

Battery-Free Connected Machine Vision with *WISPCam*

Saman Naderiparizi¹, Zerina Kapetanovic¹, Joshua R. Smith^{2,1}

¹Electrical Engineering Department and ²Computer Science and Engineering Department
University of Washington, Seattle, USA-98195

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 use it 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.

I. INTRODUCTION

The energy efficiency of computing has been improving on a steady exponential trajectory for the last 70 years at least, a trend that pre-dates integrated circuits. Today's microelectronics is about one trillion (10^{12}) times more energy efficient than early computers such as Eniac [1] [2]. Thanks to this energy efficiency scaling, it became possible in the early 2000s to power general purpose microcontrollers and low-power sensors using propagating radio waves as the only power source [3] [4] [5]. Most recently, it has become possible to power more demanding workloads, such as cameras, using only radio waves [6] [7] [8]. This article reviews our work on the *WISPCam*, an RF-powered camera that communicates via backscatter, and describes the challenges of mapping a machine vision application onto this highly constrained system.

II. RF POWERED MOBILE CAMERA

The *WISPCam* is a passive wireless RFID tag enhanced with a VGA camera that leverages RF signals for power and communication. A photo of the *WISPCam* is shown in Figure 1. The *WISPCam* has an on board low-power microcontroller enabling it to communicate with the camera to store image data as well as handling the RFID backscatter communication protocol. Additionally, the *WISPCam* utilizes a power harvester to accumulate energy from RF signals and then store the energy on a charge reservoir.

The major significance of the *WISPCam* is its ability to capture images, perform additional computation, and transmit data wirelessly without the limitations of batteries, excess wiring, or complex installation, yet still remaining a low-power system. These functionalities and features introduce many promising real-world applications for the *WISPCam*. A few of these applications include, but are not limited to, monitoring inaccessible areas, surveillance (such as home security), or

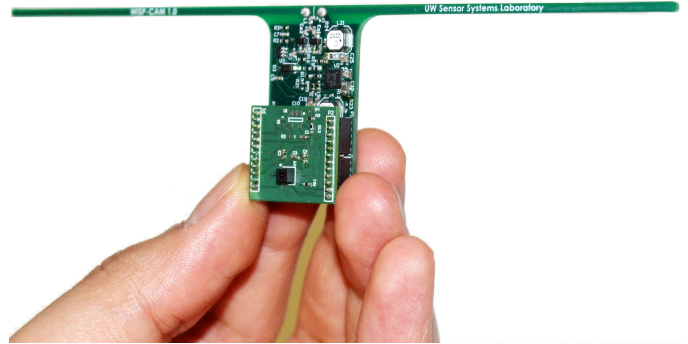


Fig. 1: Latest prototype version of *WISPCam* with an off-the-shelf cellphone VGA camera

even monitoring parking availability. In fact, the *WISPCam* can be an effective solution to the aforementioned applications because it is completely battery-free and wireless, resulting in a very low-maintenance and cost-efficient mobile vision and computation platform. Despite enabling many applications, the *WISPCam* standalone has three primary constraints:

- 1) **Communication speed limitation.** The effective bit-rate using off-the-shelf RFID readers is less than 100kbps.
- 2) **Limited computational capabilities.** *WISPCam* is an energy scavenging device, thus for it to operate properly cutting its computation corners in exchange of 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. Essentially, the *WISPCam* should decide to work around its storage, communication, and computation limitations by spreading its workload to the on-board computer and host PC (cloud). Additionally, if a device like the *WISPCam* lacks the intelligence to capture images based on a target application, it is probable that it will capture and transmit a vast amount of data that contains insignificant information. For instance, in the case of the *WISPCam* as a surveillance camera, the *WISPCam* should only capture an image whenever some form of movement is occurring in its field of view.

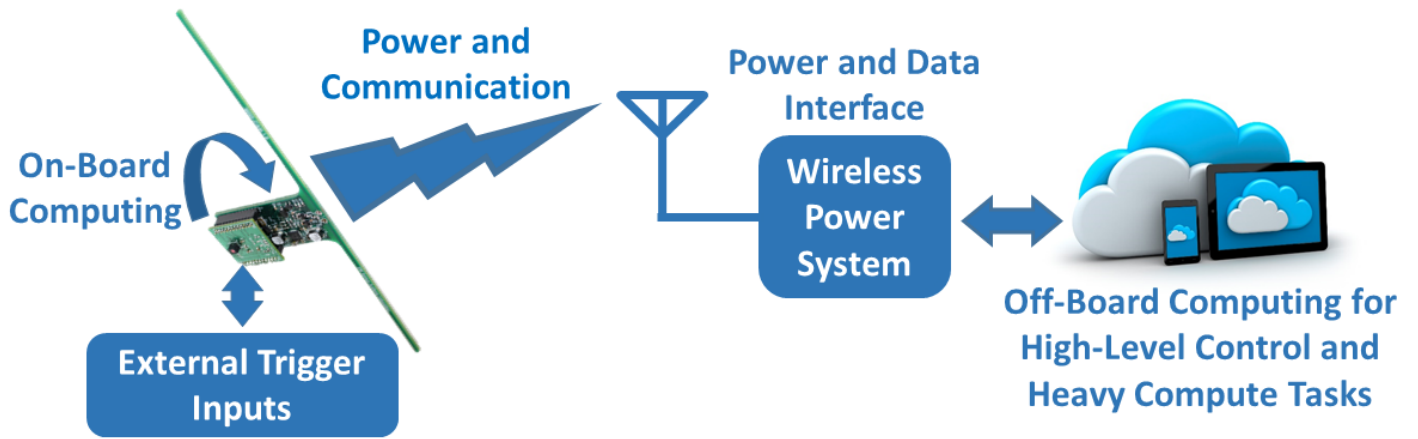


Fig. 2: High-level block diagram of *WISPCam* deployment with its necessary components that will help eliminating its limitations

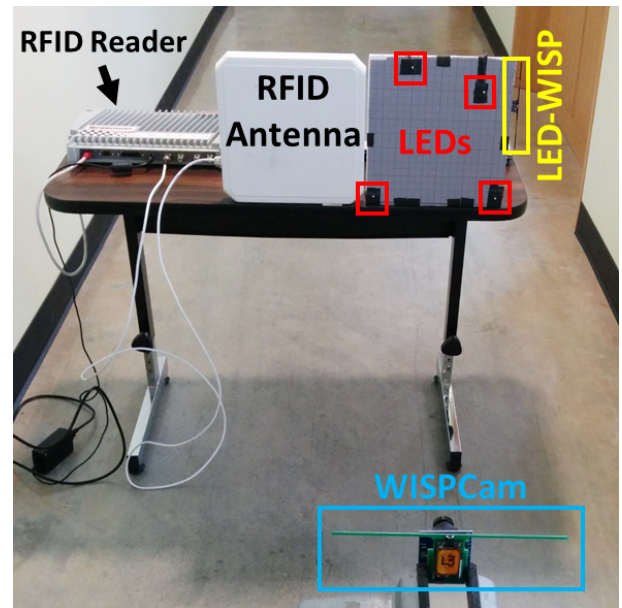
Figure 2 shows the high-level block diagram of *WISPCam* system deployment and how we can decompose and balance tasks on the cloud, allowing the *WISPCam* to overcome its limitations. In essence, the *WISPCam* can compress its backscatter data by either light computations or application based low-power triggering inputs. This will result in removing storage and communication barriers to some extent. Then in a higher level process, heavy computational tasks can be off-loaded to the cloud to overcome computation and storage limitations.

As another example, if a host PC wants to trigger the *WISPCam* and assign a task, based on the energy requirements of the task, the host should either distribute the work into smaller tasks or wait until the *WISPCam* harvests enough energy. To elaborate, image capturing is an atomic operation and triggering this operation when the total available energy is less than the minimum required energy will result to a failure and thus wasting the stored energy. This conveys that the host PC should have some notion of the *WISPCam*'s available energy. This can be achieved by having a low-power timer running at all times, which wakes the CPU on the *WISPCam* every few seconds to sample its charge level and send that over to the host PC. In the remaining of this article, we will demonstrate how we implemented the full-system deployment of the *WISPCam*.

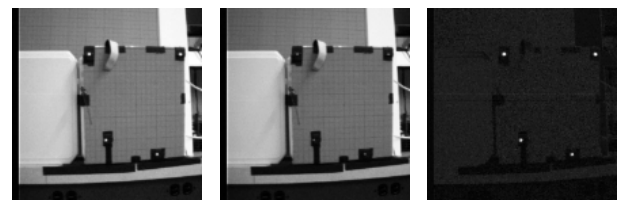
III. ON-BOARD COMPUTATION

Localization is a key aspect that allows us to take more advantage of the tag. Specifically, in the case of a *WISPCam*, knowing the location from which an image was captured can enable different applications that are location dependent. A few examples include, environmental modeling, 3D object reconstruction, and location triggered image capturing that can be potentially enabled with a location-aware *WISPCam*.

To that end, we reuse the on-board camera of the *WISPCam* to optically localize itself by having an optical cue in the field of view. The idea is to detect four pixel coordinates in the image plane that correspond to four known locations in the field of view. To enable this, four LEDs that are powered and controlled by a battery-free RFID tag (LED-WISP) are placed



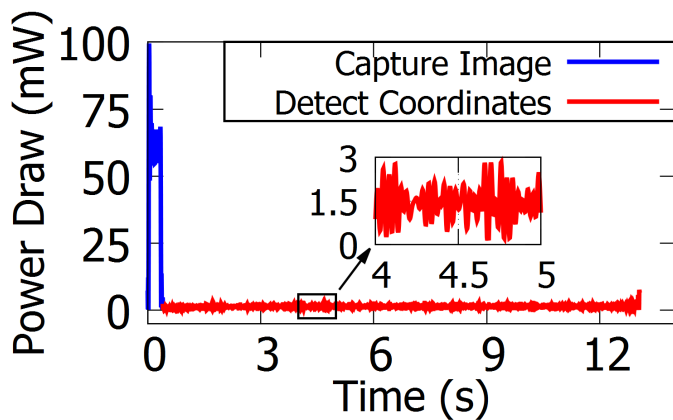
(a) Localization setup



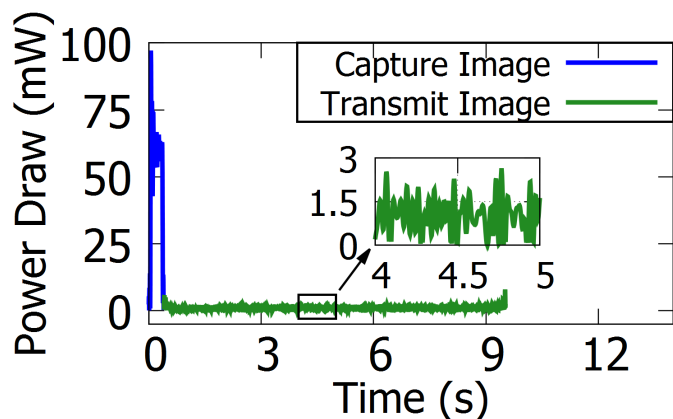
(b) Foreground Image (c) Background Image (d) Subtracted Image

Fig. 3: Localization setup and a foreground, background and subtracted images sample of *WISPCam*. *WISPCam* will detect the four LED coordinates by subtracting background and will transmit only 4 coordinates instead of an entire image.

in four known locations in the field of view of the *WISPCam*. The *WISPCam* will then captures two back-to-back images using its camera. The first image is called foreground, that has the four LEDs on, and the second image is called background, that has the four LEDs off.



(a)



(b)

Fig. 4: 4a shows detecting the LED coordinates leveraging on-board computation and 4b shows transmitting the both raw foreground and background images without any computations.

The task of the LED-WISP is to listen to the communication between the *WISPCam* and an RFID reader and then synchronize itself to the *WISPCam* whenever a known secure key is transmitted between the RFID reader and the *WISPCam*. Detecting the occurrence moment of the secure key transmission will allow LED-WISP to know exactly when the *WISPCam* is going to capture the foreground and background images. Consequently, the *WISPCam* will be ensured to have a pair of proper foreground and background images, where their subtraction will simply give four bright corners that correspond to the four LEDs coordinates. The setup of this implementation is shown in Figure 3a.

The important point here is that the foreground and background images, as well as the detection of LEDs coordinates, is being done on the *WISPCam*. Thus, instead of sending an entire image to the RFID reader, only eight coordinates are being transmitted. This reduces data transmission to the RFID reader per *WISPCam* by a factor of 1680. Therefore, this will make our localization method practical for a very dense network of the *WISPCam*. Instances of *foreground* and *foreground - background* images are shown in Figure 3.

We explored the *WISPCam*'s power consumption while doing the aforementioned on-board computation. Figure 4a shows the power consumption of the *WISPCam* leveraging

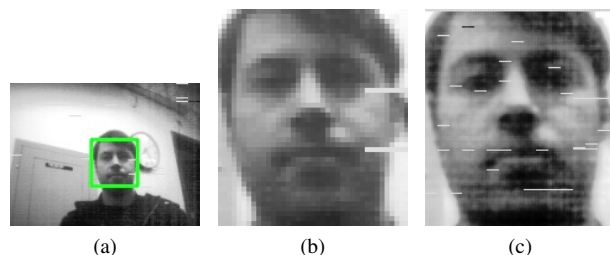


Fig. 5: *WISPCam* will capture a 160×120 low resolution image for face detection which is shown in 5a. Since this image does not have enough resolution extracting the face by zooming in will result in a poor quality image 5b. Next, *WISPCam* will capture a windowed high resolution image of the face according to the coordinates of the face 5c

on-board computation to detect LEDs coordinates, while Figure 4b shows power consumption profile in a scenario where the *WISPCam* transmits its entire image frames to the host PC without doing any computation. The energy required for capturing foreground and background images, computing LED coordinates, and transmitting raw image data is $20.5mJ$, $17.47mJ$, and $8.19mJ$, respectively. Leveraging on-board computation to compute LED coordinates is a more power hungry task than transmitting the raw image, which is due to the fact that backscatter is a passive radio. On the other hand, paying that extra energy to compute LED coordinates on-board will introduce about 1700 times less traffic on the network.

IV. CLOUD COMPUTING

Most vision algorithms are computationally intensive. Thus, it is not possible to implement them on an energy and computation limited platform like the *WISPCam*. For instance, the Viola-Jones face detection algorithm takes approximately $426ms$ on a $2.2GHz$ notebook to detect faces for a VGA image [9]. This means it is impossible to implement face detection in real-time on the *WISPCam*. Some other vision tasks, such as face recognition, are even more computationally intensive. Consequently, we require a combination of on-board and cloud computation to enable face recognition using the *WISPCam*.

On the other hand, face recognition will have a higher recall rate on higher resolution images up to a certain image resolution threshold [10]. However, the total amount of data memory available on the *WISPCam* is just enough for an image that has a resolution 10 times lower than a VGA image. Nevertheless, face detection is possible with lower image resolutions in comparison to face recognition [11]. 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. The result of this process is shown in Figure 5. The face captured by *WISPCam* has four times more pixel density in each dimensions than the initial 160×120 image.



(a) Ground-truth image from the personal camera (b) Image captured by surveillance WISPCam

Fig. 6: (6a) ground-truth image captured with a webcam camera, and (6b) image captured by WISPCam which was triggered by a PIR sensor.

V. MOTION TRIGGERED CAMERA

The WISPCam is an energy scavenging device, which means it is duty cycled based on the available energy and, by default, image capture and transmission will be triggered whenever there is enough energy stored to do so. To use the camera for surveillance purposes, it is not ideal that the WISPCam triggers only based on energy storage. In fact, the WISPCam should rely on motion detected in its field of view as well. Thus, we integrate an ultra-low-power Passive Infrared (PIR) motion sensor with the WISPCam. The PIR sensor is implemented to be continuously active, while burning less than $3\mu\text{W}$ of power. This enables the ability to detect motion and trigger the capture of an image in real-time. Figure 6 shows the ground truth and the image captured by the surveillance WISPCam.

VI. 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 work arounds 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.

VII. ACKNOWLEDGEMENTS

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

REFERENCES

- [1] 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*, vol. 33, pp. 46–54, March 2011.
- [2] J. R. Smith, "Range scaling of wirelessly powered sensor systems," in *Wirelessly powered sensor networks and computational RFID*, pp. 3–12, Springer, 2013.
- [3] M. Philipose, J. R. Smith, B. Jiang, A. Mamishev, S. Roy, and K. Sundara-Rajan, "Battery-free wireless identification and sensing," *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 37–45, 2005.
- [4] 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*, pp. 495–506, Springer, 2006.
- [5] 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*, vol. 57, pp. 2608–2615, Nov 2008.
- [6] S. Naderiparizi, A. Parks, Z. Kapetanovic, B. Ransford, and J. Smith, "Wispcam: A battery-free rfid camera," in *RFID (RFID), 2015 IEEE International Conference on*, pp. 166–173, April 2015.
- [7] 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*, (New York, NY, USA), pp. 445–449, ACM, 2015.
- [8] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith, "Powering the next billion devices with wi-fi," *CoRR*, vol. abs/1505.06815, 2015.
- [9] 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*, pp. 1–6, Oct 2012.
- [10] 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*, vol. 47, pp. 48–49, Citeseer, 2004.
- [11] 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*, vol. 74, no. 12, pp. 4329–4349, 2015.