IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 2, 2022, accepted March 26, 2022, date of publication April 11, 2022, date of current version April 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164681

A Study on the Impact of Multiview Distributed Feature Coding on a Multicamera Vehicle Tracking System at Roundabouts

SALMA ELEUCH[1], NADIA KHOUJA[1], SIMONE MILANI[2], (Member, IEEE), TOMASO ERSEGHE[2], AND FETHI TLILI[1]

[1]GRESCOM Laboratory, Higher School of Communications of Tunis (SUPCOM), University of Carthage, Carthage 1054, Tunisia
[2]Dipartimento di Ingeneria dell'Informazione, Università degli studi di Padova, 35131 Padua, Italy

Corresponding author: Salma Eleuch (salma.elleuch@supcom.tn)

ABSTRACT Visual sensor networks are one potential enabler for the evolution of the Internet of things. Due to their limited resources in terms of energy and bandwidth, it is crucial to identify appropriate approaches that take into considerations such constraints and reduce the amount of data transmitted to the gathering point (sink). In this context, this paper describes the impact of a distributed smart-camera system that exploits an analyze-then-compress strategy, on a multi-view vehicle tracking at roundabouts application. In the tested system, part of the processing is shifted to the smart cameras, i.e., the object detection/classification and feature extraction, so that only the extracted features describing moving vehicles are transmitted instead of the whole image/video. Features are further compacted by using a state-of-the-art distributed coding technique, based upon an efficient clustering method that exploits the temporal and spatial (multiple views) correlations between features. The system is tested on a real-data scenario, by evaluating the bit-rate reduction capabilities in dependence of the channel conditions, as well as the matching accuracy of the reconstructed descriptors in the specific tracking application. Both feature-wise and object-wise matching are investigated. For the chosen application scenario, a bit-rate reduction of $30-35\%$ is proved to be achievable in non-ideal channel conditions. Even more interestingly, such reduction is proved not to harm the matching accuracy (i.e., it is coherent with the target application), for which an F-score up to 0.923 is guaranteed.

INDEX TERMS Bit-rate reduction, distributed feature coding, feature matching, multi-view, resource allocation, roundabouts, smart city, traffic monitoring, vehicle tracking, visual sensor networks.

I. INTRODUCTION

Traffic monitoring [1] is one of the most important applications in smart city technologies. Multiple video solutions have been proposed for vehicle tracking and congestion estimation [2] in highways, intersections, and roundabouts; such scenarios have proved to be extremely challenging due to the high number of conflicting nodes, important occlusion probabilities, and the existence of several driving routes [3]. To overcome these challenges and to build an efficient vehicle tracking system at intersections and roundabouts, it is crucial to cover the entire area by placing multiple cameras. To this purpose, scientific and industrial research has been developing low-cost scalable visual sensor networks (VSNs), i.e., networks consisting of several connected embedded smart camera sensors that are able to process the sensed data and communicate among each other to gather the information at a central sink node with higher energy and processing capabilities [4], [5]. The use of partially-overlapped fields of views (FoVs) of cameras in these networks allow solving many practical problems, such as occlusions, illumination changes, and pose variations. Unfortunately, smart traffic monitoring applications require the processing and transmission of huge amount of data, as well as a significant energy consumption [5]. Satisfying these requirements by VSNs is challenging due to their limited computational, communication, and energy resources. This imply a urgent need for efficient visual compression architectures that minimize the transmission and processing powers, while keeping a satisfying accuracy in the final processing task, e.g., object tracking [6]–[8].

Previous works have tackled the problem following two different paradigms. The compress-then-analyze (CTA) approach compresses the acquired data (i.e., the whole image

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

or video) at the node terminal and sends it to the sink node where the visual analysis is performed [9]. Alternatively, in the analyze-then-compress (ATC) paradigm the source nodes locally process the visual data to generate a set of visual features that consist in a semantically-relevant representation of the acquired scene; features will be then delivered to the sink node for further high-level analysis [9], [10]. This second strategy can significantly reduce the amount of transmitted data and energy consumption since irrelevant details (with respect to the target task) can be discarded and coding schemes can exploit the correlation among data from different sensors. Interestingly, whenever cameras' FoVs are significantly overlapped, local features show a significant inter-view correlation (i.e., spatial correlation between features at the same time instant but from different cameras), as well as the usual intra-view correlation (i.e., temporal correlation between features from the same camera at different instants) [11]. Under the ATC paradigm, inter-view correspondences can be used to design collaborative strategies such as predictive [12] or distributed source coding (DSC) [13]–[16]. In the latter option, also known as distributed feature coding (DFC), each camera encodes its extracted features independently modulating the amount of coded information depending on the inter-view correlation, which is fully exploited at the receiver where a joint decoding of features is performed [17]. Indeed, previously-decoded descriptors from a generic view (either spatially or temporally adjacent) can be used as a side information (SI) to decode the currently-received information [18].

This paper elaborates upon, generalizes, and substantially revises the seminal conference results [19], based on a one-view vehicle tracking system. In this paper, we describe a smart-city context of a traffic monitoring in VSN application, whose primarily task is that of multi-view vehicles tracking at roundabouts. The proposed system respects the constrained resources of the camera sensor nodes in VSN and aims at reducing the communication burden between the cameras and the amount of transmitted data to the gathering point (sink). As a matter of fact, the multi-view vehicle tracking task is performed by the sink since it has much more processing capacity compared to the camera sensor nodes and it is much better to exploit the spatial correlation between received data from cameras with overlapped FoV at the sink node in such a way that unnecessary communication among those cameras are prevented. Moreover, we apply the ATC approach where camera nodes acquire video frames, extract relevant features compactly representing the captured data and transmit them to the sink to perform feature-based tracking. In order to reduce furthermore the amount of transmitted data, an appropriate feature selection stage is performed at camera nodes to select only features that pertain to the desired objects to track with respect to the specific application scenario (i.e., moving vehicles). The feature selection is realized by first detecting and classifying moving vehicles and then extracting features that describe those detected vehicles. To efficiently code the extracted features, we harness the existing DFC solution available from the literature. The coding system relies on the state-of-the-art findings of [18], an approach that takes advantage of data clustering to estimate and exploit the intra- and inter-view correlation among features from all cameras, and the powerful DSC technique that has been proven to reduce significantly the transmitted bitrate. The cameras are assumed to communicate with each other and with the sink in a multi-hop routing scheme. The data received at the sink are decoded and then used for further high-level visual analysis, i.e., matching features to track vehicles, whose expected accuracy is accurately tested in a real-data scenario and for imperfect camera calibration. We aim in this paper at evaluating the crucial impact of the DFC solution on the overall system performance (i.e., at the application level), as well as at testing its capability in a non-ideal scenario and in dependence on the most relevant system parameters (i.e., non ideal transmission channel conditions, bit puncturing and level of bit-rate reduction). Since we are interested in tracking vehicles, we focus on object-wise matching which consists in matching features related to the same object detected in two different views even if matched features don't represent the same part of that object. In other words, feature-to-feature matching errors (e.g., due to the noisy channel or punctured bits) can be tolerated if they belong to correctly matched objects in the two views. By focusing on the object-wise matching, we prove in this paper that further bitrate reduction can be achieved while maintaining good multi-view matching accuracy.

The main contributions of the present paper can be summarized as follows:

1) We analyze a multi-view vehicle tracking system at roundabouts for VSNs where the processing tasks are effectively partitioned between the cameras and the sink. This is achieved by using the impressive ATC technique, in conjunction with an appropriate feature selection stage that selects only those features that represent moving vehicles. Unlike many of the solutions currently available in the literature, that mainly rely on sending the entire image/video content [1], [20], [21], our approach is able to efficiently perform the target application task (i.e., multi-view vehicle tracking) while at the same time to control the transmission burden, which is a critical request for VSNs.
2) Our system efficiently exploits the multiple-camera views by performing tracking at the sink side, thus being able to exploit the inter-view correlation among cameras without the burden of inter-camera communication which consumes energy and shortens the network lifetime, unlike state-of-the-art proposals of the literature where tracking is carried out at the camera side with uniquely intra-view data [22] or with inter-view data gathered after cameras exchange some information about their obtained intra-view tracking results (e.g., single camera vehicle trajectories) [23], [24].
3) We analyse the impact of DFC at the application level, taking into consideration real-world scenario

impairments such as imperfect camera calibration and transmission errors (channel noise), by evaluating their impact on the multi-view matching accuracy, as well as bit-rate requirements.

4) Matching accuracy is evaluated object-wise in order to provide a reliable measure of the effective system performance for tracking vehicles at roundabouts, that is not captured by the literature since solutions using feature-based tracking approach are commonly proposing one-view tracking system based on the Kanade-Lucas-Tomasi Feature Tracker (KLT) which requires the intensities of the neighboring pixels of the detected feature to estimate their position each frame and to track them over time [20], [21], [25].

The rest of the paper is organized as follows. The related work is presented in Section II. An overview of the proposed system is available in Section III, while Section IV presents a number of experimental results that validate the system by suitably measuring the system performance, i.e., bit-rate reduction capabilities and feature matching accuracy, in a practical traffic monitoring scenario at roundabouts. Conclusions are finally drawn in Section V.

II. RELATED WORK

A. TRACKING/FEATURE TRACKING

Recently, several works on vision-based intelligent transportation systems have been proposed in the literature, in the context of traffic monitoring [20], [26], traffic data collection [1], and accident detection [27]. They are all based on three main steps, namely: 1) vehicle detection, 2) tracking, and 3) extraction of useful information from the tracking results.

The first step can be realized by performing image segmentation, or more specifically background subtraction, which consists in detecting the moving objects by exploiting the subtle differences between the foreground and the background. The most used algorithmic methods are: the Gaussian Mixture-based Background/Foreground Segmentation subtractor [28]; Mog2 [29], namely an improved version of the mixture of Gaussians (MoG) algorithm with better adaptability to illumination changes and shadows; the Gaussian Mixture Model (GMG) [30], which is a combination of statistical background estimation and Bayesian segmentation; and the universal video background subtraction (ViBe) [31], that, for each pixel, selects a set of background samples randomly in order to estimate the background model.

Concerning the second step, there exist four main types of object tracking: region-based, contour-based, model-based [2], and feature-based [20], [21], [32], [33]. In this paper, we are more interested in feature-based tracking since transmitting features requires less resources than the pixel presentation, and even in partial occlusion cases it is still possible to extract and to track features from the visible part of the vehicle [9]. Feature-based tracking consists in tracking features over time and then grouping the formed trajectories of features belonging to the same object in such a way that each object is represented by a single trajectory. Most approaches for feature-based vehicle tracking proposed in the literature are based on the KLT tracker [25] which consists in tracking a window of pixels instead of a single pixel, i.e., the intensity values of the pixel and its neighbours in order to estimate the new position of a feature in the next frame. This feature tracker can only be applied in the CTA approach since the intensity information is needed at the sink node.

B. CODING/DISTRIBUTED SOURCE CODING

Efficiently coding the local features is a crucial task, especially for feature-based applications that are deployable on distributed camera networks with limited resources such as VSNs. This issue has gained the interest of many researchers and several works for coding features have been proposed.

In [11], a coding scheme was proposed for encoding local real-valued features extracted from video sequences by exploiting both the intra-frame correlation (i.e., spatial redundancy between descriptors extracted from the same frame), and inter-frame correlation (i.e., temporal redundancy between descriptors from successive frames). In both cases, the coding procedure follows three steps: descriptor transform, quantization (lossy compression), and entropy coding. Other works addressed the problem of coding binary local features [34], [35]. Besides the spatial and temporal redundancy, many researchers, during the last few years, were interested in exploiting the inter-view correlation between cameras with overlapped fields of view. In this case, features are coded resorting to a reference set of features extracted from other views (other cameras). In [36], the authors proposed a solution to jointly encode local features extracted from different cameras. It consists in inter-view prediction at the encoder side by exchanging information between cameras with overlapping fields of view. In fact, one camera, chosen to be the base view, exchanges its extracted features with the neighboring cameras in order to be used as a reference set. In [12], another joint multi-view coding architecture is proposed to encode local binary features. In the latter solution, first features are matched and then, for each obtained correspondence, one feature is considered as a reference and encoded using intra-view coding technique while for the second feature only the differential residual is encoded.

Alternatively, multi-view feature coding can also be performed by applying DSC techniques which can be very beneficial particularly for networks where communication between cameras is prohibitively expensive. In [37], an unsupervised multi-view feature selection and distributed coding was proposed for a network of cameras with constrained energy and bandwidth resources. In this approach, the extracted local features are quantized using a global vocabulary (bag-of-words model) commonly known between the cameras and the decoder. In [18], authors described an architecture for inter-view feature coding using DSC that is able to guarantee further bit-rate savings. Each camera clusters and encodes its features independently of the other

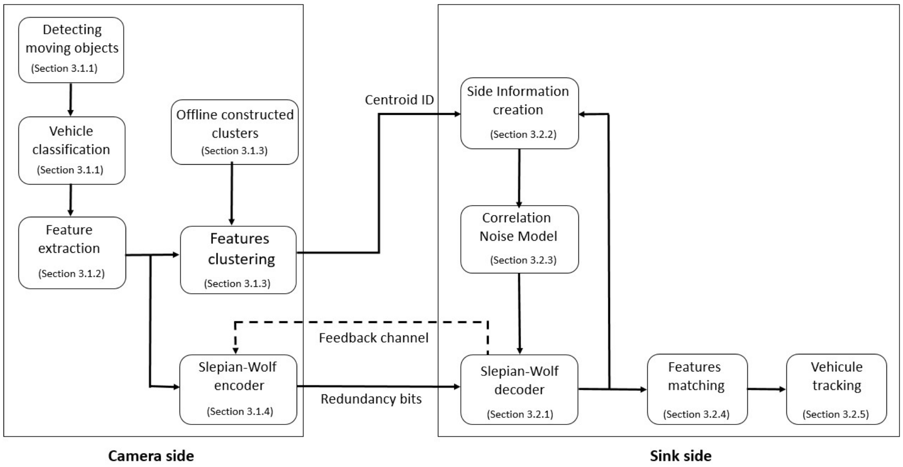

Detecting moving objects (Section 3.1.1)
Vehicle classification (Section 3.1.1)
Feature extraction (Section 3.1.2)
Offline constructed clusters (Section 3.1.3)
Features clustering (Section 3.1.3)
Slepian-Wolf encoder (Section 3.1.4)
Centroid ID
Feedback channel
Redundancy bits
Side Information creation (Section 3.2.2)
Correlation Noise Model (Section 3.2.3)
Slepian-Wolf decoder (Section 3.2.1)
Features matching (Section 3.2.4)
Vehicule tracking (Section 3.2.5)
Camera side
Sink side

FIGURE 1. System diagram.

cameras using channel coding schemes (systematic linear channel codes). Then the obtained parity bits are forwarded to the decoder where decompression and error correction are performed. A side information is used at the decoder to predict the decoding of the received descriptor. In this paper we work upon the state-of-the-art solution [18], and investigate its performance in the specific application of multi-view vehicle tracking at roundabouts, with a specific view on both bit-rate reduction and feature matching performance.

III. SYSTEM DESCRIPTION

We consider a VSN connecting multiple cameras with overlapped FoV to a sink node through low bandwidth links. The system employs an ATC paradigm where each camera collects visual data and extracts a set of local features. Since the camera nodes in the VSN have limited computational and energy resources, performing all the tasks related to vehicle tracking at the camera level is very challenging. Hence, these tasks are divided between cameras and the data gathering point. The proposed feature-based tracking system is mainly composed of three essential modules: 1) vehicle detection and feature extraction, 2) feature coding/decoding and 3) features matching and vehicle tracking. The cameras, in this case, perform the two first modules (vehicle detection and classification, feature extraction and clustering, and feature coding) while the sink performs feature decoding and the last module. Figure 1 depicts the global system architecture for the proposed multi-view features coding and tracking which will be explained in more details in the following.

A. SMART CAMERA NODES

Camera nodes (see Figure 1) are responsible of data acquisition, detecting moving objects and classifying them into one of the five possible classes: car, small truck, big truck, bus, and motor bike. Note that in case objects do not belong to the mentioned classes, they are skipped so that only the elements of interest are transmitted to the sink node [19]. Then, a set of visual features representing the moving vehicles are extracted and clustered into non-overlapping regions (see Figure 2). Finally, the cameras encode their extracted features independently and transmit the resulting data to the sink together with the feature's cluster ID.

1) VEHICLE DETECTION

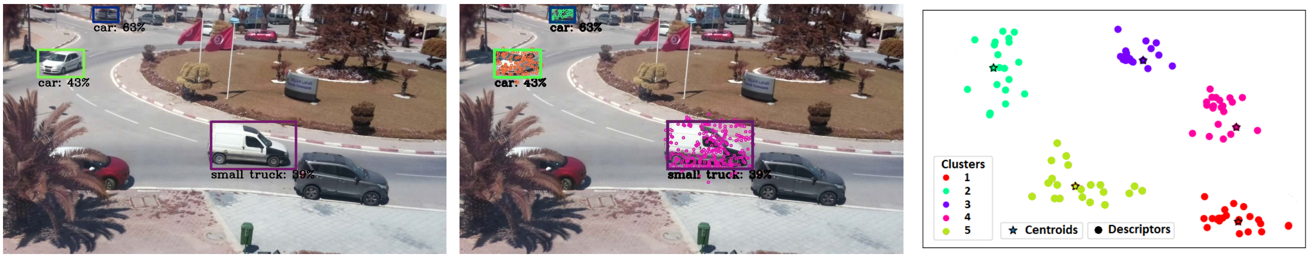
The first step to detect vehicles is image segmentation, which involves identification of moving objects at each frame. The chosen approach is background subtraction by performing the MoG subtractor [28], which is an efficient solution for modeling fast changing in illumination, providing a good compromise between the consumed computing resources and the achieved segmentation precision and for treating repeated camera shaking. Morphological transformations on the obtained binary image are also used to refine the detection. At the end of this step, any moving object is extracted including vehicles, moving trees by the wind, and pedestrians.

With the intention of selecting only moving vehicles, a verification step is added to classify each detected object. To accomplish this task, camera nodes apply the `YOLOv2` detector [38] right after the background subtraction and the morphological transformations. `YOLOv2` is a real-time object detector and classifier based on 19 convolutional neural network layers. The algorithm is trained on a huge data set collected from camera sensors (see details in Section IV-A) in order to classify objects in the images into five classes of vehicles: cars, small trucks, big trucks, motor bikes and buses. After training, the `YOLOv2` algorithm is able to detect every object in the image belonging to one of the five classes.

Figure 2(a) illustrates an example of vehicles detection and classification result where the label and the classification's confidence percentage are shown for each vehicle.

2) FEATURES EXTRACTION

To reduce the transmitted bit rate, the ATC paradigm is applied in conjunction with appropriate feature selection



(a) Vehicle detection and classification **(b)** SURF features detection **(c)** Features clustering

FIGURE 2. Pictorial representation of the tasks performed at the camera nodes.

stage, which implies that only features about the detected moving vehicles are sent to the sink node (see Figure 2(b)). Local feature vectors are a compact representation of the local content of an image patch that differs from its immediate surroundings by means of texture, color or intensity. As features are capable of describing the main characteristics of vehicles, the receiver can efficiently identify and track vehicles by simply estimating the trajectory of the received feature points from one frame to another [25]. Since an effective and robust algorithm is needed to extract relevant features capable of representing vehicles even in the presence of partial occlusion, illumination and pose change, the real-valued feature extractor speeded-up robust features (SURF) [39] is selected for its robustness, speed and low computational requirements, thus generating descriptor vectors of 64 real-valued elements.

3) FEATURES CLUSTERING

In furtherance of accurately exploiting the correlation among extracted features for the DSC [18], we classify features into groups of strong likelihood in such a way that the features belonging to the same cluster have similar descriptors (i.e., they are correlated features). Clusters must cover all possible descriptors that might represent a vehicle and each cluster is identified by a centroid. Once the clusters are defined (the output of offline constructed clusters block of Figure 1), we assume that the set $\mathcal{A}$ of centroids and the centroids' IDs are a common knowledge between both the cameras and the sink. The clustering operation is similar to vector quantization since it allows each descriptor to be coded in a very compact yet efficient way by exploiting the correlation between the descriptor and the centroid or between descriptors belonging to the same cluster. A reliable way to identify set $\mathcal{A}$ is by exploiting the `K-means` clustering algorithm that aims at grouping data points into K clusters by reducing within-cluster variances [40]. For the initialization process, the `K-means++` approach [41] is selected.

Overall, the process of feature clustering is illustrated in Figure 2(c) where the clusters and the corresponding allocated features are represented in descriptor space after dimensionality reduction using the principal component analysis (PCA) technique. For illustration purposes, we show in Figure 2(c) only 5 clusters out of 1000 and a limited number of features per cluster ($\leqslant$20).

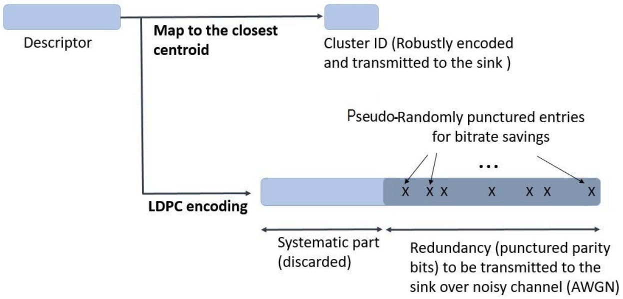

Descriptor
Map to the closest centroid
Cluster ID (Robustly encoded and transmitted to the sink)
Pseudo-Randomly punctured entries for bitrate savings
...
X X X X X X X
LDPC encoding
Systematic part (discarded)
Redundancy (punctured parity bits) to be transmitted to the sink over noisy channel (AWGN)

FIGURE 3. DSC approach at the smart camera side.

4) DISTRIBUTED SOURCE CODING (DSC)

Once features (i.e., real valued descriptors) are extracted, a quantization step is performed to the aim of converting them to binary vectors. We use a uniform scalar quantizer with 100 levels (8 bits) and a step size $\Delta = 0.02$. Each SURF descriptor with 64 float elements is therefore quantized into a binary vector of 512 bits, then encoded using the Slepian-Wolf (SW) approach [17]. In our proposed system, we implement a DSC encoder using a (6,4) regular systematic low-density parity-check (LDPC) encoder of rate 1/3. The length of the encoded vector is $n = 1536 = k \cdot 3$, with $k = 512$ bits the size of each descriptor. To construct the parity check matrix H we harnessed the predefined function `parity_check_matrix` from the `pyldpc` library in `python`, which builds a regular Parity-Check Matrix following Gallager's algorithm [42]. For each encoded descriptor, the systematic part is discarded and only the parity information as well as the cluster ID to which the descriptor belongs are transmitted to the sink node, as illustrated in Figure 3. To evaluate the bit-rate savings that can be achieved using the DSC technique, a number of bits from the redundancy part are pseudo-randomly punctured. We define the fraction of punctured bit, which is equivalent to the fraction of reduced bit-rate, as

$$\rho = \frac{\text{Number of punctured bits}}{\text{Total number of parity bits}}. \tag{1}$$

B. SINK NODE

Once the data gathering point (sink) receives the transmitted features, it uses the available information, also named side information SI (i.e., the clusters' centroids and the already decoded descriptors), in order to decode the descriptor from the received redundancy bits corrupted by noise. The approach is a standard LDPC decoder based on belief propagation (BP) algorithm, whose parameters are carefully set by the correlation noise model (CNM) block of Figure 1. Once features are successfully decoded, the sink identifies objects to track and estimates their trajectory over time by exploiting the multi-view information.

1) SLEPIAN-WOLF DECODING

The decoding process (SW decoder block in Figure 1) is based on the idea that the systematic part, which is not transmitted, is replaced by a noisy counterpart given by the SI. This can either be the cluster centroid, retrieved from the transmitted cluster ID, or already decoded descriptors available from multiple views (see details in later Section III-B2). The received bits corrupted by noise, and the SI, are then used to build the log-likelihood ratios (LLRs) that are needed to run the BP algorithm [43]. LLRs are differently constructed for the redundancy bits and for the systematic part. For the redundancy part we assume a channel with an additive white Gaussian noise (AWGN) and have [44]

$$\mathrm{LLR}_{\mathrm{red}} = \frac{2r}{\sigma^2}, \tag{2}$$

where σ^2 is the noise variance and r is the received vector corrupted by noise. The bits that were punctured (i.e., not transmitted) are simply set to LLR value zero. For the systematic part we model noise through a binary symmetric channel (BSC), for which we have [44]

$$\mathrm{LLR}_{\mathrm{sys}} = \log\left(\frac{1-p_{bit}}{p_{bit}}\right) c + \mathrm{LLR}_{\mathrm{apriori}}, \tag{3}$$

where p_{bit} is the average bit error probability, c is the BPSK map of the binary SI vector, and the *a priori* contribution is entry-wise modeled as $\mathrm{LLR}_{\mathrm{apriori}}(n) = \log((1-p_{n0})/p_{n0})$ where $p_{n0} = p(d_n = 0)$ is the probability that the n^{th} descriptor element d_n is equal to 0. All parameters are set in the CNM estimation block (see later Section III-B3 for details). The maximum number of iterations of the BP algorithm is fixed to 100 for quasi-optimal performance. If the maximum number of iterations is achieved before convergence, then sink has to request for more parity bits from camera through a feedback channel.

2) SI CREATION

By relying on the assumption that the clusters' representative vectors are known at the receiver side, the received centroid ID is used to retrieve the corresponding centroid, in its binary presentation, and to create the SI [18]. In our multi-view setup, the SI is the set of all already successfully decoded descriptors from current and previous frames and from all cameras belonging to the cluster identified by the received centroid ID, plus the binary centroid vector. In this way we make a good use of the statistical correlation between descriptors in the buffer and the target one, i.e., we exploit both the intra-correlation among features from the same camera through consecutive time frames and the inter-correlation among features from different cameras with overlapped FoV.

3) CNM ESTIMATION

Building a good statistical model that accurately depicts the correlation noise between the SI and the true descriptor is a crucial phase in the DSC procedure, which should accurately identify parameters p_{bit} and $p(d_n = 0)$ to be used in (3). Since a cluster groups features that are very similar, a precise CNM can be built. Specifically, we consider two cases, namely:

- One-view decoding mode: Whenever there is no available already decoded descriptors in the buffer, we rely on the cluster centroid c, and set p_{bit} to the average distance between the corresponding centroid and all the descriptors belonging to that cluster that comes from the correlation statistics provided by the offline feature clustering operation. We also set $p(d_n = 0) = \frac{1}{2}$, so that the a priori LLR in (3) is set to $\mathrm{LLR}_{\mathrm{apriori}} = 0$.
- Multi-view decoding mode: We adopt the CNM presented in [18] for which $p(d_n = 0) = \frac{N}{N_{SI}}$, where N is the number of times n^{th} element of descriptors from SI are equal to zero, and N_{SI} is the total number of descriptors within the SI, and we set $p_{bit} = \frac{1}{2}$ in such a way that the contribution of the centroid c alone in (3), which does not represent the full created SI, is not used.

4) FEATURES MATCHING

Feature matching implies a number of concurrent actions that are meant to strengthen the tracking performance. Observe that all the proposed criteria for selecting correct matches work on objects (i.e., cars, trucks, etc.) in such a way to enhance their detectability and their trajectories estimation accordingly to the reference application scenario. We call this approach **object-wise** matching. The concurrent actions are:

1) *intra-view matching*, that is, the idea of detecting and recognizing the same objects over time on each view which is essential for constructing the objects' trajectories on each camera separately, and
2) *inter-view matching*, that is, to identify the objects presence on multiple views in order to switch and continue tracking the same objects from one camera to the other which can be very useful to solve practical problems such as occlusions (see Sect. III-B5).

We detail them in the following.

a: INTRA-VIEW MATCHING

For the current frame i, the sink matches each grouped features (i.e., features belonging to the same bounding box and thus presenting the same object) with features from the previous frame $i-1$. Each object from frame i should be matched with at most one object from the frame $i-1$. If one object has feature matches with features corresponding to two

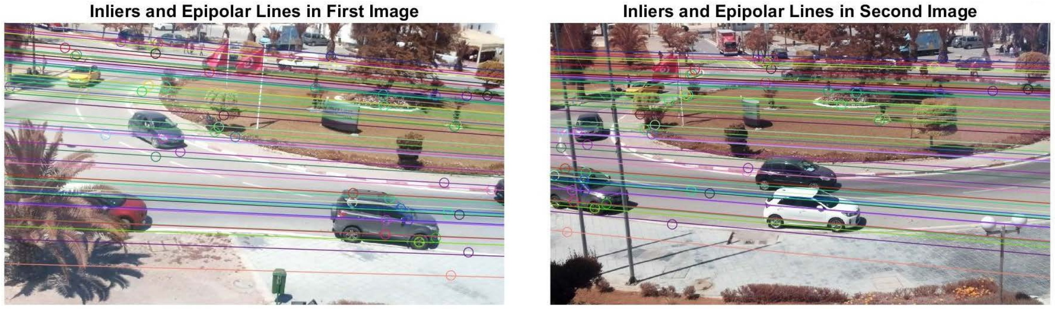

Inliers and Epipolar Lines in First Image
Inliers and Epipolar Lines in Second Image

FIGURE 4. Epipolar lines obtained by the estimated fundamental matrix **F** and the point correspondences (inliers).

or more different objects in frame $i-1$, then the selection of the best match is based on the number of matched features and their distances. In fact, the object with the higher number of matched features and the lowest distances between feature descriptors is considered as the best match and therefore, the remaining feature matches with the other objects are considered as wrong matches and they are eliminated.

In this paper, the k-nearest neighbors Brute Force (`Knn BF`) with $k = 2$ is used to match features. The algorithm is implemented in the Open computer vision (OpenCV) library [45]. The Lowe's Ratio test is performed to select the correct matches. Once matching is completed successfully for each object in frame i and all wrong matches are discarded, features trajectories from frame $i-1$ to frame i can be estimated.

b: INTER-VIEW MATCHING

For the current view v, the sink matches features representing each detected object with features from another view w. Each object from view v should be matched with at most one object from view w. Similarly to the intra-view matching, `Knn BF` and the Lowe's Ratio test are applied to select matches.

A further step is added to improve the multi-view matching accuracy which consists of exploiting the *epipolar* geometric relations [46] between the two cameras (relations between the 3D points and their projections onto the 2D images in both cameras, illustrated in Figure 4) to select the correct matches. These relations lead to some constraints between the image points, namely that a point in one view is constrained to a line, named the epipolar line, in the other view. The fundamental matrix **F** is the algebraic representation of such epipolar geometry.

In the case of non calibrated cameras, the fundamental matrix can be approximately estimated, independently of scene structure and the cameras' internal parameters, by using at least eight correspondences of imaged scene points. The resulted system of equations for the points correspondences is a linear least square problem that can be solved using the Eight-Point algorithm [47]. In the proposed system, the fundamental matrix **F** is estimated for two non calibrated cameras by considering a number of point correspondences that are selected manually. The predefined function `findFundamentalMat` from OpenCV [45] is used to compute the matrix **F** with input the point correspondences. Once the fundamental matrix **F** is computed, the epipolar lines are estimated using the predefined function `computeCorrespondEpilines` in OpenCV.

The fundamental matrix **F** and the epipolar lines are estimated offline and are assumed to be known at the sink node. Since the cameras are assumed not to be moving, they are computed only once.

5) VEHICLE TRACKING

One-view object tracking can be defined as the process of identifying and locating the object from one frame to the other from the same view as it moves in the scene, and then estimating its trajectory over time by simply performing intra-view matching of features representing that object as described in Section III-B4. As a matter of fact, each object is represented by a number of extracted features, and therefore, the object is tracked by grouping trajectories formed by these features over time. One-view tracking can be performed for each view simultaneously and independently.

The vehicle tracking module of Figure 1 performs also multi-view tracking to cope with one-view tracking loss of an object at a certain camera, i.e., a so far tracked vehicle up to the previous frame is not detected at the current frame due to partial or total occlusions or failure at detecting or classifying the object at the camera side. Indeed, for each view, upon receiving features that represent a new object to track, the system performs inter-view matching (i.e., matching the object's features from the current view with features from other available views as described in Sect. III-B4), and if a potential matched object from a second view is found for more than three consecutive frames then it is considered as correct matching and this information is stored in a database to be used later when multi-view tracking is needed. Once one-view tracking of an object fails, the system refers to the reconstructed database aforementioned and check if the corresponding object in the other view can still be located and thus switches and continues the tracking in this second view. The probability of finding another view with better sight and tracking conditions of the same object increases accordingly to the number of cameras with overlapped FoV.

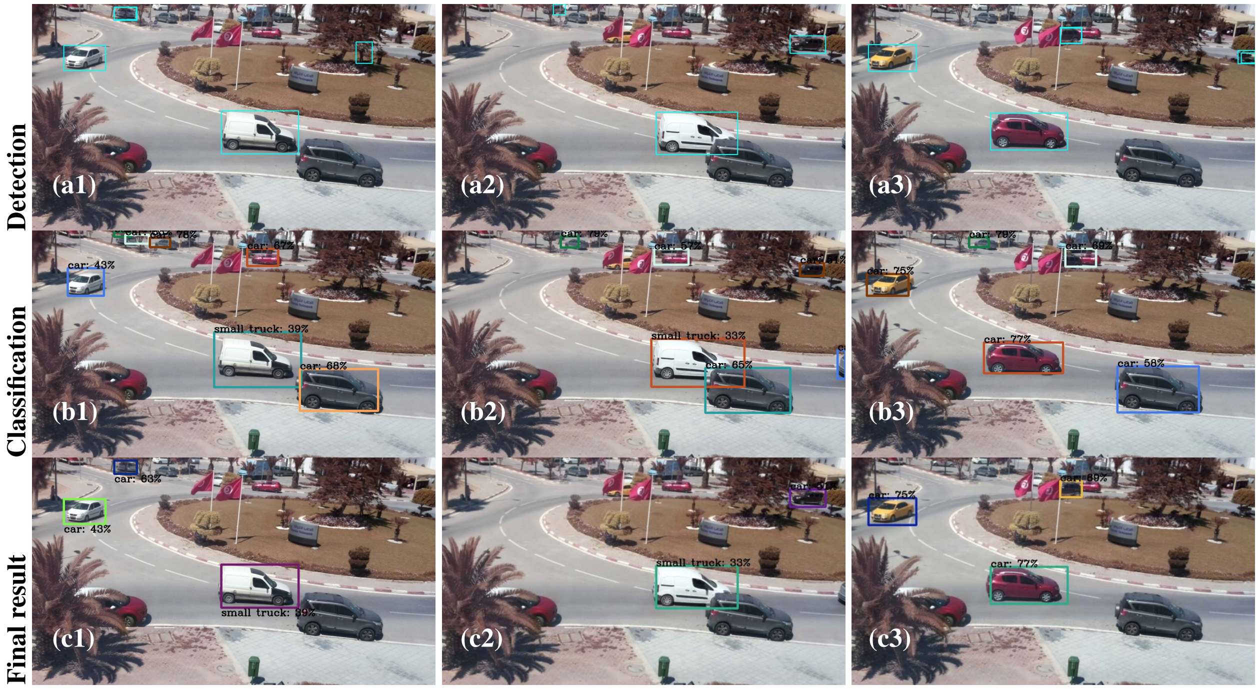


FIGURE 5. Vehicle classification: Detection and classification examples.

IV. EXPERIMENTAL RESULTS

In this section we test the proposed system on a real-case scenario, where video sequences were captured by two camera sensor nodes with overlapped FoVs, using `Pi NoIR camera v2` sensors [48] connected to `raspberry Pi 3 model B` boards [49]. Performance is evaluated for all the constituent modules of the proposed system (vehicle classifier, offline features clustering, multi-view distributed feature decoder, and multi-view vehicle tracking) in order to cover both the DSC capabilities, as well as the impact of the designed solution at the application level.

A. TRAINING AND TEST DATABASES

Considering the specificity of our multi-view vehicle tracking system at roundabouts, a specific database had to be built (see details on it in [19]). Data was collected at a roundabout in front of SUP'COM university in the city of Al Ghazela, Tunisia. The constructed data is termed SupCom roundabout database (SR) and consists of two sets of images:

- a *training set*, which was mainly used for training the vehicle detection and classification module, and which is a collection of 2058 images from videos of 3 minutes long captured by 10 camera sensor nodes. In order to cover a large number of possible vehicle poses at the roundabout, the cameras were placed in such a way that they capture the same scene from 7 different views and 2 different heights. For strenghtening the vehicle detection training purposes, 4218 images gathered from the database collected in the CBCL StreetScenes Challenge Framework [50] and the voc 2007 train databases [38] were added to the set.
- a *test set*, which is used to evaluate the classifier as well as the tracking, is instead constructed from videos of 3 minutes in length with 25 fps captured simultaneously from two camera sensors with overlapped FoV.

B. VEHICLE DETECTION AND CLASSIFICATION

The very first step in our system is vehicle detection and classification (see Figure 5), which is a very important step for the accuracy of the tracking and traffic data extraction.

The algorithm was tested on images from the SR test set for different values of threshold ranging from 10% to 90% [19]. A threshold is the minimum percentage of confidence for classifying an object to a certain class. For each confidence threshold, the true positive rate (TPR) and the false positive rate (FPR) are computed. The TPR, termed also as the recall, is the sensitivity of the classifier to correctly identifying as much as possible objects (i.e., probability of detection). The FPR, on the other hand, represents the proportion of wrongly identified objects to a certain class (i.e., probability of false alarm). In Figure 6, a receiver operating characteristic (ROC) curve is measured by expressing the TPR as a function of the FPR for all thresholds as the threshold varies.

In the case of classifying an object to the "car" class, the TPR is defined as the percentage for correctly classifying a car object to the class while the FPR is identified as the percentage of wrongly classifying an object which is not a car to the "car" class. The resulting ROC curve is illustrated in Figure 6 along with the ROC curves of both the random and perfect classifiers. The ROC curve is a way to choose the ideal threshold to make predictions that is a trade-off between

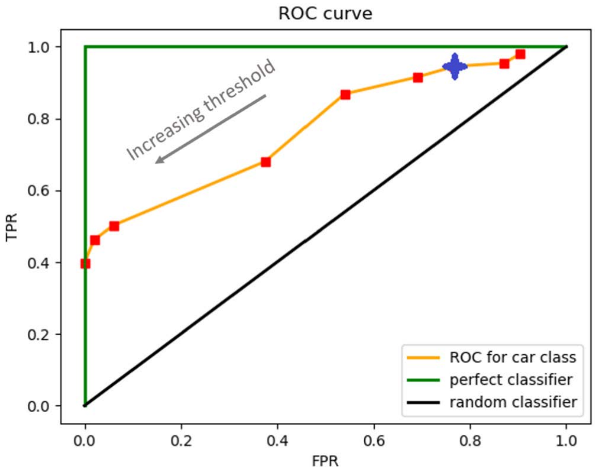


FIGURE 6. Vehicle classification: Receiver operating characteristic curve for the "car" class.

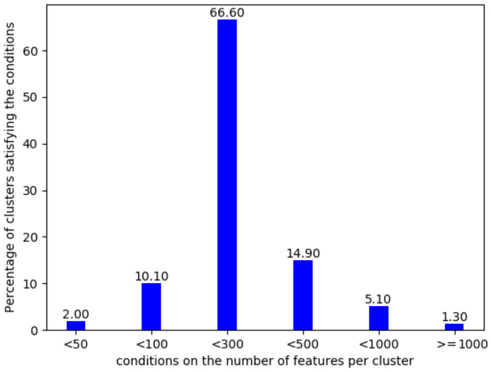


FIGURE 7. Offline constructed clusters: Features distribution within the obtained clusters from the K-means algorithm.

the TPR and the FPR. The higher is the threshold, the fewer positive predictions and the more negative predictions are made.

Considering that we are interested in tracking vehicles, detecting and classifying all existing vehicles each frame is crucial. In other words, it is more important to reduce the false negative predictions and to increase the TPR than reducing false positive predictions. In fact, incorrectly classifying an object (e.g., a person) to the "car" class is a situation that can be solved since the probability of making the same false positive prediction for more than two or three consecutive frames is very low. For this reason, we choose to work with a confidence threshold resulting in a very high TPR value as a sufficient accuracy model of classification even if the obtained FPR is slightly high. The chosen point marked by a star in Figure 6 corresponds to 30% threshold of confidence for which the TPR and FPR values are respectively 0.945 and 0.767.

The results of vehicle detection and classification are illustrated in Figure 5. The rows (a) and (b) in Figure 5 respectively represent the moving objects detected by the MoG algorithm and the classified objects by the YOLOv2 classifier for three examples. The last row, (c), instead, represents the final result which is the combination of the MoG and the YOLOv2 algorithms in order to detect moving vehicles. In all the three examples given in Figure 5, stationary vehicles are detected by YOLOv2 as it can be seen in row (b). Yet, combined with the MoG subtractor, they are eliminated in the final result. Thus the desired aim from combining the two algorithms is achieved. However, in the cases of partially occluded cars, the YOLOv2 classifier succeeded to detect and classify the car hidden by the flag correctly in image (b3) with 69% confidence but failed to detect the far car hidden by the tree in the top of picture (b2). A failure in the vehicle detection and classification step is critical to the one-view tracking results as it implies loss in track of the vehicle over time. Nevertheless, this problem can be solved in the context of multi-view tracking by exploiting information from other cameras with overlapped FoV where the same vehicle is seen and detected successfully.

C. OFFLINE CLUSTERS CONSTRUCTION

The feature clustering step is performed offline by grouping 237759 descriptors from 1750 images taken from the SR database [19]. We set the number of clusters to $K = 1000$. Figure 7 depicts the distribution of features within the obtained clusters. The vast majority of clusters (98%) contain at least 100 features.

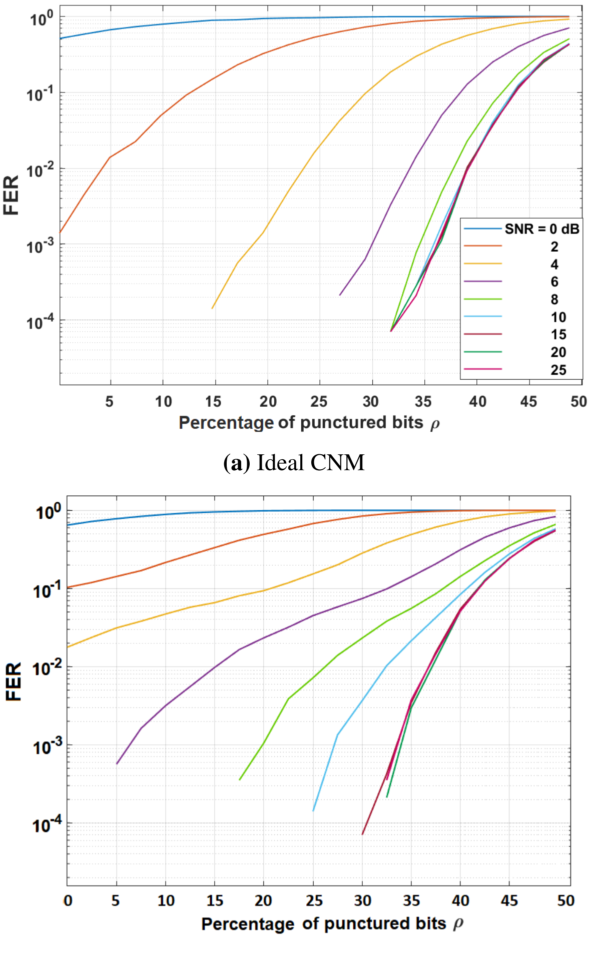
D. ERROR RATE OF THE SW DECODER

The distributed feature decoder is evaluated in two different scenarios:

- *Ideal CNM*, where we assume that the decoder has a complete knowledge about the exact likelihood between the created SI and the descriptor to decode, and thus an ideal CNM is estimated.
- *Practical CNM*. The ideal CNM can not be known by the decoder in practice, therefore, a second scenario is considered for which an approximation of the correlation between the SI and the received feature is proposed as described in Section III-B3.

Figure 8 depicts the frame error rate (FER) evaluation for different values of puncturing fraction ρ and for signal to noise ratio (SNR) values ranging from 0 to 25dB. For the sake of simplicity, only the FER evaluation is presented in this paper since exactly the same considerations, and similar interpretations, can be inferred from the bit error rate (BER).

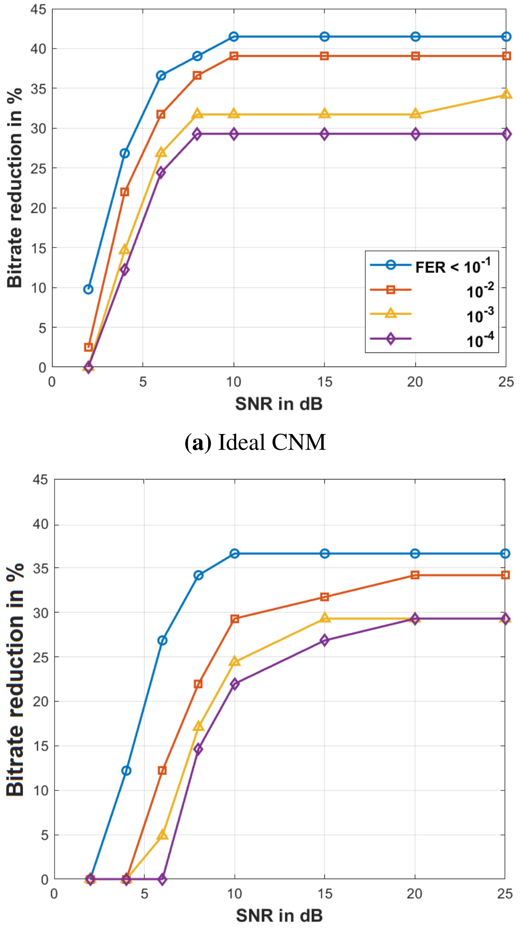
Interestingly, the practical CNM in Figure 8(b) has similar behavior to the ideal CNM in Figure 8(a), but with higher FER values and hence lower bit-rate savings. However, as the SNR increases, the gap between the two decreases, especially for SNR $\geqslant$ 15dB since we can observe a saturation for high SNR values. Starting from SNR equals to 8dB, a significant number of bits can be saved without deteriorating the decoder's efficiency (fraction ρ values starting from 29.3% and 14.6% in the ideal and practical cases, respectively). Yet, if we increase the fraction ρ significantly, i.e., greater than or



(a) Ideal CNM

(b) Practical CNM

FIGURE 8. SW decoder: FER as a function of the fraction of punctured bits ρ for different SNR values.



(a) Ideal CNM

(b) Practical CNM

FIGURE 9. SW decoder: bit-rate savings achieved for different FER thresholds.

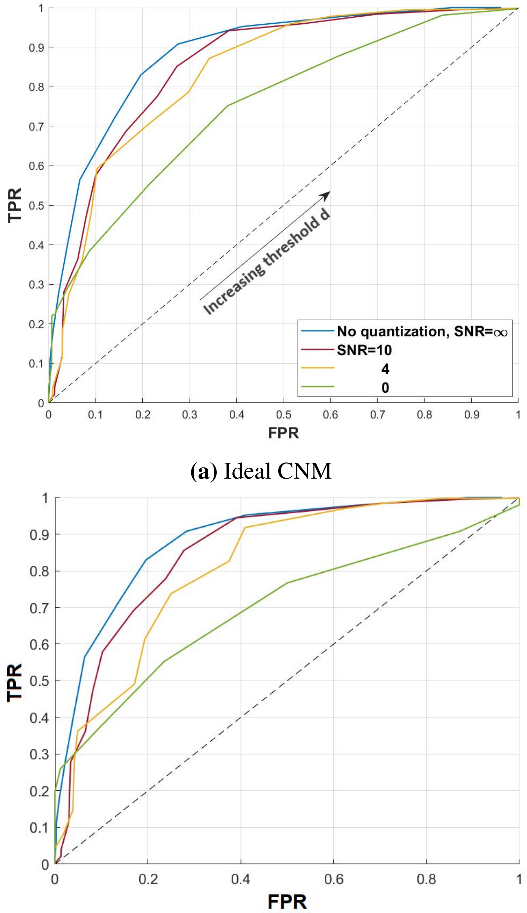
equal to 45%, the performance of the decoder breaks down and we get high FER values even for high SNRs.

In Figure 9, we provide a further insight by illustrating the bit-rate savings achieved by exploiting both the intra and inter-camera correlation at the decoder side for both the ideal and practical CNM cases. The bit-rate saving is inspected for different values of SNR and different FER thresholds under which we consider a decoding is done successfully. Raising the FER threshold entails more bit-rate savings due to the fact that we are tolerating more decoding errors. The maximum bit-rate reduction attained at high SNRs ranges from 29.3% to 41.5% in the ideal case, and from 29.3% to 36.62% in the practical case. In other words, the decoder in the ideal CNM case results in an average of 2.44% more bit-rate savings compared to the practical CNM case. As the SNR decreases, the channel becomes more noisy and the bit-rate saving deteriorates.

E. EFFECT OF DECODING ON MATCHING ACCURACY

To assess the matching accuracy of the recovered descriptors after decoding, the ROC curves were computed and illustrated in Figure 10 for the ideal and practical CNM case scenarios. We consider a true positive TP as a correct matching of features belonging to the same vehicle seen in the two views and a false positive FP as matching between features belonging to different vehicles. Hence, two features are considered correctly matched if they belong to the same vehicle even if they don't actually represent the same part of that vehicle (object-wise matching). Figure 10 depicts the ROC curves, under a 30% puncturing and for different SNRs, for both the ideal and practical CNM case scenarios, whereas Figure 11 depicts the ROC curves for different puncturing fraction ρ and for an SNR of 10dB in the best case scenario. Bearing in mind the aforementioned saturation of the multiview feature decoding at high SNRs, the ROC curves for SNRs greater than 10dB are exactly identical to the one obtained for SNR = 10dB.

Also observe the blue ROC curves which are equivalent to the performance of matching the original extracted descriptors (i.e., the ROC curves under no quantization, no puncturing and an ideal channel with SNR $= \infty$), whose controlled gap with the red ROC curves (i.e., under quantization, 30% puncturing and noisy yet sufficiently good



(a) Ideal CNM

(b) Practical CNM

FIGURE 10. **Feature matching: Object-based matching accuracy for both ideal and practical CNM scenarios, in dependence of the SNR value, and for $\rho = 30\%$ puncturing percentage.**

channel with SNR = 10dB and FER = 0) shows the limited impact of quantization on the system performance. The lower the SNR, the lower the matching accuracy we get because of the increased errors at the decoding step (increased noise in the communication channel) as it can be seen from Figure 10. Moreover, the ROC curves in the practical CNM case have similar behavior to those obtained in the ideal CNM case and hence, the same observations made previously also apply to the practical CNM scenario.

Similarly, when the puncturing fraction ρ increases, more decoding errors are generated and thus the matching accuracy is progressively deteriorating (see Figure 11). Yet, since the gap between the ROC curves obtained for the two values of the fraction $\rho = 30\%$ and 50% is small, we can reduce significantly the bit-rate while still maintaining an acceptable matching accuracy.

An example of multi-view feature matching when SNR = 10dB and $\rho = 50\%$ is provided in Figure 12 where two special examples are shown: example 1 is when a vehicle is partially occluded in one view (vehicle obj2 in view 2) and example 2 is a variation of the vehicle's pose from one view to the other (vehicle obj1 in both views). As can be easily seen, we can still match the vehicles correctly in both cases even though few features per object are matched due to corruption of reconstructed descriptors caused by decoding errors.

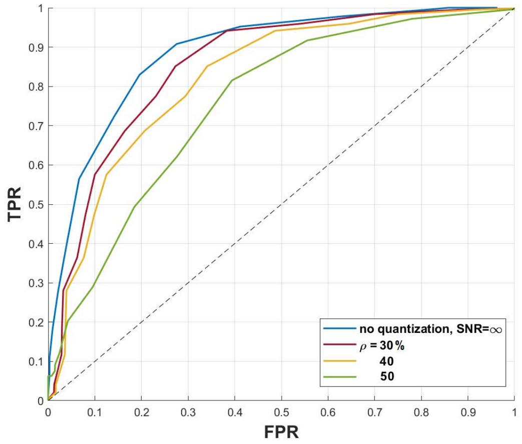


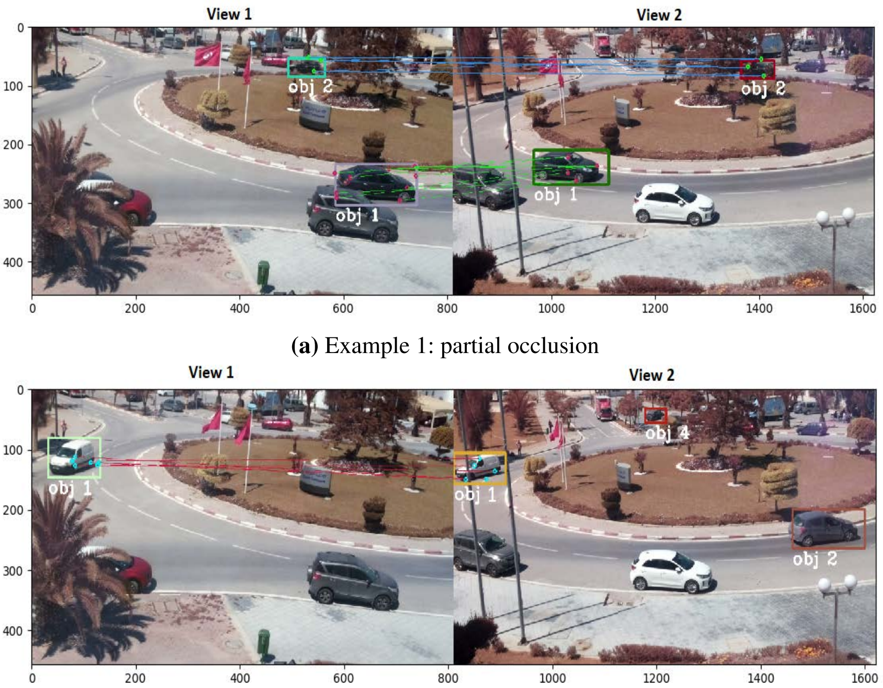
FIGURE 11. **Feature matching: ROC curves for different puncturing fraction ρ in the ideal CNM scenario.**

F. OBJECT-WISE MATCHING

Interestingly, the gap between the ROC curves obtained at high SNRs and at SNR = 0dB is small despite of the tremendous value of the FER at SNR = 0dB. This can be explained by considering the fact that we are presenting the accuracy of object-wise matching instead of feature-wise matching as aforementioned. In this context, Figure 13 shows the ROC curves obtained for applying object-wise matching (straight lines) against ROC curves in the case of feature-wise matching (dashed lines) of descriptors from our data set. In the latter case, a couple of descriptors correspondence is considered as correct match only if they represent the same part of the same object. As can be easily seen, when the feature-wise matching is performed the performance is deteriorated compared to the one obtained for the object-wise matching. This deterioration is due to the fact that we are considering non calibrated cameras or imperfect stereo camera calibration. In other words, we are slightly reducing the precision on selecting correct feature matches (features level) in order to improve the multi-view matching accuracy as long as we are matching correctly the same vehicles from both views (object level).

G. WORKING POINT IN THE ROC CURVE

Incidentally, each point on the ROC curve is the couple (TPR, FPR) computed for a specific discrimination threshold d. The latter denotes the maximum distance between two features considered correctly matched. In the case of multi-view object tracking, it is important to have high TPR and low FPR values to ensure that we are tracking the object correctly from one view to the other. As shown in Figure 10, when the threshold d is small, the selectivity of the matching function is at its highest values which means that only few



(a) Example 1: partial occlusion

(b) Example 2: variation of pose

FIGURE 12. Examples of multi-view features matching in the case of SNR = 10dB and $\rho = 50\%$.

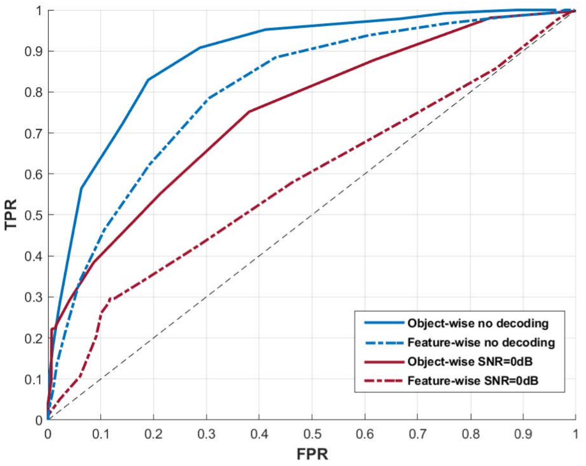


FIGURE 13. Object-wise against feature-wise matching accuracy evaluation.

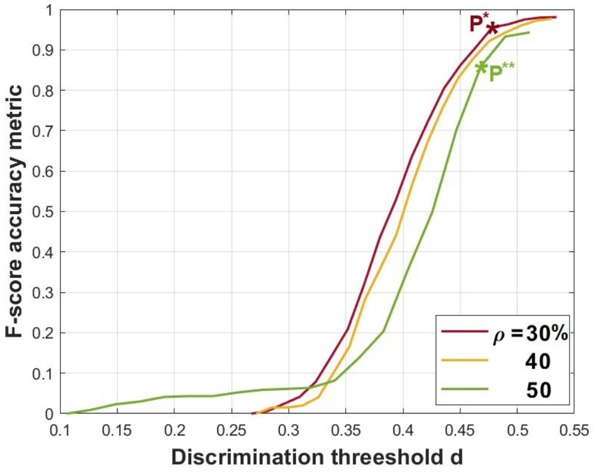


FIGURE 14. Matching accuracy metric F-score as a function of the discrimination threshold d for different puncturing fraction ρ.

matches with great resemblance (i.e., with distance smaller than d) are selected as TP and few FP are detected (low FPR values). But in this case the number of false negatives is high resulting in low TPR. With the increase of the discrimination threshold d, the matcher gets more sensitive to TP detection leading to higher TPR values at the price of an increased FPR. In fact, a high threshold d entails low specificity of the matching selection and thus more FP detection. In this case, an object from view 1 can be matched to a wrong object in view 2 leading to wrong tracking and information extraction. Hence, it is crucial to find a trade-off between the TPR and the FPR.

For the sake of simplicity, we illustrate a method to choose the discrimination threshold and the corresponding accuracy value by taking SNR = 10dB as an example. To measure the matching accuracy at each threshold point d, we used the F-score which is the harmonic mean of the precision P and recall R, that is

$$F = 2\frac{P \cdot R}{P + R}, \quad P = \frac{TP}{TP + FP}, \; R = TPR. \tag{4}$$

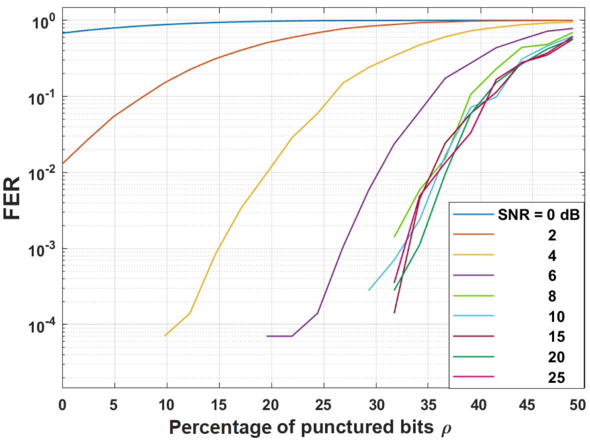
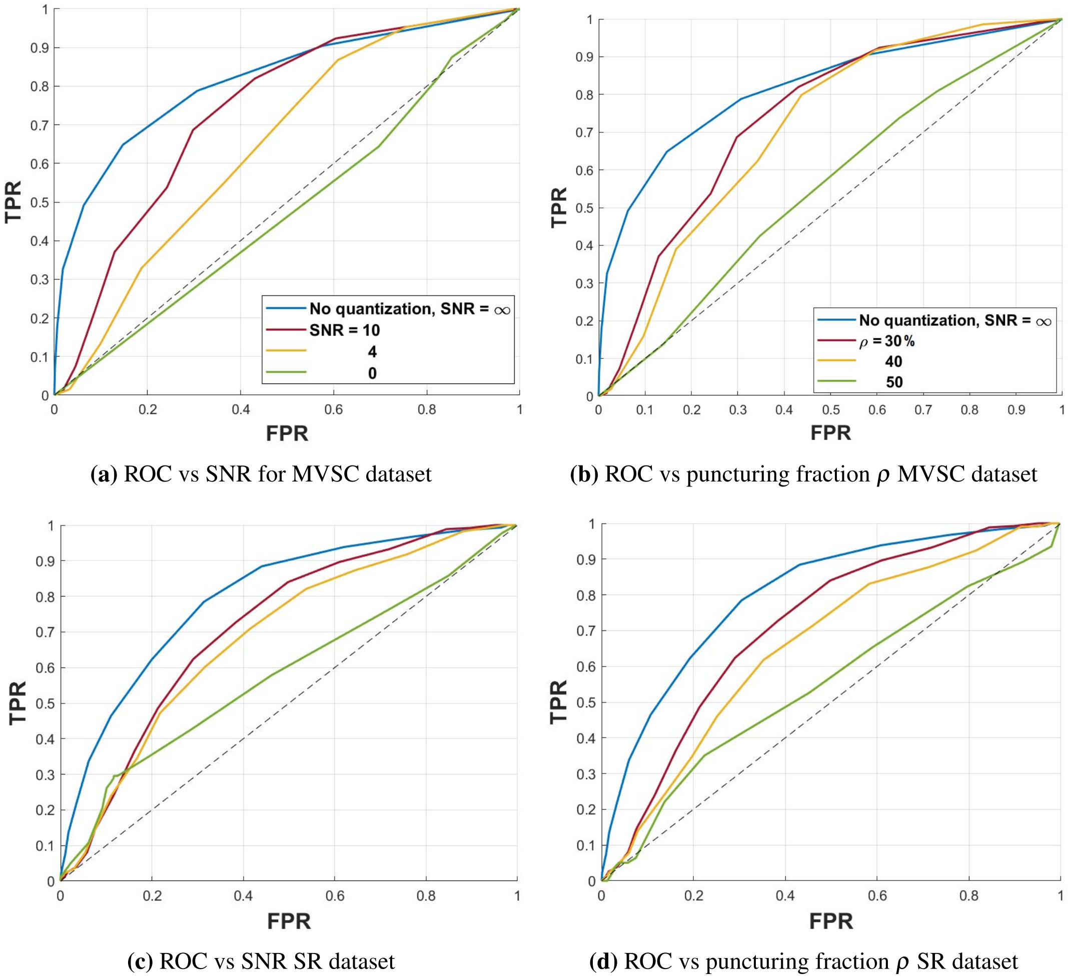


FIGURE 15. FER performance on the MVSC dataset, to be compared with Figure 8.

Figure 14 shows the evaluation of the matching accuracy for different puncturing fraction ρ by measuring the F-score as a function of the discrimination threshold d. As the threshold d increases, i.e., the TPR gets higher, the F-score increases as well. However, increasing the threshold d too much leads to high F-score values at the price of very high FPR values. Having regard to the aforementioned considerations, we chose a discrimination threshold for which a good matching accuracy ($F \geqslant 0.85$) and a low FPR (FPR $\leqslant 0.5$) are achieved. For the ROC curve obtained at SNR $=$ 10dB and $\rho = 0.3$, we chose the threshold $d^* = 0.4782$ which corresponds to the triplet (TPR$^* = 0.942$, FPR$^* = 0.383$, $F^* = 0.955$). Interestingly, an F-score value close to $F^* = 0.955$ obtained for SNR $=$ 10dB and $\rho = 30\%$ can be achieved when the puncturing fraction ρ is equal to 50% if $d^{**} = 0.4682$ is chosen as discrimination threshold. The latter corresponds to the triplet (TPR$^{**} = 0.917$, FPR$^{**} = 0.5$, $F^{**} = 0.859$).

To summarize, from what previously stated, we can infer that for any puncturing fraction ρ between 30% and 50% we can find a discrimination threshold d at which we can achieve significant bit-rate savings greater than or equal to 30% while preserving a good matching accuracy ($0.859 \leqslant F \leqslant 0.955$) when SNR $=$ 10dB.

Incidentally, in the case of feature-wise matching, the maximum achievable F-score when SNR $=$ 10dB and $\rho = 0.3$ is



(a) ROC vs SNR for MVSC dataset

(b) ROC vs puncturing fraction ρ MVSC dataset

(c) ROC vs SNR SR dataset

(d) ROC vs puncturing fraction ρ SR dataset

FIGURE 16. ROC curves obtained for feature-wise matching descriptors from both the SR and the MVSC datasets.

0.644 for the triple ($d = 0.4360$, TPR $= 0.84$, FPR $= 0.497$). As the number of bits lost increases, the maximum achievable F-score decreases until reaching the value of 0.355 when $\rho = 0.5$ which is a very low value as an accuracy metric. This leads us to deduce that, for feature-wise matching, the maximum bit-rate reduction we can obtain when SNR $=$ 10dB in such a way that we guarantee an F-measure accuracy no lower than 0.644 is 30%.

H. VALIDATING THE PROPOSED SYSTEM

To validate the proposed multi-view distributed feature decoding system, we analyzed its performance on the widely-known database MVSC [51] which is completely independent of the chosen application. The MVSC data set is a collection of high number of images containing patches sampled from 3D reconstructions of three well-known world sites: the Statue of Liberty (New York), Notre Dame (Paris) and Half Dome (Yosemite). In this paper we used only images from the Statue of Liberty database which is a set of 1758 1024 × 1024 bitmap images. Each image is composed of 256 patches (16 × 16 array of image patches). Each patch is sampled as 64 × 64 gray-scale. Along with the images, associated metadata provide information about the location, scale and orientation of each keypoint representing a patch, as well as a ground truth for matching patches. For each patch, one SURF descriptor is extracted and transmitted to the sink node. At the receiver side, we used the descriptors of the corresponding matched patches as SI for the SW decoder.

Figure 15 reports the FER performance obtained for different values of SNR and different puncturing patterns. In the comparison with Figure 8(a), note the quasi-identical behavior of the FER under ideal CNM for the MVSC and the SR datasets. Figure 16 further illustrates the ROC curves obtained for evaluating the feature-wise matching of descriptors from the MVSC dataset in the comparison to those of the SR dataset. As a matter of fact, recall that the descriptors from the MVSC database are feature-wise matched, and therefore this is the only possible means of fair comparison. As can be easily seen from Figure 16, the matching accuracy results for both databases have similar behavior over the entire SNR and puncturing fraction ρ ranges.

V. CONCLUSION

This paper presented a multi-view vehicle tracking system at roundabouts suitable for VSNs. To reduce the bit-rate and the energy consumption at the camera sensor nodes, the ATC paradigm was adopted where cameras detect moving vehicles and extract local features while the rest of the tracking process is performed at the sink node. A multi-view distributed feature coding architecture based on DSC techniques was applied to convey essential information to the sink. The extracted features from all cameras are encoded separately and jointly decoded at the receiver side. By exploiting a robust object-wise matching procedure, the proposed system is proved to achieve significant bit-rate reduction up to 30% for high SNRs $\geqslant$ 10dB while maintaining a good multi-view matching accuracy close to the performance where no quantization and coding is performed (F-score $= 0.955$). The study also showed that, for applications based on feature-wise matching we can still achieve similar bit-rate savings at the price of a deteriorated matching accuracy ($\leqslant$0.644). As a proof-of-concept, we tested the proposed system on a small network composed of two cameras, while extensions to denser networks with higher number of cameras are left to future work.

REFERENCES

[1] H. Dinh and H. Tang, "Development of a tracking-based system for automated traffic data collection for roundabouts," *J. Mod. Transp.*, vol. 25, no. 1, pp. 12–23, Mar. 2017.

[2] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2681–2698, Oct. 2016.

[3] L. A. Rodegerdts, *Roundabouts: An Informational Guide*, vol. 672. Washington, DC, USA: Transp. Res. Board, 2010.

[4] B. Tavli, K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools Appl.*, vol. 60, no. 3, pp. 689–726, Oct. 2012.

[5] F. G. H. Yap and H.-H. Yen, "A survey on sensor coverage and visual data capturing/processing/transmission in wireless visual sensor networks," *Sensors*, vol. 14, no. 2, pp. 3506–3527, Feb. 2014.

[6] M. Abdollahzadeh, H. A. Ghazijahani, and H. Seyedarabi, "Quality aware HEVC video transmission over wireless visual sensor networks," in *Proc. 24th Iranian Conf. Electr. Eng. (ICEE)*, May 2016, pp. 787–792.

[7] S. Colonnese, F. Cuomo, and T. Melodia, "An empirical model of multiview video coding efficiency for wireless multimedia sensor networks," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1800–1814, Dec. 2013.

[8] N. Cen, Z. Guan, and T. Melodia, "Interview motion compensated joint decoding for compressively sampled multiview video streams," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1117–1126, Jun. 2017.

[9] A. Redondi, L. Baroffio, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze vs. Analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2013, pp. 278–282.

[10] L. Baroffio, A. Canclini, M. C. A. Redondi, M. Tagliasacchi, G. Dan, E. Eriksson, V. Fodor, J. Ascenso, and P. Monteiro, "Enabling visual analysis in wireless sensor networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3408–3410.

[11] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Coding visual features extracted from video sequences," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2262–2276, May 2013.

[12] D. Van Opdenbosch and E. Steinbach, "Collaborative visual SLAM using compressed feature exchange," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 57–64, Jan. 2019.

[13] S. Milani and G. Calvagno, "Distributed video coding based on lossy syndromes generated in hybrid pixel/transform domain," *Signal Process., Image Commun.*, vol. 28, no. 6, pp. 553–568, Jul. 2013.

[14] J. Hou, L.-P. Chau, M. Zhang, N. Magnenat-Thalmann, and Y. He, "A highly efficient compression framework for time-varying 3-D facial expressions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1541–1553, Sep. 2014.

[15] S. Milani, "A distributed source autoencoder of local visual descriptors for 3D reconstruction," *Pattern Recognit. Lett.*, vol. 146, pp. 193–199, Jun. 2021.

[16] A. M. Nambiar, M. Tagliasacchi, and E. Magli, "Secure image databases through distributed source coding of SIFT descriptors," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2012, pp. 130–135.

[17] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[18] N. Monteiro, C. Brites, F. Pereira, and J. Ascenso, "Multi-view distributed source coding of binary features for visual sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2807–2811.

[19] S. Eleuch, N. Khouja, T. Erseghe, and F. Tlili, "Feature-based vehicle tracking at roundabouts in visual sensor networks," in *Proc. 17th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Jul. 2020, pp. 167–172.

[20] T. Furuya and C. J. Taylor, "Road intersection monitoring from video with large perspective deformation," Ph.D. dissertation, Dept. Comput. Inf. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, 2014.

[21] W. Wang, T. Gee, J. Price, and H. Qi, "Real time multi-vehicle tracking and counting at intersections from a fisheye camera," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 17–24.

[22] J. Barthélemy, N. Verstaevel, H. Forehead, and P. Perez, "Edge-computing video analytics for real-time traffic monitoring in a smart city," *Sensors*, vol. 19, no. 9, p. 2048, May 2019.

[23] M. Nikodem, M. Słabicki, T. Surmacz, P. Mrówka, and C. Dołęga, "Multi-camera vehicle tracking using edge computing and low-power communication," *Sensors*, vol. 20, no. 11, p. 3334, Jun. 2020.

[24] H.-M. Hsu, T.-W. Huang, G. Wang, J. Cai, Z. Lei, and J.-N. Hwang, "Multi-camera tracking of vehicles based on deep features Re-ID and trajectory-based camera link models," in *Proc. CVPR Workshops*, 2019, pp. 416–424.

[25] C. Tomasi and T. Kanade, "Detection and tracking of feature points," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., CMU-CS-91-132, 1991.

[26] A. Romanoni, L. Mussone, D. Rizzi, and M. Matteucci, "A comparison of two Monte Carlo algorithms for 3D vehicle trajectory reconstruction in roundabouts," *Pattern Recognit. Lett.*, vol. 51, pp. 79–85, Jan. 2015.

[27] M. Muffert, D. Pfeiffer, and U. Franke, "A stereo-vision based object tracking approach at roundabouts," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 22–32, Summer 2013.

[28] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance System*. Boston, MA, USA: Springer, 2002, pp. 135–144.

[29] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, May 2006.

[30] A. B. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 4305–4312.

[31] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[32] N. Saunier and T. Sayed, "A feature-based tracking algorithm for vehicles in intersections," in *Proc. 3rd Can. Conf. Comput. Robot Vis. (CRV)*, 2006, p. 59.

[33] T. Gao, Z.-G. Liu, W.-C. Gao, and J. Zhang, "Moving vehicle tracking based on SIFT active particle choosing," in *Proc. 3rd Can. Conf. Comput. Robot Vis. (CRV)*. QC, Canada: Springer, Jun. 2006, pp. 695–702.

[34] L. Baroffio, J. Ascenso, M. Cesana, A. Redondi, and M. Tagliasacchi, "Coding binary local features extracted from video sequences," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2794–2798.

[35] D. Van Opdenbosch, M. Oelsch, A. Garcea, T. Aykut, and E. Steinbach, "Selection and compression of local binary features for remote visual SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7270–7277.

[36] L. Bondi, L. Baroffio, M. Cesana, A. Redondi, and M. Tagliasacchi, "Multi-view coding of local features in visual sensor networks," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6.

[37] C. M. Christoudias, R. Urtasun, and T. Darrell, "Unsupervised feature selection via distributed coding for multi-view object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[38] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[39] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[40] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[41] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, Philadelphia, PA, USA, 2007, pp. 1027–1035.

[42] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[43] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Proc. NIPS*, vol. 13, 2000, pp. 689–695.

[44] T. Erseghe, *Channel Coding*. Padova, Italy: Padova Univ., 2016.

[45] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O'Reilly Media, 2008.

[46] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, "Learning epipolar geometry from image sequences," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, pp. 1–8.

[47] W. Chojnacki and M. J. Brooks, "Revisiting Hartley's normalized eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1172–1177, Sep. 2003.

[48] *Pi NoIR Camera V2*. Accessed: Apr. 10, 2022. [Online]. Available: https://www.raspberrypi.org/products/pi-noir-camera-v2/

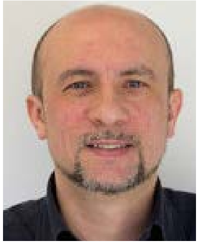
[49] *Raspberry Pi 3 Model B Board*. Accessed: Apr. 10, 2022. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[50] S. M. Bileschi, "StreetScenes: Towards scene understanding in still images," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 132692645, 2006.

[51] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

SALMA ELEUCH received the Telecommunication Engineering degree from the Higher School of Communication of Tunis (SUP'COM), Tunisia, in 2017. She is currently pursuing the joint Ph.D. degree with SUP'COM and the Università degli studi di Padova, Italy. In 2016, she participated as an Erasmus Mobility Program with the Università degli studi di Padova, where she carried out her master's thesis in the field of wireless sensor networks. She is a member of the GRESCOM Laboratory, SUP'COM, University of 7th November at Carthage, Tunisia. Her current research interests include visual sensor networks, smart traffic monitoring, distributed coding, and distributed optimization.

NADIA KHOUJA received the Engineering Diploma and the master's degrees in communications and the Ph.D. degree in communication sciences and information technologies from the High Communications School of Tunis, Tunisia, in 2002 and 2006. She worked as a Research and Development Engineer at STMicroelectrics, from 2002 to 2007. She worked on fields related to SoC and NoC modelisation and validation. She is currently an Associate Professor in telecommunications at the Institut Superieur des Etudes technologiques en communications de Tunis (ISETCOM) and a member of the GRESCOM Laboratory, SUP'COM, University of 7th November at Carthage, Tunisia. Her research interests include digital filtering, power consumption analysis, VLSI, embedded processors circuits, and FEC decoding algorithms and architectures.



SIMONE MILANI (Member, IEEE) received the Laurea degree in telecommunication engineering and the Ph.D. degree in electronics and telecommunication engineering from the Università degli studi di Padova, Padova, Italy, in 2002 and 2007, respectively. He was a Visiting Ph.D. Student with the University of California at Berkeley, Berkeley, CA, USA, in 2006. He was also a Consultant at STMicroelectronics, Agrate, Italy. He worked as a Postdoctoral Researcher with the University of Udine, Udine, Italy, the Università degli studi di Padova, and the Politecnico di Milano, Milan, Italy, from 2007 to 2013. From 2013 to 2020, he was an Assistant Professor with the Dipartimento di Ingeneria dell'Informazione, Università degli studi di Padova, where he is currently an Associate Professor. His research interests include digital signal processing, image and video coding, 3D video processing and compression, joint source-channel coding, robust video transmission, distributed source coding, multiple description coding, and multimedia forensics.

TOMASO ERSEGHE received the Laurea (M.Sc.) and Ph.D. degrees in telecommunication engineering from the Università degli studi di Padova, Italy, in 1996 and 2002, respectively. From 1997 to 1999, he was with Snell and Wilcox, an English broadcast manufacturer. From 2003 to 2017, he was an Assistant Professor (Ricercatore) at the Dipartimento di Ingeneria dell'Informazione, Università degli studi di Padova, where he is currently an Associate Professor. His research interests include coding in the finite block-length regime, social network analysis and network science, distributed algorithms, smart grid optimization, ultra-wideband transmission systems design, spectral analysis of complex modulation formats, fractional Fourier transforms and their applications, image processing, and compression.

FETHI TLILI received the Electrical Engineering degree, in 1990, the Ph.D. degree in electrical engineering, in 2000, and the Habilitation degree in telecommunications, in 2011. He was a senior consultant and an expert at CPL modems design, video technology, and radar processing for several international companies. He has been a Professor at the Higher School of Communications of Tunis (SUP'COM), Tunisia, since 1991, and a Researcher at the GRES'COM Laboratory, since 2012. His research interest include embedded systems, digital communications systems, HW architectures for signal processing, video technology, and radar processing for automotive systems.