Please cite as Mohammad A. Alsmirat and Nabil J. Sarhan. Intelligent Optimization for Automated Video Surveillance at the Edge: A Cross-Layer Approach. Simulation Modelling Practice and Theory, Volume 105, 102171, December 2020, Elsevier. DOI: https://doi.org/10.1016/j.simpat.2020.102171.

Intelligent Optimization for Automated Video Surveillance at the Edge: A Cross-Layer approach

Mohammad Alsmirat

Department of Computer Science
Jordan University of Science and Technology, Irbid, Jordan
Email: masmirat@just.edu.jo

Nabil J. Sarhan

Department of Electrical and Computer Engineering
Wayne State University, Detroit, Michigan, USA
email: nabil@wayne.edu

Abstract

The interest in large scale automated video surveillance systems and the interest in using cloud in supporting such systems has increased dramatically. Unfortunately, building such a large system requires huge resources (for processing and storage) and very high network bandwidth. This paper, proposes a framework for resources efficient intelligent automated surveillance framework that utilizes edge servers. In this framework, multiple video sources capture and send videos to a an edge server which performs an intelligent computer vision based cross-layer optimization of the video sources hardware resources and the network bandwidth. The network can be a wireless network and the sources can be mobile. The proposed solution changes the application rates and the required link layer parameters (the transmission opportunities in our case study) of the sending video sources according to the dynamic network conditions to maximize the overall accuracy of the computer vision algorithm(s) of interest in the system. The proposed framework utilizes an enhanced effective airtime estimation algorithm utilizing a *Proportional Integral Differential* (PID) controller that measure the available useful bandwidth in the network. Furthermore, we propose a bandwidth pruning mechanism to reach any desired tradeoff between the computer vision algorithm accuracy and the energy consumption of the video sources. We evaluate and present the effectiveness of the proposed framework, the effective airtime estimation

algorithm, and the proposed bandwidth pruning mechanism, through extensive experiments using OPNET.

Key words: Automated Video Surveillance, Bandwidth Allocation, Computer Vision, Cross-Layer Optimization, Detection Accuracy, Effective Airtime Estimation, Wireless Networks, WLAN, edge computing, IoT

---

1. Introduction

Large scale video surveillance systems popularity has increased dramatically during the last decade. Automated video surveillance (AVS) comes to help in real-time detection of threats and for monitoring of their progress in large scale surveillance systems efficiently.

Most of the research that can be found in the literature on AVS focuses mostly on computer vision algorithms and their robustness in object and event detection, tracking, and classification [1–9]. Moreover, despite the fact that many companies offer IP-based surveillance systems, the design of an automated, scalable, Intelligent, and massively distributed surveillance system remains a significant research problem [10, 11]. Further more, only few researches considered the scalability and the cost of video surveillance systems [10, 12–15] and most of them did not consider building intelligent systems that use computer vision. In large scale surveillance systems, balancing cost and scalability is a greatly challenging problem. That is because, to increase the system coverage, more video sources should be added which causes the increase of the required computational capability of the system and the bandwidth of the system network to process and transmit all the captured video streams by these video sources. (The fact that increasing the bandwidth also increases the computational cost applies to most practical circumstances.) Even when a distributed processing architecture is used, the cost of such a system can still be a big concern as computer vision algorithms are computationally intense especially if they employ modern deep learning algorithms and require continuous learning [2]. Another major problem that effects large scale systems is power consumption, especially if some of the system resources are battery-powered. Considering that video sensors consume orders of magnitude more resources than scalar

sensors, reducing power consumption is essential even when the power is available [16].

Edge computing comes to solve problems such as scalability and service availability [10, 17] in cloud supported systems. Solving these problems is achieved by installing a small server at the edge of the network (called edge server) that provides the services that the cloud server provides but closer to the nodes in the network. Please refer to our paper [10] that contains a thorough review on edge computing, its application, and installations options.

The system considered in this paper is an AVS system that has multiple video sources (or stations) that streams video to an edge server (station) over some network that can be wireless. As a case study in this paper, IEEE 802.11 wireless LAN (WLAN) (WiFi) network is being studied. Figure 1 show an example of such a system. As shown in the figure, the system is composed of multiple cells and the communication medium is shared between all the video sources (video sensors or cameras) in each cell. Also, the video sources can be mobile and they can battery powered. As it is also depicted in the figure, each cell have an edge server that is co-located with the cell access point and the connection between them is very high bandwidth and thus cannot be considered as a bottleneck. The edge server is used to execute the computer vision algorithms of interest and it can be used to create and send automated alerts (messages) is suspicious events or if objects are detected in the cell area. In this paper's case study, we use face detection as the computer vision algorithm of interest. The edge server also can forward the surveillance videos it received, or at lease a summery of them, to a central cloud server for central processing.

This study has been inspired by the recent results in [18, 19]. Naturally, the computer vision algorithm accuracy depends on the quality of the generated video by the sources. (Note that the transfer rate and thus the required bandwidth increase with the quality.) These studies demonstrate that the video quality sensitivity of the computer vision algorithms are not as the sensitivity of humans. For example, Figure 2 show the same image compressed using JPEG compression standard with low and high quality settings. We than applied the Viola-John face detection algorithm [20] on both images. As we can see in the figure, the algorithm detects the face in both images even that we

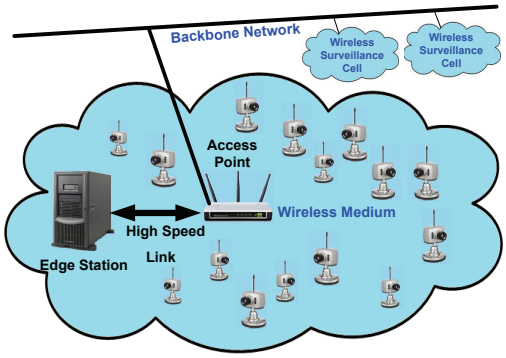


Figure 1: A Large Scale Video Surveillance System

can barley see any thing in the low quality image.

(a) Low Quality (b) High Quality

Figure 2: Compression Impact on Quality Example [Image knex0.gif from CMU/MIT image set [21]]

We address the scalability-cost and power consumption problems of large-scale AVS systems by developing an optimal bandwidth allocation solution that runs on the network edge. The proposed bandwidth allocation solution dynamically controls the sending rates of various sources in the same cell. This is achieved by formulating the

problem of bandwidth allocation as a cross-layer mathematical optimization problem of the total weighted accuracy of a computer vision algorithm (face detection accuracy in our case study) (or alternatively, the total of the computer vision algorithm weighted error) that is constrained by not exceeding the total available network bandwidth in the cell. The weights can be assigned based on many factors, including the potential threat level, placement of video sources, and location importance. Or simply, they can represent the video sources level of importance at the current time. With this formulation, we show that the problem can be solved using Lagrangian relaxation techniques. The solution employs rate-accuracy curves (i.e. accuracy functions of the rate or bandwidth), which are best to be generated for each video source in its designated location. The proposed solution considers three layers: Application, Link, and Physical. The edge server collects all the required information to solve the optimization from the sources that are connected to it and solve the optimization problem. It then distributes the solution to the video sources in the cell.

The main contributions of this paper can be summarized as follows.

- We propose a complete intelligent edge accuracy-based cross-layer optimization solution and we study its performance in details. We have conducted an initial study on this topic in [15] that is being extended greatly in this study. Up to the time of this writing, this is the first cross-layer solution that optimizes the detection accuracy in AVS systems and study its performance in this details. Only a few researches [10, 12–14, 22–25] have considered video quality (or distortion) optimization.
- The proposed solution utilizes a novel *Proportional Integral Differential* (PID) controller algorithm for estimating the effective airtime (a measure of the effective bandwidth) of any wireless medium.
- We develop a model that characterizes the video data rate and face detection accuracy error relationship.
- We propose a mechanism that reduces the bandwidth usage (we call here bandwidth pruning mechanism) to achieve the desired power consumption and detection accuracy tradeoff.

- We study the performance of the proposed framework using different video coders; MJPEG and MPEG-4 part 2.
- The proposed framework is evaluated by streaming real MJPEG videos (one of the most famous video formats in digital video surveillance systems) over a simulated network to extensively test the scalability of the system. OPNET simulator is used for network simulation.
- For a more realistic evaluation, the proposed framework also implements a concealment algorithm [26] at the edge station to reduce the impact of packet loss on video quality.

The extensive evaluation results show that the proposed framework increases the accuracy of face detection applied on the received video streams. The results also show that the accuracy based framework yields a significant power reduction. Moreover, the results show that using PID in the network effective airtime estimation algorithm make it more stable, decreases convergence time, and produces more accurate estimations.

The rest of this paper is organized as follows. Section 2 reviews some background information and the most related work in the literature. Section 3 details the proposed large scale edge accuracy based cross-layer optimization framework. Subsequently, Section 4 presents the methodology used in performance evaluation. Finally, Section 5 demonstrates the main results.

2. Background Information and Related Work

2.1. IEEE 802.11e Standard

The IEEE 802.11 wireless LAN (WLAN) [27] is one of the most used forms of wireless networks now a days. With the introduction of 802.11e standard, differential quality-of-service (QoS) levels provisioning among different access categories (AC) in the same station is enabled, which enhances the support of multimedia applications. IEEE 802.11e is an enhancement that adds QoS to the legacy 802.11 a/b/g standards. 802.11e is very important for delay-sensitive applications, such as voice over IP and video streaming. The 802.11e enhances the Distributed Coordination Function (*DCF*) and the Point Coordination Function *PCF*) in the legacy 802.11 MAC, through a new

coordination function: the Hybrid Coordination Function (HCF). Within HCF, there are two methods of channel access, similar to those defined in the legacy 802.11 MAC : *Hybrid Coordination Function Controlled Channel Access* (HCCA) and *Enhanced Distributed Channel Access* (EDCA). Both EDCA and HCCA define different Traffic Classes. These traffic classes are mapped to access categories that includes Voice, video, Best Effort, and Background where voice traffic has the highest priority and background traffic has the lowest priority. As discussed in [23, 28], HCCA is considered to be more complex and less flexible due to its nature of being a point coordination function. In this study we consider EDCA based networks. Four EDCA parameters determine traffic priorities. These parameters are:

- *Arbitration Inter Frame Space (AIFS)*. AIFS determines the time before an AC starts the transmission when the medium is found to be not busy.
- *Minimum Contention Window (CW_{min}), and Maximum Contention Window (CW_{max}).* If a packet collision happened or the medium is found busy, the AC stops accessing the medium (backs off) for a time that is picked randomly between 0 and CW, where CW is a number that is initially is equal to CW_{min}, and is incremented after every failed transmission until it reaches CW_{max}. This number is reset to CW_{min} after the next first successful transmission of the AC.
- *Transmission Opportunity Time (TXOP)*. TXOP limit determines the period of time that in which the AC can keep transmitting in, when it successfully get accesses to the medium.

From the discussion of EDCA above, different AC with different EDCA parameters allocates different fractions of the available bandwidth in the network. This means that different fractions of the network bandwidth is allocate to the same AC in two different wireless stations if they are given different EDCA parameters. In particular, to assign a fraction of the bandwidth to a station, the frequency-related parameters can be either fixed among stations and transmission opportunity limit is used to control the allocation or the transmission opportunity limit among stations can be fixed and the frequency parameters are used as the controllers. According to [29], both approaches have the same differentiation capabilities.

2.2. Automated Video Surveillance

Major prior work on surveillance systems can be summarized as follows.

- Study [30] proposed a prototype for an urban surveillance system, targeting the high-end of the security market and using a dedicated network for high-quality streaming.
- Studies [31, 32] proposed systems that send still images periodically from the video source to the user so as to reduce the bandwidth requirements.
- Knight [33] is a wide-area surveillance system that detects, tracks, and classifies moving objects across multiple cameras. It transmits videos with fixed encoding parameters to a centralized server.
- SfniX [34] is a surveillance system that supports realtime monitoring and storage of all the video streams, performs video analysis, and answers semantic database queries. Like Knight [33], it transmits videos with fixed encoding parameters to a central server for processing.
- VSAM [35] uses multiple video sensors to provide continuous coverage of people and vehicles in a cluttered area. It sends one low quality video at a time and relies on dedicated workstations for the detection, tracking, and classification of events. Some of the technologies developed by the VSAM project have been commercialized by companies, such as ObjectVideo.
- Study [19] generated rate-accuracy curves for face detection and face tracking algorithms. It simply limited the video rates of all video sources to one value, referred to as the "sweet point". In this study, we develop a comprehensive cross-layer optimization solution. We also develop an accurate rate-accuracy model using multiple datasets.

2.3. Distortion-Based Cross-Layer Optimization for Video Streaming Systems

Numerous studies have discussed cross-layer optimization in video streaming over wireless networks. In this section, we discuss only the most related and the resents ones.

Study [23] formulated and solved an optimization problem that minimizes the sum of distortion in all video streams sent to a central video server that is collocated with the access point. This work used the formulation in [36] to develop an analytical model for the calculation of the effective airtime. As will be discussed in Subsection 2.4, the analytical models involve significant simplifications, approximations, and assumptions. For examples, this study assumed that EDCA should be p-persistent, which uses backoff timer selection process that follows a geometric distribution with parameter p while the standard EDCA backoff timer follows uniform distribution. Moreover, the study calculated the optimal application rate and link layer parameters without taking the packetization overhead of the transport and the application layers into consideration.

The studies in [13, 14] addressed the aforementioned issues in [23] and improved its link-layer adaptation model and the effective airtime estimation.

The studies in [10, 12, 24, 25] extended the work in [13, 14] by implementing the framework in NS3, support mobility in the system, add cloud support for central processing, and add security to the system.

2.4. Effective Airtime Estimation

The effective airtime (EA) is a measure of the useful bandwidth of the network and it is the network time fraction used for the transmission of useful data (without traffic overhead). Solving the mathematical optimization problem the we formulate in this paper needs an accurate measurement or at least an accurate estimation of EA. This will be discussed in more details later. If we know exactly the capacity of the network, we can make the cameras in the system use the network bandwidth fully and without exceeding the network capacity. This can guarantee that the video streams that are received by the edge server is in best quality (produces the best computer vision accuracy) and it also can guarantee that packet collision and re-transmission in the network is minimized. Minimizing collisions and retransmissions is very important because collisions and retransmissions increase power consumption of the wireless cameras. In [28], an ad-hoc network effective airtime was calculated by dividing the total throughput by the physical rate of the network. In this calculation, they assume that the physical rate of all stations in the network is the same. In [23], the authors

calculated the effective airtime by developing an analytical model for video streaming from multiple stations to a central server that is based on the formulation of [36]. The main problem with these models is that they involve many assumptions, approximations, and simplifications. As a result, their developed model was given based only on CW_{min} and the number of data sources in the network. Moreover, in their model, the effective airtime increases with increasing the number of stations and reaches a value close to 100% in a network of size 30 or more.

Studies [37, 38] sought to determine the saturation throughput of a wireless node using analytical models that require solving complex nonlinear equation systems. In addition, they make many limiting assumptions, such as error-free links and constant packet size. Study [37] made additional assumptions, including infinite retry buffers and infinite MAC queue size. Studies [39–41] measured network capacity using packet propping. Other studies, such as [42] (and references within) utilized the wireless station carrier sense capability, idle and busy medium ratios, and other techniques, such as collision prediction to estimate the capacity of the wireless link in ad-hoc networks. This approach is more realistic and accurate, but unfortunately, none of these studies dealt with wireless networks in the infrastructure configuration.

In [13, 14], an online and dynamic effective airtime estimation algorithms were proposed. The algorithms stops when convergence is achieved and then needs to be rerun whenever significant changes in the network happen. More experimentations with the algorithms show that the algorithms may not always converge to an accurate value of the effective airtime because, in some cases, the algorithms are forced to stop at at the end of a period called estimation time even if the algorithms did not achieve convergence.

In this paper, we seek to reduce the convergence time of the estimation, allow the algorithm to run continuously so as to react faster to any network changes, and let the algorithm find an accurate value without forcing the convergence process.

3. Proposed Edge Detection Accuracy Cross-Layer Optimization Framework

In this paper, we consider a case study in which a large scale automated video surveillance (AVS) system, that is constructed of multiple cells, in which each cell has multiple video sources/stations (cameras and/or sensors) stream videos to an edge station using an infrastructure IEEE 802.11 WLAN network in the EDCA mode. The main two challenges in the considered system can be summarized as follows. (1) In each cell, the WLAN limited bandwidth that can be even shared with other devices, should be estimated accurately and the available bandwidth for the AVS should be distributed efficiently over the cell video sources to achieve the maximum computer vision algorithm(s) accuracy at the edge station. (2) Providing differential bandwidth assignment to different sources is required because various sources in the cell have different characteristics, including channel conditions, power constraints, and lighting conditions. In addition, different sources may have different importance levels.

Figure 1 shows a clarification of such a system. Each cell in the system has S sources. The wireless connection between the access point (AP) and each source s in each cell has different physical rate (y_s). This physical rate is affected by the source distance from the AP of the cell and the type of obstacles in the way of communication between the source and the AP. It is possible also that each source s streams different encoded video with different rate R_s. Each source s may also have different weight (w_s). The weight w_s represents the importance level of the source s at the current time and depends on many factors, including the potential threat level at the source location and the location importance. In this paper, the weights are assumed to be already preset to the video sources.

In this study, our ultimate objective is to find a solution that optimally maximizes the detection accuracy at the edge server in each cell by dynamically distributing and allocating the available bandwidth in the cell network among the video sources in the cell. All video, network, system, and environmental aspects, which may change dynamically, should be fully considered in this optimal solution. To be able to achieve that, the proposed solution should be cross-layer solution which may dynamically manages parameters of the Application layer, the Link layer, and the Physical layer in the

network stack.

3.1. Accuracy-Based Optimization Problem Formulation

To find the optimal solution for the previously described cell bandwidth distribution and allocation problem, the problem is formulated as a cross-layer optimization problem of the summation of the weighted detection accuracy error of the computer vision algorithm running at the edge station. Because the wireless medium in the cell is shared among all video sources, the airtime fraction should be determined by the bandwidth allocation solution for each video source. This can be formulated mathematically as shown in Equation 1.

$$Find \quad F^* = \arg\min_{F} \sum_{s=1}^{S} w_s * accuracyError_s(r_s) \tag{1a}$$

$$s.t. \quad \sum_{s=1}^{S} f_s = A_{eff} \tag{1b}$$

$$r_s = f_s \times y_s \tag{1c}$$

$$0 \leq f_s \leq 1 \tag{1d}$$

$$s = 1, 2, 3, ..., S, \tag{1e}$$

where $F^* = \{f_s^* | = 1, 2, 3, ..., S\}$ which is the set of the cell video sources airtime optimal fractions (f_s^*), w_s is video source s importance factor, r_s^* is video source s optimal application-layer rate, A_{eff} is the cell total effective airtime, and y_s is video source s physical-layer rate.

The mathematical Equation in (1) can be solved if the accuracy error function $accuracyError_s(r_s)$ is characterized and the effective airtime of the network is estimated.

3.2. Rate-Accuracy Characterization

As stated earlier, the solution of the optimization problem on hand requires a function that characterizes the relationship between the rate of a video and the accuracy error of a computer vision algorithm applied to the video. To keep the paper focused,

we analyze only the face detection algorithm as a case study. We experiment with both MJPEG and MPEG-4 Part 2. Due to power consumption and other considerations such as accurate frame-based video review, MJPEG are still used in many surveillance applications, especially in battery-operated cameras. The process we follow here, can be used to characterize the accuracy error on any other computer vision algorithm and this is left for another study.

For MJPEG, we use the OpenCV [43] implementation of Viola-Jones algorithm for face detection. As we are working with face detection, we used popular face image sets to determine the size-accuracy error relationship. These image sets include CMU/MIT [21], Georgia Tech [44], and SCFace [45]. We also use these image sets in our experiments to assemble the MJPEG video streams using the streamer discussed in Section 4 to load the network with video traffic. For fair evaluations and comparison of bandwidth allocation solutions, each video source should be able to stream the video at a bitrate that matches the output of the optimization. By studying the video streams that can be generated using the Georgia Tech image set, we find that these steams ranges of bitrates are very limited in the studied network. We find that by lowering the resolution of the images in the image set, a better bitrate ranges variety can be produced. For that reason, we have created two lower resolution versions of original Georgia Tech image set at 30% and 50% of the original resolution. The new image sets have resolutions of 192×144 and 320×240, respectively. Using these sets, the video sources can achieve the rates produced by the optimization more accurately. In SCFace, the images are taken by three cameras at three different distances from the subjects. Effectively, the cameras capture different resolutions of the subjects. We refer to these image sets in decreasing order of the distance as *SCFace at Distance 1*, *SCFace at Distance 2*, and *SCFace at Distance 3*. For each image set, we use MATLAB imwrite function to compress each picture in the set with quality factors ranging from 1 to 100, with 1 being the lowest, and then apply the computer vision algorithm on each image to calculate the accuracy using a predefined ground truth about the location of the faces in the image. We use two metrics for the detection accuracy: *positive index* and *negative index*. The positive index is the number of correctly detected faces divided by the total number of faces, whereas the negative index is determined as the number of incorrectly detected

faces divided by the total number of faces.

Subsequently, we can find the average size, the average positive index, and the average negative index of all images with the same quality factor. The accuracy error can then be calculated as the sum of the total error: $accuracyError = (1 - positiveIndex) + negativeIndex$.

For MPEG-4, we start by generating ground truth for the face positions in each frame of each video sequence. Subsequently, we use *FFmpeg* with the option *qscale* to compress each video with quality scales from 2 to 31, with 2 being the highest quality, and then apply the computer vision algorithm on each video frame to calculate the accuracy using the generated ground truth. As with MJPEG, we use the positive index and negative index for the detection accuracy and apply the same aforementioned procedure.

For comparison purposes, we also perform rate-distortion characterization on the MPEG-4 videos. We followed the same approach as with rate-accuracy characterization. We asses the distortion of each video frame against the uncompressed video frame using the Root Mean Square Error (RMSE) metric. The video distortion can than be calculated as the average frame distortion of all the frames in the video.

The results are shown in Figures 3 and 4 for MJPEG and MPEG-4, respectively. By curve fitting the results using Matlab, we find that the simplest function that characterizes the accuracy error is given by Equation 2

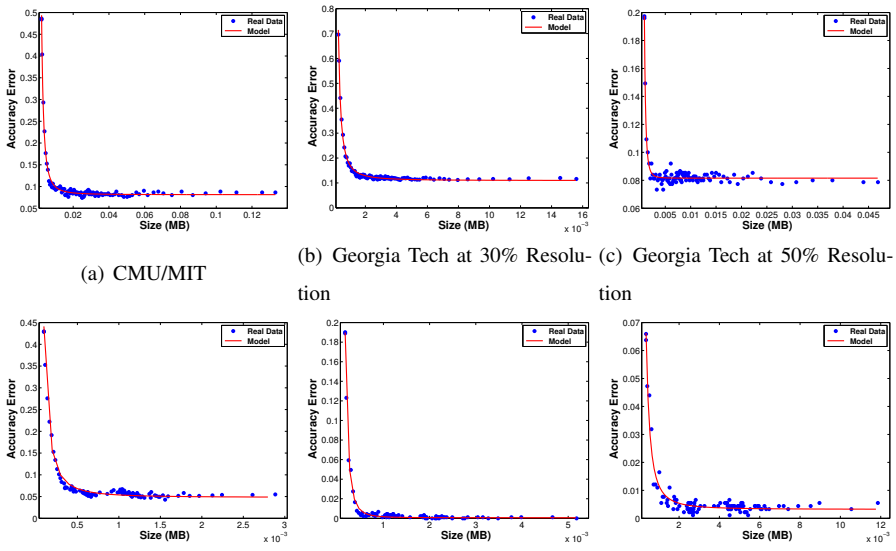
$$accuracyError = a \times I^b + c, \tag{2}$$

where I is the JPEG image size for MJPEG and the sampled frame size in MPEG-4 and a,b, and c are constants. The curve fitting results are demonstrated in Figures 3 and 4 for MJPEG and MPEG-4, respectively. The model in Equation 2 is shown by a solid blue line.

The relationship between the image/frame size I and a source video rate r_s is given by $I = r_s/\tau$, where τ is the video frame rate Given that, the accuracy error function for a source s can be given as $(a_s(r_s/\tau)^{b_s} + c_s)$.

Note that Equation 2 represents both AccuaryError and Distortion with the same model but with different constant values. The accuracy-based formulation in Equation

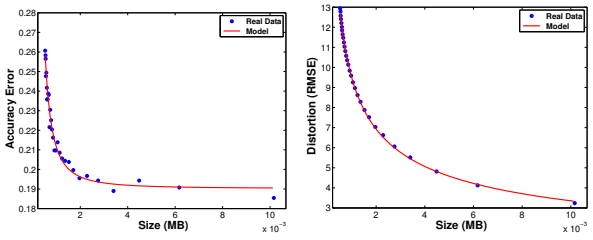
1a uses the AcuracyError shown in Equation 2. The distortion model is used to compare distortion-based optimization (the old approach proposed in [14]) with the accuracy-based optimization proposed in this paper.



(a) CMU/MIT (b) Georgia Tech at 30% Resolution (c) Georgia Tech at 50% Resolution

(d) SCFace at Distance 1 (e) SCFace at Distance 2 (f) SCFace at Distance 3

Figure 3: Rate-Accuracy Characterization for MJPEG



(a) Size-Accuracy for CIF (b) Size-Distortion for CIF

Figure 4: Rate-Accuracy and Rate-Distortion Characterization for MPEG-4

3.3. Effective Airtime Estimation

Because the existing analytical models in [23, 36] for determining the effective airtime involve significant simplifications, approximations, and assumptions, and because the online approach in [13, 14] was promising but still need some enhancement in its stability and convergence, we propose an enhanced dynamic and online algorithm for estimating the cell infrastructure wireless network effective airtime that is based on the approach in [13, 14].

The proposed algorithm is based on the approach proposed in [13, 14] but uses a *Proportional Integral Differential* (PID) controller for adjusting the currently estimated value of the effective airtime in order to achieve a faster, more stable, and more accurate estimation. In addition, the new algorithm can be run continuously so as to react faster to any changes in the network. As depicted in Figure 5, the PID controller utilizes the history values of the effective airtime and the error change rate to adjust the current effective airtime value. The error depends on the dropping rate in the network. The PID controller has three components: *proportional*, *integral*, and *differential*. These components are weighted by constants K_P, K_I and K_D, respectively. The effective airtime is changed using the proportional component based on the error immediate value. The integral and differential components consider the past and the anticipated future error values, respectively, which help the steady state error and the overshoot reduction.

The steps of the new estimation algorithms are shown in Algorithm 1. The algorithm starts as the online algorithms in [13, 14] by finding an initial value for the effective airtime (A_{eff}). The algorithm initially works in an initialization period where each video source in the cell is streaming a video, to the edge server, with a rate that is equal to the maximum network physical rate (network saturation rate) divided by the number of video sources in the network (the number of sources in the network can be distributed in the beacon sent by the AC). During this period, the edge server calculates the throughput (t_s) for each video source s as the source stream is being received by the application layer of the edge server. At the end of this initialization period, the the A_{eff} initial value is calculated as $\sum_{s=1}^{S} t_s/y_s$. This initialization step is needed only once in the first time the system estimates A_{eff}. The algorithm then continue to run

while the system is running and it does not add any extra overhead to the traffic in the network. The algorithm keeps track of time and update A_{eff} value at the end of each period of time, called here, the estimation period. During each estimation period, each video source s, while sending its video stream to the edge server, tracks its sent video packets and calculates its packet dropping rate (d_s) and at the end of the estimation period, the source sends this information to the access point (AP). The AP calculates the overall average dropping ratio as $A_\Delta = \sum_{s=1}^{S} d_s/y_s$. The PID error is then calculated as $A_{thresh} - A_\Delta$ and A_{eff} is adjusted by the PID controller to eliminate the error as illustrated in Figure 5. A_{thresh} controls the allowable dropping in the network and it should be set manually.

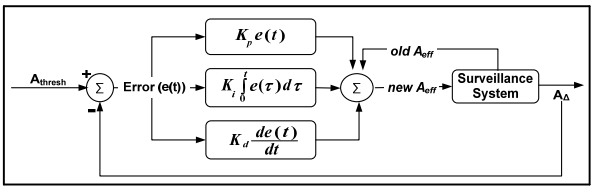


Figure 5: Simplified PID Controller for Effective Airtime Estimation

Algorithm 1: Simplified PID Algorithm for Dynamically Estimating the Effective Airtime

while *initialization period is not expired* **do**

Force each source to send in a rate equals to the maximum physical rate divided by the number of sources;

Update t_s value for each source;

end

$A_{eff} = \sum_{s=1}^{S} t_s/y_s$;

At the end of each estimation period{

$A_{\Delta} = \sum_{s=1}^{S} d_s/y_s$;

BeforeLastError=LastError;

LastError=error;

error = $A_{thresh} - A_{\Delta}$;

$A_{eff} = A_{eff} + K_P \times error - K_I \times LastError + K_D \times BeforeLastError$;

}

We use the following well-established procedure in control theory to tune the three PID parameters. (1) Set K_I and K_D to zeros and increase K_P until the output oscillates, and then set K_P to half of that value. (2) Increase K_I until any offset is corrected in adequate time. (3) Increase K_D until the reference can be reached quickly after load disturbance.

3.4. Optimization Solution

As all the requirements needed to solve Equation (1) are available, the solution goes as follows:

Step 1: To check if the formulation has a global minimum, we check if the problem is a convex programming problem. This can be done by checking that all the problem constraints are linear and convex. By checking the constrains in ((1b)-(1e)), we find that they are linear and it follows that they are convex. As all the constrains are convex and because the derivative of the sum term is monotonically non-decreasing, it follows that the optimization function $\sum_{s=1}^{S}(w_s(a_s(f_s y_s/\tau)^{b_s} + c_s))$ is also convex and it follows that the problem is a budget constrained convex programming problem.

Step 2: The easiest way to solve a budget constrained convex programming problem, is to use the Lagrangian relaxation technique [46]. The Lagrangian relaxation technique requires the problem to be written as a Lagrangian-relaxed formula. The Lagrangian-relaxed formula for our problem is written in Equation 3.

$$L(F^*, \lambda) = \sum_{s=1}^{S}(w_s(a_s(f_s y_s/\tau_s)^{b_s} + c_s)) + \lambda(\sum_{s=1}^{S} f_s - A_{eff}), \tag{3}$$

where $0 \leq f_s \leq 1$, and $s = 1, 2, 3, ..., S$. For each video source s, the Lagrangian relaxation formula is written as $f_s^* = \arg\min_{f_s} w_s \times AccuracyError_s(r_s) + \lambda f_s$. It follows that the optimal solution can be found by solving the equation

$$d(w_s(a_s(f_s y_s/\tau)^{b_s} + c) + \lambda f_s)/df_s = 0. \tag{4}$$

Thus,

$$f_s^* = (\frac{-\lambda^* \tau_s}{w_s a_s b_s y_s (y_s/\tau_s)^{(b_s-1)}})^{(1/(b_s-1))} \tag{5}$$

Now, we need to find the optimal λ^*. By substituting f_s^* as shown in Equation 5 in $\sum_{s=1}^{S} f_s = EA$, we get

$$\sum_{s=1}^{S}(\frac{-\lambda^* \tau_s}{w_s a_s b_s y_s (y_s/\tau_s)^{(b_s-1)}})^{(1/(b_s-1))} = EA, \tag{6}$$

To solve this formulation for λ^*, b_s of the accuracy error curves of all video sources should be assumed to be the same. This is not a problem because it is valid empirically. This give us λ^* as shown in Equation 7.

$$\lambda^* = (\frac{A_{eff}}{\sum_{s=1}^{S}(\frac{-\tau_s}{w_S a_s b_s y_s (y_s/\tau_s)^{(b_s-1)}})^{(1/(b_s-1))}})^{(b_s-1)} \tag{7}$$

3.5. The Allocation Algorithm

The problem solution in Equation 5 and 7 is realized in each cell as follows. λ^* is calculated by the AP using 7 and its value is sent to the video sources in the cell in the beacon packet. Once λ^* is received by a video source s, The source s calculates its airtime fraction f_s^* using Equation 5 and the video source changes its application data rate or the the video encoding rate, using the equation $r_s^* = f_s^* \times y_s$. To make sure

that each source is able to transmit the data its application layer generates, the link-layer parameters are adapted to accommodate the application rate. Either transmission opportunity duration limit (TXOP) or the frequency of the transmission parameters (AIFS, CW_{min}, CW_{max}) can be used to achieve this. The use of TXOP is easier because only one variable is needed to be managed. We use the model proposed in [13, 14] to calculate the needed TXOP.

As in [13, 14], each video source determines its TXOP limit as the time required to transmit the packets that belong to a single video frame along with all associated overhead. This value can be found by using the following equation:

$$x_s^* = [\frac{L_s}{y_s}] + [\frac{O_s N_s}{y_s}] + [(2N_s - 1)t_s] + [N_s t_a], \tag{8}$$

where: L_s represents the maximum video frame size and can be calculated by dividing the max data rate coming from the upper layers of the source by the source frame rate, y_s represents the source physical rate, O_s represents the per-packet average physical and MAC layers overhead, N_s represents the count of MAC-layer data frames in the maximum size video frame, which can be computed by dividing L_s by the average size of video data in each MAC-layer data frame, t_s represents *Short Interframe Space (SIFS)* time, and t_a represents the time required to send a MAC-layer acknowledgment packet.

3.6. Proposed Bandwidth Pruning Mechanism

As discussed earlier, with the optimization solution, each video source can determine its sending application rate. The results in Figure 3 suggest that the detection accuracy increases only slightly with the video rate (application rate) after a certain point. (That point varies based on the system, network, and environmental conditions.) Therefore, we propose a a *bandwidth pruning* mechanism to achieve any desired trade-off between the detection accuracy and power consumption. With this mechanism, each video source adjusts (reduces) its application rate if the anticipated loss in the detection accuracy is below a certain threshold. The threshold can be simply a fixed percentage or a function of the remaining battery energy in the video source and its

importance. The pruning mechanism is of significant importance, especially when the sources are battery-operated.

In this paper, we experiment with 4 levels of bandwidth pruning: 95%, 90%, 80%, and 70%. Each level specifies the percentage of the original accuracy that will be achieved after the pruning mechanism is applied. Level 95%, for example, means that the achieved detection accuracy for each video source after pruning will be 95% of that produced by the optimization solution. In other words, a 5% reduction in the accuracy will be experienced by each source. Obviously, with higher reductions in the bandwidth, greater savings in power consumption will be achieved.

4. Performance Evaluation Methodology

We use OPNET evaluate the effectiveness of the proposed optimization solution, including the effective airtime estimation algorithm and bandwidth pruning mechanism.

To ensure realistic evaluations, a real MJPEG video traffic source is implemented as a new application in OPNET. The application generates MJPEG video traffic in the form of Real-time Transport Protocol (RTP) packets. The MJPEG videos are constructed by choosing random images from standard face image sets that include CMU/MIT [21], Georgia Tech [44], and SCFace [45]. At the edge server application layer, we implement a client (a sink) that assemble the received RTP packets to recreate the sent video frames and apply the used concealment method to try to restore any missing packets. This video sink than collect the data required to calculate our performance metrics. At each video source, the streamer takes a bitrate, a frame rate, and an image set as inputs and produces a corresponding MJPEG video stream. We assume a constant bitrate of 20 frames per seconds.

In the case of MPEG-4, we implement a video streamer that takes a video playback rate and a frame rate and produces raw video packets to simulate a video stream. We use standard YUV video sequences, namely, Foreman, Mother and Daughter, News, and Silent video sequences in CIF resolution. We did not choose higher resolutions, since prior results [47] indicate that computer vision applications are not as sensitive

as humans are to resolution and quality and thus lower resolutions can be chosen to reduce power consumption.

The networks in our experiments contain a set of video sources, an AP, and an edge server that is co-located with the AP. The start time of the video sources is chosen randomly within 1 second from the simulation start time. While running, each source sends its status update in a small packet to the AP every 1 second. This small packet is called *state report* in this paper. It contains the video source weight, physical rate, dropping rate (both buffer and retransmission dropping rates), and local accuracy error model parameters (the constants a,b,c). Using all these data from all video sources in the cell, the AP(or the co-located edge server) performs the effective airtime estimation algorithm and calculates λ for the optimization solution. We preset video sources weights and physical rates using random numbers and we reuse these same random numbers in all experiments. The source weight is chosen from 5 levels between 0 and 1 keeping the sum of all video sources wights to be 1. Each source physical rate is randomly chosen from the values $\{12, 18, 24, 36, 48, \text{and } 54\}$ Mb/s which are the possible physical rates of the 802.11g standard. All the main simulation parameters and their values are summarized in Table 1.

In our analysis, we compare the proposed accuracy-based optimization solution (called *Weighted Accuracy Optimization* (WAO) in the results), with two video distortion based solutions. Namely:

- The solution in [13, 14], called *Enhanced Distortion Optimization* (EDO) in the results, which is the best performer among exiting solutions.
- A new EDO version that uses weights for video sources, called *Weighted Distortion Optimization* (WDO) in the results.

We collect and analyze the following most important **performance metrics**:

- *Weighted Accuracy*, which is the sum of the weighted detection accuracy of each video source. The accuracy for each source is calculated as the average accuracy of the received frames sent by that source. The accuracy of a dropped video frame is assumed to be 0.
- *Overall Network Load*, which is defined as the total load sent by the application

Table 1: Summary of Simulation Parameters

Parameter	**Model/Value(s)**
Number of video sources	16-76
Simulation Time	10 min
Packet Size	1024 bytes
Application Rate	Optimized, Default = Max Physical Rate / No. of Sources
Video Frame Rate	20 frames/sec
Physical characteristics	Extended Rate (802.11g)
Physical Data Rate	Random from {12Mb/s, 18Mb/s, 24Mb/s, 36Mb/s, 48Mb/s, 54Mb/s}
Weight	Random, One of five levels between [0 1]
Buffer size	256 Kb
Video TXOP limit	Optimized, Default = $3008\mu s$
Video CW_{min}	15
Video CW_{max}	31
Video AIFS	2
Short Retry Limit	7
Long Retry Limit	4
Beacon Interval	0.02 second
State Report Interval	1 second

layers of all video sources.

- *Power Consumption*, which is the average power consumption of the wireless interfaces of the video sources and is determined by using the power consumption model in [48].

5. Result Analysis

5.1. Effectiveness of the Proposed Effective Airtime Estimation

Using the aforementioned manual tuning method and extensive experimentation, we observe that the best values for the PID parameters depend on the workload and the desired value of A_{thresh}. Table 2 summarizes these results.

Table 2: Summary of PID Parameter Tuning

Workload	A_{thresh}	K_P	K_I	K_D
Georgia Tech at 30% Resolution	0.001	1	0.25	0.25
Georgia Tech at 30% Resolution	0.005	1.5	0.25	0.25
Georgia Tech at 50% Resolution	0.001	0.5	0.25	0.25
Georgia Tech at 50% Resolution	0.005	1.5	0.25	0.25
CMU/MIT	0.005	0.5	0.25	0.25
MPEG-4 CIF Resolution	0.005	1.5	0.25	0.25

Let us now compare the effectiveness of the proposed effective airtime estimation algorithm utilizing the PID controller with the estimator in [13, 14]. Figure 6 compares the output effective airtime values over time for both algorithms, referred to as "PID Estimator" and "Existing Estimator", respectively. Only the results for Georgia Tech at 30% and 50% of the original resolution are shown. The other datasets exhibit a similar behavior. The results demonstrate that the PID estimator converges much faster and has smaller overshooting and undershooting.

Figure 7 shows the average effective airtime versus the number of video sources for the three bandwidth allocation solutions. These results show that the effective airtime increases with the network size up to a point and then starts to decrease. The peak happens when the proper balance between node contention and network utilization is reached. The peak point in the figure is when the network size is 16, but this value varies with the total sending rate of the sources. In particular, the peak should happen at a smaller network size if the sources are more demanding for bandwidth and at a larger network size if the sources are less demanding. Note that the three solutions yield close airtimes, especially for larger networks.

The next experiments are to study the impact of A_{thresh} on the weighted accuracy and power consumption. Figure 8 shows the results of WAO for two networks of 52 and 56 video sources. The results show that there is a peak of the weighted accuracy metric that can be achieved with values of A_{thresh} that are smaller than 0.01. This means that when the dropping is very small, the optimal accuracy is achieved. For the power consumption metric, the results are as expected where increasing A_{thresh} increases

the sending rate of the source, which in turn increases the power consumption of the source. These results suggests that there is a trade-off between accuracy and power consumption when A_{thresh} value is smaller than 0.01 and the best value should be chosen properly.

5.2. Effectiveness of the Proposed Bandwidth Allocation Solution

Figures 9 and 10 compare the performance of various solutions (EDO, WAO, WDO) for A_{thresh} values of 0.001 and 0.005, respectively, when using the Georgia Tech dataset at 30% resolution. Figure 11 shows the same results when A_{thresh} is equal to 0.005 and for Georgia Tech at 50% resolution. Moreover, Figure 12 shows the results for the CMU/MIT dataset and A_{thresh} of 0.005. Furthermore, Figure 13 shows the same results for MPEG-4 CIF resolutions when $A_{thresh} = 0.005$.

The results show clearly that WAO achieves better performance than other solutions in all metrics. To be specific, WAO increases weighted accuracy and power consumption metrics by almost 10%. The power consumption reduction comes from the reduction of the video sources application sending rates. These results also show that using weights with distortion optimization does not improve the network performance in terms of weighted accuracy, and in some cases, it even worsen the weighted accuracy. These results means that distortion based optimization is not suitable for surveillance systems with video sources that have different importance factors.

5.3. Effectiveness of the Proposed Bandwidth Pruning Mechanism

Figures 14-16 show the effectiveness of the bandwidth pruning mechanism when used with WAO for different image datasets and different A_{thresh} values. Furthermore, Figure 17 show the same results for MPEG-4 CIF resolution when $A_{thresh} = 0.005$. Not all combinations are shown because they all follow the same overall behavior. Four levels of pruning are analyzed: 95%, 90%, 80%, and 70%, with each level specifying the percentage of the original accuracy that will be achieved after the pruning mechanism is applied. The results show that the pruning mechanism significantly reduces the overall network load and power consumption with much less reduction in the weighted

detection accuracy. For example, with 95% pruning, we can save up to 45% in power consumption by scarifying only 5% in the accuracy.

Finally, let us discuss the effectiveness of the proposed WAO solution with the pruning mechanism, compared with the other solutions (EDO, WDO, and WAO without pruning). Figures 18 and 19 compare WAO with 95% pruning to the other solutions for A_{thresh} of 0.005 and two different datasets. (The results for other datasets and values of A_{thresh} exhibit similar behavior.) The results demonstrate that while the WAO solution with 95% pruning achieves almost the same values of weighted accuracy as EDO and WDO, it yields up to 45% saving in power consumption and network load.

6. Conclusions and Future Work

We have proposed an accuracy-based cross-layer video optimization framework for automated video surveillance systems that estimates and distributes the network bandwidth among the participating video sources to minimize the sum of the weighted detection accuracy in video streams. The proposed framework uses an enhanced effective airtime estimation algorithm utilizing a *Proportional Integral Differential* (PID) controller. Moreover, we have proposed a bandwidth pruning mechanism to achieve any desired tradeoff between detection accuracy and power consumption. Furthermore, we have developed an accurate model for characterizing the rate-accuracy relationship. The framework have been evaluated by streaming real video frames in simulated wireless network using OPNET.

The following points summarizes the main results.

- By comparing the distortion-based optimization and the proposed accuracy based framework, we find that the proposed framework significantly improves the detection accuracy as well as the power consumption which happen because of sending and dropping much less video data.
- The proposed PID based effective airtime estimation algorithm is accurate and converges faster than the existing online estimation algorithms.
- With a 95% pruning level, the proposed bandwidth allocation solution achieves almost the same accuracy as the distortion-based optimization but reduces the power

consumption by up to 45%.

In future work, we will experiment with a variety of computer vision algorithms and networks.

References

[1] Y. Shatnawi, M. Alsmirat, M. Al-Ayyoub, M. Aldwairi, The impact of the number of eigen-faces on the face recognition accuracy using different distance measures, in: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), 2018, pp. 1–5. doi:10.1109/AICCSA.2018.8612837.

[2] M. A.-A. Yousef Shatnawi, Mohammad Alsmirat, Face recognition using eigen-faces and extensionneural network, in: 16th ACS/IEEE International Conference on Computer Systems and Applications, 2019.

[3] A. oza, L. Mihaylova, D. Bull, N. Canagarajah, Structural similarity-based object tracking in multimodality surveillance videos, Mach. Vision Appl. 20 (2) (2009) 71–83. doi:10.1007/s00138-007-0107-x.
URL http://dx.doi.org/10.1007/s00138-007-0107-x

[4] Y. Jia, C. Zhang, Front-view vehicle detection by markov chain monte carlo method, Pattern Recognition 42 (3) (2009) 313–321.

[5] L. Bourdev, S. Maji, J. Malik, Describing people: Poselet-based attribute classification, in: International Conference on Computer Vision (ICCV), 2011.
URL http://www.eecs.berkeley.edu/~lbourdev/poselets

[6] M. Al-Zinati, T. Almasri, M. Alsmirat, Y. Jararweh, Enabling multiple health security threats detection using mobile edge computing, Simulation Modelling Practice and Theory (2019) 101957doi:https://doi.org/10.1016/j.simpat.2019.101957.
URL http://www.sciencedirect.com/science/article/pii/S1569190X19300905

[7] M. A. Y. J. Mohammad Al-Zinati, Taha Almasri, A mobile-edge computing bio-surveillance framework for multiple biological threat detection, in: 2019 The Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019, pp. 104–109.

[8] J. C. Niebles, C.-W. Chen, , L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: Proceedings of the 12th European Conference of Computer Vision (ECCV), Crete, Greece, 2010.

[9] L. Fei-Fei, L.-J. Li, What, Where and Who? Telling the Story of an Image by Activity Classification, Scene Recognition and Object Categorization, Studies in Computational Intelligence- Computer Vision (2010) 157–171.

[10] M. A. Alsmirat, Y. Jararweh, I. Obaidat, B. B. Gupta, Internet of surveillance: a cloud supported large-scale wireless surveillance system, The Journal of Supercomputing 73 (3) (2017) 973–992.

[11] P. Korshunov, W. T. Ooi, Critical video quality for distributed automated video surveillance, in: Proceedings of the 13th annual ACM international conference on Multimedia, 2005, pp. 151–160.

[12] M. A. Alsmirat, I. Obaidat, Y. Jararweh, M. Al-Saleh, A security framework for cloud-based video surveillance system, Multimedia Tools and Applications 76 (21) (2017) 22787–22802.

[13] M. A. Alsmirat, N. J. Sarhan, Cross-layer optimization and effective airtime estimation for wireless video streaming, in: Computer Communications and Networks (ICCCN), 2012 21st International Conference on, IEEE, 2012, pp. 1–7.

[14] M. A. Alsmirat, N. J. Sarhan, Cross-layer optimization for many-to-one wireless video streaming systems, Multimedia Tools and Applications (2018) 1–23.

[15] M. Alsmirat, N. J. Sarhan, Cross-layer optimization for automated video surveillance, in: 2016 IEEE International Symposium on Multimedia (ISM), IEEE, 2016, pp. 243–246.

[16] W.-C. Feng, E. Kaiser, W. C. Feng, M. L. Baillif, Panoptes: scalable low-power video sensor networking technologies, ACM Trans. Multimedia Comput. Commun. Appl. 1 (2) (2005) 151–167.

[17] Y. Jararweh, M. Alsmirat, M. Al-Ayyoub, E. Benkhelifa, A. Darabseh, B. Gupta, A. Doulat, Software-defined system support for enabling ubiquitous mobile edge computing (2017).

[18] M. A. Alsmirat, F. Al-Alem, M. Al-Ayyoub, Y. Jararweh, B. Gupta, Impact of digital fingerprint image quality on the fingerprint recognition accuracy, Multimedia Tools and Applications (2018) 1–40.

[19] P. Korshunov, W. T. Ooi, Video quality for face detection, recognition, and tracking, ACM Trans. Multimedia Comput. Commun. Appl. 7 (3) (2011) 14:1–14:21. doi:10.1145/2000486.2000488.
URL http://doi.acm.org/10.1145/2000486.2000488

[20] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, 2001, pp. 511–518.

[21] CMU/MIT image set, http://vasc.ri.cmu.edu/idb/html/face/frontal_images.

[22] J. wei Huang, Z. Li, M. Chiang, A. K. Katsaggelos, Pricing-based rate control and joint packet scheduling for multi-user wireless uplink video streaming, in: Proc. 15th International Packet Video Workshop (PV2006), 2006.

[23] C.-H. Hsu, M. Hefeeda, A framework for cross-layer optimization of video streaming in wireless networks (2011).

[24] M. A. Alsmirat, Y. Jararweh, I. Obaidat, B. B. Gupta, Automated wireless video surveillance: an evaluation framework, Journal of Real-Time Image Processing 13 (3) (2017) 527–546.

[25] M. I. Al-Saleh, M. A. Alsmirat, Y. Jararweh, I. Obaidat, A unified key distribution and session management protocol for mobile video surveillance systems, in: 2018

Fifth International Conference on Internet of Things: Systems, Management and Security, IEEE, 2018, pp. 234–238.

[26] S. Shirani, F. Kossentini, S. Kallel, R. Ward, Reconstruction of jpeg coded images in lossy packet networks.

[27] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications (2007) C1–1184doi:10.1109/IEEESTD.2007.373646.

[28] N. Shankar, M. van der Schaar, Performance analysis of video transmission over ieee 802.11a/e wlans, IEEE Transactions on Vehicular Technology 56 (4) (2007) 2346 –2362.

[29] C.-T. Chou, S. N. Shankar, K. G. Shin, Achieving per-stream qos with distributed airtime allocation and admission control in ieee 802.11e wireless lans, in: INFOCOM, 2005, pp. 1584–1595.

[30] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, S. Harp, Urban surveillance systems: From the laboratory to the commercial world, IEEE Proceedings 89 (10) (2001) 1478–1497.

[31] V. Nair, J. Clark, Automated visual surveillance using hidden markov models, in: Proceedings of the 15th International Conference on Vision Interface (VI), 2002, pp. 88–92.

[32] X. Yuan, Z. Sun, Y. Varol, G. Bebis, A distributed visual surveillance system, in: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), 2003, p. 199.

[33] O. Javed, Z. Rasheed, O. Alatas, M. Shah, KNIGHT: A real-time surveillance system for multiple overlapping and non-overlapping cameras, in: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2003, pp. 649–652.

[34] R. Rangaswami, Z. Dimitrijevi, K. Kakligian, E. Chang, Y. Wang, The SfinX video surveillance system, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2004.

[35] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, A system for video surveillance and monitoring, Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (May 2000).

[36] Y. Ge, J. C. Hou, S. Choi, An analytic study of tuning systems parameters in ieee 802.11e enhanced distributed channel access, Computer Networks 51 (8) (2007) 1955–1980.

[37] J. Hui, M. Devetsikiotis, A unified model for the performance analysis of IEEE 802.11e EDCA, Communications, IEEE Transactions on 53 (9) (2005) 1498–1510.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1510952

[38] L. Xiong, G. Mao, Saturated throughput analysis of ieee 802.11e edca, Computer Networks 51 (11) (2007) 3047–3068.

[39] Z. Yang, K. Nahrstedt, A bandwidth management framework for wireless camera array, in: NOSSDAV, 2005, pp. 147–152.

[40] S. H. Shah, K. Chen, K. Nahrstedt, Dynamic bandwidth management in single-hop ad hoc wireless networks, MONET 10 (1-2) (2005) 199–217.

[41] S. H. Shah, K. Chen, K. Nahrstedt, I. Introduction, Available bandwidth estimation in ieee 802.11-based wireless networks (2003).

[42] C. Sarr, C. Chaudet, G. Chelius, I. G. Lassous, Bandwidth estimation for ieee 802.11-based ad hoc networks, IEEE Trans. Mob. Comput. 7 (10) (2008) 1228–1241.

[43] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools.

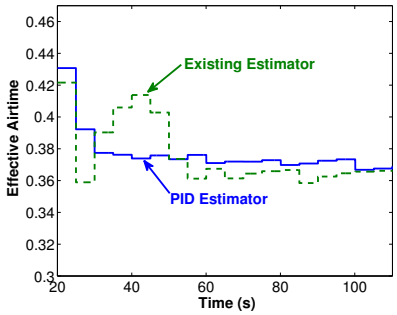
[44] Georgia tech face database, http://www.anefian.com/research/face_reco.htm.

[45] M. Grgic, K. Delac, S. Grgic, Scface - surveillance cameras face database, Multimedia Tools Appl. 51 (3) (2011) 863–879.

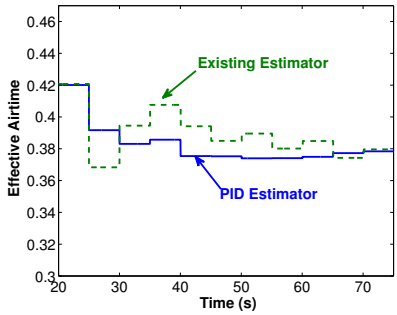
[46] A. Ortega, K. Ramchandran, Rate-distortion methods for image and video compression, IEEE Signal Processing Magazine 15 (6) (1998) 23–50.
URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=733495

[47] Y. Sharrab, N. J. Sarhan, Accuracy and power consumption tradeoffs in video rate adaptation for computer vision applications, in: Proc. of the IEEE Int'l Conf. on Multimedia & Expo (ICME 2012), Melbourne, Australia, 2012, pp. 410 – 415.

[48] Y. Sharrab, N. J. Sarhan, Aggregate power consumption modeling of live video streaming systems, in: ACM Multimedia Systems (MMSys 2013), Oslo, Norway, 2013.



(a) Georgia Tech at 30% Resolution, 32 sources, $A_{thresh} = 0.005$



(b) Georgia Tech at 50% Resolution, 32 sources, $A_{thresh} = 0.005$

Figure 6: Effectiveness of Proposed PID-Based Estimation

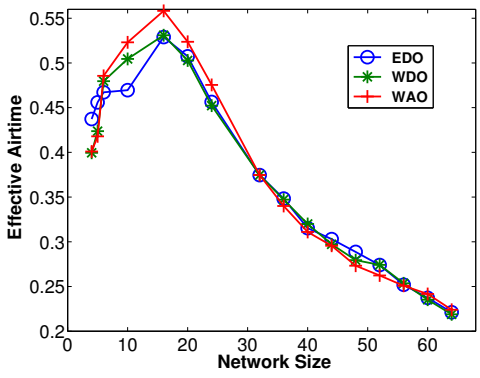
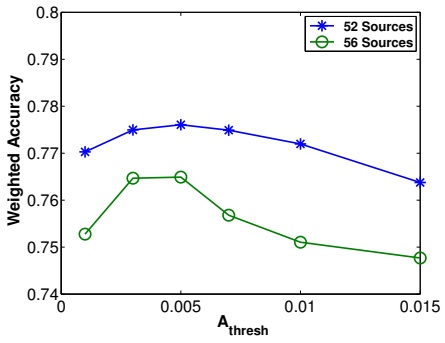
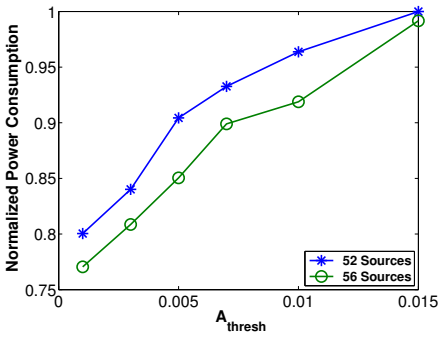


Figure 7: Comparing Various Bandwidth Allocation Solutions in Effective Airtime [Georgia Tech at 30% Resolution, $A_{thresh} = 0.005$]

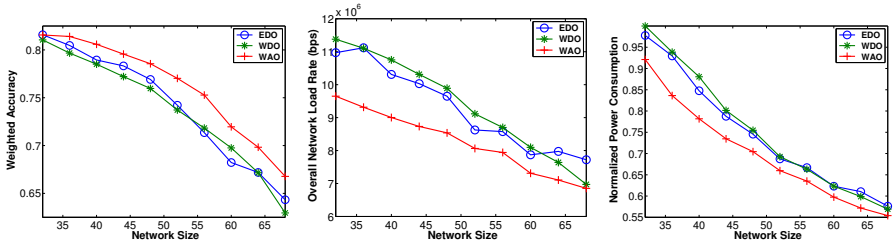


(a) Weighted Accuracy



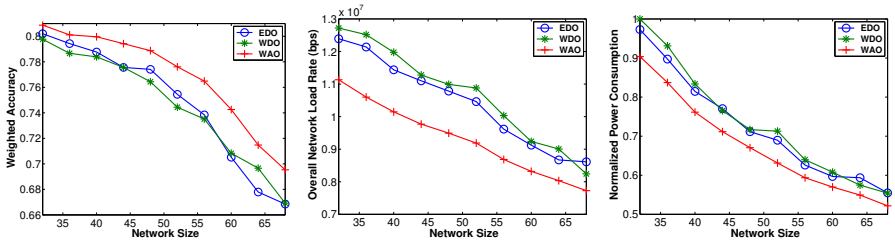
(b) Power Consumption

Figure 8: Impact of A_{thresh} with Proposed Weighted Accuracy Optimization [Georgia Tech at 30% Resolution]



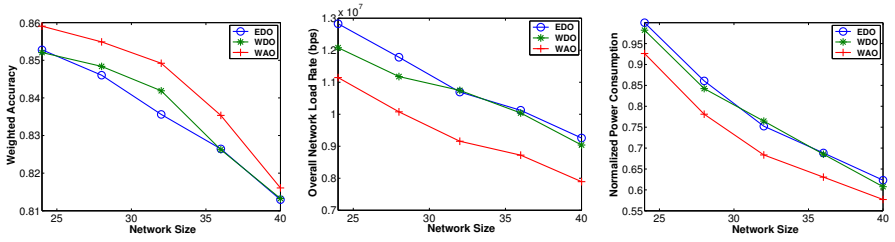
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 9: Comparing the Effectiveness of Various Bandwidth Allocation Solutions [Georgia Tech at 30% Resolution, $A_{thresh} = 0.001$]



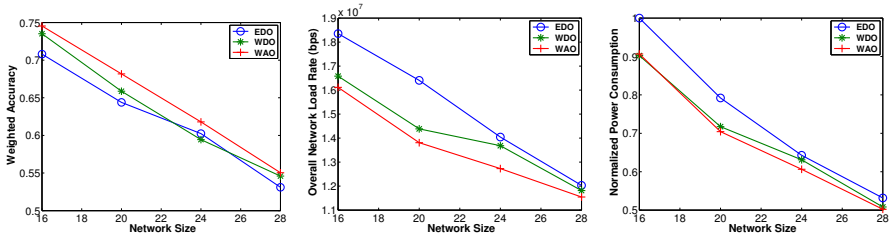
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 10: Comparing the Effectiveness of Various Bandwidth Allocation Solutions [Georgia Tech at 30% Resolution, $A_{thresh} = 0.005$]



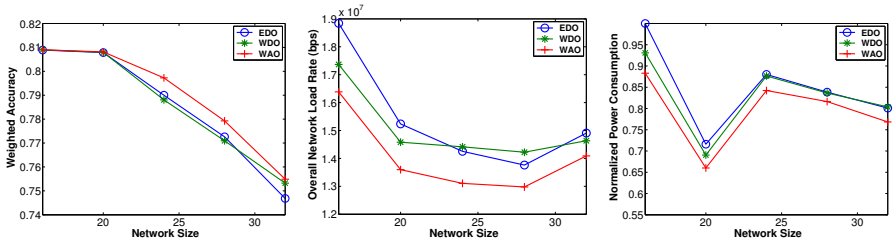
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 11: Comparing the Effectiveness of Various Bandwidth Allocation Solutions [Georgia Tech at 50% Resolution $A_{thresh} = 0.001$]



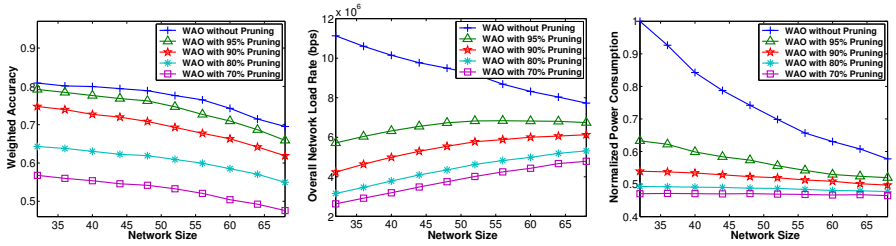
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 12: Comparing the Effectiveness of Various Bandwidth Allocation Solutions [CMU/MIT, $A_{thresh} = 0.005$]



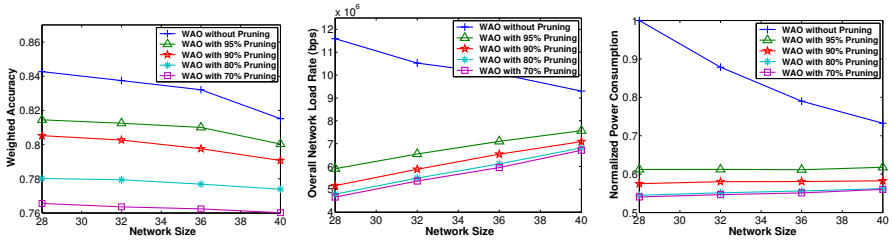
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 13: Comparing the Effectiveness of Various Bandwidth Allocation Solutions [MPEG-4 CIF Resolution, $A_{thresh} = 0.005$]



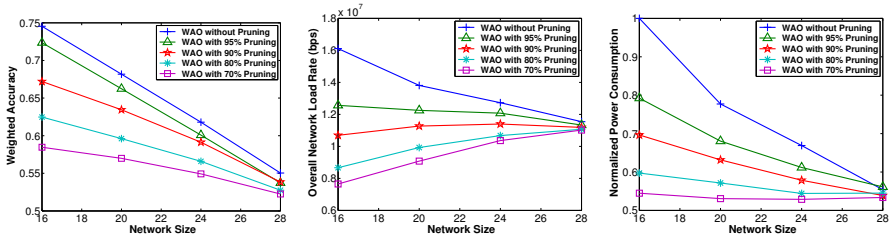
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 14: Effectiveness of Bandwidth Pruning Mechanism [WAO Solution, Georgia Tech at 30% Resolution, $A_{thresh} = 0.005$]



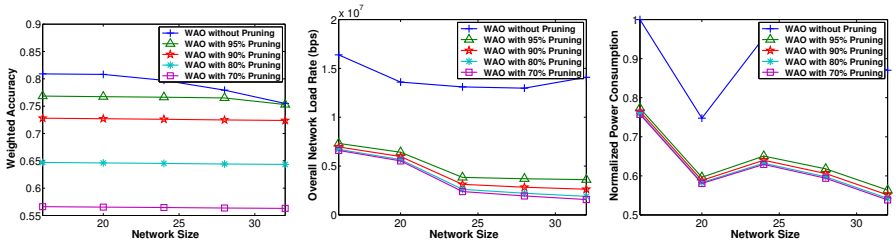
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 15: Effectiveness of Bandwidth Pruning Mechanism [WAO Solution, Georgia Tech at 50% Resolution, $A_{thresh} = 0.005$]



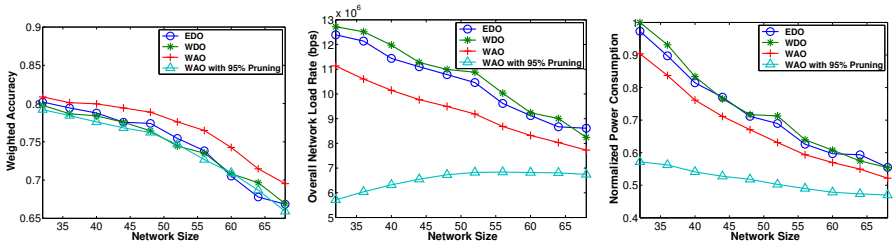
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 16: Effectiveness of Bandwidth Pruning Mechanism [WAO Solution, CMU/MIT, $A_{thresh} = 0.005$]



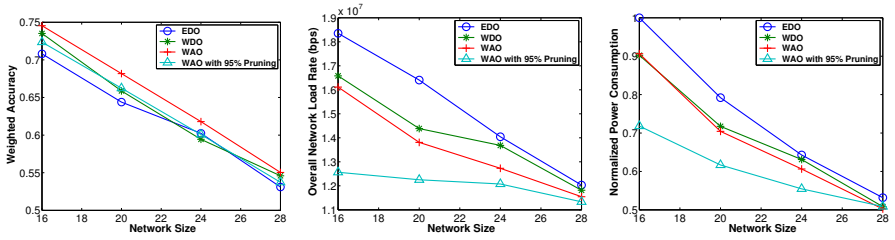
(a) Weighted Accuracy (b) Overall Network Load (c) Power Consum ption

Figure 17: Effectiveness of Bandwidth Pruning Mechanism [WAO Solution, MPEG-4 CIF Resolution, $A_{thresh} = 0.005$]



(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 18: Effectiveness of the Proposed WAO Solution with the Pruning Mechanism [Georgia Tech at 30% Resolution, $A_{thresh} = 0.005$]



(a) Weighted Accuracy (b) Overall Network Load (c) Power Consumption

Figure 19: Effectiveness of the Proposed WAO Solution with the Pruning Mechanism [CMU/MIT, $A_{thresh} = 0.005$]