

# Energy Efficient Event Driven Video Streaming Surveillance Using sleepyCAM

Tenager Mekonnen\*, Erkki Harjula\*, Arto Heikkinen\*, Timo Koskela<sup>†</sup> and Mika Ylianttila\*

\*Centre for Wireless Communication, University of Oulu, Finland

{tenager.mekonnen, erkki.harjula, mika.ylianttila}@oulu.fi

<sup>†</sup>Center for Ubiquitous Computing, University of Oulu, Finland

{timo.koskela}@oulu.fi

**Abstract**—Wireless Multimedia Sensor Networks (WMSN) are one of the emerging paradigms of the Internet of Things (IoT) that are used to retrieve content including scalar data, video and audio streams and still images from the physical environment. In contrast to scalar sensor nodes, multimedia sensor nodes perform far complex tasks which can be highly power consuming. In this paper, we present the design of energy efficient high resolution camera sensor node, that is capable of capturing full HD video at 30fps, using off-the-shelf hardware for an event driven video streaming surveillance application. In order to achieve long battery life, we use an energy efficient motion detection and power management mechanism, called sleepyCAM, which uses a low-power scalar sensor node to detect motion and wake-up a high resolution camera node when needed. We used Libellium Wasp mote platform and raspberry pi (RPI) to implement the functionality of the low-power sensor node and the HD camera node, respectively. We validate our work using a baseline setup on a standby RPi that uses scalar sensor for motion detection. The results demonstrate that with the used hardware platform, the power consumption can be reduced by more than 99 percent.

## I. INTRODUCTION

Wireless Multimedia Sensor Networks (WMSN) are among the emerging paradigms of the Internet of Things that are used to retrieve content including scalar data, video and audio streams and still images from the physical environment. Recent studies and industrial forecasts predict that future global IP traffic will be largely dominated by multimedia data [1], [2]. According to Cisco's Visual Networking Index (VNI) report, by 2020, 82 percent of global IP traffic will be dominated by video content [1].

Due to their ability of providing richer information and wider coverage, as compared to scalar sensor networks, WMSNs have vital importance in application areas that have significant scientific, social, and strategic relevance [3]. Examples include wildlife monitoring to determine guidelines for human/predator coexistence, monitoring large open areas in airports [3], monitoring crops and farm equipment in agricultural plots [4], monitoring the condition of affected people during an accident in mining industry [5], and monitoring elderly people in assisted living scenarios [6].

In contrast to scalar sensors (such as temperature and humidity sensors), camera nodes generate higher volumes of data requiring higher processing power, and thus consume more energy [3]. In video-surveillance applications, however, the visual sensors spend more than 99% of their time waiting

for an incident [7]. Thus, the waiting time energy consumption is the dominant factor affecting the total energy consumption of a video surveillance network.

Wildlife experts and researchers use HD cameras to capture moving objects for events of interest since monitoring the area at all the time is impractical [8]. Event driven surveillance cameras that capture/stream only events of interests are efficient in saving the device power, memory and bandwidth cost. There are, however, technical challenges and obstacles that need to be addressed. The three fundamental issues in a WMSN are energy consumption, data latency and data quality [9]. For an off-grid video surveillance deployment, energy consumption will be of paramount concern.

This paper asks the question: *Is it possible to achieve high data quality in a wireless multimedia sensor node and yet maintain a prolonged battery life?* To address this issue, we present the following contributions:

- Application of sleepyCAM, a low-power motion detection and power management of a camera node [10], for an event driven full HD (at 1080p) video streaming surveillance application over WiFi network.
- Implementation of event driven full HD (at 1080p) video streaming surveillance application using a standby RPi to be used as a baseline benchmark.
- A quantitative power consumption analysis and battery life estimation of both our proposed solution (sleepyCAM) and the baseline approach.

Most of the energy efficiency approaches applied in Wireless Sensor Networks (WSN) and IoT in general focus on reducing the power consumption of the nodes by introducing different levels of sleep modes during the idle time of a device [11]. Nonetheless, the currently available inexpensive hardware platform RPi [12], which is capable capturing and streaming HD video does not provide power management to support different levels of sleep modes (deep sleep, sleep, awake). If RPi is run on battery, the lifetime is closer to hours than months when it is idle, running only basic operating tasks. Using our sleepyCAM architecture, however, idle time battery life of RPi based camera sensor node can be extended by more than 108 days using a 6000mAh off-the-shelf Lithium-ion battery.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III introduces our

SleepyCAM hardware prototype and network architecture of both the baseline and our proposed solution. Section IV provides the power consumption analysis and evaluation results. Section V concludes the paper by discussing the results and highlighting future works.

## II. RELATED WORK

Energy efficiency in WMSN can be achieved by improving the design of the mote hardware architecture as in [13], [14], or by shutting down the device during the idle times and wake it up when needed such as using "wake-on-wireless" technique [15]. It is also possible to use low-power FM radios as a control signaling to turn-off the nodes' higher-power, higher-capacity radio module, such as WiFi, when it is not sending or receiving data [16]; or a combination of one or more of these techniques.

Wake-on-wireless technique reduces the idle power consumption of mobile devices by allowing the device to shut down when it is not being used. Shih et al. [15], provide an example of this method by adding low-power radio to a IEEE802.11b-enabled PDA. Using out-of-band control signaling from the low-power radio, the system maintains connectivity and wakes up the PDA when needed. Unlike PDAs, however, RPi does not have the mechanism for listening wake-on-wireless packets once it is shutdown, and hence, makes it difficult to apply the technique on it.

Similarly, a more recent studies [16], [17], suggest the use of FM radio in WiFi-enabled multimedia sensor networks as an "always-on" point-to-multipoint control channel used to turn off the WiFi radios in camera nodes. This method can minimize the wasted energy on receiving packets that are destined to other nodes. In addition, by scheduling the transmission windows of the nodes, this technique can minimize the number of collisions in a congested network and hence the energy associated with the re-transmission of packets. The limitation of this method is that it only manages WiFi radios (which contributes only a fraction of a device's total power consumption in the case of RPi and similar devices) and does not control the power state of the main hardware. In addition, this method requires an "always-on" FM radio which consumes energy, and also becomes problematic when scaling up the network in mass deployments of WMSN.

In Turducken [14], a similar type of hierarchical multi-tiered hardware architecture which we propose is used in mobile devices to enhance their availability in a distributed system. By integrating different power-level mobile devices such as PDAs, laptops and sensors into a single multi-tiered device, Turducken can function at power-levels of any of these tiers. The fundamentals of this approach can be applied to WMSNs. However, PDAs and laptops are far too expensive. In addition, unlike RPi, by design these devices do not provide interfaces for integrating with sensors and actuators.

## III. PROTOTYPE ARCHITECTURE

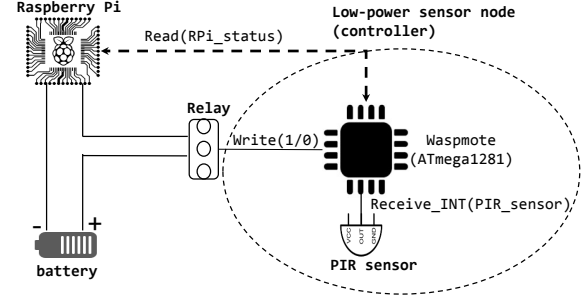


Fig. 1: SleepyCAM: Power management of camera node. [10]

### A. Prototype architecture of sleepyCAM

We have developed sleepyCAM power management mechanism (shown in figure 1) for RPi and similar hardware for realizing a low-power and high resolution wireless multimedia sensor. The hardware architecture of sleepyCAM is composed of a low-power sensor node (controller) and RPi with a camera module. The controller uses Pyroelectric Infrared (PIR) sensor to detect motion and a relay switch to power up RPi when motion is detected.

Our goal is to achieve a low-power operation mode during the waiting time, which constitutes more than 99 percent of a surveillance application lifetime. Previous studies have proposed different techniques to achieve low-power operation of portable devices during idle time by introducing sleep modes and waking-up techniques, such as using wake-on-wireless [15] and multi-tier network architecture [18], [7], [19]. However, such approaches are difficult to implement if the hardware sensor platform used as camera node does not support sleep modes. RPi is one of such microcomputers which is cheap, has great computational capability for capturing and processing high-quality multimedia data, and provides networking over Ethernet, Bluetooth Low Energy (BLE) and WiFi. Yet, RPi does not support Advanced Configuration and Power Interface (ACPI) power management and has no sleep modes. Supplementing RPi's motherboard with full ACPI functionality would naturally raise the price.

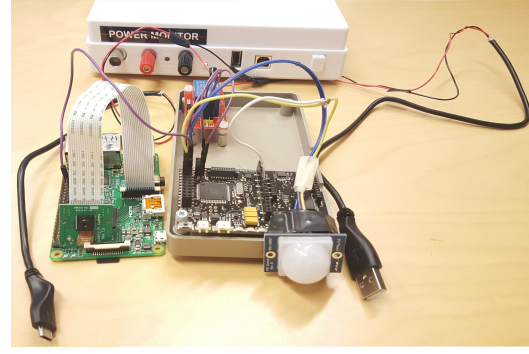
RPi can easily be configured to use the lowest possible clock setting of the CPU (under-clocking). However, this does not reduce the power consumption significantly. Thus, We designed a power management mechanism for RPi using a low power sensor node (the controller). The functional requirements for the controller are: (1) to detect motion using PIR sensor in the area under surveillance and (2) to activate the relay switch and power up RPi when motion is detected.

We used Libelium Wasp mote sensor platform [20] to implement the functionality of the controller node and RPi 3 model B with Rev 1.3 camera module as a camera sensor node. Parallax Rev B PIR sensor is attached to one of the digital pins of the controller for motion detection and a 10A 30VDC rating Songle relay switch is used to control the power of RPi.

Upon waking-up, the RPi starts streaming video to the receiving node over WiFi. It is up to RPi to decide when



(a) baseline setup



(b) sleepyCAM setup

Fig. 2: Hardware prototype architecture

to stop streaming video. The RPi can continue the streaming as long as there is activity in the area. This can be decided by performing motion detection using another PIR sensor on RPi. RPi will eventually shutdown after completing the streaming.

The last but not least step for the controller in the power management process is to detect a graceful shutdown completion of RPi so that it can deactivate the relay without corrupting the micro-sd card in RPi. For that, RPi can be configured to signal the status of the shutdown process over one of its general purpose input/output pins. The controller can read this signal using one of its digital input/output pins and deactivate the relay switch.

The software of the controller to detect motion and activate the relay is implemented in C programming language using Waspnote Pro v040 IDE. The software of the camera node for streaming video is implemented using Python programming language.

The baseline setup used as a benchmark to validate our work is a standby RPi with a camera module and a PIR sensor. The surveillance application for the baseline setup is implemented in Python. When there is an interrupt signal from the PIR sensor, the application calls a function that starts streaming video to a laptop. The hardware prototypes for both setups is shown in figure 2.

#### B. Network Architecture of an event driven WMSN

An overview of the network architecture for both the baseline and our optimized WMSN is shown in figure 3.

- *Scenario A*: presents the baseline setup. In this setup, we have a standby RPi with a camera module and a PIR sensor. When motion is detected, the RPi starts streaming video to a remote server (laptop in our case) over WiFi. In this setup, the RPi is active throughout the surveillance application time.
- *Scenario B*: presents our optimized energy efficient camera (sleepyCAM). In this setup, we have the controller node with a PIR sensor for motion detection, and a relay for managing the power of RPi. When the PIR sensor is active, the controller sets the relay switch to latch and power RPi. RPi will boot and start streaming video to the laptop. After completing the stream, RPi will shutdown and give signal to the controller to unlatch the relay switch. In this setup, the controller node is in standby all the time while the RPi is turned on only when motion is detected.

In both cases, video streaming only starts when an intruder is detected in the area under surveillance. Netcat utility is installed for writing and reading the video stream over TCP protocol on both the sending and receiving ends, RPi and laptop respectively. Similarly, in both setups, the camera nodes stream H.264 video of 10 seconds with resolution of 1920x1080 and bitrate of 17Mbs. The main difference in the two setups is that, in our proposed solution, instead of exploiting the high-computational power camera sensor node (RPi) for motion detection, we used the low-power controller

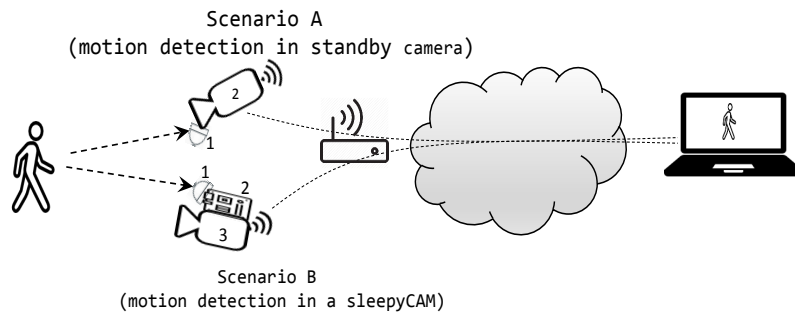


Fig. 3: Network architecture of WMSN for an event driven streaming video surveillance

node to actively monitor the area.

#### IV. POWER CONSUMPTION ANALYSIS

##### A. Evaluation setup

We used Monsoon power monitor tool [21] to measure the power consumption of both RPi and Waspote under all operation modes. Power can be fed to the device under measurement from Monsoon as shown figure 4. The measurement data can be exported as csv file from Monsoon to a workstation. We used moving average filter to smooth the data and generate the graphs using MATLAB.

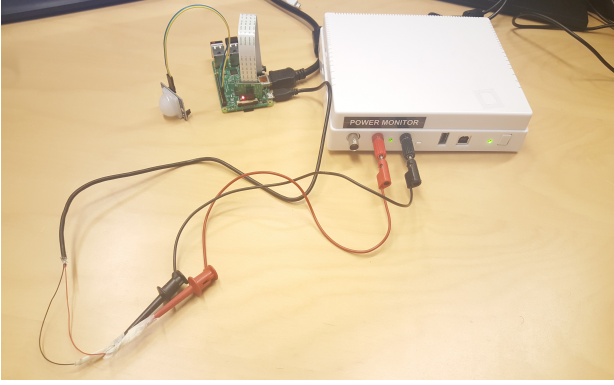


Fig. 4: measurement setup

##### B. Evaluation results

In this section, we present the power consumption transients in different operation states of both the baseline and proposed architecture using the measurement outputs from Monsoon power monitor. During the waiting time (motion detection), the fluctuations of the power transients remain almost the same with time for both setups (since there are no data transmission/reception or some tasks running in the background). However, there could be slight changes in the video streaming phase due to the traffic situation in the network.

Figure 5 shows the power consumption transients of the baseline setup (scenario A) running an event driven video streaming surveillance program. The graph shows the power consumption of a standby RPi before and after the surveillance program is launched. The part of the graph before the surveillance program started characterizes the power consumption of RPi when it is connected to a WiFi network and only basic operating system tasks are running. When the program is launched, there is a short spike and the device will enter to the waiting time mode, which is the motion detection phase. The PIR sensor basically consumes  $\sim 0.35mW$  when it is inactive and  $\sim 8.35mW$  when it is active (refer figure 6). Therefore, the influence of the PIR sensor on the overall energy consumption is marginal since RPi consumes far more power (see figure 5). When motion is detected, the camera module is activated and RPi will stream 10 seconds of video to a remote server (the laptop). During streaming, the average power consumption of the node raises to  $2430mW$  from  $1309mW$  during the waiting time (refer table I and table II).

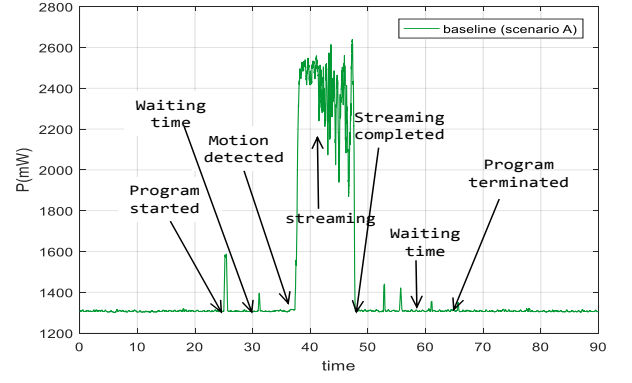


Fig. 5: Power consumption of the baseline setup (scenario A).

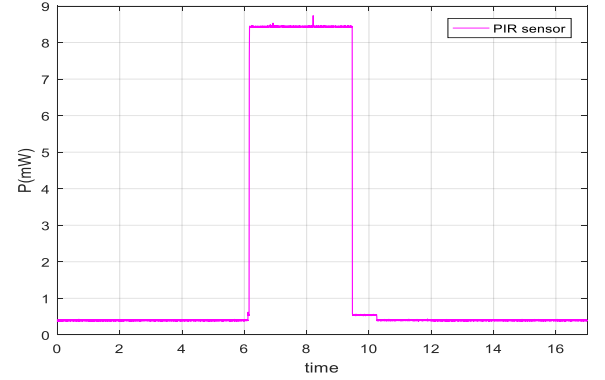


Fig. 6: Power consumption of a PIR sensor

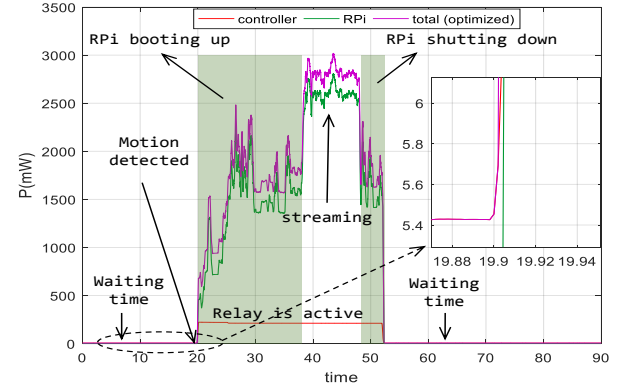


Fig. 7: Power consumption in sleepyCAM (scenario B).

Similarly, figure 7 depicts the power consumption transient of the sleepyCAM prototype running an event driven video streaming surveillance program. The graph shows the power consumption of the controller node (red line), RPi (green) and the total (purple line). As pointed out before, the motion detection part in sleepyCAM is taken care by the controller node. This node enters to the waiting mode soon after it is powered-up and initialization has been completed. The power consumption of this node during the waiting time is  $5.43mW$ . The RPi is however completely powered-off and hence consumes  $0mW$ . Thus, the total power consumption of

sleepyCAM during waiting time is  $5.43mW$  (refer table I).

When motion is detected, the controller power jumps from  $5.43mW$  to  $212mW$  and stays in this power state as long as the RPi is running. This power consumption of the controller is mainly caused by the relay. Note that, we have configured RPi to start video-streaming during the boot-up process, even before any user has logged in. As presented in table II, the RPi power during the streaming in this scenario is  $2470mW$ . Hence, the total power of sleepyCAM during streaming is  $2682mW$ .

After the streaming is completed, the RPi will shutdown itself and the controller will then deactivate the relay switch.

Figure 8a presents the power transient comparison between the baseline setup and our optimized solution using sleepyCAM. The plots are time-synchronized to better visualize the relative power gains and losses in the two scenarios. The beeswarm diagram on the right, figure 8b, presents the power consumption of the two scenarios in the waiting time.

	baseline	optimized
	Power (mW)	Power (mW)
RPi	1309	0
Controller	NA	5.43
<b>Total</b>	<b>1309</b>	<b>5.43</b>

TABLE I: Waiting time power consumption

	baseline	optimized
	Power (mW)	Power (mW)
RPi	2430	2470
Controller	NA	212
<b>Total</b>	<b>2430</b>	<b>2682</b>

TABLE II: streaming power consumption

We conclude the power consumption analysis by providing battery life estimation of the waiting time in both scenarios using a  $6000mAh$  off-the-shelf Lithium-ion battery. Battery life can be estimated from the device's current consumption using online tools like provided in [22]. The average current

consumption, nominal voltage, and hours of battery life are presented in table III. From the values in the table, sleepyCAM prototype improves waiting time battery life by more than 17300%.

	Voltage (V)	Current (mA)	Battery-life(hrs)
baseline	5.0	276	15.2
optimized	3.3	1.6	2625

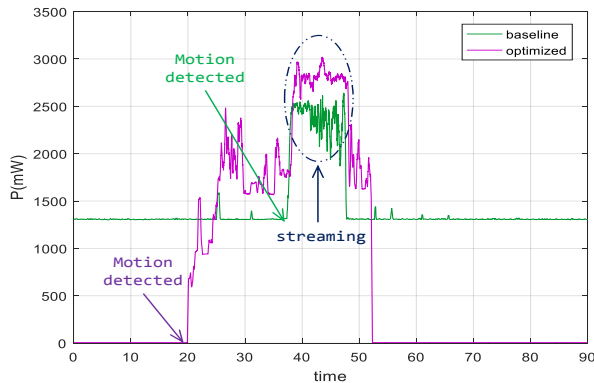
TABLE III: Battery-life of waiting time

## V. DISCUSSION AND FUTURE WORK

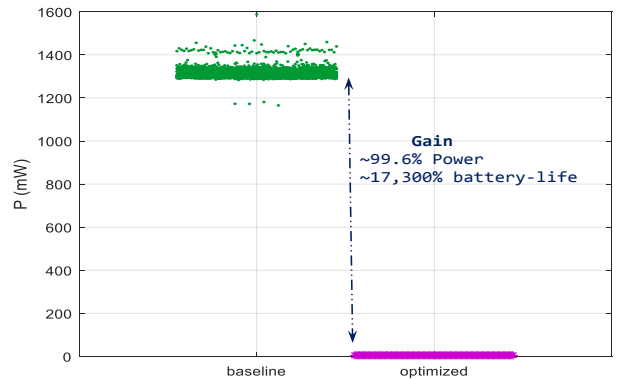
In this paper, we presented energy efficient event driven video streaming surveillance using our proposed sleepyCAM motion detection and power management mechanism. We benchmarked the performance of our proposal using a baseline setup. Both the baseline and our proposed sleepyCAM setup consist of RPi single-board computer with a camera module, to stream video when motion is detected. The fundamental difference of the two setups is the way how each detect motion. In the baseline setup, motion is detected using PIR sensor that is attached to one of the RPi's GPIO pin. In our optimized solution, however, we exploit a low-power controller node to conduct the motion detection and manage power of the camera node at the same time. Our results indicate that, more than 99 percent power consumption can be saved in the waiting time of a video streaming surveillance application using sleepyCAM hardware prototype. Similarly,  $\sim 2600$  hours (108 days) longer battery-life can be achieved in the waiting time using a  $6000mAh$  battery-house.

Finally, while our primary goal was on reducing power consumption in waiting time of event driven video streaming surveillance, we conclude the paper by providing some remarks on how each scenarios could further be improved both in the waiting and video streaming time of the surveillance application.

In scenario A, the RPi OS can be optimized to further decrease the idle power consumption by removing unused services and disabling peripheral devices that are not needed



(a) Overall power consumption comparison



(b) Waiting time power consumption comparison

Fig. 8: Power consumption comparison between baseline setup and our optimized solution using sleepyCAM

for the application. For instance, by disabling the HDMI port, it is possible to save more than 100mW of power.

In scenario B, it is possible to achieve fast boot-up using buildroot tools to generate simple, efficient and easy-to-use embedded Linux systems through cross-compilation [23], [24]. In this scenario, we also noticed that the controller power raises considerably when it powers up RPi. This is mainly due to the solenoid inside the mechanical relay switch which draws lots of current when it is set to latch during the run time of the RPi. The mechanical relay switch can be replaced by a solid state relay or even by an optocoupler (photo-isolator) with the exact load-current rating to reduce the power consumption of the controller significantly during the run-time of RPi. These are beyond the scope of this paper and we did not address them.

In our future work, we plan to extend our sleepyCAM WMSN into multi-tier WMSN. By engaging simple low power scalar sensor nodes as tier-1 device, it is possible to provide an advance alert message to sleepyCAM (tier-2). Doing so can provide enough time to sleepyCAM to wake up and stream video on time. We plan to use BLE link between tier-1 and tier-2 devices. Thus, our future work will include power consumption analysis for BLE messaging between tier-1 and tier-2 devices; and for streaming a video to a server over WiFi network.

#### ACKNOWLEDGMENT

This work is supported by University of Oulu Graduate School (UniOGS), TEKES and the European Celtic-Plus project CONVINCe and was partially funded by Finland, France, Sweden and Turkey.

#### REFERENCES

- [1] Cisco, "White paper: Cisco visual networking index: Forecast and methodology, 2015-2020," Tech. Rep. [Online]. Available: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>
- [2] "Internet of multimedia things: Vision and challenges," *Ad Hoc Networks*, vol. 33, pp. 87 – 111, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870515000876>
- [3] C. B. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "Characterizing energy consumption in a visual sensor network testbed," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, 2006, pp. 8 pp.–339.
- [4] A.-J. Garcia-Sanchez, F. Garcia-Sanchez, and J. Garcia-Haro, "Wireless sensor network deployment for integrating video-surveillance and data-monitoring in precision agriculture over distributed crops," *Computers and Electronics in Agriculture*, vol. 75, no. 2, pp. 288 – 303, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169910002553>
- [5] S. Misra, G. Mali, and A. Mondal, "Distributed topology management for wireless multimedia sensor networks: exploiting connectivity and cooperation," *International Journal of Communication Systems*, vol. 28, no. 7, pp. 1367–1386, 2015. [Online]. Available: <http://dx.doi.org/10.1002/dac.2770>
- [6] M. Hossain and D. Ahmed, "Virtual caregiver: An ambient-aware elderly monitoring system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1024–1031, 2012. [Online]. Available: <http://dx.doi.org/10.1109/TITB.2012.2203313>
- [7] P. Porambage, A. Heikkinen, E. Harjula, A. Gurtov, and M. Ylianttila, "Quantitative power consumption analysis of a multi-tier wireless multimedia sensor network," in *European Wireless 2016; 22th European Wireless Conference*, May 2016, pp. 1–6.
- [8] T. Cevik, A. Gunagwera, and N. Cevik, "A survey of multimedia streaming in wireless sensor networks: progress, issues and design challenges," *arXiv preprint arXiv:1512.03565*, 2015.
- [9] C. F. Chiasserini and E. Magli, "Energy consumption and image quality in wireless video-surveillance networks," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 5, Sept 2002, pp. 2357–2361 vol.5.
- [10] T. Mekonnen, E. Harjula, and M. Ylianttila, "Energy efficient motion detection in a high-resolution wireless surveillance camera node," in *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '16. New York, NY, USA: ACM, 2016, pp. 373–375. [Online]. Available: <http://doi.acm.org/10.1145/3012709.3017604>
- [11] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104 – 122, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128614001418>
- [12] Raspberry Pi Foundation, *Raspberry Pi*. [Online]. Available: <https://www.raspberrypi.org/>
- [13] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 360–369.
- [14] J. Sorber, N. Banerjee, M. D. Corner, and S. Rollins, "Turducken: Hierarchical power management for mobile devices," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 261–274. [Online]. Available: <http://doi.acm.org/10.1145/1067170.1067198>
- [15] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '02. New York, NY, USA: ACM, 2002, pp. 160–171. [Online]. Available: <http://doi.acm.org/10.1145/570645.570666>
- [16] F. Sousa, R. Campos, and M. Ricardo, "Energy-efficient wireless multimedia sensor networks using fm as a control channel," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, June 2014, pp. 1–7.
- [17] J. Dias, F. Sousa, F. Ribeiro, R. Campos, and M. Ricardo, "Green wireless video sensor networks using fm radio system as control channel," in *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jan 2016, pp. 1–8.
- [18] P. Kulkarni, D. Ganesan, and P. Shenoy, "The case for multi-tier camera sensor networks," in *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAY '05. New York, NY, USA: ACM, 2005, pp. 141–146. [Online]. Available: <http://doi.acm.org/10.1145/1065983.1066016>
- [19] R. Bhatt and R. Datta, "A two-tier strategy for priority based critical event surveillance with wireless multimedia sensors," *Wireless Networks*, vol. 22, no. 1, pp. 267–284, 2016. [Online]. Available: <http://dx.doi.org/10.1007/s11276-015-0971-7>
- [20] Waspote Sensor Boards, <http://www.libelium.com/products/waspote/>.
- [21] Monsoon Solutions, Inc., *Monsoon Power Monitor*. [Online]. Available: <https://www.msoon.com/LabEquipment/PowerMonitor/>
- [22] Digi-Key Electronics, *Battery Life Calculator*. [Online]. Available: <http://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life>
- [23] BuildRoot, *Making Embedded Linux Easy*. [Online]. Available: <https://buildroot.org/>
- [24] Busybox. [Online]. Available: <https://git.busybox.net/buildroot/tree/board/raspberrypi/readme.txt>