WISPCam: An RF-Powered Smart Camera for Machine Vision Applications

Saman Naderiparizi
Electrical Engineering
Department
University of Washington
Seattle, Washington
samannp@uw.edu

Zerina Kapetanovic
Electrical Engineering
Department
University of Washington
Seattle, Washington
zerinak@uw.edu

Joshua R. Smith
Computer Science and
Engineering Department
Electrical Engineering
Department
University of Washington
Seattle, Washington
jrs@cs.washington.edu

ABSTRACT

Sustained exponential improvements in the energy efficiency of microelectronics has recently enabled us to build battery-free camera systems that are powered entirely by propagating radio waves. This paper describes primitive machine vision applications built using this highly constrained, battery-free camera system. After describing the *WISPCam* system and its constraints, we show how to leverage on-board computation capabilities to compress data transmit as much as possible. Then we demonstrate application of *WISPCam* for face detection/recognition. To that end, we use *WISPCam* to capture (relatively) high resolution images of faces, without ever capturing a full frame at high resolution. This example application illustrates the issues that arise in partitioning a demanding vision application across mobile hardware that is highly constrained in power, storage, computation, and communication.

CCS Concepts

• **Computer systems organization** → *Sensor networks; Embedded hardware;*

Keywords

Battery-Free Camera, Smart Camera, RF-Powered, Machine Vision, Compression, On-Board Computation

1. INTRODUCTION

Computation efficiency has been exponentially increasing over the last 70 years. Comparing to Eniac, today's microelectronics are about one trillion (10^{12}) times more energy efficient [3, 11]. This scaling in energy efficiency has made it possible to power general purpose microcontrollers and low-power sensors in early 2000s using radio frequency (RF) signals as the only power source [9, 12, 10]. Powering cameras solely off of RF signals became possible most

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ENSys 2016 November 16, 2016, Stanford, CA, USA

© 2016 ACM. ISBN 978-1-4503-4532-3/16/11... \$15.00

DOI: <http://dx.doi.org/10.1145/2996884.2996888>

Figure 1: Latest prototype version of *WISPCam* with an off-the-shelf cellphone VGA camera

Figure 2: High-level block diagram of *WISPCam* deployment with its necessary components that will help eliminate its limitations

recently as an example of a more demanding workload [6, 8, 13]. These devices have been enabled to monitor their own energy budget with a very low-power on-board energy monitoring scheme [7], which will enable battery-free devices to make smart decisions based on their available energy. In this article we review our work on the *WISPCam*, an RF-powered camera that communicates via backscatter, and describe the challenges of mapping a machine vision application onto this highly constrained system.

2. RF POWERED MOBILE CAMERA

The *WISPCam*, shown in Figure 1, is a passive wireless RFID tag enhanced with a VGA camera that leverages RF signals for power and communication. The core of the *WISPCam* is a low-power microcontroller that handles the RFID communication protocol to exchange information with an RFID reader as well as interfacing with the camera. The *WISPCam* scavenges energy from radio waves using an energy harvester and stores this energy on a charge reservoir. While this system enables battery-free and wireless imaging it comes with a set of constraints:

(a) Current frame (b) Frame delta (c) Normalized

Figure 3: Figure 5a shows the CDF of total number of pixels that changed more than the corresponding threshold between consecutive frames. The dashed lines show the corresponding results if our normalization technique is not used. Figure 3a shows the current frame. Figures 3b, and 3c show the delta between the current and previous frame with and without applying our normalization method.

1. **Communication speed limitation.** The effective bit-rate using off-the-shelf RFID readers is less than 100kbps.
2. **Limited computational capabilities.** The *WISPCam* is an energy scavenging device, thus for it to operate properly cutting its computation corners in exchange for power requirement reduction is a vital factor.
3. **Limited data/code storage space.** Its total on-chip low-power non-volatile memory (FRAM) is limited to about 64 kilobytes of data and code memory.

As a result, the *WISPCam* requires some intelligence to partition its operation efficiently while performing tasks beyond sensor data reading. Essentially, the *WISPCam* must decide how to work around its storage, communication, and computation limitations by spreading its workload to the on-board microcontroller and a host PC. To that end, we divide the *WISPCam* operations into two subcategories of **Computationally Light Tasks** and **Computationally Demanding Tasks**. Given the computational capabilities of the *WISPCam*, it executes light tasks on-board before transmission to reduce the use of the limited communication bandwidth. In the case of demanding tasks, the *WISPCam* will off-load tasks to the cloud for further processing [5] as shown in Figure 2.

3. COMPUTATIONALLY LIGHT TASKS

The *WISPCam* is an appealing solution for applications that require dense deployment because it is cost efficient and maintenance-free. For instance, it can serve as a camera for surveillance systems. However, due to the rich nature of images, in regards to data size, a dense distribution of surveillance *WISPCam*'s would face communication bandwidth limitations.

Consider a typical surveillance camera and what it captures in its frame of view. It can be assumed that majority of the pixels that the camera sees are static background pixels, such as the sky, a building, or a street. We evaluate the case in which the surveillance camera only transmits pixels that change over consecutive frames. Previous work has shown that this process, known as Selective Change Driven Imaging, has the potential to significantly reduce data transmission [2]. Furthermore, we investigated how feasible it is in terms of computation to implement this process on the *WISPCam*. To that end, the *WISPCam* was examined in

two scenarios. In the first scenario, it was placed in a research lab to capture 210×160 pixels resolution images for 24 hours. In the second scenario, the *WISPCam* was placed facing outside collecting 210×160 pixel resolution images during a work day from 9AM to 5PM. In the following section we go through data analysis for each scenario:

3.1 Indoor Surveillance Camera

Data was collected for 24 hours, where the data gathered from 11PM to 8AM was used to analyze and characterize the noise from the camera and environment. During this period there was no motion in the lab, meaning that any changes in pixel intensity should be due to the camera or the environment injected noise. In every experiment, the pixels that change over a percentage threshold of 255 (the maximum intensity level) were considered to be significantly changed pixels and were transmitted.

Here we calculated the variation in pixel intensity for the 230 consecutive frames from 11PM to 8AM. Figure 3b illustrates the changed pixels between Figures 3a and its previous frame. Note that there are some horizontal stripes of changed pixels appearing in Figure 3b. Since the camera is triggered about every 100ms, we believe the horizontal lines are due to the 60Hz blinking of the lab lamps.

We mathematically model the noise to gain an understanding of these characteristics. Let us define $P_{i,j}$, $B_{i,j}$, and $n_{i,j}$ as the camera pixel reading, actual background intensity, and noise introduced on pixel i, j , respectively. The noise is modeled as,

$$P_{i,j} = B_{i,j} + n_{i,j} \quad \forall (i, j) \in \text{Pixel Indices} \quad (1)$$

Let us also define that $n_{i,j}$ has normal distribution with zero mean and σ^2 variance. We then assume the *WISPCam* captures a new image in a motion-free environment defined as,

$$P_{i,j}^* = B_{i,j}^* + n_{i,j}^* \quad (2)$$

The horizontal stripes in Figure 3b tell us that there might be a scaling factor between $B_{i,j}^*$ and $B_{i,j}$ (as an instance of 60Hz blinking from the lab lighting), so that $B_{i,j}^* = K_i \times B_{i,j}$. Given that there are M pixels per row and assuming a constant scale factor K_i for each row, we can find K_i as shown below by averaging pixels in a single row.

$$\begin{aligned} \sum_{j=1}^M P_{i,j} &= \sum_{j=1}^M B_{i,j} + \sum_{j=1}^M N(0, \sigma^2) \\ &= \sum_{j=1}^M B_{i,j} + N(0, M\sigma^2) \\ &\rightarrow \overline{P_i} = \overline{B_i} + N(0, \sigma^2/M) \end{aligned} \quad (3)$$

Since M is a large number, in this case 210, we approximate $N(0, \sigma^2/M)$ with zero. Thus allowing us to easily find the K_i as shown below,

$$K_i \approx \frac{\overline{P_i^*}}{\overline{P_i}} \quad (4)$$

Using the above K_i before subtracting the current frame from the previous frame will help normalize the current frame to the previous one, thus reducing the appearance of horizontal stripes. The solid lines in Figure 5a show the CDF of the total number of changed pixels after applying the normalization method.

(a) Previous Frame (b) Current Frame

(c) Delta 2.5% (d) Delta 5% (e) Delta 7.5%

(f) Recovered 2.5% (g) Recovered 5% (h) Recovered 7.5%

Figure 4: An example of the difference and recovered image for two consecutive images. It seems that 5% threshold is the right threshold.

(a) Noise CDF (b) Indoor CDF (c) Outdoor CDF

Figure 5: CDF of total number of significantly changed pixels over consecutive frames for noise analysis, indoor, and outdoor scenarios is shown in Figures 5a, 5b, and 5c respectively. There are three different thresholds for each case.

We evaluate this method over a dataset of 429 frames gathered from 8AM to 11PM during a work day in the lab. Based on Figure 5a it seems that a 1.125% threshold allows a significant amount of noise to pass through the algorithm. On the other hand a 5% threshold will let 3% of the total pixels through. Analyzing the results, it is presumed a threshold of 5% or above will guarantee prevention from most noise introduced.

Figure 4 shows two consecutive frames, their differences, and the resulting recovered image from the previous frame for the three different thresholds. Empirically picking a threshold of 5% will maintain the important information while rejecting most of the noise as well as achieving about $22\times$ reduction in data transmission.

3.2 Outdoor Surveillance Camera

We performed the same experiment for a *WISPCam* facing outside, collecting 530 images from 9AM to 5PM. The CDF of total number of changed pixels is shown in Figure 5c. It is important to note that in this scenario we did not apply our normalization method explained before because the outdoor scenario is not impacted by the same sources of noise, such as the 60Hz blinking lights. Based on Figure 6, setting the threshold to 5% will result in an acceptable recovered image while reducing the data transmission by a factor of 52.

(a) Previous Frame (b) Current Frame

(c) Delta 2.5% (d) Delta 5% (e) Delta 7.5%

(f) Recovered 2.5% (g) Recovered 5% (h) Recovered 7.5%

Figure 6: An example of the difference and recovered image for two consecutive images.

Figure 7: Handshake protocol for communication between the *WISPCam* and a host PC

4. COMPUTE-DEMANDING TASKS

Most vision algorithms are computationally intensive and therefore currently impossible to implement on an energy and computation limited platform. For instance, the Viola-Jones face detection algorithm takes approximately $426ms$ on a $2.2GHz$ notebook to detect faces for a VGA image [15]. This means it is impossible to implement real-time face detection on the *WISPCam*. Other vision tasks, such as face recognition, are even more computationally intensive. Consequently, if we were to enable these applications on the *WISPCam* a combination of on-board and cloud computation would be required.

However, face recognition will have a higher recall rate on higher resolution images up to a certain image resolution [14]. However, the total amount of data memory available on the *WISPCam* is just enough for an image with a resolution 10 times lower than a VGA image. Furthermore, face detection is possible with lower image resolutions in comparison to face recognition [4]. With this in mind, our idea is to capture and send a subsampled low resolution 160×120 image. Then, a host PC will detect the coordinates of the windows that contain faces and report them back to the *WISPCam*. Finally, the *WISPCam* will capture windowed and high resolution images of the faces and transmit them back to the PC [5].

In order to perform the necessary tasks on a PC, we developed system software using Python. To elaborate, we leveraged SLLURP [1], a pure-python Low Level Reader Protocol (LLRP) library. Integrating SLLURP with the software

Figure 8: *WISPCam* will capture a 160×120 low resolution image for face detection which is shown in 8a. Extracting the face by zooming in will result in a poor quality image 8b. Next, *WISPCam* will capture a windowed high resolution image of the face according to the coordinates received from host-PC 8c

interface, we were able to develop a simple handshake protocol to initiate the capturing of an image and perform face detection, as shown in Figure 7.

For proof of concept, we had people stand in front of the *WISPCam* for it to perform face detection. In this experiment we placed the *WISPCam* five feet away from the RFID antenna and had the *WISPCam* follow the protocol for face detection. The results of this process are shown in Figure 8. A face captured by the *WISPCam* has 16 times more pixel resolution than the initial 160×120 image.

5. FUTURE WORK

The idea of shared parking has been around for quite a while. However, to enable it a very low cost automated system is needed. The application of automated parking services is feasible even with our latency limitations on the proposed extensions to the *WISPCam*. To elaborate, most parking areas have speed bumps and entrance gates that slow down vehicles, thus one second latency would be tolerable. As a first step, we will deploy the current *WISPCam* for automated parking facilities to test its feasibility, but the long term goal is to have high-speed backscatter communication built into the *WISPCam*.

6. SUMMARY

In this article we reviewed the various features implemented on the *WISPCam*, a wireless and battery-free mobile RF powered camera. Numerous applications were demonstrated to address the possible workarounds for the limitations that the *WISPCam* faces. We presented our approach on balancing computational tasks between a host PC and the *WISPCam*. Additionally, the feasibility of some computationally demanding vision tasks were explored and evaluated. The key achievements on the *WISPCam* so far as an end-to-end system will provide new avenues for future applications, which in turn will make the *WISPCam* a more practical solution for real-life scenarios.

7. ACKNOWLEDGEMENTS

This work is funded by Intel Science and Technology Center for Pervasive Computing (ISTC-PC), NSF award number CNS 1305072, and a Google Faculty Research Award.

8. REFERENCES

- [1] sllurp – Python client for LLRP-based RFID readers. <https://github.com/ransford/sllurp>. Accessed December 2014.
- [2] J. Boluda, P. Zuccarello, F. Pardo, and F. Vegara. Selective change driven imaging: A biomimetic visual sensing strategy. *Sensors*, 2011.
- [3] J. Koomey, S. Berard, M. Sanchez, and H. Wong. Implications of historical trends in the electrical efficiency of computing. *Annals of the History of Computing, IEEE*, 33(3):46–54, March 2011.
- [4] T. Marciniak, A. Chmielewska, R. Weychan, M. Parzych, and A. Dabrowski. Influence of low resolution of images on reliability of face detection and recognition. *Multimedia Tools and Applications*, 74(12):4329–4349, 2015.
- [5] S. Naderiparizi, Z. Kapetanovic, and J. Smith. Battery-free connected machine vision with wispcam. *GetMobile: Mobile Comp. and Comm.*, 20(1), 2016.
- [6] S. Naderiparizi, A. Parks, Z. Kapetanovic, B. Ransford, and J. Smith. Wispcam: A battery-free rfid camera. In *RFID, 2015 IEEE International Conference on*, pages 166–173, April 2015.
- [7] S. Naderiparizi, A. Parks, F. Salemi Parizi, and J. Smith. umonitor: In-situ energy monitoring with microwatt power consumption. In *RFID (RFID), 2016 IEEE International Conference on*, May 2016.
- [8] S. Naderiparizi, Y. Zhao, J. Youngquist, A. P. Sample, and J. R. Smith. Self-localizing battery-free cameras. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15*, pages 445–449. ACM, 2015.
- [9] M. Philipose, J. R. Smith, B. Jiang, A. Mamishev, S. Roy, and K. Sundara-Rajan. Battery-free wireless identification and sensing. *Pervasive Computing, IEEE*, 4(1):37–45, 2005.
- [10] A. Sample, D. Yeager, P. Powledge, A. Mamishev, and J. Smith. Design of an rfid-based battery-free programmable sensing platform. *Instrumentation and Measurement, IEEE Transactions on*, 57(11):2608–2615, Nov 2008.
- [11] J. R. Smith. Range scaling of wirelessly powered sensor systems. In *Wirelessly powered sensor networks and computational RFID*, pages 3–12. Springer, 2013.
- [12] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy, and A. Mamishev. A wirelessly-powered platform for sensing and computation. In *UbiComp 2006: Ubiquitous Computing*, pages 495–506. Springer, 2006.
- [13] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith. Powering the next billion devices with wi-fi. *CoRR*, abs/1505.06815, 2015.
- [14] J. Wang, C. Zhang, and H.-Y. Shum. Face image resolution versus face recognition performance based on two global methods. In *Proceedings of Asia Conference on Computer Vision*, volume 47, pages 48–49. Citeseer, 2004.
- [15] Q. Wang, J. Wu, C. Long, and B. Li. P-fad: Real-time face detection scheme on embedded smart cameras. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6, Oct 2012.