Camaroptera: a Batteryless Long-Range Remote Visual Sensing System

Matteo Nardello
matteo.nardello@unitn.it
University of Trento
Trento, Italy

Harsh Desai
harshd@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

Davide Brunelli
davide.brunelli@unitn.it
Univeristy of Trento
Trento, Italy

Brandon Lucia
blucia@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

ABSTRACT

Batteryless image sensors present an opportunity for pervasive wide-spread remote sensor deployments that require little maintenance and have low cost. However, the reliance of these devices on energy harvesting presents tight constraints in the quantity of energy that can be stored and used, as well as limited, energy-dependent availability. In this work, we develop Camaroptera, the first batteryless, energy-harvesting image sensing platform to support active, long-range communication. Camaroptera reduces the high latency and energy cost of communication by using near-sensor processing pipelines to identify interesting images and transmit them to a far-away base station, while discarding uninteresting images. Camaroptera also dynamically adapts its processing pipeline to maximize system availability and responsiveness to interesting events in different harvesting conditions. We fully prototype the Camaroptera hardware platform in a compact, 2cm x 3cm x 5cm volume, composed of three adjoined circuit boards. We evaluate Camaroptera demonstrating the viability of a batteryless remote sensing platform in a small package. We show that compared to a system that transmits all image data, Camaroptera's processing pipelines and adaptive processing scheme captures and sends 2-5X more images of interest to an application.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; *Embedded software*; • **Hardware →** *Wireless integrated network sensors*; • **Computing methodologies → Object detection**; *Neural networks.*

KEYWORDS

Energy-harvesting, Intermittent computing, Edge computing, Computer vision, Sensor systems

1 INTRODUCTION

Improvements in energy-harvesting systems have led to the emergence of wireless IoT devices that are entirely *energy-autonomous*. Such a system collects energy (e.g., RF, solar) from its environment, buffering the energy in a battery [1] or capacitor [2]. After collecting sufficient energy, the system activates and performs some sensing, computing, or communication for its application.

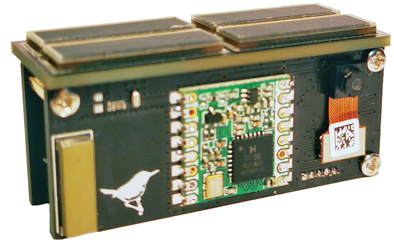

Figure 1: Camaroptera prototype

Existing ultra-low-power and batteryless IoT devices typically support limited sensing, computing and communication modalities. Today's systems primarily sense low-data-rate time-series information using sensors like accelerometers, barometers, hydrometers, and gyroscopes. These devices perform low-intensity computation, such as summarization (e.g., averaging), compression, and logging and communicate over short distances (meters), using off-the-shelf radios, such as Bluetooth Low-Energy (BLE). Unfortunately, future applications, such as signals intelligence, enhanced situational awareness, wildlife monitoring, and smart cities require more sophisticated sensing, computing, and communication modalities. Future systems must support *video and image* sensor data to directly, rather than indirectly, observe complex environmental phenomena and furthermore demand deployment into environments requiring wireless data backhaul at kilometer scale. The tandem requirements of high-data-rate sensing and long-range communication pose a key challenge for an energy-constrained system: how should a device use scarce bandwidth and energy to communicate interesting signals to a far-away base station?

In this paper, we present the development of Camaroptera, a batteryless remote image sensing system, which addresses all of these future application requirements. Camaroptera senses visual spectrum data using an ultra-low-power image sensor [3] and is equipped with a LoRa [4] radio, enabling it to communicate over long distances, even in the presence of urban signal occlusion [5]. Camaroptera is batteryless and harvests its operating energy using small solar panels, storing energy in a small supercapacitor.

Camaroptera's main contribution is two-fold. Camaroptera's first contribution is to address the challenge of energy and bandwidth scarcity by instead spending energy to apply sophisticated *near-sensor processing* to sensed data. Camaroptera processes images locally and only transmits the interesting results to a base station. Camaroptera's flexible software pipeline architecture allows deploying signal processing (e.g., compression, filtering), and machine

ENSsys '19, November 10, 2019, New York, NY

ACM ISBN 978-1-4503-7010-3/19/11…$15.00
https://doi.org/10.1145/3362053.3363491

inference using convolutional/deep neural networks (CNN/DNNs). Based on the amount of ambient energy available, Camaroptera can dynamically vary the processing it performs on input data. This allows Camaroptera to optimize the system availability by performing cheaper and quicker operations when there is less energy available and employing more sophisticated techniques when there is larger energy availability.

Camaroptera's second contribution is to develop a fully- functional, miniaturized energy-harvesting prototype. The 3D assembly of the multi-board prototype fits a 2cm x 3cm x 5cm package. The prototype works within these tight volume and surface area constraints, which limit solar panel output to a few mW and energy storage volume (e.g., 33mF at 3V). The prototype supports sophisticated software pipelines within the resource constraints of a commodity ULP MCU: 256kB of total memory, a 16MHz clock, and a 16-bit datapath.

Our measurements show that Camaroptera is a viable batteryless remote image sensor, supporting long-range communication, by leveraging efficient on-sensor computational pipelines. We first present the design overview and describe how Camaroptera answers several important research questions in meeting key design requirements. We then measure Camaroptera's basic operation in a lab environment, demonstrating its ability to collect, process, and selectively transmit image sensor data, operating entirely from its limited solar power input.

2 BACKGROUND

Energy-harvesting systems are devices that extract their operating energy from their environment, eliminating their dependence on batteries by instead using highly durable and long-lived capacitive energy storage. An energy-harvesting system may extract energy from directed or ambient radio waves [6, 7], solar radiation [2, 8], or other environmental sources and operates only intermittently, as energy is available [9–15]. Energy-harvesting devices support long-term deployments because they are limited by capacitor and IC lifetimes only, rather than relatively shorter battery lifetimes.

Recent work [1] advocates for using batteries to avoid managing capacitive energy storage and to store surplus energy. While simpler and appropriate for some tasks with low duty cycle and low compute intensity, battery-powered devices are larger, heavier, contribute to battery waste, and face lifetime issues, as fixed batteries fail and rechargeable batteries wear out with recharge cycles. Building energy-harvesting systems that rely on capacitive energy storage is an appealing design to support very large numbers of sensor systems to be deployed pervasively for long periods of time and to support high duty cycle and high-intensity computation.

In addition to freedom from energy infrastructure, pervasively deployed IoT applications must operate at potentially long distances from communications infrastructure. The recent maturation of chirp spread-spectrum long-range radio technologies, such as LoRa [16]/LoRaWAN [17], has given rise to long-range sensor data backhaul options such as OpenChirp [5]. LoRa ICs are commercially available, inexpensive, and offer communication over extremely long distances (i.e., kilometers) at very low transmit power levels (i.e., tens of mW).

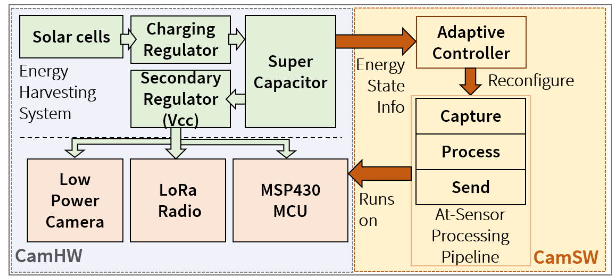

Figure 2: System overview of Camaroptera

Additionally, the ability to communicate kilometers from a tiny, energy-harvesting sensor device creates the opportunity for more devices to be deployed to more environments than is possible using legacy backhauls: for example, 4G/LTE is expensive per byte [18], Bluetooth/BLE is range-limited [19, 20], and wifi requires deploying many base stations for wide-area coverage.

While long-range communication technology is a key enabler of pervasively deployed IoT devices, the energy required for communication dominates a device's total operating energy, even at a low communication duty cycle.

Computing directly on sensor data using compute hardware co-located with those sensors reduces the need to incur the high energy cost of communication using a high-power radio. While the computational capabilities of existing ultra-low-power microcontrollers is limited, prior work has shown promise in evaluating even sophisticated sensor data processing workloads on intermittent, energy-harvesting systems. One recent study [21] found that the time to collect energy needed to communicate a 28×28 pixel image was approximately 360× greater than the time to collect energy needed to intermittently evaluate a convolutional neural network on that image. With architectural support for yet more efficient computing in ultra-low-power MCUs [22–24], the computational capabilities available to intermittent, energy-harvesting sensor nodes will increase.

3 CAMAROPTERA DESIGN

Camaroptera is a batteryless image sensor designed to support pervasive, long-term deployment. Its design is motivated by the unique challenges of deploying a high-data-rate sensor over a large area for a long time. We identify key design requirements for such a deployment and provide an overview of Camaroptera's design, which meets these goals.

3.1 Camaroptera Design Requirements

There are several key design requirements for a batteryless remote sensing system. These requirements are **(R1)** low cost, **(R2)** zero maintenance, **(R3)** geographic distribution, **(R4)** bandwidth-effective operation, and **(R5)** minimal environmental impact.

(R1) Cost. A pervasively deployed image sensing system must have a low cost. Devices must be manufactured at large scale (i.e., millions) requiring each device's cost to be low to minimize total cost. Minimizing cost minimizes operator liability as deployed devices are at risk of damage or loss due to vandalism, animal interactions [25] and weather.

(R2) Maintenance. A pervasively deployed image sensing system must operate for a long period of time without maintenance. A device should require no component or battery replacements over its lifetime. A device should operate without costly centralized power or communications infrastructure; fully autonomous and wireless operation is ideal.
(R3) Distribution. A pervasively deployed image sensing system must cover a large geographic area while respecting cost and maintenance requirements in order to lower the number of high-cost base stations that require continuous power (i.e., imposing an infrastructure cost) or large batteries (i.e., imposing a maintenance cost). Long-range wireless communication is the only viable option for a large-scale, geo-distributed remote image sensing systems.
(R4) Communication. A pervasively deployed image sensing system should make judicious use of bandwidth. Long-range wireless links typically have a low data-rate [5, 17, 21] and the system should ensure that every transmitted byte is valuable to the end-to-end application that the system implements. Moreover, to respect the cost, maintenance, and distribution requirements, the system must limit the number of required long-range receivers. The more data each sensor sends to a receiver, the fewer devices each receiver can support, and the more receivers are required [26].
(R5) Environmental Impact. A pervasively deployed image sensing system must minimize its negative impact on the environment into which it is deployed by being small, unobtrusive, and by not interfering with existing (e.g., radio) infrastructure. A device must minimize its environmental impact in the long term by minimizing the amount of hazardous chemical waste due to batteries and other toxic components. Ideally, as technology permits, pervasively deployed sensors should be manufactured from biodegradable semiconductors on thin, flexible, biodegradable organic substrates [27].

3.2 Camaroptera Design Overview

Camaroptera is a batteryless sensing, computing and communication system composed of a custom hardware platform, application-level software components and an adaptive controller. Figure 2 presents a system overview of the different blocks that compose the Camaroptera visual sensing system. Each Camaroptera device is built on CamHW, a custom hardware platform. It includes a small, low-power image sensor to collect images, a microcontroller with an embedded memory to process images, a long-range radio chip for communication, and a solar energy-harvesting power system for collecting and storing energy from the environment.

The software of Camaroptera is built around CamSW, a simple operating system and device driver layer that manages sensor data collection and provides reconfigurable, *at-sensor processing pipelines* to process collected images. It includes an *adaptive controller* that monitors the device's input power using dedicated hardware and modifies the processing tasks based on energy availability. This allows Camaroptera to maximize the system availability and responsiveness to interesting events by performing cheaper and quicker processing in times of lower energy availability. When there is more energy available to harvest, CamSW will switch Camaroptera's processing pipeline to more sophisticated tasks.
Meeting Design Requirements. Together, Camaroptera's hardware and software meet the design requirements for a pervasive, long-term image sensing device. Camaroptera's design minimizes its cost. CamHW is a low-cost 2-layer custom PCB populated entirely with COTS components and ICs. Each fully-assembled device costs around USD$50, enabling large-scale deployements. Camaroptera's batteryless energy-harvesting power system is simple to deploy and requires no maintenance once deployed. Batteryless operation requires no battery replacements and produces no battery waste. CamHW includes a LoRa IC, enabling communication over kilometer ranges. Kilometer-scale communication is a key requirement for widespread deployment without excessive base station infrastructure costs. CamSW is designed for immediate, at-sensor processing of collected sensor data. Application-specific at-sensor processing allows CamSW to identify uninteresting data and to discard them immediately, avoiding consuming energy, time, and bandwidth to send them to a base station, efficiently using scarce bandwidth.

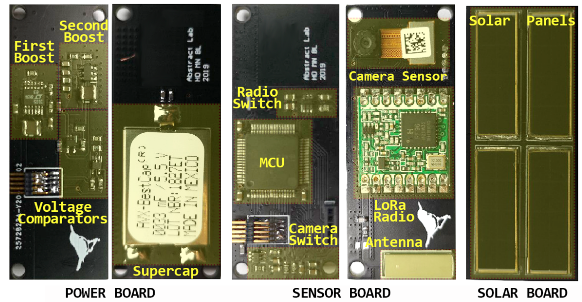

Figure 3: Camaroptera prototype PCBs

4 HARDWARE DESIGN

Camaroptera hardware platform is designed for sensing, computing, and long-range communication and is composed entirely of COTS components to limit per-device cost.

Camaroptera is composed of 3 small boards. The *sensor board* includes sensing, computing, and communication components. The *power system board* includes energy storage and power conditioning components. The *solar board* includes the device's solar panels and provides structural support for the manufactured device. Figure 3 shows a photograph of the populated sides of the three boards.

4.1 Sensor Board

The sensor board incorporates the main active components of Camaroptera's remote sensor system, including a microcontroller (MCU), a low-power image sensor, and a LoRa transceiver with a ceramic patch antenna. The sensor board hardware is agnostic to its power system and can be powered by Camaroptera's power board or by a standard 3V supply.
MCU. Camaroptera's MCU is a Texas Instruments Ultra Low-Power MCU MSP430FR5994 [28] running at 16 MHz with 8kB of SRAM and 256 kB of embedded non-volatile FRAM. The limited memory, low clock frequency, and simple architecture are key challenges to support sophisticated computations, as observed in prior work [21].
Image Sensor. Camaroptera uses a Himax HM01B0 [3] image sensor, which is an ultra-low-power CMOS image sensor. Its sensor has an active area of 320x320 pixels. Camaroptera configures the camera to operate in QQVGA (160x120) mode, capturing 8-bit grayscale images.

Transceiver. Camaroptera uses a Semtech SX1276 [4] LoRa chirp spread spectrum [16] modulation transceiver IC [29]. The chip incorporates an ultra-low-power 20 dBm power amplifier with a sensitivity over -148 dBm. Camaroptera connects this LoRa IC to a ceramic chip antenna with a maximum gain of 3.42 dBi.

4.2 Power board

The power board implements Camaroptera's energy harvesting power system, a two-stage voltage boosting circuit with hardware voltage comparators to keep system voltage in the most efficient operating range for the boosters. The power board also houses Camaroptera's supercapacitor-based energy storage.

Boosting. The first voltage boosting stage connects the solar panels to the supercapacitor using an LTC3105 [30]. The booster is a high-efficiency step-up DC/DC converter that operates down to 225mV input and supports maximum power point control (MPPC). Both of these features are important for operating on solar energy in low light. The second boosting stage connects the energy storage to the sensor board using a TPS61070 [31] synchronous voltage boost converter. This boost IC provides efficiency over 85% with input as low as 1.2V for a regulated output of 3V.

Camaroptera controls the booster operation, keeping it powered off when the supercapacitor voltage is outside its maximum efficiency region.

Voltage Thresholding. Camaroptera uses two MIC841 [32] voltage comparators with an externally adjustable hysteresis to drive the enable input of Camaroptera's second booster and the reset line of the MCU. The first comparator ensures that the second boosting stage is enabled only when the energy storage capacitor is charged between a lower threshold of 1.24V and its rated maximum of 3V. Camaroptera's second comparator holds the MCU in reset while the V_{CC} output of the second boosting stage stabilizes. The second comparator is set to a low threshold of 2.2 V, the minimum voltage to operate the LoRa modem, and a high threshold of 3V.

Energy Storage. Camaroptera stores energy in a high-density supercapacitor. Camaroptera's supercapacitor must store sufficient energy to ensure that its longest atomic task completes without exhausting the device's stored energy. Camaroptera's largest atomic task is sending a LoRa packet. We empirically determined that a BestCap [33] 33mF high-density supercapacitor with a low Equivalent Series Resistance (ESR) stores sufficient energy to send LoRa packets in several radio modes. 33mF is the largest capacitance available in the small-form-factor supercapacitor size on Camaroptera's power board; a larger capacity would require a larger supercapacitor volume and footprint size.

4.3 Solar Board

Camaroptera's solar board is entirely covered by solar panels and mounts perpendicularly to the sensor and power boards. The solar panels are an array of four IXYS [34] high-efficiency monocrystalline panels, measuring 1cm x 2cm each. The solar board provides structure and power for the assembled device with mechanical and electrical connections to the other boards.

5 SOFTWARE DESIGN

Camaroptera's software subsystem, CamSW, is centered around its use of near-sensor processing pipelines and an adaptive control mechanism that varies its operating parameters depending on the amount of incoming energy to ensure minimum quality of service requirements.

5.1 Near-Sensor processing pipeline

Camaroptera implements a multi-stage pipeline that processes an image after its capture in order to identify interesting images. The goal of the pipeline is to process images locally, which has a low time and energy cost, to avoid transmitting uninteresting images, which has a high time and energy cost.

We describe Camaroptera's processing pipeline in the context of a representative driver application that is designed to identify and transmit images containing people. This person detection pipeline has both general and application-specific stages. It has the following four stages.

Difference Filtering. We implemented a simple image differencing algorithm that compares the captured frame with the previous frame. We deem images different from one another if the number of *different* pixels exceeds a heuristically-defined threshold. We set the threshold empirically to 400 pixels by observing that human figures in our images tend to be around 20x20 pixels in size.

Camaroptera can supports more sophisticated methods, but that would introduce additional computational complexity and latency. We chose this method as it is simple, fast (requires only a few subtractions, one addition and one comparison operation) and needs the storage of a single historical image, which works well for our space constrained device.

Inference. If a new image differs from a previous one, Camaroptera runs an application-specific inference routine on that image to identify whether it is interesting to the application. For our person detection application, we have implemented a Convolutional Neural Network (CNN) the structure of which is derived from the LeNet [35] digit classification network.

As trained, even LeNet, which is a small CNN, does not fit in the 250kB of avaliable memory on our MSP430 MCU. We manually altered the network's structure and performed hyperparameter optimization to make the network fit onto the device's memory. To reduce the size of the input layer, we pass our 160x120 input image through a 4x4 average pooling layer before passing it to LeNet's first convolutional layer, reducing the input to 30x40 pixels. Next, we used the Genesis [21] network minimization tool to perform hyperparameter (i.e., structural) optimization on our trained, modified network. Genesis applies aggresive near-zero-pruning and layer separation techniques such as Singular Value Decomposition. This enables us to reduce a network's weight storage requirements, the size of its intermediate activations, the expected inference latency, and the accuracy and precision of the network. The final network occupies 20kB space for the weights, which is a significant reduction on the 3.6MB occupied before optimization. It also occupies a further 80kB for the intermediate activations and gives 78% accuracy on the test set, with 40% False Positives and 1% False Negatives.

Pre-Transmission Transformation. If an image is deemed interesting by inference, Camaroptera transforms the image to prepare it for transmission. This could include a variety of encoding and encryption, depending on application requirements. In our person detection prototype, we compress each interesting image before

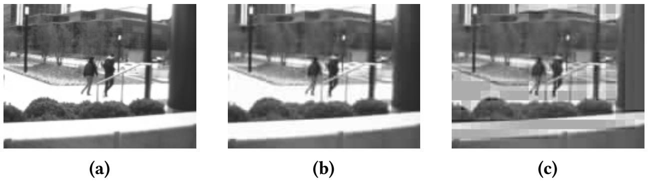

Figure 4: Comparison between the original and compressed version of a frame captured by Camaroptera. (a) Original image. (b) Floating point JPEG. (c) Fixed point JPEG.

transmission by using an optimized version of baseline JPEG compression, derived from the Moodstocks jpec encoder [36].

We modified the implementation to use fixed point arithmetic instead of floating point because our MCU does not natively support floating point operations and software emulation is extremely slow. Shifting to fixed point reduced the latency to compress a 160x120 image from 25 seconds to 7 seconds. Fixed point JPEG degrades image quality, but not excessively, as Figure 4 shows qualitatively.

We additionally optimized the transmission of JPEG headers with the observation that the first 500 bytes of the compressed bit stream remain constant for a fixed image resolution and quality factor. We store the header on the receiver and avoid sending the 500 bytes of header data, which amounts to the transmission of two LoRa packets and seconds or minutes of device operation.

Transmission. In the last stage, Camaroptera packetizes the image and transmits each of an image's packets in sequence using its LoRa radio. In this stage useful information regarding the time between the transmission of the different packets is also collected. This data will be used to infer the luminosity level and exploited by the adaptive reconfiguration routine.

5.2 Adaptive Reconfiguration Routine

Camaroptera's software adapts to changing lighting conditions by varying the operations that it performs in its pipeline. The goal of this reconfiguration is to maximize system availability, by minimizing the end-to-end latency required for capturing, processing and transmitting an image. To achieve that goal, we consider three possible modes of the CamSW processing pipeline. These are `send all`, `diff+send`, and `diff+infer+send`, which include different subsets of the pipeline stages described above. Here `diff` corresponds to Difference Filtering and `infer` corresponds to Inference.

The reconfiguration routine uses a measure of available energy to choose the appropriate operating mode. It does this by recording the supercapacitor charging rate after sending every individual radio packet. This charging rate is measured using a local timer present on the MCU. After all the packets for an image are sent, the individual charging rates are averaged and the average charging rate is used to select the best operating mode for processing the next image. The measured charging rate corresponds to the power available in the environment, which in turn corresponds to the light level.

The reconfiguration routine is pre-loaded with a table that indicates the mode with minimum end-to-end latency for a range of charging rate values. This table is based on a measured precharacterization of Camaroptera's end-to-end latencies in different modes across varying light levels. At runtime, the reconfiguration routine chooses the operating mode that minimizes latency for its measured charging rate value, thereby making the system available for recording and processing maximum interesting events.

6 EVALUATION AND RESULTS

We evaluated Camaroptera's power consumption, and end-to-end latency across a range of operating conditions in a controlled laboratory environment. The evaluation centers on a representative application that detects people outdoors. For this application, an interesting image is one containing at least one person.

6.1 Power Consumption Analysis

Camaroptera's energy harvesting system consumes 197 μW when disconnected from the sensor board, with the supercapacitor fully charged. The sensor board consumes 5.1 mW while capturing a QQVGA image: 3 mW by the microcontroller and 2.1 mW by the camera sensor. Image capture time varies from a minimum of 661 ms to a maximum of 1.62 seconds, because the camera needs more time for exposure calibration in low light conditions. Image difference computation takes 46.14 ms, and the MCU draws 5.77 mW of power. JPEG compressing a QQVGA image takes 7.23 seconds, and the MCU consumes 5.17 mW of power on average. CNN inference takes 11.9 s, with a power consumption of 4.85 mW. Sending a LoRa packet dominates power consumption, with a 363.8 mW power draw for a transmit power of 17 dBm.

The LoRa chipset's Spreading Factor (SF) and Bandwidth (BW) parameters determine its transmit latency and the effective receiver sensitivity. Receiver sensitivity is the minimum signal strength that the receiver can detect, which determines feasible transmit range. A larger-magnitude, negative receiver sensitivity translates to a larger range: the signal can travel further and attenuating more while remaining detectable at the receiver. To find the best combination of these parameters in terms of energy and range, we measured the transmission energy for sending a 255-byte packet at different SF and BW combinations, showing data in Figure 5. Our

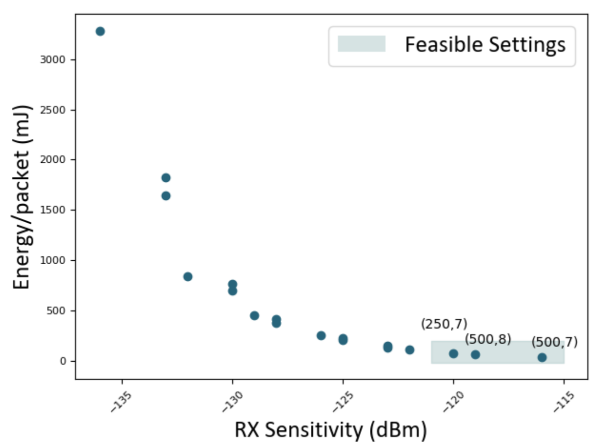

Figure 5: Relationship between LoRa radio sensitivity and energy requirement for sending a packet of 255 bytes. The operating points with feasible energy requirements for our 33mF capacitor are blue highlighted.

Table 1: Recharge time for Camaroptera operations in seconds. '+' indicates ability to run continuously due to sufficient input power.

	Illuminance (klx)										
	1.5	**5**	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**
Cap.	23.2	1.7	+	+	+	+	+	+	+	+	+
Diff.	1.6	+	+	+	+	+	+	+	+	+	+
Inf.	216.2	25.3	+	+	+	+	+	+	+	+	+
Comp.	245	14.97	6.5	+	+	+	+	+	+	+	+
Tx 500/7	307	17.3	6.6	4.24	3.4	2.6	1.9	1.7	1.3	1.1	0.7
Tx 500/8	803	44.6	16.7	9.4	6.3	4.5	4.0	3.1	2.5	2.2	1.8
Tx 250/7	1096	64.7	18.5	10.3	7.5	5.3	4.1	3.5	2.9	2.3	2.0

33 mF supercapacitor can store enough energy for any of three different operating modes – (BW/SF) 500/7, 500/8, 250/7. Each of these modes has a higher receiver sensitivity than the previous, but also a higher total energy per packet.

6.2 Performance Analysis and Results

We characterized the latency to collect energy for different sub-operations that make up Camaroptera's capture, processing, and transmission pipeline. In a controlled lab setup, we varied illuminance from 1.5 klx to 95 klx to simulate different light conditions. Table 1 shows results. We show energy collection latency to transmit a single packet for three different radio configurations. The data show that above 15 klx (outdoors on a bright day), all operations except radio transmission can run continuously without recharging because the input power is sufficient to power the MCU and camera. The radio is power limited and always draws more power than the panels provide. The data also illustrate a key trade-off: collecting energy for inference at low light levels (e.g., <5 klx) has high latency, but that latency is significantly less than the combined energy collection latency to compress and transmit an image. The data thus show that at-sensor inference minimizes end-to-end latency at low light levels.

We evaluated the fraction of interesting images captured and sent and the end-to-end frame latency. We evaluated each of the three operating modes of CamSW for these metrics and present the results in Figures 6 and 7. For evaluating the fraction of interesting images captured, we measured and averaged data from 30 trials per configuration. As we observe in Figure 6, at lower light levels, `diff+infer+send` sends the most interesting images because inference saves the time and energy that other modes spend transmitting packets. As light level increases, the benefit of sophisticated processing diminishes as inference has a flat latency cost, whereas communication recharge latencies reduce. At higher light levels, `diff+send` is superior. The CamSW adaptive controller selects the best operating mode for a light level, and hence captures the highest fraction of interesting images.

Finally, we show Camaroptera's average end-to-end per-image latency in Figure 7, across different operating modes and for different light levels, measured by averaging 40 trials per configuration. The data shows that across light levels, average latency is highest when the system sends all images without filtering, much higher than the modes with image filtering. This demonstrates the effect of Camaroptera's near-sensor processing on the overall per-image latencies, where local filtering makes the system more available to newer interesting events by discarding uninteresting ones.

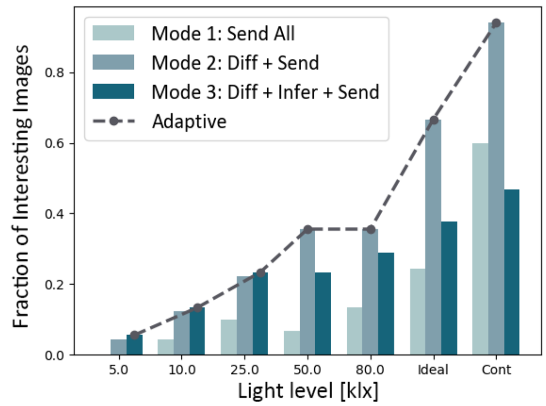

Figure 6: Comparison of the fraction of interesting images captured and sent in every mode at different light levels. The adaptive mode selects the best mode at every light level. 'Ideal' and 'Cont' are continuously powered configurations using a bench power supply. 'Ideal' case is power limited to the datasheet-maximum power of our solar panel array.

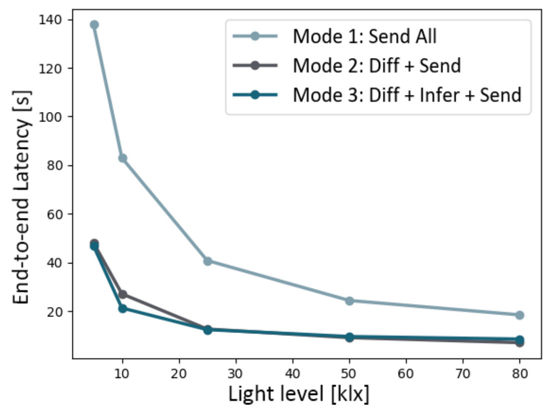

Figure 7: Average time for capturing, processing and transmitting a frame for 3 different working modes.

7 CONCLUSIONS

Developing smart devices that can autonomously adapt and reconfigure based on the environment condition where they are installed, is the key solution to maximize the ratio between acquired useful data and the available energy. In this paper we presented Camaroptera, the first batteryless remote image sensing system with the ability to communicate over extremely long distances using an active LoRa radio. Camaroptera is designed to avoid the high cost of communication using this long-range radio by processing data locally, using at-sensor processing pipelines. Results have highlighted how using machine inference on sensed data substantially reduces energy costs leading to captures and sends 2-5X more images of interest to an application.

REFERENCES

[1] Neal Jackson, Joshua Adkins, and Prabal Dutta. Capacity over capacitance for reliable energy harvesting sensors. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, IPSN '19, pages 193–204, New York, NY, USA, 2019. ACM.

[2] Alexei Colin, Emily Ruppel, and Brandon Lucia. A reconfigurable energy storage architecture for energy-harvesting devices. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '18, pages 767–781, New York, NY, USA, 2018. ACM.

[3] Himax Technologies, Inc. HM01B0 Ultra Low Power camera sensor. https://github.com/cjosephson/backcam/blob/master/hardware/datasheets/HM01B0_DS_preliminary_v06.pdf, 2018.

[4] Semtech. SX127x transceivers Datasheet. https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf, 2019.

[5] A. Dongare, C. Hesling, K. Bhatia, A. Balanuta, R. L. Pereira, B. Iannucci, and A. Rowe. Openchirp: A low-power wide-area networking architecture. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 569–574, March 2017.

[6] Alanson P Sample, Daniel J Yeager, Pauline S Powledge, Alexander V Mamishev, and Joshua R Smith. Design of an rfid-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608–2615, 2008.

[7] Hong Zhang, Jeremy Gummeson, Benjamin Ransford, and Kevin Fu. Moo: A batteryless computational rfid and sensing platform. *Department of Computer Science, University of Massachusetts Amherst., Tech. Rep*, 2011.

[8] Zac Manchester. KickSat. http://zacinaction.github.io/kicksat/, 2015.

[9] Brandon Lucia and Benjamin Ransford. A simpler, safer programming and execution model for intermittent systems. In *ACM SIGPLAN Notices*, volume 50, pages 575–585. ACM, 2015.

[10] Brandon Lucia, Vignesh Balaji, Alexei Colin, Kiwan Maeng, and Emily Ruppel. Intermittent Computing: Challenges and Opportunities. In Benjamin S. Lerner, Rastislav Bodík, and Shriram Krishnamurthi, editors, *2nd Summit on Advances in Programming Languages (SNAPL 2017)*, volume 71 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[11] Alexei Colin and Brandon Lucia. Chain: Tasks and channels for reliable intermittent programs. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, 2016.

[12] Kasım Sinan Yıldırım, Amjad Yousef Majid, Dimitris Patoukas, Koen Schaper, Przemyslaw Pawelczak, and Josiah Hester. Ink: Reactive kernel for tiny batteryless sensors. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 41–53. ACM, 2018.

[13] Josiah Hester, Kevin Storer, and Jacob Sorber. Timely execution on intermi!ently powered ba!eryless sensors. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, SenSys '17.

[14] Emily Ruppel and Brandon Lucia. Transactional concurrency control for intermittent, energy harvesting, computing systems. In *PLDI*, 2019.

[15] Kiwan Maeng, Alexei Colin, and Brandon Lucia. Alpaca: Intermittent execution without checkpoints. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, Vancouver, BC, Canada, October 22–27, 2017. ACM.

[16] Semtech. LoRa™Modulation Basics . https://www.semtech.com/uploads/documents/an1200.22.pdf, 2018.

[17] Semtech . A technical overview of LoRa®and LoRaWAN™. https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdfDS_SX1276-7-8-9_W_APP_V6.pdf, 2019.

[18] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*, 5(1):1 – 7, 2019.

[19] Bradford Campbell, Joshua Adkins, and Prabal Dutta. Cinamin: A perpetual and nearly invisible ble beacon. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 331–332, USA, 2016. Junction Publishing.

[20] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, Luca Benini, and Rajesh Gupta. Pible: Battery-free mote for perpetual indoor ble applications. In *Proceedings of the 5th Conference on Systems for Built Environments*, BuildSys '18, pages 168–171, New York, NY, USA, 2018. ACM.

[21] Graham Gobieski, Brandon Lucia, and Nathan Beckmann. Intelligence beyond the edge: Inference on intermittent embedded systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, pages 199–213, New York, NY, USA, 2019. ACM.

[22] Graham Gobieski, Amolak Nagi, Nathan Serafin, Mehmet Meric Isgenc, Nathan Beckmann, and Brandon Lucia. Manic: An energy-efficient, parallel architecture for ultra-low-power embedded systems. In *Proceedings of the 52nd IEEE/ACM International Symposium on Microarchitecture*, MICRO '19, 2019.

[23] W. J. Dally, J. Balfour, D. Black-Shaffer, J. Chen, R. C. Harting, V. Parikh, J. Park, and D. Sheffield. Efficient embedded computing. *Computer*, 41(7):27–32, July 2008.

[24] J. Balfour, W. Dally, D. Black-Schaffer, V. Parikh, and J. Park. An energy-efficient processor architecture for embedded systems. *IEEE Computer Architecture Letters*, 7(1):29–32, Jan 2008.

[25] Ting Liu, Christopher M. Sadler, Pei Zhang, and Margaret Martonosi. Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet. In *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 256–269, New York, NY, USA, 2004. ACM.

[26] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe. Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 60–71, April 2018.

[27] Ting-Jung Chang, Zhuozhi Yao, Paul J. Jackson, Barry P. Rand, and David Wentzlaff. Architectural tradeoffs for biodegradable computing. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-50 '17, pages 706–717, New York, NY, USA, 2017. ACM.

[28] Texas Instruments. MSP430FR599 Mixed-Signal Microcontrollers datasheet. http://www.ti.com/lit/ds/symlink/msp430fr5994.pdf, 2018.

[29] HopeRF. RFM95W LoRa Module. https://www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf, 2019.

[30] Linear Technology. LTC3105 DC/DC converter. https://www.analog.com/media/en/technical-documentation/data-sheets/3105fb.pdf, 2010.

[31] Texas Instrument. TPS61070 s Boost Converter. http://www.ti.com/lit/ds/symlink/tps61070.pdf, 2015.

[32] Microchip. MIC841 Comparator with Reference and Adjustable Hysteresis. http://ww1.microchip.com/downloads/en/devicedoc/20005758a.pdf, 2017.

[33] AVX. BestCap® Ultra-low ESR High Power Pulse Supercapacitors. http://catalogs.avx.com/BestCap.pdf, 2019.

[34] IXYS. IXOLAR High Efficiency SolarBIT. http://ixapps.ixys.com/DataSheet/KXOB22-01X8F_Nov16.pdf, 2016.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[36] Moodstocks. jpec - a JPEG encoder in C. https://github.com/Moodstocks/jpec, 2016.