A distributed scheme for energy-efficient event-based target recognition using Internet of Multimedia Things

Manal Alsabhan¹, Adel Soudani¹ and Manan Almusallam²

Abstract
The availability of low-cost embedded devices for multimedia sensing has encouraged their integration with low-power wireless sensors to create systems that enable advanced services and applications referred to as the Internet of Multimedia Things. Image-based sensing applications are challenged by energy efficiency and resource availability. Mainly, image sensing and transmission in Internet of Multimedia Things severely deplete the sensor energy and overflow the network bandwidth with redundant data. Some solutions presented in the literature, such as image compression, do not efficiently solve this problem because of the algorithms' computational complexities. Thus, detecting the event of interest locally before the communication using shape-based descriptors would avoid useless data transmission and would extend the network lifetime. In this article, we propose a new approach of distributed event-based sensing scheme over a set of nodes forming a processing cluster to balance the processing load. This approach is intended to reduce per-node energy consumption in one sensing cycle. The conducted experiments show that our novel method based on the general Fourier descriptor decreases the energy consumption in the camera node to only 2.4 mJ, which corresponds to 75.32% of energy-saving compared to the centralized approach, promising to prolong the network lifetime significantly. In addition, the scheme achieved more than 95% accuracy in target recognition.

Keywords
Multimedia Internet of Things, wireless multimedia sensor network, multimedia sensing, features extraction, object recognition, low-energy processing, Fourier descriptors, distributed processing

Date received: 4 August 2021; accepted: 3 March 2022

Handling Editor: Yanjiao Chen

Introduction
Internet of Multimedia Things (IoMT),¹ also referred to as Multimedia Internet of Things (MIoT),² are networks in which multimedia things can interact with one and with other things connected to the Internet to provide multimedia-based services and applications. Wireless multimedia sensor networks (WMSNs)³ are the key infrastructure for IoMT applications. WMSNs enable image-based object recognition and tracking that are in some cases difficult or impossible

¹Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
²Department of Computer Science, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

Corresponding author:
Adel Soudani, Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 57168, Riyadh 11574, Saudi Arabia.
Email: asoudani@ksu.edu.sa

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
to achieve, especially in remote and high-risk environments such as monitoring the natural habitats of wild animals, land border control, underwater marine life observation, applied in green forests instead of traditional fire lookout towers for initiating fire alarms or triggering the extinguishers in response to fire smoke clouds, and monitoring moving objects within a supervised environment.

However, sensors are challenged by the limitations associated with constrained memory, limited buffer size, processing capabilities, transmission bandwidth and Quality of Service (QoS), and energy resources. These limitations are complicated even further due to the nature and size of multimedia data.

In a multimedia-based monitoring system, camera nodes are programmed to acquire and process visual data. In a typical setting depicted in Figure 1, a camera node uses the underlying wireless technology to stream the captured visual data to a backend server. Nevertheless, camera nodes are typically powered by irreplaceable and non-rechargeable batteries. Therefore, maximizing the network lifetime is a critical challenge. To address this problem, several research efforts proposed energy-efficient routing protocols, data compression algorithms, and distributed processing models.

In WMSN-based monitoring and tracking applications, the periodic image transmission to the end-user severely impacts the energy available in the network. In depth, the transmission of large volumes of multimedia data requires an intense activity of the wireless transceiver consuming a high level of energy. Compression techniques can have a significant role in preserving the network energy through data reduction. However, most of the available compression algorithms are inadequate for low-power processing due to their high-processing complexity. In addition, the traditional compression model in which a node compresses and sends all captured images can occasionally provide the end-user with irrelevant data and exhaust the network’s bandwidth.

One potential solution to extend the network lifetime and the application viability will be to process the captured images locally, at the source sensor, to detect relevant events to the end-user and then send only a compact representation of the useful data to the remote-control server through the network. Even though this approach is suitable for image-based recognition applications, it would require careful effort to design a low-complexity scheme that provides a tradeoff between the accuracy of target recognition and the energy savings on the source sensor. However, an accurate image-based recognition process might require the invocation of sophisticated methods for features extraction. Therefore, executing the whole scheme in one node could strongly exhaust its embedded energy that questions the practical validity of the application. We think that a new method of in-network cooperative execution of the designed recognition scheme over a cluster of nodes can provide a practical solution to the problem of image-based target recognition in IoMT.

The fundamental question we are addressing is to find the best way to construct a distributed energy-efficient scheme where a cluster of nodes collaborates in image-based target recognition. The idea is to balance the processing load through a set of nodes that form the processing cluster to verify whether the captured image contains the event of interest. This approach considerably reduces the amount of data transmitted to the sink node, which preserves the source sensor’s energy and contributes to the extension of the network lifetime.

We believe that a distributed processing model can provide energy-efficient performance compared to a centralized model in which a camera sensor depletes its energy rapidly. Nevertheless, the efficiency of this event-based sensing scheme mainly depends on striking a balance between affordable computational complexity and satisfying the accuracy of target recognition.

The main contribution of this article is to design and implement a low-complexity distributed sensing scheme in WMSN for target detection and recognition based on general Fourier descriptors (GFDs). We conduct a detailed experiment in which we evaluate the performance of the proposed distributed sensing scheme. It details the analysis of the results of the specified approach and discusses its capability to achieve low-power sensing and notification. The innovation of this scheme is to reduce the communication overhead and per-node energy consumption while ensuring efficient notification to the end-user. Performance analysis shows that the proposed scheme outperforms related work in target recognition while providing considerable savings in the network’s energy levels.

The rest of the article is organized as follows. The “Related work” section discusses the literature related to this research problem. Next, in the “Methodology”
section, we detail the design of the proposed detection scheme using GFD as a feature extraction method. Finally, in the “Results and discussion” section, we discuss the implementation and the experimentation to evaluate the performance of the presented scheme. We analyze the recognition capability and present the results of this work compared to similar literature approaches. In the last section of the article, we conclude, and we provide some future highlights.

Related work

In the context of energy efficiency, substantial research was conducted to design low-energy multimedia delivery with adequate QoS. Our investigation shows that the most common approaches to assure energy efficiency in WMSN are routing protocols, data compression algorithms, and distributed data processing. The low-energy adaptive clustering hierarchy (LEACH) routing protocol is a cluster-based routing protocol used in Wireless Sensor Networks (WSN) to achieve such a purpose. Unfortunately, applying it in WMSN becomes ineffective, especially when multimedia data and the network scale increase.8 Another promising approach is to reduce the size of multimedia data, improving the packet throughput in the network. With the application of data compression techniques, the number of packets transmitted to the end-user through the network will decrease.10

Consequently, it will help extend the network lifetime, reduce congestion, and enhance service quality from the end-user perspective. However, previous studies11,12 have shown that most standard compression algorithms are developed for resourceful computers. Thus, these algorithms are not applicable in the context of WMSN as they require extensive resources and high computational capabilities, contrary to the sensor’s resource constraints. Leila et al.13 reduced the sensed data using multiple compression stages to meet a certain level of resolution based on the end application demand. However, the main drawback of this work is the high computational cost of iterative compression processes. Wang et al.14 reduced the high computational cost of wavelet-based compression15 by using a two-dimensional discrete cosine transform (2D-DCT) compression technique that avoids iterations and compresses the sensed data using only the first level of transformation. The 2D-DCT is considered an adequate compression implementation for constrained sensors. Leila et al.13 proposed a hardware implementation of compression, which is considered an unscalable solution for large-scale sensor networks because it raises the implementation cost. The novel discrete Tchebichef transform (DTT)16 was applied to the region of interest (ROI) instead of the whole image. This approach enhances the discrete cosine transform (DCT) to improve its algorithm complexity and energy efficiency.

In contrast, distributed compression1 utilizes a set of cooperating nodes in executing the compression scheme. This approach distributes the computational process across a cluster of nodes to balance the processing load, save node energy, and reduce the size of transmitted data. The Slepian and Wolf17 theorem is a compression technique of two or more correlated data streams followed by joint decoding on the receiver side. Using information from correlated sensors reduces the number of transmitted packets needed. This scheme reduces the redundancy of similar data, which will relieve the receiver’s processor and memory from accepting insignificant communication requests. However, the Slepian–Wolf theorem increases the number of exchanged packets because of node cooperation and cluster formation. In addition, the mathematical coding model needs to be investigated in the context of multimedia data to ensure its efficiency for ATmega128 microcontrollers. Wu and Abouzeid18 presented and evaluated a distributed image compression technique based on wavelet transformation. They addressed data exchange operations that drain network energy due to extensive data broadcast and communication. Evaluation results showed that the proposed scheme prolonged the network lifetime, thus demonstrating the feasibility of distributed image processing.

Xu et al.19 attest to the performance of cluster-based hybrid computing paradigm for collaborative sensing and processing in WSNs compared to distributed computation paradigms: mobile agent and client/server distributed model. The model performance proved to be energy-efficient and was therefore scalable. However, this needs further investigation to prove the efficiency of heavy data processing, like images or videos, in WMSNs.

Qi et al.20 proposed a distributed multisensory target detection method. The idea is to detect the target from different angle resolutions upon entry to the detection area boundaries. Then, the node will aggregate a notification and send it to the base station to announce the moving target location. The performance analysis indicates improved detection probability using collaborative node sensing than centralized processing using a single node. Lin et al.21 presented a distributed approach to recognize a given identity based on feature extraction from images. First, the scheme extracts and detects the face region, and then the face components are detected. Face components are distributed among nodes to be processed based on parallel processing. However, this work has some limitations in computational and processing sharing, and there is an urge to attest to the algorithm reliability under different network scales. In a multiface detection method,22 camera nodes locally execute the face boundary detection. A
sink node receives the information of interest instead of the whole image to complete the object recognition process. This low-complexity approach removes the redundant data in the captured scene and reduces data traffic in the network. However, this technique is not helpful in object tracking and recognition application because it will load the network with unsure data.

The work presented in Zam et al. uses a clustering approach to detect and track an identified target. However, this work is based on a combination of acoustic and visual sensors equipped with passive infrared motion detectors. In addition, object identification is accomplished in the sink node. On the contrary, Koyuncu et al. combine audiovisual and scalar data to enhance object recognition and classification capabilities. This approach significantly decreased the network traffic, which will consequently prolong the network lifetime. Latreche et al. are based on the image fusion algorithms where a final informative image results from combined scenes captured in the monitoring area at different distances and resolution angles. This hybrid multimedia image fusion uses integer lifting wavelet transform (ILWT) and the DCT to generate high-quality fused images. Two steps are then applied to the fused image: extracting low-frequency coefficients of the image, followed by another phase to capture the satisfactory detail coefficients of the same image.

Nevertheless, despite the proven detection accuracy of this approach, energy efficiency needs to be evaluated for ATmega128 microcontrollers. Presented work in Zam et al. introduces an energy-aware collaborative tracking and moving detection scheme for WMSN. The proposed method is based on collaborative sensors to extract a lightweight informative image from a multi-ple captured scene to decrease the computational communication cost. An attractive method for minimizing energy consumption and extending the network lifetime is to use a local event-based sensing and detection scheme. This technique can reduce image data redundancies and lower the network traffic data while preserving adequate image quality. A way to accomplish this method is to use an ROI descriptor at the sensor node to detect whether the image captures an even interest and sends the minimum required data to the end-user. This approach reduces the data transmitted to the sink node; consequently, it preserves the energy of the source sensor and the energy of the other nodes of the network. Adopting this technique requires designing an efficient image analysis method to detect the target with invariance to translation, orientation, and scaling properties.

A motion detection framework, developed for WMSNs in surveillance areas, divides the captured image into sets of small blocks. The framework then discovers differences between the blocks of the captured image and the reference image. This approach helps not only to save energy but also to keep bandwidth usage low. However, this work is intended for object appearance detection applications where the identification and classification tasks are shifted to the base station. Vasuhi et al. have used the Haar wavelet implemented locally in the sensor for object feature extraction in WMSN, but they did not address the scheme’s computational complexity and the energy consumption efficiency.

In a distributed two-hop, clustered-image transmitting scheme, camera-equipped nodes act as cluster heads and distribute the compression tasks among the nodes in the cluster. Cluster nodes participate in both the distributed compression and transmission process. This approach balances the energy consumption between the cooperated nodes, which extends the whole network lifetime. However, the transmission of a stream of images exhausted the network’s energy and increased the contention and congestion in the network. Chefi et al. used a hardware platform for energy conservation which was not considered a scalable solution because of the estimated high-cost implementation.

Nikolakopoulos et al. introduce a new image compression technique based on a quad-tree decomposition combined with an inpainting image algorithm for image restoration. The authors prove it was an efficient low-power solution for computational complexity in WMSNs compared to JPEG, LZW, and JPEG2000 compression algorithms. In Wang et al., an artificial immune system–based image pattern recognition method was presented, but the associated energy consumption was remarkably high; therefore, it is unsuitable for sensors with constrained energy resources.

Alhilal used a shape-based descriptor for target recognition locally executed at the source node. The obtained results have demonstrated an impressive gain in energy. However, the centroid distance and the curvature signatures used for the recognition capability of the presented scheme suffer from accuracy problems. Specifically, the proposed scheme has high sensitivity and variance to the characteristics of the detected objects in the images. Moreover, the proposed scheme was built around the assumption of a single object appearing in the camera scene.

Bouacheria et al. improve the Routing Protocol for Low-power and Lossy Network (RPL) to accommodate the transport of compressed videos. The RPL is mainly used to deliver scalar data with an accepted QoS assurance level. However, the authors improved it
with a new version called Multi-Instance RPL (MI-RPL) for multimedia content. The packet prioritizes video frames, which significantly improves compared to traditional RPL in terms of energy-efficient compressed video delivery and good QoS level.

In the same context, Bidai36 improves the video traffic delivery by customizing the RPL specification to support the multipath version of RPL to deal with multimedia contents. This work shows that using a multipath routing scheme will distribute the video load on many routing paths, efficiently balancing the energy consumption between the network nodes. This work provides reasonable and adequate QoS performance metrics for multimedia applications.

In this work, we aim to design and implement an efficient image-based target recognition scheme for WMSN based on distributed implementation approach for the different tasks of the scheme. The distribution of processing load over the nodes of a processing cluster is intended to reduce per-node energy consumption. Since the nodes of the processing cluster will be assigned dynamically every sensing cycle, the energy consumption will be balanced over the nodes, eventually contributing to extending the network lifetime. This article proposes a scalable and energy-efficient distributed scheme for image-based target recognition in IoMT applications.

**Methodology**

In this section, we present the design of the proposed object detection and recognition sensing scheme. The main principle is to balance the processing load across nodes that form a processing cluster. This scheme will reduce the per-node energy consumption and, consequently, extend the network lifetime.

**General description of the object recognition scheme**

In the proposed approach, an event of interest is defined by a set of distinctive features referred to collectively as the target’s signature. During the setup phase, the wireless multimedia sensor receives, through the network, the descriptor of the object to be tracked. This signature is defined by the end-user in offline mode and will be loaded onto the preconfigured processing sensors to recognize a specific target. During the runtime, the camera nodes periodically sense the surrounding environment. Once an event is detected, clustered sensors start the object detection and extraction process from the captured scene.

The proposed system is designed to be scalable for detecting different event types and to provide dynamic notification based on the application’s requirements. The execution of the scheme in-network enables locally to decide whether a new detected object is the target object in the captured scene. The scheme provides different detection notification types according to the requirements of the application. It implements the notification with a simple byte, the transmission of the detected object descriptors, or the transmission of a representation of the extracted ROI.

The recognition process in the cluster is achieved based on the low-complexity feature extraction method. Then, the extracted features are compared against the target signature. When the matching process indicates significant similarities, a notification is sent to the end-user. Otherwise, the sensed event is discarded, and the camera sends a message “No Target is Found” to the end-user and the camera sensor resumes the search for an event. This study demonstrates an efficient and scalable scheme for target detection that is low in computational complexity and high accuracy in addition to reducing per-node storage requirements and communication overhead (see Figure 2).

**Local event detection**

Background subtraction is commonly used in WMSN applications to detect an object’s appearance or movement by identifying the ROI. An ROI is defined by computing the difference between the color/pixel intensity in the captured scene frame (foreground image) and the color/pixel intensity of a static scene frame (background image).

**Background subtraction.** Piccardi37 reviewed the background subtraction methods and ranked them according to their complexity, storage requirements, and detection accuracy. The Running Gaussian Average stood out as a simple background subtraction process that provides satisfying accuracy and limited memory requirements. As an intensity-based background subtraction method, the Running Gaussian is sensitive to the image’s brightness changes. Nevertheless, its characteristics are aligned with the limited resources of sensors. Using the Running Gaussian Average algorithm to extract the ROI from the background image achieved high-speed object extraction with minimum hardware requirement and low-power consumption.38,39

Assuming a grayscale image \((M)\) composed of \(p \times p\) pixels, the background pixel value at frame \(n\) is updated by running a Gaussian probability density function as follows

\[
\beta_n = \beta_{n-1} + \alpha(F_n - \beta_{n-1})
\]

where \(\beta_n\) is the updated background average, \(F_n\) is the current frame intensity, \(\beta_{n-1}\) is the previous background average, and \(\alpha\) is an updating constant whose value ranges between 0 and 1 and represents a trade-off between stability and quick update.
A pixel is classified as foreground (i.e. belongs to an updated object) if the condition expressed in equation (2) is met

$$\hat{M} = \begin{cases} 1 & \text{if } |F_n - \beta_{n-1}| > \text{Thr} \\ 0 & \text{Otherwise} \end{cases}$$

where $\hat{M}$ is a binary image and the $\text{Thr}$ is the threshold.

**Extraction of ROI.** Recognizing an object of interest starts by isolating the set of blocks in $M$ that represents the ROI. In literature, there are several methods used to separate the ROI from the captured scene, such as row and column scanning functions, an iterative threshold approach, and a region operating segmentation algorithm.

The region operating segmentation algorithms, such as the seed region growing algorithm, are appropriate for any image type and can be used in wide applications. The algorithm grows predetermined seed pixels into regions until all of the image’s pixels have been assimilated. Nevertheless, region operating segmentation algorithms have higher processing complexity compared to thresholding approaches.

In our research, we focus on a trade-off between processing complexity and detection accuracy. Therefore, we adopt an iterative thresholding algorithm.

**Figure 2.** Flowchart of the proposed distributed event-based detection and recognition scheme.
The probabilistic approach is one technique based on the threshold method for extracting the ROI. The algorithm subdivides the image into sub-blocks and counts the total number of pixels participating in the foreground object.

Assuming a foreground block denoted by $\beta_n(j)$ and a background block denoted by $\beta_{n-1}(j)$, a new object is detected when the difference between the image blocks is significantly higher than a certain threshold $\text{Thr}$, as expressed in the following equation:

$$\sum_{j=1}^{k} |\beta_n(j) - \beta_{n-1}(j)| > \text{Thr} \quad (3)$$

This approach reduces memory occupancy and energy consumption related to pixel processing compared to row and column scanning functions and region-growing algorithms.

**Extraction of features’ vectors.** In WMSNs, the starting point for object identification and detection is extracting the essential shape features. The shape-based features provide a compact representation that is suitable for storage and communication. For a low communication overhead, the extracted feature descriptors should be presented by a minimum number of bytes to reduce the required data for end-user notification.

Once the object is detected, the scheme will extract the blocks that form the valuable area and isolate them from unnecessary blocks (see Figure 3).

In the literature, there are many shape descriptions and techniques to measure similarity. These techniques are summarized in Yang et al.\textsuperscript{42} In this phase, the presented scheme proceeds to extract the features’ vectors from the obtained ROI to complete the object recognition.

The GFD\textsuperscript{9,43} is a mathematical model that uses the Fourier transformation to transform a shape signature into a set of descriptor features. First, GFD transforms the input image $f(x_i,y_i)$ of size $N \times M$, where $f$ is defined by $f(x_i,y_i)$: $1 \leq i \leq M, 1 \leq j \leq N$ into a polar image $f(r,\theta)$ using the following equations:

$$r = ((x-x')^2 + (y-y')^2)^{0.5} \quad (4)$$

$$\theta = \tan^{-1}(y-y'/x-x') \quad (5)$$

where $x'$ and $y'$ are the mass center of the shape. Then, the Fourier transformation takes place to extract the signature feature vector set, referred to as Fourier descriptors (FDs), using the following equation:

$$\text{FD}(\rho, \varphi) = \sum_{r=0}^{R} \sum_{\theta_i=0}^{T} f(r,\theta_i)e^{j2\pi(\rho r + \varphi \theta_i)} \quad (6)$$

The parameters $\rho$ and $\varphi$ reflect the image size, $\theta_i = i2\pi/T, \; 0 \leq \varphi < T$, $R$ is radial resolution, and $T$ is angular resolution.

The GFD method is invariant to translation. However, to achieve rotation and scaling invariance, a normalization step is applied to the extracted feature vector set, as in the following equation:

$$\text{GFD} = \left\{ \frac{|\text{FD}(0,0)|}{\text{area}}, \ldots, \frac{|\text{FD}(0,n)|}{\text{FD}(0,0)}, \ldots, \frac{|\text{FD}(m,n)|}{\text{FD}(0,0)} \right\} \quad (7)$$

where $m$ is the maximum number of radius frequencies, and $n$ is the maximum number of angular frequencies. Zhang and Lu\textsuperscript{44} indicate the efficient shape descriptor using GFDs for shape representation is 52 where radial frequencies $m = 3$ and angular frequencies $n = 12$. We refer to the GFDs collectively as the detected signature $\tilde{S}$.

**Target recognition.** For target recognition, the extracted FDs are compared against the preconfigured target signature. This aim can be achieved using a similarity function $\Delta$ that measures the distance between the detected signature ($\tilde{S}$) and the reference signature ($S$). The similarity function is associated with a threshold $(T)$ that indicates the level of similarity between the compared signatures. If the difference is less than the threshold $(T)$, the detected object is declared as the target, and the user is notified of target recognition. Otherwise, the detected object is ignored, and the user is notified that the target is not detected.

Several ranking and distance measuring functions have been presented in the literature,\textsuperscript{9,26,45} which can be applied to evaluate the similarity between the compared signatures. However, the selected similarity function should be low complexity to avoid increasing the overhead of local target detection.
(Objects), we extracted the GFDs representing the signature of that class. The “Results and discussion” section provides more details on the multiclass signature extraction and the experimental dataset.

The basic idea is to compare the detected object signature ($S$) against a set of possible reference signatures (Objects). This comparison is based on Euclidean distance ($ED$), which is a lightweight computational complexity statistical measurement evaluated as seen in equation (8)

$$ED = \left( \sum_{i=1}^{N} (X_i - \bar{X}_i)^2 \right)^{0.5}$$  \hspace{1cm} (8)

where $N$ represents the total vector set, $X_i$ denotes the $i$th feature vector of the extracted signature, and $\bar{X}_i$ denotes the $i$th feature vector of the reference.

End-user notification. When the sensor identifies the target, it notifies the end-user. Notification is undertaken according to the requirements of the remote user’s applications. Thus, this step represents the potential for saving considerable time and energy consumption by sending a single-byte message, a set of feature vectors, or functional extracted blocks from the image. The end-users’ on-demand notification requests reduce bandwidth congestion by minimizing the volume of transmitted data and the need for retransmission in case of error. This approach leads to a lighter traffic load, thus prolonging the life of the entire network.

**Distributed processing cluster design**

In the context of wireless sensor networks, cluster-based processing has become an attractive method for efficient data processing\(^{[18–21,46]}\) that preserves the resident energy in each participating node and consequently extends the whole network lifetime.

A distributed image-based target detection is proposed, in which the processing load is balanced across a set of nodes in a processing cluster to reduce the per-node energy consumption and, consequently, extend the network lifetime. The distributed implementation will use collective network synergy to achieve better performance competed to a centralized implementation. In this design, both the network model and the energy consumption model are presented.

Network model. This work inspires the processing cluster design by the Low-Energy Adaptive Clustering Hierarchy-cluster based (LEACH-C) protocol\(^{[47]}\) with adaptation distributed processing requirements. Our goal is to build a scalable and energy-efficient distributed processing cluster for image-based target recognition. The proposed research work is concentrating basically on giving proof that the distributed execution of the image-based recognition scheme extends the network lifetime. For the network architecture and the nodes distribution, we assume the following:

- The network is composed of a set of camera nodes, each surrounded by sensor nodes that might participate in the distributed processing when they are selected and could also participate in the collection of other types of data.
- Each camera node has its own angle of view that is selected to avoid overlapping in the field of view with other neighbor camera nodes.
- The network’s density is high enough to ensure that each camera node has a considerable number of nodes that could be involved in the distributed processing.
- The network consists of static wireless camera sensors and conventional static sensors used for processing and communication tasks.
- There is only one camera node in each processing cluster.
- The camera node is responsible for processing cluster setup. The selection will depend mainly on the highest resident energy level of each selected collaborated node.
- Depending on the nature of the application, the communication between the sink node and the access point to the Internet Protocol (IP) network might require a wide range radio link to ensure connectivity. We think that for this purpose, the connection through low-power radio link using Low-power wide area network (LPWAN) technology such as LoRa or SigFox will be a practical solution\(^{[48]}\).

The environment in which the network will be deployed has low dynamicity. The proposed system could be deployed for a wide range of applications such as monitoring the natural habitat of wild animals, fire detection in forests, and red palm weevil detection in agriculture, and it could also be used in land border monitoring.

The distributed processing cluster scheme is iterative, where each iteration consists of two phases: (1) A cluster is formed from the camera node and selected candidates, and (2) processing tasks are distributed across the cluster to accomplish target recognition. The formation of a processing cluster is based on selecting the nodes with resident energy levels.

The scenario, depicted in Figure 4, is setting up a single processing cluster as follows:

- The camera node initiates the cluster forming request by sending a broadcast of the [ENERGY_REQUEST] packet.
- A set of processing nodes within the camera’s neighborhood will reply with their residual energy...
levels using the [ENERGY_RESPONSE] packet. The camera maintains a list of possible candidates.

- The camera node selects the two candidate nodes of the processing clusters P1 and P2, based on the highest resident energy level.
- The camera node will assign a single task for each participating node by communicating through the [JOIN] packet.
- P1 and P2 send an acknowledgment if they are not busy using the [FORM] packet. Otherwise, go back to step 1.
- After forming processing clusters, the camera node starts capturing the observed scene periodically. If there is a detected object, the camera will isolate the functional ROI and send it to the first node for further processing through the [ROI] packet.
- P1 and P2 will work together on object feature extracting, matching, and notifying steps, where P1 is responsible for receiving and extracting the GFD feature vector set, and P2 will accomplish the matching and notification step. Once the object is detected, the P2 responsible for the matching process will notify the camera.
- The camera, in its turn, will send the end-user a notification when the detected object matches the target. Otherwise, the detected object is discarded, and the end-user is updated with a message of “No Target is Found.”

At the end of this epoch of communication and processing, the processing cluster’s current cycle is terminated, and the cooperative nodes are set free.

Finally, the camera is ready to form the subsequent sensing and distributed processing cycle through a new cluster. In our work, we used the packet structure defined in the IEEE 802.15.4 standard to exchange data and to control the setup of the cluster formation and collaboration of the nodes. In depth, the communication part is designed to rely on the payload field without modifying the standard IEEE 802.15.4 packet header structure. This advantage gives the scheme the flexibility to design 10 different control messages and four data exchange messages using the structure of payload fields, as illustrated in Figure 5.

We design the communication control requests to be exchanged between the nodes to ensure the setup of the processing cluster. These requests allow to set up the processing cluster and to assign the functions to the selected nodes at the early configuration step during network setup.

When the processing cluster is formed, valuable data are exchanged between the camera nodes and the other nodes to ensure the sensing scheme’s different steps, as described in Tables 1 and 2.

Energy consumption model. In this article, we adopt the energy consumption model used in LEACH,8 as illustrated in the following equations

\[
E_{tx}(l, d) = \begin{cases} 
1 \times E_{elec} + l \times E_{mp} \times d^4, & \text{if } d \geq d_0 \\
1 \times E_{elec} + l \times E_{fs} \times d^2, & \text{if } d < d_0
\end{cases}
\]  

where \(E_{elec}\) is the energy consumed by the circuit per bit; \(d\) is the distance between sender and receiver; \(E_{fs}\) relates to free space energy depleted by the amplifier.
for a short distance, while $E_{mp}$ corresponds to multipath fading energy that is depleted by the amplifier and long distances. $d_0 = \sqrt{E_{fs}/E_{mp}}$ is the reference distance between sender and receiver. If this distance is less than $d_0$, then $E_{fs}$ is turned on; otherwise, $E_{mp}$ is demanded.

The energy consumption of a sensor node when it receives a $k$-bit packet is as follows

$$E_{rx} = k \times E_{elec} + k \times n \times E_{DA}$$

(10)

where $E_{DA}$ is the needed energy to aggregate data, $k$ is the number of bits per packet, and $n$ is the number of received messages.

**Results and discussion**

In this study, we are mainly interested in evaluating the energy consumption at the sensor’s level during recognition and notification when considering various scenarios.

**Experiment setup and parameters**

In this experiment, we assume a network area of 100 m $\times$ 100 m. The camera node is in the center of the area at position (50,50). The sink node is located at position (0,0). $N$ sensor nodes are scattered in random positions.
For this experiment, we assume $N = 10$. The MATLAB and AVRORA simulators are used to evaluate the energy consumed. AVRORA is a sensor emulator that evaluates the energy consumption level associated with the internal algorithm processing (TinyOS). MATLAB can simulate the communication between the sensor node and the sink. Table 3 lists the sensor specification. The AVRORA tool was used to study the energy consumption of the proposed scheme by estimating the per-node energy consumption for the assigned tasks and the transmission time for a set of sensor nodes like the Mica2 and TelosB sensors. The overhead related to cluster formation and communication between cooperating nodes was estimated using MATLAB simulation based on the description given in the “Energy consumption efficiency analysis.”

### Table 3. Sensor specification.

| Criteria                              | Description |
|---------------------------------------|-------------|
| Mote series                           | Mica2       |
| Sensor processor                      | ATmega128L, 868/916 MHz |
| Measurement flash                     | 512K bytes  |
| Program flash memory                  | 128K bytes  |
| Sensor data rate                      | 38.4 KBAud = 20/40 kbps |
| Network communication model           | Based on signal strength |
| Initial energy                        | 100 mJ      |
| Electric consumption energy (RX, TX)  | 5*E-5 mJ/bit |
| Transmit amplifier Energy consumption per bit in the free space model (Efs) | 1*E-8 mJ/bit/m² |
| Transmit amplifier Energy consumption per bit in the multi-path model (EMP) | 1.3*E-12 mJ/bit/m⁴ |
| Data aggregation energy (EDA)         | 5*E-9 mJ/bit/signal |
| Square root (Efs/EMP) d₀              | 8.7705 m    |
| RX: Receive; TX: Transmit.            |             |

Target recognition and performance analysis

In this section, the performance of the proposed scheme is estimated for single object detection in Mica2 sensors using images of size $(64 \times 64$ pixels 8 bpp) and $(128 \times 128$ pixels 8 bpp). To prove the efficiency and accuracy of the selected shape descriptor and attest to the ability to satisfy the recognition and identification capabilities, we implement the GFD algorithm using MATLAB.

In literature, different shape descriptor techniques such as GFD and Zernike Moments (ZM) were tested using MPEG-7 datasets. However, these datasets contain solid grayscale objects and binary animals’ shapes. However, these datasets are unfit for testing the presented scheme to recognize the tracked single object applications such as rare animal tracking.

For testing, we implement our dataset composed of six different animal classes using an 8-bitmap grayscale image with 64 by 64 pixels and 128 by 128 pixels. The images were designed to include different animal movements and different degrees of rotation, scaling, and translation for invariant testing with 168 images. Starting from the reference image, we generated a total of 28 images in each class. Each class set was divided into 15% reference images, 60% training images, and 25% testing and validating images (see Appendix 1).

We apply the GFD shape descriptor technique to our dataset to extract a set of 52-image feature vector descriptors. We chose four radiance frequencies and nine angular frequencies based on the recommendation in Zhang and Lu. The cumulative results of the extracted vectors are presented in Figure 6.

We infer from Figure 5 the correlation of the extracted features despite the changes in the animal posture, rotation, scaling, and translation level.

The GFD feature vectors demonstrate excellent identification results with almost identical feature vectors despite the shape’s invariant features. According to this result, our proposed scheme presents a robust and accurate shape descriptor for recognizing and identifying an object. It presents a high ability for capturing significant features of the sensed object. For better recognition performance to differentiate the appearing object from a native class from other possible classes, we extend the GFD for recognition ability by using a metric called MED (minimum Euclidean distance) to find the best discrimination threshold as follows

$$MED = \text{Min}(\text{other classes } ED) - \text{Max}(\text{native class } ED)$$

We use this metric to calculate different possible classification thresholds based on an empirical study on the dataset. We first calculate the minimum and maximum ED for each training dataset compared to the reference images in this work. Then we compute the MED values compared to other classes.

From this experiment, we obtained different five thresholds, and to attest the efficiency of the chosen
threshold value, and we defined classification efficiency and retrieval performance as follows

\[
\text{Classification efficiency} = \frac{m - n}{m} \times 100
\]

where \(m\) is the total number of classified images, and \(n\) is the total number of misclassified images.

The retrieval performance will be measured as follows

\[
P = \frac{r}{n} \times 100
\]

where \(P\) is the precision, \(r\) is the number of retrieved objects, and \(n\) is the total number of misclassified images.

\[
R = \frac{r}{m} \times 100
\]

where \(R\) is the recall efficiency, \(r\) is the number of retrieved objects, and \(m\) is the total number of relevant objects in the whole database.

The experiment calculated the obtained threshold values and tested them to select the best value for recognition capabilities, as illustrated below. In Table 4, each threshold value produced different classification and retrieval metrics.

As shown in Table 4, the classification efficiency guarantees the highest performance even if we choose the lowest threshold, while the retrieval efficiency shows the lowest precision percentage when choosing the highest threshold. It performs better as we decrease it while recognizing images gives a good performance under different thresholds.

Moreover, we need to keep the chosen threshold as a trade-off between classification efficiency and accurate retrieval capability.

We conclude from Table 4 that we can use optimal threshold values equal to 0.165. The chosen threshold ensures the classification efficiency while maintaining a modest retrieval performance where all the classification efficiency and retrieval performance are above 95%.

---

Figure 6. Extracted feature set using GFD for the six classes of animals (a) Horse dataset, (b) Wolves dataset, (c) Deers dataset, (d) Elephants dataset, (e) Rhino dataset, and (f) Tigers dataset.
Energy consumption efficiency analysis

Our proposed scheme consists of two main phases: (1) cluster forming phase and (2) the processing of the distributed tasks among the established cluster. To define the optimal number of neighbor nodes that are supposed to participate in the processing cluster, we decomposed the scheme into a set of subtasks. Then, we used AVRORA and MATLAB simulators to quantify each task’s time and energy consumption for only a single sensing cycle.

To analyze the energy consumption of the proposed scheme, we have used the emulator of sensor nodes AVRORA tool that makes it possible to estimate the per-node energy consumption for the assigned tasks as well as the time for sending data of a set of sensor nodes such as MICA and TelosB sensors. The communication between the different nodes of the cluster was estimated using MATLAB simulation based on the description provided in the “Methodology” section.

Based on the results of these experimentations, we divided the scheme tasks into the following atomic processing units: (1) extracting the ROI, (2) extracting the image features’ vector set, and (3) matching obtained vectors by referring to the reference vectors. This subdivision is inevitable due to its specific mathematical calculation nature. The energy consumption related to internal task processing is calculated using AVRORA, as presented in the following tables.

Table 5 shows the time and energy consumption related to the setup of the processing cluster phase. The

### Table 4. Evaluation of GFD recognition capabilities under different values of thresholds.

|                | Classified relevant images, r | Misclassified relevant images, n | Classified irrelevant images | Classified but irrelevant images | Total classified | Classification efficiency | Retrieval performance | Precision | Recall |
|----------------|-------------------------------|----------------------------------|-------------------------------|----------------------------------|------------------|--------------------------|-----------------------|-----------|--------|
|                |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Thr = 0.26     |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Horse          | 68                            | 0                                | 77                            | 77                               | 145              | 100                      | 46.8                  | 100       |
| Wolf           | 68                            | 0                                | 129                           | 129                              | 197              | 100                      | 34.5                  | 100       |
| Deer           | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Elephant       | 68                            | 0                                | 59                            | 59                               | 127              | 100                      | 53.5                  | 100       |
| Rhino          | 68                            | 0                                | 138                           | 138                              | 206              | 100                      | 33.0                  | 100       |
| Tiger          | 68                            | 0                                | 79                            | 79                               | 147              | 100                      | 46.2                  | 100       |
| Thr = 0.19     |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Horse          | 68                            | 0                                | 9                             | 9                                | 77               | 100                      | 88.3                  | 100       |
| Wolf           | 68                            | 0                                | 14                            | 14                               | 82               | 100                      | 82.9                  | 100       |
| Deer           | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Elephant       | 68                            | 0                                | 2                             | 2                                | 70               | 100                      | 97.1                  | 100       |
| Rhino          | 68                            | 0                                | 22                            | 24                               | 92               | 100                      | 73.9                  | 100       |
| Tiger          | 68                            | 0                                | 2                             | 2                                | 70               | 100                      | 97.1                  | 100       |
| Thr = 0.17     |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Horse          | 64                            | 4                                | 3                             | 3                                | 67               | 94.0                     | 95.5                  | 94.1      |
| Wolf           | 68                            | 0                                | 1                             | 1                                | 69               | 100                      | 98.5                  | 100       |
| Deer           | 65                            | 3                                | 0                             | 0                                | 65               | 95.3                     | 100                   | 95.5      |
| Elephant       | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Rhino          | 68                            | 0                                | 3                             | 3                                | 71               | 100                      | 95.7                  | 100       |
| Tiger          | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Thr = 0.165    |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Horse          | 64                            | 4                                | 2                             | 2                                | 66               | 93.9                     | 96.9                  | 94.1      |
| Wolf           | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Deer           | 65                            | 3                                | 0                             | 0                                | 65               | 95.3                     | 100                   | 95.5      |
| Elephant       | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Rhino          | 68                            | 0                                | 3                             | 3                                | 71               | 100                      | 95.7                  | 100       |
| Tiger          | 67                            | 1                                | 0                             | 0                                | 67               | 98.5                     | 100                   | 98.5      |
| Thr = 0.155    |                               |                                  |                               |                                  |                  |                          |                       |           |        |
| Horse          | 64                            | 4                                | 0                             | 0                                | 64               | 93.7                     | 100                   | 94.1      |
| Wolf           | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Deer           | 65                            | 3                                | 0                             | 0                                | 65               | 95.3                     | 100                   | 95.5      |
| Elephant       | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Rhino          | 68                            | 0                                | 0                             | 0                                | 68               | 100                      | 100                   | 100       |
| Tiger          | 65                            | 3                                | 0                             | 0                                | 65               | 95.3                     | 100                   | 95.5      |
obtained results do not include the energy consumed to capture the image by the camera node. As shown, the camera consumes more energy and takes more time than any other node due to its leading role in the clustering phase. We measured the energy consumption associated with the ROI extraction from the image set. Table 6 sums up the energy consumption for this task regarding the occupancy of the ROI in the captured image.

The results reported in Tables 6 to 8 show how the size of the ROI impacts the scheme’s efficiency on the camera and collaborated nodes in terms of processing time and energy consumption during different steps of the distributed scheme execution. We can note that for low occupancy of the object in the acquired image (small size of ROI), the energy consumption and the processing time are balanced among the nodes of the processing cluster. Consequently, the per-node lifetime is expected to be prolonged. For high ROI size, the processing in the camera node requires high-energy consumption, which reduces its lifetime considerably.

Table 9 shows the essential energy estimation for shape feature extraction based on GFD for object identification and recognition.

We infer from the obtained results that the proposed scheme could distribute the processing load between the camera node and the two selected evenly collaborated nodes in the processing cluster. This approach releases

### Table 5. Per-node energy and time consumption in forming cluster phase.

| Cluster forming task | Camera | Single neighbor node |
|----------------------|--------|----------------------|
|                      | Time (s) | Energy (%) | Time (s) | Energy (%) |
| Neighborhoods energy request | 0.000025 | 0.005 | 0 | 0.006 |
| Neighbors energy responses | 0 | 0.055 | 0.000025 | 0.005 |
| Cluster forming and acknowledgment | 0.00005 | 0.12 | 0.000025 | 0.011 |
| Total | 0.015 | 0.18 | 0.00005 | 0.021 |

### Table 6. Extracted ROI from images dataset size.

| Percentage of occupancy of the ROI in the image (64 × 64 bpp) | Percentage of occupancy of the ROI in the image (128 × 128 bpp) | Extracted ROI (bits) | Energy consumption in TX (%) | Energy consumption in RX (%) |
|---------------------------------------------------------------|---------------------------------------------------------------|----------------------|-----------------------------|-----------------------------|
| 30%                                                           | 7%                                                           | 1228                 | 0.0617472                   | 0.067584                    |
| 55%                                                           | 13%                                                          | 2252                 | 0.1132032                   | 0.123904                    |
| 65%                                                           | 15%                                                          | 2662                 | 0.1337856                   | 0.146432                    |
| 90%                                                           | 22%                                                          | 3686                 | 0.1857850                   | 0.202752                    |
| –                                                             | 30%                                                          | 4915                 | 0.2469888                   | 0.270336                    |
| –                                                             | 55%                                                          | 9011                 | 0.4528128                   | 0.495616                    |
| –                                                             | 65%                                                          | 10,649               | 0.5351424                   | 0.585728                    |
| –                                                             | 90%                                                          | 14,745               | 0.7415100                   | 0.811008                    |

ROI: region of interest.

### Table 7. Energy and time estimation for object detection and identification for an image size of (64 × 64 pixels 8 bpp).

| Object detection and identification | Camera | P1 |
|-------------------------------------|--------|----|
|                                    | Time (s) | Energy (%) | Time (s) | Energy (%) |
| Camera extracts ROI                | 0.118  | 2.25 | 0 | 0.068 |
| Camera sends ROI to P1 (ROI size%) | 0.000025 | 0.062 | 0 | 0.124 |
| ROI 30%                            | 0.000025 | 0.113 | 0 | 0.146 |
| ROI 55%                            | 0.000025 | 0.134 | 0 | 0.207 |
| ROI 65%                            | 0.000025 | 0.247 | 0 | 0.152 |
| ROI 90%                            | 0.000025 | 2.389 | 0 | 0.152 |

ROI: region of interest.
the camera from the processing. The results also show that energy depletion in the nodes of the processing clusters P1 and P2 is not proportional to the ROI size in the acquired image. When the target is recognized, the scheme will notify the end-user with different possible message types. Tables 10 and 11 illustrate the energy consumption based on the selected notification option.

As summarized in Table 12, the energy consumption related to the processing of the scheme is shared between the different nodes. The camera node consumes around 24% of energy during a single sensing cycle, while the cooperative nodes consume 76% of energy from the total consumed energy required to process the scheme using an image size of 64 by 64.

For an image of \((128 \times 128\) pixels 8 bpp), these percentages of energy consumption could reach 52% in the camera node and 48% in collaborated nodes. The camera selects candidate nodes for cluster participation based on the highest residual energy in each new sensing cycle. This step aims to distribute the processing load over the cluster’s nodes, which consequently extends the node’s lifetime.

In Figures 7 and 8, we plotted the cumulative energy consumption in the collaborative nodes during multiple sensing cycles while executing the proposed distributed processing scheme.

From Figure 7(a), we note that the total energy consumed in the network does not exceed 30% of the total network energy using images with a size of \((64 \times 64\) pixels 8 bpp). When we use images with a size of \((128 \times 128\) pixels 8 bpp) (Figure 7(b)), the total consumed energy in the network after 10 sensing cycles is around 10%, and this is because the camera sensor has consumed all the available energy and cannot initiate the processing scheme to process more acquired images.

In Figure 8, we show that the battery of the camera node is exhausted while the nodes of the network are still alive. These results attest to the importance of applying

Table 8. Energy and time estimation for object detection and identification for an image size of \((128 \times 128\) pixels 8 bpp).

| Object detection and identification | Camera | P1 |
|-------------------------------------|--------|----|
| Time (s) | Energy (%) | Time (s) | Energy (%) |
| Camera extracts ROI | 0.39 | 9.06 | 0 | 0.146 |
| Camera sends ROI to P1 (ROI size %) | ROI 15% | 0.000025 | 0.134 | 0 | 0.203 |
| | ROI 22% | 0.000025 | 0.185 | 0 | 0.270 |
| | ROI 30% | 0.000025 | 0.247 | 0 | 0.496 |
| | ROI 55% | 0.000025 | 0.453 | 0 | 0.585 |
| | ROI 65% | 0.000025 | 0.535 | 0 | 0.811 |
| | ROI 90% | 0.000025 | 0.741 | 0 | 0.84 |
| Total (average) | 0.39 | 9.442 | 0 | 0.471 |

Table 9. Energy and time estimation for feature vectors’ extraction using GFD.

| Feature vectors’ extraction using GFD | P1 | P2 |
|--------------------------------------|----|----|
| Time (s) | Energy (%) | Time (s) | Energy (%) |
| P1 extracts GFD vectors | 0.131 | 2.94 | 0.21 | 4.8 |
| P1 sends GFD vectors to P2 | 0.000025 | 0.056 | 0.21 | 4.862 |
| P2 matching the vectors to a reference | | | |
| Total | 0.131 | 2.996 | 0.21 | 4.862 |

GFD: general Fourier descriptor.

Table 10. Energy and time consumption for simple notification.

| Simple notification 1 byte | P2 | Camera |
|----------------------------|----|--------|
| Time (s) | Energy (%) | Time (s) | Energy (%) |
| P2 sends notification to the camera node | 0.000025 | 0.005 | 0.000025 | 0.011 |
| The camera sends a notification to the end-user | | | |
| Total | 0.000025 | 0.005 | 0.000025 | 0.011 |
the distributed approach of the target detection scheme. While this result concludes that the camera node is the failure point of this application, it demonstrates that selecting the nodes to be part of the processing cluster was reasonably performed, resulting in a good distribution of the energy consumption over the network.

Figure 9 plots the percentage of energy consumption in camera and collaborative nodes during the first 10 sensing cycles. We can see that each node, excluding the camera, is selected at most two times during the 10 rounds. This result shows that the proposed scheme allows invoking the nodes to ensure an equilibrium of the consumed energy among the network nodes. Figure 10 presents the cumulative energy consumption level in all network nodes during the first 10 sensory cycles. We find that the camera maintains a constant level of small consumption percentage compared to other collaborative nodes. Simultaneously, the rest of the processing nodes consume different energy levels depending on their number of invocations and their role during each processing cycle. However, the significant ROI extracted in images of 128 x 128 explains the high-energy consumption in the camera node. Conversely, using image size 128 by 128, where a significant ROI is possibly extracted, will lead to high camera energy depletion before any processing nodes and an extra co-node are demanded to maintain the network lifetime.

We suppose that the participating nodes do not execute any simultaneous tasks. We also suppose that the minimum number of nodes available in the network around the camera sensor and could participate in a processing cycle is three. In every processing cycle, only two nodes selected based on the highest residual energy level will be part of the processing cluster, allowing a fair distribution of loads for every node in the network. In the network, reducing the number of possible collaborating nodes will result in a greater chance of their selection for participation in a more significant number of subsequent sensing cycles, which will increase their power consumption faster than if they are distributed evenly. For example, using only two collaborated nodes will drain the network node’s energy level before depleting the camera. As illustrated in Figure 11(a), the scheme runs up to 25 sensing cycles while the camera still has 40% of the energy level. On the contrary, the camera is exhausted when dealing with a higher percentage of occupancy of ROI, especially in 128 by 128 image size, and the scheme will stop after only 11 sensing cycles, as shown in Figure 11(b).

Figure 12 shows the cumulative energy consumption in all 10 network nodes that are expected to participate in the processing clusters formed during all sensory cycles. The figure shows that the energy consumption was distributed over the network nodes in a balanced manner. The energy consumption in the camera node is higher than the energy consumption in the other nodes because of the heavy processing task assigned to this node. Its performance shows a gradual decline in the available level of energy until the battery is exhausted. Assigning tasks to other network nodes helps the camera perform more sensing cycles, helping to ensure an extension of its lifetime.

When comparing the algorithm’s performance to the centralized processing approach, we find that the energy consumption can reach 9.995 mJ in image (64 by 64-8 bpp) and 16.8 mJ in the image (128 by 128-8 bpp). In a distributed implementation of the presented scheme, the energy consumption in the camera node decreased to 2.39 mJ based on ROI size up to 2.46 mJ (approximately from 76.1% to 75.32% reduction) in images (64 by 64-8 bpp). A load of energy consumption for 24% in the camera and 76% in collaborated processing nodes is shown in Table 12. Each node, excluding the camera, is selected two times during the first 10 sensing cycles in the experiment. So, minimizing the number of candidates collaborating with processing nodes will increase the probability of participating more in sensing processing load.

### Table 11. Energy consumption for other notification options.

| Other notification options | Notification options | P2 | Camera |
|----------------------------|----------------------|----|--------|
|                            | Time (s) | Energy (%) | Time (s) | Energy (%) |
| P2 sends notification to the camera node | GFD feature vectors | 0.000025 | 0.062 | 0 | 0.068 |
| The camera sends a notification to the end-user | GFD feature vectors | 0.000025 | 0.062 |
|                          | ROI 15%   | 0.000025 | 0.134 |
|                          | ROI 22%   | 0.000025 | 0.185 |
|                          | ROI 30%   | 0.000025 | 0.247 |
|                          | ROI 55%   | 0.000025 | 0.453 |
|                          | ROI 65%   | 0.000025 | 0.535 |
|                          | ROI 90%   | 0.000025 | 0.741 |

GFD: general Fourier descriptor; ROI: region of interest.
While the camera will be released to 2.39 up to 9.89 mJ (approximately 86%–41.1% reduction) in images (128 by 128 pixels 8 bpp), this energy gain will prolong the camera life and improve network performance optimally improved. We highlight the system’s performance through the life cycle of the network and the extent to which it is affected by the image’s size (see Figure 12). This scheme promises to prolong the

---

**Figure 7.** Cumulative consumed energy in the nodes of the network for (a) images (64 × 64 pixels 8 bpp) and (b) images (128 × 128 pixels 8 bpp).

**Figure 8.** Residual energy in-camera node versus average residual energy in-network for (a) images of (64 × 64 pixels 8 bpp) and (b) images of (128 × 128 pixels 8 bpp).

**Figure 9.** Energy consumption in the nodes of the network after 10 sensing cycles for (a) images size of (64 × 64 pixels 8 bpp) and (b) images size of (128 × 128 pixels 8 bpp).
network life where we reach 44 sensing cycles in distributed processing compared to only 10 sensing cycles in a centralized approach (see Figure 13).

We compared the efficiency of the proposed scheme with similar research approaches designed for energy-efficient multimedia sensing. For this comparison, it is essential to state that these approaches could be classified into different categories. In depth, some research work used the approach of image compression to reduce the amount of transmitted data through the network and save energy. Banerjee and Das Bit6 have proposed an energy-aware scheme intended for image transmission in WMSN. Their approach ensures a low overhead data compression for energy saving based on the curve fitting technique. The obtained results demonstrated energy efficiency compared to other similar data compression algorithms.

Kouadria et al.16 applied an ROI-based image compression using the DTT. The DTT compression technique is an alternative to the DCT due to its low complexity and good energy consumption. However, the experimental results show that it consumes around 146.63 mJ per block of (8 × 8) pixels.

An approach of distributed compression algorithm was proposed in Zuo et al.31 It was noted that the authors’ approach consumes around 1.4 J for the compression of an image size of (512 × 512 pixels 8 bpp) which is considered as an extremely high-energy consumption processing, and it may flood the network with irrelevant data. In the same context, Nikolakopoulos et al.33 presented a compression scheme based on quadtree decomposition. The obtained results showed that it consumed 120 mJ energy to transmit an image of (128 × 128 pixels 8 bpp), and 45 mJ was required to similarly transmit an image of (64 × 64 pixels 8 bpp).

In conclusion, multimedia sensing based on image compression approaches consumes high energy compared to our presented approach. The application of a
As a solution to reduce the high-energy consumption related to the software implementation of compression algorithms, another approach based on hardware implementation was proposed. In Chefi et al.\textsuperscript{32} a design of a hardware platform was presented to reduce the consumed energy. Although the hardware implementation increases the cost, it ensures a significant gain in energy. However, this solution is not suitable for low-cost devices. We have shown that we can identify and recognize the phenomenon of interest without changing the sensor design compared with our proposed scheme.

Some other similar approaches studied the possibility of energy saving based on event-based multimedia sensing. The proposed scheme presents attractive characteristics in energy consumption. Alhilal et al.\textsuperscript{27} presented a centralized processing scheme based on the centroid distance and histogram as object descriptors. The results showed that their proposed scheme consumes 47.6 mJ for an image size of \((64 \times 64\) pixels \(8\) bpp) and 80.2 mJ for an image size of \((128 \times 128\) pixels \(8\) bpp). Besides the high computational complexity of our scheme, the centroid distance is not an accurate descriptor for target recognition. In-depth results demonstrated sensitivity to the characteristics of the detected object in the image.\textsuperscript{42} Our previous work published in Alsabhan and Soudani\textsuperscript{28} proposed a processing scheme for energy efficiency based on GFDs as a descriptor of the object’s recognition process. We provided a comparison with the deployment of ZM, showing a significant energy saving while using GFD descriptors. However, the proposed scheme was implemented in a centralized manner on the camera node. The centralized processing approach of the scheme required around 9.995 mJ for image size \((64 \times 64\) pixels \(8\) bpp) and 16.8 mJ for image size \((128 \times 128\) pixels \(8\) bpp) using GFD and 22 mJ using ZM. Applying GFD reduced the need to image preprocessing since GFD is scale, translation, and rotation invariant while ZM is invariant to rotation only. Zam et al.\textsuperscript{23} present a novel energy-aware face-detection algorithm for extracting a lightweight discriminative vector of detected face-sequence to be sent to sink with low transmission cost and high-security level. However, the efficient face recognition algorithm has been performed at the sink. The total in-node energy consumption is 2.7 J, and the in-network energy consumption is 5.1 J, so in total, it consumes around 7.8 J.

Compared to the previously mentioned work about event-based sensing, the new approach of distributed implementation demonstrated that the processing load

| Image size | Measured attribute | Camera | P1 | P2 | Camera | P1 | P2 |
|------------|--------------------|--------|----|----|--------|----|----|
| \(64 \times 64\) pixels \(8\) bpp | Cluster forming task | 0.015 | 0.00005 | 0.015 | 0.00005 |
| | Object detection and identification | 0.118 | 2.389 | 0.152 | 0.152 |
| | Feature vectors extraction using GFD | – | 0.131 | – | 0.131 |
| | Simple notification | 0.000235 | 0.011 | 0.000235 | 0.011 |
| Total | Energy % | 0.133 | 25.2% | 0.131 | 29.8% |

GFD: general Fourier descriptor.
of the camera sensor was reduced. Other tasks were migrated to other nodes, which extends the viability of the application since the camera node will execute more sensing cycles. The performance evaluation of the presented scheme shows that our work outperforms similar event-based schemes in terms of ultra-low-energy consumption associated with the clustering and in-cluster communication, extending the camera lifetime and consequently extending the multimedia application lifetime in the wireless sensor network.

The presented solutions in Bouacheria et al., Bidai compete in terms of energy efficiency and QoS assurance based on providing routing protocol solutions. Authors presented novel approaches to reach the end-user with reliable and energy-efficient multimedia delivery. These presented works provide the best solutions for applications that favor continuous data streaming delivery, such as environment live broadcast. However, in some applications of WMSN, end-users are interested in being notified when an event of interest happens using a short message that declares the object’s appearance or this notification can be promoted to have the ROI instead of having the whole captured image. Moreover, for some applications, end-users prefer to allow the network to take decisions locally for what action can be done, such as triggering the alarm when fire smoke is detected or tracking a moving object from zone to zone on the same monitoring area instead of waiting for the end-users’ response. Besides, we think that our solution provides an interesting solution to release the networks from excess data exchange by either transmitting a notification based on the end-users’ interest or minimizing the interference of end-user and deciding locally if the identified object is declared as a possible target or not. In comparison, our presented approach will release the networks from 38 J to a low computational complexity between 2.5 and 10 mJ.

In conclusion, we prove the energy efficiency of the presented work and demonstrate how it outperforms in terms of accuracy and efficiency to similar solutions presented in the literature of image detection and recognition in WMSN. Table 13 summarizes the presented comparison.

**Conclusion**

Reducing the per-node processing load appears to be a fitting suggestion to extend the network lifetime and achieve adequate performance. The idea of distributed processing of a sensing scheme over a set of nodes as a processing cluster would balance the processing over these nodes and would reduce per-node energy consumption during one sensing cycle. This work focused on the specification and design of a low-energy processing scheme intended for distributed cluster-based...
implementation. The idea is to balance the processing load through a set of nodes that form the processing cluster set to extend the network lifetime. In this work, we present our experimental results of the distributed implementation of a proposed target detection scheme based on a multimedia sensor, a wireless sensor network. The processing load is decreased from the camera to only 2.4 mJ in the distributed processing cluster (approximately 75.32% reduction using image size \(64 \times 64\) pixels 8 bpp) and 9.8 mJ (approximately 41.1% reduction using image size \(128 \times 128\) pixels 8 bpp) compared to the centralized processing paradigm. The scheme’s performance evaluation showed that the distributed approach offers an ultra-low-energy consumption associated with clustering. The energy consumption due to processing is divided among the network, approximately 24% in the camera and 76% in cooperative nodes based on the previously mentioned energy measurement. So, using 10 collaborated nodes will distribute 76% among them and lower the loss of energy in collaborated nodes to 15.2% only per cycle. We believe that the distributed approach results significantly extend the life of the camera node and testify to multimedia sensing efficiency. In future work, we will attest the performance with an environment with multiple objects’ appearances. Also, we will examine the recognition capabilities and retrieval efficiency using other signature feature extraction methods such as wavelet transformation.

**Acknowledgements**

The authors thank the Deanship of Scientific Research in King Saud University for funding and supporting this research through the initiative of Deanship of Scientific Research (DSR) Graduate Students Research Support (GSR).

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported and funded by the Deanship of Scientific Research in King Saud University through the initiative of Deanship of Scientific Research (DSR) Graduate Students Research Support (GSR).

**ORCID iD**

Manal Alsabhan https://orcid.org/0000-0001-5968-971X

**References**

1. Alvi SA, Afzal B, Shah GA, et al. Internet of Multimedia Things: vision and challenges. *Ad Hoc Netw* 2015; 33: 87–111.
2. Nauman A, Qadri YA, Amjad M, et al. Multimedia Internet of Things: a comprehensive survey. *IEEE Access* 2020; 8: 8202–8250.
3. Yadav V and Arora G. Analysis of wireless multimedia sensor network. In: Proceedings of the 2019 2nd international conference on power energy, environment and intelligent control (PEEIC), Greater Noida, India, 18–19 October 2019, pp.496–498. New York: IEEE.

4. Misra S, Reisslein M and Xue G. A survey of multimedia streaming in wireless sensor networks. IEEE Commun Surv Tut 2008; 10(4): 18–39.

5. Chiwariro R and Thangadurai N. Quality of service aware routing protocols in wireless multimedia sensor networks: survey. Int J Inform Technol 2022; 14: 789–800.

6. Banerjee R and Das Bit S. An energy efficient image compression scheme for wireless multimedia sensor network using curve fitting technique. Wirel Netw 2019; 25(1): 167–183.

7. Latreche B, Saadi S, Kious M, et al. A novel hybrid image fusion method based on integer lifting wavelet and discrete cosine transformer for visual sensor networks. Multimed Tool Appl 2019; 78(8): 10865–10887.

8. Radhika M and Sivakumar P. Video traffic analysis over LEACH-GA routing protocol in a WSN. Proced Comput Sci 2019; 165: 701–707.

9. Keyes L and Winstanley AC. Fourier descriptors as a general classification tool for topographic shapes. In: Proceedings of the Irish machine vision and image processing conference, 1999, pp.193–203, https://mural.maynoothuniversity.ie/66/2/AW-Fourier-1999.pdf.

10. Kadduchi ML, Soudani A, Lecuire V, et al. Low power hardware-based image compression solution for wireless camera sensor networks. Comput Stand Interface 2012; 34(1): 14–23.

11. Akyildiz IF, Su W, Sankarasubramaniam Y, et al. Wireless sensor networks: a survey. Comput Netw 2002; 38(4): 393–422.

12. Yick J, Mukherjee B and Ghosal D. Wireless sensor network survey. Comput Netw 2008; 52(12): 2292–2330.

13. Leila M, Lecuire V and Moureraux JM. Camera sensor networks. In: Proceedings of the 2nd international conference on image processing theory, tools, and applications (IPTA), Paris, 7–10 July 2010, pp.126–129. New York: IEEE.
34. Wang H, Peng D, Wang W, et al. Artificial immune system based image pattern recognition in energy efficient wireless multimedia sensor networks. In: Proceedings of the MILCOM 2008: military communications conference, San Diego, CA, 16–19 November 2008, pp.1–7. New York: IEEE.

35. Bouacheria I, Bidai Z, Kechar B, et al. Leveraging multi-instance RPL routing protocol to enhance the video traffic delivery in IoMT. Wirel Pers Commun 2021; 116(4): 2933–2962.

36. Bidai Z. RPL enhancement to support video traffic for IoMT applications. Wirel Pers Commun 2022; 122: 2367–2394.

37. Piccardi M. Background subtraction techniques: a review. In: Proceedings of the 2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583), The Hague, 10–13 October 2004, vol. 4, pp.3099–3104. New York: IEEE.

38. Pham DM and Aziz SM. Object extraction scheme and protocol for energy efficient image communication over wireless sensor networks. Comput Netw 2013; 57(15): 2949–2960.

39. Rehman YAU, Tariq M and Sato T. A novel energy efficient object detection and image transmission approach for wireless multimedia sensor networks. IEEE Sens J 2016; 16(15): 5942–5949.

40. Adams R and Bischof L. Seeded region growing. IEEE Trans Pattern Anal Mach Intell 1994; 16(6): 641–647.

41. Kang WX, Yang QQ and Liang RP. The comparative research on image segmentation algorithms. In: Proceedings of the 2009 first international workshop on education technology and computer science, Wuhan, China, 7–8 March 2009, vol. 2, pp.703–707. New York: IEEE.

42. Yang M, Kpalma K and Ronsin J. A survey of shape feature extraction techniques. Pattern Recogn 2008; 15(7): 43–90.

43. Zhang D and Lu G. Generic Fourier descriptor for shape-based image retrieval. In: Proceedings of the ICME’02: IEEE international conference on multimedia and expo, Lausanne, 26–29 August 2002, vol. 1, pp.425–428. New York: IEEE.

44. Zhang D and Lu G. Shape-based image retrieval using generic Fourier descriptor. Signal Process Image Commun 2002; 17(10): 825–848.

45. Larsson F, Felsberg M and Forssén PE. Linköping University post print patch contour matching by correlating Fourier descriptors. In: Proceedings of the IEEE digital image computing: techniques and applications, 2009, https://www.researchgate.net/publication/221209693_Patch_Co ntour_Matching_by_Correlating_Fourier_Descriptors

46. Sajjanhar U and Mitra P. Distributive energy efficient adaptive clustering protocol for wireless sensor networks. In: Proceedings of the 2007 international conference on mobile data management, Mannheim, 1 May 2007, pp.326–330. New York: IEEE.

47. Heinzelman WR, Chandrakasan A and Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii international conference on system sciences, Maui, HI, 7 January 2000, p.10. New York: IEEE.

48. Abboud S, Rachkidy NE, Guitton A, et al. Gateway selection for downlink communication in LoRaWAN. In: Proceedings of the 2019 IEEE wireless communications and networking conference (WCNC), Marrakesh, Morocco, 15–18 April 2019, pp.1–6. New York: IEEE.

49. Titzer BL, Lee DK and Palsberg J. AVRORA: scalable sensor network simulation with precise timing. In: Proceedings of the IPSN 2005: fourth international symposium on information processing in sensor networks, Boise, ID, 15 April 2005, pp.477–482. New York: IEEE.

50. Yadav RB, Nishchal NK, Gupta AK, et al. Retrieval and classification of shape-based objects using Fourier, generic Fourier, and wavelet-Fourier descriptors technique: a comparative study. Opt Laser Eng 2007; 45(6): 695–708.
Appendix 1. Experimental dataset of images.