IJIAM
International Journal of Informatics
and Applied Mathematics

International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 1, No. 1, 10-23

Efficient Camera Clustering Method Based on Overlapping FoVs for WMSNs

Benrazek Ala-Eddine[1], Farou Brahim[1], and Kurulay Muhammet[2]

[1] LabSTIC, Computer Science Department, 8 Mai 1945 Guelma University, POB 401
Guelma, Algeria
(benrazek.alaeddine, farou.brahim)@univ-guelma.dz

[2] Department of Computer Science, Yildiz Technical University, Istanbul, Turkey
mkurulay@yildiz.edu.tr

Abstract. The paper presents a new method of clustering cameras in a wireless multimedia sensor network based on overlapping camera fields of view. This method aims to group as many overlapping cameras as possible using the Bron-Kerbosch algorithm. The algorithm allows to find a maximum number of cliques, where they represent camera clusters with strongly overlapping fields of view. The main objective of this method is to form clusters of cameras that have a large overlap area between them in order to restrict the communication area only within the cluster. The method also avoids network congestion and reduce the redundancy of detected data to limit the rapid decrease in energy resulting from the acquisition, processing and transmission of redundant multimedia data. The simulation results show that the proposed method is more effective in extending network lifetime and reducing overhead costs.

Keywords: Clustering, WMSN, Field of View, Energy Conservation, Maximal Cliques

1 Introduction

Over the past decade, researchers have paid particular attention to Wireless Multimedia Sensor Networks (WMSNs) to develop a more efficient monitoring system that is adapted to emerging applications for smart city construction and Internet of Things systems [12], [16]. Surveillance systems play a very important role in our lives due to their varied benefits, such as the security of public places, financial places, sensitive places, and prohibited places. Surveillance systems consist of a set of intelligent cameras distributed and powered by batteries. The role of these cameras is to extract visual data captured in the environment and sharing it with other cameras or computer centers for analysis and processing to track or re-identify objects.

In traditional WMSNs, cameras are autonomous. They operate without any knowledge of the distribution or functioning of these neighbors, which leads to redundancy in the recording of the same scenes by multiple overlapping cameras due to the lack of communication and coordination between them. This increases the production of multimedia data, where the management and sharing of this multimedia data requires very high bandwidth and huge storage capacity, as well as an excessive computer centre to process and analyze this data, and all this leads to a rapid decrease in energy consumption. To reduce the amount of redundant data, researchers have implemented a new surveillance system that includes intelligent collaborative cameras. This new system coordinates the operation of the cameras between them through communication in order to avoid recording the same scenes with several cameras. Prabhu Natarajan et al. [19] defined multi-camera collaboration as a mechanism allowing several heterogeneous cameras to capture videos, analyze, share knowledge, calculate and perform collaborative control actions to submit a surveillance service.

According to Prabhu's definition, collaboration in the surveillance camera network means that there is a volume of data shared between the network cameras. When the number of cameras in systems that monitor a large area increases, the number of messages and the amount of shared information or knowledge necessarily increases. Therefore, this can lead to network overload or congestion that ultimately leads to network saturation due to the large number of messages transmitted over the network. In addition, the collaboration algorithms for all system cameras become very complex.

One of the main solutions to overcome these problems is clustering and camera planning. Clustering has several objectives, namely [3]: (i) network scalability, (ii) reduction of power consumption to extend battery and network life, (iii) stabilization of network topology, (iv) reduction of overhead costs.

The Bron-Kerbosch algorithm, presented in [8], is one of the fastest [15] and most efficient [6] algorithms for finding maximum cliques in a G(V,E) graph. The Bron-Kerbosch algorithm is a recursive algorithm based on the backtracking technique to find maximum cliques. In graph theory, a complete sub-graph is called clique, and a maximum clique is a sub-graph in which no vertices can be added without losing the property of the clique [22]. In this study, a network of surveillance cameras can be considered as a $G(E, V)$ graph, where the cameras are simulated by vertices (V) and the arcs between the vertices (E) represent the existence of overlaps between the cameras. A maximum clique or a maximum complete sub-graphic corresponds to a cluster of cameras, which are determined by the Bron-Kerbosch algorithm.

This article presents a new method of clustering cameras in a wireless camera network based on overlapping Field of Views (FoVs). This method consists in grouping as many overlapping cameras as possible using the Bron-Kerbosch algorithm that allows to find maximum cliques, the latter representing the camera clusters. The main objective of this method is to form overlapping groups of cameras with large surfaces in order to restrict the communication area to avoid network congestion and reduce redundancy of detected data, while reducing energy consumption and extending network lifetime.

The remainder of this paper is organized as follows: Section 2 presents the related work. Next, Section 3 describes how to calculate the surface area of overlaps and discusses the proposed clustering method. The results of the clustering method tests are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Works

According to the two state-of-the-art articles [17] and [11], several works in the literature have studied in depth sensor clustering algorithms in wireless sensor networks (WSNs) to develop eco-energy efficient routing protocols to extend the network lifetime. In wireless multimedia sensor networks (WMSNs), Alaei et al. [3], [4], [5], [1] and [2] have proposed several camera clustering algorithms for WMSNs based on overlapping areas between camera FoVs to establish cooperation between clusters that have been formed to detect objects. The objective of this work has been to save energy and increase network lifetime. The same principle was used by [9], where the authors have proposed a hierarchical clustering algorithm based on overlapping FoVs.

In [14], the authors have presented a clustering algorithm based on overlapping regions to reduce redundant video streams and reduce energy waste. The proposed algorithm is applied to overlapping regions resulting from the application of the scene cutting technique.

Shreya Mishra et al. [18] have proposed a method of clustering cameras according to cameras communication radius to improve the coverage of directional sensor networks. The authors modeled the clusters as circles that represent the communication range and selected the first node as the center of the circles. Unfortunately, the cameras did not have a FoV in a circle, which makes this approach inapplicable outside the communication itself.

In addition, Masoud Zarifneshat et al. [24] have proposed a distributed semi-localized clustering scheme where the camera selection process is assigned to their leader in a dynamic and cooperative manner for target tracking.

Danial et al. [10] have presented an algorithm for selecting the minimum number of cameras activated to cover all targets. They took into account the multiple views of the targets to calculate redundancy in WVSN network. This approach

requires computational power at camera level to ensure the recognition mechanism that in most cases fails for multiple reasons such as, high number of targets, environment complexity and performance limits of surveillance cameras.
Furthermore, KyDong Jung et al. [13] have introduced a clustering algorithm called FL-TEEN that uses a fuzzy inference system to improve the adaptability of cluster head selection to improve performance in terms of sensor node lifetime. This semi-automatic approach requires human intervention to generate rules for each environment. To solve the region coverage problem, Selina Sharmin et al. [23] have developed an area coverage system sensitive to network lifetime that uses a clustering mechanism based on the overlap's degree and residual energy levels. This proposal promotes network management in the event of a sensor failure on the monitoring quality gain. In [21], Premlata Sati et al. have presented an automatic FoV rotation mechanism for each camera to maximize the coverage area with the minimum number of cameras in the surveillance area and therefore avoid redundant detection. However, this approach requires the existence of panoramic cameras.

3 Proposed Clustering Method

Each camera has a detection area called FoV in which a camera can detect everything. In WMSNs, cameras' FoVs can overlap, which means that there are common sub-view areas (FoVs) between them and that the overlap of FoVs results in a loss of network energy due to redundant detection of the overlap. To solve this problem, this paper presents a clustering method based on the FoVs overlapping criterion instead of the distance between cameras criterion.
Prior to the introduction of the proposed clustering method, it is important to define some useful concepts that will be used subsequently in this paper:

- *Field of View (FoV)*: The field of view of a camera is defined by the area in which a camera can easily and accurately detect objects covered by the latter. According to [3], the FoV is modeled by the surface of an isosceles triangle (ABC), as shown in Figure 1. The vertex A is considered as the position of the camera; the two other vertices are calculated by Equations (1), (2), (3) and (4).
- *Clusters*: is a subset of cameras with overlapping FoVs. The overlap area between the FoVs of two nodes determines whether they can be in the same cluster according to the proposed clustering algorithm.

$$X_B = X_A + R_s \cdot \cos(\alpha) \tag{1}$$

$$Y_B = Y_A + R_s \cdot \sin(\alpha) \tag{2}$$

$$X_C = X_A + R_s \cdot \cos((\alpha + \theta) \bmod 2\pi) \tag{3}$$

$$Y_C = Y_A + R_s \cdot \sin((\alpha + \theta) \bmod 2\pi) \tag{4}$$

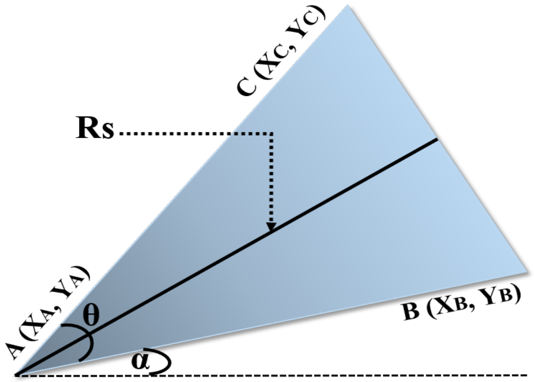


Fig. 1: FoV of camera sensor [2]

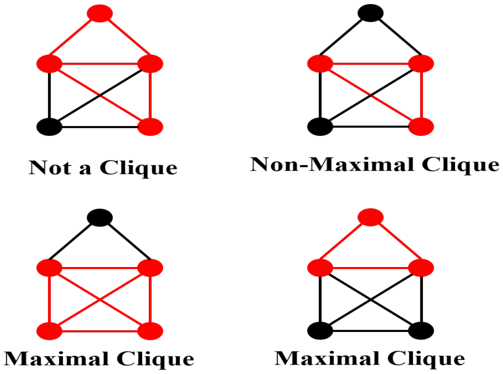


Fig. 2: Example of maximum clique

- *Isolated camera*: a camera is considered isolated if all surfaces that overlap with other cameras are null.
- *Clique*: the clique is a complete subgraph, i.e. a subset of vertices all in a two-to-one relationship.
- *Maximum clique*: is a clique that cannot be extended by including other adjacent vertices, as shown in Figure 2.

The proposed clustering method was implemented within the base station (BS), where all phases were executed only once, just after the cameras were deployed. This method consists of two main steps, as shown in Figure 3, namely: (i) calculate the area of the intersection polygons of the FoV intersection polygons for two cameras overlapping each other, (ii) apply the maximum clique discovery algorithm to cluster the cameras with the largest overlap area.

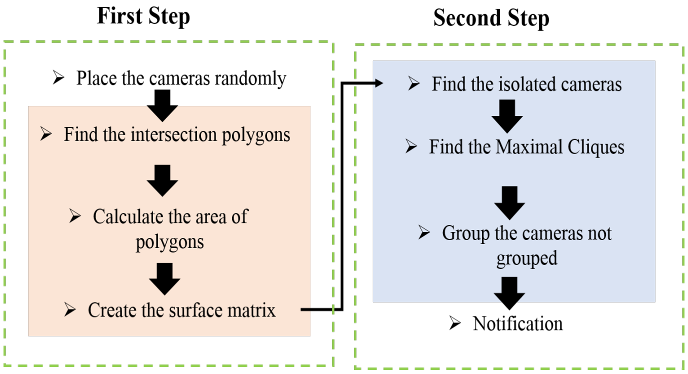

First Step
Place the cameras randomly
Find the intersection polygons
Calculate the area of polygons
Create the surface matrix
Second Step
Find the isolated cameras
Find the Maximal Cliques
Group the cameras not grouped
Notification

Fig. 3: Steps of clustering

3.1 Surface area calculation of the FoV intersection polygons

In the proposed modeling, the overlapping FoV between two cameras is represented by irregular polygons. Therefore, to determine these polygons, all polygon vertices must first be found, which are represented by the FoVs intersection points of two triangles. Figure 4 shows some examples of the intersection of two FoVs.

According to Figure 4, there are two types of intersection vertices: vertices generated by the triangles lines intersection and vertices that have the vertex of a triangle included in another triangle. First, it is necessary to search for the vertices of the triangles inside the other triangles. Therefore, Equation (5) and Conditional Expression (6) are used to determine if a point is in the triangle or not.

Let T be a triangle with the vertices $A(X_1, Y_1)$, $B(X_2, Y_2)$ and $C(X_3, Y_3)$. $M(X, Y)$ is a point in space: $m, k \in \Re$.

$$\begin{cases} X = k \cdot (X_2 - X_1) + m \cdot (X_3 - X_1) + X_1 \\ Y = k \cdot (Y_2 - Y_1) + m \cdot (Y_3 - Y_1) + Y_1 \end{cases} \tag{5}$$

$$\begin{cases} M \in ABC, & \text{if } m \geq 0 \wedge k \geq 0 \wedge m + k \leq 1 \\ M \notin ABC, & \text{Otherwise} \end{cases} \tag{6}$$

If there is a vertex among all the vertices of the triangles that verify the Conditional Expression (6), the coordinates of these vertices will be added to the list of polygon vertices.

The next step is to find the intersection points between the sides of the triangles (two by two) by solving the formula of the two Equation (7) and to find the intersection point of two lines. The Conditional Expression (9) allows to verify

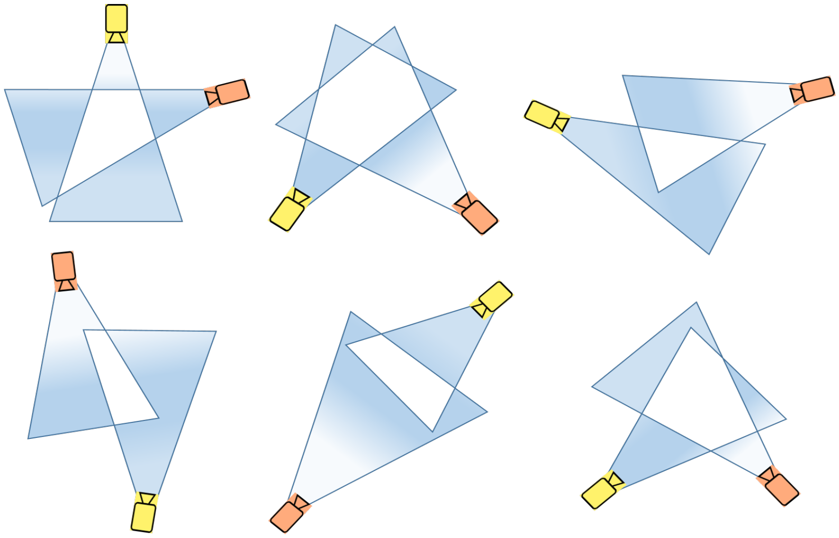


Fig. 4: Different possibilities for FoV overlapping

if the intersection point is accepted or not as a vertex of the polygon.
Let (AB) and (CD) be two lines with the following Equations (7):

$$\begin{cases} Y = aX + b \\ \\ Y = a'X + b' \end{cases} \tag{7}$$

With, $a, b \in \Re$, where $(a, b) \neq (0, 0)$ and $a', b' \in \Re$ where $(a', b') \neq (0, 0)$.

$$\begin{array}{l} ((X_A \leq X \leq X_B) \wedge (X_C \leq X \leq X_D)) \vee \\ ((X_A \leq X \leq X_B) \wedge (X_C \geq X \geq X_D)) \vee \\ ((X_A \geq X \geq X_B) \wedge (X_C \geq X \geq X_D)) \vee \\ ((X_A \geq X \geq X_B) \wedge (X_C \leq X \leq X_D)) \vee \end{array} \tag{8}$$

$$\begin{cases} P(X, Y) \text{ accepted }, & \text{if (8) is checked} \\ P(X, Y) \text{ unacceptable }, & \text{otherwise} \end{cases} \tag{9}$$

The coordinates of each accepted intersection point will also be added to the polygon's vertex list. At the end of this step, all points in the polygon will be detected and saved in a list for the next processing, as shown in Figure 5. The next step is to calculate the area of the polygons constructed by the following Equation (10) [7]:
Let $A_i(x_i, y_i)$, $i = 0, ..., n$ a polygon, where n is the number of polygon vertices such that $A_0 = A_n$.

$$Surface = \frac{1}{2} \sum_{i=0}^{n} (x_i \cdot y_{y+1}) - (x_i \cdot y_{y-1}) \tag{10}$$

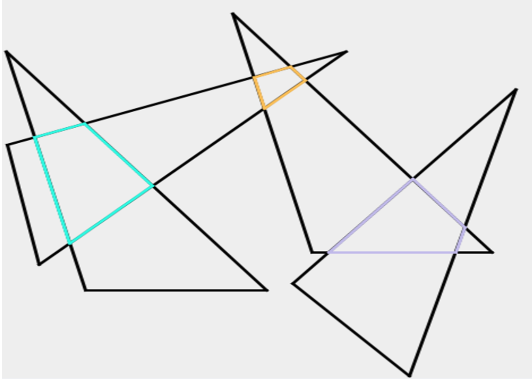
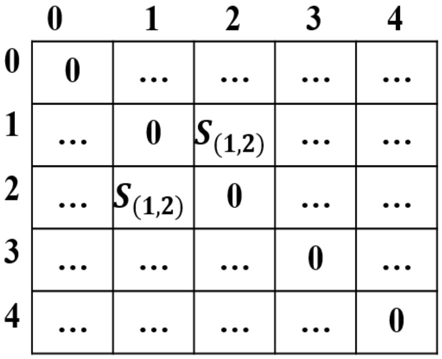


Fig. 5: Example of polygon intersection result

Each calculated surface is recorded in the surface matrix. Once all surfaces have been calculated, a symmetrical square matrix with a zero diagonal is obtained (Figure 6).



	0	1	2	3	4
0	0	...	...	...	...
1	...	0	$S_{(1,2)}$	...	...
2	...	$S_{(1,2)}$	0	...	...
3	...	...	...	0	...
4	...	...	...	...	0

Fig. 6: Matrix of surfaces

3.2 Camera Clustering

In this study, the camera network is modeled by a non-oriented graph $G(V, E)$, where the nodes V represent the cameras, the arcs E represent the existence of overlap between two cameras (nodes). The cluster of cameras is represented by the cliques found using the Bron-Kerbosch maximum clique discovery algorithm [8].

Algorithm 1 : Bron-Kerbosch “Without Pivot”

```
Input :
$P = \{V\}$ // set of all vertices in Graph $G$
$R = \{\}$ // is a possibly a clique
$X = \{\}$ // contains nodes already in some clique or processed

Proc BronKerbosch ($P$, $R$, $X$):
if $P \cup X = \{\}$ then
    Report $R$ as a Maximal Clique
end if

for each vertex $v$ in $P$ do
    BronKerbosch ($P \cap \{v\}$, $R \cup \{V\}$, $X \setminus \{v\}$)
    $P = P \setminus \{V\}$
    $X = X \cup \{v\}$
end for
```

1. Bron-Kerbosch algorithm
   The following pseudo-code represents the simplified implementation of Bron-Kerbosch's maximum clique discovery algorithm. The algorithm works as follows: Select a vertex v of P and add it to R, then delete its non-neighbors of P and X. Choose another vertex of the new set P and repeat the operation until P is empty. If P and X are empty, indicate the content of R as the maximum new clique (otherwise R contains a subset of a clique already found). Now, go back to the last vertex select and initialize P, R and X as they were before the choice, remove the vertex from P and add it to X, then expand the next vertex. If there are no more vertices in P, go back to the next level.
2. System modeling
   This section describes how to group all cameras $C = \{c_1, c_2, ..., c_n\}$ using the maximum clique search algorithm (Algorithm 1). In the proposed modeling, the definition of the maximum click (in the concept definition part) can be translated to define the concept of a group or cluster. Therefore, a camera cluster is a subset of cameras that overlap two by two, as shown in Figure 7, where the three red cameras represent a cluster. After calculating all surfaces of the intersection polygons in the first step (Section 3.1), the extraction of the clusters is carried out.
   At first, BS finds the isolated cameras and puts each camera in a single cluster. Then, it executes the Bron-Kerbosch algorithm (Algorithm 1) on the graph constructed by the surface matrix. For each execution, BS selects the clique that has the maximum cardinality among the obtained cliques and removes the cameras that exist in the chosen clique of the G to obtain well-separated groups. This process stops when all the cliques are extracted. At the end of this step, some cameras may stay ungrouped, in which case the BS adds these cameras to the cluster corresponding to the cluster of the

camera that has the greatest overlap. Once all cameras are clustered, BS informs all cameras of the identifier (ID) of their cluster.

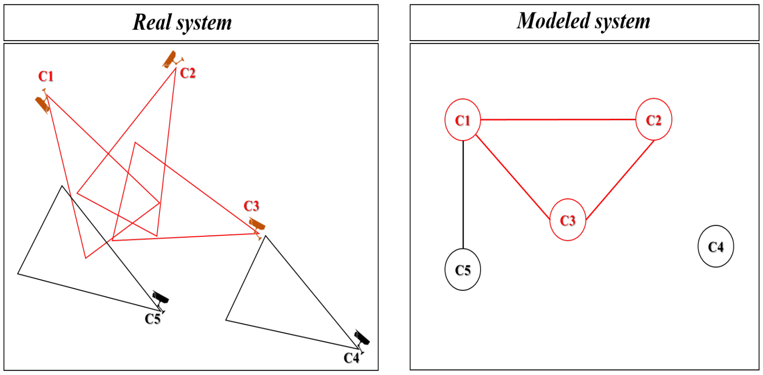


Fig. 7: Modeling example

4 Algorithm Test Results

This section discusses the results of the proposed clustering method. During simulation, all sensors in the monitoring area were configured with a FoV angle

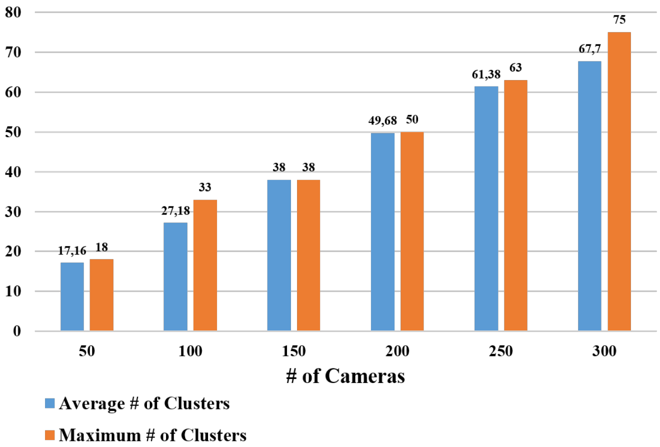


Fig. 8: Average and Maximum Number of Clusters.

$\theta = 60^o$ and R_s of 25 m and were randomly placed in the monitoring area. The system was executed 50 times for all cases where each execution represents a random spatial dispersion of the cameras, since the quality of the results is closely related to the location of the cameras.

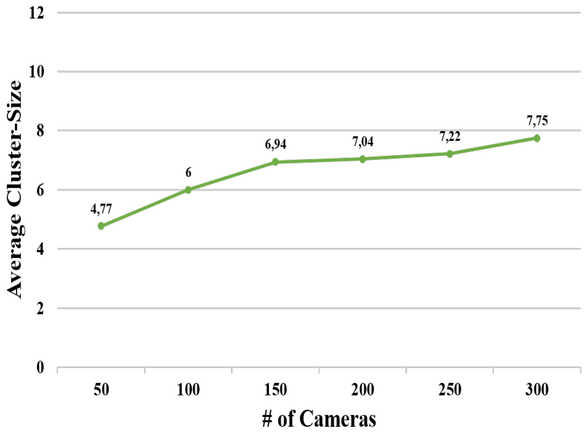


Fig. 9: Cluster-size average

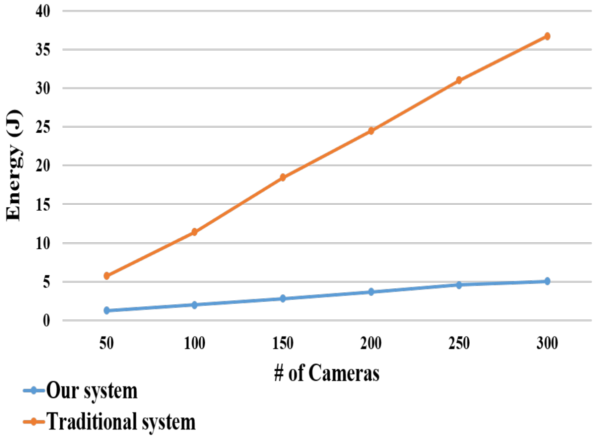


Fig. 10: Estimated energy consumption

Figure 8 presents the average and maximum number of clusters in the network. Using 300 randomly positioned cameras, an average of 67.7 clusters and a maximum of 75 clusters is obtained. The number of clusters decreases with the decrease in the number of cameras to 17.16 on average and a maximum of 18 clusters in a network includes 50 cameras. This decrease is related to the decrease in the overlap area between the cameras.

Figure 9 shows that the number of cameras in a cluster increases logarithmically with the number of cameras in the network. The number of cameras in clusters also increases due to the creation of larger overlapping areas between FoVs cameras, where the network of 300 cameras the size of the cluster reaches 7.75 cameras and in the network of 50 cameras reaches 4.77 cameras.

To evaluate the proposed method from an energetic point of view, the same method mentioned in the article [2] was used. In this method the cameras were periodically awakened (capture, processing, image sending) and at each interval, there was only one active camera in each cluster and the others were in sleep mode.

Equation (11) calculates the energy consumed by an active camera during the interval T = 1s (because the cameras operate in real time), meaning that E_{up}, E_{cap}, and E_{proc} are respectively the energies consumed to wake the camera, capture an image and process it.

$$E = E_{up} + E_{cap} + E_{proc} \tag{11}$$

The energy consumed by a cluster of average size N is calculated by Equation (12) and the total energy of the network is calculated by Equation (13), where E_{sleep} is the energy of a camera in sleep mode and C_{avg} is the average number of clusters.

$$E_{cluster} = E + (N-1) \cdot E_{sleep} \tag{12}$$

$$E_{total} = E_{cluster} \cdot C_{avg} \tag{13}$$

Figure 10 shows the average energy consumed in a network, using Cyclops [20] as camera sensor. Based on this study, the energy consumption of the collaborative camera system as compared to the non-collaborative system was reduced by 83.5

5 Conclusion

This paper presented an approach to improve the performance of existing video surveillance systems. The aim is to group neighboring cameras into a single cluster in order to restrict the communication area and reduce network congestion, and also to facilitate the implementation of collaboration algorithms between cluster cameras. The results demonstrated a reduction in data redundancy and, consequently, a reduction in the power consumption and storage space required to process and store the video stream.

For this purpose, the camera network was modeled by a non-oriented graph $G(V, E)$ where the nodes V represented the cameras, the arcs E represented the existence of an overlap between two cameras. The cluster of cameras in this modeling is represented by the cliques built on the basis of the Bron-Kerbosch maximum clique search algorithm.

There are still many challenges and outstanding questions about WMSN in order to develop a more reliable real-time video surveillance system that needs to be addressed in the near future, such as managing camera fault tolerance issues and developing new techniques to more effectively manage large volumes of multimedia data.

References

1. Alaei, M., Barcelo-Ordinas, J.M.: Mcm: multi-cluster-membership approach for fov-based cluster formation in wireless multimedia sensor networks. In: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference. pp. 1161–1165. ACM (2010)
2. Alaei, M., Barcelo-Ordinas, J.M.: A method for clustering and cooperation in wireless multimedia sensor networks. Sensors **10**(4), 3145–3169 (2010)
3. Alaei, M., Barcelo-Ordinas, J.M.: Node clustering based on overlapping fovs for wireless multimedia sensor networks. In: Wireless Communications and Networking Conference (WCNC), 2010 IEEE. pp. 1–6. IEEE (2010)
4. Alaei, M., Barcelo-Ordinas, J.M.: A collaborative node management scheme for energy-efficient monitoring in wireless multimedia sensor networks. Wireless networks **19**(5), 639–659 (2013)
5. Alaei, M., Barcelo-Ordinas, J.M.: A cluster-based scheduling for object detection in wireless multimedia sensor networks. In: Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks. pp. 50–56. ACM (2009)
6. Antoro, S., Sugeng, K., Handari, B.: Application of bron-kerbosch algorithm in graph clustering using complement matrix. In: AIP Conference Proceedings. vol. 1862, p. 030141. AIP Publishing (2017)
7. Braden, B.: The surveyor's area formula. The College Mathematics Journal **17**(4), 326–337 (1986)
8. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Communications of the ACM **16**(9), 575–577 (1973)
9. Chaurasiya, S.K., Mondal, J., Dutta, S.: Field-of-view based hierarchical clustering to prolong network lifetime of wmsn with obstacles. In: Electronics, Communication and Computational Engineering (ICECCE), 2014 International Conference on. pp. 72–77. IEEE (2014)
10. Costa, D.G., Silva, I., Guedes, L.A., Vasques, F., Portugal, P.: Optimal sensing redundancy for multiple perspectives of targets in wireless visual sensor networks. In: Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on. pp. 185–190. IEEE (2015)
11. Gherbi, C., Aliouat, Z., Benmohammed, M.: A survey on clustering routing protocols in wireless sensor networks. Sensor Review **37**(1), 12–25 (2017)
12. Gungor, V.C., Hancke, G.P., et al.: Industrial wireless sensor networks: Challenges, design principles, and technical approaches. IEEE Trans. Industrial Electronics **56**(10), 4258–4265 (2009)
13. Jung, K., Lee, J.Y., Jeong, H.Y.: Improving adaptive cluster head selection of teen protocol using fuzzy logic for wmsn. Multimedia Tools and Applications **76**(17), 18175–18190 (2017)
14. Kheirkhah, M.M., Khansari, M.: Clustering wireless camera sensor networks based on overlapped region detection. In: Telecommunications (IST), 2014 7th International Symposium on. pp. 712–719. IEEE (2014)

15. Kovács, L., Szabó, G.: Conceptualization with incremental bron-kerbosch algorithm in big data architecture. Acta Polytechnica Hungarica **13**(2), 139–158 (2016)
16. Li, S., Da Xu, L., Zhao, S.: The internet of things: a survey. Information Systems Frontiers **17**(2), 243–259 (2015)
17. Liu, X.: A survey on clustering routing protocols in wireless sensor networks. sensors **12**(8), 11113–11153 (2012)
18. Mishra, S., Chaurasiya, S.K.: Cluster based coverage enhancement for directional sensor networks. In: Next Generation Computing Technologies (NGCT), 2015 1st International Conference on. pp. 212–216. IEEE (2015)
19. Natarajan, P., Atrey, P.K., Kankanhalli, M.: Multi-camera coordination and control in surveillance systems: A survey. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **11**(4), 57 (2015)
20. Rahimi, M., Baer, R., Iroezi, O.I., Garcia, J.C., Warrior, J., Estrin, D., Srivastava, M.: Cyclops: in situ image sensing and interpretation in wireless sensor networks. In: Proceedings of the 3rd international conference on Embedded networked sensor systems. pp. 192–204. ACM (2005)
21. Sati, P., Goel, P., Goswami, S.: Enhancing coverage area in self-orienting directional sensor networks. International Journal of Information and Computation Technology **4**, 1661–1666 (2014)
22. Schaeffer, S.E.: Graph clustering. Computer science review **1**(1), 27–64 (2007)
23. Sharmin, S., Nur, F.N., Razzaque, M.A., Rahman, M.M., Alelaiwi, A., Hassan, M.M., Rahman, S.M.M.: α-overlapping area coverage for clustered directional sensor networks. Computer Communications **109**, 89–103 (2017)
24. Zarifneshat, M., Khadivi, P., Saidi, H.: A semi-localized algorithm for cluster head selection for target tracking in grid wireless sensor networks. Ad Hoc & Sensor Wireless Networks **25**(3-4), 263–287 (2015)