Introduction to the Special Issue on Embedded Systems for Computer Vision

Samarjit Chakraborty
The University of North Carolina (UNC) at Chapel Hill, US

Qing Rao
Momenta Europe GmbH, Stuttgart, Germany

We provide a broad overview of some of the current research directions at the intersection of embedded systems and computer vision, in addition to introducing the papers appearing in this special issue. Work at this intersection is steadily growing in importance, especially in the context of autonomous and cyber-physical systems design. Vision-based perception is almost a mandatory component in any autonomous system, but also adds myriad challenges like, how to efficiently implement vision processing algorithms on resource-constrained embedded architectures, and how to verify the functional and timing correctness of these algorithms. Computer vision is also crucial in implementing various smart functionality like security, e.g., using facial recognition, or monitoring events or traffic patterns. Some of these applications are reviewed in this introductory article. The remaining articles featured in this special issue dive into more depth on a few of them.

2012 ACM Subject Classification Computer systems organization → Embedded and cyber-physical systems

Keywords and Phrases Embedded systems, Computer vision, Cyber-physical systems, Computer architecture

Digital Object Identifier 10.4230/LITES.8.1.0

Funding *Samarjit Chakraborty*: Acknowledges support from the US NSF grant# 2038960.

Published 2022-11-16

Editor Samarjit Chakraborty and Qing Rao

Special Issue Special Issue on Embedded Systems for Computer Vision

1 Computer vision and embedded systems

Efficiently implementing computer vision algorithms on resource-constrained embedded systems is necessary for many application domains and is continuing to attract widespread attention in the research community [50]. We are witnessing a surge in the development of various smart and autonomous systems – such as autonomous cars, robots, drones, and industrial automation systems. All of these systems rely on environmental perception in order to generate the necessary control action. Towards this, inputs from various sensors like cameras and lidars need to be processed and their output serves as inputs to control algorithms that compute commands for realizing the desired system functionality. Hence, vision processing algorithms have to meet various architectural and resource constraints and need to be certified for functional and timing correctness in order to ensure the reliability of the entire system. This has triggered research on the development of high-performance embedded systems architectures to support computer vision solutions, e.g., using accelerators like FPGAs and GPUs, in particular for those that rely on machine learning. There has also been recent work on how to provide timing guarantees to computer vision algorithms that are a part of feedback control loops. Power and memory optimization of computer vision algorithms, and issues related to privacy and security of vision-based applications and systems have also been published during the past couple of years.

In view of these developments, we organized this special issue for the Leibniz Transactions on Embedded Systems and invited papers on a variety of topics at this intersection of computer vision and embedded systems. The topics we listed included new embedded systems architectures

– including FPGAs, GPUs, and heterogeneous MpSoCs – for computer vision, novel algorithms for computer vision targeting embedded applications, machine learning and neural networks for image and video understanding for autonomous systems, timing analysis of computer vision algorithms and architectures, performance and power analysis and management of computer vision systems, vision-based control or visual servoing systems, security and privacy issues in vision-based embedded systems, robustness issues in vision-based autonomous systems, and debugging vision-based embedded systems. Papers on applications of computer vision were also invited.

Before we introduce the three papers that comprise this special issue, we briefly review some of the recent literature on the above topics. This review is by no means complete or detailed. Its only purpose is to provide a context for this special issue and impress upon the reader the variety of work that has been done in this area, and the rich set of challenges that are still remaining.

1.1 Control + Vision

As we already hinted above, almost all autonomous systems rely on different feedback controllers that need to be implemented on resource-constrained distributed embedded systems [5]. In the past, controller *design* and their *implementation* were done separately, which resulted in a mismatch between model-level assumptions and implementation realities. This necessitated an iterative design approach and significant effort in testing and debugging. More recently, different parts of a control system are being co-designed and co-synthesized [42].

In the case of *visual servoing* systems, where a computer vision system is used to compute control inputs, co-design is also being used. More specifically, while controller design was traditionally done independently and was disconnected from the vision or perception processing techniques, recently the need for co-design between the two is being recognized [19, 53]. Holistic approaches to closed-loop behavior have also been studied in [44], instead of designing components like object detection, tracking, motion planning, and multi-sensor fusion in isolation, followed by integrating these components together. By co-designing and optimizing these components together, it has been shown that resource utilization may be significantly reduced with minimal impact on performance such as motion planning.

Several studies have focused on evaluation of vision-in-the-loop systems, since building a full prototype of such systems might be too expensive and infeasible. Hence, the evaluation is done in a progressive manner as different components of the system are designed. Here, the work in [21] has proposed an evaluation framework for *model-in-the-loop*, *software-in-the-loop*, and *processor-in-the-loop* simulation features, as the design progresses from the model of the system, to the software and its deployment on multi-core processors. The goal of this work was to evaluate the closed-loop performance of industrial motion control systems. The design of the control strategy, and the design and implementation of the vision processing system – both involve several parameters. How to jointly determine the values of these parameters in order to optimize control performance, and also resource usage, is the main problem. Today, only “point solutions” exist and we will continue to see more work in this area.

1.2 Efficient implementation of vision processing (incl. GPUs & FPGAs)

Efficient implementation of compute-intensive tasks on resource-constrained architectures, such as those seen in the automotive domain, require the use of various forms of hardware accelerators like FPGAs [48]. Towards this, [20] proposed a FPGA-based scalable and resource-efficient multi-camera GigE Vision IP core for video and image processing. This IP core, that was implemented on a Xilinx Virtex-4 FPGA, supports the connection of multi-camera interfaces to a single Gigabit

Ethernet port using an Ethernet switch. Similar FPGA-based implementations have also been studied in [29] and [49]. Since modern cars are now equipped with multiple cameras, lidar, and radar sensors, the volume of data that needs to be acquired for efficient processing is also large and efficient data acquisition techniques and communication architectures have also been studied [14].

Sensor fusion is a standard task in many domains like automotive and robotics, and can be very resource-heavy. Therefore, techniques for efficient sensor fusion has attracted considerable attention. For example, [43] proposes techniques for spatiotemporal sampling to activate Lidars only at regions-of-interest that are identified by analyzing the visual input. Such a task-based sampling approach significantly reduces the volume of data that is sensed and transferred, thereby reducing sensor fusion workload. A similar approach, where a smart camera captures only task-critical information and is driven by embedded deep neural network algorithms for real-time control of sensor parameters has been presented in [32].

As outlined above, in most vision-based control applications, the vision processing algorithms incur very heavy computational workload. The work in [8] studied approximate image processing techniques to reduce this workload, making vision-based control suitable for embedded platforms. The underlying control strategy was adapted to an approximation-aware optimal linear-quadratic-gaussian (LQG) controller, where the approximation error was modeled as sensor noise. This work may also be viewed as a *co-design* between control strategy and its implementation, along the lines of other studies like [19, 53]. Further, the tradeoffs between the degree of approximation, and the resulting closed-loop quality-of-control, and metrics like memory utilization and energy consumption have been studied in [7].

Yet another example of efficient implementation of vision-based control is in [30], which proposes identifying different *system scenarios* that are optimized, and a switched controller switches between these scenarios. Identifying only a limited number of scenarios helps with optimization that would not be possible otherwise. Along similar lines, the work in [31] attempts to reduce sensor-to-actuator delay in vision-based control applications by pipelining the vision task on a multiprocessor architecture. In particular, this work shows how such pipelining may be implemented for model predictive control, while accounting for workload variations in the different stages of the image processing pipeline. As a continuation of this study, how the number of pipeline stages impacts the quality of control has been studied in [45].

1.3 Verification and monitoring

Safety [57] and security [54] of autonomous systems is usually of crucial importance. While there has been considerable work on the functional [17] and also timing verification [47] of cyber-physical systems, and in particular controllers implementing different autonomous features, a full verification of the system also requires a verification of the vision processing and understanding algorithms that provide inputs into the controllers. Towards this, monitoring [13], debugging [12] and optimization techniques for FPGA-based implementation of vision processing [11] have also attracted considerable attention recently. However, work in this area is relatively nascent; as systems become more complex and the need for certification increases, we will see more results in this domain.

1.4 Timing predictable vision processing

While mainstream computer vision has considered *fast* vision processing algorithms, the issue of *real-time*, viz., guaranteeing that the output is produced within a deadline, has not been studied. However, when used in conjunction with control algorithms and in safety-critical systems, such real-time processing guarantees become necessary. Towards this, temporal isolation might be needed between soft real-time applications like computer vision and time-critical applications like control tasks [27]. Although many certification techniques rely on time partitioning, it has been shown that the use of multicore + accelerator platforms can disrupt such partitioning or timing isolation schemes [3]. In this context, how to design the vision processing system to be timing predictable [2, 4], and how to debug systems for timing violations have also been studied [41].

Most control systems tend to be have a certain degree of timing robustness [18], and this can be used for the design of the perception processing subsystem. How the behavior of vision-in-the-loop control systems is impacted by the processing platform and the control software executing on it has been studied in [21]. This study, in particular, evaluates the *predictability* of the embedded computational platform "CompSOC" [16], which guarantees periodic and deterministic execution of control tasks, allowing a verification of their timing properties.

Timing predictable GPU processing: A number of papers have recently investigated the role of GPUs in providing such timing guarantees. It has been established that using OpenCV-based applications on GPUs, unexpected delays might occur not only on the GPUs, but also on the host CPUs, which might jeopardize the real-time constrains associated with vision-based applications [1]. While most studies have focused on how to use GPUs to gain computational advantage, there has been relatively less work done on evaluating the real-time characteristics of GPUs – from both AMD and also NVIDIA [34]. Allocation strategies for real-time management of multi-GPU systems has been been studied in [9].

1.5 Algorithms and data structures for efficient vision processing

There exists a variety of algorithms, processing architectures, and techniques for implementing vision processing tasks on embedded computing platforms [28]. In particular, a lot of recent attention has been on neural network based processing and their implications, like ensuring consistency in their performance [51], or efficiently implementing them on resource-constrained edge devices [15]. To cite examples from specific domains – in-vehicle augmented reality applications [40] and autonomous features in cars (such as automated parking or driving) require new data structures and algorithmic techniques for vision processing [37]. More importantly, these have to be resource efficient, e.g., so that complex 3D shapes of the environment can be transmitted within the vehicle using low- to medium-bandwidth communication infrastructure [36]. In addition to suitable data structures, special compression techniques for resource efficient in-vehicle communication and computation have also been studied [38, 35].

The use of neural networks for image processing on embedded systems has been widely studied in the literature [52, 58], along with specialized hardware architectures for vision processing [26, 29]. When multiple convolutional neural networks (CNNs) with each processing a different video stream are implemented on the same resource-constrained embedded platform, then their inference and timing performance might not be satisfactory. However, reorganizing these CNNs and exploiting parallelism, pipelining, and merging the images, might lead to significant improvements [56]. Some studies have also considered different graph transformation [10] and scheduling strategies for OpenVX graphs to achieve better real-time performance [55]. OpenVX is a standard for cross-platform acceleration of computer vision algorithms, and provides a high-level of abstraction for programming vision applications [33].

In many cost-sensitive domains like automotive and also for general purpose cost-effective imaging, a common challenge is to solve problems using low-cost and resource-constrained components. For example high dynamic range images may be recovered by fusing multiple low dynamic range (LDR) images. If all the LDR images are perfectly aligned, then this fusion process is much easier. However, in reality, there are misalignments and jitters between different cameras and several studies have proposed machine learning techniques for correcting them (e.g., see [25]). This is especially relevant in the automotive and robotics domains where physical movement causes cameras to vibrate. Another example in the same direction – where intelligent algorithmic or machine learning techniques are used to utilize low-cost or resource-constrained components – is where 3D shapes of objects are created from a single image using deep neural networks in the context of autonomous driving [39].

As outlined above, there has been a tremendous growth in the use of machine learning algorithms for vision processing in various embedded computing settings, especially in the automotive domain. But a big challenge in their use is the need for a large amount of labelled training data, especially because often the data needs to be manually labeled. To address this, various techniques have been proposed to reduce the volume of such data that is needed. One such technique is referred to as *active learning*, where a model selects samples for labeling based on their uncertainty, thereby reducing the volume of data needed for various tasks like 3D object detection [46].

1.6 Other applications

While we have so far mostly focused on autonomous systems and their instantiations in domains like automotive and robotics, vision processing is also useful in various monitoring applications for example for monitoring traffic flow [23] to devise suitable traffic analysis [24] and management strategies [6]. Embedded vision processing has also been used for security and face recognition, where various efficient implementation techniques have been studied, including how to best use combinations of cloud and edge processing [22].

2 Papers in this special issue

This special issue features three papers. The first, entitled "*Susceptibility to Image Resolution in Face Recognition and Training Strategies to Enhance Robustness*" by Knoche, Hörmann and Rigoll studies how facial recognition algorithms are susceptible to the resolution of the input images. They show that recognition accuracy can dramatically drop when the image resolution is reduced. Since training images and input images that need to be recognized might not have the same resolution, this finding poses a serious problem. To addresses this, they have proposed new training strategies on state-of-the-art face recognition models, which provided significant boost in accuracy and improved robustness.

The second paper, entitled "*Micro- and Macroscopic Road Traffic Analysis using Drone Image Data*" by Kruber *et al.* discusses analysis of traffic image data that is captured using drones. This includes estimating vehicle states and trajectories, and also macroscopic statistics such as traffic flow and traffic density. Finally, the third paper, "*HW-Flow: A Multi-Abstraction Level HW-CNN Codesign Pruning Methodology*" by Vemparala *et al.* is on optimizing convolutional neural network models, that are widely used for image classification, segmentation, and object detection tasks in autonomous vehicles. Towards this, the paper proposes a framework referred to as *HW-Flow* that achieved around 2x reduction in energy and latency in a number of well-known neural networks and datasets.

We hope that readers will find these articles to be interesting and will gain new insights into this evolving area of embedded systems for computer vision. We thank everyone who submitted their research to this special issue. We also thank all the reviewers, the EiC – Alan Burns – and members of the LITES Editorial Office, especially Michael Wagner, without whose help this special issue would not have been possible.

References

1 Tanya Amert and James H. Anderson. Cupid$^{\mathrm{rt}}$: Detecting improper GPU usage in real-time applications. In *24th IEEE International Symposium on Real-Time Distributed Computing (ISORC)*, 2021.

2 Tanya Amert, Michael Balszun, Martin Geier, F. Donelson Smith, James H. Anderson, and Samarjit Chakraborty. Timing-predictable vision processing for autonomous systems. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

3 Tanya Amert, Zelin Tong, Sergey Voronov, Joshua Bakita, F. Donelson Smith, and James H. Anderson. Timewall: Enabling time partitioning for real-time multicore+accelerator platforms. In *42nd IEEE Real-Time Systems Symposium (RTSS)*, 2021.

4 Michael Balszun, Martin Geier, and Samarjit Chakraborty. Predictable vision for autonomous systems. In *23rd IEEE International Symposium on Real-Time Distributed Computing (ISORC)*, 2020.

5 Wanli Chang and Samarjit Chakraborty. Resource-aware automotive control systems design: A cyber-physical systems approach. *Found. Trends Electron. Des. Autom.*, 10(4):249–369, 2016.

6 Wanli Chang, Debayan Roy, Shuai Zhao, Anuradha Annaswamy, and Samarjit Chakraborty. Cps-oriented modeling and control of traffic signals using adaptive back pressure. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020.

7 Sayandip De, Sajid Mohamed, Konstantinos Bimpisidis, Dip Goswami, Twan Basten, and Henk Corporaal. Approximation trade offs in an image-based control system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020.

8 Sayandip De, Sajid Mohamed, Dip Goswami, and Henk Corporaal. Approximation-aware design of an image-based control system. *IEEE Access*, 8:174568–174586, 2020.

9 Glenn A. Elliott, Bryan C. Ward, and James H. Anderson. GPUSync: A framework for real-time GPU management. In *34th IEEE Real-Time Systems Symposium (RTSS)*, 2013.

10 Glenn A. Elliott, Kecheng Yang, and James H. Anderson. Supporting real-time computer vision workloads using OpenVX on Multicore+GPU platforms. In *IEEE Real-Time Systems Symposium (RTSS)*, 2015.

11 Martin Geier, Marian Brändle, and Samarjit Chakraborty. Insert & save: Energy optimization in IP core integration for FPGA-based real-time systems. In *27th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021.

12 Martin Geier, Marian Brändle, Dominik Faller, and Samarjit Chakraborty. Debugging FPGA-accelerated real-time systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2020.

13 Martin Geier, Dominik Faller, Marian Brändle, and Samarjit Chakraborty. Cost-effective energy monitoring of a Zynq-based real-time system including dual Gigabit Ethernet. In *27th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2019.

14 Martin Geier, Florian Pitzl, and Samarjit Chakraborty. GigE vision data acquisition for visual servoing using SG/DMA proxying. In *14th ACM/IEEE Symposium on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, 2016.

15 Abhinav Goel, Caleb Tung, Xiao Hu, George K. Thiruvathukal, James C. Davis, and Yung-Hsiang Lu. Efficient computer vision on edge devices with pipeline-parallel hierarchical neural networks. In *27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2022.

16 Kees Goossens, Arnaldo Azevedo, Karthik Chandrasekar, Manil Dev Gomony, Sven Goossens, Martijn Koedam, Yonghui Li, Davit Mirzoyan, Anca Mariana Molnos, Ashkan Beyranvand Nejad, Andrew Nelson, and Shubhendu Sinha. Virtual execution platforms for mixed-time-criticality systems: the compsoc architecture and design flow. *SIGBED Rev.*, 10(3):23–34, 2013.

17 Dip Goswami, Reinhard Schneider, and Samarjit Chakraborty. Re-engineering cyber-physical control applications for hybrid communication protocols. In *Design, Automation and Test in Europe (DATE)*, 2011.

18 Dip Goswami, Reinhard Schneider, and Samarjit Chakraborty. Relaxing signal delay constraints in distributed embedded controllers. *IEEE Trans. Control. Syst. Technol.*, 22(6):2337–2345, 2014.

19 Clara Hobbs, Debayan Roy, Parasara Sridhar Duggirala, F. Donelson Smith, Soheil Samii, James H. Anderson, and Samarjit Chakraborty. Perception computing-aware controller synthesis for autonomous systems. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

20 Omar W Ibraheem, Arif Irwansyah, Jens Hagemeyer, Mario Porrmann, and Ulrich Rueckert. A resource-efficient multi-camera GiGE vision IP core for embedded vision processing platforms. In *IEEE International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, 2015.

21 Chaitanya Jugade, Daniel Hartgers, Phan Dúc Anh, Sajid Mohamed, Mojtaba Haghi, Dip Goswami, Andrew Nelson, Gijs van der Veen, and Kees Goossens. An evaluation framework for vision-in-the-loop motion control systems. In *27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022.

22 Anis Koubaa, Adel Ammar, Anas Kanhouch, and Yasser AlHabashi. Cloud versus edge deployment strategies of real-time face recognition inference. *IEEE Transactions on Network Science and Engineering*, 9(1):143–160, 2021.

23 Friedrich Kruber, Eduardo Sánchez Morales, Samarjit Chakraborty, and Michael Botsch. Vehicle position estimation with aerial imagery from unmanned aerial vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

24 Friedrich Kruber, Jonas Wurst, Eduardo Sánchez Morales, Samarjit Chakraborty, and Michael Botsch. Unsupervised and supervised learning with the random forest algorithm for traffic scenario clustering and classification. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

25 Zhen Liu, Wenjie Lin, Xinpeng Li, Qing Rao, Ting Jiang, Mingyan Han, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Adnet: Attention-guided deformable convolutional network for high dynamic range imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

26 Dipan Kumar Mandal, Jagadeesh Sankaran, Akshay Gupta, Kyle Castille, Shraddha Gondkar, Sanmati Kamath, Pooja Sundar, and Alan Phipps. An embedded vision engine (EVE) for automotive vision processing. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.

27 Alejandro Masrur, Sebastian Drössler, Thomas Pfeuffer, and Samarjit Chakraborty. VM-based real-time services for automotive control applications. In *16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2010.

28 Mahmoud Méribout, Asma Baobaid, Mohamed Ould-Khaoua, Varun Kumar Tiwari, and Juan Pablo Pena. State of art iot and edge embedded systems for real-time machine vision applications. *IEEE Access*, 10:58287–58301, 2022.

29 Seifeddine Messaoud, Soulef Bouaafia, Amna Maraoui, Ahmed Chiheb Ammari, Lazhar Khriji, and Mohsen Machhout. Deep convolutional neural networks-based hardware-software on-chip system for computer vision application. *Comput. Electr. Eng.*, 98:107671, 2022.

30 Sajid Mohamed, Dip Goswami, Vishak Nathan, Raghu Rajappa, and Twan Basten. A scenario- and platform-aware design flow for image-based control systems. *Microprocess. Microsystems*, 75:103037, 2020.

31 Sajid Mohamed, Nilay Saraf, Daniele Bernardini, Dip Goswami, Twan Basten, and Alberto Bemporad. Adaptive predictive control for pipelined multiprocessor image-based control systems considering workload variations. In *59th IEEE Conference on Decision and Control (CDC)*, 2020.

32 Burhan Ahmad Mudassar, Priyabrata Saha, Yun Long, Mohammad Faisal Amir, Evan Gebhardt, Taesik Na, Jong Hwan Ko, Marilyn Wolf, and Saibal Mukhopadhyay. CAMEL: an adaptive camera with embedded machine learning-based sensor parameter control. *IEEE J. Emerg. Sel. Topics Circuits Syst.*, 9(3):498–508, 2019.

33 The Khronos Group, OpenVX: Portable, power efficient vision processing. `https://www.khronos.org/openvx/`.

34 Nathan Otterness and James H. Anderson. AMD GPUs as an alternative to NVIDIA for supporting real-time workloads. In *32nd Euromicro Conference on Real-Time Systems (ECRTS)*, volume 165 of *LIPIcs*, pages 10:1–10:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

35 Qing Rao and Samarjit Chakraborty. Efficient lossless compression for depth information in traffic scenarios. *Multim. Syst.*, 25(4):293–306, 2019.

36 Qing Rao and Samarjit Chakraborty. In-vehicle object-level 3D reconstruction of traffic scenes. *IEEE Trans. Intell. Transp. Syst.*, 22(12):7747–7759, 2021.

37 Qing Rao, Christian Grünler, Markus Hammori, and Samarjit Chakraborty. Design methods for augmented reality in-vehicle infotainment systems. In *51st Annual Design Automation Conference (DAC)*, 2014.

38 Qing Rao, Christian Grünler, Markus Hammori, and Samarjit Chakraborty. Stixel on the bus: An efficient lossless compression scheme for depth information in traffic scenarios. In *20th Anniversary International Conference of MultiMedia Modeling (MMM)*, volume 8325 of *Lecture Notes in Computer Science*. Springer, 2014.

39 Qing Rao, Lars Krüger, and Klaus Dietmayer. Monocular 3D shape reconstruction using deep neural networks. In *IEEE Intelligent Vehicles Symposium (IV)*, 2016.

40 Qing Rao, Tobias Tropper, Christian Grünler, Markus Hammori, and Samarjit Chakraborty. AR-IVI - implementation of in-vehicle augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014.

41 Debayan Roy, Clara Hobbs, James H. Anderson, Marco Caccamo, and Samarjit Chakraborty. Timing debugging for cyber-physical systems. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

42 Debayan Roy, Licong Zhang, Wanli Chang, Sanjoy K. Mitter, and Samarjit Chakraborty. Semantics-preserving cosynthesis of cyber-physical systems. *Proc. IEEE*, 106(1):171–200, 2018.

43 Kruttidipta Samal, Hemant Kumawat, Priyabrata Saha, Marilyn Wolf, and Saibal Mukhopadhyay. Task-driven rgb-lidar fusion for object tracking in resource-efficient autonomous system. *IEEE Trans. Intell. Veh.*, 7(1):102–112, 2022.

44 Kruttidipta Samal, Marilyn Wolf, and Saibal Mukhopadhyay. Closed-loop approach to perception in autonomous system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

45 Róbinson Medina Sánchez, Juan Valencia, Sander Stuijk, Dip Goswami, and Twan Basten. Designing a controller with image-based pipelined sensing and additive uncertainties. *ACM Trans. Cyber Phys. Syst.*, 3(3):33:1–33:26, 2019.

46 Sebastian Schmidt, Qing Rao, Julian Tatsch, and Alois C. Knoll. Advanced active learning strategies for object detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

47 Reinhard Schneider, Dip Goswami, Alejandro Masrur, Martin Becker, and Samarjit Chakraborty. Multi-layered scheduling of mixed-criticality cyber-physical systems. *J. Syst. Archit.*, 59(10-D):1215–1230, 2013.

48 Shreejith Shanker, Philipp Mundhenk, Andreas Ettner, Suhaib A. Fahmy, Sebastian Steinhorst, Martin Lukasiewycz, and Samarjit Chakraborty. VEGa: A high performance vehicular ethernet gateway on hybrid FPGA. *IEEE Trans. Computers*, 66(10):1790–1803, 2017.

49 Ponnan Suresh, Saravanakumar Umathurai, Celestine Iwendi, Senthilkumar Mohan, and Gautam

Srivastava. Field-programmable gate arrays in a low power vision system. *Comput. Electr. Eng.*, 90:106996, 2021.

50 George K. Thiruvathukal and Yung-Hsiang Lu. Efficient computer vision for embedded systems. *Computer*, 55(4):15–19, 2022.

51 Caleb Tung, Abhinav Goel, Fischer Bordwell, Nick Eliopoulos, Xiao Hu, Yung-Hsiang Lu, and George K. Thiruvathukal. Why accuracy is not enough: The need for consistency in object detection. *IEEE Multim.*, 29(3):8–16, 2022.

52 R. Udendhran, M. Balamurugan, Annamalai Suresh, and R. Varatharajan. Enhancing image processing architecture using deep learning for embedded vision systems. *Microprocess. Microsystems*, 76, 2020.

53 Zhilu Wang, Chao Huang, Yixuan Wang, Clara Hobbs, Samarjit Chakraborty, and Qi Zhu. Bounding perception neural network uncertainty for safe control of autonomous systems. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

54 Peter Waszecki, Philipp Mundhenk, Sebastian Steinhorst, Martin Lukasiewycz, Ramesh Karri, and Samarjit Chakraborty. Automotive electrical and electronic architecture security via distributed in-vehicle traffic monitoring. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 36(11):1790–1803, 2017.

55 Ming Yang, Tanya Amert, Kecheng Yang, Nathan Otterness, James H. Anderson, F. Donelson Smith, and Shige Wang. Making openvx really "real time". In *IEEE Real-Time Systems Symposium (RTSS)*, 2018.

56 Ming Yang, Shige Wang, Joshua Bakita, Thanh Vu, F. Donelson Smith, James H. Anderson, and Jan-Michael Frahm. Re-thinking CNN frameworks for time-sensitive autonomous-driving applications: Addressing an industrial challenge. In *25th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2019.

57 Anand Yeolekar, Ravindra Metta, Clara Hobbs, and Samarjit Chakraborty. Checking scheduling-induced violations of control safety properties. In *20th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 13505 of *Lecture Notes in Computer Science*. Springer, 2022.

58 Junxing Zhang, Shuo Yang, Chunjuan Bo, and Zhiyuan Zhang. Vehicle logo detection based on deep convolutional networks. *Comput. Electr. Eng.*, 90:107004, 2021.