2nd International Conference on Sensors Engineering and Electronics Instrumental Advances (SEIA' 2016), 22-23 September 2016, Barcelona, Spain

Power consumption considerations of an agricultural camera sensor with image processing capability

Gábor Paller and Gábor Élő
Széchenyi István University, Information Society Research & Education Group, Egyetem tér 1. Győr, Hungary
Tel.: +36 (96) 503-400
E-mail: {paller.gabor,elo}@sze.hu

Summary: This paper presents our experiments with image processing toolkits on microcontrollers through the use case of a an agricultural camera sensor capturing images in multiple spectral ranges. Night animal population estimation requires frequent capture of infrared images and transferring these images to the server is not feasible due to bandwidth limitation and/or power consumption constraints. Hence image processing capability is needed in the sensor. The paper presents the common vole detection algorithm we developed and its power-aware implementation. We emphasize the need for more modular image processing frameworks that can be deployed on microcontrollers more easily. We also present our agricultural camera sensor platform that is suitable for various detection/observation tasks.

Keywords: agriculture, infrared imaging, image processing, power efficiency

1. Introduction

Agricultural sensor use cases include capturing images for e.g. detecting drought, plant phenotype or diseases. These use cases require visible light [6], [7] or infrared imaging [1], [11]. While most applications require relatively simple sensors (e.g. capturing images several times a day), we present a case in this paper which requires more frequent sampling. This observation activity generates significant amount of data and transmitting this data from an isolated, battery-powered sensor operating far from the fixed network infrastructure is not a trivial task. This paper argues that in these use cases significant saving in power consumption can be achieved by implementing image processing capability in the sensor.

The energy consumption balance between data processing at the sensor endpoint vs. data processing at the server has already been analyzed in a general case [8]. In this paper we examine this question in a more special case, namely low-power microcontrollers as processing units, and limited communication options.

2. Common vole detection use case

AgroDat project, financed by the government of Hungary intends to develop connected sensors for the agriculture. One of the more challenging use cases we identified is animal monitoring, specifically rodent tracking. Population outbreaks of certain rodent species can cause significant damage in crop production. More aggressive rodenticides are applied according to population estimation hence this estimation is an economically important task. Detection of wild animals during mowing operations reported by [10] requires similar technical solutions.

As common voles are night animals, the sensor used for population estimation must be able to detect these animals in the darkness. Previously the availability of long-wavelength infrared (LWIR) cameras was limited due to their high cost, therefore short-wavelength infrared (SWIR) cameras (like Kinect [2]) have been used for rodent tracking. SWIR cameras, however, have the disadvantage that the bait area needs to be illuminated by infrared light which limits their effective range. Relatively low-cost LWIR cameras appeared just recently.

We experimented with FLIR Lepton camera module whether small rodents can be detected reliably. The idea is that the rodents are attracted to a bait area which is surveyed by the infrared camera. The FLIR Lepton camera operates in the 8000-14000 nm wavelength range and has a resolution of 80x60 pixels.

We made the following experiment. An animal similar to the common vole (*Phodopus sungorus*) was placed in a cage and images were captured with different distances between the camera and the animal. The background was lawn and other common foliage. The images were made in the night (Fig. 1.)

The infrared camera measures observed temperature values for each pixel. These temperature values are deduced from the infrared radiation observed in the viewport area corresponding to the pixel. In order to obtain an image with visible features, temperature range between the minimum and maximum temperature values in the input, raw image need to be mapped to intensity values (like 0-255 gray-scale) in the gray-scale image that acts as input to the image processing algorithm.

The small rodents we are looking for that are farther from the camera and therefore their observed size is smaller than a pixel size in this relatively low-resolution image look like colder than they actually are because the temperature of the elements of the foliage are calculated into the temperature measured for the pixel in question. The dynamic mapping of the temperature range in the raw input image to gray-scale representation means that as the warm object gets farther from the camera, features in the background get “brighter”.

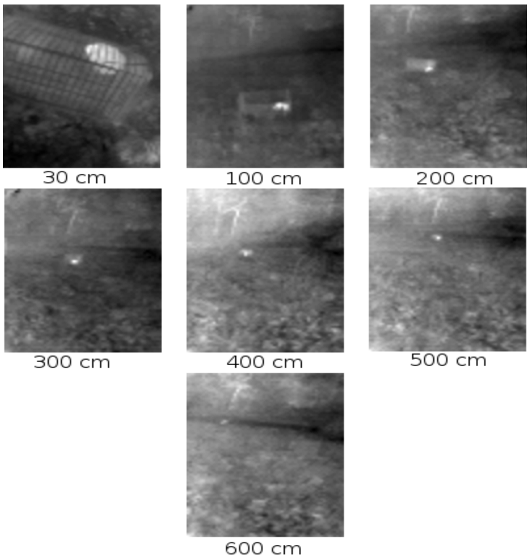

Fig. 1. Small rodent similar to a common vole (Phodopus sungorus) in long-wavelength infrared image.

3. Vole detection algorithm

The goal of the vole detection is to identify images where something relevant is captured. These images are then sent to the back-end server for further, more detailed analysis, eventually yielding the population estimate. Image processing is also important in case of extremely low-bandwidth wireless bearers like Sigfox where sending the image is not feasible. A Sigfox endpoint is able to send just 140 16-byte messages daily so sending the entire image is clearly not possible. The sensor has to extract some characteristic data (like the number of the vole-like objects identified in the image) and send this extracted data over the Sigfox network. The image itself is obtained by some other means (like auxiliary GSM network access providing only batch image upload or physical access to the mass storage (e.g. SD card) of the sensor).

The first version of the vole detection algorithm was implemented in OpenCV. The steps of the algorithm are the following.

- The greyscale image is transformed into a binary image with a fixed threshold of 204.
- Contour tracing algorithm from the OpenCV library is applied, then the resulting contours' convex hull is filled. This step gets rid of spurious noise in the image resulting from the thresholding step.
- Elements in the image are dilated then again contour traced.
- Finally the enclosing circle of each resulting contour is calculated and these circles are compared to the circles obtained from the previous iteration. Largely overlapping circles are eliminated. If a circle is found moving and its size corresponds to the size of a vole, the image is stored and/or uploaded to the server.

In order to ensure that the animal does not leave the image when the next picture is taken but also moves significantly so that the circle representing the animal has sufficiently different location, we found that a frame rate of 1 Hz yields reliable results. Depending on the use case, this frame rate would be sustained continuously or just for a short period of time. We achieved good results by taking 5 consecutive pictures with 1 Hz frame rate then interrupting the image capturing/processing for 1 minute. Compared to the steady 1 Hz frame rate, this burst operation still identified the animals reliably because once they were in the bait area, they remained there for several minutes. On the other hand, the burst operation consumed significantly less power.

We prototyped the algorithm on an embedded Linux platform (BeagleBone Black/TI AM335x 1GHz ARM Cortex-A8) and we found good efficiency in recognizing relevant images. Unfortunately the high standby consumption of these embedded Linux platforms nullified any power consumption savings [3]. The project was therefore moved to a microcontroller unit (MCU) platform. Due to its high performance (internal floating-point unit (FPU), Cortex-M4 core, up to 168 MHz clock speed), large internal flash (512 Kbytes or 1 Mbytes, depending on subtype) and RAM memory (192 Kbytes) we chose the STM32F407 MCU and attempted to port OpenCV's 2 basic modules (core, imgproc) to the MCU. Even these modules required more flash space than the relatively large flash memory of this high-end MCU. The reason is OpenCV's heavily layered software architecture and its extensive usage of support libraries (e.g. libc, libm, libz, STL, etc.) Even though the actual image processing modules are relatively small, extracting them out of the OpenCV dependency network turned out to be too complicated.

We evaluated two additional image processing frameworks. CImg [4] is a C++ template library (hence it has dependency on STL) but it is missing morphological analysis tools needed for our vole detection algorithm. CVIPTools [5] is a quite exhaustive C library but the Linux version on which the STM32F407 port is based was last maintained in 2002. This version of CVIPTools does not support graphics processing units (GPU) either. Curiously, these features are advantages when it comes to using the library on an MCU as pure C implementation eliminates the need of STL support library and not even high-end MCUs have GPU. CVIPTools has the advantage that it depends only on the standard C library (libc). We satisfied this dependency by porting the Newlib library[1] to the MCU. The flash image of the vole detection application with the relevant modules of CVIPTools and Newlib has the size of 126 Kbytes which fits conveniently into the MCU's flash memory. This demonstrates that much more complex image processing algorithms can also be implemented on this platform.

While CVIPTools and OpenCV both offer plenty of algorithms and tools, the tool set is not exactly the same. The CVIPTools version, starting from the second step, employs a different processing.

- In the second step, after the greyscale-to-binary conversion, a morphological dilating is performed followed by a morphological closing and an additional greyscale-to-binary thresholding operation.
- Objects in the image are then labeled, yielding bounding boxes for contiguous objects.
- The enclosing circles are calculated from these bounding boxes. Identification of the overlapping/moving circles is the same as in case of the OpenCV implementation.

CVIPTools (on the STM32F407 MCU) and OpenCV-based implementations (on BeagleBone Black) yield similar outputs and power consumption can be compared. The new, MCU-based implementation ported to CVIPTools consumes 0.0027 mAh when processing 5 consecutive pictures while the previous, embedded Linux-based implementation (OpenCV) needed 0.62 mAh. Moreover, the MCU is able to sleep with microamper-scale power consumption while the embedded Linux implementation consumes significant amount of power even when sleeping. In the previous iteration of the sensor [3], the sensor control logic was off-loaded to the GSM communication module (Telit GL865) that has user software execution feature due to the high power consumption of hardware responsible for the image processing function. The MCU-based implementation eliminated this more complex setup. In addition, the low power consumption in both computing and sleeping phases justifies the image processing capability in the sensor as significant saving is realized when only the relevant images are sent to the server.

We also tried to port CVIPTools and the vole detection algorithm to a much smaller microcontroller, an STM32L152RCT6. This MCU is optimized for ultra-low consumption application, has Cortex-M3 core, no FPU and up to 32 MHz clock speed. The MCU is also equipped with 256 Kbytes of flash memory and 32 Kbytes of RAM. Particularly the relatively small RAM is problematic for image processing applications but as our raw infrared image is just 9600 bytes, there was a hope that our vole detection algorithm fits into the RAM. The size of the application code (vole detection+relevant modules of CVIPTools and Newlib) was 122 Kbytes which compares favorably with the total flash size of 256 Kbytes. No matter how hard we tried, however, the object labeling step required more memory than the about 29 Kbytes available for the C heap. Also, due to the lack of FPU support, (partial) processing of one image required 420 msec which indicates that even if there was enough memory, the desired frame rate of 1 sec would be hard to achieve.

4. The camera sensor

The experiments described in the previous sections led us to construct a multi-purpose agricultural camera sensor. The head unit of the sensor can be seen in Fig. 2. This head unit is usually mounted on a pole so that the vegetation or the bait area (in case of rodent sensor) can be observed. The sensor optionally contains 4 visible-light cameras, positioned 90 degrees from each other and 1 LWIR camera. Power supply of each of these cameras can be enabled separately, allowing the developer of the sensor application to switch on the cameras only when needed.

The sensor is equipped with multiple communication options that can also be deployed optionally. GSM modem provides the capability to perform bulk image upload. Low-power wide area (LPWAN) modem (Sigfox in the current version of the camera sensor) is used for delivering short messages in a power-efficient way – like sending the number of rodents detected in the bait area.

In order to demonstrate the need for multiple communication options, Fig. 3. and Fig. 4. depicts the power consumption of sending a small data item (60 bytes) by GSM/GPRS and Sigfox. The GSM/GPRS modem was Telit GL865, the Sigfox modem was Adeunis Si868. The Sigfox modem was controlled by an Atmel ATmega2560 MCU whose power consumption in this scenario was negligible, the Telit GL865 was controlled by its own, Python-based execution logic. The GSM/GPRS scenario included network registration, PDP context activation, data transmission and network un-registration procedures. Sigfox does not need registration, the power consumption diagrams show the sending of 4 messages as the 60 bytes payload fits only into 4 16-byte Sigfox messages. The result is that GSM/GPRS needs approximately 1 mAh power consumption while the Sigfox scenario requires 0.2 mAh. Also, GPRS maximum power consumption during the scenario is much higher which allows the Sigfox option to be implemented with smaller batteries. In order to transfer data relevant to an image over the extremely low-bandwidth Sigfox network, the sensor unit must extract relevant features from the image by means of image processing. A similar experience has been reported for other low-bandwidth networks operating in the license-free spectrum used to transfer image data [9].

In case of LPWAN communication, there is an option that the relevant images are stored on an SD

1 https://sourceware.org/newlib/

card in the sensor, available off-line (when the service personnel visits the camera sensor). The SD card can also store images for batch upload operations by means of the GSM modem, if that option is installed. Another option is a large, 4 Mbytes RAM that can act as a temporary memory for image processing operations on the large images that the visible-light cameras produce.

Fig. 2. Head unit of the camera sensor

These optional features make the camera sensor a versatile platform whose application areas span from simple foliage observation (with visible-light or LWIR camera) to more complex detection tasks requiring image processing. The STM32F407 MCU does have limitations with regards to complex image processing operations but the relatively powerful ARM core and the extensive feature set of CVIPTools does permit the implementation of reasonably sophisticated image processing. Also, the communication architecture that supports power-intensive but relatively high-bandwidth (cellular) and low-power wide area (Sigfox in our case) network support permits both short message sending with very small power consumption and bulk image uploads.

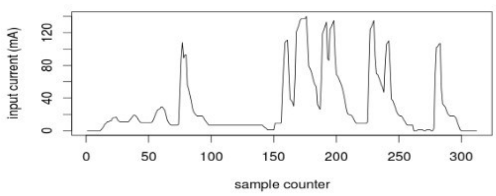

Fig. 3. Power consumption of sending a 60-byte packet by GSM/GPRS.

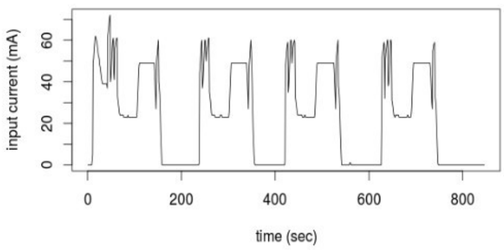

Fig. 4. Power consumption of sending a 60-byte packet by Sigfox.

5. Conclusions

Sensors are often considered to be data capture devices which just transfer the data to more powerful nodes ("servers") where the data is processed. Limited communication bandwidth or limited battery power may require more sophisticated data processing in the sensor. The common vole detection use case presented in this paper aimed to demonstrate that image processing frameworks with complex dependency structures and layered (as opposed to modular) architecture are often unsuitable for low-power environments. Also, the low-power and low-bandwidth communication options like Sigfox require that sensors communicate just the relevant features of the image and not the entire image. It is also often a requirement to transfer the images themselves for further processing on the server. This requires additional transfer mechanisms (off-line or high-power, high-bandwidth communication option) in addition to the low-power, low-bandwidth network.

References

[1]. A. Manickavasagan et al., Applications of Thermal Imaging in Agriculture–A Review, *Canadian Society for Engineering in Agricultultural, Food and Biological Systems (2005): 05-002*.

[2]. Z. Wang, S. A. Mirbozorgi, M. Ghovanloo, Towards a kinect-based behavior recognition and analysis system for small animals, *Biomedical Circuits and Systems Conference (BioCAS), Atlanta, Georgia, USA, 22-24 Oct. 2015*

[3]. G. Paller, G. Élő, Energy-efficient operation of GSM-connected infrared rodent sensor, *SENSORNETS 2016, 5th International Conference on Sensor Networks, Rome, Italy*

[4]. The Cimg Library (http://cimg.eu/).

[5]. CVIPTools (http://cviptools.ece.siue.edu/)

[6]. H.-E. Nilsson, Remote sensing and image analysis in plant pathology, *Canadian Journal of Plant Pathology, Volume 17, Issue 2, 1995.*

[7]. J. Romeo, G. Pajaresa, M. Montalvo, J.M. Guerreroa, M. Guijarroa, J.M. de la Cruz, A new Expert System for greenness identification in agricultural images, *Expert Systems with Applications, Volume 40, Issue 6, May 2013, pages 2275–2286.*

[8]. K. Kumar, Y.-H. Lu, Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?, *Computer, Volume: 43, Issue: 4, pages 51–56.*

[9]. T. Wark et al., Transforming Agriculture through Pervasive Wireless Sensor Networks, *IEEE Pervasive Computing , Volume: 6, Issue: 2, April-June 2007*.

[10]. K. A. Steen, A. Villa-Henriksen, O. Roland Therkildsen, O. Green, Automatic Detection of Animals in Mowing Operations Using Thermal Cameras, *Sensors, Volume 12, Issue 6*

[11]. N. R. Falkenberg, G. Piccinni, J. T. Cothren, D. I. Leskovar, C. M. Rush, Remote sensing of biotic and abiotic stress for irrigation management of cotton, *Agricultural Water Management, Volume 87, Issue 1, 10 January 2007, Pages 23–31*