A Wireless Sensor Network Application with Distributed Processing in the Compressed Domain

Mauricio González, Javier Schandy, Nicolás Wainstein, Martín Bertrán, Natalia Martínez, Leonardo Barboni, and Alvaro Gómez(✉)

Facultad de Ingeniería, Instituto de Ingeniería Eléctrica, Universidad de la República, Montevideo, Uruguay
{mgonzalez,jschandy,nwainstein,mbertran, nataliam,lbarboni,agomez}@fing.edu.uy

Abstract. Wireless Sensor Networks are being used in multiple applications and they are becoming popular particularly in precision-agriculture and environmental monitoring. Their low-cost enables to build distributed deployments with large spatial density of nodes. They have been traditionally used to build maps describing scalar fields varying in time and space. However, in the recent years, image capturing capable nodes have appeared allowing to measure more complex data but imposing new challenges for the processor and memory constrained nodes.

Transmission of large images over a Wireless Sensor Network is a costly operation since most of the power consumption at the node is due to the operation of its radio. Hence, it is desirable to process and extract interesting features from the images at the node in order to transmit the important information and not all the images. However, image processing is also complicated by low processor and memory resources at the node. An image is usually delivered in JPEG format by the node's camera and stored in flash memory but, with current typical node configurations, memory resources are insufficient to open the image file and perform the image processing algorithms on the pixels of the image. To overcome this limitation, image processing can be done in the compressed domain parsing the JPEG file and working directly on the Discrete Cosine Transform coefficients of the compressed image blocks as soon as they are decoded. In this article, we present an agricultural Wireless Sensor Network application that implements block based classification in the compressed domain. In this application, image-sensor nodes are placed on insect pest traps to quantify pest population in fruit trees.

Keywords: Wireless sensor network · Pest monitoring · Compressed domain · Block based classifier · JPEG · DCT

The authors are grateful for the support of CSIC-Universidad de la República and INIA-FPTA Research Project 313.

1 Introduction

Wireless Sensor Networks (WSN) are comprised of small programmable devices named nodes. Each node is composed of a micro-controller, sensors and a radio to communicate wirelessly with neighboring nodes. Their increasingly widespread usage is due to the following reasons: (i) the node low-cost enables to build distributed deployments easily scalable with large spatial density of nodes per unit area with reduced cost of installation and maintenance, (ii) these deployments enable to build maps describing scalar fields varying in time and space (e.g. temperature, humidity), (iii) the nodes operate with low current consumption, so that they can achieve several months of battery lifetime.

Most WSN applications have been traditionally restricted to scalar measuring nodes, see for example [10,14], but recently image capturing capable nodes are being integrated with the challenge of handling more complex data over the networks. With this new scenario, image processing at the node becomes important to reduce radio transmissions in order to keep the current consumption bounded and hence achieve long battery lifetime. Some works in this line are presented in [6,12].

WSNs have received considerable research and development effort in order to enhance their capabilities to be used for precision agriculture and environmental monitoring. In these areas, WSNs enable to manage the farm productivity, allowing product quality enhancement with reduced operational cost.

Image processing in the compressed domain refers to a wide range of algorithms that can be performed directly on compressed images with no prior decompression or with only partial decompression. Research in this field started in the nineties (see for example [3,15]) but the increase in power and memory resources of computers left this kind of processing almost unnecessary for most applications. With plentiful resources, compressed images can be firstly decompressed in order to apply algorithms for the usual spatial domain. However, with restricted processing and memory resources, processing in the compressed domain may make sense constituting a powerful tool. WSNs are one of these applications with low computing and memory resources that can benefit from compressed domain techniques. WSNs are not designed to transmit large amounts of data and transmitting images implies having to modify the existing network protocols to increase the performance of the system. Having the possibility to process efficiently an image at the node and transmit only interesting information, not only enhances the battery life time by reducing the amount of data transmitted (resulting in decreasing the time the radio is in active mode), but also widely simplifies the network protocols as the problem turns into a classical WSN application, where few bytes are transmitted periodically.

In WSNs, cameras that can be attached to a wireless node usually have an integrated JPEG compression engine and deliver a JPEG file. Accessing the image file stream and processing in the compressed domain is a task that can be performed efficiently in the node even with low resources.

In this article, we present an agricultural WSN application that implements block based classification in the compressed domain. The work is part of an ongoing project between our University and fruit producers that looks forward to deploying a distributed pest monitoring system. As a first step in this project, an image-sensor node is being designed capable of periodically taking an image of the inside of a pest trap, analyze the image and deliver information via the radio channel. Some previous related work with image capable WSNs in agriculture can be found in literature such as [9,16], but, up to our knowledge, the approach of block based classification in the compressed domain at the node has not been implemented before.

Following this introduction, Sect. 2 presents the main concepts of processing in the compressed domain and introduces block based classification. Section 3 introduces the pest monitoring WSN application, the implementation of the image node and the compressed domain processing at the node. Results of block based classification are shown on artificial and real images of trapped insects. Finally Sect. 4 presents some concluding remarks.

2 Processing in the Compressed Domain

2.1 JPEG Compressed Images

JPEG is a standard for lossy compression and codification of images. In the basic version of the standard, the image is tessellated in 8×8 blocks and each block is transformed from the spatial domain to the frequency domain using the Discrete Cosine Transform (DCT). For each block, the first DCT coefficient is called the DC coefficient and it is equivalent to the average intensity value of the block, the rest are known as AC coefficients associated to the other frequency components present in the block. The DCT coefficients of each block undergo quantization using different weights adapted to the different frequency components according to their perceptual importance.

This process of quantization is the step that introduces loss since the DCT coefficients are divided by their respective quantization weights and the result is rounded off. After quantization, the remaining non zero DCT coefficients of an 8×8 block are ordered following a zig-zag pattern and subsequently zero run length encoded and codified using Huffman variable length coding. When the image has colors they are represented in the luminance-chrominance YCbCr space. The Cb-Cr components have less bandwidth than the luminance so they can be further downsampled. These components can be downsampled in different proportions, for example 4:1:1 which means that for every 4 luminance blocks there is only 1 Cb and 1 Cr block. A group of such luminance-chrominance blocks (4 Y, 1 Cb and 1 Cr in the example) form a Minimum Coded Unit (MCU). The resulting JPEG file consists of a header with image information and the applied quantization and Huffman tables, followed by the variable length coded blocks. A complete explanation of the JPEG standard can be found in [17]. Figure 1 presents a diagram of the compression process in JPEG.

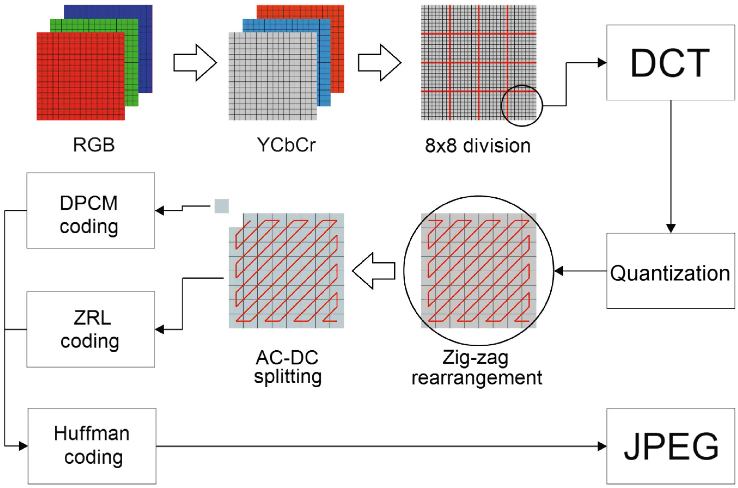

Fig. 1. Basic block diagram of the JPEG standard.

2.2 Compressed Domain Processing

The decompression of a JPEG file follows the inverse sequence of steps. Image processing in the compressed domain can be performed at different stages of the decompression process. In increasing computational complexity, the data that can be used from the JPEG file is:

- Coding length of each block:
  The coding length of a block is the number of bits that are used to represent the block in the JPEG file. This value is a measure of the complexity of the block as it is related to the entropy of the block.
- Decoded DC component of each block:
  Only the first DCT component of each block is kept while decoding. The DC component is the mean value of the block.
- Decoded DCT components of each block:
  All the DCT components of the blocks are decoded, the DC and the AC components. The coefficients can be used individually or combined as different measures of the energy of the block at several frequency sub-bands.
- Inverse DCT transformed values of selected blocks:
  The pixel values of a block can be obtained by inverse transforming the DCT coefficients if needed for some selected blocks.

Processing in the compressed domain can be performed using some or all of this data, according to the selected algorithm. Apart from the type of data, processor and memory requirements of an algorithm are less when it operates sequentially one block at a time. Requirements rise when the algorithm uses information of all the blocks and/or spatial relationships between the blocks which implies more than one traverse of the JPEG file or memory allocation for temporary data.

Since it is possible to recover the DCT coefficients, and the DCT transform is linear, every linear operation on the spatial domain can be ported to the compressed domain. Also some non linear methods can also be ported to the compressed domain. Examples of the algorithms used in the compressed domain are: (i) Arithmetic pixelwise operations [15], (ii) Contrast enhancement [7], (iii) Linear Filtering [18]. A more extensive compendium of image and video processing in the compressed domain is presented in [11].

Most relevant to this article, block based classification can be implemented efficiently in the compressed domain since good features to describe the contents of a block can be easily computed from the DCT coefficients. Several classical supervised classifiers can be trained offline with powerful processing resources and used afterwards as block based classifier with low processing requirements.

2.3 Block Based Classifier

Block based classification is an operation that can be performed block by block as soon as they are decoded. Once decoded, the 8×8 blocks are then classified as a unit, allowing for image segmentation, and/or selective transmission of blocks that contain relevant information.

An example of a simple, low computing cost supervised classifier can be implemented using Fisher's linear discriminant analysis (LDA) [4]. The classifier can use easy to compute features on the DCT coefficients of the blocks. This method finds the direction in which the feature vectors can be projected that gives the best separability between classes. The training of the classifier can be done offline, thus limiting the required online processing to calculating the features, applying a dot product with the projection vector, and classifying via a threshold.

Figure 2 shows some results on the "17 Category flower dataset" from the Visual Geometry Group (U.of Oxford) [13] (hereinafter referred to as Flowers dataset) and the "Airplane dataset" provided by Caltech Vision Group [1] (hereinafter referred to as "Airplanes" dataset). A block based classifier was trained with 450 images and tested on 250 images from the Flowers dataset. These images were randomly chosen from those that had a ground truth to compare to. A classifier was also trained for the "Airplanes" dataset In this case, 267 images were selected from those that had the sky as background and the set was randomly divided in 157 images for the training set and the remaining 110 for the test set. The features chosen on the DCT coefficients were the DC component, total AC energy of the block, relative energy of the first, second and third bands (15 features, 5 features for each image channel Y, Cb an Cr).

Considering $D = (d_0, d_1, ..., d_{63})^t$ the vector of DCT coefficients of a block in zig-zag order for one of the Y,Cb,Cr channels the features are computed as:

$$\left(d_0, \sum_{i=1}^{63} {d_i}^2, \frac{\sum_{i=1}^{2} {d_i}^2}{\sum_{j=1}^{63} {d_j}^2}, \frac{\sum_{i=3}^{5} {d_i}^2}{\sum_{j=1}^{63} {d_j}^2}, \frac{\sum_{i=6}^{9} {d_i}^2}{\sum_{j=1}^{63} {d_j}^2} \right) \tag{1}$$

Table 1. Confusion matrices at an operating point with a classifier threshold set to 1.5

		Predicted positive	Predicted negative
Flowers	Actual positive	83.83 %	14.75 %
	Actual negative	16.17 %	85.25 %
Airplanes	Actual positive	85.17 %	6.94 %
	Actual negative	14.83 %	93.06 %

Figure 2 shows the results on some images of the databases. Table 1 presents the confusion matrices for the performance on the training sets of both datasets at an operating point with a classifier threshold of 1.5.

Fig. 2. Results of block based classification. Please refer to text for complete explanation.

The classifier seems adequate for extracting the objects of interests from the background. Thus, the desired objective of selective compression can be easily met without much loss in relevant information. If only the blocks corresponding to detected objects were to be transmitted, analyzed or stored, the "interesting blocks" to whole image blocks ratio would be 42 % for "Flowers" and 13 % for "Airplanes". Note that these numbers depend largely on the proportion of "interesting blocks" to background in the image and greater efficiencies are achieved in sparsely populated images.

3 The WSN Pest Monitoring Application

Processing in the compressed domain is applied to a WSN for monitoring pest population in fruit trees. In this application, images of pest traps are acquired and analyzed to control pest population on a daily basis. Images could be transmitted over the network to be collected and analyzed on a server but this imposes large traffic in the network that implies complex network protocols and excessive energy consumption at the nodes. Reduction of transmission payload is mandatory to achieve a simple network and low energy consumption of the nodes. With this objective, block based classification is a useful tool to detect the number of insects in the trap and transmit one radio packet with this information instead of transmitting hundreds of packets with the complete image.

3.1 Application Description

The lepidopterous insect pest (moths) produces diseases in trees. The moths lay eggs from which larvae are born and they produce lesions to the fruit. The control of the pest population is implemented by means of using plastic traps with a sticky bottom side and pheromone lures. The trap can capture male adult moths attracted by the female pheromone lures. A person, who periodically travels through crops, is in charge of performing the counting of insects caught in the trap and eventually clean trap bottoms of pest crowded traps.

In an ongoing project between the University and fruit producers, a wireless image-sensor node is being designed capable of taking images inside the trap, analyze the images at the node and transmit the important information via radio channel. The final pest monitoring system will: (i) enable simple pest monitoring of large areas, (ii) simplify the maintenance of the traps since the person in charge will only be required for trap cleaning when needed, (iii) enable early alerts in case of pest infection thus allowing performing localized fumigation in the crop with reduced pesticide usage and hence avoiding environmental and water pollution.

3.2 The Wireless Image-Sensor Node (WimSN)

The designed WimSN is based on the CM3000 [5] node that features the TI MSP430F1611 16-bit RISC microprocessor with a program memory flash of 48 kB, data RAM of 10 kB and an external flash of 1 MB. The wireless communication is implemented by means of the RF Chip TI CC2420 (IEEE 802.15.4 2.4 GHz standard compliant). The selected camera module is the LinkSprite JPEG Color Camera TTL Interface-Infrared LS-Y201 [8]. It requires DC 3.3 V voltage power supply and the current consumption is around 80 mA. The camera module can capture VGA/QVGA and lower resolution images, it integrates a JPEG compression engine, serial communication and 60 degrees viewing angle lens.

The WimSN uses Contiki OS [2], an Event-Driven real time operating system (RTOS) oriented to WSN applications in constrained hardware. This RTOS manages the hardware resources and includes different libraries such as network stacks and a file system. Contiki OS scheduler manages sleep modes powering down the microprocessor when there is neither processing needed nor events scheduled in the event queue.

Figure 3 shows the implemented block diagram.

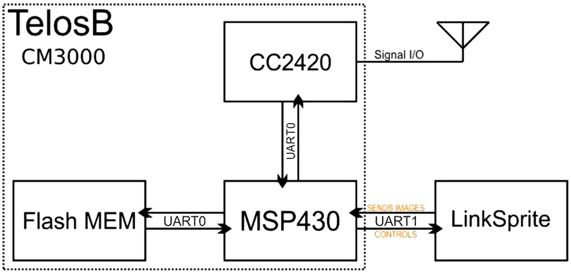

Fig. 3. Block diagram of the implemented wireless image-sensor node.

The WimSN is enclosed in a watertight compartment and it is attached to an acrylic delta shape trap. The trap bottom (which collects the trapped insects and is photographed by the WimSN camera) hangs from the delta trap and can be easily dismounted for cleaning or inspection. Figure 4 shows the 3D design and the implemented device. Figure 5 presents images acquired with the WimSN in the designed trap.

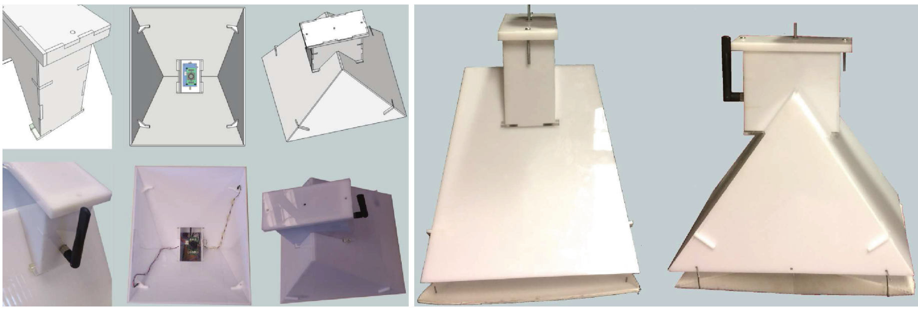

Fig. 4. The WimSN integrated to the pest trap. Left: 3D design and implemented device showing the exterior and interior of the trap. Right: the trap with the hanging bottom.

3.3 Block Based Classification

Implementation at the Node. As explained in Sect. 3.2, the microcontroller receives a JPEG compressed image from the camera and stores the image in flash memory. In order to apply LDA, the JPEG file has to be parsed and the DCT blocks have to be decoded from the file stream but no further decompression is needed. A full JPEG decompression is not only impractical but even impossible due to the reduced hardware capabilities[1], so authors used a custom minimal decoder. The implemented decoder does not perform a full JPEG decompression but obtains the DCT coefficients of each Y, Cb and Cr block per block.

A block based classifier based on LDA was implemented as explained in Sect. 2.3. In the case of the WimSN, the implemented classifier uses only features from the Y luminance channel (DC component, total AC energy of the block, relative energy of the first, second and third bands).

After acquiring an image, the JPEG file is parsed and each block is classified as soon as it is decoded. The classification indicates if a block is part of an insect or is part of the background and the coordinates of the positive blocks are stored in a list that can be transmitted afterwards.

Optionally, under the hypothesis that the moths are not overlapped, a customized labeling algorithm can be applied in order to reduce the amount of stored data. The size of a moth in the image is bigger than an 8×8 block, so one moth is detected in several blocks. Taking advantage of the fact that DCT blocks are decoded row by row of the image, when an horizontal burst of positive classified blocks is detected, only the coordinates of the middle block is stored. After the classification of all the blocks, the list is inspected to identify and merge vertically connected regions. Therefore, the system is capable of establishing the coordinates near the centroid of each insect.

Results. In the current stage of the project, trap images are still insufficient in order to train and test thoroughly the classification of insects. The classification algorithm implemented at the node is tested on a database augmented with images of artificial insects. To build this database, images were acquired with insects simulated with objects of similar shape and taking into account the variability of other aspects (variable illumination, dirt at the trap bottom, etc.). Figure 5 shows images of the trap bottom with real and simulated insects acquired with the WimSN.

Figure 6 shows an image acquired in the designed trap and the result of block based classification.

The training and testing sets of 8×8 blocks are built by manual segmentation of the insects in the images. A block in an image is considered an insect block if at least the 80 % of it is covered by a segmented insect. A dataset of 28 images (134400 8×8 blocks) was considered for testing and training. Although not extensive to do a complete train/test performance analysis, the primary results are encouraging. Considering training sets of 22 images (105600 blocks) and testing sets of 6 images (28800 blocks) the system shows an area under the curve (AUC) of the Receiver operating characteristic (ROC) of over 0.85.

[1] Note that the included flash memory would be almost filled completely with a full RAW image leaving little space for other data (considering that a 3 channel, 640×480 file occupies more than 900 kB).

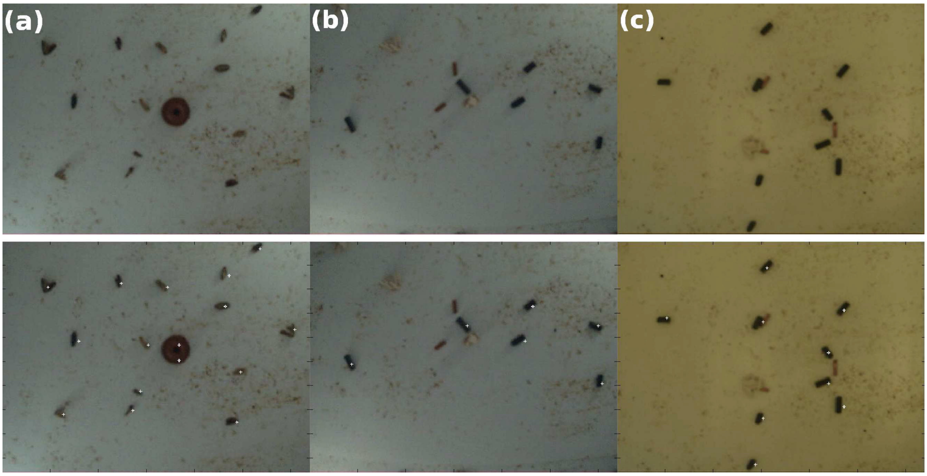

Fig. 5. Above, images acquired with the designed WimSN node. Below, same images with detected insects. (a) Image containing real insects. The round object is the pheromone lure. (b, c) Image of simulated insects. Images (a) and (b) were acquired with the trap self illumination, while image (c) was acquired with the trap in daylight.

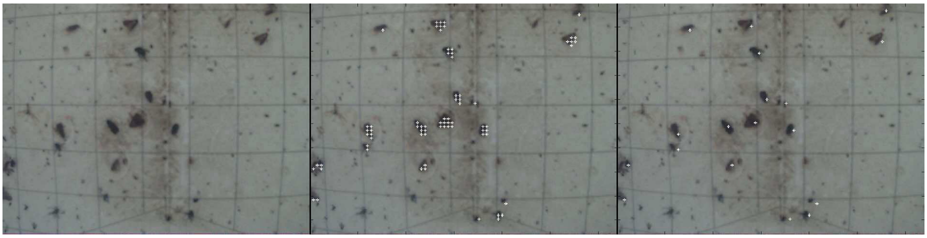

Fig. 6. (a) Image acquired with the designed WimSN and trap. (b) Position of the positive blocks detected with the block based classifier. (c) Position of the connected components detected with the optional labelling algorithm

3.4 Impact of Distributed Processing on the WSN

The impact of distributed processing in our application is twofold.

1. WSN were not designed to transmit large amounts of data. The transmission of images implies having to modify the existing network protocols to increase the performance of the system (e.g. enhance synchronization strategies over the radio channel). Having the possibility to transmit only the information of the detected insects in a couple of radio packets widely simplifies the network protocol selection as the problem turns into a classical WSN application, where few bytes are transmitted periodically. In this scenario, well established network protocols and applications already implemented in the real time operating system can be used to collect data in tree or mesh topology networks.

2. Transmitting only the information of the detected insects also enhances the battery lifetime of the nodes of the WSN. Sending JPEG images represents the 7.3 % of the consumption of battery power of the WimSN node. This could be acceptable in unicast single-hop transmissions but not for large covering WSNs with tree or mesh multi-hop topologies. For example, for a branch network with 7 hops, the node closest to the sink[2] shall transmit its own image but also receive and transmit 6 images of its' children nodes. This implies that the energy consumed by the radio activity would be increased considerably in this node degrading its energy independence.

   On the other hand, the implemented detection algorithm represents the 1.2 % of the consumption of battery power of the WimSN but retransmissions represent only 0.1 % of the energy consumption at the node. In this configuration, the estimated consumption of the battery charge is 2.62 mAh per day (acquiring and processing two images per day) which means an energy autonomy of over 30 months with typical AA batteries. Hence, with distributed processing, the energy autonomy of the nodes close to the sink is no longer an issue.

4 Concluding Remarks

Algorithms for image processing and pattern recognition in the compressed domain are useful tools for applications with low computational resources and strict energy consumption requirements. WSNs with imaging nodes are one of this kind of applications that can benefit from compressed domain techniques.

A wireless sensor node was designed and assembled for a WSN dedicated to distributed pest monitoring on fruit trees. The node implements a block based classifier in the compressed domain. In this case a simple LDA classifier was used but the experience can be easily extended to other supervised classifiers in the future. The preliminary results of the classification are promising. Although, further experimentation and a bigger image dataset is necessary to train/test the system, this step in our project has shown that the approach is valid and it enables: (i) battery lifetime enhancement, and (ii) network simplification.

The next step in our project is the deployment of some WimSN nodes in the fruit plantation in order to capture more images during the fruit growing season. In this first deployment, nodes will be working in a dual mode transmitting the images and also the local classification. Classification will be evaluated and the acquired images will allow to build an important database that will enable to incrementally retrain the classifier.

Our future work will include the evaluation of classification on the image database acquired at the plantation. That evaluation will tell us if this simple approach is sufficient or if other classifiers/algorithms are required at the nodes. Anyway, the framework for compressed domain classifiers/algorithms is already set and easy to extend.

[2] The sink node receives all the information from the network. This node is usually attached to a computer and does not have energy restrictions.

References

1. Caltech. Computational Vision Group - Archive. http://www.vision.caltech.edu/html-files/archive.html
2. The Contiki Community. Contiki - The Open Source OS for the Internet of Things.
3. de Queiroz, R.L.: Processing JPEG-compressed images and documents. IEEE Trans. Image Process. **7**(12), 1661–1672 (1998)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2012)
5. Maxfor Technology Inc., Maxfor Digital Brochure (2011)
6. Karlsson, J.: Image compression for wireless sensor networks. Master thessis in computing Science (2007)
7. Lee, S.: An efficient content-based image enhancement in the compressed domain using retinex theory. IEEE Trans. Circ. Syst. Video Technol. **17**(2), 199–213 (2007)
8. Linksprite. LinkSprite JPEG Color Camera Serial UART Interface (2012)
9. Lloret, J., Bosch, I., Sendra, S., Serrano, A.: A wireless sensor network for vineyard monitoring that uses image processing. Sensors **11**, 6165–6196 (2011)
10. López, J.A., Soto, F., Sánchez, P., Iborra, A., Suardiaz, J., Vera, J.A.: Development of a sensor node for precision horticulture. Sensors **9**(5), 3240–3255 (2009)
11. Mukhopadhyay, J.: Image and Video Processing in the Compressed Domain. CRC Press, Boca Raton (2011)
12. Nikolakopoulos, G., Kandris, D., Tzes, A.: Adaptive compression of slowly varying images transmitted over wireless sensor networks. Sensors **10**(8), 7170–7191 (2010)
13. Nilsback, M.-E., Zisserman, A.: A visual vocabulary for flower classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1447–1454 (2006)
14. Pierce, F.J., Elliott, T.V.: Regional and on-farm wireless sensor networks for agricultural systems in eastern washington. Comput. Electron. Agric. **61**(1), 32–43 (2008)
15. Shen, B., Sethi, I.K., Bhaskaran, V.: DCT domain alpha blending. In: 1998 Proceedings of International Conference on Image Processing, ICIP 98, vol. 1, pp. 857–861. IEEE (1998)
16. Tirelli, P., Borghese, N.A., Pedersini, F., Galassi, G., Oberti, R.: Automatic monitoring of pest insects traps by Zigbee-based wireless networking of image sensors. In: I2MTC-IEEE, pp. 1–5 (2011)
17. Gregory, K.: Wallace. the JPEG still picture compression standard. Commun. ACM **34**(4), 30–44 (1991)
18. Zhuang, L., Zhao, R., Yu, N., Liu, B.: SVD based linear filtering in DCT domain. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 2769–2772. IEEE (2010)