Data Reduction Using Change Coding for Remote Applications of Wireless Visual Sensor Networks

KHURSHEED AURANGZEB[1,2], **MUSAED ALHUSSEIN**[1], **AND MATTIAS O'NILS**[3]

[1]Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[2]Department of Electrical Engineering, COMSATS Institute of Information Technology Attock, Attock 43600, Pakistan
[3]Electronics Design Division, Mid Sweden University, 85170 Sundsvall, Sweden

Corresponding author: Khursheed Aurangzeb (kaurangzeb@ksu.edu.sa)

ABSTRACT The data reduction capability of image compression schemes is limited by the underlying compression technique. For applications with minor changes between consecutive frames, change coding can be used to further reduce the data. We explored the efficiency of change coding for data reduction in a wireless visual sensor network (WVSN). This paper presents an analysis of the compression efficiency of change coding for a variety of changes, such as different shapes, sizes, and locations of white objects in adjacent sets of frames. Compressing change frame provides a better performance compared with compressing the original frames for up to 95% changes in the number of objects in adjacent frames. Due to illumination noise, the size of the objects increases at its boundaries, which negatively affects the performance of change coding. We experimentally proved that the negative impact of illumination noise could be reduced by applying morphology on the change frame. Communication energy consumption of the VSN is dependent on the data that are transmitted to the server. Our results show that the communication energy consumption of the VSN can be reduced by 27%, 29%, and 46% by applying change coding in combination with JBIG2, Group4, and Gzip_pack, respectively. The findings presented in this paper will aid researchers in enhancing the compression potential of image coding schemes in the energy-constrained applications of WVSNs.

INDEX TERMS Change coding, communication energy consumption, embedded systems, image compression, wireless visual sensor network.

I. INTRODUCTION

Extensive research has been performed on image compression and wireless communications. However, the remote applications of Wireless Visual Sensor Networks (WVSNs) for large area monitoring have created new challenges due to the strict limitations on the storage, processing bandwidth, and power consumption in the Visual Sensor Nodes (VSNs). Many researchers have focused on various implementation strategies for executing vision processing tasks locally or centrally (Fig. 1). For example, in [1], images were captured and compressed in the VSN, and then transmitted to the server. In this case, communication energy consumption is higher than computational energy consumption because only image capturing and compression were executed at the VSN.

The researchers in [2] and [3] proposed executing all image processing tasks at the VSN and transmitting only the labeled features to the server. They reduced communication energy consumption, but computational energy consumption increased due to performing all tasks in the VSN. In [2], a distributed vision processing system was implemented for human pose interpretation. They accomplished real-time monitoring by employing distributed processing. By performing onboard processing, they extracted critical joints from the subjects in real time. The results from multiple cameras were transmitted to the server where the human pose was reconstructed.

SensEye [3] is a heterogeneous multi-tier network, in which the focus has been on low power, low latency detection, and low latency wakeup. SensEye can perform image processing tasks such as object detection, recognition, and tracking.

Both onboard processing and communication with a server consume a substantial portion of the total energy budget of each node. Communicating raw data to the server leads to reduced processing energy consumption, but has the disadvantage of larger communication energy consumption.
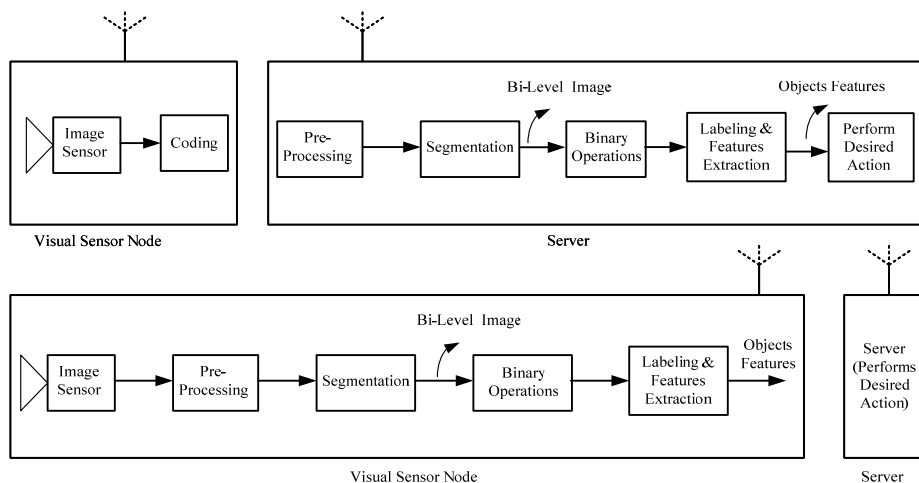
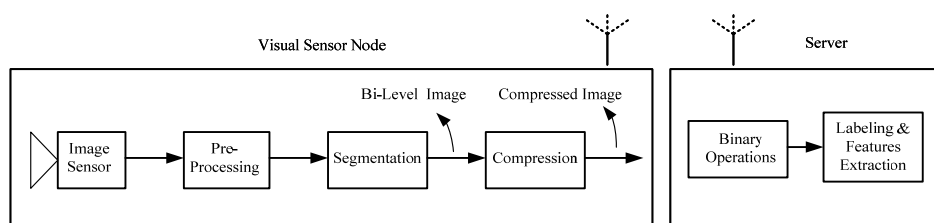**FIGURE 1.** The two extremes of image processing tasks in WVSNs.

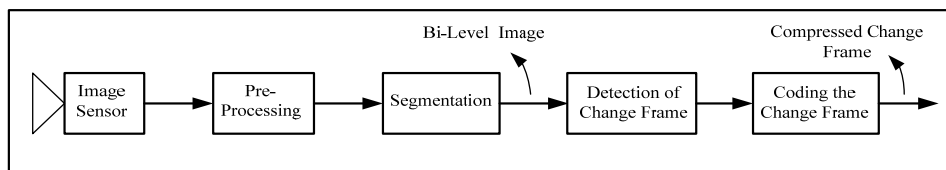**FIGURE 2.** Energy efficient architecture for WVSNs.

FIGURE 3. Sequence of operations for change coding.

Performing all processing locally and communicating the final object features to the server leads to reduced communication energy consumption, but has the drawback of larger computational energy consumption.

We previously determined in [4] and [5] that selecting an appropriate Intelligence Partitioning (IP) strategy between the VSN and the server will help in reducing the overall energy consumption of the VSN. In [6], we proposed a general architecture for some applications of WVSNs in which a binary image is compressed following pre-processing and segmentation (this architecture is shown in Fig. 2.) Based on this architecture, we explored the compression efficiency of well-known binary image compression methods in [7]. We concluded that the compression efficiency of JBIG2 [8], CCITT Group 4 [9], and Gzip_pack [10] is significantly better compared to other methods. For Gzip_pack, we first packed eight pixels of the bi-level image into one byte, and then compressed it using the Gzip command in Linux.

In some applications, adjacent frames are very similar, such as meter reading, bird/bat detection, etc. In such applications, the differences in two adjacent frames are small (only a part of the image is changing). Further data reduction can be achieved by compressing change frames rather than the original frames.

The goal of our current work is to explore the effectiveness of change coding in combination with binary image coding for data reduction in WVSNs (Fig. 3). Data reduction will lead to reduced communication energy consumption for the VSN. Change coding is a well-known concept in the image processing community, but we are evaluating its performance specifically for some applications of WVSNs. Rather than proposing a new change coding scheme, we are interested in achieving further data reduction by using change coding for specific applications in WVSNs, such as meter reading, bat/bird detection, etc.

We have explored the data reduction efficiency of three suitable bi-level image compression standards

(CCITT Group 4, JBIG2, and Gzip) in combination with change coding for a variety of changes in adjacent frames. We considered various types of changes in adjacent frames, such as the number, size, location, and shape of white objects.

The remainder of this article is organized as follows. Related work is presented in Section II. Section III presents the basic concept and architecture for change coding. A statistical model for simulating different variations in a set of frames is described in Section IV. Experimental results are discussed in Section V. Finally, conclusions are provided in Section VI.

II. RELATED WORK

In some applications of WVSNs, images represent the objects being monitored or the background. Few examples of such applications are target tracking [11]–[13], bat/bird detection [14] and Automatic Meter Reading (AMR) [15]–[17].

Images can be segmented into binary images in such applications, which leads to a significant reduction in data size. Binary image compression methods are extremely effective and have been researched extensively. The communication energy consumption of a VSN can be greatly reduced by implementing a suitable binary image compression method in combination with change coding.

Nandhini and Radha [11] developed a system that can perform both object detection and tracking with reduced complexity. They tracked detected objects using a Kalman filter. They extracted the centroids of objects from binary images using contour tracing. The centroids were then used as inputs for the Kalman filter to track the objects.

Another target tracking system for resource constrained WSNs was proposed in [12]. The authors implemented several specific signal processing algorithms for target detection, classification, and tracking. They used 5,000 binary images of humans and 4,000 binary images of non-humans for training. Another 2,000 binary images of humans and 1,500 binary images of non-humans were used for testing. The authors in [13] explored the impact of compressive sensing for target detection and tracking.

Wind energy plants have a severe impact on wildlife, with significant fatality rates for various kinds of birds and bats. The death of these birds and bats occurs due to their collision with the rotor blades of wind turbines. Vision-based monitoring systems, such as the one proposed in [14], reduce the mortality rate of birds and bats by using an optimized turbine control strategy when a bird/bat is detected.

Another application for WVSNs and binary is the automatic monitoring of energy meters. Ferrigno *et al.* [1], Elrefaei *et al.* [15], Shinde and Kulkarni [16], and Rodr_Iguez *et al.* [17] performed binary image processing for meter reading applications of WVSNs.

The compression efficiency of binary image compression methods was analyzed in [18]–[20]. The performance of lossless still image compression was explored in [18]. The authors analyzed the compression ratios of all the well-known compression methods available at the time. However, one

issue with the investigation in [18] is the absence of the latest standard, JBIG2. The other issue is the consideration of still images only. The execution times and compression ratios of various compression methods were explored in [19]. However, Kodituwakku and Amarasinghe [19] only considered textual data and still images.

Another comparison of compression methods was presented in [20]. The authors evaluated many compression schemes based on various medical images. They considered both the compression ratios and execution times of the compression methods. They concluded that compression efficiency depends on the type of images, meaning their results cannot be applied to machine vision applications.

Conventional video codecs such as JPEG2000, H.264 and MPEG-x have a complex encoding and simple decoding architecture, where raw video data is compressed by a powerful computer which can perform computationally complex operations. On the other hand, the resource-constraint VSN need to compress the data onboard before transmitting it to the server, which makes it essential for VSN to have a computationally efficient compression method. However, the conventional video coding methods are not appropriate for WVSNs because of the resource constraints such as limited onboard computational power, availability of limited energy in battery and communication bandwidth.

The performance of three video codecs for WVSNs including DISCOVER [21], H.264 Intra [22] and DCVS [23] was explored in [24]. This study is based on parameters such as computation complexity, total energy consumption, decoding complexity at receiver side, lifetime of the VSN and the quality of reconstruction. These codecs have high computational complexity which results in large energy consumption. In this study, an energy consumption model based on TelosB mote specifications was used.

Most recent work on encoding data in WVSN is based on Distributed Video Coding (DVC) [25]. DVC uses computationally less complex encoders at VSN, by shifting most of the computationally complex tasks to the server. In other words, in DVC, the computationally complex and energy consuming tasks are moved to the server, which is expected to be resourceful in both computational power and energy resources. A comprehensive review of the state of the art on DVC based codecs for WVSN was presented [26]. This review comprises of a comparative discussion of several well-known video codecs based on their functional aspects and performance comparison. The DVC is based on discrete cosine transformation (DCT), which is computationally more complex compared to the binary image compression methods.

There are many applications of WVSN but our evaluation of data reduction is for those which are based on bi-level images. The cyclist and human detection based on binary images, for guaranteeing their safety, was explored in [27]. The monitoring of birds which are flying towards the wind turbines, to divert them away and avoid possible accidents with the wind turbines has been presented in [28].
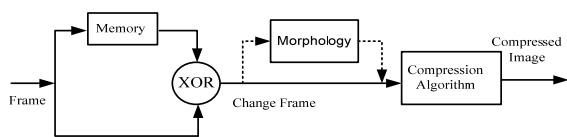
FIGURE 4. Proposed architecture for change coding.

TABLE 1. Memory requirements for saving a frame in the form of RLE.

Memory (KB)	Full circle	Semi-circle	Quarter-circle	Full ellipse	Semi-ellipse	Quarter-ellipse
RLE	1.8	1.8	1.3	2.1	2.1	1.6
Bi-level image	32	32	32	32	32	32

FIGURE 5. Problem caused by morphology and its solution.

Other studies which were based on bi-level images include the monitoring of meter readings [1], magnetic particle detection in hydraulics systems for failure prevention [29], human detection [30] and robot localization [31].

Compared to conventional video codecs and DVC, the computational complexity of binary image coding methods is very low. So, for the considered applications, further data reduction can be achieved by applying change coding in combination with binary image coding at the VSN, which is the core of our current work. Unfortunately, we could not find a relevant article in which the performance of change coding in combination with binary image coding method is explored. So, it will not be fair to present a comparison of algorithms which are computationally complex and are evaluated on different computing platforms.

In machine vision applications, images contain various objects that are different from those found in scanned textual and medical images. Thus, a thorough study on the effectiveness of change coding for machine vision applications is required.

III. CHANGE CODING

In some applications, the number of objects in the change frame (frame obtained by performing an XOR operation on two adjacent frames) is lower than the number of objects in each of the original adjacent frames. By compressing the change frame generated from two adjacent frames, we expect higher compression performance compared to simply compressing the original frames.

The additional cost of change coding is the memory required for storing frames in the form of white and black runs, and an XOR logic gate. The architecture for change coding is presented in Fig. 4. Every incoming frame is stored in the form of alternating runs of ones and zeros. Memory usage can be reduced by storing frames using Run Length Encoding (RLE). The change frame between two adjacent frames can be computed by performing an XOR operation on the respective segmented pixels of the current frame from the camera and the pixels of the previous frame from memory. The selected binary image compression algorithm is then applied to compress the processed CFs. A morphological operation (erosion, dotted lines in Fig. 4) is optional and may be performed on the CFs for removing illumination noise (illumination noise caused by segmentation errors).

VSNs have a strict memory size constraint. For storing a segmented frame in memory, 32 KB of memory is required (640 columns, 400 rows, $640 \times 400 = 25,600$ bits = 32 KB). Table 1 displays the memory requirements for saving an entire frame in the form of RLE for frames containing an average of 20 objects. The memory requirement remained nearly constant for different standard deviations in the number of objects in the original frames, and is approximately 2.1 KB for the worst case. Therefore, more than 29 KB of memory can be saved by storing a frame in memory using RLE.
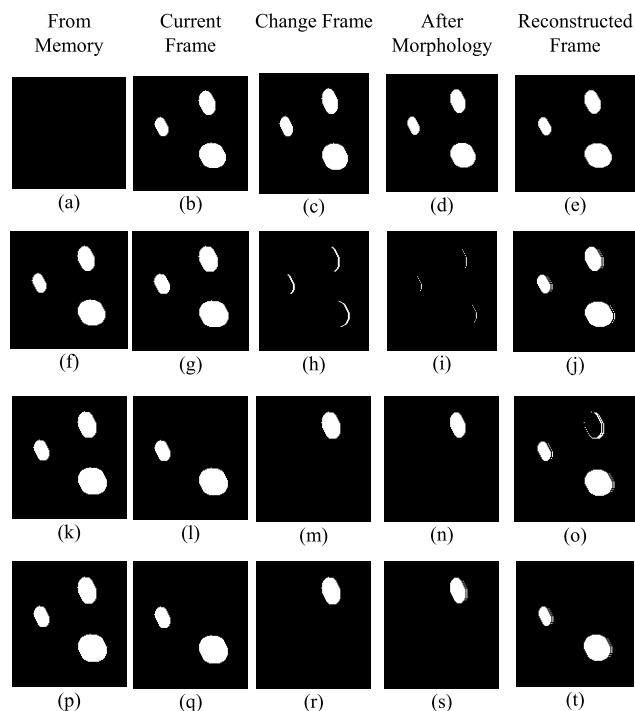
The problems caused by change coding and the solutions to those problems are presented in Fig. 5. Sections (a)-(e) illustrate the concept of change coding, while sections (f)-(o) illustrate the problems caused by change coding due to illumination noise and morphology operations.

Specifically, Fig. 5(o) shows that if an object moves away from the camera in the presence of illumination noise, then a ring shaped object will exist in the reconstructed frame. This ring shaped object in the reconstructed frame is unwanted noise caused by the loss introduced when using morphological erosion.

In order to resolve this problem, we propose the architecture presented in Fig. 6. Sections (p)-(t) in Fig. 5 are the resulting images after applying the architecture presented in Fig. 6. It should be noted that the ring shaped object is not present in the reconstructed image presented in Fig. 5(t).
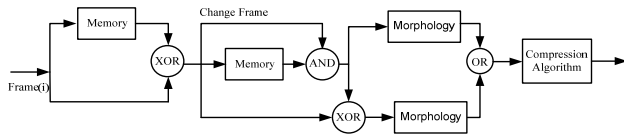
FIGURE 6. Block diagram of change coding implementation.

FIGURE 7. Effect of adding/removing objects in the OFs on the CFs.

IV. STATISTICAL MODEL FOR EVALUATION OF CHANGE CODING

Objects are randomly located in the images in many machine vision applications. The objects being monitored are typically moving. Thus, some objects may disappear/appear in adjacent frames. There could be significant changes in the shapes, sizes, and locations of the objects across a sequence of frames. The size of the compressed image is dependent on object shapes, object sizes, the number of objects, and their locations in the images.

Thus, for analyzing the efficiency of change coding, a mandatory requirement is to generate a rich set of frames. There is a need for a statistical model that is capable of generating a sequence of frames with numerous attributes, such as dissimilar locations, varying sizes, different shapes, and different numbers of objects. We have developed such a model and generated a rich set of images with these desired object characteristics. We used this set of images in combination with the selected compression standards for the performance evaluation of change coding.

Some objects may appear or disappear in continuous sets of frames. For example, in one frame there could be ten objects, out of which two (any number) may move out of the scene and will not be present in the next frame. Similarly, a few objects may move into the scene and the number of objects in the next frame will increase.

In our statistical model, we assume that μ objects are present in the frames on average. A standard deviation of σ is used to simulate the effect of added/removed objects in the sequence of frames. Additionally, we consider that the size of some objects may grow due to illumination noise.

The performance of the binary image coding standards is dependent on the changing of pixel values from one to zero and vice versa. The trend of appearing/disappearing objects was simulated in one set of frames. The increase/decrease in the size of the objects due to illumination noise was simulated in a second set. Both of the generated sets have frames with different object shapes, such as semi-ellipses, quarter-ellipses, ellipses, semi-circles, quarter-circles, and circles. In our evaluation, we used frames of size 640x400, with randomly placed objects in a black background.

The purpose of developing the statistical model was to generate frames with the desired object characteristics. In doing so, we are able to analyze the compression efficiency of change coding. Real-world performance can be investigated prior to actual system implementation by simulating real-world situations in a statistical model.
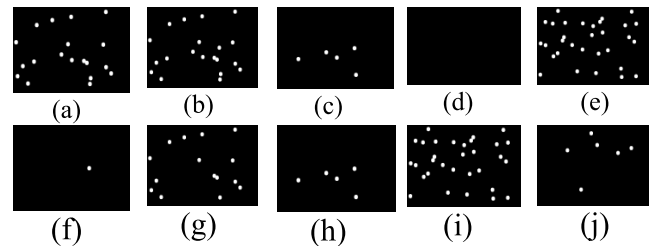
A. STATISTICAL MODEL FOR APPEARING/ DISAPPEARING OBJECTS

We generated 50 frames for each type of white object (quarter-circles, semi-circles, circles, ellipses, etc.) using our MATLAB model. In our MATLAB model, the number of objects in each of the 50 frames is determined by using a random number generator. The random number generator takes three input values: the mean (μ), the standard deviation (σ), and number of samples in the population (N).

In our analysis, the number of samples in the population is 50 frames, the mean is 20 objects, and number of objects varies from 2 to 22 (representing a 10-110% change in the number of objects in the set of frames).

The variation between 2 and 22 is used to simulate the effect of disappearing/appearing objects in a continuous set of frames. For each standard deviation, 50 frames with 20 objects on average are generated. This means that in the whole set of frames, the number of objects in various frames will be about 20. The variation in the number of objects in frames is based the standard deviation.

Five sample frames and the differences between adjacent frames are shown in Fig. 7. The placement of the objects in the sequence of frames in Fig. 7 is fully random. Fig. 7(a)-(e) represent the five Original Frames (OFs) with 20 objects on average and a standard deviation of 70%. Fig. 7(f)-(j) represent the CFs, where some frames have a higher number of objects compared to others.

We have graphed the number of objects in each of the 50 frames for the case where the standard deviation in the number of objects is 10% in Fig. 8. The horizontal axis represents the frame number in the sequence while the vertical axis represents the number of objects in each frame.

If there are 24 objects in the current frame, and the subsequent frame only contains 18 objects, then six objects will be present in the CF. If we compress the frame with 24 objects, the compressed image will be much larger than the size of the compressed CF containing only 6 objects. Similarly, if there are 16 objects in one frame and 5 more objects enter the field of view in the next frame, then the number of objects in the second frame is 21. However, the CF only contains 5 objects and its size will be smaller compared to compressing the OF containing 21 objects. The performance of change coding in relation to adding/removing objects is discussed in the results section.
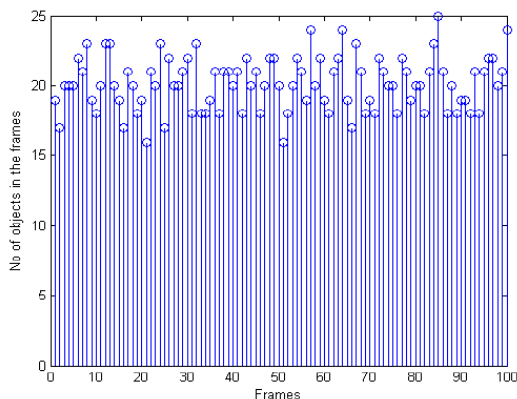
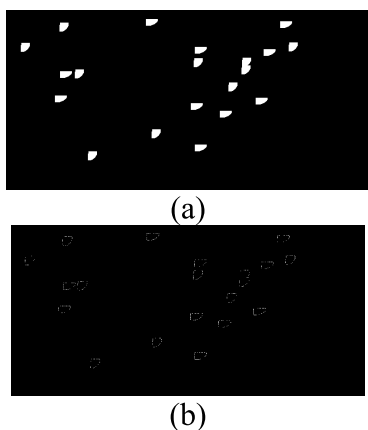**FIGURE 8.** Frames with different numbers of objects.

(a)

(b)

FIGURE 9. Effect of illumination noise on the CF.

B. STATISTICAL MODEL FOR SIMULATING ILLUMINATION NOISE

We generated 50 frames with different shapes of objects using our MATLAB model. The number of objects in all the frames is fixed at 20 and there is a chance for the size of the objects to increase by few pixels to simulate the effect of illumination noise. A sample frame and a CF between two adjacent frames are presented in Fig. 9. The objects are randomly placed in Fig. 9. Fig. 9(a) presents a frame containing exactly 20 objects with a possibility for the size of the objects to grow by a few pixels across the continuous set of frames. Fig. 9(b) presents the CF, where a few white pixels at the boundaries of the objects are present. This means that the size of the objects in the second frame increased by a few pixels at their boundaries.

Due to illumination noise, some pixels in the immediate surroundings of the objects may have a false value. We have analyzed this problem for varying numbers of pixels in the frames by assigning a false 1 value to a specified number of pixels at the boundaries of the objects.

We analyzed the performance of the compression standards for each object shape for both the OFs and CFs, where we assigned a false value to various numbers of pixels at the boundaries of the objects. Our aim is to find the intersection
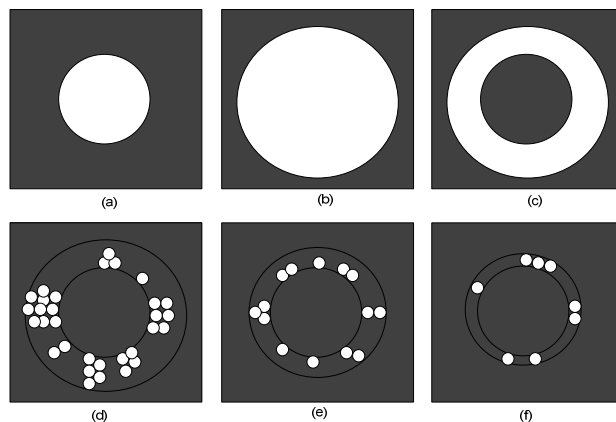

FIGURE 10. Effect of illumination noise on the location of pixels at the boundaries of objects.

between the file size of the OF coding and CF coding. The results of this analysis are discussed in both graphical and tabular form in the results section.

C. STATISTICAL MODEL FOR PLACEMENT OF NEW OBJECTS/PIXELS

The number and size of objects in a sequence of frames vary in real-world applications. Some objects may be added/removed in the sequence of frames. Similarly, the size of any of the objects may grow in the continuous set of frames due to illumination noise. In order to simulate real-world situations, the placement of new objects in the frame in our statistical model is fully random. This means that new objects can be placed anywhere in the frames. Similarly, some objects may move out of the scene. The removal of objects from the frame is also fully random in our model. This means that any of the objects in the frame can be removed. Added/removed objects can be observed in the frames in Fig. 7. It should be noted that their placement is fully random.

Another issue is the effect of illumination noise on the size of the objects. Illumination noise can affect some objects more severely than others. Similarly, it can affect some frames more severely than others. In order to simulate this effect, a false value is assigned randomly to pixels at the boundaries of the objects in our statistical model. Fig. 9 presents OF and CF. Only the changes are visible in the CF and the exact placement of the pixels is not clearly visible. The placement of the additional pixels due to illumination noise is fully random and is shown in Fig. 10, which presents the concept in a more intuitive format.

In Fig. 10, sections (a)-(c) show the growth in the radius of the objects. Ideally, if the radius of a circular object is increased by some integer value, then the CF will look like Fig. 10(c). The situation is very different in the real world. Illumination noise affects some objects more severely than others. Similarly, it affects some parts of an object more severely than others.

Any object may grow more on one side than the other sides. Three scenarios are shown in Fig. 10(d)-(f).
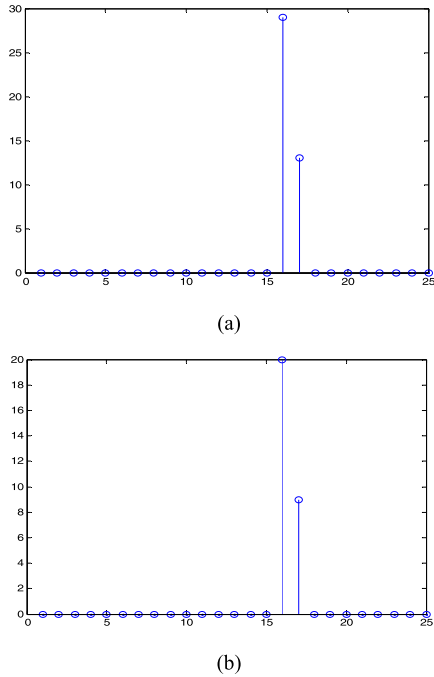
(a)

(b)

FIGURE 11. Histogram of illumination noise. (a): Histogram of frames generated using the statistical model. (b): Histogram of images captured in indoor environment.

FIGURE 12. Compression for adding/removing objects.

Fig. 10(d) shows a situation where the radius of an object is increased by 3 pixels and different sides of the objects are affected randomly. Some parts of the object grow more compared to the other parts. Fig. 10(e) and (f) contain two more scenarios, where the radius of the object is increased by 2 pixels and 1 pixel, respectively, and the placement of the pixels is random.

The size of an object may increase by 1, 2, or 3 pixels due to illumination noise (it could be increased up to any number and on any side of the object). It depends on the source of lighting in the environment. In an outdoor environment, the sources of the illumination noise are sun light, car light, etc. These may have more severe effect than indoor environment lighting. The statistical model is flexible, allowing the size of an object to grow by any specified number of pixels, while the placement of new pixels remains fully random.

In order to prove that our statistical model truly represents the illumination noise caused by external lighting in the real world (random noise), we must analyze a large number of frames and present the analysis in a compact form. We generated two sets of 50 frames to accomplish this goal. One set was generated by using our statistical model and the other set was generated by capturing real images of a page containing one circle. An image with only one white object of radius 15 is used in both sets. The CFs for both sets are detected and the average number of white pixels in the CFs is displayed graphically in Fig. 11.

The horizontal and vertical axes in Fig. 11(a) and (b) show the radius and the number of white pixels in the CFs respectively. The behavior of the number of affected pixels
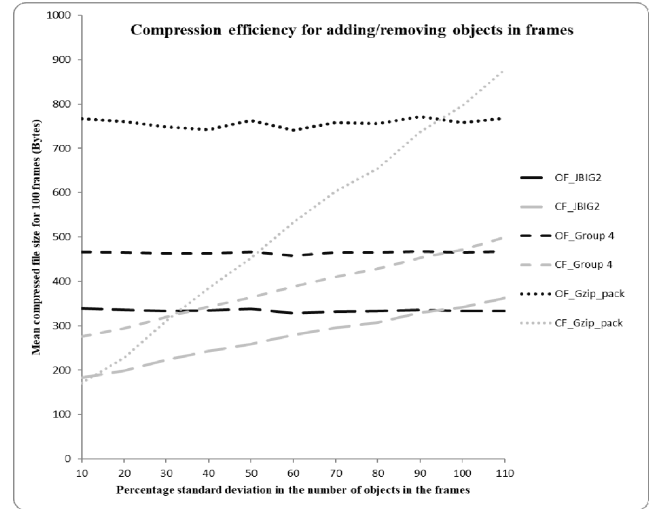
at the boundary of the object is nearly identical in both histograms.

This indicates that our statistical model is an accurate approximation of illumination noise in the real world. Only the no. of white pixels in both the histogram is different. There is an option for increasing the number of affected pixels in our statistical model. We used various affected pixels per object in the images in our simulations.

V. RESULTS
We compressed the generated frames by using the aforementioned compression methods. The results are presented in this section. We first discuss the implementation details of the compression standards, followed by the experimental results.

We used the Libtiff library [33] for CCITT Group 4 compression. The Ubuntu operating system has a gzip command, which we used for evaluating gzip compression. The JBIG2 implementation in the Leptonica image processing library is used for evaluating JBIG2 performance. We downloaded and compiled the Leptonica image processing library from [33].

The OF and CF lines in Fig. 12 and Fig. 13 represent the Original Frame and Change Frame, respectively. The compressed file sizes for the three compression standards, for both the OFs and CFs, are shown in Fig. 12. For each standard deviation, an average of 20 objects are present in the OFs. The CFs contain between 2 and 22 objects based on standard deviations of 10-110% in the OFs.

Fig. 12 shows that for each of the three compression standards, the compressed file size of the OFs is nearly constant for various standard deviations in the number of objects. The reason for this is that on average 20 objects are present in each of the 50 generated frames.

In Fig. 12, one can see that file sizes of the CFs increases with the increase in the number of objects (10-110% standard deviation creates 2-22 objects in the CFs). This is an intuitive result because the size of the compressed file is dependent on the number of objects in the frame. Specifically, it is
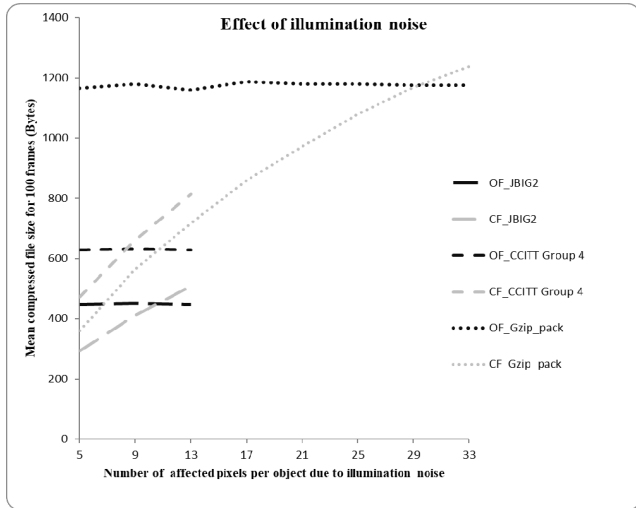
FIGURE 13. Impact of illumination noise on compression efficiency.

Method	Full circle	Semi-circle	Quarter-circle	Full ellipse	Semi-ellipse	Quarter-ellipse
Group 4	7	6	8	9	7	7
JBIG2	12	7	12	14	7	9
Gzip_pack	36	30	15	45	35	22

FIGURE 14. Effect of morphology operation on reducing the impact of illumination noise.

dependent on the no. of transitions from zeros to ones and vice versa.

The intersection between the size of OFs and CFs occurred at a standard deviation of approximately 95-100% (Fig. 12). Change coding has no benefit beyond this point. We determined this crossing point for all considered shapes of objects and it always occurred at a standard deviation between 90% to 100%.

The performance of both OF coding and CF coding is affected by illumination noise, which is presented in Fig. 13. We considered frames containing exactly 20 objects with several different numbers of affected pixels at the object boundaries. In Fig. 13, the horizontal axes represent the number of affected pixels per object while the vertical axes show the compressed file size.

Fig. 13 shows that the file size after compressing OFs is nearly constant, whereas the file size after compressing CFs increases from left to right. This increase is due to the presence of illumination noise. The number of affected pixels per object in the CFs increases from left to right, which is the main reason for the increasing file size in Fig. 13. Higher numbers of affected pixels per object cause additional transitions from zeros to ones and vice versa, causing the rise in compressed file size seen in Fig 13.

For CCITT Group 4 and JBIG2, the intersection between OF coding and CF occurred in images where illumination noise affected 8 and 12 pixels per object respectively (the shape used is quarter of circle). The intersection for Gzip_pack occurred in images where illumination noise affected 30 pixels per object. This indicates that CCITT Group 4 and JBIG2 are more sensitive to illumination noise.

We determined the intersection points for various shapes of objects. The results are presented in Table 2. For the various shapes in Table 2, the intersection point is always the highest for Gzip_pack. It should be noted that for all three compression standards, the intersection point is different for various shapes of white objects.

Illumination noise causes the size of objects to grow by varying amounts of pixels. This adversely affects the performance of change coding. The intersection point for illumination noise occurred very early compared to the case of appearing/disappearing objects in the sequence of frames. A morphology operation can be used to reduce the effect of illumination noise. By applying a morphology (in this case an erosion operation with diamond-shaped structure elements of size 2×2), one to two false isolated pixels (due to illumination noise) can be eliminated from the CFs. This will delay the intersection point until a higher number of affected pixels per object is reached (Fig. 14).

The performance of change coding depends on the application. The changes in contiguous frames are high in some applications, meaning high compression efficiency cannot be expected from change coding. However, if the adjacent frames in an application are highly similar, then excellent data size reduction will be achieved by applying change coding. For example, in meter reading applications, everything other than a few digits in adjacent frames remains constant. Thus, the CFs will contain very few digits and change coding will result in excellent compression.

In our previous work in [7], we have determined the communication energy consumption of various compression methods. We used the same procedure in our current work to determine the communication energy consumption for image coding and change coding and presented the results in Fig. 15. Fig. 15 shows that, by applying change coding in
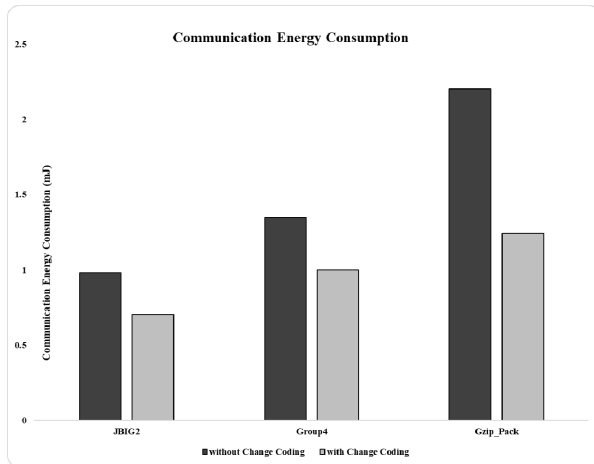
FIGURE 15. Communication energy for binary images with and without change coding.
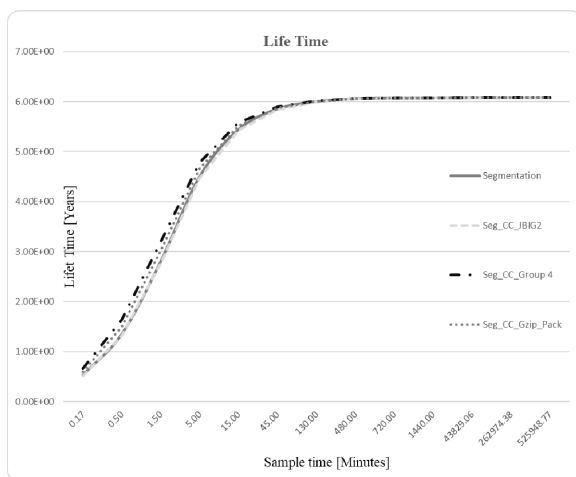

FIGURE 16. Life Time of the VSN with and without Change Coding.

combination with JBIG2, Group 4 and Gzip_pack, the communication energy consumption is reduced by 27%, 29% and 46% respectively.

In our previous work in [4], we have predicted the lifetime of the VSN based on the energy consumption of the various image processing tasks. In the current work, we have adopted the same procedure for predicting the life time of the VSN, by applying change coding at the VSN (in addition to other tasks) and the results are shown in Fig. 16.

Though the communication energy consumption of JBIG2 is lowest, but due to its high processing time, its total energy consumption is high, which resulted in lowest (worst) lifetime curve for it in Fig. 16. On the other hand, the computational complexity of Group 4 is low and its compression ratio is also good, which resulted in highest lifetime curve.

VI. CONCLUSION
We evaluated the performance of change coding in combination with three binary image coding methods: Gzip_pack,

JBIG2, and CCITT Group 4. Frames with various kinds of changes, such as different sizes, various shapes, and different numbers of white objects were used in our evaluation. We determined that image coding in combination with change coding is better than image coding alone. Furthermore, the performance of image coding in combination with change coding is better for up to 95% variance in terms of the number of objects in the set of frames. No additional saving is achieved beyond 95% changes in number of objects in the frames. Object size may increase at boundaries by a varying number of pixels due to illumination noise, which negatively affect the performance of change coding. The performance of change coding is inferior to image coding in cases where more than four pixels per object were affected by illumination noise. Our results also demonstrate that morphology can be applied to minimize the impact of illumination noise. We applied change coding to real captured images and verified the results that we obtained based on statistically generated images. Thus, we conclude that change coding in combination with CCITT Group 4 and morphology is an effective novel approach for reducing the data that needs to be communicated in WVSNs. The CCITT Group 4 is preferred due to its good compression performance in combination with less computational complexity. Reduced data size along with less computational complexity leads to reduced total energy consumption which will result in increased life time of the VSN.

APPENDIX
Standard deviation: Standard deviation is used to simulate the effect of added/removed objects in the sequence of frames. 10-110 % standard deviation represents 2-22 objects in the change frame.

Communication Energy: The energy spent on transmitting the compressed images from VSN to the server.

Illumination Noise: The noise in the captured images due to the external light in the environment.

Morphology: The process of removing small unwanted noise in the images.

REFERENCES
[1] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto, "Balancing computational and transmission power consumption in wireless image sensor networks," in *Proc. IEEE Int. Conf. Virtual Environ., Hum.-Comput. Interfaces, Meas. Syst.*, Messina, Italy, Jul. 2005, p. 6.

[2] W. Chen, A. Hamid, and K. Richard, "Real-time human posture reconstruction in wireless smart camera networks," in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Washington, DC, USA, Apr. 2008, pp. 321–331.

[3] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "SensEye: A multi-tier camera sensor network," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, New York, NY, USA, 2005, pp. 229–238.

[4] K. Khursheed, M. Imran, M. O'Nils, and N. Lawal, "Exploration of local and central processing for a wireless camera based sensor node," in *Proc. IEEE Int. Conf. Signal Electron. Syst.*, Gliwice, Poland, Sep. 2010, pp. 147–150.

[5] K. Khursheed, M. Imran, A. W. Malik, M. O'Nils, N. Lawal, and B. Thörnberg, "Exploration of tasks partitioning between hardware software and locality for a wireless camera based vision sensor node," in *Proc. Int. Symp. Parallel Comput. Electr. Eng. (PARELEC)*, Apr. 2011, pp. 127–132.

[6] M. Imran, K. Khursheed, M. O'Nils, and N. Lawal, "Exploration of target architecture for a wireless camera based sensor node," in *Proc. 28th NORCHIP Conf.*, Tampere, Finland, Nov. 2010, pp. 1–4.

[7] K. Khursheed, M. Imran, N. Ahmad, and M. O'Nils, "Selection of bi-level image compression method for reduction of communication energy in wireless visual sensor networks," *Proc. SPIE*, vol. 8437, p. 84370M, May 2012. [Online]. Available: http://dx.doi.org/10.1117/12.923716

[8] F. Ono, W. Rucklidge, R. Arps, and C. Constantinescu, "JBIG2—The ultimate bi-level image coding standard," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Vancouver, BC, Canada, Sep. 2000, pp. 140–143.

[9] *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus*, vol. 7, document Rec. ITU-T T.6, Fascicle VII.3, Terminal Equipment and Protocols for Telematic Services, The International Telegraph and Telephone Consultative Committee (CCITT), Geneva, Switzerland, 1988. Accessed: Feb. 2, 2018. [Online]. Available: http://www.itu.int/en/ITU-T/publications/Pages/recs.aspx, http://www.itu.int/rec/T-REC-T.6-198811-I/en

[10] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[11] S. A. Nandhini and S. Radha, "Compressed sensing based object detection and tracking system using measurement selection process for wireless visual sensor networks," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 1117–1122, doi: 10.1109/WiSP-NET.2016.7566310.

[12] X. Wang, S. Wang, D.-W. Bi, and J.-J. Ma, "Distributed peer-to-peer target tracking in wireless sensor networks," *Sensors*, vol. 7, no. 6, pp. 1001–1027, 2007.

[13] S. Fayed, S. Youssef, A. M. El-Helw, M. Patwary, and M. Moniri, "Compressive sensing-based target tracking for wireless visual sensor networks," in *Proc. 18th Int. Conf. Comput.*, vol. 1. 2014, pp. 44–50.

[14] N. Scholz, J. Moll, M. Mälzer, K. Nagovitsyn, and V. Krozer, "Random bounce algorithm: real-time image processing for the detection of bats and birds," *Signal, Image Video Process.*, vol. 10, no 8, pp. 1449–1456, 2016, doi: 10.1007/s11760-016-0951-0.

[15] L. A. Elrefaei, A. Bajaber, S. Natheir, N. AbuSanab, and M. Bazi, "Automatic electricity meter reading based on image processing," in *Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT)*. Nov. 2015, pp. 1–5, doi: 10.1109/AEECT.2015.7360571.

[16] M. V. Shinde and P. W. Kulkarni, "Reading of energy meter based on image processing technology," *Int. J. Electron. Commun. Comput. Eng.*, vol. 5, no. 4, pp. 298–302, Jul. 2014.

[17] M. Rodríguez, G. Berdugo, D. Jabba, M. Calle, and M. Jimeno, "HD-MR: A new algorithm for number recognition in electrical meters," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 87–96, 2014, doi: 10.3906/elk-1202-34.

[18] R. B. Arps and T. K. Truong, "Comparison of international standards for lossless still image compression," *Proc. IEEE*, vol. 82, no. 6, pp. 889–899, Jun. 1994.

[19] S. R. Kodituwakku and U. S. Amarasinghe, "Comparison of lossless data compression algorithms for text data," *Indian J. Comput. Sci. Eng.*, vol. 1, no. 4, pp. 416–425, 2010.

[20] J. Kivijärvi, T. Ojala, T. Kaukoranta, A. Kuba, L. Nyúl, and O. Nevalainen, "A comparison of lossless compression methods for medical images," *Comput. Med. Imag. Graph.*, vol. 22, no. 4, pp. 323–339, 1998.

[21] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," in *Picture Coding Symp.*, vol. 17. 2007, pp. 1103–1120.

[22] H., Kalva, "The H.264 video coding standard," *IEEE Multimedia*, vol. 13, no. 4, pp. 86–90, Oct./Dec. 2006.

[23] L.-W. Kang and C.-S. Lu, "Distributed compressive video sensing," in *Proc. IEEE Int. Conf. Acoust., Speech SP*, Taipei, Taiwan, Apr. 2009, pp. 1169–1172.

[24] N. Imran, B.-C. Seet, and A. C. M. Fong, "A comparative analysis of video codecs for multihop wireless video sensor networks," *Multimedia Syst.*, vol. 18, no. 15, pp. 373–389, Oct. 2012, doi: 10.1007/s00530-012-0258-0.

[25] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[26] N. Imran, B.-C. Seet, and A. C. M. Fong, "Distributed video coding for wireless video sensor networks: a review of the state-of-the-art architectures," *SpringerPlus*, vol. 4, no. 1, p. 513, 2015.

[27] M. Imran, B. Rinner, S. Z. Zand, and M. O'Nils, "Exploration of pre-processing architectures for field-programmable gate array-based thermal-visual smart camera," *J. Electron. Imag.*, vol. 25, no. 4, p. 041006, 2016, doi: 10.1117/1.JEI.25.4.041006.

[28] N. Ahmad, M. Imran, K. Khursheed, N. Lawal, and M. O'Nils, "Model, placement optimisation and verification of a sky surveillance visual sensor network," *Int. J. Space-Based Situated Comput.*, vol. 3, no. 3, pp. 125–135, 2013.

[29] M. Imran, K. Khursheed, N. Lawal, M. O'Nils and N. Ahmad, "Implementation of wireless vision sensor node for characterization of particles in fluids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 11, pp. 1634–1643, Nov. 2012.

[30] J. M. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin, "Foreground-adaptive background subtraction," *IEEE Signal Process. Lett.*, vol. 16, no. 5, pp. 390–393, May 2009.

[31] H. Lee, H. Dong, and H. Aghajan, "Robot-assisted localization techniques for wireless image sensor networks," in *Proc. 3rd Annu. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, vol. 1. Sep. 2006, pp. 383–392.

[32] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 337–343, May 1977.

[33] *Using The TIFF Library*. Accessed: Feb. 2, 2018. [Online]. Available: www.libtiff.org/libtiff.html

KHURSHEED AURANGZEB received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology Abbottabad, Pakistan, in 2006, the M.S. degree in electrical engineering (System on Chip) from Linköping University, Sweden, in 2009, and the Ph.D. degree in electronics design from Mid Sweden University, Sweden, in 2013. From 2013 to 2016, he was an Assistant Professor/HoD with the Electrical Engineering Department, Abasyn University, Peshawar, Pakistan. He is currently an Assistant Professor with the College of Computer and Information Science, King Saud University, Riyadh, Saudi Arabia.

MUSAED ALHUSSEIN was born in Riyadh, Saudi Arabia. He received the B.S. degree in computer engineering from King Saud University, Riyadh, in 1988, and the M.S. and Ph.D. degrees in computer science and engineering from the University of South Florida, Tampa, FL, USA, in1992 and 1997, respectively. Since 1997, he has been with the Faculty of the Computer Engineering Department, College of Computer and Information Science, King Saud University. He is currently the Founder and the Director of Embedded Computing and Signal Processing Research Laboratory. His research activity is focused on typical topics of computer architecture and signal processing and with an emphasis on VLSI testing and verification, embedded and pervasive computing, cyber-physical systems, mobile cloud computing, big data, eHealthcare, and body area networks.

MATTIAS O'NILS received the B.S. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 1993, and the Licentiate and Ph.D. degrees in electronics system design from the Royal Institute of Technology, Stockholm, Sweden, in 1996 and 1999, respectively. He is currently a Professor and the Dean of the Department of Electrical Engineering and leads a research group in embedded systems design at Mid Sweden University. His current research interests include design methods and implementation of embedded systems, with a specific interest in implementation of real-time video processing systems.

• • •