Visual IoT: Architectural Challenges and Opportunities

RAVI IYER
Intel

The emergence of ultra-low-power sensing devices along with connectivity to gateways and cloud services has led to an end-to-end Internet of Things (IoT) architecture for many real-world usages. Visual IoT is one such class of IoT that poses significant end-to-end challenges due to the need for sensing and processing of visual data. The richness of visual data provides many opportunities for analytics, while at the same time requiring high computational capabilities and therefore potentially high-bandwidth data transfer to a more powerful node in the end-to-end architecture. Memory and storage needs are also more pronounced in visual IoT solutions, requiring careful thought to developing an intelligent memory hierarchy for visual storage and retrieval. In this article, I examine the computing, memory, and interface implications for end-to-end visual IoT architectures and discuss potential solutions and tradeoffs in each of these areas. Before we start, let's go over a brief overview of visual IoT usage domains.

Visual IoT Overview

Beyond photography, cameras have been used widely in multiple domains, ranging from security (for example, surveillance and monitoring), entertainment (recording of public and personal events, such as sports and music), and, more recently, interactive environments (augmented, virtual, and merged reality; see www.intel.com/content/www/us/en/architecture-and-technology/virtual-reality-overview.html) and robotics and drones[1] (navigation, delivery, interaction, and assistance). With the emergence of depth-sensing cameras, such as Intel RealSense (www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html), analysis of the captured visual scene becomes even more attractive for many of these scenarios.

In many of these scenarios, three types of platforms compose the end-to-end IoT architecture (see Figure 1):

- visual sensing nodes that capture the data and potentially do some local processing;
- gateways, phones, or on-premise platforms that can stage the data and provide higher computing capability; and
- cloud servers that provide services for search, analytics, or simply storage.

Much like real estate, the key to efficiently architecting a visual IoT architecture is location (where to perform the computation), location (where to store the data), and location (where to enable interfaces and tools for analytics). Let's start by examining the computing location challenge and then move to memory and interfaces.

Editors' note: We invited two industry experts to discuss the opportunities and challenges surrounding the Internet of Things. The following are their views on the topic. —*Vijay Janapa Reddi and Hyesoon Kim*

Computing in Visual IoT: Partitioning and Heterogeneity

Partitioning the work in an end-to-end visual IoT architecture is a challenge, because it requires the balancing of multiple important dimensions:

- the sensing node's battery life,
- the latency of the interaction,
- throughput benefits on the server versus bandwidth costs of the transfer, and
- security and privacy implications of the data.

The partitioning problem is essentially across heterogeneous platforms as well as within heterogeneous processing elements (cores versus GPUs versus accelerators) within a platform. Also, the application can dictate the partitioning strategy in some cases, in which a subset of operations on the sensor node can deliver some minimal useful experience, while the processing at the server is used for heavier computations.

Let's take an example scenario of a visual agent monitoring a home environment. The key aspects of a home agent include

1. anomaly detection,
2. saliency and summarization,

0272-1732/16/$33.00 © 2016 IEEE

Published by the IEEE Computer Society

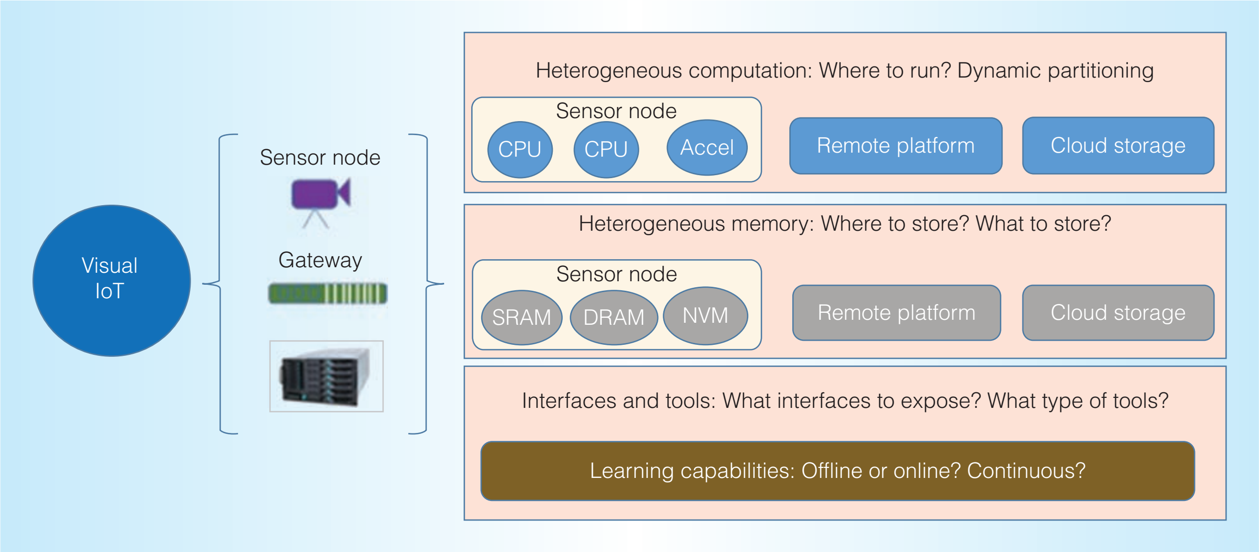
Visual IoT
Sensor node
Gateway
Heterogeneous computation: Where to run? Dynamic partitioning
Sensor node
CPU
CPU
Accel
Remote platform
Cloud storage
Heterogeneous memory: Where to store? What to store?
Sensor node
SRAM
DRAM
NVM
Remote platform
Cloud storage
Interfaces and tools: What interfaces to expose? What type of tools?
Learning capabilities: Offline or online? Continuous?

Figure 1. Visual IoT. Example implications on architecture research include how to dynamically partition the computation across the heterogeneous architectures, manage the memory across the end-to-end system, and integrate offline/online learning capabilities and tools.

3. detection of patterns of behavior, and
4. recognition and interaction with a person in the home from a Q&A standpoint.

The scenario becomes even more complicated if the visual agent is mobile (like a robot) versus a static visual agent in which the backgrounds can be predetermined. For aspects 1 and 4, the response time is critical, so local processing is desirable, whereas for aspects 2 and 3, batch processing is more useful because of the large amount of data needed before processing.

Such scenarios are common and require careful examination of whether the processing can be performed on the computing core of the sensor node itself, offloaded to a local accelerator, sent to a gateway within the home, or offloaded to a private cloud where the analysis can be accomplished. A static solution would end up determining how to employ the most efficient engine (fixed function or configurable accelerator) at each node in the end-to-end architecture. Instead of statically determining this heterogeneous architecture and partitioning balance, a dynamic partitioning solution is even more suitable if the solution has to be customized for different homes and similar environments. As a result, solutions such as remote offloading are becoming more important from a flexibility and customization point of view.[2] Research and development in heterogeneous architectures with partitioning capabilities that retain flexibility while maximizing efficiency and customizability will continue to be predominant for visual IoT, as well as other rich environments.

Memory in Visual IoT: Saliency, Storage, and Hierarchy

Another major challenge in the end-to-end visual IoT architecture is management of the visual data. Although the richness of visual data is attractive, it is also true that much of the visual data captured can be potentially discarded, and only a summary typically needs to be retained. The key is to figure out what visual data summary must be retained by potentially extracting the salient segments of the visual stream. The basis for saliency depends entirely on the usage in question. For example, in the home scenario, the salient aspects might be the key activities that happened throughout the day and the anomalies and novel occurrences that were identified. In a recent study,[3] the authors demonstrated the ability to summarize a video by optimizing for similarity and coverage. Analyzing such algorithms and capabilities and converting them into appropriate computing and memory implementations[4] is a critical area of research for future visual sensor nodes, as well as the gateways and servers that maintain the data.

Beyond saliency and summarization of frames, it is also critical to identify key entities and activities in visual data to enable fast search and indexing. The question becomes what metadata needs to be extracted and where such metadata should be stored (on the sensor node, in the gateway, or on the cloud server). In addition, there is a question of what type of memory is most suitable for the metadata in question. This calls for an end-to-end heterogeneous memory architecture consisting of different memory types, ranging from cache to DRAM to nonvolatile memory to storage. Identifying the right balance of such heterogeneous memory across each of the nodes in the end-to-end architecture is critical as visual IoT usages explode and cause bandwidth challenges for retrieval of data.

Interfaces and Tools for Visual IoT: Learning and Development

Finally, it is important to consider appropriate interfaces for visual IoT platforms. For example, as machine learning techniques get adopted to analyze sensor data, it becomes important to understand how to take advantage of both offline and online learning techniques. As an example, if the visual agent wants to understand gestures made by the people in a home, it is extremely useful to enable interfaces and tools that allow the agent to train on and download these capabilities. By making such capabilities broadly available, developers will be able to provide many analytics capabilities employing rich sensor data and potentially crowdsourced training data. Especially as sensor nodes become more capable (such as the Intel Curie Module[5] with pattern matching capability), new tools that enable developers to use such capabilities (such as the Intel Knowledge Builder toolkit [http://software.intel.com/en-us/intel-knowledge-builder-toolkit]) are critical for the rapid deployment of IoT solutions.

Visual IoT is a rapidly growing class of usages with the proliferation of smart cameras with increasing capabilities. Future areas of research include developing heterogeneous architectures and dynamic partitioning capabilities across end-to-end visual IoT, examining heterogeneous memory stores for visual data management and retrieval, and tools and interfaces for fast deployment of analyzing visual and other IoT solutions. MICRO

References

1. K. Kaplan, "The Future of Drones: Market Prepares for Takeoff," Intel, Sept. 2016; http://iq.intel.com/drone-economy-prepares-takeoff.
2. H. Eom et al., "OpenCL-Based Remote Offloading Framework for Trusted Mobile Cloud Computing," *Proc. Int'l Conf. Parallel and Distributed Systems*, 2013, pp. 240–248.
3. S. Chakraborty, O. Tickoo, and R. Iyer, "Adaptive Keyframe Selection for Video Summarization," *Proc. IEEE Winter Conf. Applications of Computer Vision*, 2015, pp. 702–709.
4. T. Lee et al., "Low-Complexity HOG for Efficient Video Saliency," *IEEE Int'l Conf. Image Processing*, 2015; doi:10.1109/ICIP.2015.7351505.
5. "Intel Curie Module & Intel IQ SW Fact Sheet," Intel, Aug. 2015; www.intel.com/content/www/us/en/wearables/intel-curie-fact-sheet.html.

Ravi Iyer is a senior principal engineer, CTO, and director in New Business Initiatives at Intel. He is an IEEE Fellow. Contact him at ravishankar.iyer@intel.com.

Toward a Self-Learning and Energy-Neutral IoT

Emre Ozer
ARM

A typical Internet of Things (IoT) device comprises five components: sensor, microcontroller, memory, battery/energy harvester, and radio. It is a device that collects, preprocesses, stores, and transfers data received from a sensor to a host (for example, a reader via RFID, a smartphone via Bluetooth, or the cloud through a gateway) wherein data processing is performed. The microcontroller is mainly responsible for control and simple data preprocessing, and the radio is used to transmit short data packets. Hence, the battery can last for years before it is recharged or replaced.

Such a simple IoT device is no longer adequate because emerging applications (such as medical, structural/environmental monitoring, and e-textiles) demand ambient intelligence or cognition and real-time response from IoT devices. A new class of IoT devices called *self-learning IoT devices* will emerge to provide cognitive services, such as situational awareness, anomaly detection, activity, and pattern and emotion recognition, which are essentially machine learning algorithms. Real-time response from self-learning IoT devices is needed because an anomaly or critical activity must be detected or recognized in situ and reported immediately, because transmitting the sensor data via radio to its host to do this will be costly in terms of energy and latency.[1] For this reason, the cognitive action must take place in the device, not in the host. For example, an implantable chip must detect an abnormal condition in the organ and must take an action in real time. It cannot afford to wait for a critical decision to be made by the host.

A self-learning IoT device will accommodate multiple sensors and a more powerful computation engine to perform computationally intensive sensor fusion and to run machine learning algorithms. It will need a good size on-device memory (SRAM and nonvolatile memory) to store the code and buffer the streaming sensor data before and after data fusion. Integrating multiple sensors, a relatively higher-performance computation engine, and more on-chip storage in a self-learning IoT device will consume more energy than a simple IoT device, and will put incredible pressure on the battery. Self-learning IoT devices will be deployed in such environments in which recharging or replacing the battery is not possible—for example, an IoT device implanted in a human body, integrated into a building's foundation, or embedded in the textile fabric. Hence, the battery in the device must operate for a long time (for example, more than 10 years) and be charged by multiple energy harvesters that are integrated into the device to harvest ambient energy (such as thermal, vibration, solar, pressure) in order to charge the battery. This is a concept called *energy neutrality,*[2] in which the battery will always be charged by energy harvesters in the device[3] and should never be recharged by human intervention.

The next phase in the IoT's evolution is self-learning and energy-neutral devices having the properties of cognition, real-time response, and perpetual energy. The main challenge is to run computation and memory-intensive sensor fusion and machine learning algorithms in a device powered only by the harvested energy. This opens up opportunities to design novel computation engines, memory subsystems, and energy management units, considering not only energy efficiency but also energy neutrality. The computation engine in the device must be equipped with single-instruction, multiple data/digital signal processing capabilities and be coupled with one or more machine learning hardware accelerators (such as a deep neural network). Alternatively, the computation engine can be tightly coupled with sensors that will stream data directly to the computation engine, such that it may have an analog preprocessing front end tightly coupled to the sensors, and the machine learning algorithms will run on the engine's digital back end engine. Today, state-of-the-art microcontrollers have up to 4 Mbytes of flash memory and much less static RAM, and future IoT devices will not have orders of magnitude larger on-chip storage because of the cost issues. Machine learning algorithms—in particular, deep neural networks—take up a significant storage space, so it will be a challenge to store a large number of network parameters on chip. Hardware and software compression techniques will be used to deal with the large parameter space in deep neural networks. Also, alternative machine learning algorithms that are more adaptive, resource efficient, and energy efficient (such as nonparametric Bayesian methods[4]) can be developed for self-learning and energy-neutral IoT devices. The management of the harvested energy is a critical process, and the data must be sensed, fused, stored, and processed, and the response given, before the harvested energy in the battery depletes. The harvested energy management must be performed by a combination of innovative software and hardware techniques, such as the prediction of the harvested energy before task execution, or new instructions to control the energy harvesting process.

Self-learning and energy-neutral IoT devices will also emerge in the printed electronics world.[5] Printed electronics offers cost-effective fabrication of electronics with low-cost substrates and materials (such as plastic and paper), simpler processing and patterning steps, and disposability. It has found applications in sensors, RFIDs, solar cells, batteries, and displays in the fields of medical, wearable, textile, automotive, and packaging applications. Smart printed devices have already been demonstrated as smart tags, labels, packages, e-textiles, and wearables. For example, T.E. Halterman built a printed alarm armband that monitors the vital signs of hospital patients.[6] The flexible armband contains a solar panel, piezoelectric speaker, temperature sensor, and power supply circuit, all of which are organic components in a wearable form factor. It is self-powered by the solar panel, and the speaker sounds an alarm when the temperature sensor measures a temperature between 36.5 to 38.5 degrees Celsius. These early demonstrators are the precursors of future self-learning and energy-neutral printed IoT devices. The main advantage of printed electronics is that they allow low-cost customization thanks to the low-cost flexible substrate and materials, and do not require costly clean rooms, unlike silicon. This offers a unique opportunity, in particular, to customize the computation engine to the needs of the cognitive application that will be running on the device. For example, an energy-efficient support vector machine (SVM) can be designed as a custom computation engine (rather than using a less energy-efficient general-purpose computation engine) and printed for a single-use smart packaging product, because it will run only the SVM. This will not be possible in silicon, because customization (that is, ASIC) is extremely costly. Thus, printed electronics will pave the way to low-cost customization of efficient computation engines for future printed self-learning and energy-neutral IoT devices.

Future IoT devices will become more intelligent and aware of their environment, and will integrate more capable computation engines to perform cognitive activities. However, these devices will still be constrained by energy efficiency and limited energy capacity, as in today's dumb IoT devices. They will be so deeply embedded that they will not be accessible to replace or recharge their batteries, and will have to depend on energy harvesters to become self-sustained or energy-neutral. This will be even more prominent for printed electronic devices that will be manufactured for a single use. The main engineering challenge is to design an energy-neutral device that will be deployed in critical missions and stay operational for a long time, but at the same time run computationally complex machine learning algorithms. Nevertheless, this challenge brings up unique opportunities for system architects, designers, and software developers to come up with holistic solutions not only for the self-learning and energy-neutral IoT devices in silicon, but also in emerging printed electronics. MICRO

References

1. R.C. Carrano et al., "Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, 2014, pp. 181–194.
2. M. Magno et al., "Infinitime: A Multisensor Energy Neutral Wearable Bracelet," International Green Computing Conference (IGCC), 2014.
3. A.S. Weddell et al., "A Survey of Multisource Energy Harvesting Systems," *Proc. Conf. Design, Automation and Test in Europe*, 2013, pp. 905–908.
4. Y. Raykov et al., "Predicting Room Occupancy with a Single Passive Infrared (PIR) Sensor through Behavior Extraction," *Proc. ACM Int'l Jt. Conf. Pervasive and Ubiquitous Computing*, 2016, pp. 1016–1027.
5. S. Khan, L. Lorenzelli, and R. Dahiya, "Technologies for Printing Sensors and Electronics over Large Flexible Substrates: A Review," *IEEE Sensors J.*, vol. 15, 2015, pp. 3164–3181.
6. T.E. Halterman, "Flexible, 3D Printed, Solar Powered Thermal Alarm for Patient Monitoring," *3D Print*, 26 Feb. 2015; http://3dprint.com/47116/3d-printed-fever-alarm.

Emre Ozer is a principal research engineer at ARM Research in Cambridge. Contact him at emre.ozer@arm.com.