2014 IEEE International Conference on Distributed Computing in Sensor Systems

Real-time Distributed Visual Feature Extraction from Video in Sensor Networks

Emil Eriksson, György Dán, Viktoria Fodor
School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
{emieri,gyuri,vfodor}@kth.se

***Abstract*—Enabling visual sensor networks to perform visual analysis tasks in real-time is challenging due to the computational complexity of detecting and extracting visual features. A promising approach to address this challenge is to distribute the detection and the extraction of local features among the sensor nodes, in which case the time to complete the visual analysis of an image is a function of the number of features found and of the distribution of the features in the image. In this paper we formulate the minimization of the time needed to complete the distributed visual analysis for a video sequence subject to a mean average precision requirement as a stochastic optimization problem. We propose a solution based on two composite predictors that reconstruct randomly missing data, and use a quantile-based linear approximation of the feature distribution and time series analysis methods. The composite predictors allow us to compute an approximate optimal solution through linear programming. We use two surveillance videos to evaluate the proposed algorithms, and show that prediction is essential for controlling the completion time. The results show that the last value predictor together with regular quantile-based distribution approximation provide a low complexity solution with very good performance.**

***Index Terms*—Image analysis; wireless sensor networks**

I. Introduction

Low cost cameras and networking hardware make a new class of sensor networks viable, namely, visual sensor networks (VSNs), where visual information is captured at one or several cameras and processed and transmitted through several network nodes, until the useful information reaches a central unit. Such systems can have both industrial and consumer applications, including supervision and surveillance systems, and remote monitoring, or as components for autonomous systems, like automotive navigation [1], [2]. VSNs, however, differ from more traditional sensor networks, where the transmission of sensed information requires little bandwidth and the complexity of the information processing is rather low. VSNs may instead capture high bitrate video sequences, and information processing is performed locally in the network in order to deliver only useful information to the sink node. The information processing needed for visual analysis, such as for tracking and for object recognition, is however computationally intensive even using state-of-the-art algorithms like FAST and BRISK [3], [4].

A promising solution to allow real-time processing of the visual information is to distribute the processing tasks among several sensor nodes in the network, as it allows the use of the processing capacity of several nodes. Nevertheless, the time it takes for a particular node to perform the processing depends on the available communication and computational resources of the node and, importantly, on the image content, which is not known prior to performing the processing. Therefore, optimizing the distribution of the processing tasks among the network nodes is a challenging task.

In this paper we consider visual analysis of video sequences based on local feature descriptors [3], [4], which are widely used for object recognition and tracking. The precision of the visual analysis task depends on the number of descriptors used, with a known target value. The camera node distributes the workload of interest point detection and descriptor extraction by assigning image sub-areas to the processing nodes. The processing workload of a node depends both on the size of the sub-area and on the number of detected and extracted descriptors. Our goal is to maintain good visual analysis performance, which requires that the total number of interest points detected be close to the target value, and to allow real-time video analysis, which requires that the network nodes should receive the pixel information of the assigned sub-area, and should complete the extraction of the descriptors as fast as possible.

We formulate this problem as a multi-objective stochastic optimization problem. To solve the optimization problem we leverage the temporal correlation among the consecutive images in the video sequence. The temporal correlation allows us to develop a predictor of the detection threshold, such that the number of descriptors is close to the required number. To minimize the completion time, we find the optimal schedule of the transmissions to the processing nodes, and we predict the optimal division of the image into sub-areas using a percentile-based approximation, such that the time of completing the feature extraction is minimized. Numerical results show that prediction is essential to achieve our objectives, and that the proposed prediction algorithms combine low computational complexity with good prediction performance.

The rest of the paper is organized as follows. In Section II we review related work. In Section III we describe the considered system and in Section IV we provide the problem formulation. In Section V we develop the proposed predictors and in Section VI we identify the optimal scheduling order.

The project GreenEyes acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 296676.

978-1-4799-4618-1/14 $31.00 © 2014 IEEE
DOI 10.1109/DCOSS.2014.30

In Section VII we present smiulation results and we conclude the paper in Section VIII.

II. Related work

The challenge of networked visual analysis is addressed in [3], [4], defining feature extraction schemes with low computational complexity. To decrease the transmission bandwidth requirements, [5], [6] propose lossy image coding schemes optimized for descriptor extraction, while [7], [8], [9] give solutions to decrease the number and the size of the descriptors to be transmitted. In [10] the number and the quantization level of the considered descriptors are jointly optimized to maximize the accuracy of the recognition, subject to energy and bandwidth constraints.

To decrease the transmission requirements of feature extraction in the case of video sequences [11] selects candidate descriptor locations based on motion prediction, and transmits and processes these areas only. In [12], [13] intra- and inter-frame coding of descriptors is proposed to decrease the transmission requirements.

Our work is motivated by recent results on the expected transmission and processing load of visual analysis in sensor networks [10], [14], [15]. Measurements in [14] demonstrate that processing at the camera or at the sink node of the VSN leads to significant delays, and thus distributed processing is necessary for real-time applications. The requirement of prediction based system optimization is motivated by the statistical analysis of a large public image database in [15], showing that the number and the spatial distribution of the descriptors have high variability and depend significantly on the image content. Thus, the temporal correlation in the video sequence needs to be utilized to achieve the efficient control of the visual analysis parameters. Finally, experiments in [10] show that the processing delay and the energy consumption increase linearly with the image size and with the number of detected descriptors. Consequently, to limit the time needed for descriptor extraction, the number of descriptors need to be controlled, and the workload allocation has to consider both the size of the sub-areas and the distribution of the descriptors.

Optimal load scheduling for distributed systems is addressed in [16], in the framework of Divisible Load Theory, with the general result that minimum completion time is achieved, if all processors finish the processing at the same time. Usually three decisions need to be made: the subset of the processors used, the order they receive their share of workload, and the division of the workload. Unfortunately, the results are specific to a given system setup. Works closest to ours address tree networks with heterogeneous link capacities and processor speeds [17], concluding that scheduling should be in decreasing order of the transmission capacities, while the processing speed does not affect the scheduling decision. However, [18] shows that the optimal scheduling order may be different if the processing has constant overhead, and under equal link capacities the scheduling should happen in decreasing order of the processing speeds. As we show in the paper, this result can not be used in general either, for example, in our scenario where unicast and multicast transmissions are combined, and the link transmission capacities differ.

III. Background and System model

We consider a VSN consisting of a camera node $\mathcal{C}$, a set of processing nodes $\mathcal{N}$, $|\mathcal{N}| = N$, and a sink node $\mathcal{S}$. The camera node captures a sequence of images. Each image is transmitted to and processed at nodes in $\mathcal{N}$, and finally the results are transmitted to $\mathcal{S}$ where the visual analysis task is completed.

A. Communication model

The nodes communicate using a multicast/broadcast capable wireless communication protocol, such as IEEE 802.15.4 or IEEE 802.11. Transmissions suffer from packet losses due to wireless channel impairments. As measurement studies show [19], [20], losses at the receivers can be modeled as independent and the loss burst lengths have low mean and variance in the order of a couple of frames [21], [22]. Therefore, a widely used model of the loss process is a low-order Markov-chain, with fast decaying correlation and short mixing time. In the system we consider, the amount of data to be transmitted to the processing nodes is relatively large, and therefore it is reasonable to model the average transmission time from $\mathcal{C}$ to a node $n \in \mathcal{N}$ as a linear function of the amount of transmitted data. The average per pixel transmission time, including potential retransmissions, is referred to as the transmission time coefficient and denoted by C_n. As the throughput is close to stationary over short timescales, C_n can be estimated [23]. When using multicast or broadcast transmission, the throughput is determined by the receiver with lowest achievable throughput.

B. Feature detection and extraction

A sequence $\{Z_i\}$, $i = 1, \ldots, I$, of images is captured at $\mathcal{C}$. Each image has a height of h and a width of w pixels. For each image, $\mathcal{C}$ sends the image data to the processing nodes, which perform interest point detection and feature descriptor extraction.

Interest point detection is performed by applying a blob detector or an edge detector at every pixel of the image area [24], [25], [4]. The detector computes a response score for each pixel based on a square area centered around the pixel, with side length $2ow$ pixels, where o depends on the applied detector. A pixel is identified as an interest point if the response score exceeds the detection threshold $\vartheta \in \mathbf{\Theta} \subseteq \mathbb{R}^+$. The number of interest points detected in an image depends on the image and on the detection threshold ϑ, we thus describe the number of interest points detected in image i is by an integer valued, left continuous, non-negative, decreasing step function $f_i(\vartheta)$ of the detection threshold ϑ. f_i is not known before processing image i; we model it as a random function chosen from the family of integer valued, left continuous, non-negative, decreasing step functions. The inverse function f_i^{-1} : $\mathbb{N} \to \mathbf{\Theta}$ can be defined as $f_i^{-1}(m) = \max\{\vartheta | f_i(\vartheta) = m\}$.

The maximum exists because f_i is a left continuous, decreasing step function. We denote the sequence of thresholds used in the images by $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_I)$.

In order to distribute the workload among the processing nodes in $\mathcal{N}$, the camera node divides each image i into at most N sub-areas. Sub-area $Z_{i,n}$ is then assigned to processing node n. For simplicity, we consider that the sub-areas are slices of the image formed along the horizontal axis. This scheme was referred to as area-split in [15]. We specify the sub-areas by the horizontal coordinates of the vertical lines separating them, normalized by the image width w, which we refer to as the cut-point location vector $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,N})$, $x_{i,1} < \ldots < x_{i,N} = 1$. For notational convenience, we define $x_{i,0} = 0$, the left edge of image i, and $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_I)$, the sequence of cut-point vectors used for the trace. Since interest point detection at a pixel requires a square area around the pixel to be available, all points within ow pixels of the horizontal coordinate $x_{i,n}$ need to be transmitted to both node n and $n+1$. We call o the overlap, and we express its value normalized by the image width w (hence the multiplication above). We consider that $\frac{1}{N} >> o$, which holds if the image size is reasonably large.

The number of interest points detected in sub-area $Z_{i,n}$ depends on the image, the detection threshold ϑ_i and on the cut-point location vector $\boldsymbol{x}_i$. We thus describe the number of interest points detected in sub-area $Z_{i,n}$ by the function $f_{i,n}(\vartheta_i, \boldsymbol{x}_i)$, and we define the vector function $\boldsymbol{f}_i(\vartheta_i, \boldsymbol{x}_i) = (f_{i,1}(\vartheta_i, \boldsymbol{x}_i), \ldots, f_{i,N}(\vartheta_i, \boldsymbol{x}_i))$. The function $\boldsymbol{f}_i(\vartheta_i, \boldsymbol{x}_i)$ can be modeled as a random function from the family of integer valued vector functions with $\sum_{n=1}^{N} f_{i,n}(\vartheta, \boldsymbol{x}_i) = f_i(\vartheta)$. We consider that the time it takes to detect the interest points is a linear function of the size of the sub-area (not including the overlap) with rate $P_{d,px,n}$, and of the number of interest points detected with rate $P_{d,ip,n}$.

As the next step, a feature descriptor is extracted for each interest point. The time it takes to extract the descriptors is a linear function of the number of interest points detected with rate $P_{e,ip,n}$.

To validate this model, we performed interest point detection and feature descriptor extraction on a BeagleBone Black single board computer for 3 different image sizes using OpenCV [26]. The results shown in Figure 1 confirm that the computation time can be well approximated by a linear function. Similar results were reported on an Intel Imote2 platform in [14].

When node n completes the extraction of descriptors within area $Z_{i,n}$, it transmits them to $\mathcal{S}$, where various computer vision tasks can be performed. In order for $\mathcal{S}$ to be able to perform its computer vision tasks, it requires M^* interest points to be detected in each image. To optimize the distributed processing, ϑ and $\boldsymbol{x}$ should be selected based on information available at $\mathcal{S}$. Since for each already transmitted image i the sink has access to the parameters $(\vartheta_i, \boldsymbol{x}_i)$, as well as all the interest point descriptors, it knows the location and score of each detected interest point. It can therefore calculate $f_i(\vartheta)$ for any $\vartheta \geq \vartheta_i$, i.e., the total workload the system would have had with detection threshold ϑ. Nevertheless, if $f_i(\vartheta_i) < M^*$ then $f_i^{-1}(M^*)$ cannot be computed by $\mathcal{S}$. Similarly, $\mathcal{S}$ can compute $\boldsymbol{f}_i(\vartheta, \boldsymbol{x}_i)$ for any $\vartheta \geq \vartheta_i$ and any cut-point location vector $\boldsymbol{x}$. We use Υ_i to denote the information available to $\mathcal{S}$ about image i, and Υ_{i-} to denote the information available about all images previous to image i.

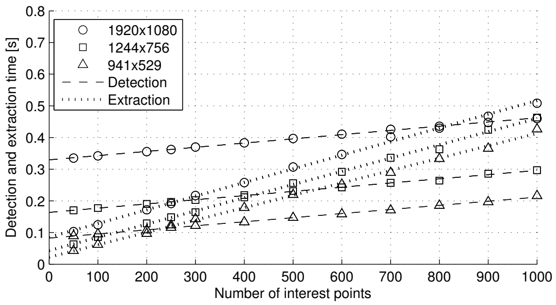

Figure 1: Average time for performing detection and extraction as a function of the number of interest points found for three image sizes. The linear regression shows a good fit.

IV. Problem Formulation

Based on the model of the wireless links and of the detection and extraction of features, we first express the transmission and processing times of the N processing nodes as a function of the threshold ϑ_i and the cut-point location vector $\boldsymbol{x}_i$. We then define the performance metrics and formulate our objective.

A. Transmission and Processing Time

Assume the nodes are numbered by the order in which they receive their data from the camera node $\mathcal{C}$, and let us consider a node n. Initially, node n is idle while all preceding nodes receive their data. It then starts receiving data once $\mathcal{C}$ starts to transmit the overlap shared between nodes n and $n-1$, followed by the data destined to node n only, and finally the overlap shared between nodes n and $n+1$. Once node n has received the data, feature detection and descriptor extraction are performed on the sub-area $Z_{i,n}$. Finally the descriptors are transmitted to $\mathcal{S}$. However, as the size of a descriptor is small compared to the image data in the case of modern binary descriptors like BRISK [4], we do not consider the time it takes to transmit them to $\mathcal{S}$. Figure 2 illustrates the phases for $N = 3$.

In the following we provide matrix expressions for the transmission, processing and finally for the task completion time for the case when $2o < x_{i,n-1} - x_{i,n}$, i.e., an overlap spans only two nodes; similar expressions can be obtained for $2o \geq x_{i,n-1} - x_{i,n}$. Let G_j be an $N \times 1$ column vector, and let D_j and E_j be $N \times N$ matrices. Also, let us use the notation $C_n^M \triangleq \max(C_n, C_{n+1})$ as a shorthand for the effective transmission time coefficient for multicast transmission to nodes n and $n+1$.

The average time node n spends idling before it receives the first bit can be expressed in matrix notation as

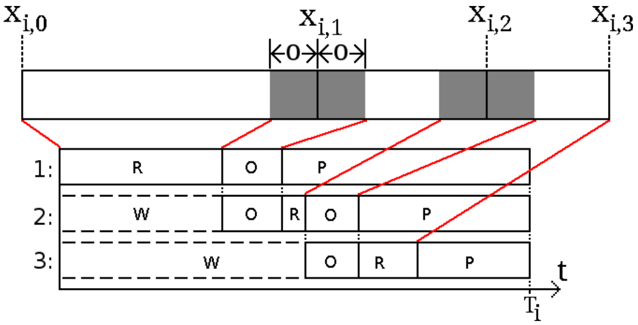

Figure 2: Example with $N = 3$ processing nodes. The image is cut along the horizontal axis, the subareas and the overlaps are transmitted to and processed by the processing nodes. A node is either (w)aiting to receive data, is (r)eceiving individual data, is receiving (o)verlapping data, or is (p)rocessing data.

$T_{idle,i} = D_1\boldsymbol{x}_i + G_1$, where

$$d_{1,m,n} = \begin{cases} hwC_n, & m = n+1 \\ hwC_n - hwC_{n+1}, & m > n+1 \\ 0, & otherwise, \end{cases} \text{ and}$$

$$g_{1,m} = \begin{cases} 0, & m = 1 \\ -hwoC_1 + \sum_{j=2}^{m-1}\left(2hwoC_{j-1}^M - 2hwoC_j\right), & m > 1. \end{cases}$$

Node n receives the overlaps with its neighbours in multicast transmission. As nodes 1 and N are assigned the edge pieces of the image, they will only receive a single overlap. The time to receive the overlap is

$T_{overlap,i} = G_2$, where

$$g_{2,n} = \begin{cases} 2hwoC_1^M, & n = 1 \\ 2hwoC_{n-1}^M + 2hwoC_n^M, & 1 < n < N \\ 2hwoC_{N-1}^M, & n = N. \end{cases}$$

The average time it takes node n to receive the non-overlapping data depends on the size of the sub-area $Z_{i,n}$

$T_{transmit,i} = D_3\boldsymbol{x}_i + G_3$, where

$$d_{3,m,n} = \begin{cases} hwC_n, & m = n \\ -hwC_{n+1}, & m = n+1 \\ 0, & otherwise \end{cases} \text{ and}$$

$$g_{3,n} = \begin{cases} -hwoC_n, & n \in 1, N \\ -2hwoC_n, & otherwise. \end{cases}$$

The time it takes to perform interest point detection is a function of the size of sub-area $Z_{i,n}$ and of the number of detected interest points, and can be expressed as

$T_{detect,i} = D_4\boldsymbol{x}_i + E_4\boldsymbol{f}_i\left(\vartheta_i, \boldsymbol{x}_i\right)$, where

$$d_{4,m,n} = \begin{cases} \frac{hw}{P_{d,px,n}}, & m = n \\ -\frac{hw}{P_{d,px,n+1}}, & m = n+1 \\ 0, & otherwise \end{cases} \text{ and}$$

$$e_{4,m,n} = \begin{cases} \frac{1}{P_{d,ip,n}}, & m = n \\ 0, & otherwise. \end{cases}$$

Finally, the time needed for descriptor extraction is a function of the number of detected interest points

$T_{extract,i} = E_5\boldsymbol{f}_i\left(\vartheta_i, \boldsymbol{x}_i\right)$, where

$$e_{5,m,n} = \begin{cases} \frac{1}{P_{e,n}}, & m = n \\ 0, & otherwise. \end{cases}$$

Let us define $D \triangleq D_1 + D_3 + D_4$, $E \triangleq E_4 + E_5$, and $G \triangleq G_1 + G_2 + G_3$. Using this notation we can express the expected completion time of each node n for image i, $T_i\left(\vartheta_i, \boldsymbol{x}_i\right) = \left(T_{i,1}\left(\vartheta_i, \boldsymbol{x}_i\right), \ldots, T_{i,N}\left(\vartheta_i, \boldsymbol{x}_i\right)\right)$ as

$$T_i\left(\vartheta_i, \boldsymbol{x}_i\right) = D\boldsymbol{x}_i + E\boldsymbol{f}_i\left(\vartheta_i, \boldsymbol{x}_i\right) + G, \tag{1}$$

which is a non-linear vector function of ϑ_i and $\boldsymbol{x}_i$.

B. Performance optimization

We are interested in two key aspects of the VSN's performance. First, we want to ensure that the VSN can perform the visual analysis task at the required level of mean average precision. The mean average precision can be controlled by the number of detected interest points. We therefore define our first performance metric to be the squared error in detected interest points in image i compared to the target value M^* required by the computer vision task

$$e_i^D\left(\vartheta_i\right) = \left(f_i\left(\vartheta_i\right) - M^*\right)^2, \tag{2}$$

and we define the corresponding mean square error as $e^D\left(\boldsymbol{\vartheta}\right) = \frac{1}{I}\sum_{i=1}^{I} e_i^D\left(\vartheta_i\right)$. We define the optimal detection threshold for image i as $\vartheta_i^* = \min(\theta_i^*)$, where $\theta_i^* = \{\vartheta | e_i^D\left(\vartheta\right) = 0\}$.

Second, we are interested in minimizing the time it takes to complete the detection and the extraction of all descriptors. We therefore define our second performance metric based on the VSN's completion time, which we define as the largest completion time among all processing nodes. We define the squared completion time error of the VSN for image i as the squared difference compared to the smallest possible VSN completion time

$$e_i^C\left(\vartheta_i, \boldsymbol{x}_i\right) = \left(\max_n\left(T_i\left(\vartheta_i, \boldsymbol{x}_i\right)\right) - \max_n\left(T_i\left(\vartheta_i^*, \boldsymbol{x}_i^*\right)\right)\right)^2, \tag{3}$$

and the mean squared completion time error as $e^C\left(\boldsymbol{\vartheta}, \boldsymbol{x}\right) = \frac{1}{I}\sum_{i=1}^{I} e_i^C\left(\vartheta_i, \boldsymbol{x}_i\right)$, where the optimal cut-point location vector $\boldsymbol{x}_i^* \in \arg\min_{\boldsymbol{x}_i}\max_n\left(T_i\left(\vartheta_i^*, \boldsymbol{x}_i\right)\right)$.

Observe that both (2) and (3) depend on the functions f_i and $\boldsymbol{f}_i$, which are not known prior to processing image i. By modeling f_i and $\boldsymbol{f}_i$ as random functions, we can formulate our problem as a stochastic multi-objective optimization problem

$$\text{lexmin}(\mathbb{E}[e^D\left(\boldsymbol{\vartheta}\right)], \mathbb{E}[e^C\left(\boldsymbol{\vartheta}, \boldsymbol{x}\right)]) \tag{4}$$

s.t.

$$\boldsymbol{\vartheta} \in \Theta^I, \boldsymbol{x} \in \mathcal{X}^I, \tag{5}$$

where $lexmin$ stands for lexicographical minimization, and we are looking for an expected value efficient solution [27]. Since the choice of ϑ_i and $\boldsymbol{x}_i$ for image i does not influence the error at images $j > i$, this problem is equivalent to solving

$$\text{lexmin}(\mathbb{E}[e_i^D\left(\vartheta_i\right)], \mathbb{E}[e_i^C\left(\vartheta_i, \boldsymbol{x}_i\right)]) \tag{6}$$

s.t.

$$\vartheta_i \in \Theta, \boldsymbol{x}_i \in \mathcal{X}, \tag{7}$$

for every image i based on the information Υ_{i-}. We therefore search for the solution in the form of a predictor $\tau^*\left(\Upsilon\right)$ that minimizes the expected square error

$$\tau^* \in \arg\min_{\tau} \mathbb{E}[e_i^D\left(\tau\left(\Upsilon_{i-}\right)\right)], \tag{8}$$

and a predictor $\gamma^*(\Upsilon)$ that minimizes the expected squared completion time error

$$\gamma^* \in \arg\min_{\gamma} \mathbb{E}[e_i^C(\tau^*(\Upsilon_{i-}), \gamma(\Upsilon_{i-}))]. \tag{9}$$

In what follows we develop and analyze predictors with low complexity and little overhead suitable for sensor networks.

V. Predictive Completion Time Minimization

Solving the prediction problems (8) and (9) with conventional methods is not straightforward for two reasons. First, since $f_i(\vartheta)$ and $f_{i,n}(\vartheta, \boldsymbol{x})$, and thus (2) and (3) are step functions in ϑ and $\boldsymbol{x}$, the sets of minimizers $\theta_i^* = \{\vartheta | e_i^D(\vartheta) = 0\}$ and $\Xi_i^* = \{\boldsymbol{x} | e_i^C(\vartheta^*, \boldsymbol{x}) = 0,\ \vartheta^* \in \theta_i^*\}$ may not be singletons. Second, if $f_i(\hat{\vartheta}_i) < M^*$ then θ_i^* is unknown and can not be used for prediction. Third, a predictor for solving (9) should predict the distribution of interest point locations.

A. Controlling the Workload

We first address the problem of estimating θ_i^* when $f_i(\hat{\vartheta}_i) < M^*$. Let us consider an image i for which the predicted detection threshold $\hat{\vartheta}_i$ results in $f_i(\hat{\vartheta}_i) < M^*$. We want to estimate a $\hat{\vartheta}_i^* \in \theta_i^*$ that can be used to predict $\hat{\vartheta}_{i+1} \in \theta_{i+1}^*$. The approach we describe in the following uses preceding images for which $f_j(\hat{\vartheta}_j) \geq M^*$ for estimating the slope of the function f_i^{-1} around M^* [28]. Let $\mathcal{I}_{i-}$ be the set of indices of the images before image i for which the estimated detection threshold $\hat{\vartheta}_j$ resulted in at least M^* interest points, i.e., $f_j(\hat{\vartheta}_j) \geq M^*\ \forall j \in \mathcal{I}_{i-}$.

We can use these images to obtain the backward estimate of the slope of the function f_i^{-1} at M^* by using the linear regression in the backward direction proposed in [28]. In the backward direction (i.e., less than M^* interest points) we can compute the regression for arbitrary difference $d < M^*$ based on the available data Υ_{i-}. For a particular difference $d = M^* - f_j(\vartheta)$ after simplification we obtain

$$\beta_{i-}^b(d) = \frac{1}{|\mathcal{I}_{i-}|} \sum_{j \in \mathcal{I}_{i-}} \frac{f_j^{-1}(M^*) - f_j^{-1}(M^* - d)}{d}, \tag{10}$$

which is the average backward difference quotient of f^{-1} at M^* over the images in $\mathcal{I}_{i-}$. Using the backward regression coefficient we obtain the estimated threshold

$$\hat{\vartheta}_i^{b*} = \hat{\vartheta}_i - (f_i(\hat{\vartheta}_i) - M^*)\beta_{i-}^b, \tag{11}$$

which is the minimum variance unbiased estimator of ϑ_i^* [28]. Given $\hat{\vartheta}_i^{b*}$, we can use a time series model to predict $\hat{\vartheta}_{i+1}$.

B. Distribution-based Cut-point Location Vector Selection

We address the minimization of the completion time in two steps. First, we consider a given ordering of the processing nodes and provide an algorithm to approximate the cut-point location vector $\boldsymbol{x}_i$ that minimizes the completion time for the ordering. Second, in Section VI we show how to find the ordering that allows the smallest completion time.

Without loss of generality we consider that sub-area $Z_{i,n}$ has to be processed by node n, and for image i we need to find the cut-point vector $\boldsymbol{x}_i$ that minimizes $e_i^C(\vartheta_i, \boldsymbol{x}_i)$.

Let us assume that the distribution of the interest points' horizontal coordinates $F_i(\vartheta_i, x)$ is known, thus $\boldsymbol{f}_i(\vartheta_i, \boldsymbol{x})$ can be computed for an arbitrary cut-point location vector $\boldsymbol{x}$. We can then compute the cut-point location vector $\boldsymbol{x}_i^*$ for image i that minimizes $e_i^C(\vartheta_i, \boldsymbol{x}_i)$ by solving the integer programming (IP) problem

$$\min t \tag{12}$$

s.t.

$$D\boldsymbol{x}_i + E\boldsymbol{f}_i(\vartheta_i, \boldsymbol{x}_i) + G \leq t\mathbf{1} \tag{13}$$

$$x_{i,n-1}w - x_{i,n}w \leq -2o\ \forall n \tag{14}$$

$$x_{i,n}w \in \{1, \ldots, w\}\ \forall n \tag{15}$$

where (13) is componentwise, (14) enforces that the cut-point coordinates are increasing, (15) ensures they are aligned with pixels, and $\mathbf{1}$ is a $N \times 1$ column vector of ones.

Using the IP (12)-(15) for the considered VSN faces two challenges. First, the distribution $F_i(x)$ is not available until the processing of image i is completed, at which point solving the IP problem is no longer necessary. Second, even if one knew $F_i(x)$ before processing image i, solving the IP problem would be computationally intensive. We address these challenges in the following.

C. Percentile-based Cut-point Location Vector Selection

The biggest challenge in solving (12)-(15) is that it needs a prediction of the distribution $F_i(\vartheta_i, x_i)$ of interest points in image i. This prediction would require predicting the locations and appearance/disappearance of all interest points for every image, which is computationally infeasible. To avoid this problem, we propose to approximate the distribution $F_{i-1}(\vartheta_i, x)$ of interest points through its percentiles, and to predict the approximation of the distribution $F_i(\vartheta_{i+1}, x)$ for the optimization through *predicting the percentiles*. Here we focus on the approximation and the optimization, and will compare various predictors in Section VII.

We approximate the distribution $F_i(\vartheta_i, x)$ with the distribution $\tilde{F}_i(\vartheta_i, x)$, obtained as the linear interpolation of $F_i(\vartheta_i, x)$ between its values at Q percentiles, denoted by $\boldsymbol{\xi} = \xi_1, \ldots, \xi_Q$,

$$\tilde{F}_i(\vartheta_i, x) = \frac{x - \xi_{q-1}}{\xi_q - \xi_{q-1}}\pi_q + \Pi_{q-1}, \tag{16}$$

where $\xi_0 = 0$, $\xi_{q-1} < x \leq \xi_q$, $\pi_q = F_i(\vartheta_i, \xi_q) - F_i(\vartheta_i, \xi_{q-1})$ is the portion of interest points in the interval $\xi_{q-1} < x \leq \xi_q$, and $\Pi_{q-1} = F_i(\vartheta_i, \xi_{q-1})$ is the portion of interest points left of ξ_{q-1}. $\tilde{F}_i(\vartheta_i, x)$ is a non-decreasing, non-negative, continuous, piecewise linear function, which we can use to compute the approximate number of interest points assigned to node n for cut-point location vector $\boldsymbol{x}_i$ as

$$\tilde{f}_{i,n}(\vartheta_i, \boldsymbol{x}_i) = M^* \left(\tilde{F}_i(\vartheta_i, x_{i,n}) - \tilde{F}_{i,x}(\vartheta_i, x_{i,n-1})\right). \tag{17}$$

We can use (17) to express the approximate time needed for interest point detection

$$
\begin{aligned}
\tilde{T}_{det,i} &= \tilde{D}_4 \boldsymbol{x}_i + \tilde{G}_4, \text{where}\\
\tilde{d}_{4,m,n} &= \begin{cases} \frac{hw}{P_{d,px,n}} + \frac{M^*}{P_{d,ip,n}} \frac{\pi_q}{\xi_q - \xi_{q-1}}, & m = n \\ \frac{-hw}{P_{d,px,n+1}} - \frac{M^*}{P_{d,ip,n+1}} \frac{\pi_r}{\xi_r - \xi_{r-1}}, & m = n+1 \\ 0, & \text{otherwise} \end{cases}\\
\tilde{g}_{4,n} &= \frac{M^*}{P_{d,ip,n}} \left(\frac{\xi_{r-1}\pi_r}{\xi_r - \xi_{r-1}} - \frac{\xi_{q-1}\pi_q}{\xi_q - \xi_{q-1}} + \Pi_{q-1} - \Pi_{r-1} \right), \forall n
\end{aligned}
$$

and the approximate time needed for descriptor extraction

$$
\begin{aligned}
\tilde{T}_{ext,i} &= \tilde{D}_5 \boldsymbol{x}_i + \tilde{G}_5, \text{where}\\
\tilde{d}_{5,m,n} &= \begin{cases} \frac{M^*}{P_{e,ip,n}} \frac{\pi_q}{\xi_q - \xi_{q-1}}, & m = n \\ -\frac{M^*}{P_{e,ip,n+1}} \frac{\pi_r}{\xi_r - \xi_{r-1}}, & m = n+1 \\ 0, & \text{otherwise} \end{cases}\\
g_{5,n} &= \frac{M^*}{P_{e,ip,n}} \left(\frac{\xi_{r-1}\pi_r}{\xi_r - \xi_{r-1}} - \frac{\xi_{q-1}\pi_q}{\xi_q - \xi_{q-1}} + \Pi_{q-1} - \Pi_{r-1} \right), \forall n
\end{aligned}
$$

By forming the matrices $\tilde{D} \triangleq D_1 + \tilde{D}_4 + \tilde{D}_5$ and $\tilde{G} \triangleq G_1 + G_2 + G_3 + \tilde{G}_4 + \tilde{G}_5$, we obtain for the approximate completion times of the nodes the set of linear equations

$$\tilde{T}_i = \tilde{D}\boldsymbol{x}_i + \tilde{G}. \tag{18}$$

The cut-point location vector $\tilde{\boldsymbol{x}}_i^*$ that minimizes (18) can be obtained by solving the integer-linear programming problem

$$\min t \tag{19}$$

s.t.

$$
\begin{aligned}
\tilde{D}\boldsymbol{x}_i + \tilde{G} &\leq t\mathbf{1} && (20)\\
x_{i,n-1}w - x_{i,n}w &\leq -2o \;\; \forall n && (21)\\
x_{i,n}w &\in \{1, \ldots, w\} \;\; \forall n && (22)
\end{aligned}
$$

Since (20) is piece-wise linear, a linear relaxation of the problem can be solved efficiently, and the rounding error is negligible if the distribution is reasonably smooth. Observe that by using $Q = f_i(\vartheta_i)$ percentiles, the approximate distribution $\tilde{F}_x(\vartheta, x)$ is a linear interpolation of $F_x(\vartheta, x)$.

An important question is how close to optimal would be the completion time of the processing with this approximate solution. To answer this question we introduce $T_i^N(\vartheta_i, \boldsymbol{x}_i^*) = \max_n(T_i(\vartheta_i, \boldsymbol{x}_i^*))$, the optimal processing completion time in the VSN based on (12)-(15), $\tilde{T}_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) = \max_n(T_i(\vartheta_i, \tilde{\boldsymbol{x}}_i^*))$ the optimal completion time with the linear interpolation according to (19)-(22), and finally $T_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*)$ the experienced completion time if $\tilde{\boldsymbol{x}}_i^*$ is applied for the real distribution $F_x(\vartheta, x)$. In the following we give a bound on the maximum difference between $T_i^N(\vartheta_i, \boldsymbol{x}_i^*)$ and $\tilde{T}_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*)$

Proposition 1. *For any $\epsilon > 0$ there exists Q such that $T_i^N(\vartheta_i, \boldsymbol{x}_i^*) \leq \tilde{T}_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) + \epsilon$.*

Proof: As $T_i^N(\vartheta_i, \boldsymbol{x}_i^*) \leq T_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*)$, we prove $T_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) \leq \tilde{T}_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) + \epsilon$. Despite the linear interpolation, for each node n the components of $\tilde{T}_{i,n}(\vartheta_i, \tilde{\boldsymbol{x}}_i^*)$ and $T_{i,n}(\vartheta_i, \tilde{\boldsymbol{x}}_i^*)$ are identical: the transmission time, the sub-area size dependent part of the detection time, and the detection and extraction times that depend on the interest points in the percentiles following $\tilde{x}_{i,n-1}^*$ and preceding $\tilde{x}_{i,n}^*$. If we define $\Delta_i = max_x|F_i(\vartheta_i, x) - \tilde{F}_i(\vartheta_i, x)|$, we can obtain the worst case bound $\tilde{\epsilon}_n = T_{i,n}(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) - \tilde{T}_{i,n}(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) \leq 2\Delta_i \left(\frac{1}{P_{d,ip,n}} + \frac{1}{P_{e,n}}\right)$, and consequently $\tilde{\epsilon} = T_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) - \tilde{T}_i^N(\vartheta_i, \tilde{\boldsymbol{x}}_i^*) \leq 2\Delta_i \max_n \left(\frac{1}{P_{d,ip,n}} + \frac{1}{P_{e,n}}\right)$.

Let us now consider Q quantiles. The number of interest points between neighboring quantile points is $\frac{f_i(\vartheta_i)}{Q} - 1 = \Delta_i$, and consequently we have

$$\tilde{\epsilon} \leq 2\left(\frac{f_i(\vartheta_i)}{Q} - 1\right) \max_n \left(\frac{1}{P_{d,ip,n}} + \frac{1}{P_{e,n}}\right). \tag{23}$$

Thus, $\tilde{\epsilon} \leq \epsilon$ for any $Q \geq \frac{2f_i(\vartheta_i)T_p}{\epsilon + 2T_p}$, with $T_p = \frac{1}{P_{d,ip,n}} + \frac{1}{P_{e,n}}$. ■

D. On-line Cut-point Location Vector Optimization

So far we considered minimizing the expected completion time, assuming that data are transmitted with the expected transmission time coefficients C_n. The actual transmission times are however random, and would differ from the expected values. In the following we address whether one should recompute the cut-point location vector after the data transmission to node m completes, to further minimize the completion time of the distributed processing.

Let us consider image i and denote the expected time of completing the transmission to node m by $\tau_{i,m}(\vartheta_i, \boldsymbol{x}_i^*)$, and the expected remaining time until completing the processing of the image by $\tau_{i,m+}(\vartheta_i, \boldsymbol{x}_i^*)$, such that $\tau_{i,m}(\vartheta_i, \boldsymbol{x}_i^*) + \tau_{i,m+}(\vartheta_i, \boldsymbol{x}_i^*) = T_i^N(\vartheta_i, \boldsymbol{x}_i^*)$ according to (12)-(15). Let us furthermore denote by $\tau_{i,m}^m(\vartheta_i, \boldsymbol{x}_i^*)$ the experienced time of completing the transmission to node m, using the optimal cut-point vector $\boldsymbol{x}_i^*$. Also we denote by $\boldsymbol{x}^m$ and $\boldsymbol{x}^{m+}$ the first m and the remaining $N - m$ elements of vector $\boldsymbol{x}$.

Proposition 2. *For all $m = 1 \ldots N-1$, $\boldsymbol{x}_i^{m+*} = \{x_{i,m+1}^*, x_{i,m+2}^*, \ldots x_{i,N}^*\}$, that is, the cut-point location vector calculated according to (12)-(15) minimizes the expected completion time $T_i^{N,m}(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}])$ for any given $\tau_{i,m}^m(\vartheta_i, \boldsymbol{x}_i^*)$.*

Proof: We prove the theorem by contradiction. $T_i^{N,m}(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}])$ can be minimized by minimizing the remaining expected completion time $\tau_{i,m+}(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}])$, where $\boldsymbol{x}_i^{m+}$ is arbitrary.

Assume now that there exists $\boldsymbol{x}_i^{m+} \neq \boldsymbol{x}_i^{m+*}$, such that $\tau_{i,m+}^m(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}]) < \tau_{i,m+}^m(\vartheta_i, \boldsymbol{x}_i^*)$. Then, exchanging $\boldsymbol{x}_i^{m+*}$ with $\boldsymbol{x}_i^{m+}$, the completion time expected before the start of the transmission of the first subarea of image i would be $\tau_{i,m-}(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}]) + \tau_{i,m+}(\vartheta_i, [\boldsymbol{x}_i^{m*}, \boldsymbol{x}_i^{m+}]) < T_i(\vartheta_i, \boldsymbol{x}_i^*)$, which is a contradiction. ■

Thus, the optimal cut-point location vector need not be recomputed after the transmission starts.

VI. Scheduling order

From [16] it is known, that for given scheduling order, that is, given order of transmission to the processing nodes, the task completion time is minimized, if all the processing nodes completes the processing at the same time, while the achievable minimum depends on the scheduling order. Below we show that the existence of data transmission overlap affects the optimal scheduling method. We show that to minimize the completion time decisions need to be made: i) on the order of the transmission to the utilized processors, ii) on the number of processors to be utilized, and iii) whether the overlap should be transmitted multicast or by separate, unicast transmission to the two involved processors.

Let us consider the simplified case, when the processing time is proportional to the amount of received data, that is, $P_n = P_{d,px,n}$ and $P_{d,ip,n} = P_{d,e,n} = 0$, and there are only two processing nodes $\mathcal{N} = \{A, B\}$.

Proposition 3. *If overlap is not required, i.e. $o = 0$, the completion time is minimized by scheduling the nodes in increasing order of per bit transmission time.*

Proof: Here we recall the proof for $N = 2$. The extended version can be found in [17]. Consider two processing nodes, A and B, with $C_A \leq C_B$ and arbitrary processing capacities P_A and P_B. When node A is scheduled before node B, the completion times for image i are

$$T_{i,AB} = hw \begin{bmatrix} x_{i,1}C_A + \frac{x_{i,1}}{P_A} \\ x_{i,1}C_A + (1 - x_{i,1})C_B + \frac{1-x_{i,1}}{P_B} \end{bmatrix}. \tag{24}$$

This gives optimal cut-point location, under which the processing at node A and B completes at the same time

$$x^*_{i,1,AB} = \frac{P_A(1 + C_B P_B)}{P_A + P_B + C_B P_A P_B}, \tag{25}$$

and the resulting minimum completion time is

$$T^*_{i,AB} = \frac{(1 + C_A P_A)(1 + C_B P_B)}{P_A + P_B + C_B P_A P_B}. \tag{26}$$

Scheduling the nodes in the reverse order gives

$$T_{i,BA} = hw \begin{bmatrix} x_{i,1}C_B + (1 - x_{i,1})C_A + \frac{1-x_{i,1}}{P_A} \\ x_{i,1}C_B + \frac{x_{i,1}}{P_B} \end{bmatrix}, \tag{27}$$

The optimal cut-point location in this case is

$$x^*_{i,1,BA} = \frac{P_B(1 + C_A P_A)}{P_A + P_B + C_A P_A P_B}, \tag{28}$$

and minimum completion time becomes

$$T^*_{i,BA} = \frac{(1 + C_A P_A)(1 + C_B P_B)}{P_A + P_B + C_A P_A P_B}. \tag{29}$$

Assume, $T^*_{i,BA} < T^*_{i,AB}$. As (26) and (29) differ only in one term in the denominator, $T^*_{i,BA} < T^*_{i,AB} \rightarrow C_A > C_B$, which contradicts the initial assumption $C_A \leq C_B$. ■

Now we introduce transmission overlap $o > 0$.

Proposition 4. *Consider overlap $o > 0$. There exists some configuration of per bit transmission times and processing rates for which the scheduling order in increasing per bit transmission times is not optimal.*

Proof: Consider, as before $\mathcal{N} = \{A, B\}$, with $C_A \leq C_B$ and arbitrary P_A and P_B. The overlap is transmitted via multicast transmission with C_B.

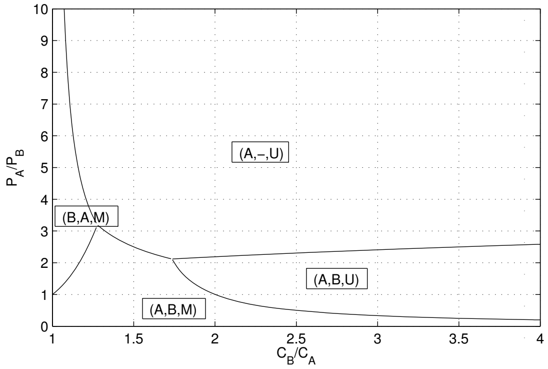

Figure 3: Optimal transmission scheduling schemes as a function of C_B/C_A and P_A/P_B, for $C_A = 1/P_A$, $C_A \leq C_B$ and $o = 0.2$.

Transmitting first to node A the completion times are

$$T_{i,AB} = hw \begin{bmatrix} (x_{i,1} - o)C_A + 2oC_B + \frac{x_{i,1}}{P_A} \\ (x_{i,1} - o)C_A + (1 - x_{i,1} + o)C_B + \frac{1-x_{i,1}}{P_B} \end{bmatrix}, \tag{30}$$

and in the reverse order they become

$$T_{i,BA} = hw \begin{bmatrix} (x_{i,1} + o)C_B + (1 - x_{i,1} - o)C_A + \frac{1-x_{i,1}}{P_A} \\ (x_{i,1} + o)C_B + \frac{x_{i,1}}{P_B} \end{bmatrix}. \tag{31}$$

The optimal cut-point location and the related minimum completion time can be calculated as in Proposition 3. The expressions are rather cumbersome in this case. However, as $\frac{C_B}{C_A} \rightarrow \infty$, they give $T^*_{i,BA} < T^*_{i,AB}$, if

$$\frac{P_A}{P_B} > \frac{1 - o}{o} \tag{32}$$

There is thus a ratio of processing rates for which reversed scheduling order, with increasing per bit transmission times, is optimal. ■

Similar derivations provide the (C_A, P_A, C_B, P_B) parameter combinations where the unicast transmission of the overlap area is preferable, and when only one of the processors should be utilized. Leaving the exact expressions aside, in Figure 3 we show representative results, with parameters $C_A = 1/P_A$, $C_A \leq C_B$ and $o = 0.2$ Under given $C_A P_A$ product and o value, the optimal transmission scheduling is a function of the ratios C_B/C_A and P_A/P_B. Only a single processor, processor A should be used in the parameter region marked with (A,-,U), that is, when the relative transmission time to A is low, and its relative processing speed is high. Moreover, the border of this region does not depend on the value of the overlap. If there is a significant difference in the transmission times, but the processing speeds are similar, then both of the processors should be used, and the overlap areas should be transmitted separately with unicast transmission. In the case of unicast transmission, it always holds that the fastest link should be scheduled first. This region is marked as (A,B,U) on the figure. Finally, according to Proposition 4, when the multicast transmission of the overlap area is optimal, the scheduling

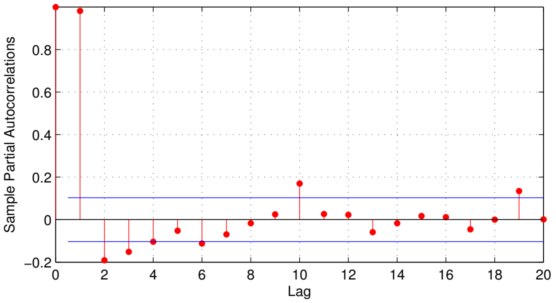

Figure 4: The partial autocorrelation function of the optimal threshold ϑ_i^* time series of the Pedestrian video sequence.

order depends on the ratio of the processing speeds, leading to parameter regions (A,B,M) and (B,A,M).

VII. Numerical Results

We performed simulations to evaluate the proposed algorithms on two surveillance video traces, both with resolution 1920 × 1080, and frame rates of 25 frames per second. One trace, referred to as the "Pedestrian" trace, consists of 375 frames and shows a pedestrian intersection with people moving horizontally across the field of view, covering and uncovering interest points in the background. The other trace, referred to as the "Rush hour" trace, consists of 473 frames and shows a road with vehicles moving slowly along the camera's line of sight, leading to mostly minor changes in the horizontal distribution of interest points. The characteristics of the Pedestrian trace make feature extraction optimization more challenging.

The VSN we consider uses the BRISK [4] scheme for interest point detection and feature description extraction, with $M^* = 400$ as the target number of interest points. When not otherwise noted the VSN has $N = 6$ processing nodes, all with equal processing rates similar to those of an Intel iMote ($P_{d,px} = 9{\times}10^4$ px/s, $P_{d,ip} = 94$ ip/s, $P_e = 25$ ip/s) and transmission time coefficients ($C = 6.7 \times 10^{-8}$ s/bit), and we use $Q = 10$ quantiles for the approximation of the interest point distribution $F_i(\vartheta, x)$, i.e., $\tilde{F}_i(\vartheta, \xi_q) = \frac{q}{Q}, q = 1, 2, \ldots, Q$.

We normalize the performance results to the performance of a non-adaptive offline scheme, and we compare the performances to that of a simple last value predictor denoted by $Y(i-1)$. The offline scheme has complete knowledge of all parameters in each frame; it uses a static detection threshold $\vartheta^s = \arg\min_\vartheta e^D(\vartheta)$ and a static cut-point location vector that minimizes the completion time assuming the interest point distribution is $F(\vartheta, x) = \frac{1}{I}\sum_{i=1}^{I} F_i(\vartheta_i^*, x)$. The last value predictor assumes that the content of image i is identical to that of image $i-1$.

A. Detection threshold prediction

Figure 4 shows the partial autocorrelation function of the optimal threshold values ϑ_i^* for the Pedestrian trace, similar results were found for the Rush hour trace. The figure suggests that autoregressive (AR) models up to order 10 should be considered for predicting $\hat{\vartheta}_i$.

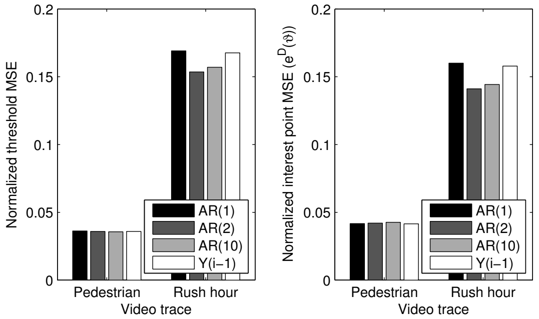

Figure 5: Mean square error of four threshold predictors in terms of threshold ϑ_i and in terms of detected interest points ($e^D(\boldsymbol{\vartheta})$).

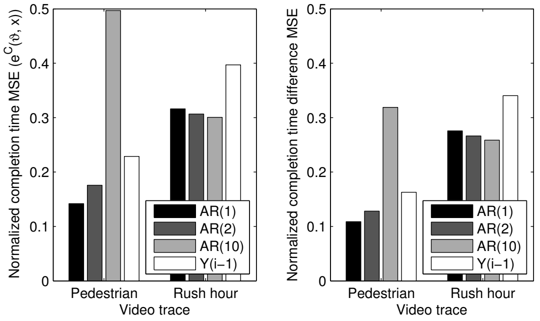

Figure 6: Mean square error of the completion time (e^C) and of the completion time difference of four percentile predictors.

Figure 5 shows the performance in terms of MSE of three AR models and of the last value predictor $Y(i-1)$. The AR models are initially trained using the first 100 frames of the trace and are then retrained after each frame. The left plot shows the MSE of the threshold prediction, i.e., $\frac{1}{I}\sum_{i=1}^{I}(\vartheta_i^* - \hat{\vartheta}_i)^2$, the right plot shows the MSE in terms of detected interest points, i.e., $e^D(\boldsymbol{\vartheta})$. The MSE results are normalized by the corresponding MSE of the offline scheme. The figure shows that threshold prediction decreases the MSE compared to the offline scheme by a factor of 5 to 20 depending on the trace. At the same time the gain of using a higher order predictor is small when compared to the last value or the AR(1) predictor, especially for the Pedestrian trace.

B. Completion time minimization

Figure 6 shows results for the completion times using the proposed percentile based prediction, i.e., each of the Q percentile points is predicted by an AR model or by the last value predictor. Prediction again decreases the MSE by up to a factor of 10 compared to the offline scheme. The two traces show different results in the performance of the predictors. For the Rush hour trace there is some advantage of choosing a higher order predictor, although the marginal performance gain decreases as the order increases. In this case the choice of the predictor should be based on the trade-off between the achieved performance and the computational complexity of training the predictor. For the Pedestrian trace, however,

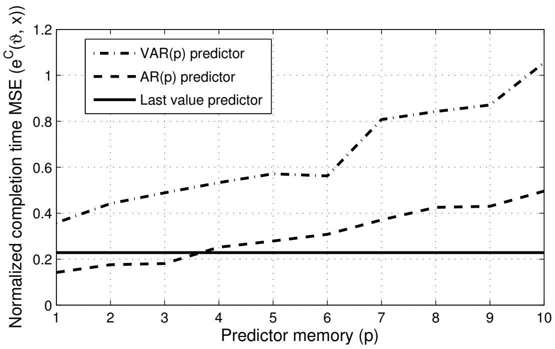

Figure 7: Mean square error of completion times for different predictors and for varying p, under percentile prediction.

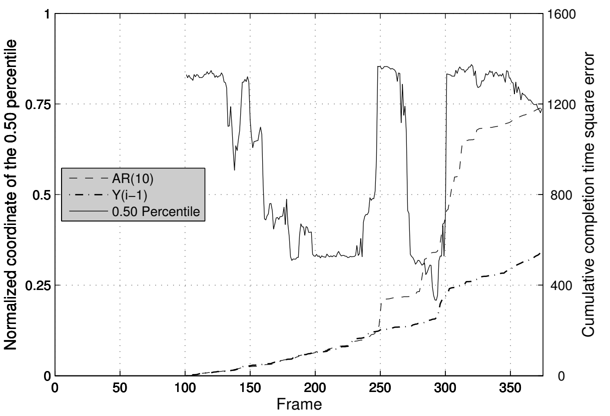

Figure 8: Cumulative square errors of two different predictors, together with the coordinate of the 0.50 interest point distribution percentile.

we see very different results, as the performance does not seem to improve with higher prediction order; the AR(10) performs significantly worse than the last value predictor. In the following we discuss the reasons for this counter-intuitive result.

Figure 7 shows the completion time MSE achieved when using AR and vector AR (VAR) predictors for percentile prediction as a function of the predictor order p, again normalized by the MSE of the offline scheme. We see that the performance of the AR predictor decreases with increased prediction order. Interestingly, the VAR predictor, which could capture any correlation between the different percentile coordinates, performs consistently worse than the independent AR predictors.

To explain the reason for the poor performance of high order predictors, Figure 8 shows the evolution of the cumulative square error (i.e., not normalized by the number of images I) for the sequence of images for the AR(10) and the last value predictor. The results confirm that due to the longer memory of the AR(10) predictor it needs longer time to adjust to large and sudden changes in the image contents. We see, for instance, that a large portion of the total square error for AR(10) emerges during frames 250–320. These frames correspond to a 3 second part of the trace where a tight cluster of interest points in the right side of the scene is first revealed, concealed, and then revealed again very suddenly. Another reason why the last value predictor can outperform the AR predictors is that the error criterion used to train the predictors is not the deviation from the minimal completion time but the error in predicting the percentile coordinates. As the interest points tend to appear in clusters, a small error in percentile prediction and cut-point selection can produce a large discrepancy between the actual number of interest points in the slices.

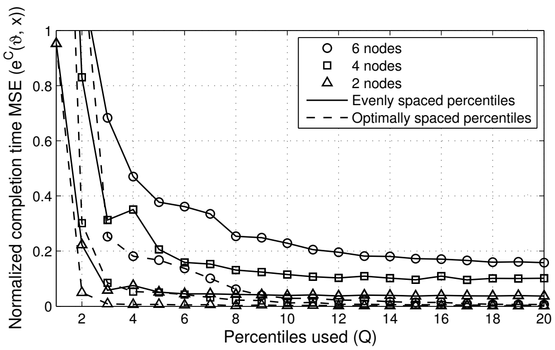

Figure 9: Mean difference in completion times as a function of the number of percentiles used for distribution approximation.

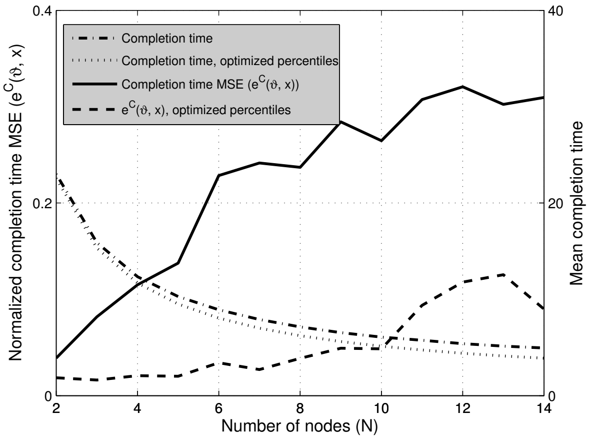

Figure 10: Completion time MSE and mean completion time as a function of N.

C. Approximation of the interest point distribution

So far we used quantiles as the percentiles for approximating the interest point distributions. Figure 9 compares the MSE of the quantile based approximation to an approximation that chooses the percentiles so as to minimize the square error of the approximation. The predictor used is the last value predictor. The figure shows that optimizing the percentiles improves the prediction performance significantly and reduces the number of percentiles needed for the same performance, especially when the number of processing nodes is high ($N = 6$). However, achieving this performance improvement comes at the price of optimizing the percentile locations, which is again computationally intensive.

In Figure 10 we show the MSE (left axis) of the last value predictor as a function of the number of nodes N for $Q = 10$ percentiles. We see that the relative gain of performing the percentile optimization increases as N increases. On the right axis we show the mean completion times achieved using the two approximations. It is worth noting that the difference in terms of mean completion time is rather small, which indicates that the large difference in terms of MSE is due to occasional large errors caused by the quantile-based approximation, which are penalized by the quadratic error function. Consequently, if large completion times can be tolerated occasionally then the quantile based approximation with the last value predictor constitute a computationally simple algorithm with good performance.

VIII. Conclusion and Future Work

We considered the problem of minimizing the completion time of distributed interest point detection and feature extraction in a visual sensor network. We formulated the problem as a stochastic multi-objective optimization problem. We proposed a regression scheme to support the prediction of the detection threshold so as to maintain a target number of interest points, and a prediction scheme based on a percentile-based approximation of the interest point distribution for minimizing the completion time. Our numerical results show that prediction is essential for achieving good system performance. The gain of high order predictors is moderate in general, and depending on the characteristics of the video trace it may even be detrimental to system performance to use higher order prediction models. Our results show that the simple AR(1) and the last value predictors together with a quantile-based approximation of the interest point distribution offer good performance at low computational complexity, making them good candidates for use in visual sensor networks.

Our model could be extended to fast fading and correlated wireless channels and to dynamically evolving network topologies, in which case node unreachability needs to be handled. Another interesting direction for future work could be to maximize the network lifetime under completion time constraints which may require pipelined processing.

References

[1] M. Cesana, A. Redondi, N. Tiglao, A. Grilo, J. Barcelo-Ordinas, M. Alaei, and P. Todorova, "Real-time multimedia monitoring in large-scale wireless multimedia sensor networks: Research challenges," in *EURO-NGI Conference on Next Generation Internet (NGI)*, 2012.

[2] A. Marcus and O. Marques, "An eye on visual sensor networks," *IEEE Potentials*, vol. 31, no. 2, pp. 38–43, 2012.

[3] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[4] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[5] L.-Y. Duan, X. Liu, J. Chen, T. Huang, and W. Gao, "Optimizing JPEG quantization table for low bit rate mobile visual search," in *Proc. of IEEE Visual Communications and Image Processing Conference (VCIP)*, 2012.

[6] J. Chao, H. Chen, and E. Steinbach, "On the design of a novel JPEG quantization table for improved feature detection performance," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2013.

[7] V. R. Chandrasekhar, S. S. Tsai, G. Takacs, D. M. Chen, N.-M. Cheung, Y. Reznik, R. Vedantham, R. Grzeszczuk, and B. Girod, "Low latency image retrieval with progressive transmission of CHoG descriptors," in *Proc. of the ACM Multimedia Workshop on Mobile Cloud Media Computing*, 2010.

[8] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[9] A. Redondi, L. Baroffio, J. Ascenso, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization of binary descriptors," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2013.

[10] A. Redondi, M. Cesana, and M. Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2012.

[11] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[12] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[13] L. Baroffio, M. Cesana, A. Redondi, S. Tubaro, and M. Tagliasacchi, "Coding video sequences of visual features," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2013.

[14] A. Redondi, L. Baroffio, A. Canclini, M. Cesana, and M. Tagliasacchi, "A visual sensor network for object recognition: Testbed realization," in *Proc. of International Conference on Digital Signal Processing (DSP)*, 2013.

[15] M. A. Khan, G. Dan, and V. Fodor, "Characterization of SURF interest point distribution for visual processing in sensor networks," in *Proc. of International Conference on Digital Signal Processing (DSP)*, 2013.

[16] V. Bharadwaj, D. Ghose, and T. Robertazzi, "Divisible load theory: A new paradigm for load scheduling in distributed systems," *Cluster Computing*, vol. 6, no. 1, pp. 7–17, 2003.

[17] V. Bharadwaj, D. Ghose, and V. Mani, "Optimal sequencing and arrangement in distributed single-level tree networks with communication delays," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 9, pp. 968–976, 1994.

[18] B. Veeravalli, X. Li, and C.-C. Ko, "On the influence of start-up costs in scheduling divisible loads on bus networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 12, pp. 1288–1305, 2000.

[19] C. Tang and P. K. McKinley, "Modeling multicast packet losses in wireless lans," in *Proc. of ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2003.

[20] J. Lacan and T. Perennou, "Evaluation of error control mechanisms for 802.11b multicast transmissions," in *Proc. of International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2006.

[21] J. Hartwell and A. Fapojuwo, "Modeling and characterization of frame loss process in IEEE 802.11 wireless local area networks," in *Proc. of IEEE Vehicular Technology Conference. (VTC-Fall)*, 2004.

[22] R. Guha and S. Sarkar, "Characterizing temporal SNR variation in 802.11 networks," *IEEE Transactions on Vehicular Technology,*, vol. 57, no. 4, pp. 2002–2013, 2008.

[23] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of ieee 802.15.4 using measurements and simulations," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2006.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[25] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. of European Conference on Computer Vision (ECCV)*, 2010.

[26] "OpenCV." [Online]. Available: http://opencv.org/

[27] F. B. Abdelaziz, "Solution approaches for the multiobjective stochastic programming," *European Journal of Operations Research*, vol. 216, pp. 1–16, 2012.

[28] E. Eriksson, G. Dán, and V. Fodor, "Prediction-based load control and balancing for feature extraction in visual sensor networks," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.