Distributed Object Recognition in Visual Sensor Networks

Stefano Paris
Mathematical and Algorithmic Sciences Lab
France Research Center - Huawei Technologies Co. Ltd.
LIPADE - Université Paris Descartes
stefano.paris@huawei.com, stefano.paris@parisdescartes.fr

Alessandro Redondi, Matteo Cesana, Marco Tagliasacchi
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Milan, Italy
Email: {name.surname}@polimi.it

***Abstract*—This work focuses on Visual Sensor Networks (VSNs) which perform visual analysis tasks such as object recognition. There, the goal is to find the image in a reference database which is the *closest match* to the image captured by camera sensor nodes. Recognition is performed by relying on visual features extracted from the acquired image, which are matched against a database of labeled features in order to find the closest image match. The matching functionalities are often implemented at a central controller outside the VSN. In contrast, we study the performance trade-offs involved in distributing the matching functionalities inside the VSN by letting sensor nodes performing parts of the matching process. We propose an optimization framework to optimally distribute the matching task to in-network sensor nodes with the goal of minimizing the overall completion time of the recognition task. The proposed optimization framework is then used to assess the performance of distributed matching, comparing it to a traditional, centralized approach in realistic VSN scenarios.**

***Index Terms*—Cache Placement, Object Recognition, Visual Sensor Networks**

I. Introduction

Visual Sensor Networks (VSNs) extend the application fields of traditional wireless sensor networks by including sensing nodes capable of acquiring and processing visual signals such as still images or videos. VSNs may have a significant impact in all application scenarios where capillary visual analysis tasks are needed at large scales. As an example, in the context of smart cities, the availability of battery-operated visual nodes may provide a much more complete coverage of the urban landscape, reaching a wider area and limiting the costs of the required infrastructure to support applications for traffic monitoring, smart parking metering, environmental monitoring, disasters management, etc. [1].

Such application scenarios require the implementation of different visual tasks in VSNs, including object recognition, face recognition, image retrieval, classification and tracking.

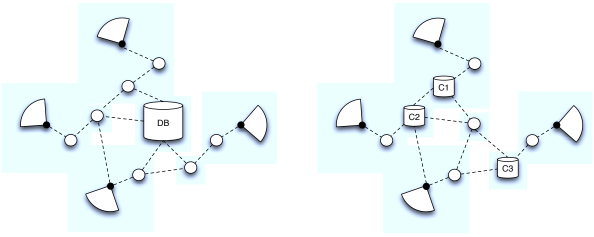
In this work, we mainly focus on those visual analysis tasks based on image retrieval such as object recognition. There, the goal is to find the object image in a reference database which is the *closest match* to the image captured by camera nodes. Recognition is performed by relying on visual features extracted from the acquired image, which are matched against a database of labeled features in order to find the closest image match. The common approach is to implement the matching functionalities at the very boundaries of the VSN at one or multiple central controller which feature high processing power, and null energy limitations. Roughly speaking, the completion time of the visual task depends on the time taken for processing visual features (at the cameras and at the servers) and on the transmission time to deliver such features to the central server(s).

We claim here that in case of bandwidth-limited multi-hop VSNs, the transmission time may become predominant, thus calling for effective solutions to move the matching functionalities closer to the camera nodes. To this extent, we study in this paper the performance trade-offs involved in distributing the features matching inside the VSN by letting network nodes performing parts of the matching process, as illustrated in Figure 1. For the task at hand, first we throughly characterize the processing time on low-power sensor nodes required for running nearest neighbors search, which is at the very heart of the matching process for image retrieval tasks. Then we propose a mathematical formulation for the *Distributed Object Recognition (DOR)* problem, in which the matching task is distributed to in-network sensor nodes, by splitting and moving parts of the reference database to particular sensor nodes in the VSN. Numerical results on sample VSNs show that a significant reduction of the completion time can be achieved by distributing object recognition tasks.

The rest of this paper is organized as follows. Section II comments on related work in the field highlighting the main contributions of the current work. Section III provides a qualitative description of the reference pipeline for object recognition tasks in VSNs, while Section IV introduces the network model and the assumptions considered in this work. Section V gives a mathematical programming formulation for the *DOR* problem. Section VI illustrates and analyzes numerical results that show the validity of the proposed approach to improve the efficiency of distributed visual tasks in VSNs. Concluding remarks are discussed in Section VII.

II. Related Work

Broadly speaking, our work is naturally related to resource placement problems in communication networks. The common objective is to decide where to place specific resources at network nodes while meeting network- and end-user-related



(a) Centralized (b) Distributed

Fig. 1: Centralized and distributed VSN scenarios. Black circles represents cameras, while white circles identify sensor nodes, which can be selected to host portions of the database.

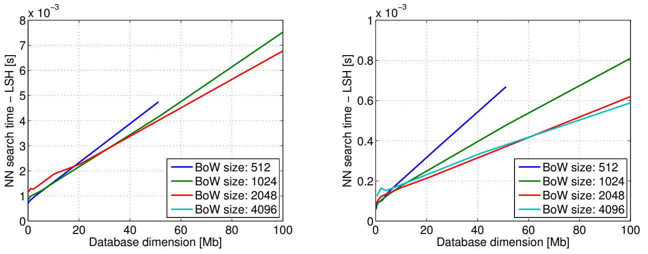
quality constraints. In [2], the focus is on the optimal placement of database replicas to minimize the cost of accessing a database while accounting for replication and replicas update costs. Along the same lines, the problem of placing servers and proxies in the Internet has been extensively studied in the literature [3]. More recently, under the push of novel networking paradigms like Content Centric Networking (CCN) and Information Centric Networking (ICN), the focus has slightly moved from where to place servers and proxies to where to place contents. Tang *et al.* investigate in [4] the problem of placing the object replicas in content distribution systems with the goal of minimizing the replication cost while meeting QoS requirement for the end users accessing the contents. In [5], a content placement approach is proposed to support video on demand applications on top of peer-to-peer systems. Although our modeling approach bears some similarities with classical frameworks for resource placement, the aforementioned related work generally refers to traditional communication networks ranging from the Internet to mobile ad hoc networks, whereas we focus here on VSNs, which have distinctive features in terms of bandwidth/hardware limitations. Moreover, we are not (only) concerned in placing contents, servers or gateways at sensor nodes as in [6], but we also target the distribution of matching tasks based on algorithms to find the nearest neighbors in multi-dimensional spaces. In this last field, recent work has addressed the performance evaluation and improvement of algorithms to compute the nearest neighbors in multi-dimensional spaces. Aly et al. [7] introduce a distributed k-d tree implementation where they place a root k-d tree on top of all the other trees (leaf trees) with the role of selecting a subset of trees for searching purposes, showing the higher throughout with respect to using independent trees. Muja and Lowe propose a through performance evaluation of different approximate alternatives to compute the nearest neighbors further assessing the speed-up in case distributed versions of the algorithms are executed on parallel machines with multiple cores[8]. In [9], authors present a distributed system for matching high-dimensional multimedia objects (DIMO), which provides multimedia applications with the basic function of computing the K nearest neighbors on large-scale datasets. In [10], two schemes for parallelizing the KD-tree search method are proposed: in the Independent KD-tree scheme, each node store a portion of the original dataset and computes an independent and local KD-tree. On the other side, in the Distributed KD-tree version, one single KD-tree is centrally computed, a central node keeps the upper part of the tree locally whereas subbranches of the search tree are offloaded to other nodes. To summarize, our work is, to the best of our knowledge, one of the first attempt to study distributed matching solutions to support visual task of image retrieval in VSNs.

III. Object Recognition Pipeline

Generally, an image retrieval system is based on a two-steps process. First, the input image is processed in order to extract local or global features, which concisely represent the content of the image itself. Such features are then matched against a database of labeled features in order to find the closest image match. The process can be customized by properly choosing (i) the particular algorithms used to extract image features and (ii) how the matching process is performed.

In this work, we consider the Bag of Visual Words (BoW) approach [11], in which local features of an image (each one corresponding to a salient point in the image) are quantized into visual *words*, which are defined by a fixed-size dictionary (generally computed by a k-means clustering performed on the features of a number of training images). For each image, a signature is produced, in the form of a histogram, which counts the number of times a particular visual word occurred in the image. Image matching is then accomplished using these signatures (i.e., comparing histograms), instead of matching every single local feature, providing very fast retrieval.

Even though the BoW model allows to represent one image with a single histogram, linear search may be still costly in the case of large databases. However, it is possible to use approximate nearest-neighbor search algorithm in order to speed up the search process, still retrieving the nearest neighbor with high probability. As an example, Locality Sensitive Hashing has been shown to be very effective for fast matching of BoW histograms. However, it is worthy analyzing the performance of such fast nearest neighbors algorithm when executed on low-power sensor nodes. For the task at hand, we implemented a BoW search engine based on multi-probe LSH [12], by relying on the OpenCV FLANN library [13]. The search engine accepts an image as input, extracts its BoW representation and search the nearest BoW histogram in a known dataset. The reference hardware is composed of an ARM-based BeagleBone Linux computer equipped with a 500 MHz ARM Cortex A8 CPU and 256 MB of RAM. This device can be used to implement a visual sensor node by connecting a camera and a low-power radio transceiver [14]. Figure 2 shows the time needed to retrieve the BoW nearest neighbor using multi-probe LSH on a BeagleBone (Figure 2(a)) operating at 500 MHz and on a MacBook Pro (Intel Core Duo, 2.3 GHz) (Figure 2(b)) when varying (i) the dimension of the search database and (ii) the dimension of the BoW histogram (i.e., the number of visual words in the BoW dictionary). As one can see the search time grows linearly with



(a) BeagleBone (b) MacBook Pro

Fig. 2: Processing time of the BoW algorithm as a function of the DataBase and the dimension of the BoW histogram using a (a) BeagleBone and a (b) MacBook Pro.

the database size: we model this relation as $\tau = p \cdot d + o$, where τ is the search time, p the search speed, d the database size and o a processing overhead. Moreover, the search procedure on a resource-constrained platform such as the BeagleBone is one order of magnitude slower than on a powerful machine. However, as we shall see in the next sections, it is still possible to reduce the total latency by properly distributing the database in the network.

IV. System Model and Problem Statement

We consider a visual sensor network composed of a set $\mathcal{N}$ of wireless nodes divided in cameras (the subset $\mathcal{S}$) and relay nodes (the subset $\mathcal{R} = \mathcal{N} \setminus \mathcal{S}$), as illustrated in Figure 1.

Each camera $s \in \mathcal{S}$ acquires an image and processes it to extract its BoW histogram which needs to be transmitted to the database for matching, wherever it is located. Let L be the size of the BoW query histogram and T the period of acquisition of an image from a camera. Assuming the processing of a continuous stream of images, each camera generates a bitrate identified as $\rho_s = L/T$.

We aim at reducing the visual task completion time by selecting a subset of relay nodes that host shares of a given database of size β and further perform the matching algorithm. Each relay node features a CPU and a non-volatile memory, which limit the possible placements of the database's portions. The processing power and storage capacity of each relay node $i \in \mathcal{R}$ are denoted by p_i and b_i, respectively. The capacity of the wireless link $(i,j) \in \mathcal{L}$ that can be established between any two nodes is defined as c_{ij}; for each wireless link $(i,j) \in \mathcal{L}$ in the network, the set $\mathcal{I}(i,j)$ contains all the interfering links, namely all the links that cannot be simultaneously activated with the link (i,j), due to self-interference. Table I summarizes the notation used throughout the paper.

V. Distributed Object Recognition Problem

The DOR problem can be defined as follows: given a set of cameras and a set of relay nodes, find for each camera a subset of nodes in the network and a portion of the original database to be assigned to each element in the subset, such that the overall visual analysis task completion time is minimized.

TABLE I: Basic notation used in the paper.

Parameters	
$c_{ij} = c$	Capacity of wireless link (i,j) (bit/s).
p_i	Processing speed of node i (s/bit)
o_i	Processing overhead of node i (s)
b_i	Storage capacity of node i (images)
L	Average packet size (bit)
β	Database size (bit)
T	Inverse of frame rate (s)
ρ_s	Bitrate generated by camera s (bit/s)
ν	Response generated by a matching node (bit/s)

The nodes hosting a database portion and thus performing in-network matching tasks are called hereafter "matching nodes". As we let each camera have its own set of matching nodes, the original database may be replicated several times in the network. The DOR problem can be formalized as a Mixed Integer Linear Programming (MILP) model by using the following decision variables and constraints.

Binary decision variables x_i^s ($s \in \mathcal{S}$ and $i \in \mathcal{R}$) indicate which node is selected as matching node for camera s, namely the relay that hosts a portion of the database used by camera s for recognizing captured images ($x_i^s = 1$ if node i hosts the portion of the database for camera s, $x_i^s = 0$ otherwise). The portion of the database hosted on node i for camera s is instead identified by the continuous variable d_i^s. Furthermore, let variables $f_{ij}^s \in \mathbb{R}^+$ and $y_{ij}^s \in \mathbb{N}$, $(i,j) \in \mathcal{L}$, $(i,j) \in \mathcal{L}$, denote the traffic flow and the number of visual queries generated by camera s and routed on link (i,j) towards all matching nodes used by s. Binary variables z_{ij}^s ($s \in \mathcal{S}$ and $(i,j) \in \mathcal{L}$) provide the wireless links that are used to carry the traffic generated by camera s. Therefore, $z_{ij}^s = 1$ indicates that link (i,j) is used by camera s to transmit $y_{ij}^s > 0$ visual queries towards some of (or all) the corresponding matching nodes. We observe that the subsets of relays used by cameras as matching nodes may a each other. Put another way, a relay $i \in \mathcal{R}$ can be selected as matching node to host portions of the database used by different cameras. Finally, we define the variables $n_i^s \in \mathbb{N}$ and $\theta \in \mathbb{R}^+$ in order to enumerate the nodes on the tree connecting a camera to its matching nodes and minimize the worst latency (due to transmission and processing), respectively. Specifically, n_i^s denotes the level of relay i on the multicast tree used by camera s to transmit its traffic, whereas θ represents the maximum transmission and processing time on a branch of the multicast tree rooted at the camera and terminating in the matching nodes.

Given the above definitions and notation, the DOR problem amounts to the following mathematical program:

$$\min \ \theta \tag{1}$$

$$\text{s.t. } \frac{L}{c} \cdot \left(n_i^s - 1\right) + p_i \cdot d_i^s + o_i \cdot x_i^s \le \theta \qquad \forall i \in \mathcal{R}, s \in \mathcal{S} \tag{2}$$

$$\sum_{(s,i) \in \mathcal{L}} f_{si}^s - \sum_{(i,s) \in \mathcal{L}} f_{is}^s = \sum_{j \in \mathcal{R}} x_j^s \cdot (\rho_s + \nu) \qquad \forall s \in \mathcal{S} \tag{3}$$

$$\sum_{(i,j) \in \mathcal{L}} f_{ij}^s - \sum_{(j,i) \in \mathcal{L}} f_{ji}^s = -x_k^s \cdot (\rho_s + \nu) \qquad \forall k \in \mathcal{R}, s \in \mathcal{S} \tag{4}$$

$$\sum_{(s,i) \in \mathcal{L}} y_{si}^s - \sum_{(i,s) \in \mathcal{L}} y_{is}^s = \sum_{j \in \mathcal{R}} x_j^s \qquad \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{5}$$

$$\sum_{(i,j) \in \mathcal{L}} y_{ij}^s - \sum_{(j,i) \in \mathcal{L}} y_{ji}^s = -x_k^s \qquad \forall k \in \mathcal{R}, s \in \mathcal{S} \tag{6}$$

$$\sum_{s\in\mathcal{S}} f^s_{ij} + \sum_{s\in\mathcal{S}} f^s_{ji} \le c_{ij} \qquad \forall (i,j)\in\mathcal{L}, (j,i)\in\mathcal{L} \quad (7)$$

$$\frac{f^s_{ij}}{c_{ij}} \le y^s_{ij} \qquad \forall (i,j)\in\mathcal{L}, s\in\mathcal{S} \quad (8)$$

$$\sum_{s\in\mathcal{S}} \frac{f^s_{ij}}{c_{ij}} + \sum_{(u,v)\in\mathcal{I}(i,j)} \sum_{s\in\mathcal{S}} \frac{f^s_{uv}}{c_{uv}} \le 1 \qquad \forall (i,j)\in\mathcal{L} \quad (9)$$

$$\sum_{s\in\mathcal{S}} d^s_i \le b_i \qquad \forall i\in\mathcal{R} \quad (10)$$

$$\sum_{i\in\mathcal{R}} d^s_i \ge \beta \qquad \forall s\in\mathcal{S} \quad (11)$$

$$\frac{d^s_i}{b_i} \le x^s_i \qquad \forall i\in\mathcal{R}, s\in\mathcal{S} \quad (12)$$

$$n^s_j - n^s_i \ge 1 - |\mathcal{N}| \left(1 - z^s_{ij}\right) \qquad \forall (i,j)\in\mathcal{L}, s\in\mathcal{S} \quad (13)$$

$$n^s_s = 1 \qquad \forall s\in\mathcal{S} \quad (14)$$

$$\frac{f^s_{ij}}{c_{ij}} \le z^s_{ij} \qquad \forall (i,j)\in\mathcal{L} s\in\mathcal{S} \quad (15)$$

$$z^s_{ij} \le y^s_{ij} \qquad \forall (i,j)\in\mathcal{L} s\in\mathcal{S} \quad (16)$$

$$d^s_i, f^s_{ij} \ge 0 \qquad \forall (i,j)\in\mathcal{L}, i\in\mathcal{R}, s\in\mathcal{S} \quad (17)$$

$$x^s_i, y^s_{ij} \in \{0,1\} \qquad (i,j)\in\mathcal{L}, \forall i\in\mathcal{R}, s\in\mathcal{S} \quad (18)$$

The objective function (1) coupled with constraints (2) minimizes the worst latency for object recognition tasks, which comprises the time for transmitting visual queries from a camera to the farthest matching node and the processing time due to matching in the database portion hosted by that matching node. Constraints (3) and (4) define the flow balance at node j. The term $\sum f^s_{ji}$ accounts for the total traffic generated by a camera $s \in \mathcal{S}$, while terms $\sum f^s_{ij}$ and $\sum f^{as}_{ji}$ represent the total incoming and outgoing traffic originated from camera s, respectively. Constraints (5) and (6) prevent traffic splitting among several links. Put another way, they force the utilization of a single path to route the traffic generated by a camera towards the corresponding matching node. The set of constraints (7) ensures that the total traffic routed on the forward and reverse links connecting two devices i and j does not exceed the channel capacity, denoted by c_{ij}, while (8) force the variable $y^s_{ij} = 1$ whenever the link (i, j) is used to transmit the traffic generated by the camera s. In contrast, constraints (9) further limit the maximum amount of traffic that can be routed on a wireless link considering all simultaneous transmissions over its interfering links (i.e., all links that cannot be simultaneously activated). Constraints (10) set a limit on the maximum amount of portions of the database that can be stored into any network node equal to the storage capacity of the node, whereas the set of constraints (11) force the relays selected for a camera as matching nodes to store jointly the whole database (the aggregated amount of information stored on all matching nodes used by a camera must be equal to the complete database size). Constraints (12) denote coherence constraints to force the activation of node i as matching node (i.e., $x^s_i = 1$) whenever some storage is reserved for camera s ($d^s_i > 0$). The set of constraints (13)-(16) enumerates sequentially all nodes on the path connecting a camera to its matching nodes. The value n^s_i represents therefore the level of network node i in the tree rooted at camera s and terminating at each relay selected as matching nodes (i.e., $i \in \mathcal{R}$ such that $x^s_i = 1$). Such value can be used to compute the number of links traversed on a path from a camera and its matching nodes, thus minimizing the worst transmission time. Finally, constraints (17) ensure the positiveness of flow and storage variables, while (18) ensure the integrality of binary decision variables.

The following proposition holds on the complexity of the DOR problem.

Proposition V.1. *The DOR problem is NP-hard.*

Proof: Let's prove that DOR problem is NP-hard by considering a simplified instance of the DOR problem where the set of matching nodes assigned to each camera is fixed, that is, variables x^s_i become parameter of the problem. In this case, the DOR problem becomes a problem of shortest path multi-commodity with non-splittable flows which is known to be NP-hard [15]. Thus, the DOR problem contains as a special case an NP-hard problem, which makes the DOR problem itself NP-hard. ■

VI. Numerical Results

In this section, we illustrate the results obtained solving the DOR problem, evaluating the impact of several network parameters, like the number of cameras and relay storage size. We first describe the experimental methodology of our simulations, then we discuss the performance of the proposed DOR approach, comparing it with a centralized scheme.

A. Experimental Methodology

We consider VSNs where nodes are randomly scattered over a circular area of radius 40 m. Cameras are deployed uniformly at the external border of the area, whereas 15 relay nodes, are randomly placed inside the area to simulate a typical VSN deployment. To evaluate the effect of the number of cameras, we vary their number in the range $[2, 6]$. Each camera node is characterized by the parameters derived from the analysis of the testbed presented in Section III. Regarding the search time, we set $p_i = 68.75$ $\mu s/Mb$ and $o_i = 625$ μs. Moreover, we observe that $\beta = 100$ MB is the maximum amount of database information that can be stored in the RAM of a BeagleBone when using nearest neighbor search algorithms like those proposed in [8]. Nonetheless, 100 MB of information permits to store 100000 image representations using the BoW model [11] with 1 $k$bit for each image descriptor/signature. In our simulations, we consider the utilization of a single ZigBee channel for all devices and the transmission power is set to 10 mW. The reception and carrier sense thresholds are set according to the sensitivity of the 802.15.4 compliant CC2420 transceiver[1]. Furthermore, the interference graph is computed assuming the utilization of an ARQ mechanism as error recovery technique (i.e., we assume DATA-ACK message exchange among network nodes involved in data communications). With this settings and with the standard packet size of 107 bytes, nearby nodes can achieve a throughput of roughly 80 kbps [16]. The path loss, which is necessary to evaluate the sensitivity of the receiving

[1] Available on-line http://www.ti.com/lit/ds/symlink/cc2420.pdf

node, is computed according to the Friis propagation model. We underline that all above assumptions do not affect the proposed DOR algorithm, which is general and can be used to solve any network scenario.

The performance of the proposed DOR approach are compared against those of a centralized approach where matching is performed at a single relay node with unlimited energy budget (i.e., the central controller). In the following, the centralized benchmark will be referred to as Centralized Object Recognition (COR). The two approaches are compared with respect to two performance metrics, namely analysis task latency and energy consumption. The former includes the transmission time to propagate the query on the path(s) from the camera node to the matching node(s), the processing time spent by the matching node(s), and the transmission time to propagate back the query response on the reverse path(s) from the matching node(s) to the camera nodes. The latter metric is measured by computing the network lifetime as the number of object recognition tasks which can be performed by camera nodes over time, that is:

$$\xi = \min\left\{ \frac{\overline{E}}{E_i} : i \in \mathcal{R} \right\}. \tag{19}$$

where $\overline{E}$ is the total energy budget at a node (e.g., 32.5 kJ), whereas E_i represents the energy spent by each node i, when all cameras send their request towards their matching nodes. The value of E_i for the generic relay i is defined as sum of the energy spent for transmission and the computation, as follows:

$$\begin{aligned} E_i \quad &= E_i^{tx} + E_i^{cpu} = P_i^{tx} \cdot \left(\frac{Q}{c} \sum_{s\in\mathcal{S},(i,j)\in\mathcal{L}} y_{ij}^s + \frac{L}{c} \sum_{s\in\mathcal{S},(j,i)\in\mathcal{L}} y_{ji}^s \right) + \\ &+ P_i^{cpu} \cdot \sum_{s\in\mathcal{S}} x_i^s \cdot (p_i \cdot d_i^s + o_i). \end{aligned} \tag{20}$$

In Equation (20), P_i^{tx} and P_i^{cpu} represents the transmission and processing power of relay node i, respectively. The value of $\frac{Q}{c}$ is the time spent to transmit/forward a query, whereas $\sum_{(i,j)\in\mathcal{L}} y_{ij}^s$ represents the number of outgoing transmissions from node i for camera s. Similarly, the value of $\frac{L}{c}$ is the time spent to transmit a response (whose size is equal to a packet), whereas $\sum_{(i,j)\in\mathcal{L}} y_{ji}^s$ represents the number of incoming transmissions of node i for camera s, which corresponds to the transmissions of the corresponding responses sent back by matching nodes on the reverse path(s). In contrast, $x_i^s \cdot (p_i \cdot d_i^s + o_i)$ represents the time spent by matching node i to perform the object recognition task for the requests sent by camera s.

Note that, due to the high computational and space complexities of the ILP model, we could not scale beyond the network sizes and time epochs discussed above (i.e., 15 relays and 6 cameras). Indeed, the maximum computational time to solve the problem using the CPLEX solver on an Intel i7-3770 Processor with 8 cores, clock speed of 3.4 Ghz and 8 GByte of RAM was approximately equal to 4 hours. For each network scenario we performed 40 independent measurements, computing very narrow 95% confidence intervals, which we do not show for the sake of clarity.

B. Performance Evaluation

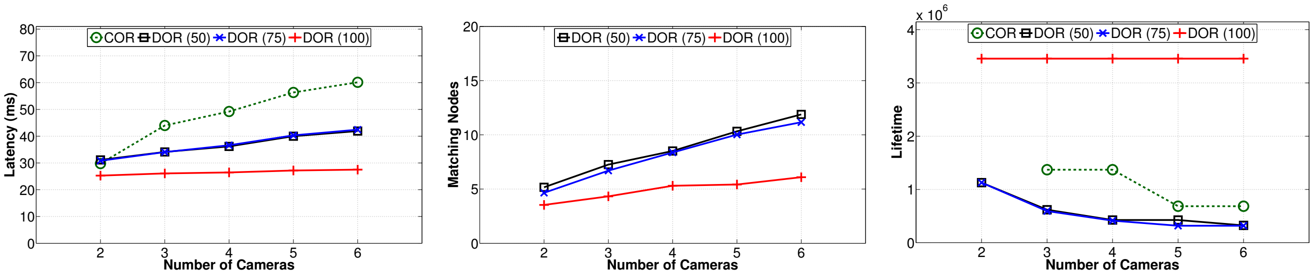
We first evaluate the effect of the number of cameras on the performance of the COR and DOR schemes. Specifically, in the network scenario described above, we progressively remove one camera to decrease their number from 6 to 2, thus making gradually the topology asymmetric (at the beginning 6 cameras are uniformly distributed on the circular edge). Figures 3 shows the *latency*, the *number of matching nodes*, and the *network lifetime* obtained using the proposed optimization approaches for different storage capacities of relay nodes, namely 50, 75, and 100 Mb. Note that the number of matching nodes are illustrated only for the DOR scheme, since the centralized optimization is not affected by the processing time and only one powerful device, which replaces a relay node, performs the visual tasks.

As illustrated in the figures, the DOR scheme achieves the best performance in terms of latency as the number of cameras increases, thus confirming the validity of the distributed matching/object recognition approaches for increasing the system's responsiveness. The COR approach performs slightly better than the DOR with 2 cameras, since in this latter case there likely exists a relay node directly connected to both cameras (recall that the topology is highly asymmetric with 2 and 3 cameras). However, as long as the number of cameras increases and the topology becomes more symmetric, the DOR solutions outperform the COR approach, since it can activate individual matching nodes for each camera no more than 2 hop away. This is further confirmed by Figure 3(b), which shows that the DOR approach increases the number of matching nodes as the number of cameras increases, since it selects the closest relays to act as matching nodes for each camera.

We further observe from Figure 3(a) that the storage capacity slightly affects the maximum latency experienced by camera nodes, since in sparse networks only a subset of cameras can place the largest portion of the database on the closest relay (e.g., 75 MB on the first hop, and 25 MB on the second hop). Only if the relay's storage size matches the database size, all cameras can surely put all their information on the 1-hop neighbor. For example, the maximum latency experienced by 6 cameras results approximately 18 ms and 24 ms, with relay's storage size equal to 100 and 75 MB, respectively.

The DOR approach puts extra burden onto the matching nodes which are requested to perform energy-consuming matching functionalities. To this extent, the DOR approach may lead to a reduced network lifetime if compared to the COR approach.

Figure 3(c) illustrates the *network lifetime* defined in Equation (19) of the COR and DOR approaches for different relay's storage sizes as a function on the number of cameras. Note that the network lifetime for the COR approach is not defined for two cameras, since the server that performs the visual tasks



(a) Latency (b) Matching Nodes (c) Lifetime

Fig. 3: Object recognition *latency*, processing overhead, and number of matching nodes as a function of the number of cameras.

is always placed 1-hop away from the two cameras and no intermediate node is involved in the visual analysis process (both the server and cameras have not energy constraints).

As expected, when the storage size of the relay is lower than the database size, the DOR scheme always reduces the network lifetime due to the energy spent by matching nodes to execute object recognition tasks on their portion of database. However, the lifetime gap between the distributed and centralized approaches decreases as the number of cameras increases, since the relay nodes close to the server spend more energy for the transmission of the traffic generated by and directed to cameras. In contrast, with the DOR approach almost all relay nodes spend the same amount of energy for the transmission, since the traffic is distributed more fairly within the network.

The network lifetime loss caused by the increase of the energy spent for processing purposes is practically offset by the improvement of the system responsiveness achieved with the DOR approach (cf. the latency in Figure 3(a)). Indeed, when the server is placed at least 2-hop away from cameras like in the symmetric topology with 6 cameras, the lifetime of the network deployed according to the DOR scheme decreases by 35% with respect to the COR approach, whereas the latency improves by 30%, considering a storage capacity of 50 Mb.

VII. Conclusion

Motivated by the need to reduce the completion time of visual tasks of object recognition in Visual sensor Networks (VSNs), we considered in this paper the opportunity of distributing in the VSN the matching task, by letting in-network sensor nodes play the role of matching nodes. Such approach has the advantage of moving the matching nodes closer to the camera nodes, with a consequent reduction in the overall task latency. On the other side, the matching functionalities put an extra burden onto the matching sensor nodes, in terms of consumed energy and required data storage space. We have proposed a MILP formulation to optimally select in-network sensor nodes to play the role of matching nodes with the objective of minimizing the visual task completion time, while accounting for node-related and network-related resource constraints. The MILP formulation was finally used to evaluate the trade-off between the reduction of the visual task completion time and the loss in network lifetime.

Acknowledgment

The project GreenEyes acknowledges the financial support of the FET programme within the 7^{th} Framework Programme of the European Commission, under FET-Open grant 296676.

References

[1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury. Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, 14(6):32–39, 2007.

[2] M. L. Fisher and D. S. Hochbaum. Database location in computer networks. *Journal of the ACM*, 27(4):718–735, 1980.

[3] J. Wu, S.F. Shih, P. Liu, and Y. Chung. Optimizing server placement in distributed systems in the presence of competition. *Journal of Parallel and Distributed Computing*, 71(1):62–76, 2011.

[4] X. Tang and J. Xu. Qos-aware replica placement for content distribution. *IEEE Trans. on Parallel and Distributed Systems*, 16(10):921–932, 2005.

[5] B. Tan and L. Massoulié. Optimal content placement for peer-to-peer video-on-demand systems. *IEEE/ACM Trans. on Networking*, 21(2):566–579, 2013.

[6] A. Capone, M. Cesana, D.D. Donno, and I. Filippini. Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach. *Computer Communications*, 33(10):1151–1161, 2010.

[7] M. Aly, M. Munich, and P. Perona. Distributed kd-trees for retrieval from very large image collections. In *British Machine Vision Conference (BMVC)*, August 2011.

[8] M. Muja and D. Lowe. Scalable nearest neighbour algorithms for high dimensional data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2014.

[9] A. Abdelsadek and M. Hefeeda. Dimo: Distributed index for matching multimedia objects using mapreduce. In *ACM Multimedia Systems Conference*, pages 115–126, New York, NY, USA, 2014. ACM.

[10] L. Qiaomin, J. Yang, B. Zhang, R. Wang, N. Ye, and M. Yan. Distributed face recognition in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014.

[11] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[12] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases*, pages 518–529, 1999.

[13] M. Muja and D.G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*, pages 331–340, 2009.

[14] A. Canclini, L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi. Comparison of two paradigms for image analysis in visual sensor networks. In *ACM Conference on Embedded Networked Sensor Systems*, pages 62–63, New York, NY, USA, 2013. ACM.

[15] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *IEEE Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1975.

[16] S. Paniga, L. Borsani, A Redondi, M. Tagliasacchi, and M. Cesana. Experimental evaluation of a video streaming system for wireless multimedia sensor networks. In *IFIP Med-Hoc-Net*, pages 165–170, 2011.