image sensing, can be advantageous, but on its own, it does not solve the problem of achieving efficient computation.

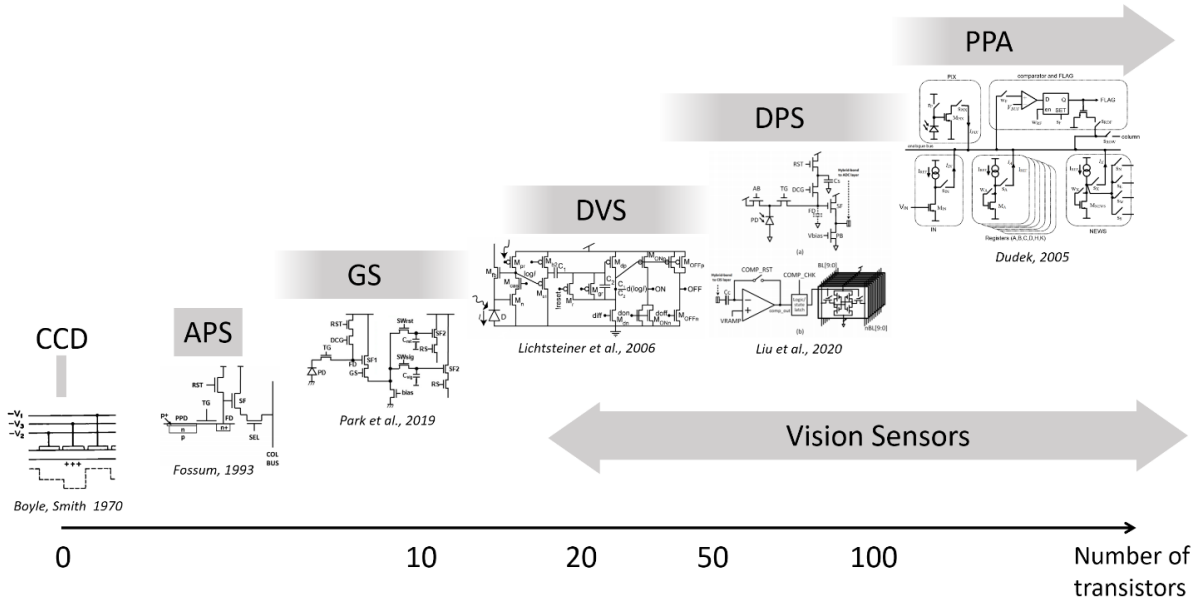

Figure 2. Pixel-level circuits in image sensor devices. Charge Coupled Devices (CCD) image sensor pixels are transistor-less (104), early CMOS image sensor pixels contained 1 transistor, modern Active Pixel Sensor (APS) CMOS devices include 3-4 transistors per pixel (2), Global shutter (GS) sensors contain up to 10 transistors (4, 105), Dynamic Vision Sensor (DVS) devices include 15-30 transistors per pixel (7,8), Digital Pixel Sensors (DPS) include around 100 transistors in ADC converters and memory in each pixel (5,6), Vision Sensors include 20-500 transistors in a processing circuit per pixel (16-25), Pixel Processor Arrays (PPA) contain pixel-level processors including 150+ transistors (31-33,48-49).

VISION CHIPS WITH PIXEL LEVEL PROCESSORS

The potential to integrate sophisticated in-pixel processing in a CMOS imaging device has been recognised by researchers early on, and numerous academic prototypes have been implemented over the years. Many of these devices have been optimised to perform some useful image processing operation using bespoke transistor circuits embedded into the pixels of the imaging array. These include the pioneering work of Mahowald on local contrast detectors (14). The in-pixel processing here is modelled on the way an animal retina carries out certain image processing tasks right next to the photoreceptors, before sending the pre-processed data down the optic nerve to the brain (15). The natural vision system optimisations, carried out by evolution, indicate the advantageous aspects of near-sensor massively-parallel computation, and bio-inspired circuits have been a feature of several vision sensor developments (16-18). Others have followed more abstract models, computing certain useful image processing operations such as motion detection (19,20), optic flow (21,22), gaussian pyramid extraction (23), background subtraction (24,25), etc. In these developments, the function of the device is fixed by the hardware circuit implemented in each pixel. Some configurability has been demonstrated, for instance by implementing programmable convolution kernels (26-28) or local binary patterns (29,30), but in general, although special-purpose vision sensors can be efficient, their application is limited due to the pre-defined functionality of the device.

The concept of in-pixel processing can be taken one step further, increasing the sophistication and flexibility of the device, with the integration of a complete (albeit simple) programmable processor core within each pixel of the image sensor. We term such a device a Pixel Processor Array (PPA). The concept is illustrated in Figure 3. It has been demonstrated that a usable pixel-level processor can be constructed with fewer than two hundred transistors (31-33). Each pixel-processing element includes local memory and arithmetic-logic circuits that implement elementary operations. As most low-level vision algorithms are data-parallel, with identical instructions executed for each pixel in the image, a suitable computer architecture is that of a SIMD (Single Instruction Multiple Data) machine, with one controller broadcasting microinstructions to the processor array. Local communications allow pixel-processors to exchange data with their nearest neighbours.Local activity flags allow for conditional data-dependent program execution.

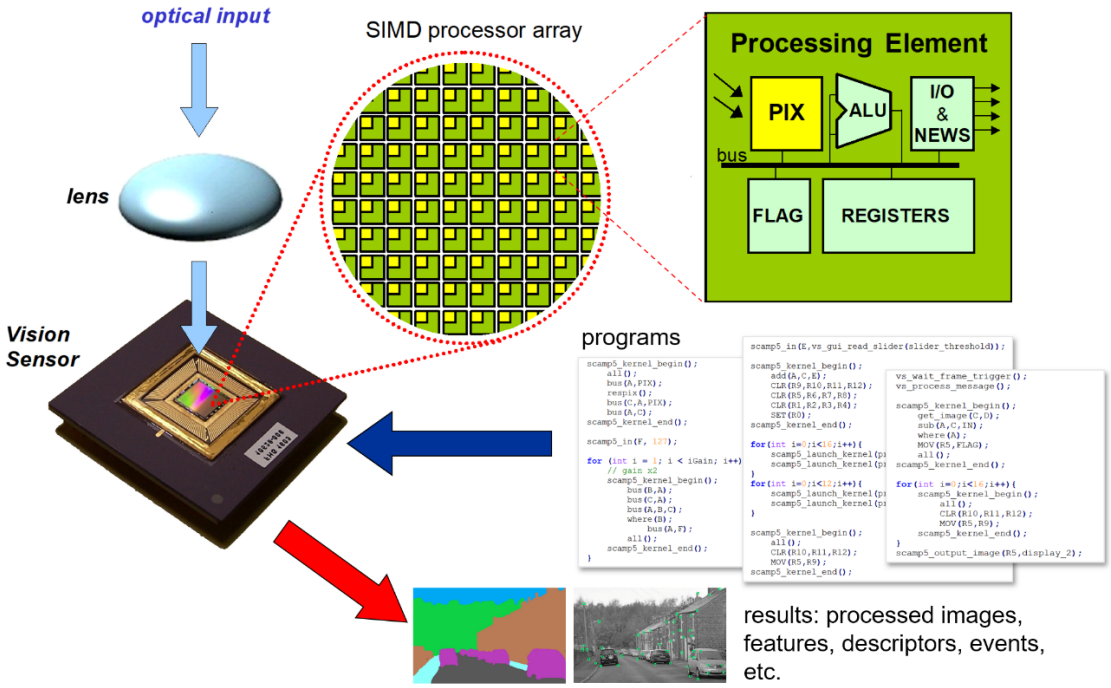

Figure 3. Pixel Processor Array (PPA). Each pixel of the Vision Sensor device incorporates a Processing Element, including photosensor circuit (PIX) and a microprocessor-like datapath, including local memories (Registers), arithmetic-logic unit (ALU), local communications (NEWS), I/O circuits and local activity control (FLAG). The processing elements execute software programs, carrying out image computations on-sensor and outputting high-level information to the rest of the system.

With this approach, the functionality of the vision sensor device is no longer determined at the design stage through the specific hardware circuitry embedded into pixels, but it is determined entirely by the software that is executed on the pixel-processor hardware. The system therefore becomes a general-purpose device, capable of implementing a large variety of computer vision algorithms right inside the pixels of the image sensor.

Such an approach is not only universal, it is also efficient in terms of hardware use. Multiple functionalities are achieved in a single device, through in-pixel hardware that is time-multiplexed. With a limited number of in-pixel transistors (comparable with that of many special-purpose vision sensors), the PPA offers vast algorithmic possibilities of a general-purpose universal machine. In practice, the specific microarchitecture, amount of local memory, and other implementation details, will constrain the scope of possible algorithms that can be implemented.

PIXEL-LEVEL SIMD COMPUTING

The idea of pixel-parallel computing for computer vision, using SIMD arrays, dates back to the early days of computer hardware, with the development of pioneering processors such as ILLIAC IV (34), CLIP (35), BLITZEN (36), motivated, to a large extent, by the data parallelism inherent in image processing tasks. A massively parallel, fine-grained processor array is ideal for executing low-level vision computations that require identical operations to be performed on every pixel in the image. The nearest-neighbour processor communications in a 2D array are also ideal for image processing tasks such as convolution kernels, that compute results based on spatially localised pixel neighbourhoods. The mapping of these algorithms onto pixel-parallel arrays is straightforward, and large total computational power (operations per second) can be easily achieved, with thousands of processors working in parallel. At the same time, greater power efficiency (operations per watt) is the result of processor-memory colocation in these processor arrays. Despite these advantages, the technology at the time did not allow the construction of integrated circuits comprising sufficiently large 2D processor arrays to fulfil the promise of fine-grain massively parallel computing. The SIMD ideas have evolved through early vector supercomputers (37), to multi-argument ALU's in digital signal processors and to "multimedia extensions" on commodity microprocessors (38). They have also inspired the development of general-purpose GPU architectures. Nowadays, the high-performance specialised computing hardware, dealing with data-parallel problems, often involves SIMD execution units, but the integration of fully pixel-parallel SIMD arrays remains a challenging problem. This is due to the very tight area constraints needed to integrate practical image-size arrays on a single chip. Silicon technology however is catching up, bringing image-size pixel-parallel SIMD computing into the realm of possibility.

Pioneering work on the integration of digital SIMD processor arrays into pixels of the imaging array has been reported in (39-42). The processing elements on the devices contained a few bits of memory, and bit-serial (i.e. using 1-bit datapath) processing circuitry. Later developments (31,43) have extended local memory to 64-bits. This is sufficient to perform many low-level image processing operations on gray-scale images. As computations are performed using digital logic circuits, the analog photosensor signal needs to be digitised, and a comparator-based ramp ADC circuit is typically employed for this purpose in each pixel.

It should be noted that the majority of special-purpose vision chips reviewed earlier (16-30) have been implemented using analog processing circuitry. This has the advantage of a direct interface to the analog image sensor, and compact implementation of various computational primitives (44). However, the functionality of these analog devices is baked into the silicon, fixed at the point of their circuit design. Some level of configurability in traditional analog systems can be achieved by adjusting parameters, or re-routing (switching) connections between hardware units, as in continuous-time "analog computers" (45) used decades ago to solve differential equations before digital computing technology took over. It is commonly assumed that software-programmable computers are a "digital" technology, however, an important insight is that it does not need to be the case. The first work on an analog instruction set computer was probably done by Masuda, Yoneda and Kasai (46). Our early work (47) introduced the idea of an *analog microprocessor*, a software-programmable system that has a microarchitecture akin to that of a digital processor, but where datapath operations are performed using analog switched-current circuits. This has enabled the

implementation of pixel-parallel SIMD processor arrays with analog (32,33) and mixed-signal processing elements (48), reaching performance levels and array sizes sufficient for practical applications (e.g. the 256 x 256 array reported in (49)). Related work based on processing elements with the analog implementation of convolution kernels embedded into a SIMD control structure, has been reported in (50-53) and column-based programmable analog compute units were used in (54).

In addition to fine-grained, processor-per-pixel SIMD architectures, several recent developments have explored alternative strategies, for instance using column-parallel processors (55-58), multiple pixels per processor (59,60), or combinations of pixel-parallel and more coarse-grained processor arrays (61,62). It remains to be seen which of these architectural alternatives will prove to be most useful in practice. Currently, the silicon integration technology still places severe limits on the amount of circuitry that can be placed inside pixels. Furthermore, the vast majority of academic prototypes have been built using standard planar CMOS technologies which compromises light sensitivity and image resolution of a pixel-parallel processor. Implementations using 3D wafer-stacked integration technologies have been considered, e.g. (55,60,63,64). This has many advantages, as illustrated in Figure 4. With the increasing commercial availability of these technologies, it can be expected that the pixel-level processing will become more widely used in the near future.

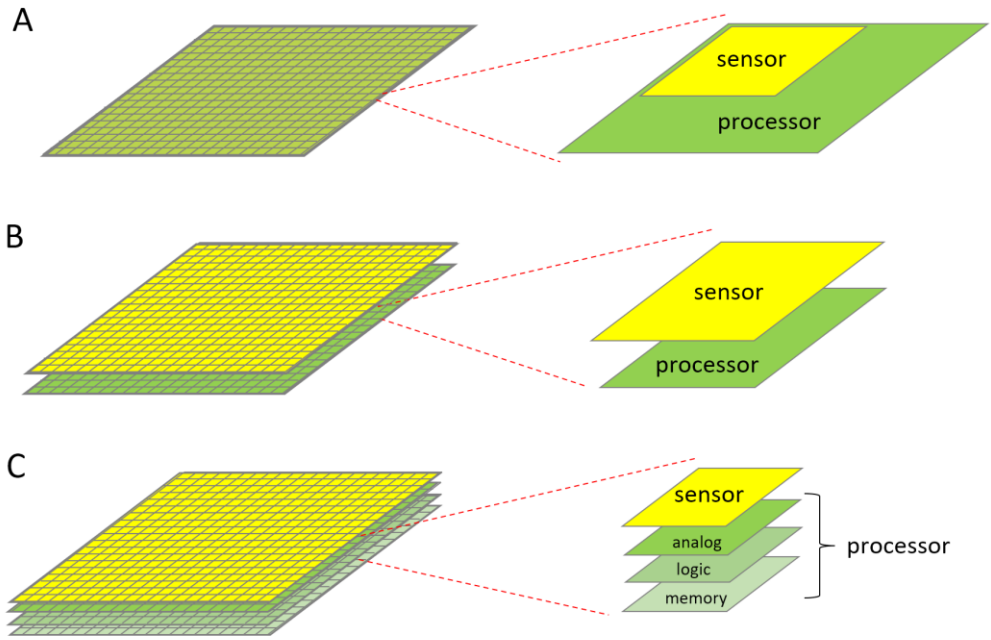

Figure 4. Pixel-parallel vision sensors in 3D stacked technologies. (**A**) In a standard planar CMOS technology, sensors and processors share pixel area in a 2D space, (**B**) Current Cu-Cu wafer bonding technologies allow placing processor circuitry underneath the image sensor. Light sensitive area is maximised whereas overall pixel size is reduced. (**C**) Advanced 3D stacking technologies may allow the distribution of in-pixel processors over multiple silicon tiers. Each tier could be fabricated using process technology optimised for sensing, analog circuits, digital logic, memory, etc.

THE SCAMP-5 VISION SYSTEM

Here we briefly overview the SCAMP-5 smart camera system. This system has been used in many of the robotic applications reviewed later in this article and represents the state-of-the art in PPA technology in terms of chip design, system design and the associated software development kit.

The system is shown in Figure 5. Whereas the core PPA functionality is provided by a custom integrated circuit, the SCAMP-5 vision chip (49), most of the peripheral components (a microcontroller, FPGA device, ADC and DAC converters, etc.) have been implemented using off-the-shelf electronic devices assembled on

a printed circuit board. This substantially increases the size and power consumption of the entire camera system, which may limit some practical applications at this point, but provides an easy to use research prototype. It can be assumed that the system could be feasibly shrunk to a much smaller size, with one System-on-a-Chip integrated circuit and a few external components, in a technically straightforward, albeit laborious development task. The SCAMP-5 vision chip comprises a 256x256 SIMD processor array and interfacing circuitry. The processor array integrates sensing, processing and memory, and executes instructions broadcast to it by an external microcontroller unit (MCU) device. In this system, an ARM M0 processor core is used for this purpose. The camera system also has another processor core (M4) that is used for interfacing to the host system, especially during debugging and program development, but can be also used to execute high-level application code in tandem with the SCAMP-5 device.

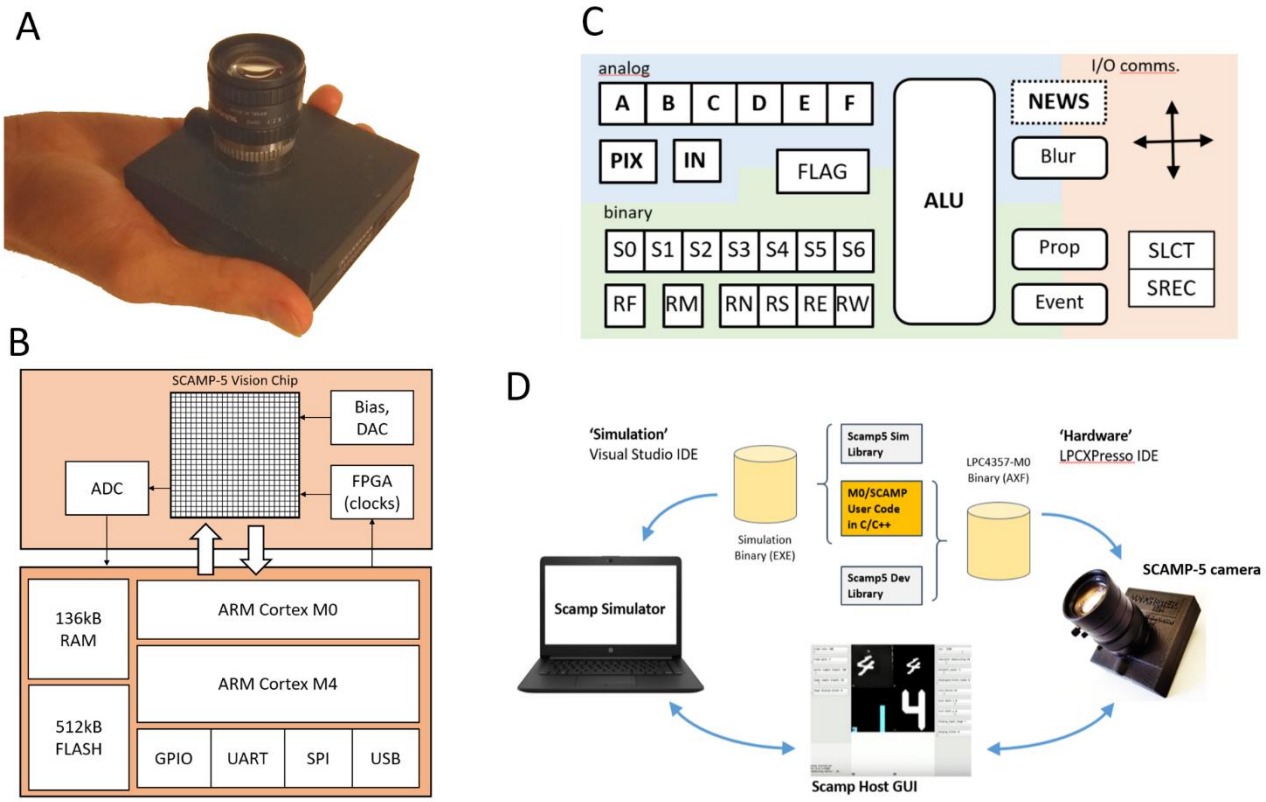

Figure 5. SCAMP-5 smart camera system, used in many of the applications reported in this article. (**A)** Prototype camera. (**B**) System architecture. At the core of the system is the SCAMP-5 vision chip comprising an array of 256x256 processing elements, control and readout circuits. The camera integrates peripheral circuitry including biasing, clocking, analog/digital converters, and microcontroller unit comprising two 32-bit processor cores, memory and interfaces. (**C**) Architecture of the processing element on the SCAMP-5 chip, it includes analog and binary registers, some special-purpose registers, arithmetic-logic unit, interface and control circuitry; processors communicate directly with 4 nearest neighbours. (**D**) Software development and debugging tools, including system simulator, and user interface, enable construction of libraries of useful image processing functions, that can be then joined into a complete vision application.

It needs to be emphasised that the success of unconventional computer hardware in applications such as robotics requires considerable effort developing software tools and libraries to enable easy application development and streamline integration into existing systems. On the SCAMP-5 system, the overall program is written in C++ then compiled and executed on the MCU, while PPA operations are triggered by execution kernels (coded in assembler, using a custom instruction set) that issue appropriate control signals to the processor array. The PPA acts as a co-processor to the MCU, acquiring images and operating on data located in its internal array registers.

The SCAMP-5 chip executes instructions at a rate up to 10MHz. This relatively low clock frequency helps to achieve low power consumption, while the total peak performance of the 65,536 parallel processor cores

reaches 0.65 TOPS (trillion operations per second). The peak power consumption of the processing cores is 1.25W, yielding efficiency of 0.5 TOPS/W. The chip has very low static power consumption of 0.2mW, and therefore low power operation in a mW range can be achieved on less demanding tasks at lower frame rates (65). The performance and efficiency metrics are particularly impressive if we consider that this chip was fabricated in a two-decades old silicon technology (a 180nm CMOS process which was commercially available in 1998). For context, a 32-bit ARM M0 processor core fabricated in a 180nm technology executes 0.012 TOPS/W. The NVIDIA Tegra X2, benefiting from a much more efficient 16nm FinFET technology, provides peak performance of 1.5 TOPS (16-bit floating point operations), while consuming 15W, i.e. 0.1 TOPS/W. However, caution should be taken when using metrics such as "operations per second", especially when dealing with very diverse architectures and data formats. In particular, SCAMP-5 arithmetic operations are executed in an analog datapath, with the precision equivalent to about 8-bit integer (49) but involving peculiarities of analog signal processing in terms of computation error and noise. The achievable performance on actual tasks is always a much better indication of the capabilities and efficiencies of unconventional computing systems.

A notable feature of the SCAMP-5 array are its *global* and *event-based* readout modes. Although the results of image processing can be read-out as binary or gray-level image frames, which is often done during program development/debugging phase, the most powerful feature of the PPA is that the images can be processed internally, and only the results of computations are ever transmitted out of the sensor device. These might be, for instance, only spatial coordinates of a sparse set of white pixels, in an otherwise black image, or a scalar representing a summation of an entire image-sized array of numbers. A sparse region-of-interest image readout is also possible. These readout modes allow operation at very high frame rates, e.g. over 1000 frames per second (fps), with heavy-lifting done by the PPA array, and the rest of the control algorithm executed on the MCU.

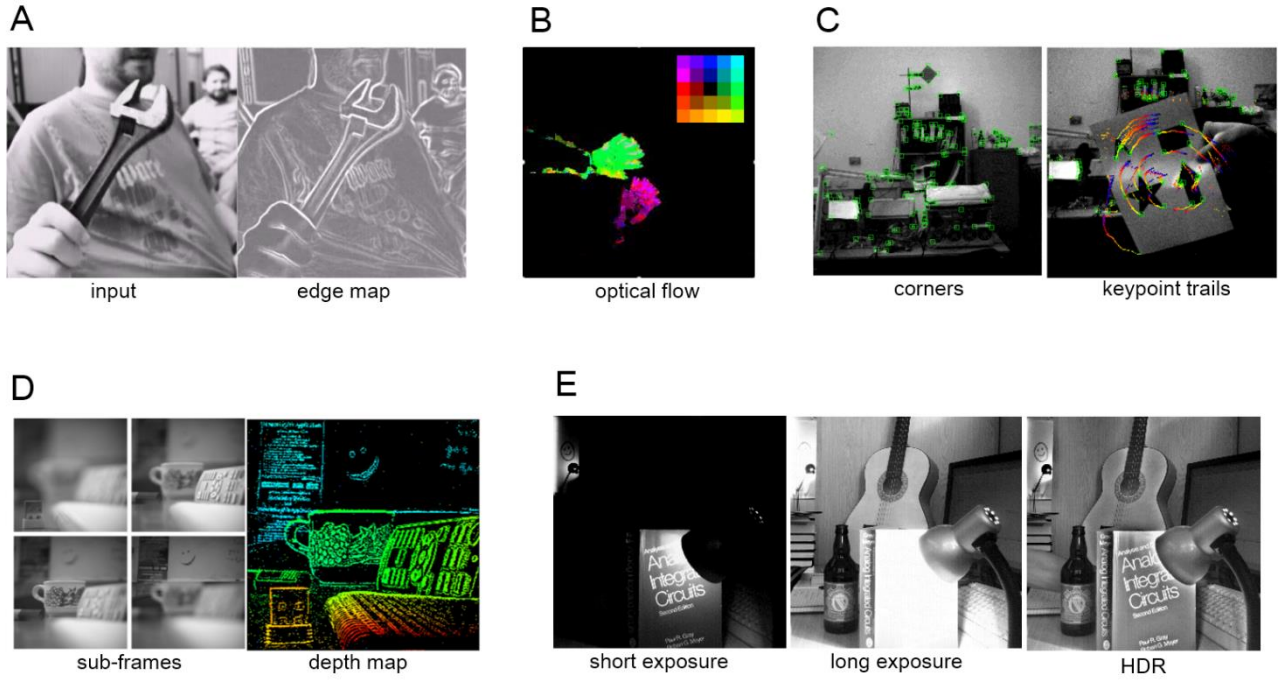

Figure 6. Image processing algorithms executed on the SCAMP-5 PPA. (**A**) Sobel edge detection, execution time 5.8μs; (**B**) Optical flow, image shows two objects moving in different directions, inset shows directional colour coding, block matching algorithm using 5x5 blocks matched over 5x5 grid takes ~0.4 ms; (**C**) Corner detection using FAST-16 algorithm. Keypoint trails show past keypoints, the test card was rotating. Images are output for illustration, when outputting only keypoint coordinates the system can operate at 2000 fps (68); (**D**) Depth from focus, subframes are acquired by sweeping the focus of a liquid lens at 60 Hz, up to 128 sub-frames are processed to determine a combined depth map at 60 fps; in the image, red is near, blue is far; this requires spatial contrast maximisation algorithm running at 7000 fps (67); (**E**) High dynamic range imaging using tone mapping by combining hundreds of frames; the image is progressively exposed to achieve locally balanced image intensity at each pixel (66).

Figure 6 illustrates several basic algorithms, executed by the SCAMP-5 system, and their execution time. Basic pixel-parallel operations such as convolution filters, corner extraction or optic flow computation are easily implemented in a PPA. As video frames are processed internally, algorithms that require high-speed transfer and processing of a very large number of images per second, and are not feasible in conventional systems, can be considered. For instance, high-dynamic range (HDR) imaging can be based on acquiring hundreds of images, to generate one combined tone-mapped image frame (66). Monocular depth mapping can be achieved, using a lens with fast-sweeping focus, using the PPA to compute when objects come in and out of focus (67). The locations of corner features can be extracted at a rate of several thousand frames per second, outputting only coordinates of extracted keypoints (68). Bandwidth reduction is one important opportunity for on-sensor processing. For example, consider feature extraction at 2000 fps. If the outputs are the (x,y) coordinates of corner features (16 bits), and 250 features are read-out per frame, that results in data bandwidth of 8 Mbps (mega bits per second). This represents data reduction of 128x as compared with 8-bit gray-level image read-out, that would require over 1 Gbps at this frame rate. The PPA is a versatile system, and relatively complex algorithms can be implemented entirely on-sensor, for instance object tracking (65,69), segmentation (70), or implementations of convolutional neural networks (CNN) for classification (71), or gesture recognition (72). In all these applications, image arrays are reduced to a computation result by the PPA, and only the final answer (one, or several bytes) is transmitted off chip per each image frame.

APPLICATIONS IN AGILE ROBOTICS

Vision sensors promise high-speed, low-latency and low-power computing in a compact package. These are very desirable features from the point of view of the requirements of mobile robotics, in particular on fast moving, agile and lightweight vehicles. With the bulk of processing and substantial data reduction done on-sensor, a simple microcontroller-based processing system may be sufficient to carry out the remaining processing tasks, while still achieving high frame and control system rates. For example, optic flow chips have enabled low-weight, low-power vision systems to be constructed to facilitate bio-inspired flight control strategies (73,74). There has also been notable progress in applying DVS devices to robotic demonstrations, highlighting the low-latency capabilities of these devices (9, 10) including applications on agile aerial vehicles (11-13).

In this review, we will focus on applications of fully-programmable PPA devices to agile robotic systems. The use of PPA devices in high-speed robotics has been pioneered by Ishikawa et al. (75), with several demonstrations related to target tracking and visual servoing. The use of CNN-based vision chips on mobile platforms has also been considered (76). Many of the early demonstrations suffered from the limited performance of early-days hardware and lack of software development tools, making the development of applications exploiting PPA capabilities challenging. The introduction of the SCAMP-5 system has provided not only a high performance plug-and-play hardware system, but also a comprehensive software development suite including simulation tools and a C++ based development flow (77). A variety of hardware interfaces, including Universal Serial Bus (USB) and Serial Peripheral Intrerface (SPI), a software stack, including custom Applications Programmers Interface (API), integration into Robot Operating System (ROS), and remote Transmission Control Protocol/Internet Protocol (TCP/IP) based interface, allows for the integration of the system into various hardware configurations. Our research work (78) has demonstrated how PPAs can be meaningfully used for a variety of key robotic tasks. These range from object tracking to visual odometry to pictorial mapping all on-sensor.

Agile object tracking and detection

An agile platform tracking another moving object serves as a testbed for rapid reaction and closed loop control, especially if the motion is unpredictable. In (79) a vision-based control strategy is implemented for tracking a mobile target. The vision algorithm running on the SCAMP-5 PPA recognises the target in the image, extracts the location, and transmits the coordinates to the host system. The strategy enables a small, agile quadrotor to track a wheeled vehicle from close range using minimal computational effort (Fig. 7A). In this case, the vehicle follows dual-pendulum chaotic trajectories whilst the observing drone is tasked to keep the vehicle in view under high acceleration and rapid changes in direction. A state observer is used to smooth out the PPA predictions of the target location and, importantly, estimate its velocity. Experimental results also demonstrate that it is possible to continue to re-acquire and follow the target during short periods of loss

in target visibility. In a similar manner to target tracking it is possible to also task the PPA with fast object detection. In (80), we demonstrate an agile car moving at speeds of 4 m/s involving rapid changes of direction based on PPA target detection. The vehicle makes fast decisions on the path to follow through detecting coded gates in a manner akin to slalom skiers by recognizing the side, left or right, that the vehicle should follow (Fig. 7B). All vision processing is done on the PPA, which transmits object locations to the host controller, at rates exceeding 2000 frames/second. Furthermore, one area that has been attractive due to its challenge for human pilots is drone racing. International challenges have been created to push drone agility accordingly. In (81) we demonstrate an agile drone moving on a race course with high acceleration, going through gates (Fig. 7D). This uses a combination of a roughly known map of the race track together with local gate pose estimates computed on the PPA. Another approach to path planning and visual obstacle detection has been presented in (82). Demonstrating the strategy for combining PPA-based feature extraction and neural network post-processing, the PPA extracts a series of visual features from the image frames, summarising these into a compact representation describing total feature intensities in 10 receptive fields (image regions). A classifier is then trained on these feature descriptors and infrared proximity sensor data, such that the proximity detection can be eventually achieved using vision data only.

On-sensor visual odometry (VO)

Estimating relative motion is a useful competence for planning and mapping in robotics. Conventional visual odometry algorithms rely on sparse features (83) in part due to computational constraints when using traditional visual pipelines. This often results in challenges with viewpoint invariance for the few features selected. However, with a PPA operating at much higher framerates, the visual algorithm is simplified as visual search can be denser and operate on a much reduced area from one frame to another. In (84) we demonstrate visual odometry (VO) operating at over 1000 fps on the focal plane. The algorithm is massively parallelized and applies a motion model to estimate the image distortion that best matches the previous image with the current one. This enables basic egomotion estimation on-sensor (Fig. 7C). Importantly we do not require additional computation and the motion parameters are the output of the PPA. Alternative PPA VO algorithms that use a feature-estimation approach have also been developed by others (85). The algorithm in (84) has been extended to correct the image perspective for a quadrotor according to its inertial measurement unit (IMU) (86). In this case, the PPA demonstrates how it can receive input from an external sensor (IMU) and use this information to make changes in-real-time to its processing by perspective correcting the current frame accordingly. This further demonstrates the simplifications on the visual algorithm achieved by the fast and massively parallel nature of the SCAMP-5 PPA. Another key capability of a PPA is the potential to perform more than one function or competence on a single sensor, primarily enabled by the on-sensor processing power available. In (87) we combine visual odometry with HDR imaging and target detection to reset the VO estimates for a drone surveying a target area. This functionality would be appropriate in a GPS denied or limited availability setting.

Pictorial mapping and localisation

Another fundamental competence in robotics is the ability to map an environment and use that map for knowing where the platform is. This assists with motion planning and task execution. In terms of mapping there are various representation options, from full 3D or higher dimensional maps to image-only or *pictorial* maps, e.g. (88,89), which via a collection of images can describe a space in a topological manner. As the robot can locate itself with respect to such a map, the platform has the ability to perform navigational tasks even if the map is metrically inconsistent. In (90) we demonstrate how the SCAMP-5 PPA can perform both pictorial mapping and localisation entirely on-sensor and at the point of image capture, outputting for each input frame an index relative to the stored images. Mapping here consists of collecting images based on criteria of sufficient difference from the previous image. This assumes the platform is navigating on a path and for a purpose, i.e. not undergoing Brownian motion. The PPA will then store, in its pixel-level distributed memory, a representation of each mapped image which will later be used for localisation. Localisation is implemented in a way similar to factored sampling on a particle filter with a motion model estimating a vicinity of images (locations) where the platform believes it is at (Fig. 7E). The massively parallel nature of the PPA allows for storing of hundreds of image locations and the location representation has been tested with both small images which have been shown to be robust to changes in the wild (91,92) or binary descriptors. This on-sensor mapping and localisation is capable of running at >500fps without further optimisation.

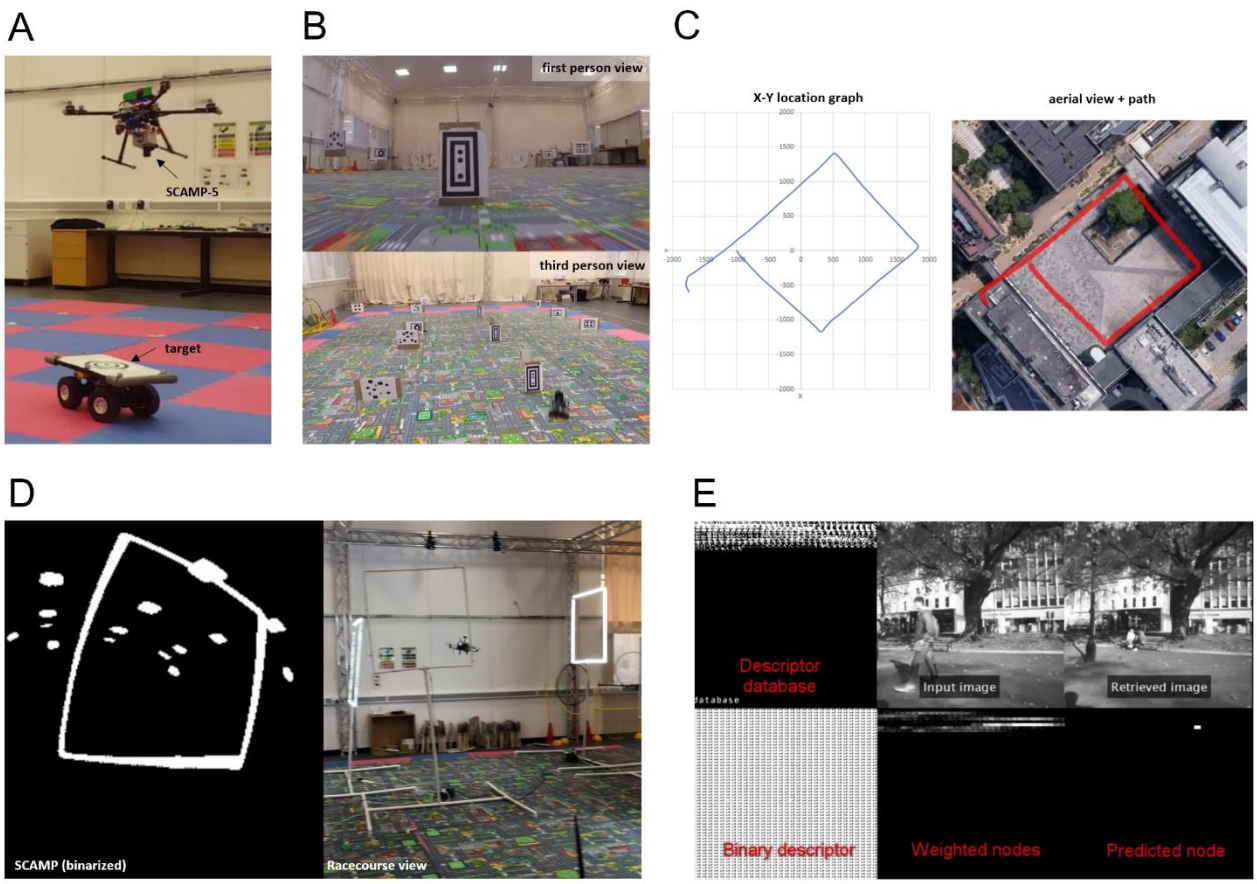

Figure 7. Robotic applications with PPAs. **A)** Closely tracking a mobile target that follows unpredictable chaotic paths with a quadrotor (79). **B)** An agile nonholonomic vehicle makes decisions at 2,000 fps on navigation trajectories based on visual targets when traveling at 4m/s (80). **C)** Visual odometry on a PPA at 1,000 fps (84). **D)** A quadrotor with a PPA estimates gate pose to fly through on a drone race course (81). **E)** Pictorial mapping and localisation on the focal plane (90). Insert shows L-R & T-B: Pictorial map, input image, retrieved image, input image descriptor representation, weighted location nodes and predicted location node.

SUMMARY AND OUTLOOK

In this article, we have reviewed recent progress on applications of PPA devices in agile robotics. Implementing robot vision systems, especially under stringent power and weight constraints, is challenging due to the amount of computing power needed to make sense of the visual world. Conventional video cameras generate a formidable amount of raw data, resulting in vast amounts of often irrelevant information that is sent and processed through the visual pipeline. This presents an opportunity for unconventional vision sensor hardware that minimises this information flow, extracting features and other high-level information at source. PPA devices allow processing to be carried out directly within the image sensor, extracting relevant data, and outputting only high-level information, instead of image frames.

It is well understood that the speed and efficiency of a computing system is mostly limited by the time and energy cost associated with data transfers, rather than the arithmetic or logic operations involved in computations. In a PPA, pixel data is processed right next to where it is acquired, eliminating communication bottlenecks between the sensor and the processor. Furthermore, processing and memory are co-located. The pixel-parallel fine-grained processor array architecture is a form of in-memory computing, as every processing element contains both execution units and local memory, storing pixel data and the intermediate results of computations. The massively-parallel processor array, intimately integrated within the image sensor fabric, provides high computational power, while offering dramatic reductions in the size, weight and

power consumption of the overall system. The pixel-level processors are fully general-purpose, software-programmable entities, capable of carrying out fundamental image processing tasks, such as filtering, extracting features and keypoints, calculating optic flow and providing other higher-level information that can be processed at substantial frame-rates. Bespoke sensing strategies, for specific applications, can be created simply by re-programming the system.

Performing vision computations directly on the focal plane minimises the latency and maximises the throughput of the vision system. The work to date has demonstrated potential advantages of this technology in enabling a variety of agile robotic tasks and systems. To bring these devices into widespread use, research is still needed in hardware, software, and applications.

Hardware developments are expected in system architecture, circuit design and implementation. Although we focussed here on processor-per-pixel architectures, alternative designs have been proposed to include multiple pixels per processor, and hierarchies of processor arrays. Physical separation of sensing and processing arrays, still on the same chip, or in the same package, offers some advantages of on-sensor processing while simplifying pixel design, allowing for smaller pixels, thus greater image resolutions and smaller lenses. It remains to be seen, which of the architectural alternatives will prove most successful in practice. With the ability to integrate more transistors inside pixels, we expect the elegance and simplicity of a processor-per-pixel arrangement, together with its highest degree of parallelism, to be of continuing interest.

Current generation PPA hardware systems are still academic prototypes, often built on older silicon technologies. Despite this, demonstrated computational performance as well as energy efficiency, are competitive with the most advanced silicon devices available today. Using more aggressively scaled state-of-the-art silicon fabrication technologies will enable substantial improvements in speed and efficiency. Wafer stacking is of particular promise. Nowadays, the typically used 2-tier stack uses a CMOS image sensor (CIS) technology that optimises photodetector performance in the top tier, and a digital CMOS technology that optimises logic density and speed in the bottom tier. This has already proven advantageous for implementing complex pixel designs (5,6,8). It can be envisaged that further advances in 3D integration, including fully stacked multi-tier thru-silicon-via (TSV) technology (93) will bring about the possibility of optimising the in-pixel circuitry for both analog and digital performance, as well as considerably expanding the capabilities of in-pixel processors, with individual pixels potentially spanning multiple vertically-integrated silicon wafers. As the designs move from research prototypes to commercial developments, we can expect to see increased integration of a full system-on-a-chip, including the pixel-processor arrays, controllers, application processors and memory.

The key to unlocking the full potential of new hardware is the availability of software tools. Present day PPA devices have idiosyncratic instruction sets; at the level of the processor array, code is still manually programmed in assembler (embedded as inline code in a C++ framework). Pioneering work on PPA code compilation (94-96) and automated code generation for domain-specific problems (97) points the way towards future work on the software toolchain. With the limited availability of hardware prototypes, the software development environment based on PC-based emulation of PPA devices might provide the research community with the tools necessary to evaluate new technologies. To that end, we have provided free access to our SCAMP-5 development and simulation framework (77), as well as interfaces allowing integration of the PPA simulator with a virtual robot simulator (98). The continuing refinement of PPA technologies, both in terms of hardware and software, and their increasing availability, will be essential to establish these firmly in the landscape of computing hardware for robotic systems.

Further work is needed on algorithms that make best use of the unconventional hardware technology. The vast majority of traditional vision algorithms stem from methods developed for serially processed, relatively low framerate, single images. With a PPA, approaches that are not optimal in conventional systems (e.g. depth from focus (67), or high dynamic range (66) that rely on processing thousands of frames per second) are feasible. At the same time, constraints not present in conventional systems (limited amount of local memory, nearest-neighbour connectivity patterns) provide a challenge. This opens up the scope for developing efficient PPA-based solutions, from basic operations e.g. image rotations (81) or global

summation (99), to strategies such as pooling of processing element resources (100, 101). The ability to shift a large proportion of the visual pipeline to the pixel array, provides the opportunity to consider a greater scope for pixel-level computations (102). It is likely that complete visual perception systems might be achievable, where the majority of computation is carried out on the sensor level.

The key to the PPA approach is the information extraction and data compression at pixel level, which results in efficient and high speed throughput of information from the sensor device to the rest of the system. The kind of sparse data that needs to be generated by the sensor, needs to be optimised for each application. A promising solution to this might be through optimising the entire pipeline, from sensing strategies through to host-level processing. PPA devices allow us not only to process the image data, but to also affect the light sensing itself (e.g. at a level of programmable exposure for each pixel). The work on "neural sensors" (103), where deep learning techniques were used to optimise the image acquisition of high-speed imaging and high-dynamic range imaging tasks, signposts this direction. The approach can be extended to other tasks, and of particular interest would be the end-to-end optimisation of the entire visual perception-action task, from sensing through to motor control. This remains an open problem, and PPA devices provide flexibility that can help to address this issue at the sensor level.

In terms of practical applications, possible today, we have demonstrated applications in agile robotics such as target tracking, odometry, obstacle avoidance and localisation. These could be extended to provide a front-end to a full high-speed simultaneous localisation and mapping (SLAM) system, and more complex navigation scenarios. Another aspect is human-machine interactions, with potential applications in low-latency gaze tracking and gesture recognition. The use of PPA devices extends beyond robotic systems, to more general computer vision applications, and along with the refinement of the microelectronic chip integration and packaging technologies, continued progress in this field is to be expected. PPAs can also find an opportunity in the development of better privacy-aware systems that put judicious attention on the data that is transferred out of them. We thus predict that PPA based vision sensors, capable of producing highly-informative, sparse data, at high temporal resolution, low latency and low power, will increasingly find application in a variety of future robotic systems.

References and Notes

1. Eki, Ryoji, Satoshi Yamada, Hiroyuki Ozawa, Hitoshi Kai, Kazuyuki Okuike, Hareesh Gowtham, Hidetomo Nakanishi et al. "9.6 A 1/2.3 inch 12.3 Mpixel with On-Chip 4.97 TOPS/W CNN Processor Back-Illuminated Stacked CMOS Image Sensor." In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, pp. 154-156. IEEE, 2021.
2. Fossum, Eric R., and Donald B. Hondongwa. "A Review of the Pinned Photodiode for CCD and CMOS Image Sensors." *IEEE Journal of the Electron Devices Society* 2, no. 3 (2014): 33-43.
3. Y. Oike, "Evolution of Image Sensor Architectures With Stacked Device Technologies," in *IEEE Transactions on Electron Devices*, doi: 10.1109/TED.2021.3097983.
4. Park, Geunsook, Alan Chih-Wei Hsuing, Keiji Mabuchi, Jingming Yao, Zhiqiang Lin, Vincent C. Venezia, Tongtong Yu, Yu-Shen Yang, Tiejun Dai, and Lindsay A. Grant. "A 2.2 μm stacked back side illuminated voltage domain global shutter CMOS image sensor." In *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 16-4. IEEE, 2019.
5. Sakakibara, Masaki, Koji Ogawa, Shin Sakai, Yasuhisa Tochigi, Katsumi Honda, Hidekazu Kikuchi, Takuya Wada et al. "A 6.9-μm Pixel-Pitch Back-Illuminated Global Shutter CMOS Image Sensor With Pixel-Parallel 14-Bit Subthreshold ADC." *IEEE Journal of Solid-State Circuits* 53, no. 11 (2018): 3017-3025.
6. Liu, Chiao, Lyle Bainbridge, Andrew Berkovich, Song Chen, Wei Gao, Tsung-Hsun Tsai, Kazuya Mori et al. "A 4.6 μm, 512× 512, Ultra-Low Power Stacked Digital Pixel Sensor with Triple Quantization and 127dB Dynamic Range." In *2020 IEEE International Electron Devices Meeting (IEDM)*, pp. 16-1. IEEE, 2020.
7. Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbruck. "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change." In *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*, pp. 2060-2069. IEEE, 2006.

8. Finateu, Thomas, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu et al. "5.10 A 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 GEPS readout, programmable event-rate controller and compressive data-formatting pipeline." In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 112-114. IEEE, 2020.
9. Conradt, Jörg, Matthew Cook, Raphael Berner, Patrick Lichtsteiner, Rodney J. Douglas, and Tobi Delbruck. "A pencil balancing robot using a pair of AER dynamic vision sensors." In *2009 IEEE International Symposium on Circuits and Systems*, pp. 781-784. IEEE, 2009.
10. Delbruck, Tobi. Lang, M. Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor, *Frontiers in Neuroscience*, Volume 7, 2013, DOI 10.3389/fnins.2013.00223
11. Mueggler, Elias, Basil Huber, and Davide Scaramuzza. "Event-based, 6-DOF pose tracking for high-speed maneuvers." In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2761-2768. IEEE, 2014.
12. Vidal, Antoni Rosinol, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios." *IEEE Robotics and Automation Letters* 3, no. 2 (2018): 994-1001.
13. Pijnacker Hordijk, Bas J., Kirk YW Scheper, and Guido CHE De Croon. "Vertical landing for micro air vehicles using event-based optical flow." *Journal of Field Robotics* 35, no. 1 (2018): 69-90.
14. Mead, Carver A., and Misha A. Mahowald. "A silicon model of early visual processing." *Neural networks* 1, no. 1 (1988): 91-97.
15. Gollisch, Tim, and Markus Meister. "Eye smarter than scientists believed: neural computations in circuits of the retina." Neuron 65, no. 2 (2010): 150-164.
16. Zaghloul, Kareem A., and Kwabena Boahen. "A silicon retina that reproduces signals in the optic nerve." *Journal of neural engineering* 3, no. 4 (2006): 257.
17. Culurciello, Eugenio, Ralph Etienne-Cummings, and Kwabena A. Boahen. "A biomorphic digital image sensor." *IEEE journal of solid-state circuits* 38, no. 2 (2003): 281-294.
18. Leñero-Bardallo, Juan A., Philipp Häfliger, Ricardo Carmona-Galán, and Ángel Rodríguez-Vázquez. "A bio-inspired vision sensor with dual operation and readout modes." *IEEE Sensors Journal* 16, no. 2 (2015): 317-330.
19. Zhao, Bo, Xiangyu Zhang, and Shoushun Chen. "A CMOS image sensor with on-chip motion detection and object localization." In *2011 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4. IEEE, 2011.
20. Y. M. Chi, U. Mallik, M. A. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "CMOS camera with in-pixel temporal change detection and ADC," IEEE J. Solid-State Circuits, vol. 42, no. 10, pp. 2187–2196, Oct. 2007
21. Lei, Ming-Han, and Tzi-Dar Chiueh. "An analog motion field detection chip for image segmentation." *IEEE transactions on circuits and systems for video technology* 12, no. 5 (2002): 299-308.
22. Stocker, Alan A. "Analog integrated 2-D optical flow sensor." *Analog Integrated Circuits and Signal Processing* 46, no. 2 (2006): 121-138.
23. Suárez, Manuel, Victor Manuel Brea, Jorge Fernandez-Berni, Ricardo Carmona-Galan, Diego Cabello, and Angel Rodriguez-Vazquez. "Low-power CMOS vision sensor for Gaussian pyramid extraction." *IEEE Journal of Solid-State Circuits* 52, no. 2 (2016): 483-495.
24. N. Cottini, M. Gottardi, N. Massari, R. Passerone, and Z. Smilansky, "A 33 μW 64×64 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, Mar. 2013.
25. García-Lesta, Daniel, Paula López, Victor M. Brea, and Diego Cabello. "A CMOS Vision Sensor for Background Subtraction." In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5. IEEE, 2020.
26. D. Ginhac, J. Dubois, B. Heyrman, and M. Paindavoine, "A high speed programmable focal-plane SIMD vision chip," Anal. Integr. Circuits Signal Process., vol. 65, no. 3, pp. 389–398, Dec. 2010
27. Jendernalik, Waldemar, Grzegorz Blakiewicz, Jacek Jakusz, Stanislaw Szczepanski, and Robert Piotrowski. "An analog sub-miliwatt CMOS image sensor with pixel-level convolution processing." *IEEE Transactions on Circuits and Systems I: Regular Papers* 60, no. 2 (2013): 279-289.
28. Lefebvre, Martin, Ludovic Moreau, Rémi Dekimpe, and David Bol. "A 0.2-to-3.6 TOPS/W Programmable Convolutional Imager SoC with In-Sensor Current-Domain Ternary-Weighted MAC

Operations for Feature Extraction and Region-of-Interest Detection." In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, pp. 118-120. IEEE, 2021.
29. Zhong, Xiaopeng, Qian Yu, Amine Bermak, Chi-Ying Tsui, and May-Kay Law. "A 2pJ/pixel/direction MIMO processing based CMOS image sensor for omnidirectional local binary pattern extraction and edge detection." In *2018 IEEE Symposium on VLSI Circuits*, pp. 247-248. IEEE, 2018.
30. Berkovich, Andrew, Michela Lecca, Leonardo Gasparini, Pamela A. Abshire, and Massimo Gottardi. "A 30 µw 30 fps 110× 110 pixels vision sensor embedding local binary patterns." *IEEE Journal of Solid-State Circuits* 50, no. 9 (2015): 2138-2148.
31. Lopich, Alexey, and Piotr Dudek. "A SIMD cellular processor array vision chip with asynchronous processing capabilities." *IEEE Transactions on Circuits and Systems I: Regular Papers* 58, no. 10 (2011): 2420-2431.
32. Dudek, Piotr, and Peter J. Hicks. "A general-purpose processor-per-pixel analog SIMD vision chip." *IEEE Transactions on Circuits and Systems I: Regular Papers* 52, no. 1 (2005): 13-20.
33. Dudek, P., and S. J. Carey. "General-purpose 128x128 SIMD processor array with integrated image sensor." *Electronics Letters* 42, no. 12 (2006): 678-679.
34. G. H. Barnes, R. M. Brown, M. Kato, D. J. Kuck, D. L. Slotnick and R. A. Stokes, "The ILLIAC IV Computer," in *IEEE Transactions on Computers*, vol. C-17, no. 8, pp. 746-757, Aug. 1968, doi: 10.1109/TC.1968.229158.
35. Fountain, Terry J., K. N. Matthews, and Michael J. B. Duff. "The CLIP7A image processor." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, no. 3 (1988): 310-319.
36. Heaton, R. A., and Donald W. Blevins. "BLITZEN: A VLSI array processing chip." In *1989 Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 12-1. IEEE, 1989.
37. Espasa, Roger, Mateo Valero, and James E. Smith. "Vector architectures: past, present and future." In *Proceedings of the 12th international conference on Supercomputing*, pp. 425-432. 1998.
38. Peleg, Alex, and Uri Weiser. "MMX technology extension to the Intel architecture." *IEEE micro* 16, no. 4 (1996): 42-50.
39. Bernard, Thierry M., Bertrand Y. Zavidovique, and Francis J. Devos. "A programmable artificial retina." *IEEE Journal of Solid-State Circuits* 28, no. 7 (1993): 789-798.
40. Paillet, Fabrice, Damien S. Mercier, and Thierry M. Bernard. "Making the most of 15k-lambda-2 silicon area for a digital retina PE." In *Advanced Focal Plane Arrays and Electronic Cameras II*, vol. 3410, pp. 158-167. International Society for Optics and Photonics, 1998.
41. Ishikawa, Masatoshi, Kazuya Ogawa, Takashi Komuro, and Idaku Ishii. "A CMOS vision chip with SIMD processing element array for 1 ms image processing." In *1999 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. First Edition (Cat. No. 99CH36278)*, pp. 206-207. IEEE, 1999.
42. Eklund, J-E., Christer Svensson, and Anders Astrom. "VLSI implementation of a focal plane image processor-a realization of the near-sensor image processing concept." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 4, no. 3 (1996): 322-335.
43. Lopich, Alexey, and Piotr Dudek. "A general-purpose vision processor with 160× 80 pixel-parallel SIMD processor array." In *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, pp. 1-4. IEEE, 2013.
44. Mead, Carver, and Mohammed Ismail, eds. *Analog VLSI implementation of neural systems*. Vol. 80. Springer Science & Business Media, 2012.
45. Small, James S. "General-purpose electronic analog computing: 1945-1965." *IEEE Annals of the History of Computing* 15, no. 2 (1993): 8-18.
46. S. Masuda, S. Yoneda and T.Kasai, "Sampled-data charge processor." *International journal of electronics* 58, no. 5 (1985): 743-760.
47. Dudek, Piotr, and Peter J. Hicks. "A CMOS general-purpose sampled-data analog processing element." *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 47, no. 5 (2000): 467-473.
48. Carey, Stephen J., David RW Barr, Bin Wang, Alexey Lopich, and Piotr Dudek. "Mixed signal SIMD processor array vision chip for real-time image processing." *Analog Integrated Circuits and Signal Processing* 77, no. 3 (2013): 385-399.

49. Carey, Stephen J., Alexey Lopich, David RW Barr, Bin Wang, and Piotr Dudek. "A 100,000 fps vision sensor with embedded 535 GOPS/W 256×256 SIMD processor array." In *2013 Symposium on VLSI Circuits*, pp. C182-C183. IEEE, 2013.
50. Cembrano, Gustavo Liñan, Ángel Rodríguez-Vázquez, Servando Espejo Meana, and Rafael Domínguez-Castro. "ACE16k: a 128× 128 focal plane analog processor with digital I/O." *International journal of neural systems* 13, no. 06 (2003): 427-434.
51. Poikonen, Jonne, Mika Laiho, and Ari Paasio. "MIPA4k: A 64× 64 cell mixed-mode image processor array." In *2009 IEEE International Symposium on Circuits and Systems*, pp. 1927-1930. IEEE, 2009.
52. Di Federico, Martin, Pedro Julián, and Pablo S. Mandolesi. "SCDVP: A simplicial CNN digital visual processor." *IEEE Transactions on Circuits and Systems I: Regular Papers* 61, no. 7 (2014): 1962-1969.
53. Massari, Nicola, and Massimo Gottardi. "A 100 dB dynamic-range CMOS vision sensor with programmable image processing and global feature extraction." *IEEE journal of solid-state circuits* 42, no. 3 (2007): 647-657.
54. Dupret, Antoine, Jacques-Olivier Klein, and Abdallah Nshare. "A DSP-like analogue processing unit for smart image sensors." *International Journal of Circuit Theory and Applications* 30, no. 6 (2002): 595-609.
55. Yamazaki, Tomohiro, Hironobu Katayama, Shuji Uehara, Atsushi Nose, Masatsugu Kobayashi, Sayaka Shida, Masaki Odahara et al. "4.9 A 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing." In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 82-83. IEEE, 2017.
56. Lindgren, Leif, Johan Melander, Robert Johansson, and B. Moller. "A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging." *IEEE Journal of Solid-State Circuits* 40, no. 6 (2005): 1350-1359.
57. Doege, Jens, Christoph Hoppe, Peter Reichel, and Nico Peter. "A 1 megapixel HDR image sensor SoC with highly parallel mixed-signal processing." In *Int. Image Sensor Workshop (IISW '15)*. 2015.
58. Yamashita, Hirofumi, and Charles G. Sodini. "A CMOS imager with a programmable bit-serial column-parallel SIMD/MIMD processor." *IEEE transactions on electron devices* 56, no. 11 (2009): 2534-2545.
59. Schmitz, Joseph A., Mahir K. Gharzai, Sina Balkır, Michael W. Hoffman, Daniel J. White, and Nathan Schemm. "A 1000 frames/s vision chip using scalable pixel-neighborhood-level parallel processing." *IEEE Journal of Solid-State Circuits* 52, no. 2 (2016): 556-568.
60. Millet, Laurent, Stephane Chevobbe, Caaliph Andriamisaina, Lamine Benaissa, Edouard Deschaseaux, Edith Beigne, Karim Ben Chehida et al. "A 5500-frames/s 85-GOPS/W 3-D stacked bsi vision chip based on parallel in-focal-plane acquisition and processing." IEEE Journal of Solid-State Circuits 54, no. 4 (2019): 1096-1105.
61. Verdant, Arnaud, Antoine Dupret, Patrick Villard, Laurent Alacoque, Hervé Mathias, and Flavien Delgehier. "A 120µW 240× 110@ 25fps vision chip with ROI detection SIMD processing unit." In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2412-2415. IEEE, 2013.
62. Zhang, Wancheng, Qiuyu Fu, and Nan-Jian Wu. "A programmable vision chip based on multiple levels of parallel processors." *IEEE Journal of Solid-State Circuits* 46, no. 9 (2011): 2132-2147.
63. Carmona-Galán, Ricardo, Ákos Zarándy, Csaba Rekeczky, Péter Földesy, Alberto Rodríguez-Pérez, Carlos Domínguez-Matas, Jorge Fernández-Berni et al. "A hierarchical vision processing architecture oriented to 3d integration of smart camera chips." *Journal of Systems Architecture* 59, no. 10 (2013): 908-919.
64. Dudek, Piotr, Alexey Lopich, and Viktor Gruev. "A pixel-parallel cellular processor array in a stacked three-layer 3D silicon-on-insulator technology." In *2009 European Conference on Circuit Theory and Design*, pp. 193-196. IEEE, 2009.
65. Carey, Stephen J., David RW Barr, and Piotr Dudek. "Low power high-performance smart camera system based on SCAMP vision sensor." *Journal of Systems Architecture* 59, no. 10 (2013): 889-899.
66. Martel, Julien NP, Lorenz K. Müller, Stephen J. Carey, and Piotr Dudek. "Parallel HDR tone mapping and auto-focus on a cellular processor array vision chip." In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1430-1433. IEEE, 2016.
67. Martel, Julien NP, Lorenz K. Müller, Stephen J. Carey, Jonathan Müller, Yulia Sandamirskaya, and Piotr Dudek. "Real-time depth from focus on a programmable focal plane processor." *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, no. 3 (2017): 925-934.

68. Chen, Jianing, Stephen J. Carey, and Piotr Dudek. "Feature extraction using a portable vision system." In *IEEE/RSJ Int. Conf. Intell. Robots Syst., Workshop Vis.-based Agile Auton. Navigation UAVs*. 2017.
69. Martel, Julien NP, and Yulia Sandamirskaya. "A neuromorphic approach for tracking using dynamic neural fields on a programmable vision-chip." In *Proceedings of the 10th International Conference on Distributed Smart Camera*, pp. 148-154. 2016.
70. Alonso-Montes, Carmen, David López Vilariño, Piotr Dudek, and Manuel G. Penedo. "Fast retinal vessel tree extraction: A pixel parallel approach." *International Journal of Circuit Theory and Applications* 36, no. 5-6 (2008): 641-651.
71. Bose, Laurie, Piotr Dudek, Jianing Chen, Stephen J. Carey, and Walterio W. Mayol-Cuevas. "Fully embedding fast convolutional networks on pixel processor arrays." In *European Conference on Computer Vision*, pp. 488-503. Springer, Cham, 2020.
72. Liu, Yanan, Jianing Chen, Laurie Bose, Piotr Dudek, and Walterio Mayol-Cuevas. "Direct Servo Control from In-Sensor CNN Inference with A Pixel Processor Array." *arXiv preprint arXiv:2106.07561* (2021).
73. Duhamel, Pierre-Emile, Judson Porter, Benjamin Finio, Geoffrey Barrows, David Brooks, Gu-Yeon Wei, and Robert Wood. "Hardware in the loop for optical flow sensing in a robotic bee." In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1099-1106. IEEE, 2011.
74. Briod, Adrien, Jean-Christophe Zufferey, and Dario Floreano. "Optic-flow based control of a 46g quadrotor." In *Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments, IROS 2013*
75. Ishikawa, Masatoshi, Akio Namiki, Taku Senoo, and Yuji Yamakawa. "Ultra high-speed robot based on 1 kHz vision system." In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5460-5461. IEEE, 2012.
76. Arena, Paolo, Luigi Fortuna, Mattia Frasca, Guido Vagliasindi, and Adriano Basile. "CNN wave based computation for robot navigation on ACE16K." In *2005 IEEE International Symposium on Circuits and Systems*, pp. 5818-5821. IEEE, 2005.
77. Chen, Jianing, Stephen J. Carey, and Piotr Dudek. "Scamp5d vision system and development framework." In *Proceedings of the 12th International Conference on Distributed Smart Cameras*, pp. 1-2. 2018.
78. Project Agile. www.project-agile.org. Retrieved July 2021.
79. C. Greatwood, L. Bose, T. Richardson, W. Mayol-Cuevas, J. Chen, S.J. Carey and P. Dudek, "Tracking control of a UAV with a parallel visual processor", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017.
80. Liu Y, Bose L, Greatwood C, hen J, Fan R, Richardson T, Carey SJ, Dudek P, Mayol-Cuevas W. Agile reactive navigation for a non-holonomic mobile robot using a pixel processor array. IET Image Processing. 1–10. 2021.
81. C. Greatwood, L. Bose, T. Richardson, W. Mayol-Cuevas, R. Clarke, J. Chen, S. J. Carey, and P. Dudek, "Towards drone racing with a pixel processor array," in 11th international micro air vehicle competition and conference (IMAV), Madrid, Spain, p. 76–82. 2019.
82. J.Chen, Y. Liu, S.J.Carey and P.Dudek, "Proximity Estimation Using Vision Features Computed on Sensor", Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2020, June 2020.
83. D. Nister, O. Naroditsky and J. Bergen, "Visual odometry," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004, pp. I-I, doi: 10.1109/CVPR.2004.1315094.
84. L. Bose, J. Chen, S. J. Carey, P. Dudek and W. Mayol-Cuevas. "Visual Odometry for Pixel Processor Arrays", International Conference on Computer Vision (ICCV), Venice, Italy, 2017.
85. Murai, Riku, Sajad Saeedi, and Paul HJ Kelly. "BIT-VO: Visual Odometry at 300 FPS using Binary Features from the Focal Plane." In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8579-8586. IEEE, 2020.
86. C. Greatwood, L. Bose, T. Richardson, W. Mayol-Cuevas, J. Chen, S.J. Carey and P. Dudek. "Perspective Correcting Visual Odometry for Agile MAVs using a Pixel Processor Array", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.
87. A. McConville, L. Bose, R. Clarke, W. Mayol-Cuevas, J. Chen, C. Greatwood, S. Carey, P. Dudek, T. Richardson. Visual Odometry Using Pixel Processor Arrays for Unmanned Aerial Systems in GPS Denied Environments. Frontiers Robotics and AI. Sept, 2020.

88. M. Cummins, P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. The International Journal of Robotics Research. 27(6):647-665. 2008.
89. M. J. Milford, G. F. Wyeth and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. pp. 403-408 Vol.1. 2004.
90. H. Castillo-Elizalde, Y. Liu, L. Bose and W. Mayol-Cuevas. Weighted Node Mapping and Localisation on a Pixel Processor Array. IEEE ICRA, 2021.
91. H. Aoki, B. Schiele, and A. Pentland, "Realtime personal positioning system for wearable computers," in 2012 16th International Symposium on Wearable Computers. IEEE Computer Society, Oct 1999.
92. M. Milford, "Visual route recognition with a handful of bits," Proc. 2012 Robotics: Science and Systems VIII, pp. 297–304, 2012.
93. Shen, Wen-Wei, and Kuan-Neng Chen. "Three-dimensional integrated circuit (3D IC) key technology: through-silicon via (TSV)." *Nanoscale research letters* 12, no. 1 (2017): 1-9.
94. Martel, Julien NP, and Piotr Dudek. "Vision chips with in-pixel processors for high-performance low-power embedded vision systems." In *Workshop on Architectures and Systems for Real-time Mobile Vision Applications (ASR-MOV), International Symposium on Code Generation and Optimization, CGO'16*. 2016.
95. Wang, Bin, and Piotr Dudek. "Coarse grain mapping method for image processing on fine grain cellular processor arrays." In *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1-6. IEEE, 2012.
96. T. Komuro, S. Kagami, M. Ishikawa, and Y. Katayama, "Development of a bit-level compiler for massively parallel vision chips," in Proc. IEEE 7th Int. Workshop on Computer Architecture for Machine Perception (CAMP'05), Jul. 2005, pp. 204–209
97. Debrunner, Thomas, Sajad Saeedi, and Paul HJ Kelly. "Auke: Automatic kernel code generation for an analogue simd focal-plane sensor-processor array." *ACM Transactions on Architecture and Code Optimization (TACO)* 15, no. 4 (2019): 1-26.
98. Y. Liu, J. Chen, L. Bose, P. Dudek, W. Mayol-Cuevas. Bringing A Robot Simulator to the SCAMP Vision System. Arxiv, 2021.
99. L. Bose, P.Dudek,J.Chen and S.Carey, "Sand Castle Summation For Pixel Processor Arrays", Internations Workshop on Cellular Processor Arrays and Applications, CNNA 2021 (accepted)
100. Martel, Julien NP, Miguel Chau, Piotr Dudek, and Matthew Cook. "Toward joint approximate inference of visual quantities on cellular processor arrays." In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2061-2064. IEEE, 2015.
101. Bose, Laurie, Jianing Chen, Stephen J. Carey, Piotr Dudek, and Walterio Mayol-Cuevas. "A camera that CNNs: Towards embedded neural networks on pixel processor arrays." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1335-1344. 2019.
102. Davison, Andrew J. "FutureMapping: The computational structure of spatial AI systems." *arXiv preprint arXiv:1803.11288* (2018).
103. Martel, Julien NP, Lorenz K. Mueller, Stephen J. Carey, Piotr Dudek, and Gordon Wetzstein. "Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors." *IEEE transactions on pattern analysis and machine intelligence* 42, no. 7 (2020): 1642-1653.
104. W. S. Boyle and G. E. Smith, "Charge coupled semiconductor devices," in The Bell System Technical Journal, vol. 49, no. 4, pp. 587-593, April 1970, doi: 10.1002/j.1538-7305.1970.tb01790.x.

Funding: This work was supported by Engineering and Physical Sciences Research Council, grant EP/M019284/1 (PD) and Engineering and Physical Sciences Research Council, grant EP/M019454/1 (WM, TR); **Author contributions:** Conceptualization: PD, WM, TR. Investigation: PD, WM, TR, SC, JC, LB, YL, CG. Writing – original draft: PD, WM. Writing – review & editing: TR, SC, LB, JC. **Competing interests:** PD is also affiliated with Pixelcore. TR is also affiliated with Perceptual Robotics. LB, SC and JC are currently employed by Pixelcore. CG is currently employed by Perceptual Robotics. WM is also affiliated with Amazon. YL declares no competing interests.

Submitted 5 August 2021, Accepted 7 June 2022, Published 29 June 2022