Chapter 1

VIDEOWEB

Optimizing a Wireless Camera Network for Real-time Surveillance

Hoang Thanh Nguyen
Center for Research in Intelligent Systems
University of California, Riverside
nthoang@cs.ucr.edu

Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside
bhanu@cris.ucr.edu

Abstract Wireless camera networks provide a unique opportunity for collaborative surveillance. Performance evaluation and optimization of camera networks, however, has seldom been addressed. This chapter fills this gap by detailing an approach by which individual cameras and a whole network of cameras can be simultaneously optimized in terms of Pareto efficiency using multi-objective optimization of performance metrics. Experiments are performed on a set of 37 wireless cameras from a testbed built from the ground up at the University of California at Riverside.

Keywords: wireless camera networks, multi-objective optimization

1. Introduction

We describe the development and optimization of a new laboratory called VideoWeb to facilitate research in processing and understanding video in a wireless environment. While research into large-scale sensor networks has been carried out for various applications, the idea of massive video sensor networks consisting of cameras connected over a wireless network is largely new and relatively unexplored. The VideoWeb laboratory entails constructing a robust network architecture for a large number of components, including cameras, wireless routers and bridges, and video processing servers. Hardware and equipment selection needs to take into account a number of factors, including durability, performance, and cost. In addition, VideoWeb requires a number of software applications including those for data recording, video analysis, camera control, event recognition, anomaly detection, and an integrated user interface.

Challenges for the design of VideoWeb include creating a wireless network robust enough to simultaneously support dozens of high-bandwidth video cameras at their peak performance, providing power and connectivity to cameras, building a server farm capable of processing all the streaming data in real-time, implementing a low-latency control structure for camera and server control, and designing algorithms capable of real-time processing of video data.

This chapter is organized as follows: In Section 2 we cover related work and contributions. Section 3 discusses the requirements and specifications used in designing the system and discusses the technical challenges and solutions for actual implementation. Section 4 describes the VideoWeb testbed. Section 5 delves into characterizing the performance metrics from which to evaluate the system. Section 6 concludes with closing comments and lessons.

2. Related Work and Contributions

While many camera network platforms have been proposed [1–3], including systems with calibrated [4] or customized camera hardware nodes [5–8], there has been little discussion on how to one should go about in configuring or evaluating the performance of the network that they've built. This chapter makes the following contributions:

1 We detail useful guidelines in designing a large-scale wireless camera network with regard to selecting the hardware, software, and architecture

2 We perform Pareto optimization of 37 individual cameras in the network with respect to performance metrics such as frame rate, resolution, compression, and lag time

3 We perform simultaneous optimization for a set of 12 outdoor wireless cameras in order to gain insight into the performance trade-offs when using a large network under real-world conditions

3. Building the Camera Network

3.1 Choosing the Type of Network

There are many types of camera networks (e.g., wired vs. wireless, multi-hop wireless, distributed vs. central processing), but the most important factor in deciding what kind of network to build is determining the primary application. For instance, if a network's primary concern is surveillance (where reliability may be paramount, e.g., there may be a legally or contract-mandated uptime), a hard-wired network may be the only way to satisfy said requirements. A wireless network, on the other hand, provides more freedom and allows cameras to go where hard-wired cameras cannot (restricted only by power source).

3.2 Choosing the Right Camera

Choosing the wrong camera can be a costly mistake when building a large video network. When selecting a camera, a number of factors should be taken into consideration. Besides cost, these may include:

- **Wired vs. Wireless cameras.** Deciding between a wired or wireless camera is often a trade off between whether or not speed and reliability can be sacrificed in order to gain flexibility and freedom in placement. Cameras which will connect to the processing location (whether it be a central or distributed server) with dedicated wire connections (e.g., Ethernet, audio/video cables) excel in providing improved speed and reliability. This comes at the cost of restricting installation locations to those which can be reached via physical cables and installation may prove to be very labor-intensive, expensive, or simply unfeasible. Wireless cameras on the other hand allow greater freedom in placement as well as offering the opportunity of mobility (in the case of non-stationary cameras, e.g., robots, field sensors), but may sacrifice speed, reliability, and/or security.

- **IP vs. Analog CCTV.** Digital vs. analog in the context of video cameras is often an issue of convenience. Traditional analog closed-circuit TV (CCTV) systems are often simpler and more cost-efficient, but search and retrieval of data is cumbersome and any applications beyond surveillance and monitoring may be awkward or require dedicated and difficult to customize processing systems for each application. IP systems, on the other hand, can be more costly and/or complex, but output digital streams easily processed on computers and can even be accessed anywhere in the world simply by putting them on an Internet-accessible connection. If the video streams will be subject to constant or routine processing, analysis, or retrieval, IP cameras offer greater convenience and all the benefits of cheap digital storage, but may require additional network and software training for those only familiar with traditional CCTV systems.

- **Single-hop vs. Multi-hop wireless.** If wireless cameras are to be used, there are two primary ways they can reach their processing/storage destination: via a single-hop connection (cameras connect directly to wireless router/receivers) or via multi-hop connections (cameras connect to other cameras and pass on data before reaching the router/receiver). Multi-hop networks impose additional complexity and hardware as well as increased latency, but gain flexibility and wireless coverage by essentially turning every camera into a repeater node; these are more-suited for cameras with on-board processing capabilities. Single-hop networks are recommended if it is viable (i.e., network routers can be installed in locations in which all cameras can reach) for purposes of lower latency and reduced hardware requirements.

- **External vs. On-camera processing.** Whether or not to perform processing on-camera or deferring processing to external computers/systems is impacted by camera capability/programmability and network latency and bandwidth. For instance, a multi-hop network may be too slow to permit active tracking if video needs to first be passed through several sensors before reaching a processor, whose control commands then need to be relayed across several more sensors before the camera ever receives the command to “pan left”. On-camera processing can also reduce bandwidth consumption of the network (e.g, transmitting only areas of interest as opposed to full-frame video), while external processing allows a greater range of control and processing power.

- **Pan/Tilt/Zoom (PTZ) vs. Static cameras.** As the name implies, PTZ cameras offer active panning, tilting, and/or zooming capabilities whereas static cameras retain a permanent fixed field of view and orientation. PTZ cameras have the advantage of being able to cover larger areas (as a whole) and can zoom in or out to obtain better views of a scene as appropriate. This comes at the cost of increased complexity by requiring (manual or automated) control in order to take advantage of this capability. Static cameras on the other hand, are often less expensive and provide consistent scene coverage, but may require more installations to cover the same area as PTZ cameras and may do so with compromised quality (camera placement is often a balance between sacrificing area coverage for close-up detail).

- **Pan/Tilt/Zoom speed and magnification.** If PTZ cameras are to be used, the responsiveness of such commands should be taken into consideration when choosing between models, as some cameras may respond or move too slowly to be useful for applications such as active tracking. Since these specifications are often omitted by camera manufacturers, it is strongly recommended to trial cameras and testing if their PTZ speed is adequate before purchasing. In addition, the level of *optical* zoom may be important depending on the detail required for specific applications and the camera's physical distance from the scene. For most applications, digital zoom is worthless (at the raw capture stage) and should only be done in data processing.

- **Progressive vs. Interlaced cameras.** All other things equal, progressive cameras should be chosen over interlaced cameras where possible. While interlaced cameras can usually perform on-camera de-interlacing to avoid the combing artifacts inherent to interlaced video, such techniques tend to wash out fine detail for static objects and result in ghosting effects on moving objects ones (the alternative, processing only every other line in the video, also effectively halves the vertical resolution). There may be some exceptions to choosing a progressive camera, such as when a CMOS-sensor progressive camera has a rolling shutter which is so slow that its video exhibits noticeable skew on moving objects (also known as the "jello effect" as often seen in handheld cameras when the camera is panned too quickly), but even this may be preferred over the combing or ghosting artifacts from interlaced video.

- **Bandwidth: video format, resolution, and frame rate.** Resolution and frame rate go hand in hand as they will (in addition to video format) directly affect the bandwidth required for transmitting and storage required for archiving. Typical video cameras offer VGA resolution (640×480) at 30 frames per second, but newer high-definition (e.g., 720p or 1080p) cameras are becoming more readily available. While 640×480 resolution may be usable for many computer vision processing applications, those interested in face recognition (or better yet, face reconstruction) may find VGA to be particularly challenging to work with. Networks with particularly demanding requirements may want to consider specialty cameras, e.g., super high-resolution cameras, hardware-stitched 360° cameras, or even high-speed cameras, though these tend to demand a premium. The output format of the camera will also affect image quality; in addition to the traditional and easy-to-decode Motion JPEG codec (essentially a large series of JPEG images concatenated together), many cameras also offer MPEG-4 output for reduced bandwidth and/or higher quality using the same bandwidth via interframe compression. Decoding the video for custom-built applications may be more difficult with MPEG-4 however, and video artifacts caused by stream corruption (e.g., network congestion, dropped packets) may appear less appealing.

3.3 Choosing and Configuring the Network Hardware

The network hardware has a single purpose: to connect the cameras to the processing location(s) and to be as transparent as possible. Factors to consider when selecting network hardware include:

- **For Wired networking.** If IP cameras are being used, it is recommended to install the highest-rated network cable available (Cat-6 ethernet cable as of this writing) which can still reach its destination (generally 100 meters for gigabit ethernet or 55 meters for 10-gigabit ethernet using Cat-6a). The cost difference may be marginal (over Cat-5/5e, for instance) while providing overhead in robustness in the event that newer higher-bandwidth cameras are installed to replace aging cameras. Ethernet extenders may be required if cable lengths exceed cable specifications.

- **For Wireless networking: 802.11g vs. 802.11n vs. RF.** If wireless IP cameras are used, it will likely be a choice between 802.11g and the newer 802.11n. If the choice is available (e.g., wireless bridges are being used to turn an ethernet camera into a wireless camera), 802.11n from our experience is a *major* upgrade from 802.11g for both increasing network throughput and signal strength. How much of an improvement may be influenced by congestion in the operating frequency range due to other wireless networks in the area. Determining a selection between analog RF transmitters, on the other hand, can be more difficult as the performance will vary more widely based on the power, frequency, and data being transmitted, as well as the environment. It is

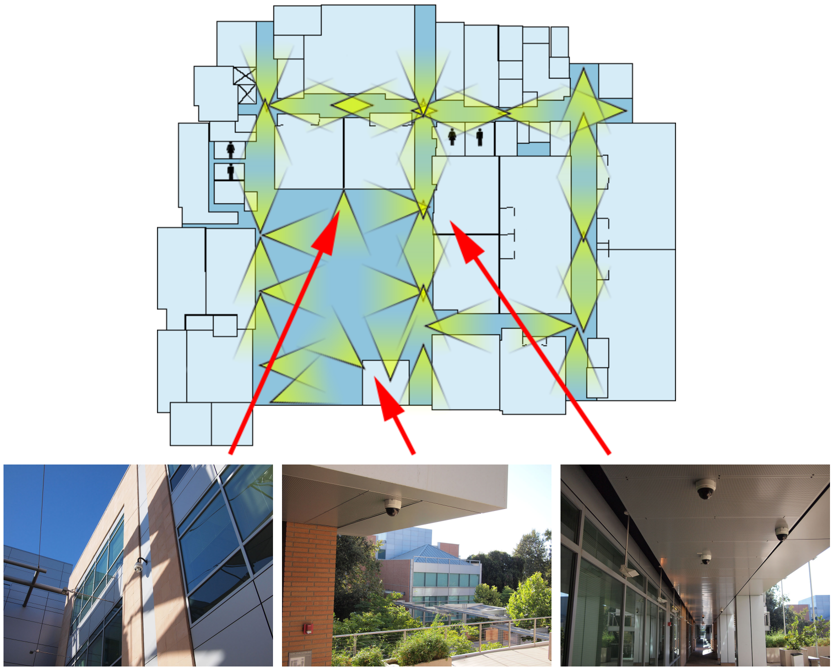

Figure 1.1. 37 camera locations were selected for complete outdoor coverage of the 14,300 square foot second floor of Engineering Building Unit II at the University of California, Riverside. Locations were manually selected and evaluated to ensure that usable fields of view were available for every square inch of the building from at least two viewpoints.

recommended to get a sample transmitter and to test each location cameras will be installed; this goes the same for wireless IP cameras, though wireless repeaters can be more-easily installed to extend ranges. In addition, selected wireless routers should offer (at minimum) gigabit capabilities, especially if a large number of cameras are expected to connect to it.

4. The VideoWeb Wireless Camera Network

The VideoWeb testbed is a network of 80 (37 outdoor and 43 indoor) wireless cameras. Goals in building the network included:

- the maximum coverage of the building's exterior floor (see Figure 1.1)

- the capability to perform real-time surveillance from a central or distributed server through a web interface

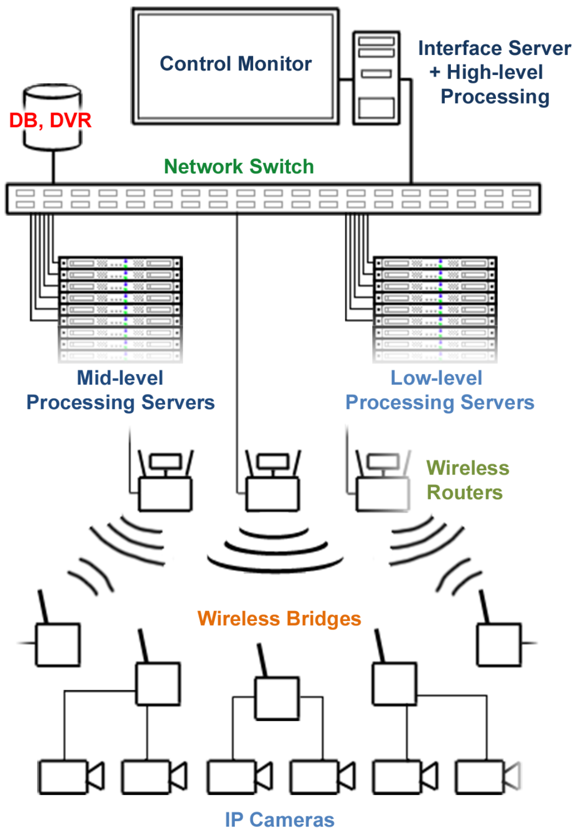

Figure 1.2. Overall architecture of the VideoWeb network. Top down: a single interface is used for direct control of any server/camera and high-level processing (e.g., user-defined face recognition). The server connects to a switch which hosts a database and joins two sets of servers: a series of mid-level (e.g., feature extraction) and low-level processors (e.g., detecting moving objects). The switch connects to routers which communicate with wireless bridges connected to the IP cameras.

- the capability for active tracking of subjects through the network
- the capability to control and perform arbitrary processing on any subset of camera feeds
- technological longevity of the system and robustness to outdoor weather

With these objectives in mind, the following design decisions were made:

- wireless connectivity for all the cameras for flexible placement and reduced installation costs

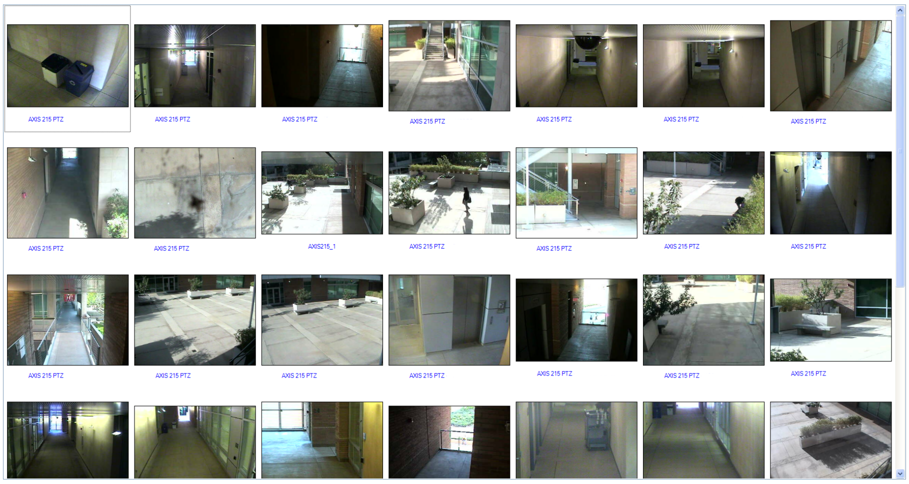

Figure 1.3. Sample video streams from the VideoWeb network.

- pan/tilt/zoom IP cameras are selected to allow active tracking as well as permit easier integration software-wise (control and streaming is all handled via simple HTTP commands)

- wireless bridges are used to provide wireless connectivity to the cameras for upgradability (e.g., advancements in wireless protocols) and are configured into a single-hop network to reduce network latency (the central server approach does not require on-camera processing)

- tiered processing architecture for simplifying the delegation of control and processing responsibilities across servers

- a 32-server processing rack with expandability from 128 cores to 256 (using dual socket motherboards)

The completed architecture of the network is seen in Figure 1.2 and sample video streams are shown in Figure 1.3. In-depth details about the network can be found in [9].

5. Experiments for Performance Characterization and Optimization of the Video Network

5.1 Optimizing Camera Configuration

Depending on the task or application, there are numerous "optimal" ways to configure a network. For instance, maximizing video resolution and quality may be paramount for biometrics, particularly in face recognition where a large number of pixels on the face is beneficial to identifying features. Surveillance and alarm systems, on the other hand, may find reliability more important. For instance, it may be more important that every moment is recorded with minimal skipping (not only for evidence in the event of an incident, but also because security applications often employ vision-based motion detection). Object tracking in turn, may benefit most by sacrificing resolution in exchange for a high sustained frame rate.

Configuring the network may consist of changing camera parameters (e.g., resolution, compression) as well as physical network parameters (e.g., number of cameras per bridge, number of bridges per router, number of routers per square foot). The later is helpful in introducing a metric for minimizing labor and monetary cost. We define 5 metrics for measuring camera network performance, the first two of which are used as configuration parameters.

1 *Resolution* (in pixels) - This measures the size of each video frame in pixels (the higher, the better). This parameter consists of 4 levels on the cameras (704×480, 704×240, 352×240, and 176×120).

2 *Video compression* - This parameter represents the amount of *lossy* video compression applied to the video by the camera. For M-JPEG streams on the cameras, this represents JPEG compression and ranges from 0 to 100 (the lower, the better). In our experiments, we test 5 of these levels (0, 20, 30, 60, and 100).

3 *Average frame rate* (in frames per second) - This measures the number of *complete* frames received per second, averaged over the duration of a measurement trial (the higher, the better). The frame rate may range from 0 to a maximum frame rate of 30 on the cameras.

4 *Standard deviation of frame rate* - This measures the consistency of the video. For instance, there may be two video streams both 20 frames per second each, but the first may output a constant

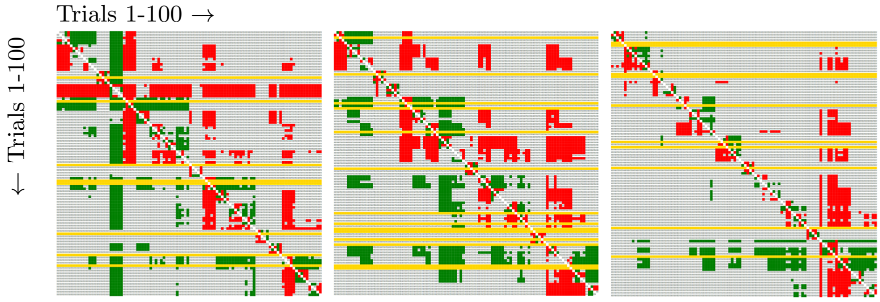

Figure 1.4. Measurement comparison matrices for 3 individual cameras. While cameras may exhibit variable performance even when using the same configurations, some configurations may be inherently better than others and exhibit similar performance across the network. To discover these configurations, 100 trials are performed on each camera under a variety of parameter configurations (i.e., resolution and compression) and each recorded measurement is compared for Pareto efficiency against the other 99 trials. This results in a symmetric matrix where vertical and horizontal axes indicate the measurements M_i and M_j, respectively (i.e., the top-leftmost square in each matrix indicates the relationship of M_1 against M_{100}). Red indicates that a particular M_i is inferior to a particular M_j, green indicates superiority, and a solid horizontal yellow line denotes rows which are completely Pareto-efficient (i.e., either superior or non-inferior against all other 99 trials).

20 frames per second while the second video may be sporadic and go from 30 to 0 to 10, back to 30 and so forth (but still average to 20 in the end). This metric is useful in evaluating the stability of the video (the lower the deviation, the better) and is measured by recording the delay between every two frames (in seconds with millisecond resolution) and calculating the standard deviation.

5 *Longest lag time between two complete frames* (in milliseconds) - This metric records the longest amount of time taken between any two consecutive frames (the lower, the better). This is insightful for evaluating a video stream's reliability (that is, it measures the longest amount of time a camera is "blind"). In addition to a depressed frame rate, this may be attributed to dropped/partial frames by the camera or data corruption/dropped packets undergone during transit.

5.2 Multi-objective Optimization Using Pareto Efficiency

We use the concept of Pareto efficiency to define which configuration of parameters is "better" than another. While this does not always tell a user which configuration should be used for a particular application, it serves to reduce the large number of possible configurations by showing which of those are usually “inferior”; a user only has to consider a configuration from the (potentially) much smaller Pareto set rather than every possible combination.

Inferiority and Non-Inferiority. Let M_1 be a vector of measurements of certain metrics for a camera and let M_2 be another trial of measurements on the same camera, but under a different parameter configuration. M_1 is said to be **inferior** to M_2 if and only if:

- every measurement in M_2 is equal to or outperforms the corresponding measurement in M_1
- one or more measurements in M_2 outperform the corresponding measurements in M_1

“Outperforms” is metric-specific and means “greater than” or “less than” depending on how the metric is defined (e.g., a *higher* frame rate outperforms a *lower* frame rate and a *lower* lag outperforms a *longer* lag). M_2 is said to be superior to or *dominates* M_1 if M_1 is inferior to M_2. Finally, M_1 and M_2 are both said to be **non-inferior** if neither is superior nor inferior to one another.

In order for a measurement M_i to be **Pareto-efficient** (amongst a set), it must be non-inferior to every other measurement in that set. That is, it possesses at least one *advantage* over every other measurement when compared one-on-one (e.g., M_1 has higher frame rate against M_2, lower lag against M_3, ..., higher resolution than M_n). The Pareto set is the set of all Pareto-efficient measurements and ideally, allows a user to discard a large percentage of inferior parameter configurations from consideration when setting the cameras.

Data Collection. Data collection consists of varying the resolution and compression parameters and recording measurements from the cameras during streaming. Two tests are performed: for individual optimization and simultaneous optimization. For individual camera optimization, each of the 37 cameras is streamed individually [9]. For simultaneous optimization of the network, a set of 12 outdoor cameras located in a courtyard are simultaneously configured and streamed. This allows us to receive insight into the strengths and limitations of the cameras individually as well as from the network as a whole. In total, we iterate through 4 resolutions (704×480, 704×240, 352×240, and 176×120) and 5 levels of compression (0, 20, 30, 60, and 100) each. Five measurement trials are captured for each of the cameras per configuration (100 trials

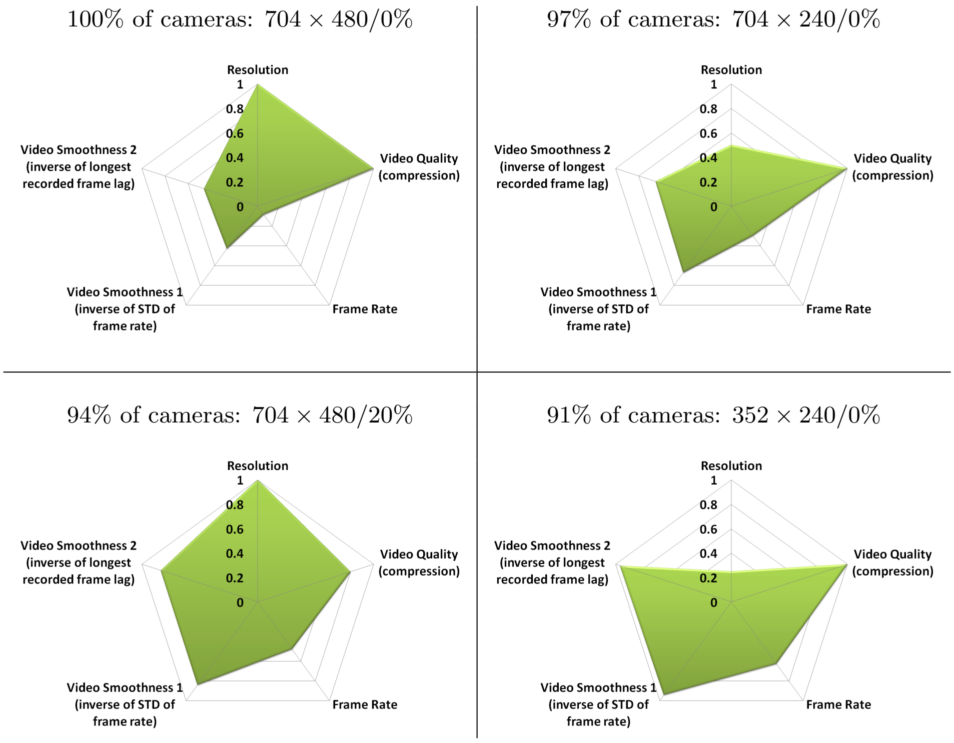

Figure 1.5. Top 4 dominating camera configurations as chosen by the 37 cameras. Graphs are ordered by the percentage of cameras in which they were Pareto-efficient.

total per camera). Each trial consists of streaming from the cameras for 3 minutes.

Camera footage is tested at 5 various points in the day across all cameras. This exposes the data to a variety of video footage ranging from bright open areas with upwards of 20 moving people in the scene, to dark and grainy footage of cameras monitoring lonely halls.

After data collection is completed, each camera is optimized individually to minimize camera, bridge, or router bias. For the simultaneous optimization, the average performance of all 12 cameras as a whole is used for optimization. This is done in $O(n^2)$ via exhaustive search (where n is the number of trials to compare), comparing each measurement to every other measurement on the same camera. With 20 configurations and 5 trials per configuration, each camera produces a symmetric $100{\times}100$ matrix. The resolution/compression pairs which result in the Pareto-efficient measurements for each camera are later aggregated against the entire network.

Compression Resolution \	100	60	30	20	0
176×120	46%	**66%**	46%	**51%**	**74%**
352×240	34%	46%	26%	34%	**91%**
704×240	**51%**	29%	17%	**54%**	**97%**
704×480	34%	31%	**63%**	**94%**	**100%**

Figure 1.6. Pareto efficiency of configurations when cameras stream independently.

Compression Resolution \	100	60	30	20	0
176×120	20%	0%	0%	20%	**60%**
352×240	20%	**60%**	20%	40%	40%
704×240	20%	40%	40%	**80%**	40%
704×480	20%	**60%**	**60%**	**60%**	20%

Figure 1.7. Pareto efficiency of configurations when all cameras stream simultaneously.

5.3 Evaluation Results

After over 100 hours of data collection at varying times of day across two weeks, the Pareto sets for all 37 individual cameras and 12 simultaneous are calculated (see Figure 1.4 for sample matrices). Considering only configurations in the Pareto sets eliminates (on average) approximately half of the tested configurations as inferior and redundant.

After aggregating the resolution/compression parameters of the Pareto sets for the entire camera network, we found that, surprisingly, *every* configuration tested was in the Pareto set for at least one camera. This suggests that there is no global network-wide consensus that any camera configuration is inferior to any other; every (tested) setting was Pareto efficient for at least some camera. Calculating the percentages of the Pareto set memberships, however, reveals that the cameras tend to exhibit a “preference” for certain configurations over others (see Figures 1.5 and 1.6). This is in line with the previous observation that roughly half of the tested configurations are not preferred (less than a majority agreement between the cameras).

The simultaneous optimization test, however, reveals that bandwidth and network limitations play a larger role in overall performance and that configurations with high Pareto efficiency percentages in individual testing (such as 704×480 and 0 compression) achieve Pareto efficiency in only 20% of the trials when this setting is applied for all cameras (Figure 1.7). Simultaneous optimization also shows us better compromises when a large number of cameras stream saturate the network (e.g., 704×240 and 20 compression).

It is not surprising to see higher percentages on configurations with either the maximum resolution or minimal compression since they already optimize at least one metric by definition. However, configurations such as $176\times120/60\%$ and $704\times240/20\%$ reveal local optimum which is potentially very useful for some practical applications of the video network. Using a more fine-tuned set of compression levels, we would likely be able to find more such points, aiding in the creation of a useful set of presets for specialized applications.

The presented multi-objective approach can also be used to optimize network parameters for specific applications. This can be done by quantifying application performance (e.g., face detection rate, face recognition rate, smoothness of tracked objects trajectories) and adding them to the multi-objective metrics.

6. Conclusions

We have designed an software-reconfigurable architecture for a wireless network of a large number of video cameras and implemented a working system by building the servers, installing the cameras, writing the software, and configuring the network to support it. Further, we gained insight into configuring the network's cameras by defining a set of metrics and discovering Pareto-efficient camera configurations by performing multi-objective optimization on a large volume of real data recorded by the system.

The idea persists that if one has a network of cameras rated at 30 frames/second (FPS), one will be able to obtain the said 30 frames/second regardless of network configuration or parameters. Though this may be true in controlled test environments, the performance expectation should not be so optimistic for real-world wireless implementations. Even using the most preferred Pareto-efficient configurations on a non-congested network, it is shown that frame rates will most certainly suffer and that trade-offs must be made.

During a large workshop hosted in the building with a large number of wireless Internet users, however, it was observed that frame rates of the cameras would periodically drop and we later found that these drops coincided with breaks given during the workshop. Suspicious that a number of open and local 802.11g networks may be congesting our network, a cluster of bridges were upgraded from 802.11g to 802.11n. In daily usage, frame rates were seen to reach up to 20 FPS for even the most bandwidth-intensive configurations (such as 704×480 resolution with 0% compression) where they were previously achieving typically only 3 FPS (even when other bridges in the network were not in use). While this makes a case for upgrading to 802.11n, this also suggests that network congestion from other networks may play a large role in frame rates and that networks may wish to operate in a dedicated frequency range.

In situations when even hardware upgrades can still not achieve sufficient performance, however, we would like to emphasize that partial data is still important. Rather than having algorithms which assume that the data consists entirely of complete video frames (and are only capable of processing such frames), real-time computer vision algorithms should take advantage of as much information as is available to them; the constant stream of partial frames which may only be missing the last few rows of data can still be tremendously useful for a number of applications.

Acknowledgments

This work was supported in part by NSF grants 0622176, 0551741, 0905671, and ONR grant N00014-07-1-0931.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[2] W.-T. Chen, P.-Y. Chen, W.-S. Lee, and C.-F. Huang, “Design and implementation of a real time video surveillance system with wireless sensor networks,” in *IEEE Vehicular Technology Conference (VTC Spring 2008)*, May 2008, pp. 218–222.

[3] H. Park, J. Burke, and M. B. Srivastava, “Design and implementation of a wireless sensor network for intelligent light control,” in *IPSN 2007*, 2007, pp. 370–379.

[4] M. Quinn, R. Mudumbai, T. Kuo, Z. Ni, C. D. Leo, and B. S. Manjunath, “Visnet: A distributed vision testbed,” in *ICDSC 2008*, Sept. 2008, pp. 364–371.

[5] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L.-C. Chang, J. Tygar, and S. Sastry, “Citric: A low-bandwidth wireless camera network platform,” in *ICDSC 2008*, Sept. 2008, pp. 1–10.

[6] C. Park and P. H. Chou, “eCAM: ultra compact, high data-rate wireless sensor node with a miniature camera,” in *SenSys ’06*, 2006, pp. 359–360.

[7] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides, “A lightweight camera sensor network operating on symbolic information,” in *First Workshop on Distributed Smart Cameras 2006*, Nov. 2006.

[8] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin, “Camera mote with a high-performance parallel processor for real-time frame-based video processing,” in *PAVSS 2007*, Sept. 2007, pp. 69–74.

[9] H. T. Nguyen, B. Bhanu, A. Patel, and R. Diaz, “VideoWeb: Design of a wireless camera network for real-time monitoring of activities,” in *ICDSC 2009*, Aug. 2009.