Intelligent Computing
A SCIENCE PARTNER JOURNAL

REVIEW ARTICLE

In-Sensor Visual Perception and Inference

Yanan Liu[1], Rui Fan[2], Jianglong Guo[3*], Hepeng Ni[4*], and Muhammad Usman Maqboo Bhutta[5]

[1]School of Microelectronics, Shanghai University, Shanghai,China. [2]The College of Electronics and Information Engineering, the State Key Laboratory of Intelligent Autonomous Systems, and Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai, China. [3]School of Science, Harbin Institute of Technology, Shenzhen, China. [4]School of Mechanical and Electrical Engineering, Shandong Jianzhu University, Jinan, China. [5]Robotics Lab, The Chinese University of Hong Kong, Hong Kong, China.

*Address correspondence to: guojianglong@hit.edu.cn (J.G.); nihepeng20@sdjzu.edu.cn (H.N.)

Citation: Liu Y, Fan R, Guo J, Ni H, Bhutta MUM. In-Sensor Visual Perception and Inference. *Intell. Comput.* 2023;2:Article 0043. https://doi.org/10.34133/icomputing.0043

Submitted 16 September 2022
Accepted 3 June 2023
Published 26 September 2023

Conventional machine vision systems have separate perception, memory, and processing architectures, which may exacerbate the increasing need for ultrahigh image processing rates and ultralow power consumption. In contrast, in-sensor visual computing performs signal processing at the pixel level using the collected analog signals directly, without sending data to other processors. Therefore, the in-sensor computing paradigm may hold the key to realizing extremely efficient and low power visual signal processing by integrating sensing, storage, and computation onto focal planes using either novel circuit designs or new materials. The focal-plane sensor-processor (FPSP), which is a typical in-sensor visual computing device, is a vision chip that has been developed for nearly 2 decades in domains such as image processing, computer vision, robotics, and neural networks. In contrast to conventional computer vision systems, the FPSP gives vision systems in-sensor image processing capabilities, thus decreasing system complexity, reducing power consumption, and enhancing information processing efficiency and security. Although many studies on in-sensor computing using the FPSP have been conducted since its invention, no thorough and systematic summary of these studies exists. This review explains the use of image processing algorithms, neural networks, and applications of in-sensor computing in the fields of machine vision and robotics. The objective is to assist future developers, researchers, and users of unconventional visual sensors in understanding in-sensor computing and associated applications.

Introduction

Vision is one of the most important perception methods and is extremely useful for information collection and interpretation [1]. It is highly desirable to develop ultrahigh-speed and ultralow-energy visual information processing methods and technologies for applications in machine vision, robotics, the Internet of Things (IoT), and artificial intelligence (AI). System latency, power consumption, and privacy issues are 3 major constraints that may hinder the further development and wider applications of conventional machine vision systems and their associated technologies [2,3]. In contrast to the mammalian retinal mechanism, wherein raw signals can be rapidly processed through several layers of cells (Fig. 1A), a considerable time lag can be introduced during visual signal digitization, storage, and transmission processes in a conventional machine vision system. Latency is a bottleneck that prevents quick responses to dynamic changes, resulting in substantial inefficiencies as irrelevant data are ferried through the entire system. In addition, the use of external image processors, such as CPU/GPU/VPU/DSPs, tends to result in a higher power consumption, which is not compatible with portable tasks (Fig. 1B). Moreover, the data deluge that results from ubiquitous sensors may obscure useful information, thereby encouraging terminal sensors to extract only a limited amount of critical information [4]. Thus, a substantial amount of data movement from the sensing chip to the processing units is reduced [5]. Furthermore, there is a considerable need to extract crucial information from raw analog signals as opposed to collected images, particularly in privacy-sensitive scenarios.

To overcome these issues, data processing should occur as close as possible to the time of signal collection using the paradigm of in-sensor computing [6]. Bioinspired by the mammalian retina (Fig. 1A), the role of the vision sensor in this approach is to acquire visual information and to digest it, producing highly compressed information instead of video frames (Fig. 1C). Additionally, in-sensor visual computing offers image-free visual signal processing, which ensures data confidentiality. In-sensor computing is an interdisciplinary research area closely related to existing technologies, including sensors, analog signal processing, near-sensor computing, and in-memory computing (Fig. S1). In-sensor computing devices integrate perception, temporary storage, data processing, and analysis with raw analog signals within a sensing chip. Although near-sensor computing reduces the physical distance between sensing and computing, data movement from sensors to processing remains necessary. Memristors are typically used for both memory and computing [7] by utilizing tunable resistances as synaptic weights. The objective of "in-sensor" computing [6,8] is to sense, extract, analyze, store, and compute sensory signals in an in situ manner within sensors. In contrast to conventional digital electronics-based sensor technology, which mainly focuses on data collection with external data processing, in-sensor signal processing emphasizes sensing, signal storage, and the preprocess, where the analog signal is collected. By integrating sensing, storage, and computing on the sensing plane of the sensory chip, low power and efficient edge computing is enabled for the embedded system, which is meaningful in the area of the IoT. Avoiding the need to send redundant information to the cloud reduces the pressure on the central computation and data transmission bandwidth. The idea of in-sensor visual computing is bioinspired by the mammalian brain and visual system, where the retina preprocesses visual information and then sends extracted signals to the brain through optical nerves [9]. Currently, this in-sensor visual computing technology can be applied to many emerging hardware systems based on advanced large-scale circuit designs such as SCAMP vision systems, photodiode arrays [10,11], event cameras [12], and memristors [13–15].

Recently, progress has been made in the development of in-sensor computing devices. Currently, 2 main types of in-sensor computing architecture are available:

(a) In-sensor architecture by integrating sensing, memory, and computing units: A focal-plane sensor-processor (FPSP) [16] integrates visual sensing, storage, and computing units on the focal plane under the architecture of a cellular neural network

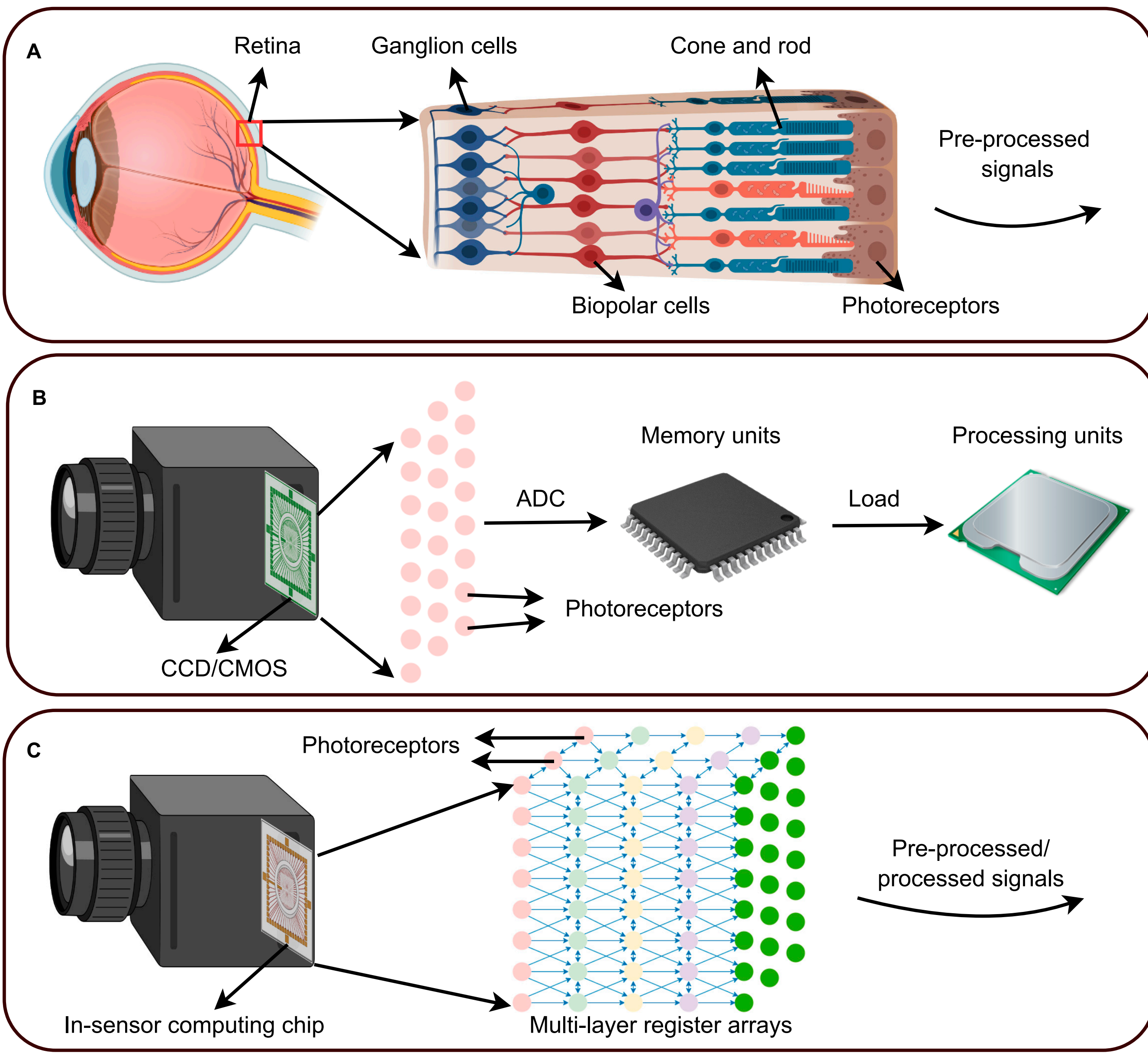

Fig. 1. The origin of in-sensor visual computing. (A) The concept of in-sensor computing is bioinspired by the retina mechanism, wherein visual signals can be generated and pre-processed by different types of cells [108]. (B) Conventional machine vision system: Light density needs to be read and converted to digital data before being loaded into memory and processing units for meaningful information extraction. (C) Visual data can be generated, stored, and processed in sensors through the bioinspired hardware design.

(CeNN) (Fig. 2A). For each processing element (PE), the generated analog signals from the pixel can be transferred to temporal memory units through the bus and then processed using the arithmetic logic unit (ALU) and registers. Each PE plays the role of a cell, interacting with its neighbors for signal exchange and processing. Hence, in-sensor visual inference is realized by the hardware CeNN and its synaptic weights in memory. The representative devices under the focal-plane sensor-processor (FPFS) architecture mainly include the SCAMP pixel processor array (PPA), Q-Eye [17], MIPA4k [16], asynchronous–synchronous focal-plane sensor-processor array (ASPA) [16], kovilta [18], and Aistorm Mantis2 [19].

(b) Detect-and-memorize materials for in-sensor computing architectures: Material-based detect-and-memorize (DAM) devices (Fig. 2B) have recently been proposed to mimic the functional mechanism of photonic synapses for implementing artificial neural networks [5,20]. Among emerging materials and devices, memristors are representative because they facilitate sensing, temporal memory, and computing capabilities when combined with other photosensitive devices [10]. Specifically, visual signals generated from photoreceptors such as photodiodes can be further processed within artificial networks composed of memristors with tunable resistances as weights. Table S1 lists the differences between the 2 rising in-sensor computing architectures.

Among the new types of emerging sensors, the SCAMP PPA [21–25] is comparatively mature in terms of its development history, sensing resolution, and practicality. Therefore, this review mainly concentrates on studies of the SCAMP PPA with other relevant devices as the related work. The SCAMP PPA is a visual device that operates directly on the current, enabling focal-plane image processing with useful results as an output from raw sensor data. The motivation is to design a fully programmable general-purpose single-instruction multiple-data (SIMD) cellular processor array for a novel world–machine interface that can sense, store, and reason without relying on external centralized processing units [26,27].

Using an all-in-sensor scheme, a device can perceive its environment and efficiently generate useful results with reduced energy consumption, latency, and data bandwidth [5,6]. Considering these advantages of the emerging SCAMP PPA, this paper reviews the research progress with in-sensor processing technology using SCAMP PPAs, aiming to introduce a new visual inference solution that can overcome the aforementioned bottlenecks. Specifically, this study investigated image processing algorithms and neural networks, covering everything from low-level image processing methods to high-level image inference, and their associated applications. The work most closely related to our review is [28], which has

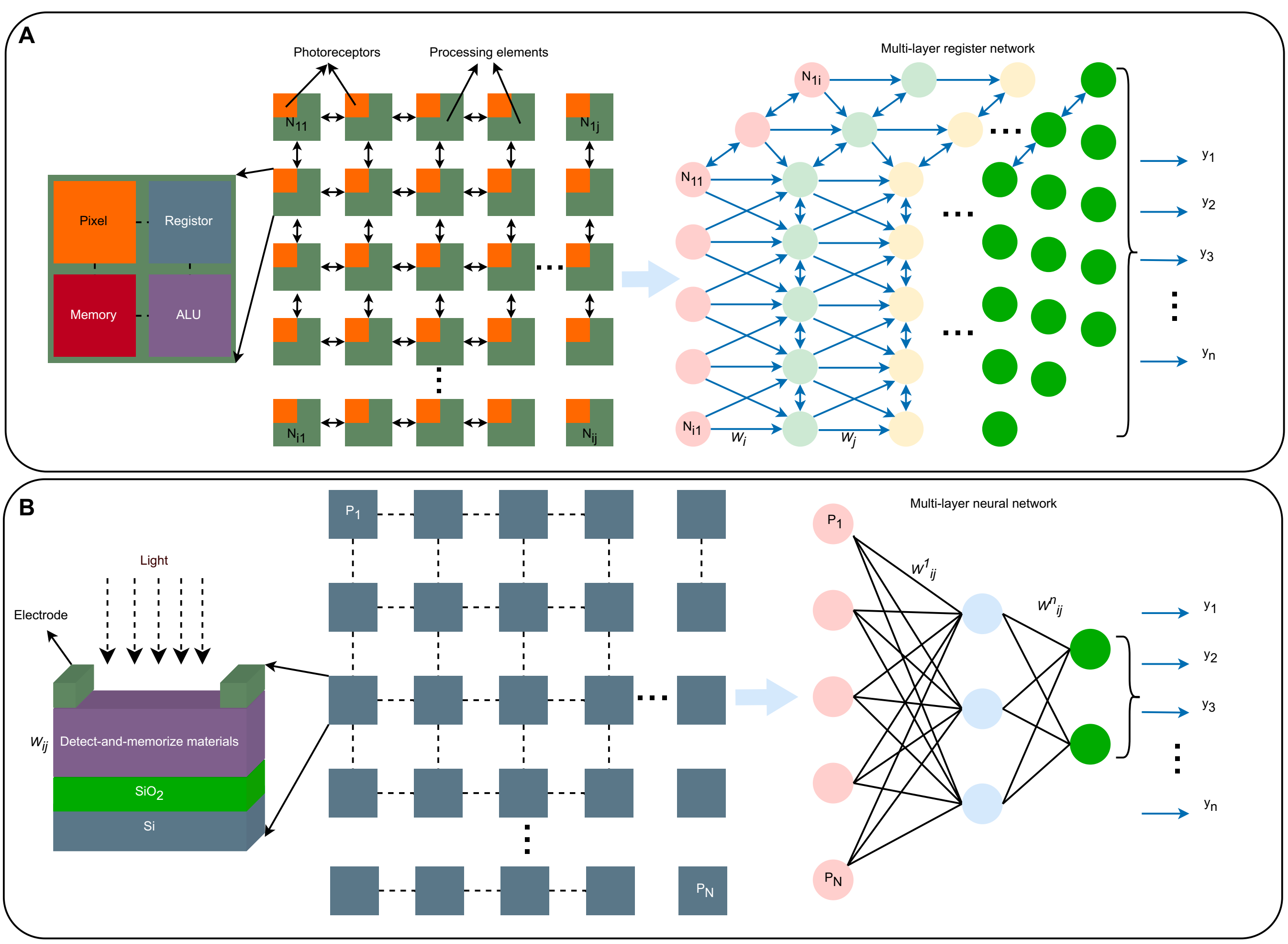

Fig. 2. In-sensor perception and computing architectures and their associated artificial networks. (A) An in-sensor cellular network can be created with an array of PEs that integrates sensing, memory, and computing units. (B) A neural network with DAM materials.

a specific focus on integrated circuit design and robot-oriented applications (Table S3).

Hence, this study reviews in-sensor computing algorithms and their applications developed on the SCAMP PPA and compares them with conventional visual sensors and other types of in-sensor computing visual sensors, inspiring researchers and developers researching unconventional computing with emerging visual sensors. The sections of this review are hardware, software, development frameworks, and applications, as shown in Fig. S2. In particular, the “Introduction” section provides an overview of in-sensor visual computing topics. The “Hardware” section summarizes the current devices in the scope of sensory-level computing with their advantages over conventional visual sensors. Associated platforms and frameworks are introduced in the “Software” section, followed by the “Algorithms” section covering low-level to high-level in-sensor visual information processing algorithms. By leveraging the established hardware and software infrastructure, the “Applications” section illustrates the applications of our research in state estimation and robotics. The “Challenges and Future Trends” and “Conclusion” sections present the existing challenges with potential future routes and summaries, respectively. The provided supplementary materials include comparison tables.

Hardware

Focal-plane sensor-processor

A representative FPSP device, the SCAMP vision system is an emerging in-sensor visual computing device. Currently, the most recent version of the SCAMP series system is SCAMP-5d (Fig. 3), which comprises 256 × 256 PEs. The SCAMP-5d vision system is a general-purpose, programmable, massively parallel vision system [29] that has many applications in the fields of robotics [31–34] and computer vision [30–32]. For the PPA shown in Fig. 1C, the photodetector converts light into an analog signal that can be directly and parallelly processed on analog registers (AREGs). In contrast to the current hardware design structure of computer vision systems, the

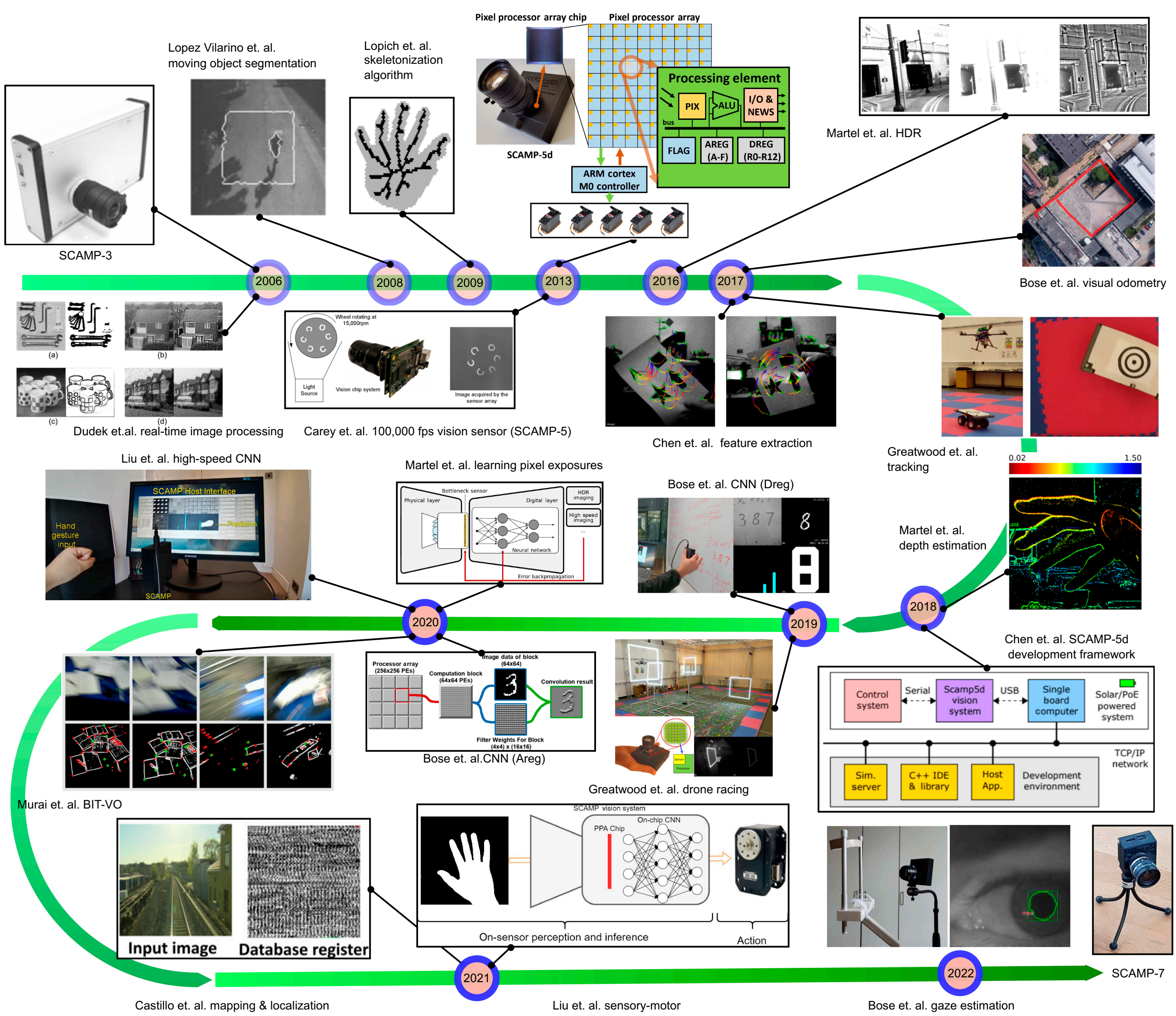

Fig. 3. Milestones of key studies with SCAMP PPAs during the last 15 years from low-level image processing to high-level pattern recognition and motion control.

PPA bypasses the analog/digital conversion (ADC) after sensing and directly operates on the analog electric current using an arithmetic unit, thereby accelerating the signal processing speed and avoiding the bottlenecks of the ADC and the data transmission process.

The PPA is a hardware implementation of a CeNN with a new optimization of a mixture of analog and digital computing using AREGs and digital registers (DREGs), respectively. The studies based on the PPA reviewed in this work utilize the parallel nature of the CeNN architecture for efficient and high-performance computing, where each "cell" is intricately connected with its 4 neighbors and information can be shared efficiently. Hence, the PPA can be modeled as a CeNN architecture for visual information computing. The CeNN processing circuit architecture was first proposed by Chua and Yang [33] and was followed by the CeNN universal machine [34] as a prototype. Subsequently, as a new circuit architecture and parallel computing paradigm, it has gained widespread popularity in academia, serving as the basis of a substantial number of research outputs and applications in pattern recognition [35], image processing [36], and biological vision modeling [37]. With the aforementioned hardware features, the SCAMP PPA has the following advantages over conventional machine systems:

Efficiency and low latency: It is clear from Fig. 1C that in-sensor computing bypasses signal digitization, transmission, and storage processes onto external devices, hence enabling high-speed image processing [29] and convolutional neural network (CNN) inference [38], which can be integrated with agile mobile robot platforms [39–42]. In addition, PE distribution and simultaneous instruction execution on the PEs enable efficient parallel signal processing. Carey et al. [29] demonstrated object detection with a frame rate of 100,000 frames per second (FPS) using the SCAMP vision system, and Liu et al. [38] proposed a binary shallow neural network on the PPA with a binary classification problem of up to 17,000 FPS. This work demonstrates the efficiency of image processing in a sensor once the parallelism mechanism of the PPA is fully exploited.

Low power consumption: According to Fig. 1C, no external processing units or data processing is needed; hence, the power consumption can be decreased substantially. The maximum power cost of the SCAMP-3 vision system for complex object tracking and counting is 29 mW [43]. The overall power consumption for the image processing and CNN inference tasks within a SCAMP-5 vision system is less than 2 W [44]. This feature qualifies the SCAMP vision system for use on mobile platforms, which typically have a short battery life. In addition, according to the power consumption test from [45], given 8 popular kernel filters, the SCAMP PPA generates the same convolution results with considerably lower power consumption (>20 times) at a higher speed (>100 times) compared with common CPUs and GPUs.

Data security and privacy protection: A unique but nonnegligible feature of in-sensor analog computing with the PPA is its inherent feature of data security and privacy protection. Data security is feasible because of the focal-plane analog information processing without an ADC, extra data recording, storage, or intermediate transmission procedures. Generally, the only output after analog computing is the extracted useful target information without redundant information, which renders it difficult to obtain the original data for sensitive information or user re-identification [46]. Hence, privacy can be strictly protected using in-sensor processing mechanisms.

In-sensor computing devices

Conventional sensors act primarily as information collectors. In recent years, with the development of techniques for integrated circuit design and the growing need for low power and low latency edge computing, sensors have gradually become integrated with the ability to process signals independent of general-purpose computers. The goal of near-sensor processing is to use a dedicated machine learning accelerator chip located on the same printed circuit board [47], or 3-dimensional (3D)-stacked with the complementary metal-oxide semiconductor (CMOS) image chip [48]. Although this enables CMOS image chip data to be processed close to the sensor rather than in the cloud, data transport expenses between the sensing and processing chips still exist. By 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 S2 lists the differences between in-sensor computing devices.

Aistorm Mantis2 [19]: The Mantis system is based on the event-driven charge domain for analog signal processing without digitization and provides an "always on" solution for analog signal processing. A key feature claimed by Aistorm is noise canceling techniques associated with analog signals. In addition, AI can be integrated into chips for various applications. However, the latest Mantis product has a resolution of only 96 × 96, which is challenging for tasks that normally require higher resolution.

Eye-RIS [49]: The Eye-RIS commercial vision system on chip extends CMOS pixel functionality with image storage (7 grayscale images and 4 binary images) and digital/analog signal processing ability. Specifically, a 32-bit reduced instruction set computer (RISC) is integrated with a vision sensor for image postprocessing after parallel in-sensor preprocessing. The resolution of the Eye-RIS vision sensor is 176 × 144. Notably, Eye-RIS's overall functional diagram is similar to that of the SCAMP vision system, where the counterpart of the RISC is the M0 microcontroller in the SCAMP PPA [50]. However, the most significant difference is that the Eye-RIS has a digital image co-processor (DICop) that handles geometric transformations and can send the results back to the pixel level for further processing.

Memristor-based devices [51]: Memristor-based hardware provides platforms for deploying neural networks using the programmable resistance within the integrated circuits to mimic the synaptic connections in a human brain [13–15,51,52]. However, it only integrates storage and processing functions, which can be regarded as in-memory computing. Hence, signals must be input from sensors or other storage devices. Therefore, these systems are typically integrated with other sensory systems for information processing.

Dynamic vision sensor [12]: A dynamic vision sensor (DVS) produces data in the form of sparse contrast-change events that facilitate low latency visual processing using external computational hardware [53–55]. These binary events are generated by in-sensor processing according to brightness changes. Although the pixels in a DVS have a primitive in-sensor processing ability that works by binarizing brightness changes, they achieve an ultra-high-speed response to the environment while working with external hardware computing units, enabling enormous potential for robotics and computer vision in challenging environments [56].

Other emerging sensor devices: Mennel et al. [10] used a 2D semiconductor (WSe_2) photodiode array as the vision sensor, with the photoresponsivity matrix storing the connecting weights of the neural network and both supervised and unsupervised learning for classification. However, laser light and a set of optical systems are required to project images onto a chip, which prevents its practical usage. Song et al. [11] proposed a CMOS-based processing-in-pixel (PIP) architecture, where image convolution (8-bit weight configuration) can be run for image preprocessing before the image data are read. In addition, Datta et al. [57] proposed the processing-in-pixel-in-memory paradigm, in which the first few convolutional layers of a CNN can be processed, and the compressed data are then sent to other near-sensor processors.

Software

For emerging vision sensors, providing user-friendly development and simulation tools is essential for researchers and engineers for idea exploration and application validation. Hence, this section introduces the platforms and frameworks of the SCAMP PPA for efficient prototyping.

Development framework

Currently, comprehensive development tools exist for this emerging in-sensor computing device that can aid researchers

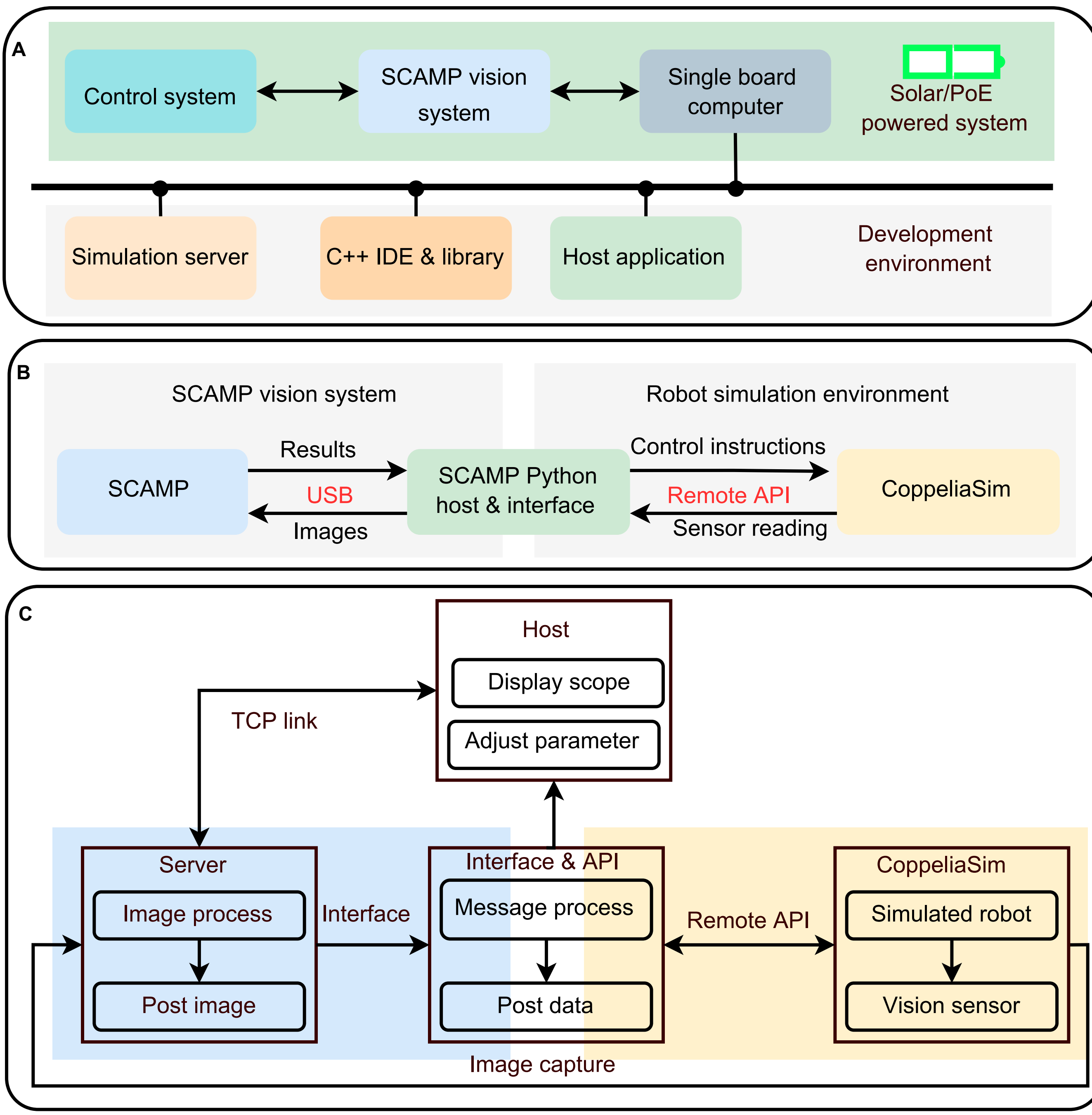

Fig. 4. Development frameworks for the SCAMP vision system. (A) Development framework for the SCAMP vision system. (B) Semi-simulation framework of the SCAMP-5d vision system and virtual environment. (C) Full-simulation framework of the SCAMP-5d vision system for robot applications.

and engineers in developing their projects. Chen et al. [58] developed guidelines and documents, libraries (low- and mid-level basic image processing algorithms), simulations, and development environment configurations (https://scamp.gitlab.io/scamp5d_doc/) (Fig. 4A). This system, called Scamp5d, utilizes a SCAMP-5 vision chip, which is equipped with sensor-level SIMD parallel processing capabilities. A dual-core ARM microcontroller was employed to manage the vision chip, offer supplementary computational abilities, and provide input/output (IO) interfaces. The vision system was programmed using C++. Scamp5d can be connected to microcontrollers, single-board computers, or other hardware using common IO buses and a USB 2.0 port. It also enables remote debugging, configuration, and reprogramming of the network. The software interface of Scamp5d is openly designed for easy integration with other software systems such as the Robot Operating System. As a standalone vision system, Scamp5d can output highly processed data instead of video streams, rendering it appropriate for applications such as miniature robots and distributed sensor networks. In addition, simulation of the vision chip can be executed through cross-compiling.

Similar to the SCAMP vision system development software framework, the KOVILTA vision system [18] software development framework is a modular and scalable framework designed to be easy to use and extend. The framework is based on a set of core components that provide the basic functions of a vision system, such as image acquisition, image processing, and object detection. The framework also provides a set of extension points that enable users to add new features and functionalities to the vision system. The KOVILTA vision system software development framework is divided into 2 main layers: the core layer and the application layer. The core layer provides the basic functions of the vision system, including image acquisition, image processing, and object detection. The application layer provides the user interface and functions specific to the application. The core layer of the KOVILTA vision system software development framework is implemented in C++. It comprises a set of libraries that provide the basic functions of the vision system. These libraries include the following: an image acquisition library that provides functionality for acquiring images from cameras; an image processing library that provides functionality for image processing tasks such as resizing, cropping, and filtering; and an object detection library that provides functionality for detecting objects in images. The application layer of the KOVILTA vision system software development framework is implemented in Python. The application layer comprises a set of scripts that provide the user interface and functionality specific to the application.

Semi-simulated and fully simulated platform

To further explore robot-related applications with a SCAMP vision system, Liu et al. [59] proposed a semi-simulated platform (Fig. 4B), wherein a real SCAMP can communicate with the CoppeliaSim robot simulator through a remote application programming interface (API). Using this platform, customers can directly process the camera readings after they are transmitted to the real SCAMP, and then send them back to the instructions generated by in-sensor computing to the entities in the simulator. Based on an earlier semi-simulated platform, Fan et al. [60] (Fig. 4C) developed a fully simulated environment integrating the SCAMP server development environment and the CoppeliaSim robot simulator, wherein the simulated SCAMP vision system and the robot simulator can communicate bidirectionally through a remote API.

Kernel filter compiler

Image convolution is necessary for mid-level image processing or convolutional neural networks. In addition to the aforementioned convolution with binary/ternary weights, approximated kernel filters for convolution are proposed by [45] with automatic code generation. Each full-precision coefficient in the kernel filters is approximated using a combination of multiple additions/subtractions and divisions (Eq. 1).

$$\alpha \approx \sum_{i=0}^{n} a_i / 2^i, \qquad a_i \in \{-1, 0, 1\}, \tag{1}$$

where n denotes the approximation depth. This work provides an effective way to approximate full-precision kernel filters and automatically generate codes for SCAMP PPA hardware. Using a similar coefficient approximation strategy, Stow et al. [61–63] automatically generated codes for the PPA, which outperformed the earlier AUKE in terms of simultaneous kernel optimization and generated more efficient codes.

Algorithms

In-sensor low- and mid-level image processing

Early algorithms for the SCAMP PPA mainly focused on low-level image processing and machine vision methods to enhance image quality and extract basic textures with combinations of inherent built-in functions based on SCAMP-3, with a resolution of 128 × 128. Specifically, these developed image processing methods are closely related to the cellular architecture of the SCAMP vision system because the PPA is a cellular processor array. Wang [64] contributed to early explorations of image processing on the PPA covering coarse grain processing, image skeleton extraction, and background detection algorithms.

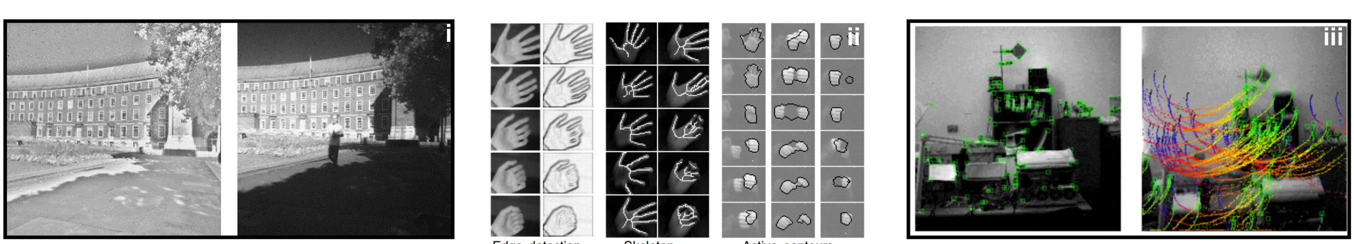

Fig. 5. Overview of SCAMP PPA-based algorithms and applications. In-sensor image processing. (i) High dynamic range (HDR). (ii) Edge, skeleton, and contour detection. (iii) Corner detection.

Barr et al. [65] presented tracking of multiple randomly moving objects using SCAMP-3, and Carey et al. [66] performed counting of 5 preset objects at 25,000 FPS and single-object tracking at 100,000 FPS using the SCAMP-5 vision system. Table S4 lists the main research on the PPA, which is illustrated in detail in the following context.

High dynamic range

Image enhancement occurs along with the image capture process on the SCAMP PPA, whereas conventional image enhancement only occurs after the image data have been obtained. As an important image enhancement method, high dynamic range (HDR) has been fully exploited in sensors with PPA.

HDR is an important low-level image preprocessing method for obtaining rich image information, even under extreme lighting conditions, such as a mixture of both dim and strong light intensities. Conventional image sensors rely on either global or rolling shutters to generate a sequence of images with different exposure time settings and then combine them into an HDR image, which means that both the exposure and image capture processes limit the efficiency of HDR imaging [30,67,68]. In 2006, Dudek [69] proposed sensor-level adaptive sensing and image processing using SCAMP-3 [43,70], wherein different exposure settings were combined for an image with a wide dynamic range. Inspired by Dudek's work, Martel [68] significantly contributed to the field of HDR images using the SCAMP-5 vision system (Fig. 5, i). An example of in-sensor HDR image generation is [71], wherein pixel-wise exposure can be conducted under various lighting conditions to generate HDR images, followed by automotive applications [72]. Furthermore, Bose et al. [31] utilized HDR images to extract binary edges as robust input information under various illumination conditions for visual odometry estimation. However, the use of iterative exposure for different regions of the image slows image preprocessing. To accelerate HDR imaging, Martel et al. [30] proposed learning shutter functions to expose each pixel independently using an end-to-end training strategy. In their study, a U-Net was trained for the exposure functions of each individual PE sensor, and these trained functions were compiled into a sensor for inference. So et al. [73] further demonstrated a new codesign of in-sensor irradiance encoding and decoding for snapshot HDR imaging, which could enable more applications such as photography and adaptive machine vision under varying light conditions.

Contour and skeleton extraction

Contour is an important feature of objects within an image that can help identify different entities. The initial contour extraction algorithms were proposed based on pixel-level snakes with exceptionally low latency in [74]. Subsequently, in 2007, Alonso-Montes et al. [75] proposed an in-sensor automatic retinal vessel tree extraction method based on a CeNN. The shared key methods in [69,74,75] iteratively extract contours based on the active contour model and CeNN. In 2008, Dudek et al. [76] proposed an image preprocessing method based on cellular automata for robotic scenarios. The skeleton within the binary image shows the size, position, and simplified shape of the object. Fast image skeletonization [77] was implemented in [78] based on wave-trigger propagation/collision. Razmjooei and Dudek [79] proposed an approximate Euclidean distance transform that uses simple and efficient shifting operations. Examples of contour and skeleton extraction are shown in Fig. 5 (ii).

Local visual feature detection

Other image processing methods such as background extraction have been exploited by Wang and Dudek [80,81]. For higher-level feature extraction, edge features can be obtained by deploying Sobel kernel filters or Laplacian filters, which were used in later works on focal-plane visual odometry [31] and neural networks [82]. As for other features, such as corner point extraction (Fig. 5, iii), Chen [32] utilized DREGs based on the Fast16 algorithms, which were used in subsequent visual odometry work [83]. Other methods have been exploited for different image processing tasks. Wang and Dudek [84] proposed a simple coarse-grain mapping method to process higher resolution than the PPA resolution by storing sub-images in different registers. Furthermore, to alleviate the problem of scarce hardware resources on the PPA, Martel et al. [85] proposed algorithms to trade off the PE resolution and registers by grouping several pixels into "super-pixels," enabling them to run more complicated algorithms.

Based on the aforementioned low- and mid-level image processing methods, researchers are motivated to exploit more general high-level image processing by leveraging state-of-the-art innovations in the field of computer vision, such as neural networks.

In-sensor neural network computing

High-level image processing tasks such as object classification, localization, and segmentation are mainly implemented using in-sensor neural network inference. Table S5 lists and compares existing neural networks with the SCAMP vision system.

Convolutional neural network for classification

The deployment of a neural network on the PPA is a breakthrough because it enables the PPA to be open to more possibilities with advanced general methods. With the use of CNNs, several types of tasks, such as classification, regression, localization, and segmentation, which typically rely on powerful GPUs/CPUs, can be feasible.

Research on CNN implementation and inference within the PPA was pioneered by Bose et al. [44,86], in which a CNN with a single convolutional layer was implemented on the PPA array and a fully connected layer was implemented on its controller chip. This work performs 16-bit image convolution operations using 4×4 DREG "Super Pixel" blocks and demonstrates live digit classification using the MNIST dataset at approximately 200 FPS. In their work, ternary $\{-1, 0, 1\}$ kernel filters were stored on the flash memory of the PPA system and effectively encoded in the instructions/operations sent to the PPA array, performing convolutions sequentially. To fully exploit the PPA's parallel computing characteristics and further improve CNN inference efficiency, Bose et al. [87] proposed the idea of in-pixel weight storage, in which the network weights are directly stored within the registers of the PPA's PEs. This enabled both the parallel computation of multiple convolutions and the implementation of a fully connected layer on the PPA array, resulting in $\times 22$ faster CNN inference (4,464 FPS) on the same digit recognition task. Based on these 2 studies, Liu et al. [38] proposed a high-speed lightweight neural network (Fig. 6, i) using BinaryConnect [88] with a new convolution implementation method that enabled varying convolutional strides. This study demonstrated 4 different classification tasks with frame rates ranging from 2,000 to 17,500 FPS with different stride setups. Subsequently, based on this network, direct servo control using

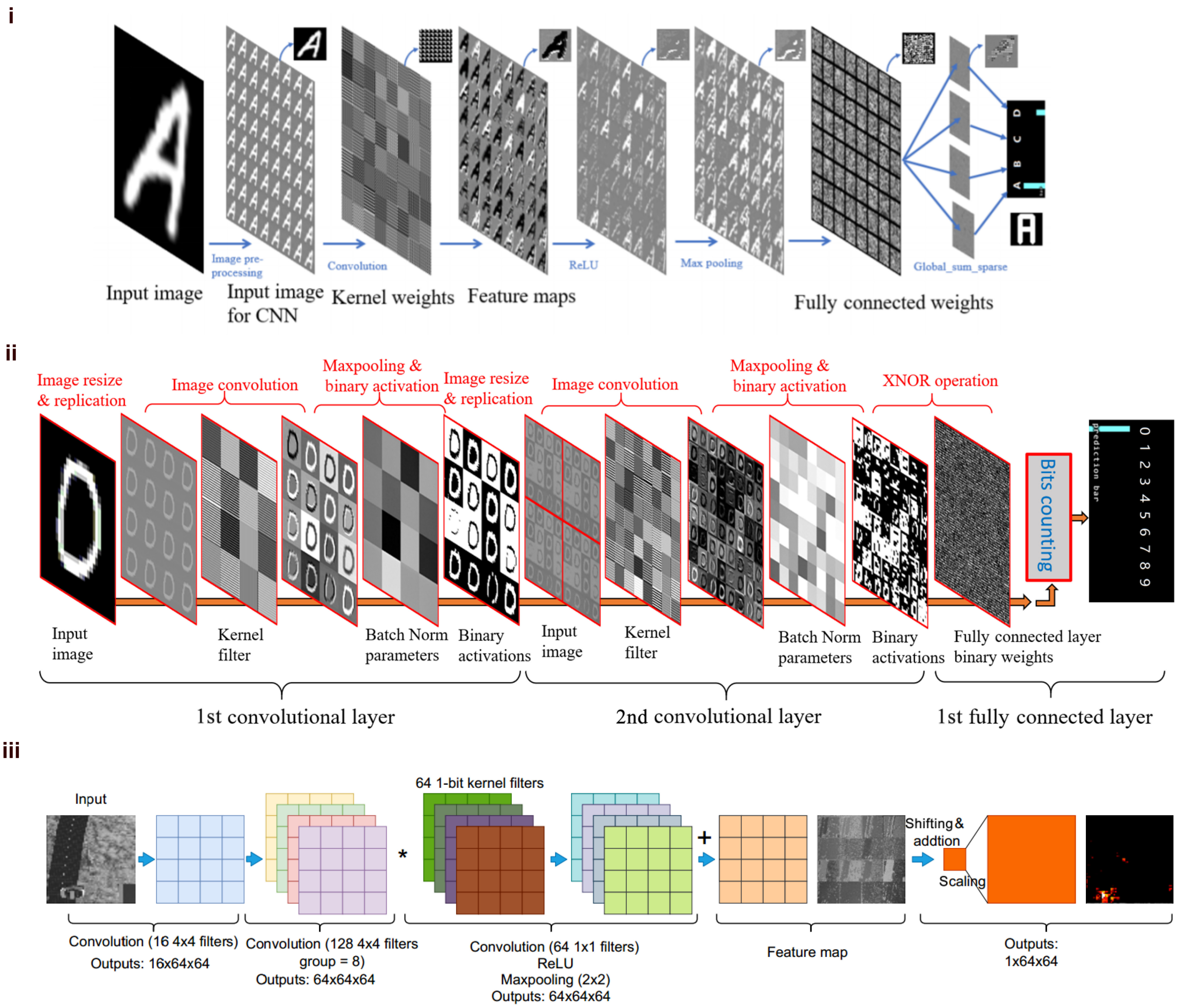

Fig. 6. Overview of SCAMP PPA-based algorithms and applications. In-sensor neural networks. (i) High-speed binary neural network. (ii) Binarized neural network. (iii) Binarized fully convolutional network (FCN).

CNN results [89] and simulated robot tracking from a drone [59] using the in-sensor CNN computing results were exploited.

AnalogNet2 [90,91] extended the initial work [92], implementing a CNN that achieved 96.9% accuracy on the MNIST dataset at a speed of 2,260 FPS; however, it required all fully connected layers to be performed externally to the PPA array with only 3 convolutional kernel filters on the PPA as the first layer. Notably, in [82], a recurrent neural network was implemented on the microcontroller of the SCAMP with features extracted from the sensor. Consequently, these 2 studies combined in-sensor feature extraction and offline network linear layers. Liu et al. [93] proposed a binarized CNN (binary weights and activations) with a batch norm for both classification and coarse segmentation (Fig. 6, ii). To handle classification problems with more labels and segmentation tasks, they proposed the idea of dynamic model swapping by uploading weights of trained models onto PPA registers, targeting multiple simpler subtasks.

Fully convolutional neural network for coarse segmentation and localization

Previous work on PPA CNN inference performed classification, and no neural network architecture for segmentation and localization existed. To expand neural network architecture on the PPA, Liu et al. [94] (Fig. 6, iii) introduced fully convolutional neural networks based on binarized neural networks with 3 convolutional layers. In their work, object localization and coarse segmentation were implemented based on heat map extraction. In addition, to simplify computation and reduce the number of intermediate parameters, group convolution was employed. Considering the limited AREG and DREG resources available on sensors, a dynamic model swapping scheme can be a solution for running multiple networks in an application. A neural network can be decomposed into several smaller networks within the storage and computing capacity of the PPA, and these networks can be performed in sequence to generate the final inference results.

Applications

In-sensor state estimation

2D and 3D estimations are some of the most common applications of machine vision systems. These methods are typically computationally expensive; therefore, powerful processing units are required. However, a conflict exists between the power consumption and processing efficiency of embedded vision systems, which require both low latency and low power costs. The SCAMP PPA offers a solution to these issues because of its capability for high-speed signal processing with low power consumption. This section investigates in-sensor 2D and 3D estimation.

Pose estimation

Bose et al. [31] proposed in-sensor 4-degree-of-freedom (DoF) visual odometry entirely in sensors by mapping a real-time input image with the previous keyframe through image scaling, shifting, rotation, and alignment (Fig. 7, ii). They demonstrated visual odometry (VO) estimation at frequencies greater than 1,000 Hz with a power cost of approximately 2 W. Subsequently, Murai et al. [83] proposed 6-DoF visual odometry based on the edge and corner points extracted in the sensor and postprocessing on a computer with a frame rate of 300 FPS (Fig. 7, i). They leveraged feature edge and corner extraction methods [32] and calculated the visual odometry off the sensor using a method similar to that of standard VO systems [95]. They combined in-sensor feature extraction and a ready-to-use VO computing method off-sensor, which might be a direction in the future to combine efficient image preprocessing in-sensor and high-volume postprocessing with a powerful CPU/GPU, particularly when facing a shortage of storage and general calculation resources for large-scale computing.

Depth estimation

The SCAMP vision system can also operate with other accessories to share the computational burden of additional applications. Martel et al. [96–98] mounted a controllable liquid lens to generate a semi-dense map in real time. Theirs was the first study on depth estimation that leveraged external physical accessories (Fig. 7, iii). Using this focus-tunable lens, a large amount of computational pressure on the sensor was relieved. In an experiment, their method achieved sparse depth images at a frame rate of greater than 25 FPS, providing 32 depth levels. This in-sensor feature extraction and post-image processing of the controller scheme are also widely used in many different applications [82,83] (Fig. 7, iii) where the task requirement for storage and computing resources is greater than the capacity of the PPA as mentioned previously.

Gaze estimation

Considering the size of the PPA sensor chip and its low power consumption, it could potentially be deployed on wearable devices such as glasses and virtual reality devices. Bose et al. [99] showed gaze tracking at greater than 10,000 Hz with a processing delay of less than 0.1 ms. The PPA enables effective visual data processing at the point of light acquisition (Fig. 7, iv). Their work decreased data transmission from the sensor to processing from whole pictures to a handful of contextual bytes by extracting information necessary for gaze tracking on the PPA, saving substantial power and time and enabling a speed considerably higher than that of conventional camera sensors.

In-sensor computing for robots

Short battery life and limited load are the 2 main factors that prohibit long-term use of mobile robots in various applications. In-sensor computing and visual devices may be the keys to solving this problem with emerging hardware designs. The portable PPA system (171 g) can perform in-sensor processing for spatial AI to reduce the data transmission pressure between the sensor and main processor, thereby improving overall

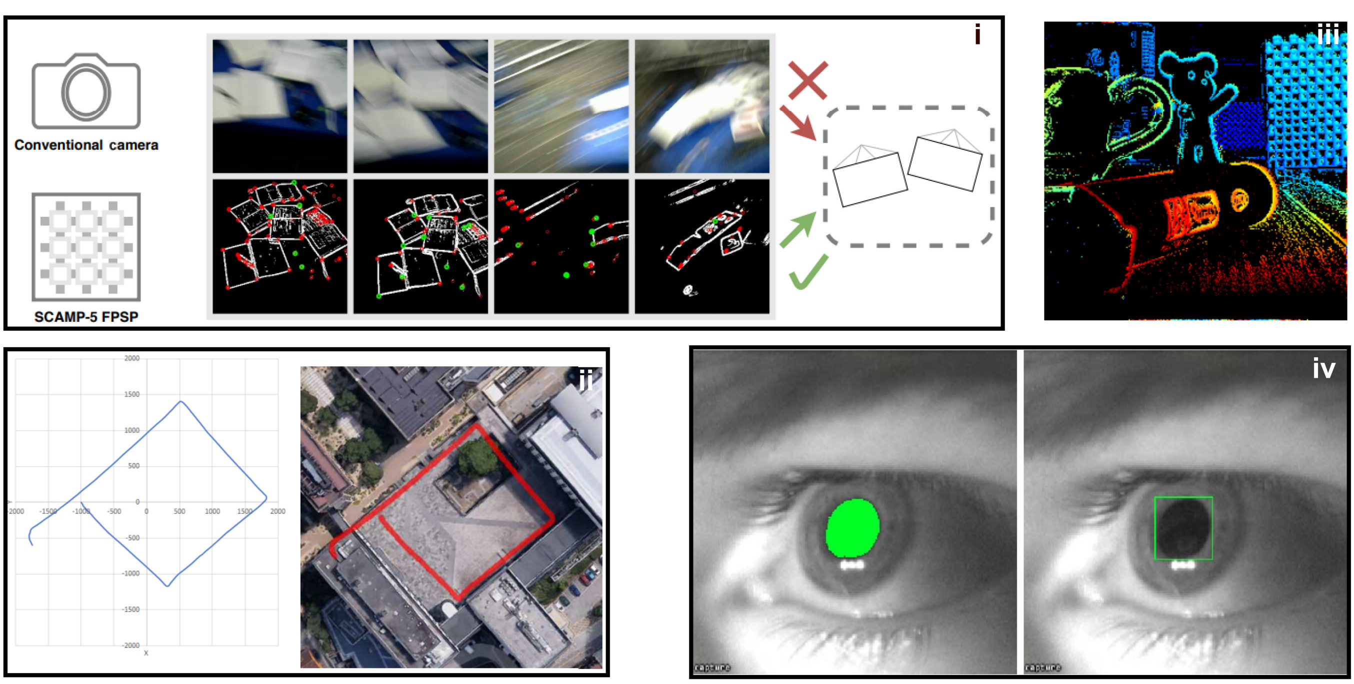

Fig. 7. Overview of SCAMP PPA-based algorithms and applications. State estimation. (i) Six-DoF binary feature visual odometry (BIT-VO). (ii) Four-DoF VO. (iii) Depth estimation. (iv) Gaze estimation.

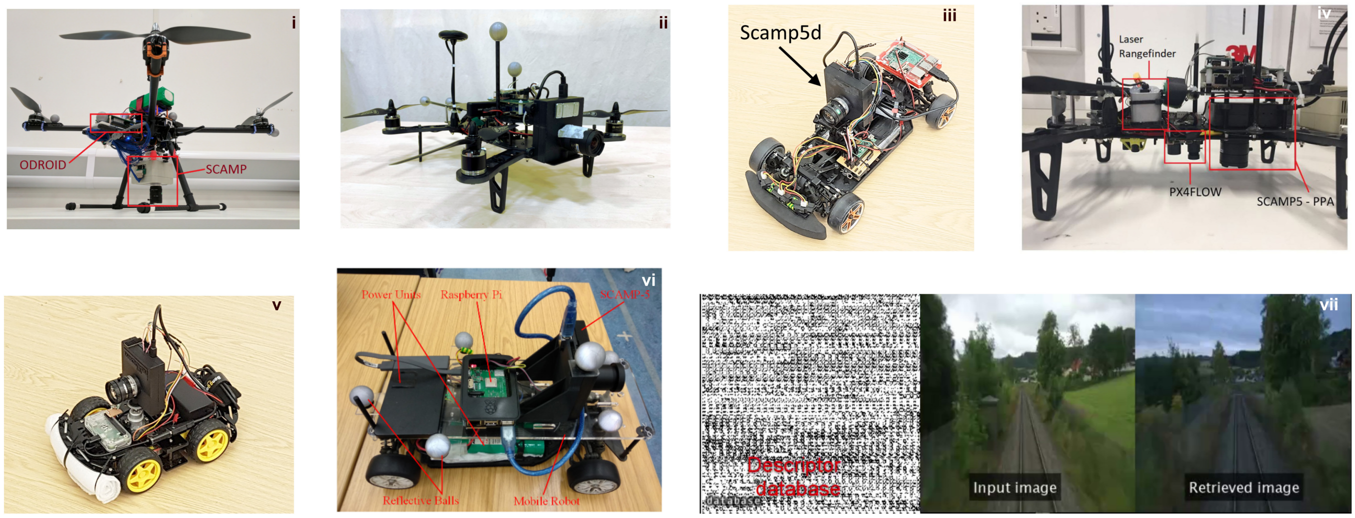

Fig. 8. Overview of SCAMP PPA-based algorithms and applications. Robot applications. (i) Ground target tracking. (ii) Drone racing. (iii) Robot random exploring with a car-like robot. (iv) VO with a drone. (v) Robot random exploring with a fixed-wheel robot. (vi) Agile reactive navigation. (vii) Mapping and localization.

processing efficiency while maintaining low power consumption [100].

In-sensor perception and navigation for quadrocopters

The SCAMP-5d vision system has been integrated into quadrocopter systems for target tracking, visual odometry, and racing. Greatwood et al. [40,41,101] performed various experiments by integrating a SCAMP-5d vision system with a quadrotor. Figure 8 (i) shows a flight control system in terms of hardware integration and a control block diagram, where a preset target can be tracked with useful information extracted from the sensor, even for short periods of target loss [40]. In this application, direct in-sensor target position extraction reduces the pressure of image capture, transmission, and processing for the entire system. Subsequently, based on a similar hardware platform, Greatwood et al. proposed in-sensor visual odometry using perspective correction in an agile micro-air vehicle. Subsequently, a drone (Fig. 8, ii) racing within a preset environment was demonstrated by leveraging the efficient image processing ability of PPA [101], wherein the target position can be estimated at approximately 500 FPS. McConville et al. [42] applied the in-sensor visual odometry developed by Bose et al. [31] to an unmanned aerial system for real-time and control purposes (Fig. 8, iv). They demonstrated that the SCAMP-5 PPA sensor can be used for position estimation in outdoor flights, potentially enabling navigation and recovery in global navigation satellite system-denied environments.

In-sensor perception and navigation for mobile vehicles

Liu [39] et al. proposed reactive agile navigation on a nonholonomic ground vehicle up to 3.88 m/s (Fig. 8, vi) using the PPA to robustly recognize preset patterns in a complex environment. Using a preset fixed pattern for target tracking is, although extremely efficient and accurate, limited to generalized environments that have useful random features. Therefore, Chen et al. [82] used in-focal-plane feature extraction from the environment and ran a recurrent neural network on the microcontroller using this extracted information to estimate the proximity to the ambient objects for obstacle avoidance purposes (Fig. 8, iii and v) with a speed ranging from 0.64 to 1.8 m/s in the experiment. Furthermore, Liu et al. [89] demonstrated a direct visual servo control using CNN inference results, which is promising for future vision-motor control platforms such as ground vehicles to have a lightweight servo control system without external control units.

Castillo-Elizalde et al. [102] proposed 1D mapping and localization by extracting features from the input images as the database and then localizing the incoming image by comparing it with the database and applying the motion model. In their work, 2 methods were utilized in different scenarios to downsample the original images: direct resizing and a local binary pattern. Their implementation achieved a running speed of over 300 Hz on large public datasets that encompassed more than 2,000 locations. It operates with a power consumption of 2.5 W and 500 GOPS/W. The potential applications of this study are to give mobile robots the ability to efficiently create in-sensor maps and localize (Fig. 8, vii).

Challenges and Future Trends

As illustrated, the SCAMP PPA is a versatile in-sensor computing device designed using novel electric circuits. This unique hardware design enables various image processing algorithms ranging from low- and mid-level image processing to high-level neural network inference. These algorithms have a wide range of applications in the fields of machine vision and robotics. Although the PPA has unique advantages over conventional machine vision systems, several limitations associated with this technology exist, which are summarized as follows:

1. The current resolution of the SCAMP-5d vision system is 256×256 for grayscale images, which inherently forbids applications that require accurate localization and detection with RGB clues.

2. Computing and RAM resources are scarce. The PPA offers only 7 in-sensor DREGs and 13 AREGs for both calculation and temporary memory, blocking the development and deployment of more sophisticated algorithms.

3. Analog noise and computing errors are nonnegligible for in-sensor computing with AREGs, particularly when it comes to iterative massive parallel computing.

4. More advanced parallel algorithms, particularly neural networks for image processing adaptive to the unique cellular circuit design, are still needed.

However, we regard these limitations as motivations for manufacturing optimization and research opportunities rather than obstacles. Engineers and researchers are developing and optimizing next-generation SCAMP vision systems based on the following aspects:

1. Hardware platforms: The next-generation in-sensor computing PPA requires higher resolution and more computing resources, a more powerful microcontroller, better AREGs with less noise, and more advanced manufacturing techniques to decrease the bulk of the vision system.

2. Model deployment methods: More advanced programming and development tools, particularly for neural networks, are necessary to simply download a neural network model into the SCAMP PPA and avoid manually implementing the CNN layer by layer.

3. High-performance neural network quantization and deployment: Considering the issue of reduction in accuracy while deploying a quantized neural network in a sensor, a high-performance neural network quantization method is needed. Further network compression approaches need to be investigated for enhanced neural network compression performance. In addition, associated network deployment techniques should be proposed to bridge the performance gap between off-sensor simulation and in-sensor inference.

4. Unconventional computing in sensors: Considering the aforementioned limitations, novel computing architectures and methods for on-sensor computing need to be investigated. For example, advancements have been made in unconventional computing methods such as neural cellular automata [103], elementary cellular automata, neuromorphic computing, transformers [104], graph neural networks [105], and reservoir computing [106]. In particular, cellular automata-based reservoir computing [107] has achieved considerable progress in sequential information processing, which is inspired by the mechanism of recurrent neural networks.

5. Sensor fusion: A variety of studies and applications have been proposed with a single grayscale SCAMP PPA. However, some research, such as research in stereo vision or simultaneous localization and mapping (slam), that typically needs multiple sensors is complicated for a single SCAMP PPA. Hence, the fusion of multiple SCAMP PPAs or a PPA with other types of sensors is a future direction that could further explore the in-sensor capabilities of the SCAMP PPA.

6. Edge computing: Edge computing is the process of physically bringing computational capacity closer to the source of data, which is typically from an IoT device or a sensor. Edge computing obtains its name from the way computational power is sent to a device or network's "edge." Data are processed faster, bandwidth is increased, and data sovereignty is ensured via edge computing. The in-sensor computing device is suitable as a sensory terminal for edge computing because of its signal processing ability.

Notably, the current SCAMP-5d vision system is manufactured using 180-nm techniques, and the whole vision system costs less than 2 W of power. The next-generation SCAMP vision system under development will achieve higher performance with less power consumption. Our team is working on the co-development and co-optimization of circuit design, integration technologies, and associated algorithms needed to successfully deploy in-sensor image processing with the PPA for both academic and commercial applications.

Conclusion

Power consumption, processing latency, and data security are key issues that limit the development of conventional machine vision systems owing to their separate hardware architectures for perception, storage, and processing. These difficulties become more apparent as image resolution increases. However, a new visual sensor hardware architecture with in-sensor visual information processing may be the key to resolving these issues. Hence, this study analyzed the recent advancements in in-sensor computing using different types of visual sensors. Specifically, we focused on the PPA as a major example because of its consistent research history and various applications. Relevant types of visual sensors with in-sensor computing capabilities were also introduced. This paper reviews in-sensor work with the PPA during the last 15 to 20 years, in which key algorithms and applications have been introduced, enabling technological progress ranging from low-level image processing to visual odometry, neural networks, and mobile robot navigation. Despite its many limitations, numerous research studies and applications based on the SCAMP-5 vision system are underway, primarily because its unique electric circuit design enables low power, efficient, and secure image processing for embedded systems. More meaningful and in-depth research outputs with a better fabricated PPA (the SCAMP-7 vision system) are foreseeable in the near future. To realize the vast potential of in-sensor processing technologies, collaboration between chip engineers, image processing researchers, and roboticists is required, starting with low-level VLSI design and progressing to fundamental algorithms and practical applications. The SCAMP PPA can be regarded as an emerging, interdisciplinary, and fertile research platform for studying analog signal processing, machine learning, fundamental parallel image algorithms, and novel simultaneous perception and processing for mobile/embedded systems.

Acknowledgments

Funding: This work was funded by the National Natural Science Foundation of China (grant 62202285), the Science and Technology Commission of Shanghai Municipality (grant 22511104500), and the Fundamental Research Funds for the Central Universities. **Competing interests:** The authors declare they have no competing interests.

Data Availability

Data are available upon request.

Supplementary Materials

Figs. S1 and S2
Tables S1 to S5

References

1. Frisby JP, Stone JV. Seeing: The computational approach to biological vision. Cambridge (MA): MIT Press; 2010.

2. Lao J, Yan M, Tian B, Jiang C, Luo C, Xie Z, Zhu Q, Bao Z, Zhong N, Tang X, et al. Ultralow-power machine vision with self-powered sensor reservoir. *Adv Sci*. 2022;9(15):Article e2106092.
3. Wan T, Shao B, Ma S, Zhou Y, Li Q, Chai Y. In-sensor computing: Materials, devices, and integration technologies. *Adv Mater*. 2022;Article e2203830.
4. Semiconductor Research Corporation and Semiconductor Industry Association, Interim report for the decadal plan for semiconductors; 2020. p. 1–21.
5. Pan W, Zheng J, Wang L, Luo Y. A future perspective on in-sensor computing. *Engineering*. 2022.
6. Zhou F, Chai Y. Near-sensor and in-sensor computing. *Nat Electron*. 2020;3(11):664–671.
7. Ielmini D, Wong H-SP. In-memory computing with resistive switching devices. *Nat Electron*. 2018;1(6):333–343.
8. Chai Y. In-sensor computing for machine vision. *Nature*. 2020;579:32–33.
9. Hubel DH. *Eye, brain, and vision*. New York City: Scientific American Library/Scientific American Books; 1995.
10. Mennel L, Symonowicz J, Wachter S, Polyushkin DK, Molina-Mendoza AJ, Mueller T. Ultrafast machine vision with 2D material neural network image sensors. *Nature*. 2020;579(7797):62–66.
11. Song R, Huang K, Wang Z, Shen H. A reconfigurable convolution-in-pixel CMOS image sensor architecture. arXiv 2021. arXiv.2101.03308.
12. inivation, Dynamic vision platform. 2021. https://inivation.com/dvp/.
13. Wang T-Y, Meng JL, Li QX, He ZY, Zhu H, Ji L, Sun QQ, Chen L, Zhang DW. Reconfigurable optoelectronic memristor for in-sensor computing applications. *Nano Energy*. 2021;89:Article 106291.
14. Wang Y, Gong Y, Yang L, Xiong Z, Lv Z, Xing X, Zhou Y, Zhang B, Su C, Liao Q, et al. Mxene-zno memristor for multimodal in-sensor computing. *Adv Funct Mater*. 2021;31:2100144.
15. Sun L, Wang Z, Jiang J, Kim Y, Joo B, Zheng S, Lee S, Yu WJ, Kong BS, Yang H. In-sensor reservoir computing for language learning via two-dimensional memristors. *Sci Adv*. 2021;7:eabg1455.
16. Zarándy Á. *Focal-plane sensor-processor chips*New York City: Springer Science & Business Media; 2011.
17. Rodríguez-Vázquez Á, Domínguez-Castro R, Jiménez-Garrido F, Morillas S, Listán J, Alba L, Utrera C, Espejo S, Romay R. *The eye-ris cmos vision system, Analog circuit design*New York City: Springer; 2008. p. 15–32.
18. Kovilta. https://kovilta.fi/technology/.
19. Aistorm, Ai in sensor. [accessed 22 Nov 2021] https://aistorm.ai/.
20. Zhang J, Dai S, Zhao Y, Zhang J, Huang J. Recent progress in photonic synapses for neuromorphic systems. *Adv Intell Syst*. 2020;2:1900136.
21. Dudek P. A programmable focal-plane analogue processor array [thesis]. The University of Manchester; 2000.
22. Carey S, Dudek P. Vision chip with high accuracy analog S^2I cells. Paper presented at: 2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA); 2014 Jul 29–31; Notre Dame, Indiana, USA. p. 1–2.
23. Carey SJ, Barr DR, Wang B, Lopich A, Dudek P. Mixed signal SIMD processor array vision chip for real-time image processing. *Analog Integr Circ Sig Process*. 2013;77:385–399.
24. Lopich A, Dudek P, A general-purpose vision processor with 160 × 80 pixel-parallel simd processor array. Paper presented at: Proceedings of the IEEE 2013 Custom Integrated Circuits Conference; 2013 Sep 22–25; San Jose, CA, USA. p. 1–4.
25. J. N. Martel and P. Dudek, "Vision chips with in-pixel processors for high-performance low power embedded vision systems," in *ASR-MOV Workshop, CGO*, vol. 6, 2016, p. 14.
26. Dudek P, Hicks PJ, An simd array of analogue microprocessors for early vision. Paper presented at: PREP'99: Postgraduate Research in Electronics, Photonics and Related Fields; 1999 Jan 05–07; UMIST, Manchester. p. 359–362.
27. Dudek P, Hicks PJ. A CMOS general-purpose sampled-data analog processing element. *IEEE Trans Circuits Syst II Analog Digit Signal Process*. 2000;47:467.
28. Dudek P, Richardson T, Bose L, Carey S, Chen J, Greatwood C, Liu Y, Mayol-Cuevas W. Pixel processor array for sensor-level computer vision on agile robots. *Sci Robot*. 2022.
29. Carey SJ, Lopich A, Barr DR, Wang B, Dudek P. A 100,000 fps vision sensor with embedded 535GOPS/W 256×256 SIMD processor array. Paper presented at: 2013 Symposium on VLSI Circuits; 2013 Jun 12–14; Kyoto, Japan. p. C182–C183.
30. Martel JN, Mueller LK, Carey SJ, Dudek P, Wetzstein G. Neural sensors: Learning pixel exposures for HDR imaging and video compressive sensing with programmable sensors. *IEEE Trans Pattern Anal Mach Intell*. 2020;42(7): 1642–1653.
31. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W. Visual odometry for pixel processor arrays. Paper presented at: 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 4604–4612.
32. Chen J, Carey SJ, Dudek P, Feature extraction using a portable vision system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop Vision-Based Agile Autonomous Navigation; 2017.
33. Chua LO, Yang L. Cellular neural networks: Theory. *IEEE Trans Circuits Syst*. 1988;35:1257.
34. Roska T, Chua LO. The cnn universal machine: An analogic array computer. *IEEE Trans Circuits Syst II Analog Digit Signal Process*. 1993;40:163.
35. Orovas C, Cellular associative neural networks for pattern recognition [thesis]. University of York; 2000.
36. Rosin PL. Training cellular automata for image processing. *IEEE Trans Image Process*. 2006;15:2076.
37. Torralba AB, Analogue architectures for vision: Cellular neural networks and neuromorphic circuits [thesis]. Institute national Polytechnique Grenoble, Laboratory of Images and Signals; 1999.
38. Liu Y, Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W. High-speed lightweight cnn inference via strided convolutions on a pixel processor array. Paper presented at: The 31st British Machine Vision Conference (BMVC 2020); 2020 Sep 07–10; Virtual Conference.
39. Liu Y, Bose L, Greatwood C, Chen 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 Process*. 2021;15(9):1883–1892.
40. Greatwood C, Bose L, Richardson T, Mayol-Cuevas W, Chen J, Carey SJ, Dudek P. Tracking control of a UAV with a parallel visual processor. Paper presented at: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2017 Sep 24–28; Vancouver, British Columbia, Canada. p. 4248–4254.

41. C. Greatwood, Bose L, Richardson T, Mayol-Cuevas W, Chen J, Carey SJ, Dudek P. Perspective correcting visual odometry for agile mavs using a pixel processor array. Paper presented at: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2018 Oct 01–05; Madrid, Spain. p. 987–994.
42. McConville A, Bose L, Clarke R, Mayol-Cuevas W, Chen J, Greatwood C, Carey S, Dudek P, Richardson T. Visual odometry using pixel processor arrays for unmanned aerial systems in GPS denied environments. *Front Robot AI*. 2020;7:126.
43. Carey SJ, Barr DR, Dudek P. Low power high-performance smart camera system based on SCAMP vision sensor. *J Syst Archit*. 2013;59(10):889–899.
44. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W. A camera that CNNs: Towards embedded neural networks on pixel processor arrays. Paper presented at: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–Nov 02; Seoul, Korea (South). p. 1335–1344.
45. Debrunner T, Saeedi S, Kelly PH. Auke: Automatic kernel code generation for an analogue simd focal-plane sensor-processor array. *ACM Trans Archit Code Optim*. 2019;15(4):Article 59.
46. Malekzadeh M, Clegg RG, Cavallaro A, Haddadi H. Privacy and utility preserving sensor-data transformations. *Pervasive Mob Comput*. 2020;63:Article 101132.
47. Pinkham R, Berkovich A, Zhang Z. Near-sensor distributed dnn processing for augmented and virtual reality. *IEEE J Emerg Sel Top Circuits Syst*. 2021;11:663.
48. Sony, Intelligent vision sensors with AI processing functionality [accessed 14 May 2020] https://www.sony.com/en/SonyInfo/News/Press/202005/20-037E/.
49. Eye-RIS, Eye-ris v1.3 hardware description [accessed 16 Sep 2022] https://imaging.teledyne-e2v.com/products/2d-cmos-image-sensors/eye-ris/.
50. Sotoa LA, Morillas S, Listán J, Jiménez A, Arena P, Patané L, De Fiore S. Embedding the anafocus' eye-ris vision system in roving robots to enhance the action-oriented perception. *Neurophysiology*. 2009;7365:821634.
51. Yao P, Wu H, Gao B, Tang J, Zhang Q, Zhang W, Yang JJ, Qian H. Fully hardware-implemented memristor convolutional neural network. *Nature*. 2020;577:641.
52. Thomas A. Memristor-based neural networks. *J Phys D Appl Phys*. 2013;46:Article 093001.
53. Lungu IA, Liu S-C, Delbruck T. Fast event-driven incremental learning of hand symbols. Paper presented at: 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS) (IEEE, 2019); 2019 May 18–20; Hsinchu, Taiwan.
54. Lungu IA, Corradi F, Delbrück T. Live demonstration: Convolutional neural network driven by dynamic vision sensor playing roshambo. Paper presented at: 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE, 2017); 2017 May 28–31; Baltimore, MD, USA.
55. Linares-Barranco A, Rios-Navarro A, Tapiador-Morales R, Delbruck T. Dynamic vision sensor integration on fpga-based cnn accelerators for high-speed visual classification. arXiv 2019. arXiv:1905.07419.
56. Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba G, Censi A, Leutenegger S, Davison A, Conradt J, Daniilidis K, Scaramuzza D. Event-based vision: A survey. arXiv 2019. arXiv:1904.08405.
57. Datta G, Kundu S, Yin Z, Teja Lakkireddy R, Mathai J, Jacob A, Beerel PA, Jaiswal AR. P2m: A processing-in-pixel-in-memory paradigm for resource-constrained tinyml applications. arXiv 2022. arXiv:abs/2203.04737.
58. Chen J, Carey SJ, Dudek P. Scamp5d vision system and development framework. Paper presented at: Proceedings of the 12th International Conference on Distributed Smart Cameras; 2018.
59. Liu Y, Chen J, Bose L, Dudek P, Mayol-Cuevas W. Bringing a robot simulator to the scamp vision system. Paper presented at: 2021 IEEE International Conference on Robotics and Automation (ICRA) Workshop: On and Near-sensor Vision Processing, from Photons to Applications (IEEE, 2021); 2021.
60. Fan W, Liu Y, Xing Y. Fully-simulated integration of scamp5d vision system and robot simulator. arXiv 2021. arXiv:2110.06386.
61. Stow E, Murai R, Saeedi S, Kelly PH. Cain: Automatic code generation for simultaneous convolutional kernels on focal-plane sensor-processors. arXiv 2021. arXiv:2101.08715.
62. Stow EJ, Murai R, Saeedi S, Kelly PHJ. Automatic code generation for simultaneous convolutional kernels on cellular processor arrays. arXiv 2020. arXiv.2101.08715.
63. Stow E, Kelly PHJ. Super-pixel compilation for pixel processor arrays. Paper presented at: 2021 IEEE International Conference on Robotics and Automation (ICRA) workshop: On and Near-sensor Vision Processing, from Photons to Applications (IEEE, 2021); 2021.
64. Wang B. Pixel-parallel image processing techniques and algorithms [thesis]. University of Manchester; 2014.
65. Barr DR, Carey SJ, Dudek P. Low power multiple object tracking and counting using a scamp cellular processor array. Paper presented at: 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications (IEEE, 2012); 2012 Aug 29–31; Turin, Italy.
66. Carey SJ, Barr DR, Wang B, Lopich A, Dudek P. Locating high speed multiple objects using a scamp-5 vision-chip. Paper presented at: 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications (IEEE, 2012); 2012 Aug 29–31; Turin, Italy.
67. Martel JN, Sandamirskaya Y, Dudek P. A demonstration of tracking using dynamic neural fields on a programmable vision chip. Paper presented at: Proceedings of the 10th International Conference on Distributed Smart Camera (2016); 2016 Sep 12; New York, USA.
68. Martel JN. Unconventional processing with unconventional visual sensing: Parallel, distributed and event based vision algorithms & systems [thesis]. ETH Zurich; 2019.
69. Dudek P, 2006 International Workshop on Computer Architecture for Machine Perception and Sensing (IEEE, 2006); 2006 Aug 18–20.
70. Dudek P, Carey S. General-purpose 128 spl times 128 simd processor array with integrated image sensor. *Electron Lett*. 2006;42:678.
71. Martel JN, Müller LK, Carey SJ, Dudek P. Parallel hdr tone mapping and autofocus on a cellular processor array vision chip. Paper presented at: 2016 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE, 2016); 2016 May 22–25; Montreal, QC, Canada.
72. Martel JN, Müller LK, Carey SJ, Dudek P, A real-time high dynamic range vision. Paper presented at: 15th International Workshop on Cellular Nanoscale Networks and their Applications (VDE, 2016); 2016 Jun 7–12; Boston, MA, USA.

73. So HM, Martel JN, Dudek P, Wetzstein G. Mantissacam: Learning snapshot high-dynamic-range imaging with perceptually-based in-pixel irradiance encoding. arXiv 2021. arXiv:2112.05221.
74. Dudek P, Vilariño DL. A cellular active contours algorithm based on region evolution. Paper presented at: 2006 10th International Workshop on Cellular Neural Networks and Their Applications (IEEE, 2006); 2006 Jul 1.
75. Alonso-Montes C, Vilarino D, Penedo M. Cnn-based automatic retinal vascular tree extraction Paper presented at: 2005 9th International Workshop on Cellular Neural Networks and Their Applications (IEEE, 2005); 2005.
76. Dudek P, Hülse M, Barr DR. *Cellular automata and non-static image processing for embodied robot systems on a massively parallel processor array*, Automata-2008: Theory and Applications of Cellular Automata Luniver Press; 2008. p. 504–510.
77. Wang B, Mroszczyk P, Dudek P. A new method for fast skeletonization of binary images on cellular processor arrays. Paper presented at: 2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA) IEEE; 2014 Jul 29–31; Notre Dame, IN, USA.
78. Mroszczyk P, Dudek P. Trigger-wave collision detecting asynchronous cellular logic array for fast image skeletonization. Paper presented at: 2012 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE, 2012); 2012 May 20–23; Seoul, Korea (South).
79. Razmjooei S, Dudek P. Approximating euclidean distance transform with simple operations in cellular processor arrays. Paper presented at: 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010) (IEEE, 2010); 2010 Feb 3–5; Berkeley, CA, USA.
80. Wang B, Dudek P. Amber: Adapting multi-resolution background extractor. Paper presented at: 2013 IEEE International Conference on Image Processing (IEEE, 2013); 2013 Sep 15–18; Melbourne, VIC, Australia.
81. Wang B, Dudek P. A fast self-tuning background subtraction algorithm. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014.
82. Chen J, Liu Y, Carey SJ, Dudek P. Proximity estimation using vision features computed on sensor. Paper presented at: 2020 IEEE International Conference on Robotics and Automation (ICRA) (IEEE, 2020); 2020 May–Aug 31; Paris, France.
83. Murai R, Saeedi S, Kelly PH. Bit-vo: Visual odometry at 300 fps using binary features from the focal plane. arXiv 2020. arXiv:2004.11186.
84. Wang B, Dudek P. Coarse grain mapping method for image processing on fine grain cellular processor arrays. Paper presented at: 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications (IEEE, 2012); 2012 Aug 29–31; Turin, Italy.
85. Martel JN, Chau M, Cook M, Dudek P. Pixel interlacing to trade off the resolution of a cellular processor array against more registers. Paper presented at: 2015 European Conference on Circuit Theory and Design (ECCTD) (IEEE, 2015); 2015 Aug 24–26; Trondheim, Norway.
86. Bose L, Chen J, Carey SJ, Dudek P, Mayol-Cuevas W. Live demonstration: Digit recognition on pixel processor arrays. Paper presented at: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019); 2019 Jun 16–17; Long Beach, CA, USA.
87. Bose L, Dudek P, Chen J, Carey SJ, Mayol-Cuevas WW. Fully embedding fast convolutional networks on pixel processor arrays, European Conference on Computer Vision (Springer, 2020); 2020 Aug 23–28; Berlin, Heidelberg.
88. Courbariaux M, Bengio Y, David J-P. Binaryconnect: Training deep neural networks with binary weights during propagations. Paper presented at: NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems; 2015 Dec 7; Cambridge, MA, USA.
89. Liu Y, Chen J, Bose L, Dudek P, Mayol-Cuevas W. Direct servo control from in-sensor cnn inference with a pixel processor array. Paper presented at: 2021 IEEE International Conference on Robotics and Automation (ICRA) workshop: On and Near-sensor Vision Processing, from Photons to Applications (IEEE, 2021); 2021.
90. Guillard B. Optimising convolutional neural networks for super fast inference on focal-plane sensor-processor arrays [thesis]. Imperial College London; 2019.
91. Wong MZ, Guillard B, Murai R, Saeedi S, Kelly PH. Analognet: Convolutional neural network inference on analog focal plane sensor processors. arXiv 2020. arXiv:2006.01765.
92. Wong M, Saeedi S, Kelly PH. Analog vision-neural network inference acceleration using analog simd computation in the focal plane [thesis]. Imperial College London; 2018.
93. Liu Y, Bose L, Fan R, Dudek P, Mayol-Cuevas W. On-sensor binarized cnn inference with dynamic model swapping in pixel processor arrays. *Front Neurosci*. 2022;16:Article 909448.
94. Liu Y, Bose L, Lu Y, Dudek P, Mayol-Cuevas W. On-sensor binarized fully convolutional neural network with a pixel processor array. arXiv 2022. arXiv:2202.00836.
95. Klein G, Murray D. Parallel tracking and mapping for small ar workspaces. Paper presented at: 2007 6th IEEE and ACM international symposium on mixed and augmented reality (IEEE, 2007); 2007 Nov 13–16; Nara, Japan.
96. Martel JN, Muller LK, Carey SJ, Muller J, Sandamirskaya Y, Dudek P. Real-time depth from focus on a programmable focal plane processor. *IEEE Trans Circuits Syst I: Regular Papers*. 2017;65(3):925–934.
97. Martel JN, Müller LK, Carey SJ, Dudek P. High-speed depth from focus on a programmable vision chip using a focus tunable lens. Paper presented at: 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE, 2017); 2017 May 29–31; Baltimore, MD, USA.
98. Martel JN, Müller LK, Carey SJ, Dudek P. Paper presented at: 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE, 2017). p. 1–1.
99. L. Bose, J. Chen, S. J. Carey, P. Dudek, Pixel processor arrays for low latency gaze estimation. Paper presented at: 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (IEEE, 2022); 2022.
100. Davison AJ. Futuremapping: The computational structure of spatial ai systems. arXiv 2018. arXiv:1803.11288.
101. Greatwood C, Bose L, Richardson T, Mayol-Cuevas W, Clarke R, Chn J, Carey S. Towards drone racing with a pixel processor array, 11th International Micro Air Vehicles. Paper presented at: Conferences and Competitions (2019); 2019.
102. Castillo-Elizalde H, Liu Y, Bose L, Mayol-Cuevas W. Weighted node mapping and localisation on a pixel processor array. Paper presented at: 2021 IEEE International Conference on Robotics and Automation (ICRA) (IEEE, 2021); 2021 May 30–Jun 5; Xi'an, China.

103. Mordvintsev A, Randazzo E, Niklasson E, Levin M. Growing neural cellular automata. *Distill*. 2020;5: Article e23.
104. Vaswani A et al. Attention is all you need. *Adv Neural Inf Proces Syst*. 2017;5998–6008.
105. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans Neural Netw*. 2008;20(1):61.
106. Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D, Hirose A. Recent advances in physical reservoir computing: A review. *Neural Netw*. 2019;115:100.
107. Yilmaz O. Reservoir computing using cellular automata. arXiv 2014. arXiv:1410.0162.
108. Kaneda M. Signal processing in the mammalian retina. *J Nippon Med Sch*. 2013;80(1):16–24.