Detecting External Disturbances on the Camera Lens in Wireless Multimedia Sensor Networks

Cesare Alippi, *Fellow, IEEE*, Giacomo Boracchi, Romolo Camplani, and Manuel Roveri

***Abstract*—Assessing the quality of images acquired by nodes of a wireless multimedia sensor network (WMSN) is a critical issue, particularly in outdoor applications where external disturbances such as the presence of water, dust, snow, or tampering on the camera lens may seriously corrupt the acquired images. In this paper, we address the problem of determining when the lens of a microcamera mounted on a WMSN node is affected by an external disturbance that produces blurred or obfuscated images. We show that such a problem can be solved by considering change detection tests applied to a measure of the blur intensity. Interestingly, both measuring the blur and the change detection test do not require any assumption about the content of the observed scene, its nature, or the characteristics of the blur. The proposed methodology is thus flexible and effective for a large class of monitoring applications. In particular, it is attractive when nodes cannot continuously acquire images (e.g., because of energy limitations) and background subtraction methods are not feasible. Two solutions, with different hardware costs, have been designed, which can detect disturbances on the camera lens at either/both the node or/and the remote station level.**

***Index Terms*—Change detection tests, degradation detection, digital image analysis, digital image processing, wireless multimedia sensor networks (WMSNs).**

I. Introduction

IN RECENT years, wireless multimedia sensor networks (WMSNs) [1], [2] have gained an increasing interest in the wireless sensor network (WSN) community, e.g., see applications in [3]–[6]. Such networks differ from traditional WSNs for the presence of audio and visual sensors that provide photos, videos, and audio streams. WMSN applications range from tracking and surveillance to traffic control and environmental monitoring: not rarely, nodes are requested to operate in outdoor harsh environments. Noticeable examples of nodes provided with visual sensors are given in [7]–[10].

This paper addresses an often-forgotten problem that regularly arises in WMSN nodes designed to work in a real environment: the presence of external disturbances on the camera lens that affect the acquired image. Possible sources of disturbances are rain and sprinkle drops, humidity, dust, and tampering attacks, which introduce blur or dim effects in the acquired images. Fig. 1 shows some images taken with water drops insisting on the camera lens. As one would expect, there may be a significant loss in the image quality, which also depends on the type and entity of the disturbance. Since WMSNs are energy-constrained embedded systems, we should not waste energy in processing or, worse, sending images to the remote station when the blur intensity impairs its information content. Furthermore, when such a disturbance is detected in a node, different actions can be accomplished to tackle its presence. For instance, all units in a neighborhood of the disturbance-affected node can increase their field of view, and the node can activate different sensors (such as microphones and passive infrared/microwave motion detectors) by using intelligent control actions, e.g., see [11], or, in the worst case, request human intervention. In any case, any image-based algorithm in execution on the node requires a preliminary procedure to determine whether the acquired image is blurred/obfuscated or not to give a confidence to the available information and drive consequent actions. The same holds for algorithms acting at the cluster level. As a consequence, a mechanism for automatically diagnosing the status of the image acquisition system is requested in WMSN applications operating in outdoor environments. In particular, in monitoring applications that perform sporadic image acquisition, blur detection should not be based on image comparisons (e.g., comparison between the current image and an estimated background image) since the scene between two consecutive images can significantly change.

The detection of external disturbances insisting on the camera lens remains an open research issue. For instance, the method proposed in [12] aims at detecting tamper attacks that obscure the camera field of view. Nevertheless, the method has been designed for systems performing continuous acquisition, and it compares the acquired images with a learned background of the monitored scene. As previously discussed, this could be impractical in several WMNS applications.

In principle, any algorithm for automatic blur detection can be used to determine when disturbances insisting on the camera lens are corrupting the image acquisition system. The algorithms presented in [13] and [14] move in this direction by determining if an image is blurred or contains blurred areas and by classifying each image/area as out of focus or motion blurred. Unfortunately, these algorithms are based on computationally demanding feature extraction steps, which are hardly executable on low-performance WMSN nodes.

The effects of atmospheric elements such as rain, fog, and haze on acquired images have been studied under the

Manuscript received January 21, 2010; revised March 4, 2010; accepted March 5, 2010. Date of publication May 6, 2010; date of current version October 13, 2010. This work was supported in part by the INTERREG EU project M.I.A.R.I.A. (an adaptive hydrogeological monitoring supporting the alpine integrated risk plan) Italy–Switzerland action 2007–2013 and in part by the FIRB Project In.Sy.Eme (Integrated Systems for Emergencies) 2008–2010. The Associate Editor coordinating the review process for this paper was Dr. Emil Petriu.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy (e-mail: alippi@elet.polimi.it; boracchi@elet.polimi.it; camplani@elet.polimi.it; roveri@elet.polimi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2010.2047129

0018-9456/$26.00 © 2010 IEEE

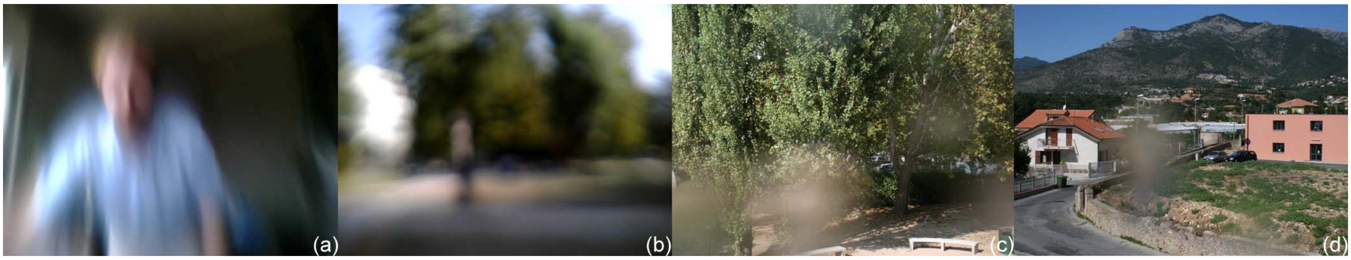

Fig. 1. Example of images acquired with some drops of water affecting the camera lens. Drops introduce blur and dim areas in the resulting images. In turn, they may affect the whole image as in (a) and (b) or only some areas of the image as in (c) and (d).

assumption that these elements are present between the camera and the observed scene, i.e., they are not insisting on the camera lens. In particular, [15] considers the case where water drops appear in the depth of field of the camera with the consequence that drops can be assumed as optical lenses that reflect and refract the light, producing a wide-angle view of the environment. An algorithm to detect and remove falling raindrops as time-varying fluctuations in video sequences is presented in [16]. Other ad hoc image enhancement algorithms have been proposed for compensating the effects of bad weather conditions on images or videos (e.g., [17] and [18] introduce methods to remove fog and haze from a single image). These algorithms consider the light scattering produced by small droplets of dust or mist in the atmosphere to perform effective contrast enhancement. However, images acquired with drops or other external disturbances insisting on the camera lens are significantly different from all the aforementioned cases, as they are significantly blurred and the suggested solutions are not meant for blur removal; rather, they perform contrast enhancement. The related literature suggests to jointly estimate and remove blur in a single blurred image (blind deblurring) to address the blur removal problem. Unfortunately, these solutions cannot be used to enhance the considered images, as they assume the blur to be spatially invariant [19], [20] in contrast with the spatially variant nature of the blur introduced by disturbances on the camera lens (particularly when the external disturbances do not uniformly cover the camera lens). In addition, blur removal algorithms assume the input image to be blurred and do not provide hints for deciding whether that is the case or not.

This paper proposes a novel method for monitoring the status of the image acquisition system to detect in advance a possible *structural* information loss due to perturbations insisting on the camera lens (e.g., drops, tampering, and dust). As such, we do not address the image enhancement issue but signal the presence of an external disturbance when it arises. To accomplish such a task in a time-variant context without assuming strong hypotheses about the observed scene and its dynamics, we require two independent steps: 1) measuring the blur intensity in the acquired images and 2) detecting the change in blur intensity, i.e., detecting the presence of the disturbance on the camera lens. Two solutions providing different detection capabilities, computational complexities, and power consumption are proposed here to meet the requirements of a typical WMSN scenario. The solutions act at either

1) the *node level*, where both the blur measure and the change detection test are executed on a WMSN node; or
2) the *network level*, with the blur measure computed at the node level and the outcoming estimate sent to a base station to undergo the change detection phase.

In other words, the two suggested philosophies use the same figure of merit to quantify the blur in the acquired images but implement different change detection tests based on the available hardware resources to detect a—possible—blur presence.

This paper extends the work presented in [21] as it introduces the node-level solution, provides the analysis of its algorithmic complexity, and strengthens the experimental section.

The structure of this paper is described as follows. Section II introduces the observation model, and Section III describes the blur change detection solutions in detail. The experimental campaign, which includes both real and synthetic testbeds, is finally presented in Section IV; experiments will be tailored to the drop case, but the methodology can easily be extended to cover similar types of external disturbances.

II. Observation Model

We have already seen in Section I that disturbing elements insisting on the camera lens are typically out of focus and induce blur or dim effects on the acquired image z. This phenomenon can be modeled as the result of a degradation operator $\mathcal{D}$ applied to the error-free and unknown image y (the *original* image), i.e.,

$$z = \mathcal{D}[y]. \tag{1}$$

The squared brackets are used to indicate the argument of an operator. Here, $\mathcal{D}$ takes into account blur and noise according to the widely accepted additive model [22], i.e.,

$$z(x) = \mathcal{D}[y](x) = \mathcal{B}[y](x) + \eta(x), \qquad x \in \mathcal{X} \tag{2}$$

where x indicates the pixel coordinates, $\mathcal{X}$ is the discrete image grid, $\mathcal{B}$ is the blurring operator, and η is the noise term. Typically, the blurring operator $\mathcal{B}$ is assumed to be linear (e.g., see [23]), leading to the final model, i.e.,

$$\mathcal{B}[y](x) = \int_{\mathcal{X}} y(x)h(x,s)ds. \tag{3}$$

$h(x,\cdot)$ represents the blur point spread function (PSF) at x, which is assumed to be a nonnegative function, as it performs local smoothing on y. The model described in (3) is very general and hosts different behaviors induced by the presence of drops/dust on the camera lens.

We then consider the general case where each sensor node of the network acquires a sequence of N observations $\{z_i\},\ i = 1,\ldots,N$, i.e.,

$$z_i(x) = \mathcal{B}_i[y_i](x) + \eta(x), \qquad i = 1,\ldots,N \tag{4}$$

with $\eta(x)$ being a stationary noise. The sequence of the original images $\{y_i\},\ i = 1,\ldots,N$, may significantly change in their content, depending on the monitored scene. As a consequence, a naive approach exploiting direct comparisons among two consecutive observations z_i and z_{i+1} may easily fail, being difficult to distinguish if different observations are due to different original images (i.e., $y_i \neq y_{i+1}$) or to a change in the blurring operator $(\mathcal{B}_i \neq \mathcal{B}_{i+1})$.

III. Detecting Changes in the Degradation Process

The proposed method requires analyzing the observations in $\{z_i\},\ i = 1,\ldots,N$, to determine a possible change in the degradation operator $\mathcal{D}$ (change associated with an external presence on the camera lens). Since the noise is assumed to be stationary, a structural change in the image acquisition system, such as the presence of drops/dust, reflects a change in the blur operator $\mathcal{B}$: detecting a change in the blur operator implies detection of a structural change affecting the image acquisition system.

A. Measuring the Blur

As one could imagine, it is hard to devise an index or figure of merit able to measure the actual blur of an image given a generic blurring operator $\mathcal{B}$. What the related literature suggests instead is to indirectly measure the blur, by relying on some details or frequency information present in the observed image z_i, as done when identifying the optimal camera parameters (e.g., focal length, aperture, and exposure) before taking a shot [24]–[27]. The underlying philosophy onto which these measures rely reflects the intuitive idea that the blur suppresses the high-frequency components of an image by local smoothing. Based on such observation, most of blur measures are actually estimates of the energy content of the image in high frequency. In the same direction, here, we consider the blur measure

$$m_i = M[z_i] = \int_{\mathcal{X}} \|\nabla z_i(x)\|_1 \, dx \tag{5}$$

where $\|\cdot\|_1$ refers to the $\mathcal{L}^1$ norm. In the discrete domain, the image derivatives are computed by means of differentiating filters (here, the Sobel ones [22]). Note that M is indirectly a measure of the total energy of the image details; as such, it is particularly sensitive to the image content (M is low when computed on blur-free images having few details, as well as in images heavily corrupted by blur). However, this measure can also be used on partially blurred images, and it guarantees a very low computational complexity.

Let us discuss how the noise term η influences a sequence of blur measures $\{m_i\},\ i = 1,\ldots,N$. Typically, noisy images have larger blur measures compared to the corresponding noise-free images: the larger the σ is, the larger the blur measure is. In addition, since the noise is stationary, it does not introduce anomalies in the sequence of blur measures. We expect that when the blur measure is dominated by the noise, detecting a decrease in the blur measure according to $\mathcal{B}$ becomes much more challenging.

B. Detecting the Change

Change detection tests are statistical techniques that, by monitoring the behavior of a process over time, detect a possible change in its behavior. In the considered case, the process under monitoring is the degradation operator $\mathcal{D}$ that corrupts the sequence of unknown original images $\{y_i\},\ i = 1,\ldots,N$, and gives the observation sequence $\{z_i\},\ i = 1,\ldots,N$; see [28] for another application of change detection tests in WSNs. Among the large range of solutions (*data driven*, *analytical*, or *knowledge based*) present in the literature to assess a change in processes [29], we focus on data-driven techniques since they do not require any *a priori* information about the process under investigation. The most common data-driven techniques for change detection generally require a design-time configuration phase to configure the test parameters either by exploiting *a priori* information or through a trial-and-error approach [29]–[31]. In our problem, we suggest to use two adaptive self-configuring statistical tests, i.e., the adaptive cumulative sum (CUSUM) and the computational intelligence-based CUSUM (CI-CUSUM) [32], for their effectiveness in detecting abrupt changes and smooth drifts. Both tests are general, do not require any information about the process under monitoring, and exploit an initial sequence $\{m_i\},\ i = 1,\ldots,T$, of blur measures computed from T external disturbance-free images for the automatic configuration of their parameters. Such a sequence allows the tests for both estimating the probability density function (pdf) of m_i in the absence of external disturbance (i.e., the null hypothesis Θ^0) and defining alternative hypotheses Θ^1s representing the "not being in Θ^0" to address any type of nonstationary change (the alternative hypotheses are automatically defined during the training phase).

To guarantee an accurate estimate of test parameters, the authors suggest to consider a reasonable large training sequence, e.g., $T > 400$. Both tests work on subsequences of blur measures (in our experiments, we considered subsequences of 20 blur measures) and estimate the transition from Θ^0 to Θ^1 by measuring the log-likelihoods between the pdf in the absence of drop/dust and the pdf's of all the alternative hypotheses at subsequence τ (one at a time), i.e.,

$$r(\tau) = ln \frac{N_{\Theta^1}(\phi_\tau)}{N_{\Theta^0}(\phi_\tau)} \tag{6}$$

where ϕ_τ is the average value of the blur measures of the τth subsequence, and N_Θ is a multivariate Gaussian distribution parameterized in Θ.

The log-likelihood ratio has an important property: a change in the pdf of the process under monitoring can be detected by analyzing the sign of the log-likelihood ratios. Both tests are able to detect the presence of drops in the images by sequentially checking whether the ϕ_τs have been generated according to a pdf associated with Θ^0 or one of the alternative hypotheses. When one of the cumulative sums of the $r(\tau)$ overcomes an automatically defined threshold, the test detects a change in the statistical behavior of the ϕ_τs (a detailed description of both tests is given in [32]). Since the log-likelihood ratio compares couples of pdf's, both tests have a number of running log-likelihood ratios that are equal to the number of alternative hypotheses defined by the test (i.e., each log-likelihood ratio compares the null hypothesis with one of the alternative hypotheses). The main difference between the adaptive CUSUM and the CI-CUSUM test consists of the set of considered features. More specifically, the adaptive CUSUM assesses changes in the mean and the variance of the process under monitoring, while the CI-CUSUM exploits a larger set of features (i.e., not only the mean and variance but also features derived from the pdf and the cumulative density function, as well as features inspired by change detection tests present in the literature), and it is more accurate at the expense of a significant increase in computational complexity. Obviously, a higher number of alternative hypotheses guarantee a more effective exploration of the hypotheses space and, hence, a larger change detection ability. The selection between the adaptive CUSUM and the CI-CUSUM test is thus strictly related to the available computational resources and the desired detection accuracy. We suggest considering the adaptive CUSUM test for the node solution (i.e., the change detection test is directly executed on nodes), while the CI-CUSUM test is the suitable choice for the network solution at either the cluster heads or the remote control station.

IV. Experiments

The proposed methods have been tested on two applications. The first benchmark refers to a sequence of synthetically generated observations (Application D1); the second refers to a real sequence of images (Application D2). In both cases, our goal is to detect the presence of water drops. Four figures of merit have been suggested to assess the performance of the proposed solutions.

- DL Detection latency. It represents the number of images required to detect a change in the blurring process after the drop arrival.
- FP False positives. It measures the number of blur changes erroneously detected by the test.
- FN False negatives. It measures the number of blur changes not detected by the test.
- ET Execution time (in seconds) of the adaptive CUSUM and the CI-CUSUM test estimated with Matlab.[1]

Execution times have separately been evaluated for the training phase needed to configure the test parameters (ET training) and for the operational phase (ET operational). The configuration set accounts for 500 blur measures computed from blur-free observations; the validation set accounts for 1500 ones.

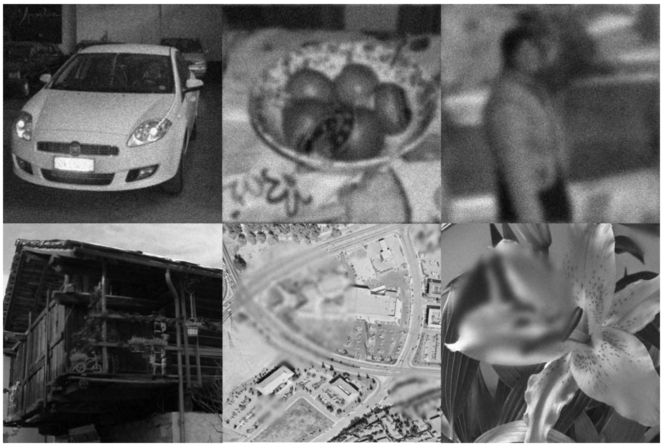

Fig. 2. Example of synthetically generated observations. First row: the blur affects the whole image, $\sigma = 0.08$, and $\nu = 1, 4, 8$, respectively. Second row: the blur affects only some part of the image, $\sigma = 0.02$, and $\nu = 1, 4, 8$, respectively.

A. Application D1

A set of sequences of observations has been generated according to (4). Each sequence contains 2000 observations obtained from 75 gray-scale 512 × 512 pixel original images, scaled in the [0, 1] value interval. In each sequence, the first 1000 observations are blur-free, i.e., $\mathcal{B}_i = \mathcal{I},\ i = 1, \ldots, 1000$, where $\mathcal{I}$ stands for the identity operator; the others have been affected by a blurring operator $\mathcal{B}_i,\ i = 1001, \ldots, 2000$, having the PSFs of (3) defined as

$$h(x, s) = \begin{cases} \delta(x - s), & x \in \mathcal{X}_0 \\ g(x - s), & x \in \mathcal{X}_1 \end{cases}, \qquad \mathcal{X}_0 \cup \mathcal{X}_1 = \mathcal{X} \qquad (7)$$

where $\mathcal{X}_0 \cap \mathcal{X}_1 = \emptyset$, δ is the Dirac's delta function, and g a Gaussian kernel of standard deviation ν. Sets $\mathcal{X}_0$ and $\mathcal{X}_1$ denote blur-free and blurred areas, respectively. Therefore, the considered blur may affect only some parts of the original image, and within the blurred areas, the blurring operator is space invariant. We considered two different cases: in the first, the blurring operator affects the whole image (i.e., $\mathcal{X}_1 = \mathcal{X}$, FULL blur); in the second, the blurring operator affects only some parts of the image (i.e., $\mathcal{X}_1 \subset \mathcal{X}$, PART blur). In this latter case, the sets $\mathcal{X}_0$ and $\mathcal{X}_1$ are the same for all sequences. In each sequence, the noise term is Gaussian $\eta \sim N(0, \sigma^2)$ added to both the blur-free and blurred images.

For both the FULL and the PART blur, we considered eight values of the standard deviation of the Gaussian kernel g of (7) $\nu = 1, 2, \ldots, 8$ and four values of σ; the standard deviation of η ranges from 0.02 to 0.08 (with a step of 0.02). For each parameter pair (ν, σ), we generated 100 different sequences to compute the figures of merit for the two solutions. Fig. 2 shows that observations generated with such parameters, at least for high values of ν, are very similar to images acquired with a drop on the camera lens, such as those in Figs. 1 and 5.

Tables I and II show the change detection ability of the network and the node-level solutions, respectively. On one hand, the network-level solution guarantees less FPs than the node-level one, thanks to the superior detection ability of the

[1] Reference platform: Intel Core 2 Duo 2.53-GHz CPU, no parallel threads.

TABLE I
NETWORK-LEVEL SOLUTION: FP AND FN EVALUATED ON DATA SYNTHETICALLY GENERATED IN APPLICATION D1

			ν							
Blur	σ	Detection	1	2	3	4	5	6	7	8
FULL	0.02	*FP*(%)	10	18	18	7	10	10	14	7
		FN(%)	1	0	1	0	4	2	0	0
FULL	0.04	*FP*(%)	14	13	9	9	12	16	11	13
		FN(%)	6	0	0	1	0	3	0	1
FULL	0.06	*FP*(%)	8	15	9	9	9	6	9	17
		FN(%)	2	2	3	1	3	2	0	0
FULL	0.08	*FP*(%)	9	11	4	12	4	13	10	8
		FN(%)	6	0	1	1	1	0	1	0
PART	0.02	*FP*(%)	11	8	6	11	8	15	15	13
		FN(%)	34	17	13	11	11	3	6	5
PART	0.04	*FP*(%)	12	11	10	7	4	11	10	9
		FN(%)	37	16	10	12	7	7	6	5
PART	0.06	*FP*(%)	12	11	19	12	10	8	8	14
		FN(%)	38	19	4	12	7	8	8	5
PART	0.08	FP(%)	5	12	8	12	13	8	13	16
		FN(%)	36	20	8	11	11	7	7	4

TABLE II
NODE-LEVEL SOLUTION: FP AND FN EVALUATED ON DATA SYNTHETICALLY GENERATED IN APPLICATION D1

			ν							
Blur	σ	Detection	1	2	3	4	5	6	7	8
FULL	0.02	*FP*(%)	21	17	18	10	19	13	18	10
		FN(%)	2	0	0	0	0	0	0	0
FULL	0.04	*FP*(%)	15	14	14	17	8	20	15	15
		FN(%)	3	0	0	0	0	0	0	0
FULL	0.06	*FP*(%)	16	19	16	10	17	15	17	13
		FN(%)	3	0	0	0	0	0	0	0
FULL	0.08	*FP*(%)	7	18	9	13	12	17	15	18
		FN(%)	2	0	0	0	0	0	0	0
PART	0.02	*FP*(%)	16	11	14	20	17	19	12	15
		FN(%)	69	41	34	24	25	27	23	19
PART	0.04	*FP*(%)	16	11	19	17	19	16	19	16
		FN(%)	65	51	38	32	27	22	23	23
PART	0.06	*FP*(%)	18	15	16	20	15	15	19	14
		FN(%)	71	50	33	25	27	31	23	21
PART	0.08	*FP*(%)	16	16	10	18	14	14	10	24
		FN(%)	70	42	33	33	32	24	24	21

CI-CUSUM test. This is particularly evident in the "PART" case at low values of ν, where the node-level solution is not able to detect the presence of blur (FNs in the last rows of Table II). On the other hand, the node-level solution guarantees very low detection delays (the test is very quick in detecting changes) and a reduced computational complexity. Figs. 3 and 4 show the detection latency on the considered data set for both the node- and network-level solutions. In both cases, the amount of images required to detect the presence of blur decreases as ν increases. We observe that the FP values are independent from the values of ν (the standard deviation of the Gaussian PSF), while FNs decrease as ν increases: the higher the blur, the easier its detection. In particular, when the blur corrupts only part of the image (the "PART" case), the values of FN are higher than when the blur corrupts the whole image (the "FULL" case). Moreover, at low values of ν, the node-level solution is not able to reliably detect the presence of drops (see the FN in the last rows of Table II). Both the network- and node-level solutions are able to cope with the considered noise levels: the values of FPs, FNs, and DLs show that the detection performance is not altered. We comment that it is extremely important to provide a reduced number of FPs, as these false alarms are sent over the network and they may result in a waste of resources. Policies at the unit and cluster levels could be implemented to reduce FPs by exploiting information coming from neighboring units. The execution time averaged over the algorithm runs is given in Table III and shows that the node solution is considerably faster than the network one during both the training and operational phases.

B. Application D2

The second application refers to a set of 25 uncompressed video sequences acquired in five different dynamic scenarios (three outdoor and two indoor). Each sequence is composed of 2000 frames (320 × 240 pixels) recorded by an integrated webcam of a laptop computer. Each frame has been converted into gray scale by averaging the *red–green–blue* values of each pixel. The first 1000 frames are drop-free, while the next 1000 have been acquired with some water drops on the camera lens. Fig. 5 shows, along rows, six frames taken from a video sequence (one per each scenario). Similar to Application D1, the training phase of both tests exploits the first 500 drop-free images of each sequence. The figures of merit have been evaluated by averaging the results of the 25 video sequences; the comparison between the performance of the two solutions is presented in Table IV.

The experimental results on this data set are in line with those of Application D1: the network-level solution guarantees lower FPs than the node solution, which, on the contrary, provides a prompter detection ability (lower values of DL) and reduced execution time. When processing these video sequences, the FPs are determined by accidental and unpredictable occluding objects that do not appear in the training set, as, for example, the shadowed face and the transparent plastic bag appearing in the sequence illustrated in the third and fourth rows of Fig. 5, respectively. In fact, different from the synthetically generated sequences of Application D1, here, the training set might not be fully representative of all the original images y_i that the WMSN node has to face in working conditions. Occluding objects may also induce FN when they are shown in the training set, as the decay in the blur measures due to the drop arrival could. Motion blur, which frequently occurs in images of dynamic scenes acquired in low-light conditions, may also cause both FPs and FNs since it causes loss of details in the observations and subsequent decay of the blur measures. This challenging problem could be (at least) partially addressed by integrating lighting information and exposure times in the change detection tests. The detection performance can be improved by considering longer training sequences and, when possible, by updating the training set with user-supervised and disturbance-free images acquired during the test execution.

Fig. 6 shows the blur measures m_i associated with the sequence including the frames depicted in the second row

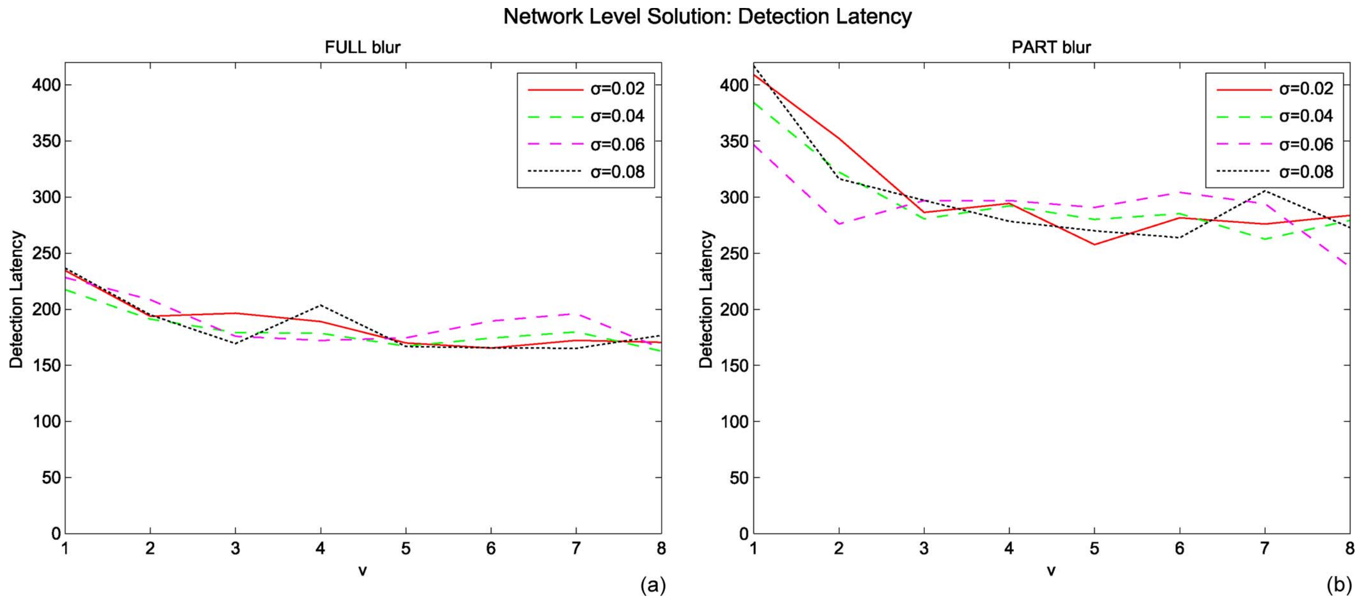

Fig. 3. DL of the network solution as a function of ν (the standard deviation of the Gaussian PSF), computed for different values of σ (the noise standard deviation). (a) Plot of the DL when the blur affects the whole image. (b) Plot of the DL when the blur affects only some areas.

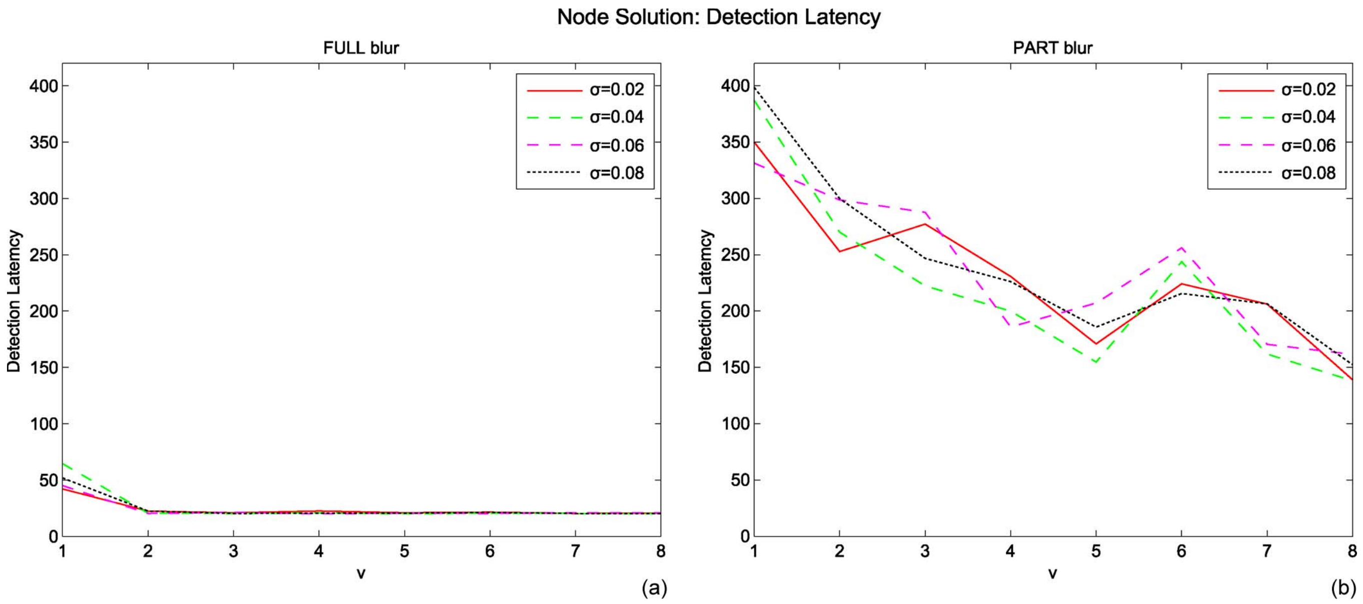

Fig. 4. DL of the node solution as a function of ν (the standard deviation of the Gaussian PSF), computed for different values of σ (the noise standard deviation). (a) Plot of the DL when the blur affects the whole image. (b) Plot of the DL when the blur affects only some areas.

TABLE III
APPLICATION D1: ET AVERAGED OVER THE $2 \times 8 \times 4 \times 100$ ALGORITHM RUNS FOR BOTH THE NETWORK-LEVEL AND NODE-LEVEL SOLUTIONS

solution	ET training (s)	ET operational (s)
network	0.125 s	0.461 s
node	0.010 s	0.003 s

of Fig. 5: in this case, both solutions detect the drop arrival within the subsequence ending at frame 1180 ($DL = 180$). Fig. 7 shows the blur measures m_i computed from the sequence illustrated in the third row of Fig. 5: in this case, the network solution detects the drop arrival within the subsequence ending at frame 1240 ($DL = 240$), while the node solution bears a false positive at frame 880, because of a sudden decay in the blur measures.

C. Computational Complexity of the Node-Level Solution

As expected, the execution times reported in Tables III and IV show that the adaptive CUSUM has a significantly lower computational complexity than the CI-CUSUM in both the training and operational phases. In fact, the adaptive CUSUM assesses changes by solely inspecting variations in the mean and variance of ϕ_τs, while the CI-CUSUM considers a larger set of features to improve the detection ability.

We present a detailed analysis of the computational complexity on the adaptive CUSUM to justify its use at the node level.

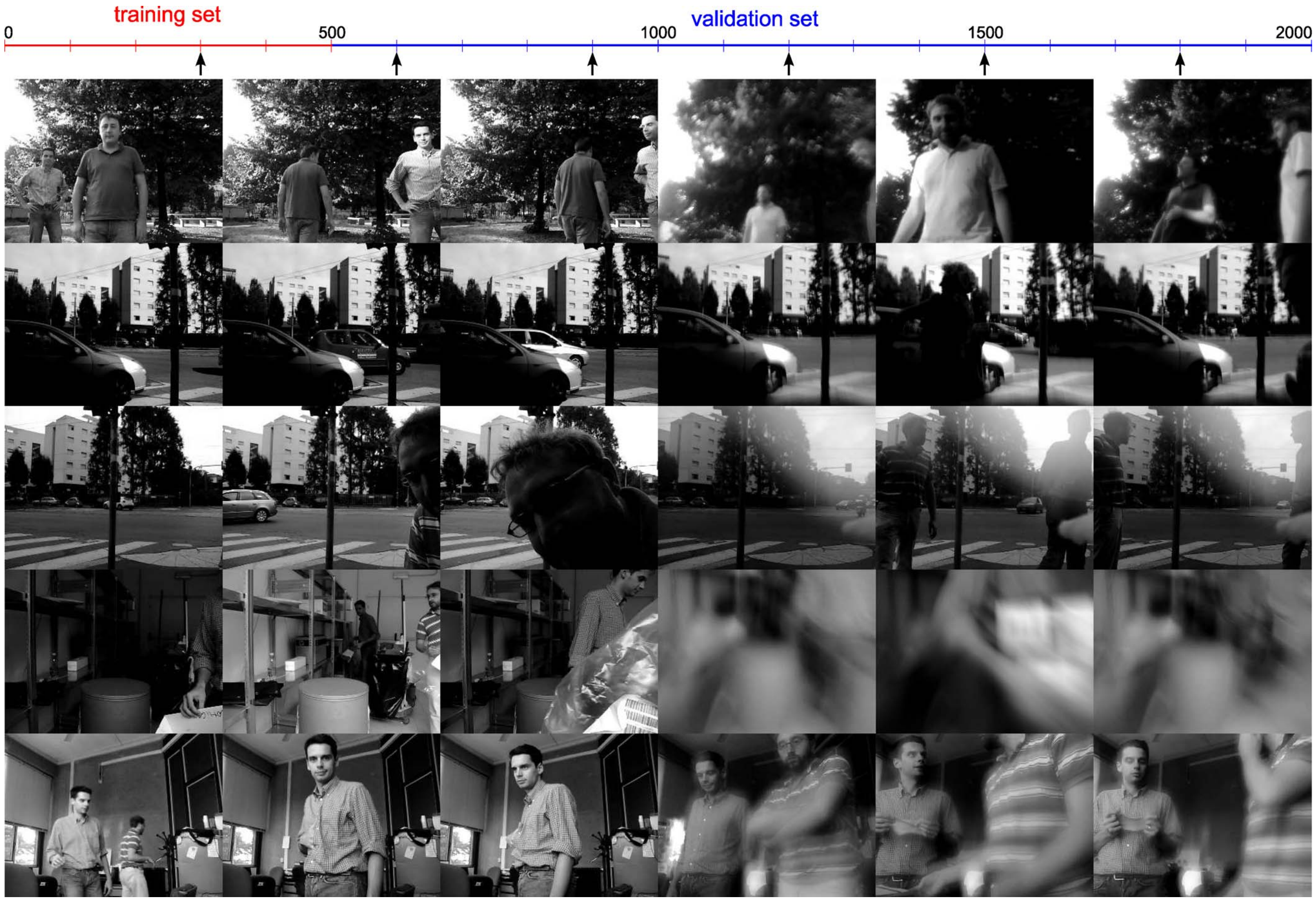

Fig. 5. Example of observations composing the video sequences. Each row shows six frames taken from an acquired video sequence. In all sequences the drops appear at frame 1001.

TABLE IV
APPLICATION D2: DETECTION PERFORMANCE EVALUATED FOR BOTH THE NETWORK-LEVEL AND NODE-LEVEL SOLUTIONS. FP AND FN HAVE BEEN COMPUTED OVER 25 VIDEO SEQUENCES, WHILE THE VALUES OF DL AND ET HAVE BEEN AVERAGED OVER THE 25 RUNS

solution	FP(%)	FN(%)	DL	ET training (sec)	ET operational (s)
network	16	4	181.0	0.127 s	0.461 s
node	24	4	44.667	0.009 s	0.003 s

The evaluation of the blur measure by (5) is discretized and implemented with $(2(s-1)+4)$ integer operations per pixel, with s being the number of nonzero coefficients of the convolutional filter used for computing image derivatives. In our Matlab implementation, $s=6$ (requiring 14 integer operations per pixel); s can be reduced to 2 when a 1-D filter is used (six integer operations per pixel).

The operational phase of the adaptive CUSUM, which constitutes the computational load in working conditions requires, for each subsequence of 20 blur measures:

1) computing the mean of the 20 blur measures of the frame (20 floating-point operations);
2) computing two logarithms (2×35 floating-point operations);
3) evaluating a 1-D standard Gaussian function in three points (3×28 floating-point operations).

The number of floating-point operations has been estimated with Matlab. It follows that the execution of the adaptive CUSUM test requires 175 floating-point operations per each subsequence of 20 blur measures. Such a reduced and computationally light sequence of operations can realistically be executed on a WMSN node having limited hardware and energy resources such as the [7]–[10] node platforms.

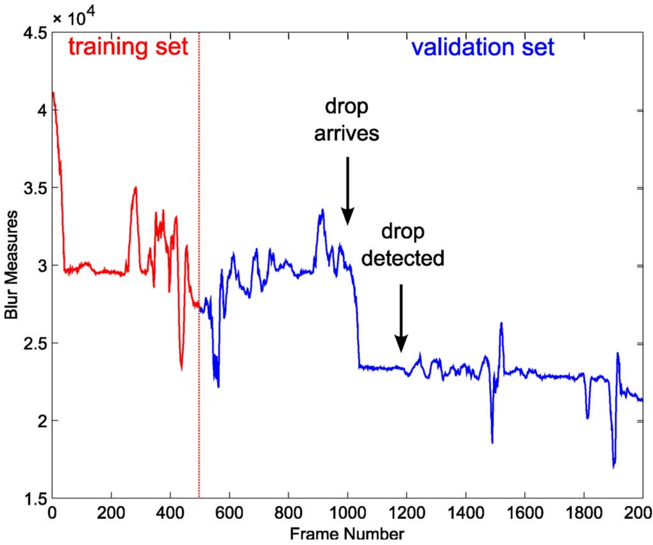

Fig. 6. Blur measures (5) computed in the sequence shown in the second row of Fig. 5.

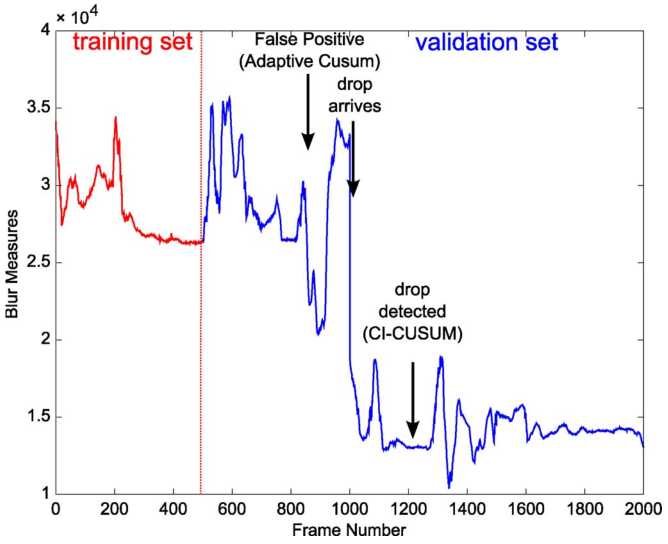

Fig. 7. Blur measures (5) computed in the sequence shown in the third row of Fig. 5.

V. Conclusions

This paper has presented two different solutions for detecting the presence of external disturbances on the camera lens in WMSN nodes. Such aspect is particularly relevant in the WMSN community since nodes quite often operate outdoors in harsh environments and it is important to continuously assess the status of the image acquisition system. The proposed solutions have combined a simple and easy-to-compute blur measure and a change detection test and have been proven effective on both synthetically generated images and video sequences acquired from a webcam.

The analysis of the computational complexity has shown that the node-level solution can be implemented on WMSN nodes deployed in critical environments. Ongoing work regards the implementation of the node-level solution on an STMicroelectronics prototype board, which represents the current state of the art in low-power smart cameras [10]. This board is equipped with the ST-VS6724 2-megapixel camera [33] and the ST STR912FA microcontroller [34], running at 96 MHz with 96-Kb SRAM. The board is able to process images in real time, delivering video streams at 30 fps. Furthermore, we are investigating strategies at the cluster level to improve the detection performance and reduce the number of FPs by exploiting local knowledge; such clusters can be generated as in [35].

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[2] W. Shaw, I. Lee, and X. Fan, “Wireless multimedia sensor networks,” in *Guide to Wireless Sensor Networks, Computer Communications and Networks*. New York: Springer-Verlag, 2009, pp. 561–582.

[3] Y. Kim, R. Evans, and W. Iversen, “Remote sensing and control of an irrigation system using a distributed wireless sensor network,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 7, pp. 1379–1387, Jul. 2008.

[4] J. Lin, W. Xiao, F. Lewis, and L. Xie, “Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1886–1896, Jun. 2009.

[5] M. Bertocco, G. Gamba, A. Sona, and S. Vitturi, “Experimental characterization of wireless sensor networks for industrial applications,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1537–1546, Aug. 2008.

[6] A. Carullo, S. Corbellini, M. Parvis, and A. Vallan, “A wireless sensor network for cold-chain monitoring,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1405–1411, May 2009.

[7] Jennic Ltd: Jn5148 Wireless Modules Documentation. [Online]. Available: http://www.jennic.com/products/modules/jn5148modules

[8] Sun labs: Sun spot, [Online]. Available: http://www.sunspotworld.com/index.html

[9] L. Ferrigno, A. Pietrosanto, and V. Paciello, “Low-cost visual sensor node for bluetooth-based measurement networks,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 2, pp. 521–527, Apr. 2006.

[10] M. Magno, D. Brunelli, L. Thiele, and L. Benini, “Adaptive power control for solar harvesting multimodal wireless smart camera,” in *Proc. 3rd ACM/IEEE ICDSC*, Sep. 2–30, 2009, pp. 1–7.

[11] R.-E. Precup, S. Preitl, E. M. Petriu, J. K. Tar, M. L. Tomescu, and C. Pozna, “Generic two-degree-of-freedom linear and fuzzy controllers for integral processes,” *J. Franklin Inst.*, vol. 346, no. 10, pp. 980–1003, Dec. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6V04-4X01PFH-1/2/305fb2150ae03114c6ff76e9ea15e6e8

[12] A. Aksay, A. Temizel, and A. E. Cetin, “Camera tamper detection using wavelet analysis for video surveillance,” in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, 2007, pp. 558–562.

[13] R. M. Chong and T. Tanaka, “Image extrema analysis and blur detection with identification,” in *Proc. Int. IEEE Conf. Signal-Image Technol. Internet-Based Syst.*, 2008, pp. 320–326.

[14] R. Liu, Z. Li, and J. Jia, “Image partial blur detection and classification,” in *Proc. IEEE CVPR*, 2008, pp. 1–8.

[15] K. Garg, Appearance of a raindrop, Comput. Sci. Dept., Columbia Univ., New York. [Online]. Available: http://www1.cs.columbia.edu/CAVE/publications/pdfs/GargTR04.pdf

[16] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 1, pp. 528–535.

[17] R. T. Tan, “Visibility in bad weather from a single image,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[18] R. Fattal, “Single image dehazing,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, Aug. 2008.

[19] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006.

[20] Y. Yitzhaky, R. Milberg, S. Yohaev, and N. S. Kopeika, “Comparison of direct blind deconvolution methods for motion-blurred images,” *Appl. Opt.*, vol. 38, no. 20, pp. 4325–4332, Jul. 1999.

[21] C. Alippi, G. Boracchi, and M. Roveri, “Detecting drops on lens in wireless multimedia sensor network nodes,” in *Proc. IEEE Int. Workshop Robot. Sens. Environ.*, Lecco, Italy, Nov. 2009, pp. 128–133.

[22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA: Addison-Wesley, 2001.

[23] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, PA: Inst. Phys. Publishing, 1998.

[24] M. Subbarao, T. Choi, and A. Nikzad, “Focusing techniques,” *J. Opt. Eng.*, vol. 32, no. 11, pp. 2824–2836, Nov. 1993.

[25] S. K. Nayar and Y. Nakagawa, “Shape from focus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 824–831, Aug. 1994.

[26] J. Kautsky, J. Flusser, B. Zitová, and S. Simberová, “A new wavelet-based measure of image focus,” *Pattern Recognit. Lett.*, vol. 23, no. 14, pp. 1785–1794, Dec. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/B6V15-45X0BW6-4/2/ca922e4f295957d8e12762a64a6105c7

[27] S.-Y. Lee, Y. Kumar, J.-M. Cho, S.-W. Lee, and S.-W. Kim, “Enhanced autofocus algorithm using robust focus measure and fuzzy reasoning,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1237–1246, Sep. 2008.

[28] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, “An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 335–344, Feb. 2010.

[29] L. H. Chiang, E. L. Russel, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. New York: Springer-Verlag, 2001.

[30] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[31] J. Zhang, E. B. Martin, and A. J. Morris, “Process performance monitoring using multivariate statistical process control,” *Proc. Inst. Elect. Eng.—Control Theory Appl.*, vol. 143, no. 2, pp. 132–144, Mar. 1996.

[32] C. Alippi and M. Roveri, “Just-in-time adaptive classifiers—Part I: Detecting nonstationary changes,” *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1145–1153, Jul. 2008.

[33] STMicroelectronics: St-vs6724 Camera Documentation. [Online]. Available: http://www.st.com/stonline/products/families/imaging/camera/vs6724.htm

[34] STMicroelectronics: Str91fa reference manual. [Online]. Available: http://www.st.com/stonline/products/literature/ds/13495.pdf

[35] H. He, Z. Zhu, and E. Mäkinen, “A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks,” *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 973–982, Jun. 2009.

Cesare Alippi (SM’94–F’06) received the Dr.Ing. degree (*summa cum laude*) in electronic engineering and the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 1990 and 1995, respectively.

He has completed research work in computer sciences at the University College, London, U.K., and the Massachusetts Institute of Technology, Cambridge. He is currently a Full Professor of information processing systems with the Dipartimento di Elettronica e Informazione, Politecnico di Milano. He is the author of more than 120 technical papers published in international journals and conference proceedings. His research interests include application-level analysis and synthesis methodologies for embedded systems, computational intelligence, and wireless sensor networks.

Dr. Alippi is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Giacomo Boracchi received the M.S. degree in mathematics from the Universitá Statale degli Studi di Milano, Milan, Italy, in 2004 and the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 2008.

He was Researcher with the Tampere International Center for Signal Processing, Tampere, Finland, in 2004. He is currently a Postdoctoral Researcher with the Dipartimento di Elettronica e Informazione, Politecnico di Milano. His research interests include mathematical and statistical methods for signal and image processing, and computational intelligence.

Romolo Camplani received the Dr.Ing. degree in electronic engineering and the Ph.D. degree in electronic and computer science from the Università degli Studi di Cagliari, Cagliari, Italy, in 2003 and 2008, respectively, and the M.Eng. degree in embedded system design from the Università della Svizzera Italiana, Lugano, Switzerland.

He is currently a Postdoctoral Researcher with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy. His research interests include wireless sensors networks, adaptive routing algorithms, and power-aware embedded applications.

Manuel Roveri received the Dr.Eng. degree in computer science engineering from the Politecnico di Milano, Milan, Italy, in June 2003, the M.S. degree in computer science from the University of Illinois at Chicago in December 2003, and the Ph.D. degree in computer engineering from the Politecnico di Milano in May 2007.

He is currently an Assistant Professor with the Dipartimento di Elettronica e Informazione, Politecnico di Milano. His research interests include computational intelligence, adaptive algorithms, and wireless sensor networks.