

Chapter 1

In-Sensor Visual Devices for Perception and Inference



Yanan Liu, Hepeng Ni, Chao Yuwen, Xinyu Yang, Yuhang Ming, Huixin Zhong, Yao Lu, and Liang Ran

Abstract The traditional machine vision systems use separate architectures for perception, memory, and processing. This approach may hinder the growing demand for high image processing rates and low power consumption. On the other hand, in-sensor computing performs signal processing at the pixel level, directly utilizing collected analogue signals without sending them to other processors. This means that in-sensor computing may offer a solution for achieving highly efficient and low-power consumption visual signal processing. This can be achieved by integrating

Y. Liu

School of Microelectronics, Shanghai University, Shanghai, China

e-mail: yanan.liu@ieee.org

H. Ni (✉)

School of Mechanical and Electrical Engineering, Shandong Jianzhu University, Jinan, China

e-mail: nihepeng20@sdjzu.edu.cn

C. Yuwen (✉)

GAC R&D Center, Guangzhou, China

e-mail: yuwenchao@gacrnd.com

X. Yang

Lancaster University, Bailrigg, Lancaster, UK

e-mail: xinyu.yang@ieee.org

Y. Ming

School of Computer Science, Hangzhou Dianzi University, Hangzhou, China

e-mail: yuhang.ming@ieee.org

H. Zhong

Department of Computer Science, University of Bath, Bath, UK

e-mail: hz877@bath.ac.uk

Y. Lu

School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China

e-mail: sunnylyyao@vip.qq.com

L. Ran

Pixelcore.ai, Beijing, China

e-mail: light.ran@pixelcore.ai

sensing, storage, and computation onto focal planes with novel circuit designs or new materials. This chapter aims to describe the proposed image processing algorithms and neural networks of in-sensor computing, as well as their applications in machine vision and robotics. The goal of this chapter is to help developers, researchers, and users of unconventional visual sensors understand their functioning and applications, especially in the context of autonomous driving.

1.1 Introduction

The importance of vision as a means of perception cannot be overstated, as it enables efficient collection and interpretation of information [1]. To apply this capability to fields like machine vision, robotics, Internet of Things (IoT), and artificial intelligence (AI), there is a pressing need to develop visual information processing methods and technologies that operate at ultra-high speeds while consuming minimal energy [2, 3]. The conventional machine vision systems and their associated technologies face major constraints in terms of system latency, power consumption, and privacy issues [2, 3]. Unlike the mammalian retina mechanism that rapidly processes raw signals through several layers of cells, the visual signal digitization, storage, and transmission processes involved in conventional machine vision systems can introduce significant time latency, which hinders quick responses to dynamic changes and results in inefficiencies due to irrelevant data processing [2, 3]. Additionally, external image processors like CPU/GPU/VPU/DSPs consume high amounts of power, which is unfavorable for portable tasks [2, 3]. The overwhelming amount of data generated by ubiquitous sensors may obscure the useful information, thus necessitating the extraction of critical information by terminal sensors to reduce data movement from the sensing chip to processing units [4, 5]. Moreover, privacy-sensitive scenarios may require the extraction of crucial information from raw analog signals rather than collected images.

To address these challenges, a paradigm shift towards in-sensor computing is proposed [6]. This approach is inspired by the mammalian retina (Fig. 1.1a) and involves the vision sensor not only acquiring visual information but also processing it to produce highly compressed information instead of video frames (Fig. 1.1c). In-sensor computing offers image-free visual signal processing, which ensures data confidentiality. This interdisciplinary field encompasses various technologies, including sensors, analogue signal processing, near-sensor computing, and in-memory computing (Fig. 1.3). In-sensor computing devices are sensors that integrate perception, temporary storage, and data processing and analysis with raw analogue signals within the sensing chip. While near-sensor computing can reduce the physical distance between sensing and computing, data movement from sensors to processors is still necessary. In-memory computing uses memristors for both memory and computing [7], utilizing tunable resistance as the synaptic weights.

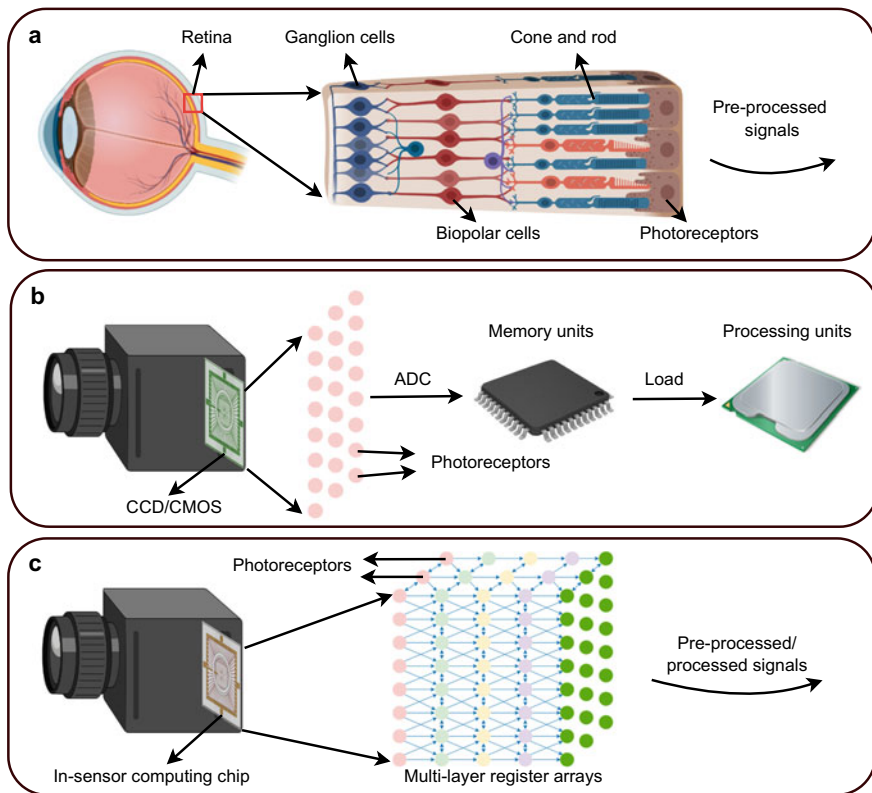


Fig. 1.1 The origin of in-sensor visual computing. **a** The concept of in-sensor computing is bio-inspired by the retina mechanism where visual signals can be generated and pre-processed by different types of cells [8]. **b** Conventional machine vision system: light density needs to be read out first and converted to digital data which being loaded into memory and then processing units for meaningful information extraction. **c** Visual data can be generated, stored, and processed in sensor through the bio-inspired hardware design

This chapter firstly illustrates the common in-sensor visual computing hardware architecture. Then various emerging in-sensor computing visual sensors are introduced in terms of hardware, software, algorithms, and applications within the category of in-sensor computing architecture. Finally, a summary and future prospective of in-sensor visual computing technology are made in the conclusion.

1.2 In-Sensor Computing Devices

1.2.1 Architecture

In recent years, progress has been made in the development of in-sensor computing devices. By far, there are mainly two types of in-sensor computing architectures:

(1) In-Sensor architecture by integrating sensing, memory, and computing units:

A Focal-Plane Sensor Processor (FPSP) [9] integrates visual sensing, storage and computing units on the focal plane under the architecture of cellular neural networks (Fig. 1.2a). As for each Processing Element (PE), the generated analogue signals from the pixel can be transferred to the temporal memory units through the bus and processed using ALU units and registers. Each PE plays a role as a cell interacting with its neighbours for signal exchange and processing. Hence, the in-sensor visual inference is realised by the hardware cellular neural network and its synaptic weights in memory. The representative devices under the FPFS architecture mainly include the SCAMP Pixel Processor Array (PPA), Q-Eye [10] MIPA4k [9], Asynchronous-Synchronous Focal-Plane Sensor-Processor Chip (ASAP) [9], KOVA1 [11], and Aistorm Mantis2 [12], where the SCAMP PPA is comparatively mature with continuous research and application outputs.

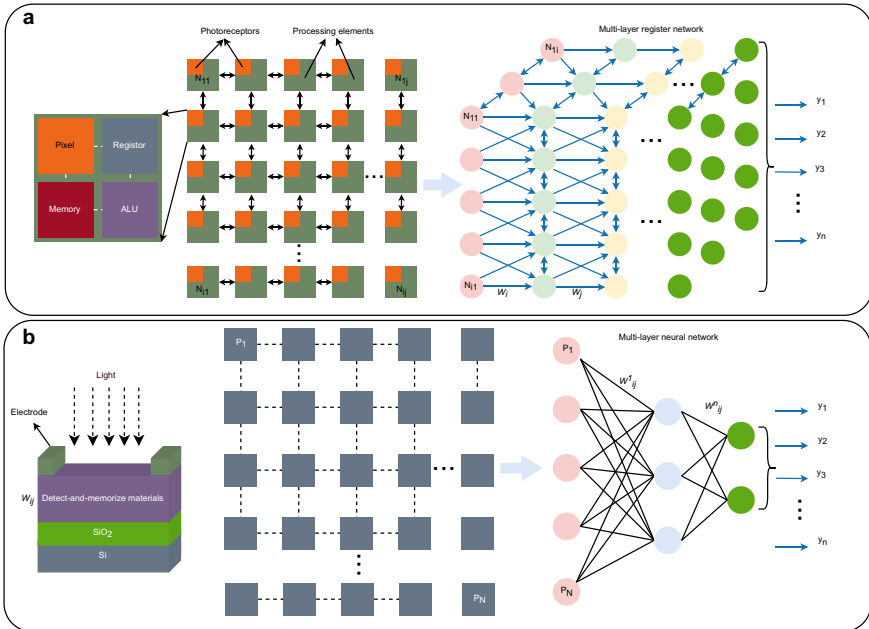
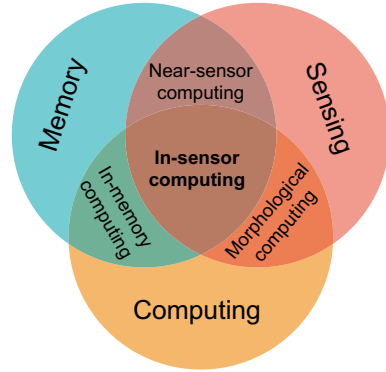


Fig. 1.2 In-sensor perception and computing architectures and their associated artificial networks. a An in-sensor cellular network can be built with an array of PEs which integrates sensing, memory, and computing units. **b** A neural network with detect-and-memorisate materials

Table 1.1 In-sensor computing architecture comparison between FPFS and DAM scheme

Scheme	Maturity level	Speed	Application	General-purpose	Programmable	Efficiency
FPFS [15, 16]	Mature	High-speed	Many	Yes	Yes	High
DAM [5, 6, 17–19]	Immature	Ultra-fast	Few	No	Partial	Ultra-high

Fig. 1.3 The position of in-sensor computing in the existing knowledge



(2) **Detect-and-memorise materials for in-sensor computing architectures:** Material-based detect-and-memorise (DAM) devices (Fig. 1.2b) have recently been proposed to mimic the functional mechanism of the photonic synapses to implement artificial neural networks [5, 13]. Among these emerging materials and devices, the memristor is representative as it facilitates sensing, temporal memory, and computing capability when combined with other photo-sensitive devices [14]. Specifically, visual signals generated from photoreceptors such as photodiodes can be further processed within the artificial networks composed of memristors with tunable resistance as the weights.

Table 1.1 shows the difference between the two rising in-sensor computing architectures. As can be seen from Table 1.1, the DAM-based in-sensor computing sensors are new and immature compared to the scheme by sensor, memory, and computing integration. Hence, this chapter mainly reviews devices and algorithms based on the first architecture scheme (Fig. 1.3).

1.2.2 Focal-Plane Sensor Processor (FPFS)

Conventional sensors mainly play the role of information collectors. In recent years, with the development of techniques on integrated circuit design and the growing need for low-power and lower-latency edge-computing, a sensor has gradually been

Table 1.2 List of in-sensor and in-memory processing vision systems

System names	Resolution	In-sensor storage	Speed (FPS)
Aistorm Mantis2	96×96	Gray-scale image	50 K [12]
Eye-RIS	176×144	7 gray + 4 binary	10 K [22]
Photodiode array	–	–	20 M bins [14]
SCAMP PPA	256×256	7 gray + 13 binary	100 K [23]
Kovilta's KOVA1	96×96	–	> 1 K [24]
MIPA4k	64×64	–	> 1 K [25]
DVS	In-senor pre-processing	Binary events	> 10 K [26]

integrated with the ability of signal processing independent from general-purpose computers. The goal of near-sensor processing is to use a dedicated machine learning accelerator chip on the same printed circuit board [20], or perhaps 3D-stacked with the CMOS image chip [21]. Despite the fact that this allows CMOS image chip data to be processed closer to the sensor rather than in the cloud, data transport expenses between the sensing chip and the processing chip still exist. In contrast, the in-sensor computing paradigm aims to embed processing capability for each individual pixel. This section introduces classic in-sensor visual computing devices. Table 1.2 lists the differences among these above-mentioned in-sensor computing devices.

1.3 SCAMP-5d Vision System and Pixel Processor Array

1.3.1 Introduction

SCAMP vision system is one of the emerging in-sensor visual computing devices. Currently, the most up-to-date version of SCAMP series system is the SCAMP-5d (Figs. 1.4 and 1.5) which consists of 256×256 processing elements (PEs) weighted 171 g with a normal lens. SCAMP-5d vision system is a general-purpose programmable massively parallel vision system [23] that was invented, designed, and developed by University of Manchester. By far, SCAMP-5d enjoys many applications in the field of robotics [27–30] and computer vision [31–33]. As for the PPA shown in Figs. 1.3 and 1.4, the photo-detector converts light into an analogue signal which can be directly parallelly processed on AREG. Different from the current hardware design structure of computer vision systems, the PPA gets rid of the Analogue/Digital Conversion (ADC) after sensing and directly operates on analogue electric current using an arithmetic unit, hence accelerating the signal processing speed and, in the meantime, avoiding the bottleneck of ADC and data transmission process. However, errors can be introduced when performing arithmetic operations or temporal information storage on AREG [6].

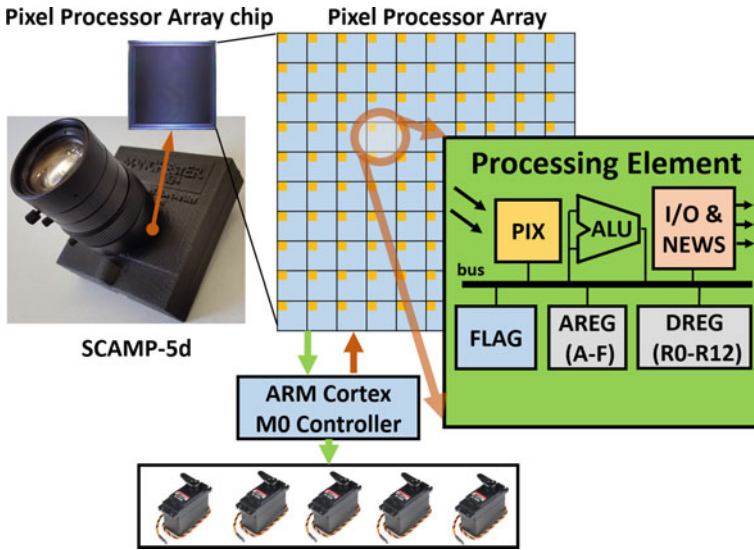


Fig. 1.4 SCAMP-5d vision system and the pixel processor array (PPA). SCAMP-5d consists of PPA with 256×256 Processing Elements (PE) and ARM Micro-controller where parallel image processing is conducted on PPA by directly operating on analogue signal (electric current from PIX, which is proportional to the light intensity) within AREG and bit operation within in DREG. Hence, there is no need for time-consuming and energy-inefficient analogue-to-digital conversion. Bio-inspired by the efficient information processing of neurons connected by synapses, PPA is designed to have highly interconnected PE and registers where information can be shared and accessed adjacently enabling efficient parallel machine vision computing. ARM micro-controller is in charge of sending instructions to the PPA, receiving the processed information from the PPA, and more fully-connected layers for deeper CNN

In terms of hardware techniques, the PPA integrates information storage on registers, image processing and analogue information operation (arithmetic operation, shifting, etc.), digital/bit operation, and logical operations. As can be seen from Fig. 1.4, for each Processing element (PE), there are seven read/write AREG (A to F) which can be used for signed value storage and computation with basic arithmetic operations, such as addition, subtraction, division, etc. In addition, thirteen 1-bit DREG (R0 to R12) in each PE (256×256 in total) can execute the Boolean logical operations, such as AND, OR, XNOR, and NOT [15] with information after binary thresholding on AREG. Each register in PE executes identical instructions synchronously under SIMD instructions, hence enabling parallel image processing. In addition, the FLAG register can activate different areas of registers given corresponding patterns for more flexible operation. With the neighbour access function where each pixel is able to communicate with its four neighbours (north, west, east, south), an efficient parallel image shifting can be implemented easily. Instructions for the PPA are dispatched by the ARM-based micro-controller with a Cortex M0 running at 204MHz. The analogue operations is executed at 5 MHz and digital at 10MHz.

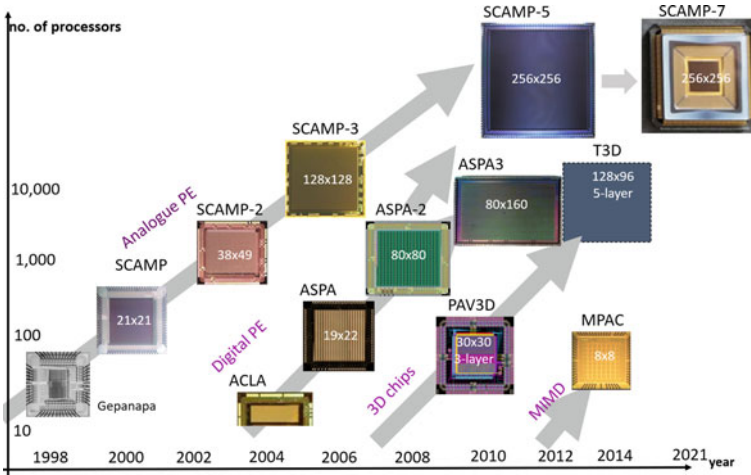


Fig. 1.5 The development process of SCAMP series vision system from the University of Manchester. This review mainly focuses on the SCAMP-3 and SCAMP-5 vision system because their higher resolution and performance would enable more research and applications. *Source* Piotr Dudek’s talk in Second International Workshop on Event-based Vision and Smart Cameras (CVPRW) [39]

Other I/O functions, such as USB2.0, GPIO, SPI, and UART, are performed on Cortex M4 Core [15]. Notice that other names for a similar type of focal-plane sensor processor can be seen from [9] with names, e.g. ASPA (Asynchronous-synchronous Focal Plane Sensor Processor Array), FPSP (Focal-Plane Sensor-Processor).

The PPA is a hardware implementation of Cellular Neural Network (CeNN) with the new optimisation on a mixture of both analogue and digital computing using AREGs and DREGs, respectively. The studies based on the PPA reviewed in this work utilise the parallel nature of the CeNN architecture for efficient and high-performance computing, where each “cell” is intricately connected with its four neighbours and information can be shared efficiently. Hence, the PPA can be modelled as a CeNN architecture for visual information computing. The CeNN processing circuit architecture was first proposed by Leon Chua and Lin Yang [34], followed by the CeNN universal machine [35] as a prototype. After that, as an invention of new circuit architecture and a parallel computing paradigm, it enjoys widespread popularity in academia with a substantial number of research outputs and applications in pattern recognition [36], image processing [37], and biological vision modelling [38]. With above-mentioned hardware features, the SCAMP PPA mainly consists of the following advantages over conventional machine systems.

Efficiency and Low Latency: It is obvious to see from Fig. 1.1c that in-sensor computing gets rid of signal digitisation, transmission, and storage processes onto external devices, hence enabling high-speed image processing [23] and CNN inference [40] which can be integrated with agile mobile robot platforms [27–30]. In addition, the PE distribution and simultaneous instruction execution on PEs allow

efficient parallel signal processing. Carey et al. [23] demonstrate an object detection with a frame rate of 100,000 per second using the SCAMP vision system and Liu et al. [40] propose binary shallow neural network on the PPA with binary classification problem at up to 17,000 FPS. These works show the efficiency of image processing in a sensor once the parallelism mechanism of the PPA is fully taken advantage of.

Low power consumption: According to Fig. 1.1c, there are no external processing units or data processing needed, hence the power consumption can be saved to a large degree. The maximum power cost of the SCAMP-3 vision system for a complex object tracking and counting system is 29 mW [41]. And the overall power consumption on image processing and CNN inference tasks within a SCAMP-5 vision system is lower than 2 W [42]. This feature makes the SCAMP vision system suitable for mobile platforms, usually with short battery life. In addition, according to the power consumption test from work [43], given 8 popular kernel filters, the SCAMP PPA generates the same convolution results with much lower power consumption (>20 times) at a higher speed (>100 times) compared to common CPUs and GPUs.

Data Security and Privacy Protection: An unique but non-negligible feature of in-sensor analogue computing with the PPA is its inherent feature of data security and privacy protection. Data security is feasible because of the focal-plane analogue information processing without ADC, extra data recording, storage, or intermediate transmission procedures. Usually, the only output after analogue computing is the extracted useful target information without redundant information, which is hardly reversible to get the original data for sensitive information or user re-identification [44]. Data security and privacy protection have become prominent challenges with the emergence of the internet of things. Smart devices such as autonomous vehicles, domestic robots, and smart household appliances are usually equipped with perceptual sensors and collect data pervasively in public and private spaces, threatening users' privacy and data security. Conventional sensors usually directly upload raw data to the cloud for data processing [45], which can be a fault line of data security. When data is processed manually or the network is attacked, crucially sensitive data can be directly obtained. The acquired data can then be applied to determine individuals' habits (e.g., motion sensors) or to conduct surveillance (e.g., facial recognition systems), which can cause significant violations of EU GENERAL DATA PROTECTION REGULATION Article 25.¹ In sensor computing first enables only valuable information to be extracted as it's output, without redundant information. Moreover, the minimised raw data are further mocked by the analog signals, which leads to re-identification almost impossible. Hence, users' privacy can be strictly protected with in-sensor processing mechanisms.

¹ <https://gdpr-info.eu/art-25-gdpr/>.

1.3.2 Algorithms and Applications

This section mainly illustrates algorithms and their potential applications (Fig. 1.7) based on the SCAMP-5 vision systems (Table 1.3 and Fig. 1.6).

1.3.2.1 Image Enhancement

Image enhancement comes along with the imaging process on the PPA compared to the conventional image enhancement which only happens after the image data is captured. Later, other methods are exploited on different image processing tasks. For example, Wang et al. [67] proposed a simple coarse grain mapping method to process bigger images than the PPA resolution itself by temporarily storing sub-images into different registers.

Table 1.3 List of main studies with the SCAMP PPA

Image processing methods	Applications	References
Background extraction	Segmentation	[46, 47]
Contour extraction	Object detection	[48–50]
Skeleton extraction	Shape simplification	[51–53]
HDR	Image enhancement	[31, 48, 54, 55]
Feature corner/edge extraction	Edge/feature-based VO	[33]
Target detection/localisation	High-speed object tracking	[23, 56, 57]
Neural network	High-level inference	[40, 42, 58–61]
Depth estimation/visual odometry	Robot navigation	[29, 30, 32, 60, 62–65]
Automatic code generation	Neural network inference, face detection	[43, 66]



Fig. 1.6 Examples of two images with(left)/without(right) HDR algorithms towards the same scene in an outdoor environment

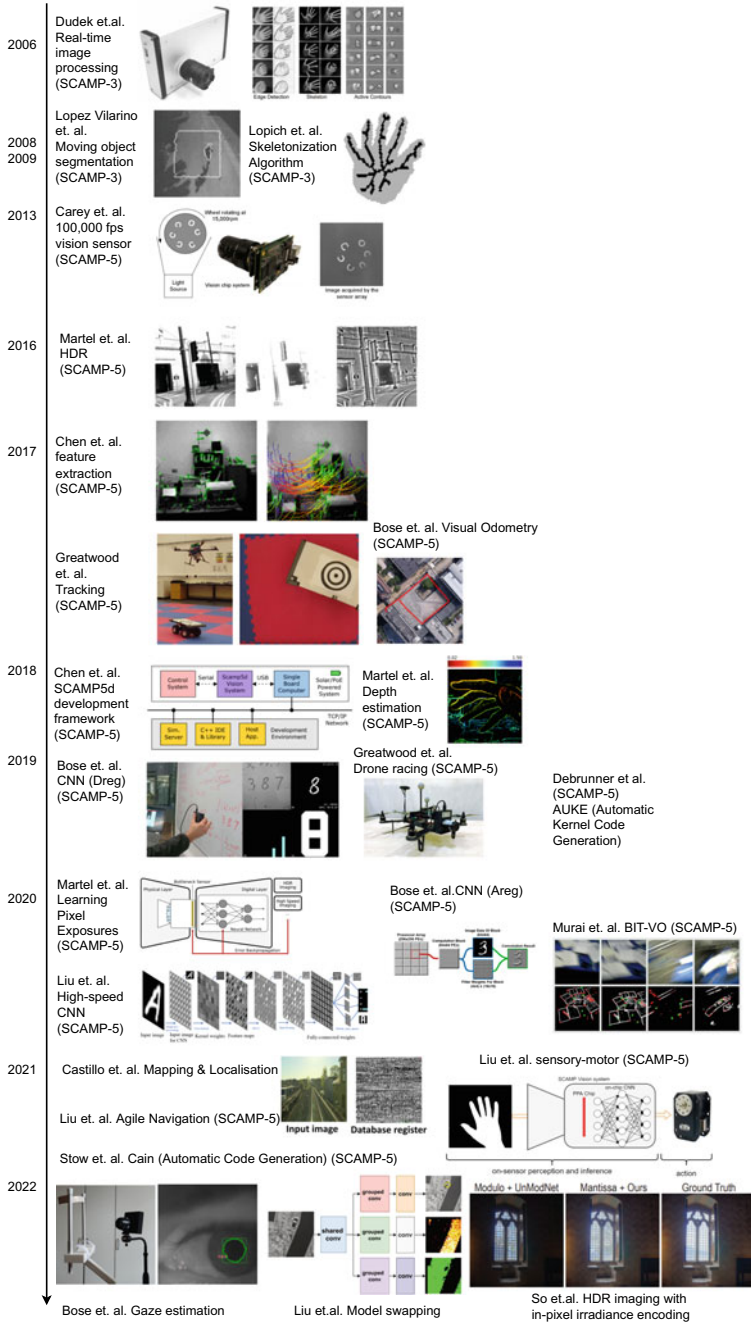


Fig. 1.7 Milestones SCAMP PPA-based work and key SCAMP PPA studies and applications during last 16 years

HDR (Fig. 1.6) is a basic low-level image pre-processing method to obtain rich image information even facing extreme lighting conditions, such as the mixture of dim and strong light intensity. However, conventional image sensors rely on either a global or rolling shutter to form an image, which limits the efficiency of HDR imaging [31, 68]. Back in 2006, Dudek [48] proposed sensor-level adaptive sensing and image processing with SCAMP-3 [41, 69], where different exposure settings are combined for an image with a high dynamic range. Martel et al. [55] make significant contributions in this area using the PPA. The first HDR image generation in-sensor is from [54] where pixel-wise exposure can be controlled to generate HDR images, followed by automotive applications [70]. Furthermore, Bose et al. [32] take advantage of the HDR image to extract edges as the robust input information for visual odometry estimation. However, the usage of iterative exposure for different regions of the image slows down the image pre-processing. To speed up the HDR imaging, Martel et al. [31] propose the learning shutter function for PEs to expose each pixel independently with an end-to-end training strategy. They obtain an exposure function by training a U-Net neural network and compiling these trained functions on the sensor for inference. Later, So et al. [71] presented in-pixel control for snapshot HDR imaging with irradiance encoding.

1.3.2.2 Contour and Skeleton Extraction

Contours are important features for objects within an image, which can help to identify different entities. Contour extraction algorithms were proposed based on a pixel-level snake with very low latency [49]. In 2007, Alonso-Montes et al. proposed the in-sensor automatic retinal vessel tree extraction based on the Cellular Neural Networks [50]. The shared key concept for these works [48–50] is to extract contour iteratively based on the active contour model and Cellular Neural Networks. In 2008, [72] proposed an image pre-processing method based on the cellular automata for a robotic scenario. The skeleton within a binary image shows the object size, position, and simplified shape. Fast image skeletonization [51] is implemented by [52] based on the wave-trigger propagation/collision. Examples of image contour and skeleton extraction based on the SCAMP PPA can be seen in Fig. 1.8.



Fig. 1.8 Examples of image contour and skeleton extraction using SCAMP PPA. Left: Extracted Retinal Vascular Tree, Figure from [50]. Right: Extracted skeletons, Figure from [47]

Other Feature Extraction Methods: Other image processing methods, such as background extraction, is exploited by Wang et al. [46, 47]. For higher-level feature extraction, the edge feature can be obtained by deploying Sobel kernel filters or Laplacian filters, which are used in the later work for focal plane visual odometry [32] and neural networks [60]. As for other features, such as corner points extraction, Chen [33] utilised the DREG for corner points extraction based on the FAST16 algorithm, which is used in later work on visual odometry [65]. Based on the above-mentioned low- and mid-level image processing methods, researchers are motivated to exploit more general high-level image processing with up-to-date techniques by taking advantage of the earlier milestone work and the state-of-art progress, such as neural networks which would be illustrated in Sect. 1.3.2.4.

1.3.2.3 In-Sensor Visual Feature Extraction for Robots

Two major constraints that preclude mobile robots from long-term and diverse applications are their short battery life and limited load. Emerging sensors may hold the key to solving this challenge due to their unique low-level hardware design. The portable SCAMP-5d vision system (171 g including the lens) can perform spatial AI processing in-sensor, reducing data transfer pressure between sensing and the main processor, hence increasing overall processing efficiency while maintaining low power consumption [73].

(a) SCAMP PPA on a Quadcopter

The SCAMP-5d vision system has been integrated into quadcopter systems for target tracking, visual odometry and racing. Greatwood et al. perform various experiments by integrating a SCAMP-5d vision system and a quadrotor [28, 29, 74]. Figure 1.9 shows a flight control system in terms of hardware integration and control block diagram, where a pre-set target can be tracked with extracted useful information on sensor even facing short periods of target tracking loss [28]. In this application, the direct in-sensor target position extraction releases the pressure of image capturing, transmission and processing for the whole system. Later, Greatwood et al. proposed the in-sensor visual odometry using perspective correction on an agile micro air

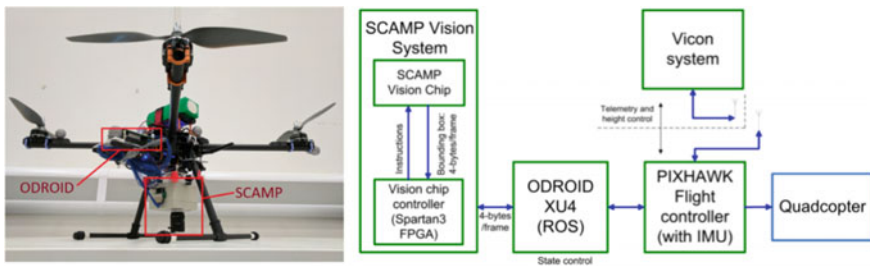


Fig. 1.9 Left: The integration of a quadrotor and SCAMP-5 vision system for object tracking. Right: a diagram of system hardware (Figure from [28])



Fig. 1.10 Quadrotor setup for the drone racing with a front-facing SCAMP (Figure from [74])

vehicle based on a similar hardware platform. After that, a drone racing Fig. 1.10 within a pre-set environment is demonstrated by taking advantage of the efficient image processing ability on the PPA [74], where the target position can be estimated at around 500 FPS. McConville et al. [30] apply the in-sensor visual odometry developed by Bose et al. [32] on an unmanned aerial system for real-time control purposes.

(b) SCAMP PPA for Mobile Robot Reactive Navigation

In terms of navigation with a SCAMP PPA, Liu et al. [27] proposed reactive agile navigation on a non-holonomic ground vehicle using PPA by robustly recognising pre-set patterns out of complex environment background. Although being very efficient and accurate, using a pre-set fixed pattern for target tracking is difficult to expand in the generalised environment where there are usually random features. With this in mind, Chen et al. [60] use in-focal plane feature extraction from the environment to perform a recurrent neural network on the M4 micro-controller using this extracted information to estimate the proximity to the ambient objects for obstacle avoidance purposes. A similar pattern of concentric circles was employed in [27, 28, 30] to effectively extract the dot centre in the circles out of the complex environment (Fig. 1.11).

(c) In-Sensor Computing for Mapping and Localisation

Mapping and localisation are useful techniques for robot navigation. In-sensor mapping and localisation are lightweight and low power cost solutions for mobile platforms. Castillo-Elizalde et al. [75] for the first time proposed 1-D mapping and localisation technique. For this method, features are firstly extracted as the database from a sequence of images. Then, the incoming image can be localised by comparing with the database and the prior knowledge of the motion model. In their work, two methods were utilised to down-sample the original images: direct resizing and local binary pattern to apply them to different localisation situations.

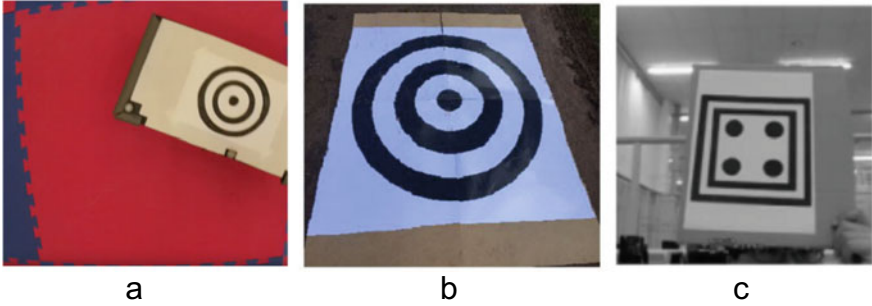


Fig. 1.11 Tracking pattern for the drone and ground vehicle. **a** Tracking a ground vehicle [28], **b** Tracking a moving target while performing a visual odometry [30], **c** Tracking a fixed pattern with a mobile ground vehicle [27]

(d) Pose and Depth Estimation

For decades, egocentric state estimation has been studied using conventional cameras, emerging DVS devices, and CPU/GPUs. In recent years, there have been some studies utilising SCAMP PPA. For example, Bose et al. [32] for the first time, proposed in-sensor 4 Degree-of-Freedom (DoF) visual odometry wholly on the sensor by mapping the real-time input image with the previous keyframe through image scaling, shifting, rotation and alignment. They demonstrate the visual odometry estimation at over 1000 Hz with around 2 W power cost. Debrunner et al. [76] use the SCAMP to estimate its global motion with the tiling method at 60 FPS with a low power cost of 100.2 mW. After that, Murai [65] proposed 6 DoF visual odometry based on edge and corner points extracted on sensor and post-processing on a computer with a frame rate of 300 FPS. They take advantage of feature edge, and corner extraction methods [33] and calculate the visual odometry off sensor using a similar strategy with the standard Visual Odometry (VO) systems [77]. Although they combine in-sensor feature extraction and ready-to-use VO computing method off the sensor, it is promising to be a direction in the future to combine the efficient image pre-processing in-sensor and high-volume post-processing with a powerful CPU/GPU, especially when facing storage shortage and general calculation resources for the large-scale computing.

In addition, the SCAMP vision system can also work with other accessories to share the computation burden for more applications. For example, Martel et al. [62–64] mounted a controllable liquid lens to generate a semi-dense map in real-time, which is the first work on depth estimation to take advantage of external physical accessories. With this focus-tunable lens, a vast amount of computation pressure on the sensor is relieved. This in-sensor feature extraction and post-image processing on controller scheme are also widely used in many different applications [60, 65], where the task requirement of storage and computing resources is out of the capacity of the PPA.

1.3.2.4 Research Progress on Neural Networks with a SCAMP PPA

The algorithms of SCAMP PPA proposed earlier mainly focus on low-level image processing and/or machine vision methods to enhance image quality and extract basic textures with combinations of inherent built-in functions based on SCAMP-3 and SCAMP-5 with a PE resolution of 128×128 and 256×256 , respectively. It should be noticed that these developed image processing methods are deeply related to the hardware design of the SCAMP vision system. For example, common methods used in this period are cellular-based algorithms, including cellular neural networks, because the SCAMP PPA itself is a cellular processor array.

Research on neural network inference with the SCAMP PPA has been active in recent years. Table 1.4 lists the main research work in the area of neural networks, which covers fully convolutional neural networks and binary convolutional neural networks using DREG or AREG with various datasets and applications. High-level image processing, such as object classification, localisation and segmentation in sensor, is achieved with the neural network. The deployment of neural networks onto the PPA is a breakthrough since it enables the PPA open to more possibilities with universal methods, which is unlike the conventional development methods with some combinations of low-level image processing methods for specific tasks. With the use of CNN, several types of tasks, such as classification, regression, localisation,

Table 1.4 Different convolutional neural network implementation with SCAMP and performance comparison.

Network	Filter number	Layers (Conv+FC)	Dataset	Accuracy	Frame rate (fps)	In-sensor/ Near-sensor
Bose [42]	16 5×5	1 + 1	MNIST	$\approx 94.2\%$	210	1Conv/1FC
Bose [58]	16/64 4×4	1 + 1	MNIST	$\approx 93\%$	>3000	In sensor
–	16+16 4×4	2 + 1	MNIST	92–94%	224	In sensor
Liu [40]	64 4×4	1 + 1	8 Plankton	80.5 %	4016	In sensor
–	16 4×4	1 + 2	8 Hand gestures	<98.7%	2092	1Conv+1FC/1FC
–	16 4×4	1 + 1	Roshambo	<97.73%	8264	In sensor
–	64 4×4	1 + 1	0,1 in MNIST	<99.1%	17543	In sensor
Liu FCN [59]	16+64 4×4 64 1×1	3 + 0	Simulation	–	283	In sensor
Liu [78] Binarized CNN	16+64 4×4	2 + 2	EMNIST	<86.74%	178	2Conv+1FC/ 1FC
Chen [60]	–	0 + N	Collected indoor	–	–	Near sensor
AnalogNet [61]	3 3×3	1 + 3	MNIST	96.9%	2260	1Conv/2FC

and segmentation, can be feasible, hence enabling more applications. Table 1.4 shows the neural network-related work based on the SCAMP PPA vision system.

The research on CNN implementation and inference within PPA is pioneered by Bose et al. [79] where a CNN with a single convolutional layer performed upon the PPA array and a fully-connected layer upon its controller chip (M0). They performed 16-bit image convolution operations using 4×4 DREG “Super Pixel” blocks and demonstrated live digit classification based on MNIST dataset at around 200 FPS. In their work, the ternary $\{-1, 0, 1\}$ kernel filters are stored on the flash (M4) of the PPA system, and are effectively encoded in the instructions/operation sent to the PPA array, performing convolutions sequentially. Furthermore, a mobile car localisation task is then explored using synthetic datasets, where the pre-processed edge information is mainly the clues for network inference. Notice that the localisation is realised by classifying the car’s position along the x and y axis, respectively. To fully take advantage of PPA’s parallel computing characteristics and to further improve the CNN inference efficiency, Bose et al. [58], for the first time, proposed the idea of in-pixel weight storage, where the network’s weights are directly stored within the registers of the PPA’s PEs. This method enabled both parallel computations of multiple convolutions, and implementation of a fully connected layer upon the PPA array, resulting in a $\times 22$ faster CNN inference (4464 FPS) on the same digit recognition task. Based on these two works, [40] further proposes a high-speed lightweight neural network using BinaryConnect [80] with a new method for computing convolutions upon the PPA, allowing for varying convolutional strides. This work demonstrated four different classification tasks with frame rates ranging from 2,000 to 17,500 per second with different stride setups. Later, based on this network, a direct servo control using CNN results [81] and a simulated robot tracking from a drone [82] with in-sensor CNN computing results are exploited. In addition, the AnalogNet2 [61, 83] extends the earlier work in [84], implementing a CNN which reaches 96.9% accuracy on the MNIST dataset at a speed of 2260 fps. However, their method requires all fully connected layers to be performed externally to the PPA array with only 3 convolutional kernel filters implemented in sequence on the PPA as the first layer. More kernel filters would significantly slow down the inference process. Notice that, in our work [60], a recurrent neural network is implemented on the micro-controller with features extracted on a sensor. In this manner, the fully-connected layer of a neural network can be deployed similarly with conventional embedded devices. It is notable that Martel et al. trained a neural network of exposure time for each individual pixel off the sensor for HDR imaging and video compressive sensing [31].

Furthermore, work [78] binarized CNN with batch norm both for classification and coarse segmentation. To deal with the classifications application with more labels and more segmentation tasks, they propose the idea of dynamic model swapping by uploading weights of trained models in sequence or according to the last inference result, targeting multiple sub-tasks decomposed from a more sophisticated task. They then demonstrate a servo control directly using the CNN inference results [81], which potentially indicates that motion control platforms, such as a ground vehicle or drones can have a light-weight servo control system without using external control units in the future.

Notice that the preceding neural network-related work mainly focuses on classification or classification-based localisation, both of which require fully connected layers. However, the parameters in fully-connected layers are typically substantially larger than those in convolutional layers due to the dense connections of each individual neuron. Thus, work developed a fully-convolutional neural network (FCN) [59], not only presenting in-sensor image segmentation and localisation but also eliminating dense layers for a smaller memory footprint [85].

1.3.2.5 In-Sensor Cellular Automata

The PPA itself is a cellular neural network architecture where each ‘cell’ is closely connected with its four neighbours, hence information can be shared efficiently. With this in mind, the author is inspired to explore the possibility to perform cellular behaviour, such as Conway’s game of life (demonstration shown from²) and elementary cellular automata (demonstration Rule 90 seen from³) based on the theory of cellular automata [86]. With the rule of the game of life, all ‘cells’ can update their states (alive or dead) in-sensor as fast as $53\ \mu\text{s}$ for each iteration based on the bit-operation with DREG. As can be seen from Fig. 1.12, a Sierpiński triangle is efficiently generated based on bit operation on the sensor with $730\ \mu\text{s}$ of 255 iterations to fill the whole chip.

(a) Elementary CA

One-dimensional CA is one of the simple CA algorithms. Some classical updating rules, such as Rule 30, Rule 90, and Rule 110, can be implemented by logic bit

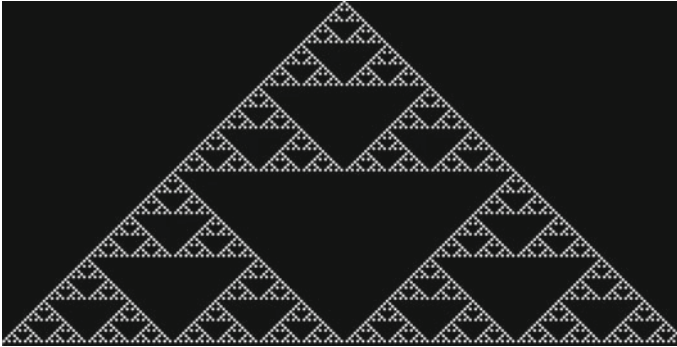


Fig. 1.12 Our demonstration of elementary cellular automata with Rule 90 on the SCAMP PPA. This pattern is generated from top to bottom. We have made this project available from https://github.com/yananliusdu/1D_CellularAutomata

² https://youtu.be/X_t4c3f-T4s.

³ https://youtu.be/HgPvoK5EJ_s.

operations. Let use L,C,R as the three continuous grids and C1 is the grid that should be updated, then rules can be represented as follows:

```
R30: C1 = L XOR (C OR R )
R90: C1 = XOR(L, R)
R110: C1 = XOR(OR(R, C), AND(R, C, L))
```

The pseudo codes for R90 can be shown as follows:

```
\begin{code}{ElementaryCellularAutomataR90}
for(int i = 1; i <= iterations; i++)
{
    scamp5_kernel_begin();
    MOV(R5,R6); //R5 = R6
    DNEWS(R0,R6,west); //R0 = shift R6 to right for one step
    DNEWS(R6,R0,west);
    XOR(R7,R5,R6); // R7 = XOR(R5,R6)
    DNEWS(R0,R7,north);
    DNEWS(R6,R0,east);
    OR(R6,R5); R6 = OR(R6,R5)
    scamp5_kernel_end();
}
\end{code}
```

Implementation for R30 and R110 can also be done similarly with R90 using corresponding unit logic operations.

(b) In-Sensor Conway's Game of Life based on the CA

The rule of Game of Life is a new independent non-linear computing scheme.

Figure 1.13 shows an updating 3×3 block to illustrate the rule and implementation using a cellular neural network. The pseudo code for the Game of Life can be represented as follows according to its definition:

```
if B0 = 1 & SUM(B1, ..., B8) <= 2
    B0 = 0
If else B0 = 0 & SUM(B1, ..., B8) == 3
    B0 = 1
else
    B0 = 0
```

where $B_i \in \{0, 1\}, i = \{0, 1, 2, \dots, 8\}$.

This chapter tries to implement and run the Game of Life using the DREG of SCAMP-5 vision systems and its parallel nature. We use three DREGs 3-bit

Fig. 1.13 The update block for the Game of Life

B1	B2	B3
B4	B0	B5
B6	B7	B8

$R6, R7, R8$ to count the number of ‘1’s around the pivot $B0$. $R5$ is the source binary image in this illustration. $R1, R2, R3, R4$ and $R9, R10$ are DREGs to temporarily store intermediate states.

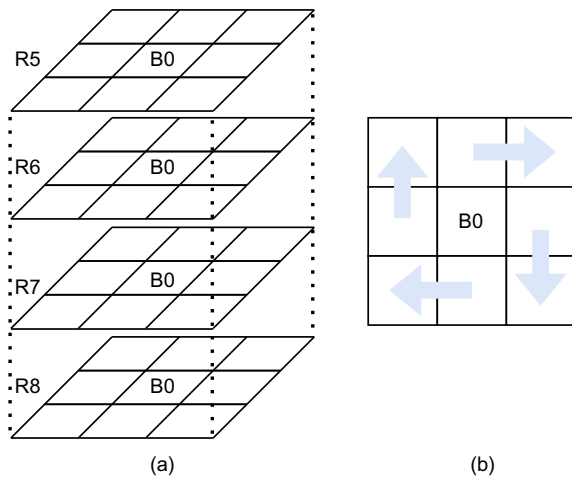
Neighbour Number Counting

As the first step of the 2D cellular automata, the number of live neighbours should be counted. As shown in Fig. 1.14a, $R5$ is the source binary image. $R6, R7, R8$ are 3-bits to represent the number of live neighbours of $R5$ in the corresponding position. For example, if $R5(B0)$ has 2 live neighbours, then $R6(B0) = 0, R7(B0) = 1, R8(B0) = 0$. In conclusion, binary digits $R6, R7, R8$ are used to record the number of live neighbours of corresponding cell. Figure 1.14b shows for each counting step (8 in total around a pivot, $R6, R7, R8$ are updated according to the states of each cell.

State Update for Cells

With live neighbour information stored in three DREGs, cells’ state can be updated according to the rule of the Game of Life as described in the aforementioned pseudo code. We make the codes of Game of Life on the SCAMP-5 vision system available from <https://github.com/yananliusdu/GameofLife>. In the future, more image

Fig. 1.14 Neighbour number counting using DREGs



processing-related work can be potentially explored as long as proper update rules and associated steps are trained with neural network methods [37, 87].

1.4 Eye-RIS

1.4.1 Introduction

Eye-RIS [22, 88] commercial vision system on Chip (VSoC) extends CMOS pixel functionality with image storage (7 gray-scale images and 4 binary images) and digital/analogue signal processing ability. Specifically, a 32-bit RISC (Reduced Instruction Set Computer) is integrated with a vision sensor for image post-processing after the parallel in-sensor pre-processing (Fig. 1.15). The resolution of the Eye-RIS vision sensor is 176×144 . Notice that Eye-RIS’s overall functional diagram is similar to that in the SCAMP vision system, where the counterpart of RISC is the M0 microcontroller in the SCAMP PPA [89]. The most significant difference, though, is that the Eye-RIS has a DICop part, which is a digital image co-processor that handles geometric transformations and can send the results back to the pixel level for more processing.

The Eye-RIS Vision System on Chip (VSoC) is an autonomous device combining a parallel CMOS image sensor processor with 32-bit RISC microprocessor performing post-processing and system control tasks, several I/O and high-speed communication ports that allow the system to communicate and/or to control external systems, and on chip memory. The combination of massive parallel image pre-processing in the sensor with complex image post-processing in the microprocessor results in ultra compact implementation of a vision system able to perform complex machine vision algorithms at speeds of several thousands of images per second. The Eye-RIS VSoC

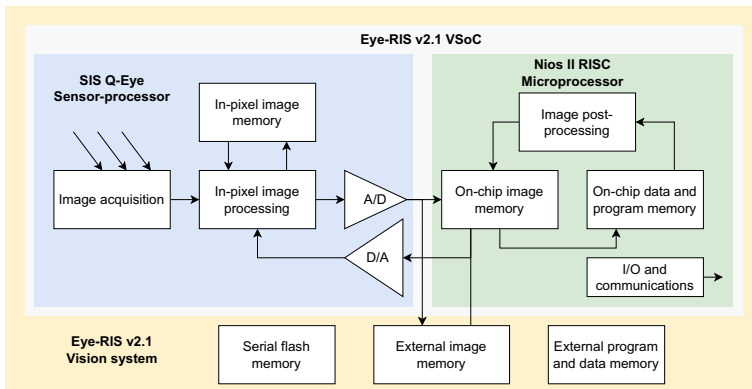


Fig. 1.15 Eye-RIS v2.1 VSoC architecture

features a complete application development software environment allowing easy control of the device and is optimized for industrial applications requiring image sensing, image processing, and decision making at extreme frame rates.

1.4.2 Applications

The applications with Eye-RIS consist of automotive, machine vision, security, games and battery powered products. Caballero-Garcia and Jimenez-Marrufo [90] proposed a series of techniques to deploy image processing algorithms on the Eye-RIS Vision System on Chip (VSoC) for various applications. One unique characteristic using the Eye-RIS vision platform compared to conventional visual sensors is the simultaneous image acquisition and early processing in the analogue domain.

Paper [91] aims to describe how the AnaFocus' Eye-RIS family of vision systems has been successfully embedded within the roving robots developed under the framework of SPARK and SPARK II European projects to solve the action-oriented perception problem in real time. With the ability of low power cost and efficient parallel image processing, Eye-RIS has been equipped to many different mobile robot platforms, such as Rover II wheeled robot and Gregor III hexapod robot. Visual homing, object tracking, and navigation using landmarks are demonstrated based on the robot platforms and in-sensor real-time image processing algorithms (Fig. 1.16).

Optical flow based on the Lucas and Kanade by Guzman et al. [92] (Fig. 1.17) is implemented on the Eye-RIS platform taking advantage of both analogue and digital signals processing using Q-Eye and Nios II RISC respectively (Fig. 1.15). In the experiment, the optical flow estimation reaches over 25 fps which can be used in the area of robotics in real time. Specifically, the optical flow constraint equation:

$$uI_x + vI_y + I_t = 0 \quad (1.1)$$

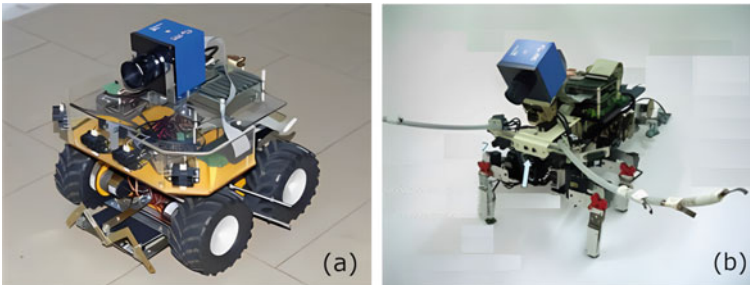


Fig. 1.16 Eye-RIS VSoC equipped onto robot platforms. **a** Rover II wheeled robot, **b** Gregor III hexapod robot (Figure from [91])

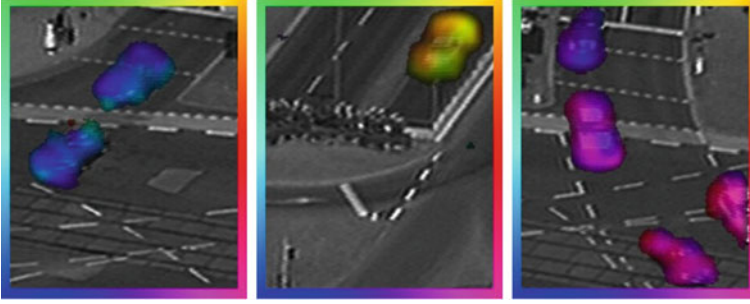


Fig. 1.17 Optical flow estimation in traffic sequences using Eye-RIS VSoC architecture (Figure from [92])

where the partial derivatives of I are denoted by subscripts, which can be obtained from the image. u and v are the x and y components of the optical flow vector, which are the optical flow vectors that need to be found out.

Lucas–Kanade method [93] is a classical method to deal with the optical flow constraint problem. According to the assumption of Lucas–Kanade method, given a small pixel block, 3×3 for example, the optical flow remains identical within this small block. Then, we can have following equation group:

$$\begin{aligned}
 uI_{x1} + vI_{y1} &= -I_{t1} \\
 uI_{x2} + vI_{y2} &= -I_{t2} \\
 &\dots \\
 uI_{xn} + vI_{yn} &= -I_{tn}
 \end{aligned} \tag{1.2}$$

Equation 1.2 can then be represented as

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \dots & \dots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \dots \\ -I_{tn} \end{bmatrix} \tag{1.3}$$

we assign Eq. 1.3 as $A\vec{V} = -b$, here the least squares method can be used to get optical flow vector \vec{V} , then $A^T A \vec{V} = A^T (-b)$, hence, the optical flow vector can be obtained through:

$$\vec{V} = (A^T A)^{-1} A^T (-b) \tag{1.4}$$

In detail, Eq. 1.4 can be shown as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n I_{xi}^2 & \sum_{i=1}^n I_{xi} I_{yi} \\ \sum_{i=1}^n I_{xi} I_{yi} & \sum_{i=1}^n I_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^n I_{xi} I_{ti} \\ -\sum_{i=1}^n I_{yi} I_{ti} \end{bmatrix} \tag{1.5}$$

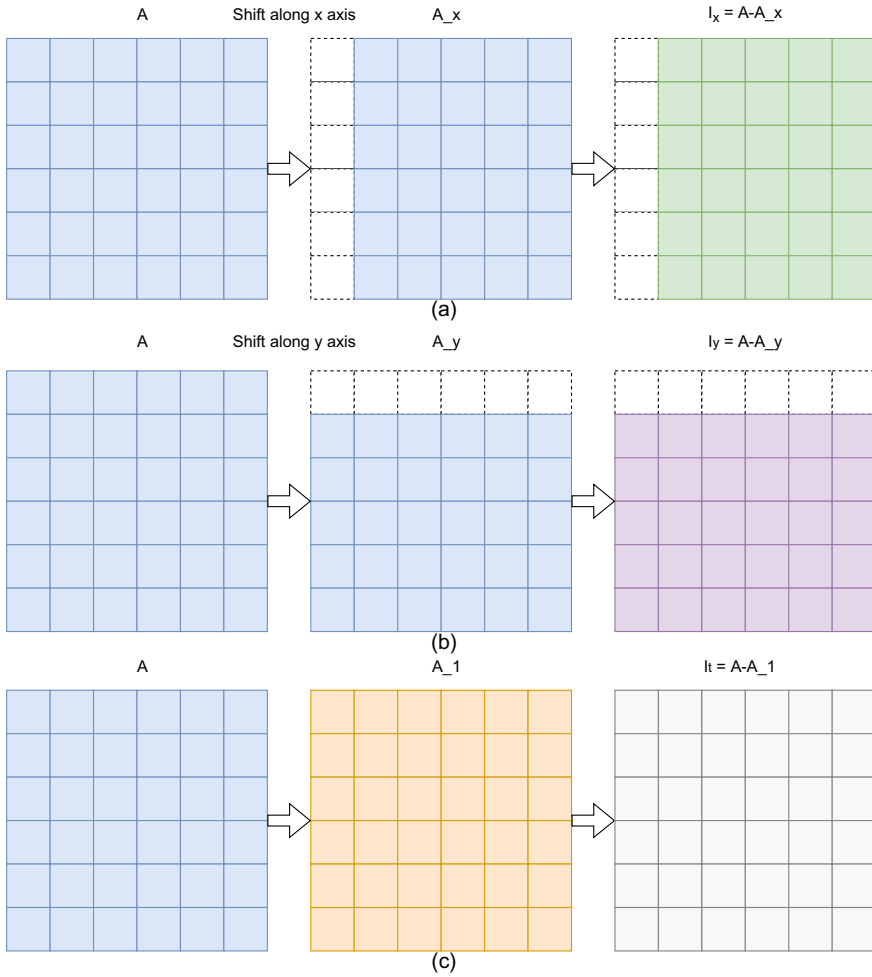


Fig. 1.18 The parallel calculation of I_x, I_y, I_t in the focal plane using analogue signals

Through Eq. 1.5, the optical flow vector can be estimated through a sequence of images. These intensity derivatives including I_x, I_y, I_t can be efficiently obtained by shifting and subtracting between frame sequences as illustrated from Fig. 1.18. After that, the optical vector can then be calculated using conventional computing units. As for the implementation of optical flow using the above-mentioned formulations, work [22] takes advantage of both in-sensor pre-processing and post-processing with computing units achieving up to 28.9 fps.

In the study of [94], authors explored different methods of point tracking on the platform of Eye-RIS, which is able to equip Unmanned Aerial Vehicles (UAVs) with the ability of on-board sensing and computing with low load and power consumption.

Nicolosi et al. [95] applied the Eye-RIS vision platform to control welding process by using the cellular neural network based visual algorithms. [96] extended their work onto vision based closed loop control for partial penetration welding of overlap joints.

Work [97] proposed algorithms for the cellular neural network to detect rapidly approaching object which is bio-inspired by mammalian retina that consists of looming sensitive circuit based on local interaction of cells.

1.5 Kovilta's KOVA1

1.5.1 Introduction

The KOvilta Vision Array (KOVA1) (depicted in [11]) employs a meticulously designed pixel-level processing circuitry that utilizes an efficient combination of analogue and digital (mixed-mode) computation techniques to execute a diverse range of pre-programmed operations. These operations include automated sensor adaption, grayscale filtering, segmentation, and complex object-level visual analysis. The selection of operations to be performed at the pixel-level hardware can be customized based on the specific requirements of the application, thereby enhancing the overall implementation efficiency.

The KOVA1 is Kovilta's inaugural silicon rendition of the KOvilta Vision Array structure, and it comprises a 96×96 pixel focal-plane processor array fabricated using 180nm CMOS technology. The sensor-processor chip is integrated into a miniature smart camera system equipped with FPGA-based control and Ethernet I/O. In this focal plane processor architecture, each pixel-cell of the sensor array includes a reconfigurable processing element that operates directly on the output of the analog photodiode. This allows real-time data compression for capturing images with high dynamic ranges without compromising quality. Additionally, processing in parallel on the pixel plane enables rapid low-level feature analysis and eliminates the need for time and energy-consuming long-distance data transfers from the sensor to an external processor. The sensor output may include only the essential feature data, such as the presence of an object or a set of object coordinates or features, thereby reducing the amount of hardware needed for further external image content analysis.

The KOVA1 camera system employs pixel cells that are connected within their immediate neighbourhood, allowing for direct information exchange during image analysis activities at the sensor level. Moreover, local memories integrated at the pixel level facilitate the storage of multiple full images or intermediate processing outcomes on the sensor plane. An FPGA chip is utilized to manage the program execution and I/O of the sensor-processor chip in the KOVA1 embedded camera system. As the control and I/O structures consume only a small fraction of the FPGA's resources, supplementary visual analysis operations can be implemented on the FPGA to enhance the on-chip sensor-level processing. The KEDE software

environment from Kovilta is used to program and manage the camera, with programming and data I/O facilitated by an Ethernet connection, while direct CMOS-signal outputs bypass the network interface. Additionally, the camera's internal memory is capable of storing program code. It can operate autonomously without the need for a PC connection and transmit output data to an Ethernet-connected or directly controlled device.

1.5.2 Applications

Line detection: Santti et al. [98] proposed a line detection method by combining KOVA1 and FPGA for low-level and mid-level feature extraction respectively. This line detection method can be applied for industrial inspection and control applications with its performance of high-speed processing and low power consumption.

Seam and Spatter Tracking: In Lahdenoja et al. [24, 99] and Santti et al.'s work [100], KOVA1 is utilised for seam tracking for real-time robot path optimisation during a high power laser welding process. The reason to use this type of pixel-level sensor lies in the ability to control pixel-wise exposure periods facing significant intensity differences in a laser welding task. Hence, the effective binary feature points can be extracted on the focal plane and then this compressed information is sent to the FPGA for laser beam location extraction using Hough transform. With a frame rate over 1000 fps, the combination of in-sensor image pre-processing and FPGA-based Hough operation enables a real-time optical seam tracking for robot control. In addition, spatter can also be tracked in laser and manual arc welding [101] in extreme radiated light intensity conditions (Fig. 1.19).

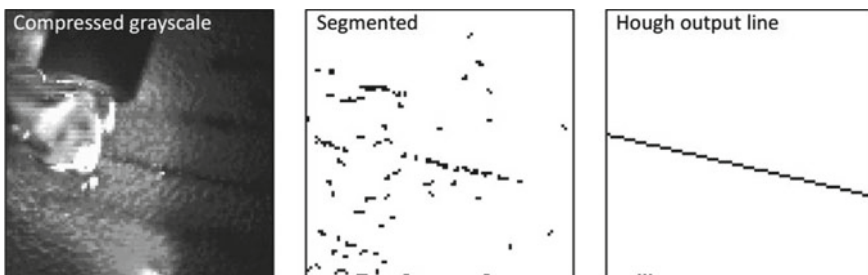


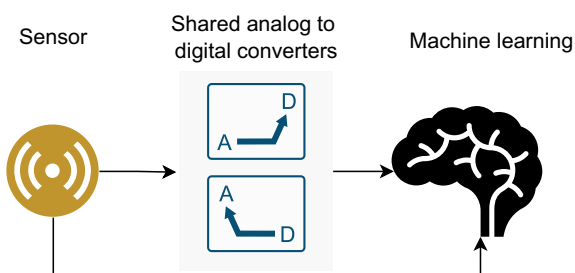
Fig. 1.19 Straight line extraction process using KOVA1. Left to right: figure captured with KOVA1, binary feature extraction using pixel-level process, estimated beam line with binary features using FPGA. Figure from [100]

1.6 Aistorm Mantis2

Mantis system is based on the event-driven charge domain for analogue signal processing without digitisation and provides an “always on” solution for analogue signal processing. One of the key features claimed by Aistorm is the noise cancelling techniques associated with the analogue signals. In addition, artificial intelligence can also be integrated into a chip for various applications. A 96×96 AI-in-Sensor machine learning SoC that is particularly well suited for classification jobs is the AIS-C100A Mants. It has a robust post wake-up ML capabilities as well as a configurable “always on” wake-up CNN engine that can be used to activate external micro-controllers when an object of interest is recognised. On a single monolithic device with the fewest possible external components, all necessary supporting circuitry is provided, including power management, timing, artificial intelligence, and communications in addition to a (up to) 40 mA LED driver enabling both linear and PWM control. An SPI port is used for communication. The on-board camera’s photos and videos as well as those transferred via the SPI connection can both be processed by Mantis. To provide the best contrast for AI calculations, the exposure time can be either internally or externally regulated. The AIS-C100A is housed in an OLGA package of 6.4×6.4 mm. The AI-in-Imager solutions that can directly take pixel data in its native charge form are AISTorm’s Mantis Family of AI-in-Imager processors. The end result is the only method in the world that can wake up a person, face, object, or action based on an image (Fig. 1.20).

There are several businesses that provide analogue AI solutions, but fundamental physical noise and bandwidth constraints prevent these products from being successful. The approach used by AISTorm is charge domain processing. This technology uses charge to create AI-in-Sensor processing opportunities for picture or audio improvement that simply cannot be realised through any other means. Other analogue methods’ noise and bandwidth restrictions are solved through charge domain processing, a revolution in processing. Over both analogue and digital solutions, charge domain solutions are preferable because they can immediately take IoT sensor data without the expense, power, or delay of digitization. The major applications proposed using the Mantis AI-in-Sensor chip cover motion tracking, gesture clas-

Fig. 1.20 AIS technology SKIPS digitization and moves directly to processing



sification, smart home, wearable devices (glasses, headsets), household appliances, gaming devices, automotive, and voice processes as claimed in their application website <https://aistorm.ai/applications/>.

1.7 Other In-Sensor Computing Devices

MIPA4k: MIPA4k [25] is an earlier sensory-level image processing sensor with similar architecture to KOVA1. A number of algorithms can be carried out based on the MIPA4k with the use of mixed signals. These algorithms mainly cover edge detection [102], local binary pattern [103], locally adaptive image sensing [104], space-dependent binary image processing [105], and object segmentation and tracking [106]. Similar early cellular vision chips also include ACE16K [107].

Memristor-Based Devices: Yao et al. [108] Memristor-based hardware is a platform to deploy the neural network using the programmable resistance within the integrated circuits mimicking the synaptic connections in a human brain [17, 108–111]. However, it integrates only storage and processing functions, which can be regarded as in-memory computing. Hence, signals should be input from sensors or other storage devices. They are thus usually integrated with other sensory systems for information processing. Lee et al. [19] take advantage of photo-diode and memristor crossbar for primary visual information process aiming to extract useful information from the input images. In-Sensor visual computing with memristor or new materials are currently not mature enough to support various practical applications for machine vision tasks. By far, there are few practical applications using memristor in-sensor computing devices.

Dynamic Vision Sensor (DVS): Invention [26] DVS produces data in the form of sparse contrast-change events that facilitate low-latency visual processing using external computational hardware [112–114]. These binary events are generated from in-sensor processing according to the brightness changes. Although the pixels in a DVS have a primitive in-sensor processing ability by binarising brightness changes, it achieves an ultra-high-speed response to the environment when working with external hardware computing units, enabling an enormous potential for robotics and computer vision in a challenging environment [115].

Other Emerging Sensor Devices: Mennel et al. [14] use a 2D semiconductor (WSe_2) photodiode array as the vision sensor, the photoresponsivity matrix to store the connecting weights of the neural network, where both supervised and unsupervised learning for classification are present. However, laser light and a set of optical systems are needed to project images onto the chip, which prevents it from having practical usage. Song et al. [116] proposed a CMOS-based PIP (Processing-in-Pixel) architecture where image convolution (8-bit weight configuration) can be performed as the image preprocessing before image data is read out. In addition, Datta et al. proposed the Processing-in-Pixel-in-memory paradigm where the first few convolutional layers of a CNN can be processed and the compressed data then sent to other near-sensor processors [117].

1.8 Conclusion

In-sensor visual computing is an emerging technology that enables efficient extraction of information solely on the sensing chip of the sensory system, facilitating signal collection, storage, and processing. Vision is the primary sense for human beings, and it accounts for a vast majority of information acquisition worldwide. Despite significant advances in computer vision using conventional camera vision systems and associated methodologies, various limitations, such as system latency and power consumption, persist. Image digitisation, storage, and transmission introduce latency, which is a bottleneck that impedes conventional machine vision systems from promptly responding to environmental changes. Additionally, traditional machine vision systems with loosely integrated sensors and processors have high energy costs, weight, and size, making them unsuitable for portable tasks.

To address the limitations of conventional machine vision systems, this chapter explores a new visual information processing scheme using a focal plane sensor processor (FPSP) that directly processes signals where they are collected, thereby avoiding issues with latency, power consumption, and size. The chapter focuses on mobile robotic control systems that utilize in-sensor computed results for multiple navigation research, and investigates novel parallel visual inference approaches, particularly machine learning-based algorithms, to extract higher-level information from analogue signals. To achieve this, a lightweight and high-speed binary convolutional neural network is proposed to categorize objects using efficient addition/subtraction and bit shifting operations. However, implementing neural networks on the focal plane is challenging due to hardware resource constraints and analogue noises. Therefore, this work proposes purely binarised convolutional neural networks with both binary weights and activations, trained with batch normalisation and adaptive threshold to binarise activations and alleviate noise. With this approach, only a small amount of extracted information is obtained, allowing for more efficient data transmission with less bandwidth, and enabling the establishment of an edge computing platform based on the neural network and PPA.

Furthermore, prior research has explored visual sensors for information collection but not for signal processing or motion control; however, this chapter investigates the direction of image processing on the sensing chip and servo motor control using the sensor's digested data directly. Thus, by merging in-sensor neural network inference and direct servo motor control, a sensory-motor system is presented. Moreover, with our proposed dynamic model swapping scheme, more sophisticated classification tasks than earlier work can be achieved. Lastly, a new in-sensor neural network architecture, fully convolutional neural networks, is presented for localisation and coarse segmentation tasks without using the fully connected layers. To deploy this new architecture of a three-layer neural network on the sensor, group convolution is introduced and implemented, with both binary weights and activations, making the fully convolutional neural network compact enough to be embedded on the sensor.

Acknowledgements This work is funded by National Natural Science Foundation of China (grant 62202285), EPSRC FARSCOPE CDT for future robotics and autonomous system, and China Scholarship Council (grant 201700260083).

References

1. Frisby JP, Stone JV (2010) Seeing: the computational approach to biological vision. MIT Press, Cambridge, MA
2. Lao J, Yan M, Tian B, Jiang C, Luo C, Xie Z, Zhu Q, Bao Z, Zhong N, Tang X et al (2022) Ultralow-power machine vision with self-powered sensor reservoir. *Adv Sci* 2106092
3. Wan T, Shao B, Ma S, Zhou Y, Li Q, Chai Y (2022) In-sensor computing: materials, devices, and integration technologies. *Adv Mater* 2203830
4. Interim report for the decadal plan for semiconductors (2020) Semiconductor research corporation and semiconductor industry association, pp 1–21
5. Pan W, Zheng J, Wang L, Luo Y (2022) A future perspective on in-sensor computing. *Engineering*
6. Zhou F, Chai Y (2020) Near-sensor and in-sensor computing. *Nat Electr* 3(11):664–671
7. Ielmini D, Wong H-SP (2018) In-memory computing with resistive switching devices. *Nat Electr* 1(6):333–343
8. Kaneda M (2013) Signal processing in the mammalian retina. *J Nippon Med School* 80(1):16–24
9. Zarándy Á (2011) Focal-plane sensor-processor chips. Springer Science & Business Media, New York City
10. Rodríguez-Vázquez Á, Domínguez-Castro R, Jiménez-Garrido F, Morillas S, Listán J, Alba L, Utrera C, Espejo S, Romay R (2008) The eye-ris cmos vision system. *Analog circuit design*. Springer, New York City, pp 15–32
11. <https://kovilta.fi/technology/>
12. Aistorm, Ai in sensor. Accessed 22 Nov 2021. <https://aistorm.ai/>
13. Zhang J, Dai S, Zhao Y, Zhang J, Huang J (2020) Recent progress in photonic synapses for neuromorphic systems. *Adv Intell Syst* 2(3):1900136
14. Mennel L, Symonowicz J, Wachter S, Polyushkin DK, Molina-Mendoza AJ, Mueller T (2020) Ultrafast machine vision with 2d material neural network image sensors. *Nature* 579(7797):62–66
15. Chen J, Carey SJ, Dudek P (2018) Scamp5d vision system and development framework. In: *Proceedings of the 12th international conference on distributed smart cameras*, pp 1–2
16. Dudek P, Hicks PJ (1999) An simd array of analogue microprocessors for early vision. In: *Proceedings of the conference on postgraduate research in electronics, photonics and related fields (PREP'99)*, pp 359–362
17. Wang Y, Gong Y, Yang L, Xiong Z, Lv Z, Xing X, Zhou Y, Zhang B, Su C, Liao Q et al (2021) Mxene-zno memristor for multimodal in-sensor computing. *Adv Funct Mater* 31(21):2100144
18. Lyapunov N, Zheng XD, Yang K, Liu HM, Zhou K, Cai SH, Ho TL, Suen CH, Yang M, Zhao J et al (2022) A bifunctional memristor enables multiple neuromorphic computing applications. *Adv Electr Mater* 8(7):2101235
19. Lee D, Park M, Baek Y, Bae B, Heo J, Lee K (2022) In-sensor image memorization and encoding via optical neurons for bio-stimulus domain reduction towards visual cognitive processing
20. Pinkham R, Berkovich A, Zhang Z (2021) Near-sensor distributed dnn processing for augmented and virtual reality. *IEEE J Emerg Selected Topics Circuits Syst* 11(4):663–676
21. Sony, Intelligent vision sensors with ai processing functionality. Accessed 14 May 2020. <https://www.sony.com/en/SonyInfo/News/Press/202005/20-037E/>

22. Rodríguez-Vázquez A, Domínguez-Castro R, Jiménez-Garrido F, Morillas S (2010) A cmos vision system on-chip with multicore sensory processing architecture for image analysis above 1,000 f/s. In: *Sensors, cameras, and systems for industrial/scientific applications XI*, vol 7536. Spie, pp 213–223
23. Carey SJ, Lopich A, Barr DR, Wang B, Dudek P (2013) A 100,000 fps vision sensor with embedded 535gops, w 256×256 simd processor array. In: *2013 symposium on VLSI circuits*. IEEE, pp C182–C183
24. Lahdenoja O, Sääntti T, Laiho M, Paasio A, Poikonen JK (2014) Seam tracking with adaptive image capture for fine-tuning of a high power laser welding process. In: *Seventh international conference on machine vision (ICMV 2014)*, vol 9445. SPIE, pp 374–380
25. Poikonen J, Laiho M, Paasio A (2009) Mipa4k: a 64×64 cell mixed-mode image processor array. In: *2009 IEEE international symposium on circuits and systems*. IEEE, pp 1927–1930
26. inivation (2021) Dynamic vision platform. <https://inivation.com/dvp/>
27. Liu Y, Bose L, Greatwood C, Chen J, Fan R, Richardson T, Carey SJ, Dudek P, Mayol-Cuevas W (2021) Agile reactive navigation for a non-holonomic mobile robot using a pixel processor array. *IET Image Process* 1–10
28. Greatwood C, Bose L, Richardson T, Mayol-Cuevas W, Chen J, Carey SJ, Dudek P (2017) Tracking control of a uav with a parallel visual processor. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp 4248–4254
29. Greatwood C, Bose L, Richardson T, Mayol-Cuevas W, Chen J, Carey SJ, Dudek P (2018) Perspective correcting visual odometry for agile mavs using a pixel processor array. In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp 987–994
30. McConville A, Bose L, Clarke R, Mayol-Cuevas W, Chen J, Greatwood C, Carey S, Dudek P, Richardson T (2020) Visual odometry using pixel processor arrays for unmanned aerial systems in gps denied environments. *Front Robot AI* 7
31. Martel JN, Mueller LK, Carey SJ, Dudek P, Wetzstein G (2020) Neural sensors: learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors. *IEEE Trans Pattern Anal Mach Intell* 42(7):1642–1653
32. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W (2017) Visual odometry for pixel processor arrays. In: *Proceedings of the IEEE international conference on computer vision*, pp 4604–4612
33. Chen J, Carey SJ, Dudek P (2017) Feature extraction using a portable vision system. In: *IEEE/RSJ international conference intelligent and robotic systems, Workshop vision-based Agile Auton. Navigation UAVs*
34. Chua LO, Yang L (1988) Cellular neural networks: theory. *IEEE Trans Circuits Syst* 35(10):1257–1272
35. Roska T, Chua LO (1993) The cnn universal machine: an analogic array computer. *IEEE Trans Circuits Syst II: Analog Digit Signal Process* 40(3):163–173
36. Orovas C (2000) Cellular associative neural networks for pattern recognition. PhD dissertation, Citeseer
37. Rosin PL (2006) Training cellular automata for image processing. *IEEE Trans Image Process* 15(7):2076–2087
38. Torralba AB (1999) Analogue architectures for vision cellular neural networks and neuro-morphic circuits. Doctorat thesis, Institute national Polytechnique Grenoble, Laboratory of Images and Signals
39. Dudek P (2019) Scamp-5: vision sensor with pixel parallel simd processor array. <https://youtu.be/D3VcmkQiPR4>
40. Liu Y, Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W (2020) High-speed light-weight cnn inference via strided convolutions on a pixel processor array. In: *The 31st British machine vision conference (BMVC 2020)*
41. Carey SJ, Barr DR, Dudek P (2013) Low power high-performance smart camera system based on scamp vision sensor. *J Syst Archit* 59(10):889–899

42. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W (2019) 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
43. Debrunner T, Saeedi S, Kelly PH (2019) Auke: automatic kernel code generation for an analogue simd focal-plane sensor-processor array. *ACM Trans Archit Code Optim (TACO)* 15(4):1–26
44. Malekzadeh M, Clegg RG, Cavallaro A, Haddadi H (2020) Privacy and utility preserving sensor-data transformations. *Pervasive Mobile Comput* 63:101132. <https://www.sciencedirect.com/science/article/pii/S1574119220300201>
45. Lin W, Liang C, Wang JZ, Buyya R (2014) Bandwidth-aware divisible task scheduling for cloud computing. *Softw: Pract Exp* 44(2):163–174
46. Wang B, Dudek P (2013) Amber: adapting multi-resolution background extractor. In: 2013 IEEE international conference on image processing. IEEE, pp 3417–3421
47. Wang B, Dudek P (2014) A fast self-tuning background subtraction algorithm. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 395–398
48. Dudek P (2006) Adaptive sensing and image processing with a general-purpose pixel-parallel sensor, processor array integrated circuit. In: 2006 International workshop on computer architecture for machine perception and sensing. IEEE, pp 1–6
49. Dudek P, Vilariño DL (2006) A cellular active contours algorithm based on region evolution. In: 2006 10th international workshop on cellular neural networks and their applications. IEEE, pp 1–6
50. Alonso-Montes C, Vilarino D, Penedo M (2005) Cnn-based automatic retinal vascular tree extraction. In: 2005 9th international workshop on cellular neural networks and their applications. IEEE, pp 61–64
51. Wang B, Mrosczyk P, Dudek P (2014) A new method for fast skeletonization of binary images on cellular processor arrays. In: 2014 14th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–2
52. Mrosczyk P, Dudek P (2012) Trigger-wave collision detecting asynchronous cellular logic array for fast image skeletonization. In: 2012 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 2653–2656
53. Razmjooei S, Dudek P (2010) Approximating euclidean distance transform with simple operations in cellular processor arrays. In: 2010 12th international workshop on cellular nanoscale networks and their applications (CNNA 2010). IEEE, pp 1–5
54. Martel JN, Müller LK, Carey SJ, Dudek P (2016) Parallel hdr tone mapping and auto-focus on a cellular processor array vision chip. In: 2016 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1430–1433
55. Martel JN (2019) Unconventional processing with unconventional visual sensing: parallel, distributed and event based vision algorithms & systems. PhD dissertation, ETH Zurich
56. Carey SJ, Barr DR, Wang B, Lopich A, Dudek P (2012) Locating high speed multiple objects using a scamp-5 vision-chip. In: 2012 13th international workshop on cellular nanoscale networks and their applications. IEEE, pp 1–2
57. Barr DR, Carey SJ, Dudek P (2012) Low power multiple object tracking and counting using a scamp cellular processor array. In: 2012 13th international workshop on cellular nanoscale networks and their applications. IEEE, pp 1–2
58. Bose L, Dudek P, Chen J, Carey SJ, Mayol-Cuevas WW (2020) Fully embedding fast convolutional networks on pixel processor arrays. In: European conference on computer vision. Springer, pp 488–503
59. Liu Y, Bose L, Lu Y, Dudek P, Mayol-Cuevas W (2022) On-sensor binarized fully convolutional neural network with a pixel processor array. [arXiv:2202.00836](https://arxiv.org/abs/2202.00836)
60. Chen J, Liu Y, Carey SJ, Dudek P (2020) Proximity estimation using vision features computed on sensor. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE, pp 2689–2695
61. Wong MZ, Guillard B, Murai R, Saeedi S, Kelly PH (2020) Analognet: convolutional neural network inference on analog focal plane sensor processors. [arXiv:2006.01765](https://arxiv.org/abs/2006.01765)

62. Martel JN, Müller LK, Carey SJ, Müller J, Sandamirskaya Y, Dudek P (2017) Real-time depth from focus on a programmable focal plane processor. *IEEE Trans Circuits Syst I: Regul Papers* 65(3):925–934
63. Martel JN, Müller LK, Carey SJ, Dudek P (2017) High-speed depth from focus on a programmable vision chip using a focus tunable lens. In: 2017 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1–4
64. Martel JN, Müller LK, Carey SJ, Müller J, Sandamirskaya Y, Dudek P (2017) Live demonstration: depth from focus on a focal plane processor using a focus tunable liquid lens. In: 2017 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1–1
65. Murai R, Saeedi S, Kelly PH (2020) Bit-vo: visual odometry at 300 fps using binary features from the focal plane. [arXiv:2004.11186](https://arxiv.org/abs/2004.11186)
66. Stow E, Murai R, Saeedi S, Kelly PH (2021) Cain: automatic code generation for simultaneous convolutional kernels on focal-plane sensor-processors. [arXiv:2101.08715](https://arxiv.org/abs/2101.08715)
67. Wang B, Dudek P (2012) 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. IEEE, pp 1–6
68. Martel JN, Sandamirskaya Y, Dudek P (2016) A demonstration of tracking using dynamic neural fields on a programmable vision chip. In: Proceedings of the 10th international conference on distributed smart camera, pp 212–213
69. Dudek P, Carey S (2006) General-purpose 128/spl times/128 simd processor array with integrated image sensor. *Electr Lett* 42(12):678–679
70. Martel JN, Müller LK, Carey SJ, Dudek P (2016) A real-time high dynamic range vision system with tone mapping for automotive applications. In: CNNA 2016; 15th international workshop on cellular nanoscale networks and their applications. VDE, pp 1–2
71. So HM, Martel JN, Wetzstein G, Dudek P (2022) Mantissacam: learning snapshot high-dynamic-range imaging with perceptually-based in-pixel irradiance encoding. In: 2022 IEEE international conference on computational photography (ICCP). IEEE, pp 1–12
72. Dudek P, Hülse M, Barr DR (2008) Cellular automata and non-static image processing for embodied robot systems on a massively parallel processor array. In: Automata-2008: theory and applications of cellular automata. Luniver Press, pp 504–510
73. Davison AJ (2018) Futuremapping: the computational structure of spatial ai systems. [arXiv:1803.11288](https://arxiv.org/abs/1803.11288)
74. Greatwood C, Bose L, Richardson T, Mayol-Cuevas W, Clarke R, Chen J, Carey SJ, Dudek P (2019) Towards drone racing with a pixel processor array. In: 11th international micro air vehicles, conferences and competitions, pp 73–79
75. Castillo-Elizalde H, Liu Y, Bose L, Mayol-Cuevas W (2021) Weighted node mapping and localisation on a pixel processor array. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE
76. Debrunner T, Saeedi S, Bose L, Davison AJ, Kelly PH (2019) Camera tracking on focal-plane sensor-processor arrays. In: Proceedings of the workshop on programmability and architectures for heterogeneous multicores (MULTIPROG), Vancouver, BC, Canada, vol 15
77. Klein G, Murray D (2007) Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, pp 225–234
78. Liu Y, Bose L, Chen J, Fan R, Dudek P, Mayol-Cuevas W (2021) On-sensor cnn parallel computing with a pixel processor array. *IEEE Trans Comput Imaging*, manuscript
79. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W (2019) A camera that cnns: towards embedded neural networks on pixel processor arrays. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV)
80. Courbariaux M, Bengio Y, David J-P (2015) Binaryconnect: training deep neural networks with binary weights during propagations. In: Advances in neural information processing systems, pp 3123–3131
81. Liu Y, Chen J, Bose L, Dudek P, Mayol-Cuevas W (2021) Direct servo control from in-sensor cnn inference with a pixel processor array. In: 2021 IEEE international conference on robotics and automation (ICRA) Workshop: on and near-sensor vision processing, from photons to applications. IEEE

82. Liu Y, Chen J, Bose L, Dudek P, Mayol-Cuevas W (2021) Bringing a robot simulator to the scamp vision system. In: 2021 IEEE international conference on robotics and automation (ICRA) workshop: on and near-sensor vision processing, from photons to applications. IEEE
83. Guillard B (2019) Optimising convolutional neural networks for super fast inference on focal-plane sensor-processor arrays. PhD dissertation, Imperial College London
84. Wong M, Saeedi S, Kelly PH (2018) Analog vision-neural network inference acceleration using analog simd computation in the focal plane. PhD dissertation, Master's thesis, Imperial College London-Department of Computing
85. Liu Y (2022) On-sensor visual inference with a pixel processor array. PhD dissertation, University of Bristol
86. Wolfram S (1984) Cellular automata as models of complexity. *Nature* 311(5985):419–424
87. Mordvintsev A, Randazzo E, Niklasson E, Levin M (2020) Growing neural cellular automata. *Distill* 5(2):e23
88. Eye-RIS, Eye-ris v1.3 hardware description. Accessed 16 Sept 2022. <https://imaging.teledyne-e2v.com/products/2d-cmos-image-sensors/eye-ris/>
89. Sotoa LA, Morillas S, Listána J, Jiménez A, Arenab P, Patané L, De Fiore S, Embedding the anafocus' eye-ris vision system in roving robots to enhance the action-oriented perception
90. Caballero-García D, Jimenez-Marrufo A (2014) Visual routines for cognitive systems on the eye-ris platform. In: Spatial temporal patterns for action-oriented perception in roving robots II. Springer, pp 249–316
91. Soto LA, Morillas S, Listán J, Jiménez A, Arena P, Patané L, De Fiore S (2009) Embedding the anafocus' eye-ris vision system in roving robots to enhance the action-oriented perception. In: Bioengineered and bioinspired systems IV, vol 7365. SPIE, pp 80–90
92. Guzmán P, Díaz J, Agís R, Ros E (2010) Optical flow in a smart sensor based on hybrid analog-digital architecture. *Sensors* 10(4):2975–2994
93. Lucas BD, Kanade T et al (1981) An iterative image registration technique with an application to stereo vision. Vancouver, vol 81
94. Zarándy Á, Pencz B, Németh M, Zsedrovits T (2014) Implementation of visual navigation algorithms on the eye-ris 1.3 system. In: 2014 14th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–2
95. Nicolosi L, Abt F, Tetzlaff R, Hofler H, Blug A, Carl D (2009) New cnn based algorithms for the full penetration hole extraction in laser welding processes. In: 2009 IEEE international symposium on circuits and systems. IEEE, pp 2713–2716
96. Abt F, Heider A, Weber R, Graf T, Blug A, Carl D, Höfler H, Nicolosi L, Tetzlaff R (2011) Camera based closed loop control for partial penetration welding of overlap joints. *Phys Procedia* 12:730–738
97. Fülöp T, Zarándy Á (2010) Bio-inspired looming object detector algorithm on the eye-ris focal plane-processor system. In: 2010 12th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–5
98. Sántti T, Lahdenoja O, Paasio A, Laiho M, Poikonen J (2014) Line detection on fpga with parallel sensor-level segmentation. In: 2014 14th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–2
99. Lahdenoja O, Sántti T, Poikonen J, Laiho M, Paasio A (2013) Characterizing spatters in laser welding of thick steel using motion flow analysis. In: Scandinavian conference on image analysis. Springer, pp 675–686
100. Sántti T, Poikonen JK, Lahdenoja O, Laiho M, Paasio A (2015) Online seam tracking for laser welding with a vision chip and fpga enabled camera system. In: 2015 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1985–1988
101. Lahdenoja O, Sántti T, Laiho M, Poikonen J (2014) Spatter tracking in laser-and manual arc welding with sensor-level pre-processing
102. Poikonen J, Laiho M, Paasio A (2010) Anisotropic filtering with a resistive fuse network on the mipa4k processor array. In: 2010 12th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–5

103. Lahdenoja O, Poikonen J, Laiho M (2010) Extracting local binary patterns with mipa4k vision processor. In: 2010 12th international workshop on cellular nanoscale networks and their applications (CNNA). IEEE, pp 1–5
104. Poikonen J, Laiho M, Paasio A (2009) Locally adaptive image sensing with the 64x64 cell mipa4k mixed-mode image processor array. In: 2009 European conference on circuit theory and design. IEEE, pp 93–96
105. Laiho M, Poikonen J, Paasio A (2009) Space-dependent binary image processing within a 64x64 mixed-mode array processor. In: 2009 European conference on circuit theory and design. IEEE, pp 189–192
106. Laiho M, Poikonen J, Paasio A (2010) Object segmentation and tracking with asynchronous grayscale and binary wave operations on the mipa4k. In: 2010 12th international workshop on cellular nanoscale Networks and their applications (CNNA). IEEE, pp 1–4
107. Linan G, Espejo S, Dominguez-Castro R, Rodriguez-Vázquez A (2002) Architectural and basic circuit considerations for a flexible 128×128 mixed-signal simd vision chip. *Analog Integr Signal Process* 33(2):179–190
108. Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang JJ, Qian H (2020) Fully hardware-implemented memristor convolutional neural network. *Nature* 577(7792):641–646
109. Wang T-Y, Meng J-L, Li Q-X, He Z-Y, Zhu H, Ji L, Sun Q-Q, Chen L, Zhang DW (2021) Reconfigurable optoelectronic memristor for in-sensor computing applications. *Nano Energy* 89:106291
110. Sun L, Wang Z, Jiang J, Kim Y, Joo B, Zheng S, Lee S, Yu WJ, Kong B-S, Yang H (2021) In-sensor reservoir computing for language learning via two-dimensional memristors. *Sci Adv* 7(20):eabg1455
111. Thomas A (2013) Memristor-based neural networks. *J Phys D: Appl Phys* 46(9):093001
112. Lungu IA, Liu S-C, Delbruck T (2019) Fast event-driven incremental learning of hand symbols. In: 2019 IEEE international conference on artificial intelligence circuits and systems (AICAS). IEEE, pp 25–28
113. Lungu I-A, Corradi F, Delbrück T (2017) Live demonstration: convolutional neural network driven by dynamic vision sensor playing roshambo. In: 2017 IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1–1
114. Linares-Barranco A, Rios-Navarro A, Tapiador-Morales R, Delbruck T (2019) Dynamic vision sensor integration on fpga-based cnn accelerators for high-speed visual classification. [arXiv:1905.07419](https://arxiv.org/abs/1905.07419)
115. Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, Censi A, Leutenegger S, Davison A, Conrath J, Daniilidis K et al (2019) Event-based vision: a survey. [arXiv:1904.08405](https://arxiv.org/abs/1904.08405)
116. Song R, Huang K, Wang Z, Shen H (2021) An ultra fast low power convolutional neural network image sensor with pixel-level computing. [arXiv:2101.03308](https://arxiv.org/abs/2101.03308)
117. Datta G, Kundu S, Yin Z, Lakkireddy RT, Beerel PA, Jacob AP, Jaiswal A (2022) P2m: a processing-in-pixel-in-memory paradigm for resource-constrained tinyml applications. [arXiv:abs/2203.04737](https://arxiv.org/abs/2203.04737)