Energy Consumption and Data Amount Reduction Using Object Detection on Embedded Platform

Boris Snajder(✉), Zoran Kalafatic, and Vedran Bilas

Faculty of Electrical Engineering and Computing,
University of Zagreb, Zagreb, Croatia
boris.snajder@fer.hr

Abstract. High resolution image handling often results with high energy burden for battery-powered devices, such as sensor nodes in WSN. Motivation for this study is assessment of energy consumption of the sensor node with high-resolution camera, featuring image processing. We present a selection of object detection algorithms and evaluate their efficiency. To verify applicability of those algorithms, we acquired image sequence that correspond to applications of pests detection in agriculture. We verified considered algorithms' performances: recall, precision and expected reduction of the data amount. Energy required to execute considered algorithms was measured on ARM processor based platform. Our results show that object extraction on a node can provide reduction of the data amount by up to three orders of magnitude. While simple algorithms can lead to lower overall energy consumption of the node, the more complex algorithm provides better performances, but at a cost of prohibitively high energy consumption.

Keywords: Wireless sensor networks · Energy efficiency · Image processing · Object detection · Data amount reduction

1 Introduction

Wireless sensor networks (WSN) are widely used in environmental monitoring, where sensor nodes acquire information from their surrounding and route them to a server or a user, Fig. 1. Appearance of small and affordable CMOS image sensors, that can be easily used in embedded devices, provides opportunity for WSN usage in visual inspection and monitoring of environment [1,2]. As the sensor nodes are usually battery-powered embedded devices, they have limited energy budget, that ultimately limits their lifetime. Thus, in WSN research field there is big emphasis on power management methods [3,4]. Also, as wireless communication is an energy intensive task, wireless protocols targeted for use in WSN, such as a ZigBee, incorporates various means to reduce sensor node energy consumption [5]. However, data intensive operations, such as image acquisition, processing and transmission, can result with significant increase in energy consumption [6].

M. Magno et al. (Eds.): S-Cube 2016, LNICST 205, pp. 118–129, 2017.
DOI: 10.1007/978-3-319-61563-9_10

In available literature [7,8] description of several sensor nodes with image sensors can be found. However, as a consequence of limited energy and computing resources, all of those sensor nodes feature relatively low resolution image sensor. In some applications, such as pests detection, it is necessary to use a high resolution image sensor [9]. As camera resolution increases, an increase in energy consumption can be expected, as the sensor nodes have to handle even more data. High energy consumption of the node with high resolution image sensor can be met with solar cells [10]. Alternatively, image processing on the nodes can be used to reduce the data amount to be transferred and consequently to lower energy consumption [11]. The data amount reduction in WSN can be achieved either by image compression or by object detection and extraction, where only fragments of the image containing objects of interest, are sent [12]. Depending on the data amount, that the nodes have to send, different wireless protocols provide lower energy consumption [13].

Thus, the goal of our research is an evaluation of several object detection and extraction algorithms, including assessment of their energy consumption in applications that require high resolution image sensors. The remainder of the paper is organized as follows. Section 2 gives an overview of considered algorithms for object detection and extraction. Description of methods and measurement setups, based on low cost embedded platforms, is presented in Sect. 3. Section 4 presents the results and Sect. 5 concludes the paper.

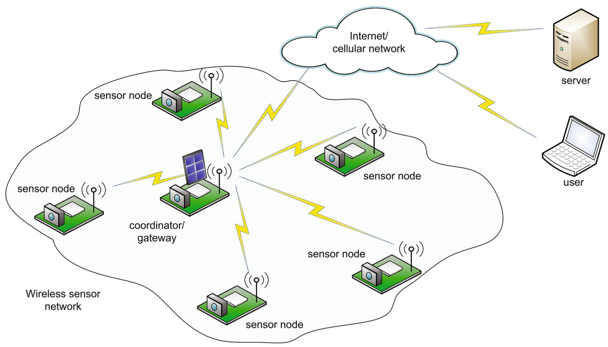

Fig. 1. Illustration of WSN in environmental monitoring application.

2 Object Detection Algorithms

Required resolution of the camera is application driven. For example, in pests detection applications spatial resolution of approximately $50\,\mu\text{m} \times 50\,\mu\text{m}$ is required [9]. To achieve that spatial resolution, if camera field of view is $10\,\text{cm} \times 10\,\text{cm}$, a 5 MPix image sensor should be used. Lossy image compression is not a suitable option for data amount reduction in this application, since size of some anatomy details of pests will be just a few pixels. Thus, focus will be on the algorithms for object detection and extraction.

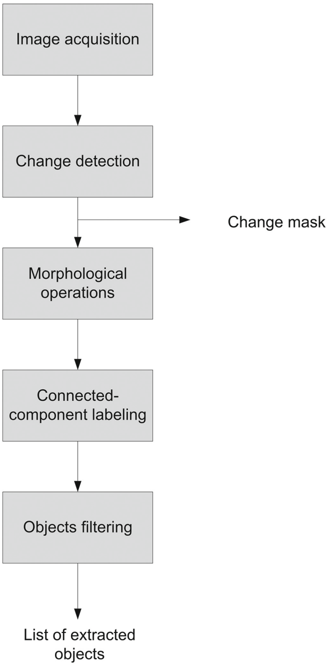

Fig. 2. Objects detection and extraction flowchart

In pests detection applications, the sensor nodes acquire images of sticky traps. Static nature of background in this application reduces the object detection problem to a change detection problem [12,14]. Steps required for objects detection and extraction are depicted in Fig. 2, with change mask estimation as the first step. As change mask contains information about changed pixels, this step will define overall performance, i.e. recall and precision [15], of the algorithms. In the case of static and homogenous background, the appearance of objects can be detected by various change detection algorithms. We considered three typical approaches [14,16]:

- Background subtraction,
- Color change detection,
- Difference of Gaussians.

Morphological operations, connected component labeling and size filtering are used for objects extraction from change mask. Result of objects extraction is a list of objects, defined by their positions and sizes. As the expected size of pests is known *a priori*, filtering can be applied to discard objects that are either too big or too small.

2.1 Background Subtraction

Except possible changes in scene illumination, background is static and homogenous, while trapped pests appear as objects of lower intensity. Thus, they can be detected by subtracting pixels intensity of current image $I_N(\mathbf{x})$ from reference background image $I_0(\mathbf{x})$:

$$I_{DIFF}(\mathbf{x}) = I_0(\mathbf{x}) - I_N(\mathbf{x}). \tag{1}$$

As subtracting is done on pixels intensity, firstly conversion to grayscale image is done. To eliminate small changes, that are result of noise and variances in illumination conditions, a threshold operation is applied and a change mask $C_N(\mathbf{x})$ is obtained:

$$C_N(\mathbf{x}) = \begin{cases} 1, & \text{if } I_{DIFF}(\mathbf{x}) > \tau \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where τ is value of threshold. If pixel intensity values are in interval $[0, 1]$, difference $I_{DIFF}(\mathbf{x})$ and τ values are in interval $[-1, 1]$. Usually, threshold value is empirically chosen. This is the most simple method in determining that some pixel has changed. As variations in illumination conditions change intensity of pixels, it can be expected that this method will be sensitive to changes in illumination [14].

2.2 Color Change Detection

Typical color of sticky traps used in pests monitoring is yellow. Thus, color information of each pixel can be used to detect change. Color image sensors usually provide color information in RGB or YUV color spaces, while the hue of color, and thus deviation from the expected value, is easily defined in HSV color space. Hue component $H(\mathbf{x})$, that carries color information, can be converted from RGB color space using expression:

$$H(\mathbf{x}) = \begin{cases} \frac{G-B}{max(R,G,B)-min(R,G,B)} \cdot 60^\circ, & \text{if } max(R,G,B) = R \\ \frac{B-R}{max(R,G,B)-min(R,G,B)} \cdot 60^\circ + 120^\circ, & \text{if } max(R,G,B) = G \\ \frac{R-G}{max(R,G,B)-min(R,G,B)} \cdot 60^\circ + 240^\circ, & \text{if } max(R,G,B) = B, \end{cases} \tag{3}$$

where R, G, B are values of pixel color components in RGB color space. Values of the hue component are in interval $[0^\circ, 360^\circ]$, while hue value of yellow pixel is around 60°. Thus, the change mask can be found by thresholding all pixels whose hue component differ from background for more than threshold value τ:

$$C_N(\mathbf{x}) = \begin{cases} 1, & \text{if } H(\mathbf{x}) < (C-\tau) \text{ or } H(\mathbf{x}) > (C+\tau) \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In comparison to the background subtraction approach, color change detection does not require reference image.

2.3 Difference of Gaussians

Difference of two low-pass Gaussian (DoG) filters can be used show local changes in pixel values, thus it can be used for object detection [16]. Difference of Gaussians $D(\mathbf{x}, \sigma, k)$ can be expressed as:

$$D(\mathbf{x}, \sigma, k) = (G(\mathbf{x}, k\sigma) - G(\mathbf{x}, \sigma)) * I_N(\mathbf{x}), \tag{5}$$

where σ is standard deviation of the first Gaussian filter, $k\sigma$ is standard deviation of the second Gaussian filter. If there is big enough local change in pixel intensity, such as pest on sticky trap, there will be increase in value of $D(\mathbf{x}, \sigma, k)$. As pests size is known, it is possible to run only one DoG, with predetermined values of standard deviation that maximizes DoG response for the targeted application. The values of σ and k, that provide the best object detection performance, are found empirically. To provide better energy efficiency, the Gaussian filters were implemented using 1-D filters. Before filtering, the image is converted to grayscale.

3 Methods and Experimental Setups

Delivery of reliable information is as much important as energy consumption of the node. Thus, it is necessary to verify performance of objects detection algorithms. For that purpose algorithm performances were evaluated on image sequences, including measurement of energy consumption.

3.1 Image Acquisition

To be able to verify object detection algorithms in different conditions, an image sequence containing more than hundred images was acquired using a measurement setup depicted in Fig. 3. The measurement setup is consisted of a mechanical mounting, a yellow sticky trap, a Raspberry Pi and a 5 MPix OV5640 camera. A software run on the Raspberry Pi acquired a new image every hour during daylight. A result is acquisition of images in different illumination conditions, ranging from direct sunlight on the sticky trap, to the illumination conditions of cloudy days. When the sticky trap is mounted to the setup, the software was restarted triggering immediate image acquisition. Thus, the first image in the sequence does not contain any objects and can be used as the background reference image. All acquired images were stored without compression in RGB color space and indexed to ease algorithms evaluation, resulting with image size of approximately 14.4 MB.

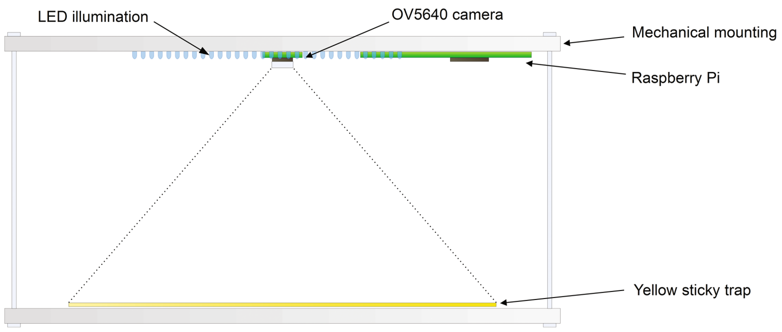

Fig. 3. Illustration of setup for image acquisition

3.2 Algorithms Performance Evaluation

For the purpose of performance evaluation, the considered algorithms were implemented in MATLAB, while recall, precision and F-measure were used for the algorithms performance evaluation. To reduce run time, assessment of influence of threshold and standard deviation values on algorithms performance was done a smaller subset consisted of four images. On the each test image at least a dozen objects were present, while size and position of all objects were manually labeled. Thus, for every test image there was associated labeled list with bounding boxes of all objects used as the ground truth. A list of extracted objects from the test images is then compared to the list of labeled objects. Recall R and precision P are calculated using expressions:

$$R = \frac{TP}{TP + FN}, \tag{6}$$

$$P = \frac{TP}{TP + FP}, \tag{7}$$

where TP is number of true positives, FN is number of false negatives and FP is number of false positives. Extracted object is considered as true positive if its center is located within area defined by some object from labeled list. Number of false positives FP equals to the number of extracted objects that do not match any object from labeled list, while number of false negatives FN is number of objects in labelled list that weren't matched by any extracted object.

Threshold value τ, and values of σ and k in case of DoG, affect recall and precision. Typically, selection of parameters that leads to increase in precision will result with lower recall, and *vice versa*, while in ideal scenario both values would be equal to one. Thus, to ease comparison of algorithms F-measure can be used:

$$F_{meas} = 2\frac{R \cdot P}{R + P}. \tag{8}$$

After algorithms parameters that provide the best performance on the test images set are determined, each algorithm was run on the whole image sequence to evaluate its performances in different illumination conditions.

As primary goal is to lower overall energy consumption of the node, the expected amount of data required to describe extracted objects is estimated for each algorithm. Expected amount of data then can be used for estimation of energy required for wireless communication.

3.3 Energy Consumption

The measurement setup used for energy consumption during algorithms execution is shown in Fig. 4. The embedded platform used for energy consumption measurement is based on AllWinner A20 dual core processor. Power supply voltage was 5 V. The current consumption of the embedded platform is measured over a shunt resistor and a National Instruments NI USB-6221 acquisition card [17], with the sampling rate of 100 ksps and the resolution of 16 bits. Processor executed algorithms implemented using OpenCV 3.0 library [18]. To ensure maximal performance, OpenCV library was compiled with NEON instructions support enabled. Start and completion of each step in object detection and extraction was signaled using GPIO lines. Energy consumption of algorithms' each step was calculated by integration of power supply voltage and measured current product. Algorithms were run on 5 MPix images and each measurement was repeated 15 times.

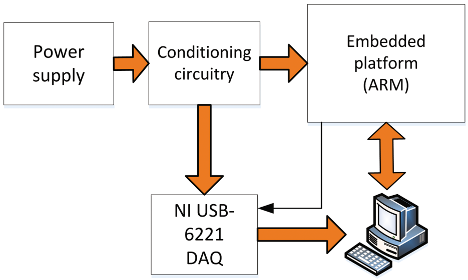

Fig. 4. Setup for measuring energy consumption of embedded platform during algorithms execution

4 Results

One of the acquired images is shown in Fig. 5. As expected, the background of the image is yellow, while pests show as darker objects on the background. Typical size of trapped pests range from 10 to 30 pixels. Also, nonuniform illumination of the scene, caused by surrounding vegetation, can be observed.

Fig. 5. A representative image acquired with the set-up given in Fig. 3. (Color figure online)

4.1 Performance of the Algorithms

Performances of the considered algorithms and their dependence on threshold value is evaluated through a recall-precision curve, Fig. 6. As expected, with threshold value selection each algorithm can be adjusted towards achieving higher recall or higher precision. The background subtraction algorithm can achieve recall $R = 1$, but at the same time precision drops to $P \approx 0.7$. Maximal value of F-measure the background subtraction algorithm achieves is $F_{meas} = 0.85$ for the threshold value $\tau = 0.25$. Decreasing threshold value results with higher recall and lower precision. While the color change detection algorithm achieves almost the same maximal value for F-measure $F_{meas} = 0.84$, its overall performances are lower compared to the performance of the background subtraction algorithm. Maximal value of F-measure is achieved for $\tau = 7.5°$. As in the case of the background subtraction algorithm, increase of threshold value decreases recall and increases precision. The best performance on the test images achieves DoG algorithm with precision $P = 1$, recall $R = 1$ and $F_{meas} = 1$, thus ideal detection. This is achieved for the threshold value of $\tau = 0.05$. The value of standard deviation was set to $\sigma = 10$ and $k = 4$, as those settings provided the best performance on the test images.

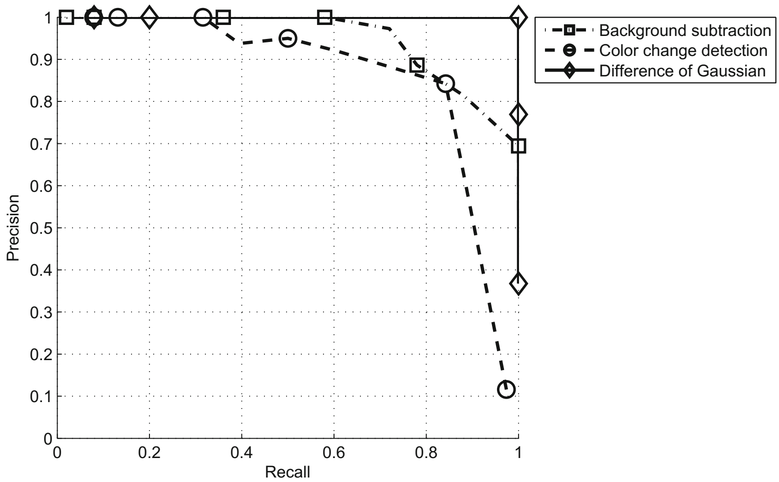

Fig. 6. Precision-recall curve of considered algorithms run on the test images

Based on the performance evaluation on the test images, the parameter for each algorithm were set and the algorithms were evaluated on the whole image sequence. The results show that DoG performs best in all illuminations conditions, although for some of the images it does not achieve values of recall or precision equal to 1. The DoG achieves recall higher than $R > 0.8$ on all images. As the background subtraction is susceptible illumination variations, change in illumination causes recall and precision to drop as low as 0.6. The color change detection algorithm is much more susceptible to illumination changes, so we observed a significant drop in recall and precision with non-homogeneous illumination, where for some images precision was lower than $P < 0.1$.

The average data required to describe detected objects with the considered algorithms are given in Table 1. If all extracted objects are sent as detected, the average data amount, depending on the used algorithm, is in the range between 26 kB and 52 kB. The average data amount to be sent can be further reduced by sending only newly detected objects. In that case the DoG and the background subtraction algorithms result with approximately 4.5 kB of data. As uncompressed full resolution image was more than 14 MB in size, data reduction in all cases is significant. Higher average data amount that the color change detection generates is result of its lower precision. Blurring of objects' edges due to low-pass filtering results with overestimated objects size and the higher data amount that the DoG algorithm generates compared to the background subtraction algorithm.

4.2 Energy Consumption

Measured energy consumption required for execution of each step of the considered algorithms is shown in Fig. 7. The background subtraction algorithm achieves

Table 1. Average amount of data

Algorithm	Average amount of data in list of extracted objects	Average amount of data in difference between lists of current and previous image
Background subtraction	26.217 kB	4.576 kB
Color change detection	52.186 kB	32.045 kB
Difference of Gaussians	45.887 kB	4.514 kB

the lowest energy consumption of $E_{BS} = 3.783 \pm 0.029$ J, while the execution time is approximately 2 s. The step of change mask estimation takes only 68 ms to execute and it consumes approximately 125 mJ of energy. Steps that are required to extract objects from the change mask consume around 3.33 J, while conversion from RGB color space to grayscale image consumes additional 326 mJ.

More complex operation required for the estimation of change mask using the color change detection algorithm results in higher energy consumption. The change mask estimation in this case consumes 1.73 J. As the color change algorithm does not require conversion to grayscale image and the steps required for objects extraction from the change mask are the same as in the case of the background subtraction algorithms, total energy consumption of the color change algorithm is $E_{CC} = 5.270 \pm 0.032$ J.

As the standard deviation used in the DoG has to reflect the expected size of objects to be detected, using 3σ rule results with a kernel size of 60×60 pixels. This results with a long execution time for the change mask estimation, and consequently with high energy consumption of approximately 70 J. Total energy consumption for the object extraction using the DoG algorithm is $E_{DoG} = 73.487 \pm 0.784$ J, while the execution time is approximately 43 s.

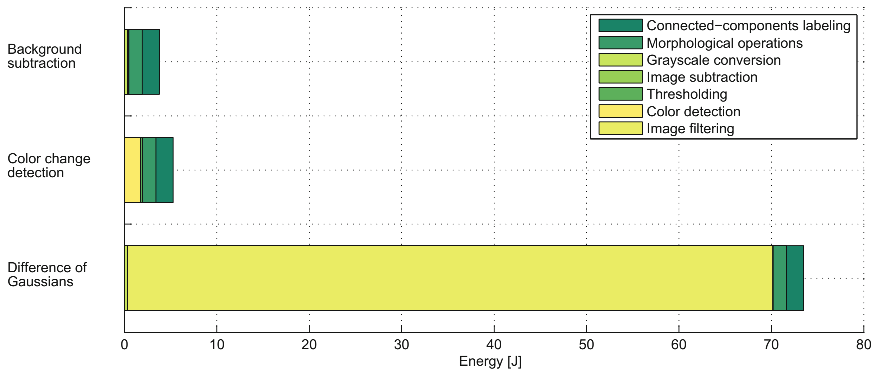

Fig. 7. Measured energy consumption for execution of each step of the considered algorithms.

5 Conclusion

Presented results show that considered algorithms are suitable for detecting objects on static background, while the DoG achieved the best performance. However, when run on the representative image sequence, all algorithms show drop in performance due to nonhomogeneous illumination conditions. As the DoG algorithm is sensitive to local changes in illumination, it achieves the best performance even in those conditions, while the color change detection performs poorly. Further, a lower data amount that the DoG achieves when comparing extracted objects from current and previous image, suggest more robust object detection compared to the background subtraction algorithm.

Better performance of the DoG algorithm comes at cost of significantly higher energy consumption. According to energy consumption analysis of wireless transfer presented in [13], transmission of 10 MB of data using ZigBee requires around 60 J of energy, while Wi-Fi requires around 9 J. Thus, image processing on the node can provide lower overall energy consumption when the background subtraction or the color change detection algorithms are used. However, energy consumption of DoG algorithm is higher than energy required to transfer whole 5 MPix image. Thus, when considering usage of image processing as means for node energy consumption reduction, it is required to take into account the energy constraint of the node, efficiency of the wireless communication, expected amount of data and required reliability of object detection.

Further, as results suggest lower performance in nonhomogeneous illumination conditions, in future we will assess possibility of using LED illumination.

References

1. Pham, C.: Communication performance of low-resource sensor motes for data-intensive applications. In: 2013 IFIP on Wireless Days (WD), pp. 1–8. IEEE (2013)
2. Jeličić, V., Ražov, T., Oletić, D., Kuri, M., Bilas, V.: Maslinet: a wireless sensor network based environmental monitoring system. In: MIPRO: Proceedings of the 34th International Convention, pp. 150–155. IEEE (2011)
3. Asorey-Cacheda, R., García-Sánchez, A.J., García-Sánchez, F., García-Haro, J., González-Castano, F.J.: On maximizing the lifetime of wireless sensor networks by optimally assigning energy supplies. Sensors **13**(8), 10219–10244 (2013)
4. Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: Energy conservation in wireless sensor networks: a survey. Ad Hoc Netw. **7**(3), 537–568 (2009)
5. Oliveira, L.M., Rodrigues, J.J.: Wireless sensor networks: a survey on environmental monitoring. J. Commun. **6**(2), 143–151 (2011)
6. Akyildiz, I.F., Melodia, T., Chowdhury, K.R.: Wireless multimedia sensor networks: applications and testbeds. Proc. IEEE **96**(10), 1588–1605 (2008)
7. Tavli, B., Bicakci, K., Zilan, R., Barcelo-Ordinas, J.M.: A survey of visual sensor network platforms. Multimedia Tools Appl. **60**(3), 689–726 (2012)
8. López, O., Rach, M., Migallon, H., Malumbres, M., Bonastre, A., Serrano, J.: Monitoring pest insect traps by means of low-power image sensor technologies. Sensors **12**, 15801–15819 (2012)

9. Boissard, P., Martin, V., Moisan, S.: A cognitive vision approach to early pest detection in greenhouse crops. Comput. Electron. Agric. **62**(2), 81–93 (2008)
10. Fukatsu, T., Watanabe, T., Hu, H., Yoichi, H., Hirafuji, M.: Field monitoring support system for the occurrence of leptocorisa chinensis dallas (hemiptera: Alydidae) using synthetic attractants, field servers, and image analysis. Comput. Electron. Agric. **80**, 8–16 (2012)
11. Ferrigno, L., Marano, S., Paciello, V., Pietrosanto, A.: Balancing computational and transmission power consumption in wireless image sensor networks. In: IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS), Giardini Naxos, Italy, 18–20 July 2005
12. Aziz, S.M., Pham, D.M.: Energy efficient image transmission in wireless multimedia sensor networks. IEEE Commun. Lett. **17**(6), 1084–1087 (2013)
13. Snajder, B., Jelicic, V., Kalafatic, Z., Bilas, V.: Wireless sensor node modelling for energy efficiency analysis in data-intensive periodic monitoring. Ad Hoc Netw. **49**, 29–41 (2016)
14. Radke, R., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. IEEE Trans. Image Process. **14**(3), 294–307 (2005)
15. Powers, D.M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation (2011)
16. Kong, H., Akakin, H.C., Sarma, S.E.: A generalized Laplacian of Gaussian filter for blob detection and its applications. IEEE Trans. Cybern. **43**(6), 1719–1733 (2013)
17. DAQ M Series NI USB-621 x User Manual, National Instruments (2009). http://www.ni.com/pdf/manuals/371931f.pdf
18. Bradski, G., Kaehler, A.: OpenCV Library: Computer Vision with the OpenCV Library. O'Reilly Media Inc., Sebastopol (2008)