7-11-2006

Knowledge-Based Video Compression for Robots and Sensor Networks

Chris Williams Williams
University of South Florida

Knowledge-Based Video Compression for Robots and Sensor Networks

by

Chris Williams

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Robin Murphy, Ph.D.
Miguel Labrador, Ph.D.
Phillip DuMas

Date of Approval:
July 11, 2006

Keywords: teleoperation, frame rate, bandwidth regulation, human factors, user contention

Table of Contents

List of Tables

List of Figures

Knowledge-Based Video Compression for Robots and Sensor Networks

Chris Williams

ABSTRACT

Robot and sensor networks are needed for safety, security, and rescue applications such as port security and reconnaissance during a disaster. These applications rely on real-time transmission of images, which generally saturate the available wireless network infrastructure. Knowledge-based Compression is a strategy for reducing the video frame transmission rate between robots or sensors and remote operators. Because images may need to be archived as evidence and/or distributed to multiple applications with different post processing needs, lossy compression schemes, such as MPEG, H.26x, etc., are not acceptable. This work proposes a lossless video server system consisting of three classes of filters (redundancy, task, and priority) which use different levels of knowledge (local sensed environment, human factors associated with a local task, and relative global priority of a task) at the application layer of the network. It demonstrates the redundancy and task filters for realistic robot search scenarios. The redundancy filter is shown to reduce the overall transmission bandwidth by 24.07% to 33.42%, and when combined with the task filter, reduces overall transmission bandwidth by 59.08% to 67.83%. By itself, the task filter has the capability to reduce transmission bandwidth by 32.95% to 33.78%. While Knowledge-based Compression generally does not reach the same levels of reduction as MPEG, there are instances where the system outperforms MPEG encoding.

Chapter One

Introduction

Knowledge-based Compression offers a strategy for regulating video frame rate transmission between field robots and remote operators at the application layer. It is an alternative to conventional methods of video transmission such as MPEG which may be either unavailable or unsuited to the specific task at hand. Because a large majority of field operations perform image processing on the receiving end of the transmission, also known as post processing, a lossy compression scheme, such as MPEG, H.26x, etc., is not acceptable. Knowledge-based Compression allows for the sending of complete image frames without first having to encode them, while still restricting the amount of information transmitted over the channel.

Knowledge-based Compression consists of three filters for regulating bandwidth usage; the filters are the redundancy filter (F_r), task filter (F_t), and priority filter (F_p). The layout of the system can be seen in Fig. 1.

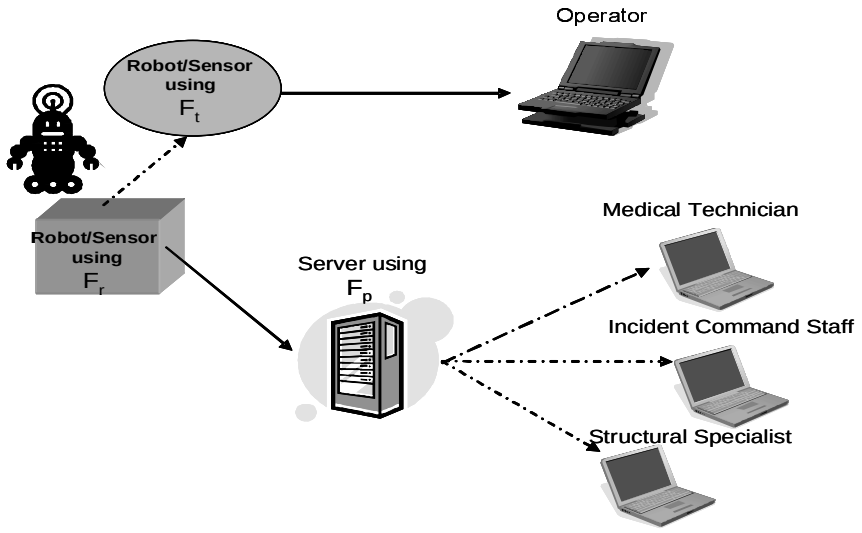

Figure 1. Example of a System Implementing Knowledge-based Compression.

1.1 Motivation

Safety, security, and rescue applications such as port security and reconnaissance after a disaster often presume that remote devices will provide real-time imagery to multiple remote users over wireless networks. However, experience shows that real-time imagery from even a few sensors or robots will saturate a wireless network (e.g., ShadowBowl). Therefore, mechanisms are needed to reduce video transmissions. However, the safety, security and rescue domain poses additional constraints which render lossy compression algorithms such as MPEG or H.26x unacceptable. The constraints are as follows.

1. Each user application may require its own post-processing of the imagery. Post-processing computer vision algorithms often cannot be performed on images which have undergone lossy compression.
2. The imagery may need to be used for forensic assessment or evidence at a later date. Therefore, the complete video stream must be stored and any video compression must be reversible. Off-board storage of video from a robot or sensor is highly desirable since the field device may be destroyed as the incident unfolds or may simply fail at an inopportune moment.

This suggests that complete video or video which has been reduced with a reversible algorithm should be periodically check pointed to a secure storage site.

1.2 Research Question

Given the motivation above, the following research question emerges:
How can multi-agent video transmission be reduced while not affecting post processing and meeting the requirements of the consumers?

Therefore addressing the above question requires knowledge about the human factors involved in regulating and modifying video sent to safety, security, and rescue *agents*. As well as a thorough understanding of constraints present in the existing system used to manage the robots and sensors of the operation. Mainly, the proposed strategy

must be able to successfully reduce bandwidth consumption while not hampering the agent's ability to effectively perform their specific task in the operation. The strategy must also ensure that data integrity is preserved in some form so that other agents have access to uncompromised data either during or after the operation.

Because of the demands imposed, rather then following the typical structure of bandwidth management systems which may reside on the data link and/or network layers, the proposed strategy, dubbed Knowledge-based Compression, was decided to be implemented on the application layer. This allows the system to utilize real time data in a dynamic environment to change compression accordingly. A look at the OSI model can be seen in Fig. 2 on the following page.

Once a system implementing Knowledge-based Compression is in place the following claims can be made:

1. With the system as a whole, consisting of Information, Processing, and Server agents, service is reversible if the following constraints are met: The redundancy filter is applied before the task filter for the Immediate-Processing agent, or the redundancy filter is applied before the priority filter for the Post-Processing agent.

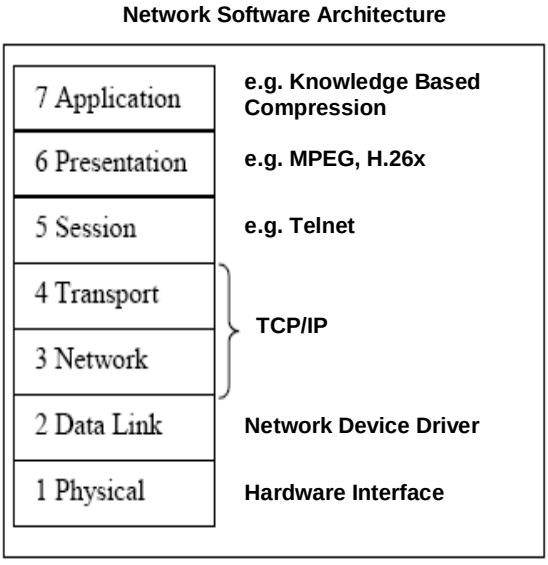

Figure 2. Open System Interconnection (OSI) Model.

2. The redundancy filter will reduce bandwidth consumption while maintaining complete information of the images and be completely reversible.
3. Each of the three filters will reduce bandwidth consumption.
4. The system as a whole will maintain *complete* information, while supporting multiple agents. Where the information is said to be complete when the result obtained from an algorithm, filter, or some other form of processing performed on the information is identical to the result obtained when the same processing is performed on the original information.

This thesis considers only robot sensors but the results should be applicable to sensor networks.

1.3 Contribution

Knowledge-based Compression offers a unique strategy to confront the issue of video bandwidth consumption when presented with constraints which prevent the use of typical methods. The implications of the strategy are evident to many different areas which include scientists, end-users, and companies. Knowledge-based Compression combines research from previous studies and expands on those ideas to create a fully functional toolset which can be applied to a system to reduce the amount of bandwidth needed and manage users. There's been very little work up until now on a system which dynamically regulates video compression based on context, human factors, and server congestion. Instead those factors were investigated and used separately. The research performed for this thesis offers a consolidated report of the various fields of contribution and how they relate to safety, security, and rescue applications.

Aside from the academic contribution, Knowledge-based Compression offers a very real advantage to the current standard methods of video transmission in safety, security, and rescue operations. The results presented in chapter four of this thesis demonstrate that Knowledge-based Compression has the following effects on bandwidth consumption. First, that the F_r portion of Knowledge-based Compression has the

capability to reduce bandwidth consumption by at least 24.07% to 33.42%, and that the F_t portion of Knowledge-based Compression has the capability to reduce transmission bandwidth by at least 32.95% to 33.78% in a realistic search and rescue scenario. Lastly, that the combined F_r and F_t portions of the system have the capability to reduce bandwidth consumption by at least 59.08% to 67.83% in a realistic search and rescue scenario. Knowledge-based compression also caters to the individual user by modifying the video feed based on their needs and requirements.

1.4 Approach: Knowledge-based Compression

Knowledge-based Compression takes advantage of three main characteristics of video intensive multi-agent networks in order to regulate bandwidth through the use of its filters. These three characteristics are:

1. Not all frames transmitted over the network contain useful information.
2. The importance of various video characteristics changes depending on which mode of operation the robot is in.
3. Different users require different qualities of video feeds, and by taking advantage of knowing which user is currently viewing the feed, that feed's characteristics can be customized to fit the requirements of that user.

The first characteristic is due to the fact that if the robot is stationary and no context change is occurring then the robot is technically transmitting identical video frames with no added benefit to the viewer.

The second characteristic was based on research by Murphy *et al.* [1] established that there are two distinct modes in teleoperated robots in confined spaces which take up 51% and 49%, respectively, of the task duration: navigation and search. The search time often consists of very little movement or context change occurring in the robot's field of view; however the robot currently does not regulate transmission based on its operation. For instance if a robot is navigating, frame rate is more important than resolution since the operator is more concerned with accurately knowing where the robot is without a delay.

Whereas if the robot is searching an area, resolution is more important than frame rate since the area is unlikely to change much over time; however being able to accurately survey the area is important.

The third characteristic stated that different users require different qualities of video feeds, and by taking advantage of knowing which user is currently viewing the feed, that feed's characteristics can be customized to fit the requirements of that user. A general user may not need to view the video stream at full frame rate and quality, so depending on server congestion their stream can be modified to save bandwidth. A medical operator on the other hand, may need to view the video stream at full frame rate and quality since their situation is much more critical than that of a general user.

The three characteristics mentioned above are taken care of by the different filters of Knowledge-based Compression, which again were F_r, F_t, and F_p. The purpose of F_r is to reduce transmission overhead by restricting the amount of unneeded video segment frames sent over the network using a local context (e.g. if the robot is stationary and the environment around the robot isn't changing, then reduce the frame rate transmission until the situation changes). This filter, in theory, has the potential to reduce the overall transmission rate by as much as 96.6% in periods of no or little change. F_r restricts frames prior to them being processed by either F_t or F_p. F_t regulates the video transmission to robot operators depending on what mode of operation the robot is in. By monitoring the operators use and the robot, the filter determines which mode of operation the robot is in and regulates frame transmission accordingly. F_p restricts and regulates frame transmission based on user type and server load. Lower priority users may have their video stream quality restricted or even halted until server congestion diminishes.

Although various algorithms exist which reduce the overhead of video transmission, the nature of the video transmitted by the robots restricts the use of them. Because un-encoded transmission may be necessary if computer vision algorithms are applied downstream it restricts the use of conventional video transmission technologies such as MPEG, H.26x, etc. Another shortcoming of the conventional video transmission methods is that they are often too displaced from the actual method of use. That is to say, that they

lack the ability to adapt to the needs of the user. Such is the case with the needs of search and rescue operations; there is need for a system which is able to tie in priority policies and contention management (e.g. certain users may require more or less information about the robot or sensors environment).

1.5 Thesis Organization

The thesis is organized is organized as follows. Chapter two provides an overview of related work in human factors on robot teleoperation, commercial approaches to bandwidth regulation, and academic approaches to bandwidth regulation. The related work section also includes a video classification and motion detection systems section which includes references to useful resources used in the design of the individual filters. Chapter three describes Knowledge-based Compression and the three filters which make up Knowledge-based Compression: F_r, F_t, and F_p. Chapter four describes the experiments and results detailing Knowledge-based Compression's performance in simulated search and rescue scenarios, and archived search and rescue footage. Chapter five summarizes the findings related to the Knowledge-based Compression, their implications, and recommended future work.

Chapter Two

Related Literature

This chapter describes the related literature which contributed to the creation of Knowledge-based Compression. The chapter is broken up into four different sections on the various topics related to the development of Knowledge-based Compression. Section 2.1 describes work related to the human factors element of Knowledge-based compression. Section 2.2 covers some of the different commercial approaches to regulating bandwidth for systems with unique constraints and requirements. Similar to section 2.2, section 2.3 covers the academic side of regulating bandwidth for unique systems. Finally, section 2.4 covers methods related to video classification and motion detection.

2.1 Human Factors

Knowledge-based Compression is motivated in part by the current understanding of the effect of frame rate regulation and stream resolution on users' perception and performance. The human factors literature concludes that task performance and Quality of Perception are largely independent of frame rate and that resolution quality of the image is preferred over frame rate. The literature is discussed in the following subsections. Specifically, the effect of varying frame rates on group communications is examined in 2.1.1.1. In 2.1.2 the effect of frame rate on the quality of service versus the quality of perception is discussed. In 2.1.3 research is presented which states that resolution is more preferred by users than frame rate when watching video. Finally, 2.1.4 examines research which proposes that teleoperators have different requirements for frame rate and resolution based on the situation. Therefore, the human factors literature demonstrates

that frame rate, resolution, and color and be adapted to fit the requirements of different situations.

2.1.1 Group Communications

In a study on the effect of varying frame rate on group communication, Anderson, Jackson, McEwan, and Mullin [2] found that there was no effect on the outcome of the task in the presence of lower frame rates. The work examined the effect of varying frame rate, 5 Hz - 25 Hz, on group communication, which was an important indication of team performance. The teams were given a design task in which they had to collaborate with each other to create an advertisement poster from a set of shared images. The images were the same for each of the groups however the order in which the images were displayed was different. Although it was determined that frame rate did not change the overall outcome of the task, the experiment did show that speakers in the low frame rate condition used longer referring expressions with more content material. This trait was used both initially and on repeat mentions although only speakers in the two party were affected and not speakers in groups of four. The paper concluded that although low video quality had no effect on final outcome, it makes speakers more communicatively cautious. Given that Murphy *et al.* [3] concludes that two humans working cooperatively have nine times better task performance than a single teleoperator, the communication findings suggest that low video quality could have an unexpected impact on team performance.

2.1.2 Quality of Service vs. Quality of Perception

In [4], Guinea and Gulliver examined the effect of varying fps rates (5hz, 15hz, 25hz) on Video-Eye paths, user Quality of Perception (QoP), and the impact of clip type on User QoP. The frame rates were static on a test by test basis meaning that the video was prerecorded using 5, 15, or 25 fps and maintained constant throughout the length of that video clip. The researchers classified quality of perception as not only a user's satisfaction

with the quality of a multimedia presentation, but also his/her ability to analyze, synthesize, and assimilate informational content of multimedia. The researchers concluded that overall, the users had no difference in their ability to understand and assimilate what they were watching in the presence of differing frame rates. The results did show, however, that the users' overall subjective level of quality and enjoyment did drop with the frame rate for video sequences which had a great deal of motion and detail, such as a rugby sports clip.

2.1.3 Quantization vs. Frame Rate

In a study presented by McCarthy, Miras, and Sasse [5], a comparison was done on the impact of resolution quality versus frame rate on user preference in a purely subjective scenario, meaning that the user was not required to perform a task but simply observe a video stream and rate their QoP. The researchers found that users prefer high-resolution images to high frame rate, and concluded that the "high motion = high frame rate" rule does not apply to small screens. The experiments were run using different combinations of quantization, i.e. compression, and frame rate, with quantization levels varying from 2 to 24 and frame rates varying from 6 to 24. Tests were done to examine effects of fps vs. quantization on a palmtop device, however the tests were also done on CIF-sized (Common Interchange Format 352 x 288 pixels) video and the results held that resolution quality was preferred over frame rate. User surveys found that at the lowest frame rate of 6 fps, participants still found the quality acceptable 80% of the time. As the quantization level increased the user acceptance decreased with a sharp drop in user acceptance occurring when the quantization level was greater than 8.

2.1.4 Frame Rate vs. Resolution: Dependent on Context

In a Theory Paper proposed by Winfield [6], it is argued that the use of standard hardware and software components, including Wireless Local Area Network technology and

Internet Protocols, can bring considerable benefits to the design of teleoperated robots for applications in inspection or surveillance. Winfield proposed that the teleoperator has different demands on video depending on the context of the situation. The study simplifies the situation into two contexts: navigation and inspection. During navigation, Winfield suggests that there is a greater emphasis on frame rate so that obstacles can be seen and evasive actions taken in time to avoid collision. Resolution is given less priority since the operator must only identify that the obstacle is there not see the fine details of the obstacle. During inspection the opposite is true; the resolution should be high in order to give the clearest possible image of the area being inspected. Frame rate can be sacrificed since little to no motion is going on and therefore no evasive action is needed.

2.2 Commercial Approaches

As noted in the motivation for the design of Knowledge-based Compression, the standard methods used for video transmission sometimes don't meet the needs of an application and therefore a different approach must be found or developed. In this section both standard methods for video transmission as well as commercially adapted methods of video transmission are presented. In section 2.2.1 the standard methods of video transmission for both wired and wireless networks are presented. Sections 2.1.2, 2.1.3, and 2.1.4 present approaches to video transmission that are more situationally based than the standard methods of transmission.

2.2.1 MPEG and H.26x

The current standards in multimedia and web compression of video fall under the guidelines of MPEG for wired connections and H.26x for wireless transmission. With MPEG individual objects within a scene are tracked separately and compressed together to create an MPEG4 file. When motion occurs in the video, MPEG4's rate control algorithm adjusts the bit rate by lowering the quality of the compressed video in order to

maintain the bandwidth transmission rate for the more complex data being transmitted. MPEG compression works through the use of three different frame types which are classified as either INTRA (I) or INTER (P or B). INTRA (I) frames, have no motion compensation performed but are used as reference for other frames. INTER (P or B) frames both include motion compensation, where P frames use the previous I or P frames as reference for motion compensation and also are used as reference frames for other INTER frames. B frames use both the previous and successive I or P frames as references for motion compensation but B frames are not used as reference frames for INTER frames.

H.26x is a popular standard used for wireless applications due to its focus on low bit rate. The protocol uses the same types of frames as MPEG, however the protocol is different from some MPEG implementations, in which I frames are periodically used mainly for indexing, In the H.263 standard, I frames are seldomly used, just to refresh the visual quality of the video transmission. A quantization parameter was also added to the standard to control bit rate, if the quantization parameter is kept constant while encoding then the quality of the frames will remain relatively constant but bit rate will fluctuate with scene changes and frame type. The opposite holds true in that if the goal when encoding is to keep the bit rate constant than quality will usually suffer.

As mentioned previously, both of the standards reside on the presentation layer and therefore include an extra level of abstraction from the proposed system. Because of this they do not allow for easy online modification of the video transmission if the context in which the video is being sent changes (e.g., the user decides that they no longer need a continuous transmission of video and instead only require that relevant snapshots be sent). Also since both of the standards' use of encoding involves sending bits and pieces of the captured video's information when transmitting, they hinder the capability of running image processing algorithms on the receiving end of the transmission [7], [8], [9], [10], [11].

2.2.2 Snapshot View

In some instances full motion video is unneeded and so the recorded video is broken down into relevant frames before being transmitted to the destination. For instance, with corporate security, companies are paying large amounts of money to track video information but are throwing away a large portion of it. By reducing video tracking to just relevant portions, bandwidth could be saved. This method of keeping only the "relevant" frames is a form of data mining, or making better use of the video collected by reducing excess and improving organization, Friedrick [12]. Aside from the corporate security sector, the military makes use of intelligent data storage as well, Page [13]. Footage from UAV's, Unmanned Aerial Vehicles, was stored in video archives consisting of thousands upon thousands of hours of video footage. One approach to reducing and organizing the video archive was through processing the video into relevant frames which effectively relate the entire sequence of the video.

2.2.3 Editing on the Fly

In scenarios which require fast response time in a very mobile situation (e.g., storm trackers), video is captured and broken down into individual frames. The frames are then edited manually on the fly before being sent back to the station. The result is a group of still images which can be looped together to generate the impression of video at between 5 to10 fps. TV stations in Oklahoma, Texas, and other tornado-prone areas have relied on satellite phone and cell phone based video transmission equipment for more than 10 years to deliver the initial images of storms, though recently upgraded equipment has allowed for faster transmission and more editing capabilities. The equipment is essentially a computer with an interface that allows the user to edit and view the video frames captured on the camera. The computer connects the camera to either a cell phone, or a satellite phone, to transmit the pictures back to the station, Whitney [14].

2.2.4 Moving to Server as Transmission Point

Some experts have moved away from the trend of finding better compression algorithms or larger bandwidth pipelines and instead have focused on better network management of video transmission. Rather than receiving data directly from the transmission point, which are typically tight links, a more optimal method suggested is to have the transmission first sent to a server with a much wider pipeline. From there users could access the video with a lower risk of congesting the pipeline, as is the case with the typically narrow link capacity of the camera transmission point. A typical web camera acting as a server may allow up to 4 users to log into it without degradation, however the service will degrade as a larger amount of users attempt to connect. By moving the video to a company's central server it can more readily act as a storage facility and allows for ease of access and improved video quality, Friedrick [15].

2.3 Academic Approaches

With the increasing popularity of wireless devices coupled with current demands for bandwidth intensive multimedia applications it is of no surprise that bandwidth management has become a widely researched topic in the computer science community. While a large area of research is focused on lower level layers of internet architecture with a goal of improving the bandwidth management throughout the network, other studies have been conducted that focus on improving bandwidth with a specific application environment in mind. In section 2.3.1 a dynamic bandwidth management system for ad hoc wireless networks is discussed. In 2.3.2 a set of papers on research related to bandwidth management in cellular networks is presented.

2.3.1 Dynamic Bandwidth Management for Ad Hoc Wireless Networks

In a paper written by Chen, Nahrstedt, and Shah [16], their research proposed a Quality of Server (QoS) bandwidth management system for an ad hoc wireless network which can react independently or in conjunction with a QoS aware link level scheduler. The system maintains a cross layer interaction between the application and link layer and manages the amount of bandwidth an application needs from the application layer. The goal of the system is to provide fairness in the absence of distributed link level weighted fair scheduling (*DWFS*). In case weighted fair scheduling becomes available, the system assists it by supplying the scheduler with weights and adjusting them dynamically as network and traffic characteristics vary. To obtain these weights, the system converts the bandwidth requirement of the application into a channel time requirement. The Bandwidth Manager then allots each flow a share of the channel time depending on its requirement relative to the requirements of other flows in the network. In the wireless network 802.11 standard each node within another node's range must share a channel of communication, in order to prevent interference when transmitting the nodes only transmit during their specific allotted channel time.

2.3.2 Bandwidth Management Systems for Cellular Networks

Another area of recent research has been directed towards improving bandwidth management among cellular customers. The research attempts to improve bandwidth management through analyzation of the individual user's current position in relation to neighboring transmission cells. The user's position is then used for dynamically reallocating resources based on that position, as well as the users expected use of additional multimedia services. The systems proposed by Sungwook, Varshney [17]; Kim, Oliveira, and Suda [18]; and Horibe, Zhang [19] offer different approaches to handoff and bandwidth reservation between base stations to prevent losses. For example, one of the proposed systems employs user mobility information to reserve available channels for a

user from the adjacent base stations according to the user's current position Horibe, Zhang[19]. The degree of channel reservation is determined based on the power levels that the user receives from the adjacent base stations.

2.4 Video Classification and Motion Detection Systems

Since today's bandwidth speeds are able to support more and more streaming multimedia applications there is a growing need for technology which maximizes the potential of these applications. Of particular interest are video surveillance applications and the image processing techniques used in conjunction with them. One area of work in video surveillance technology involves unsupervised motion detection and analyzation; and more specifically, trying to eliminate false positive/negative analysis results without diminishing the real time capabilities of the cameras due to increased overhead [20]. Another area of similar background involves the creation of classification algorithms which are able to divide an image into relevant segments based on the dominant image motion in the video [20]. However, classification algorithms tend to not be limited by the same constraints as video surveillance since the algorithms need not be run in real time.

In both classification and motion detection algorithms there normally exists a three step process: preprocessing, thresholding, and analysis or likelihood determination which is applied to the set of images [20], [22], [23], [24]. Preprocessing is the initial step and involves locating all of the areas of suspected change between two images. Thresholding is applied after preprocessing and attempts to clump together areas of change and eliminate as much noise or insignificant change as possible. The final step involves a quantitative analysis of the results of the first two steps and varies depending on the application. Determining the correct algorithm to use in a given application environment is an area of research in itself. The remaining subsections offer varying approaches to performing the three processes; specifically, section 2.4.1 deals with preprocessing and thresholding, and 2.4.2 goes over the likelihood detection process.

2.4.1 Preprocessing and Thresholding

For preprocessing, image differencing has become a common choice for motion detection [22], [23], [24], [25]. The main advantage to image differencing is that it is simplistic and therefore requires a small amount of computations and overhead. However due to image differencing's simplicity it is very sensitive to noise and illumination changes [26]. If the time between image comparison is fairly short in an application then the drawbacks to differencing can be overlooked since errors are likely to be less of an issue than if the image turn over were to take place during the period of every few months. If overhead is not an issue to an application or if there is little room for errors then there are other more robust algorithms available which take into consideration intensity and texture differences [26], [27], [28]. Thresholding, much like preprocessing, should be chosen based on the specific application in mind since that will determine the effectiveness of the thresholding method. The Poisson noise model and stable Euler number offer comparatively good quality results for the amount of calculations and overhead generated [24].

2.4.2 Likelihood Detection

As stated previously, likelihood determination can be achieved using a number of different methods. However, if a gray area of analysis exists where results can be inconclusive then fuzzy logic control can be a viable option for analysis [29], [30]. Fuzzy logic control operates through the use of a rule base which the designer uses to create fuzzy variables and sets of rules for the variables. Employing the correct number of fuzzy sets is a source of contention among different researchers; however depending on the amount of precision needed between 5 and 11 fuzzy sets for each fuzzy variable is recommended [30]. When taken into context with a motion detection algorithm, a fuzzy logic controller can be used to analyze the results of the first two steps of the algorithms processing. Once the controller has run the results through the rule set, it generates a decision as to whether or not the perceived motion detected constitutes a distinct change in the environment.

Chapter Three

Knowledge-Based Compression

This chapter describes the approach and implementation to employing Knowledge-based Compression. It's important to note that although this chapter does propose a whole strategy for setting up Knowledge-based Compression, for implementation it only covers F_r and F_t. This focus is due to the fact that F_r and F_t have more concrete specifications then $F_{p,}$ since F_p must be modeled to fit a specific scenario type and system. Section 3.1 provides a detailed structure overview of Knowledge-based Compression, giving a run down of the premise for which the system was created as well as the guarantees the system offers if used. The following sections are each dedicated to one of the three filters used by Knowledge-based Compression. Section 3.2 and its subsections describe the motivation, design, and implementation behind the redundancy filter. Section 3.3 and its subsections describe the motivation, design, and implementation behind the task filter. Section 3.4 and its subsections describe the motivation and design behind the priority filter. Section 3.5 and its subsections describe the overhead in implementing the strategy.

3.1 System Structure Overview

The Knowledge-based Compression approach assumes there are three types of agents (Information, Processing and Server) producing and/or consuming video data. Three different classes of filters (redundancy, task, priority) encapsulating relevant knowledge are used to reduce bandwidth consumption while ensuring that the needs of the agents are met and an archive is maintained.

The agents are as follows, and can be seen in Fig. 3 below. The Information agent (A_i) is typically either a robot or sensor which is recording some type of data for use in a

safety, security, and rescue operation. The Processing agent (A_p) uses data received from the Information agent and takes one of two forms: Immediate-Processing (A_{p_i}) and Post-Processing (A_{p_p}). The Immediate Processing class of the Processing agent performs tasks directly with the Information agent. The Post-Processing class of the Processing agent does not interact directly with the Information agent, but passively observes its progress and may run algorithms or filters on its own end of the communication (e.g., a medical specialist examining video footage sent back by the operator). The Server agent (A_s) archives data and video footage and acts as an intermediary point between a Post-Processing agent and the robot or sensor in order to ensure data integrity.

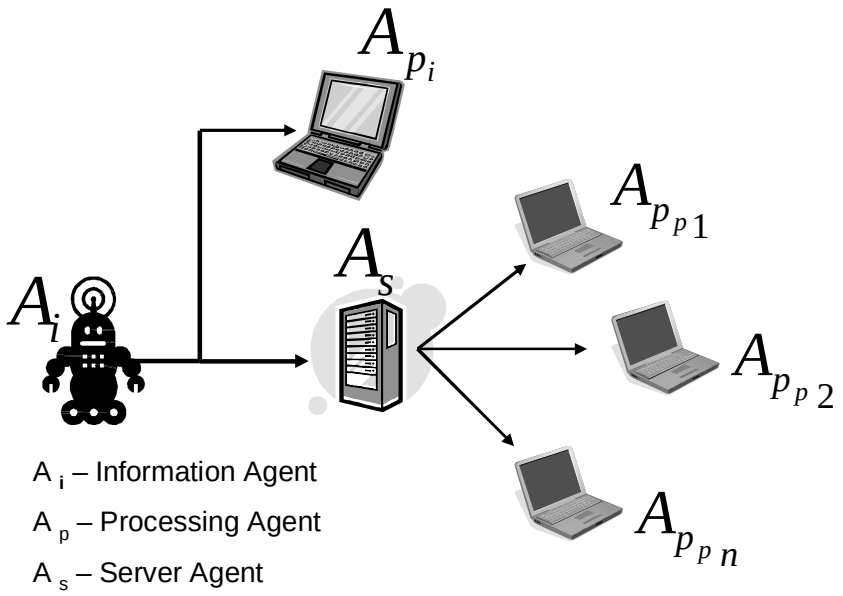

Figure 3. Agent Diagram.

There are three types of filters that we propose to address the issue of conserving bandwidth while ensuring that the operations on the video are either reversible or at a point where modification will no longer compromise future needs for the video. The filters are a redundancy filter (F_r), a task filter (F_t), and a priority filter (F_p).

The redundancy filter operates by removing redundant frames from the video transmission, specifically frames which satisfy a condition which is a function of a *threshold* and a *tolerance* for sequential image sequences. The redundancy filter does not

require any external input in order to function and can be completely contained on the robot or sensor's system.

The task filter, as its name suggests, filters data based on the specific task or tasks being performed. The task filter can be used with either implicit or explicit information. If implicit information is used then the task filter has previous knowledge about the nature of the tasks being performed and can characterize tasks based on information observed. If the task filter operates explicitly, then it will regulate data based on specific commands received denoting the task type.

The priority filter is composed of a set of filters whose conditions are satisfied by the function of a server-load and an agent type. Server-load refers to the amount of traffic the Server agent is currently experiencing based on the agents connected and the type of requests being sent. Agent-type is defined as a function of priority and the parameter set P, and P consists of values for frame rate, frame type, and resolution.

If the above system proposal is followed then the following claims can be made:

1. With the system as a whole, consisting of Information, Processing, and Server agents, service is reversible if the following constraints are met: The redundancy filter is applied before the task filter for the Immediate-Processing agent, or the redundancy filter is applied before the priority filter for the Post-Processing agent.
2. The redundancy filter will reduce bandwidth consumption while maintaining complete information of the images and be completely reversible.
3. Each of the three filters will reduce bandwidth consumption.
4. The system as a whole will maintain complete information, while supporting multiple agents. Where the information is said to be complete when the result obtained from an algorithm, filter, or some other form of processing performed on the information is identical to the result obtained when the same processing is performed on the original information.

An outline of the proposed system can be seen in Fig. 4 on the following page.

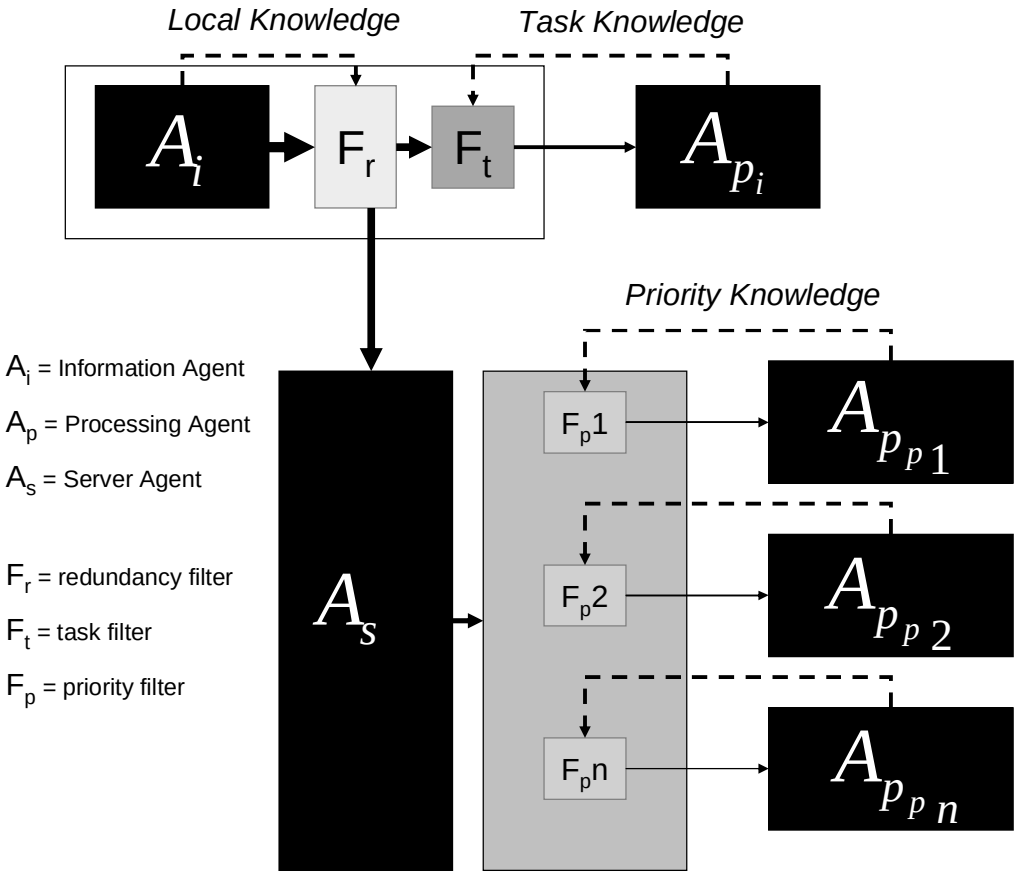

Figure 4. Knowledge-based Compression Overview.

3.2 Redundancy Filter

This section covers the approach and design involved in creating the Redundancy Filter (F_r). Specifically section 3.2.1 covers the underlying motivation for the creation of the redundancy filter, and 3.2.2 goes over the design and implementation of F_r.

3.2.1 Motivation

The underlying motivation behind the design of F_r was to conserve bandwidth while maintaining complete frame data for both archival and post-processing purposes. A common characteristic of video transmission is that not all frames transferred over the

network contain useful information. This characteristic is due to the fact that if the robot or sensor is stationary and no context change is occurring then they are technically transmitting identical video frames with no added benefit to the viewer. F_r was designed to address this issue by comparing consecutive frames to each other, and then removing redundant frames by reducing the rate at which frames are transmitted.

3.2.2 Design and Implementation

As mentioned previously, F_r operates by removing redundant frames from the video transmission, specifically frames which satisfy the function of F_r(*threshold, tolerance*) for the two images I_1 and I_2. Where F(*threshold, tolerance*) holds true when $I_1 \approx I_2$. F_r was designed to meet the following criteria:

1. F_r must be able to reduce bandwidth consumption while not affecting the systems' ability to maintain complete information.
2. Safety, security, and rescue operations are constantly changing. Therefore, the filter must be able to accurately categorize redundant frames in real time.
3. Due to the nature of safety, security, and rescue operations false positives are more acceptable than false negatives (i.e., it is better to have more frames that are redundant than to lose frames which are unique).

One possible implementation that allows F_r to satisfy the above criteria is discussed in detail below. It's possible that another method of identifying context change within the video stream could be used, however the implementation listed below has been fine tuned and tested in order to guarantee that both run time and accuracy are within the constraints listed above.

F_r identifies redundancy through the function of F(*threshold, tolerance*). The *threshold* and *tolerance* values used in the function can either be specified by system defaults or manually by the agent. The system defaults for threshold and tolerance are set to 10 pixels for threshold and 5% for tolerance. The values were determined through a series of empirical tests. The empirical tests tested a range of tolerance and threshold

values and employed false positives and negative analysis. A sample set of frames generated in the lab were passed into the filter, and the accuracy of the filter in detecting the different context changes was recorded. The test helped identify a set of values which are sensitive enough to identify context change but not overly sensitive so that they incorrectly categorize context change.

Other important values for F_r include median filter size, how often F_r is run, and the frame rate. The median filter is important to Fr because it is used to reduce the amount of noise present in the images to ensure that context change isn't incorrectly identified due to noise. The median filter size was originally tested using a 3x3 filter but was found to be less effective at reducing noise then the 5x5 filter, and larger filters were found to smooth too much.

In calculating how often F_r needs to be run in order to accurately identify context change there are a few considerations. First, the filter must be run often enough to categorize change, but not too often or it will overwhelm the system's resources. Another reason to limit how often F_r runs is that very little is expected to happen between 2 consecutive frames when transmitting at 30 frames per second since the time gap is relatively small. Therefore, a delay of between 300 and 500 msec between runs was determined to be an acceptable duration because it allows the filter to run often enough to detect change but not too often to where it overwhelms the system. The final implementation utilizes a run time of 500 msec and was found to be accurate in detecting context change while not overwhelming the systems resources.

Finally, the reduced frame rate was chosen to approximately match the run time of the filter, or 2 fps. Since the filter is set to run at least every 500ms, than having a new frame to compare with ensures that it doesn't take a full second before a change can be identified, i.e. if it was set to 1 frame per second then it's possible that the same frame could be tested against itself since the filter runs more then once per second. The rest of the algorithm is described in detail below accompanied by a block outline of the algorithm in Fig. 5.

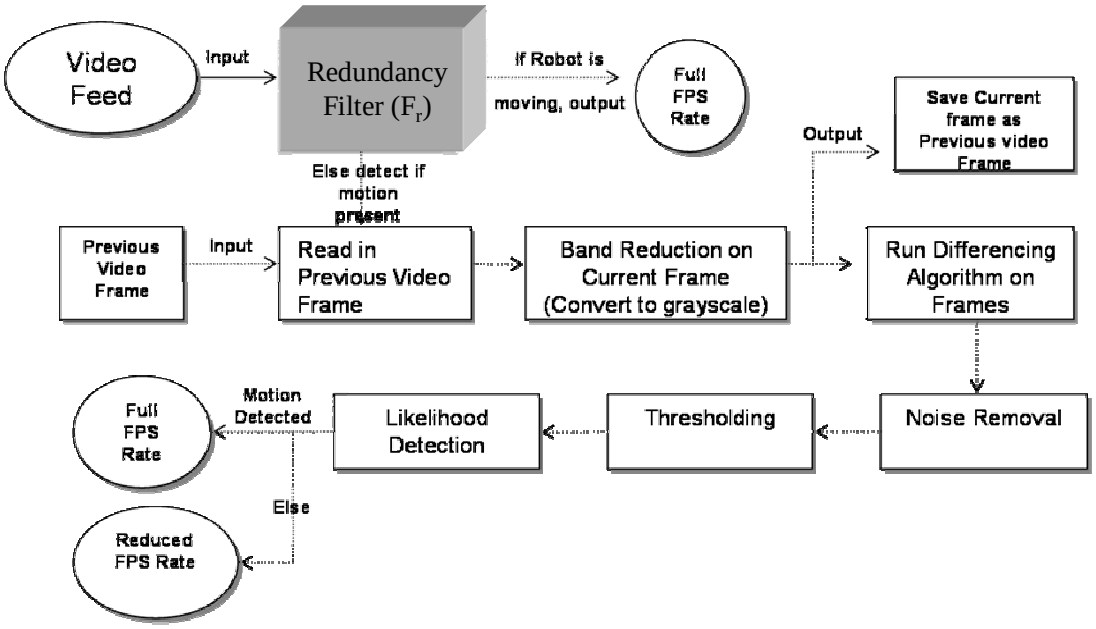

Figure 5. An Outline of the F_r Algorithm.

The function is composed of the following algorithm, with a block outline of the algorithm detailed in Fig. 5 above:

1. Image I_1 is passed into the function to check for redundancy.
2. A check is run to see if the robot or sensor is currently moving;
    a. If yes, the redundancy check is over and the function returns false for redundancy, or a frame rate of 30 (industry standard).
    b. If no, the algorithm continues.
3. I_1 is passed through a *band reduction* algorithm to convert the image from RGB to grayscale for later analysis. Where band reduction is the process by which the number of image bands are reduced. For instance, RGB possesses 3 bands whereas grayscale has 1 band. By reducing the number of bands to a single band it enables later processing such as image differencing and likelihood detection to be performed much easier.
4. A check is run to see if the algorithm has been run before by looking for the previously tested image I_2;

    a. If yes, the algorithm continues.
    b. If no, the algorithm saves I_1 as I_2 and returns false for redundancy, or a frame rate of 30.
5. I_1 and I_2 are run through a simple *differencing* algorithm to obtain the result R_1. Where differencing is the process in which image I_1 is subtracted from image I_2 on a 1 to 1 pixel basis. If the resulting pixel set is close to 0 then there is little change between the two images.
6. R_1 is run through a *median filter* using a 5x5 mask to reduce noise. A median filter is a non-linear digital filtering technique often used to remove noise from images or other signals. It involves creating a box, in this case 5 pixels by pixels, and replacing the center pixel with the median of its neighbor's values.
7. R_1 is then run through a *thresholding* process which tags all pixels that are within range of a predefined *threshold* to obtain the result R_2;
    a. If a pixel is above the *threshold* it is set to black, meaning a pixel value of 0.
    b. If a pixel is below the *threshold* it is set to white, meaning a pixel value of 255.
8. Likelihood detection is then performed on R_2 which makes its decision on redundancy based on a predefined *tolerance* value;
    a. If the percentage of white to black pixels is above the *tolerance* then the image is tagged as non-redundant.
    b. If the percentage of white to black pixels is below the *tolerance* then the image is tagged as redundant.
9. The algorithm examines the image tag returned by Likelihood detection and then returns the suggested frame rate to transmit at based on the result;

a. If redundant then the algorithm suggests a frame rate of 2 frames per second. A frame rate of 2 was chosen to approximately match the run time of the filter. Since the filter is set to run at least every 500ms then having a new frame to compare with ensures that it doesn't take a full second before a change can be identified.
b. If non-redundant then the algorithm suggests a frame rate of 30 frames per second.

3.3 Task Filter

This section covers the approach and design involved in creating the Task Filter (F_t). Specifically section 3.3.1 covers the underlying motivation for the creation of the Task Filter, and 3.3.2 goes over the design and implementation of F_t.

3.3.1 Motivation

The motivation for F_t is based on the safety, security, and rescue characteristic that the relative importance of frame rate, resolution, and image format changes depending on which mode of operation the robot is in. Specifically, Murphy *et al.* [1] established that there are two distinct modes in teleoperating robots in confined spaces which take up 51% and 49%, respectively, of the task duration: navigation and search. By taking advantage of the different formatting settings needed based on the operation F_t is able to conserve bandwidth while not affecting the ability to perform post-processing on the outgoing image feed.

3.3.2 Design and Implementation

As stated in the motivation, F_t was designed to regulate bandwidth based on two distinct tasks present in teleoperation: navigation and search. The search time often consists of

very little movement or context change occurring in the robots' field of view, whereas the complete opposite is typically present for navigating. The robot currently does not regulate transmission based on either operation. F_t takes advantage of these differences and tailors the formatting to suit the current operation. For instance if a robot is navigating, frame rate is more important than resolution since the operator is more concerned with accurately knowing where the robot is without a delay. Whereas if the robot is searching an area resolution is more important than frame rate since the area is unlikely to change much over time; however being able to accurately survey the area is important. Therefore, the task being regulated has the following criteria:

1. When the robot is in its search phase, increase the resolution to 100%, decrease the frame rate, and change the image format to color.
2. When the robot is navigating, perform the opposite, and decrease the resolution to either 65% or 20% depending on speed, increase the frame rate, and change the image format to grayscale.

In order to moderate the effect of image formatting changes on the operator when switching from navigating to searching an additional category was added. Rather then simply changing from searching to navigating, instead navigating was broken down into two different options, navigating fast and navigating slow. As such, three different formatting options are used, with each option corresponding to a specific mode. The resolution settings used for the different modes were chosen based on the amount of savings and the observed image quality. A test was run examining the resulting image size for each resolution setting in 5 percent increments for a sample image. The final resolution values chosen were 100%, 65%, and 20%.

The final implementation of F_t for Knowledge-based Compression is as follows. First, due to the fact that F_t resides after F_r in Knowledge-based Compression, F_t does not regulate frame rate any further then already specified by F_r. Since F_r removes redundancy in the image feed by restricting the frame rate during periods of no detected change it already satisfies the criteria set forth by the task. The remaining implementation is described by the algorithm below:

1. The image I_1 is passed into F_t.
2. F_t requests the current voltage of the robots motor controls;
    a. If the voltage is unchanged from the set minimum, then F_t classifies the operation as Searching.
    b. If the voltage is between the set minimum and *threshold*, then F_t classifies the operation as Navigating: Slow.
    c. If the voltage is greater than the set *threshold*, then F_t classifies the operation as Navigating: Fast.
3. F_t sets the resolution based on the operation tag;
    a. Searching - I_1 Resolution left at 100% of normal.
    b. Navigating: Slow – I_1 Resolution set to 65% of normal.
    c. Navigating: Fast – I_1 Resolution set to 20% of normal.
4. F_t sets the image format based on the operation tag;
    a. Searching – I_1 format left as RGB color.
    b. Navigating – I_1 format changed to grayscale.
5. F_t returns formatted image I_1.

An outline of F_t and its relation to F_r can be seen in Fig. 6 below.

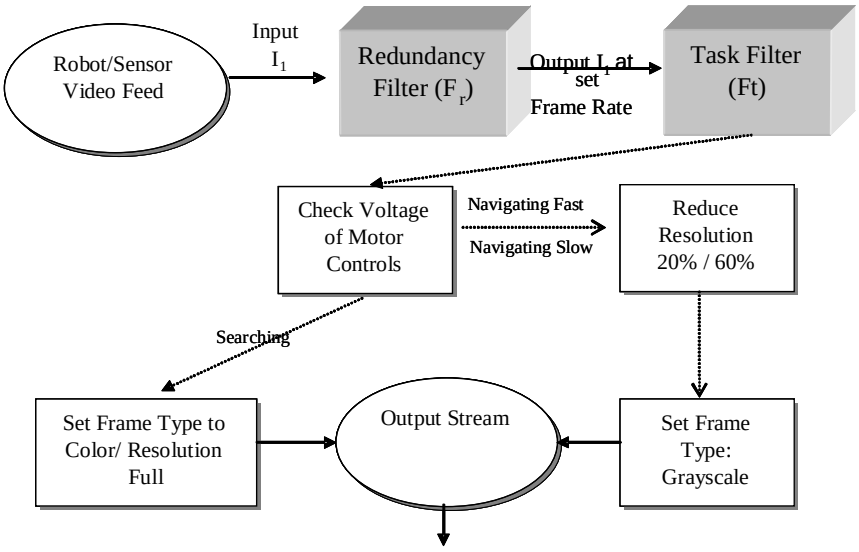

Figure 6. Outline of the F_t Algorithm.

3.4 Priority Filter

This section covers the approach and design involved in creating the Priority Filter (F_p). Specifically section 3.4.1 covers the underlying motivation for the creation of the Priority Filter, and 3.4.2 goes over the design of F_p. There are two subsections of 3.4.2 which address the Priority Policies and Contention Management portions of F_p respectively.

3.4.1 Motivation

The motivation for the creation of F_p is based on the safety, security, and rescue characteristic that different agents require different qualities of video feeds, and by taking advantage of knowing which agent is currently viewing the feed, that feed's characteristics can be customized to fit the requirements of that agent. The goal of F_p is to regulate and conserve bandwidth where possible while maximizing the amount of users able to simultaneously access current and archived video feeds.

3.4.2 Design

The priority filter is composed of a set of filters called Priority Policies and Contention Management. The two filters' conditions are satisfied by the function of a server-load and an agent type. Server-load refers to the amount of traffic the Server agent is currently experiencing based on the agents connected and the type of requests being sent. Agent-type is defined as a function of priority and the parameter set P, and P consists of values for frame rate, frame type, and resolution. The set of filters for F_p were designed with two goals in mind, first to set up a filter which restricts video format based on agent type. Second, that F_p is able to determine server load and restrict or block user access during times of contention.

As mentioned in section 3.1, the system structure overview, Knowledge-based Compression guarantees image completeness provided F_p takes place sometime after F_r.

This is made possible by first sending the video frames at the rate specified by F_r to A_s. A_s then stores the video stream locally for archival purposes, after which any requests made are serviced by copies of the original video streams. As such, any modification performed by F_p's filters will not affect A_s's ability to maintain complete information. The two filters used by F_p will be described in detail in the following two subsections.

3.4.2.1 Priority Policies

The priority policies of the system are managed by separating agents into separate tier groups based on their agent type, with the first tier representing the highest priority level attainable by an agent and the fourth tier representing the lowest priority level. When contention occurs (e.g., system resources become strained), a higher level tier will supersede lower level tiers. That is to say that if system resources are low, then the system may temporarily suspend a lower level tier in favor of a higher level tier. Each tier is assigned a default set of parameters, consisting of resolution, frame rate, and frame type (i.e. grayscale or color). Higher level tiers will have higher quality defaults.

The default values were created using an outline of possible agents of the system and their corresponding needs. The values were chosen to give a gradient of possible options available to users and are meant to be used as a sample. It is expected that values are to be changed to fit different types of scenarios and can be adjusted by the server admin.

There were four policy tiers developed which were based on an outline of possible agents of the system and their needs. The tiers and their corresponding needs can be seen in Fig. 7. When available server resources are high, tiers are able to request an upgrade to their default parameters. The implementation allows for an upgrade to the next following tier if resources are available (e.g., a Third Tier agent could request parameters normally only allowed to a Tier 2 or higher user).

Tier 1 Agent Types: Mission Specialist (MS): Medical Technician
Needs:
Color Video/Frames
Real-Time* (or as close to real time as possible)
Lossy Compression / Encoding acceptable
Frame Rate may vary depending on the task at hand
Tier 2 Agent Types: Facilitator, MS: Structural, MS: Search
Needs:
Grayscale Video/Frames Ok
Frame Rate may vary depending on the task at hand
Lossy Compression / Encoding acceptable

Tier 3 Agent Types: Incident Command Staff, MS: General Specialist
Needs:
Lossless Compression/Encoding may be needed if image enhancing performed
Frame Rate may vary depending on the task at hand
Color or Grayscale video frames may be used depending on the task
Tier 4 Agent Types: Generic
Needs:
No guarantees made about service offered by system.

Figure 7. Agent Tier Overview.

Looking at the two tiers in Fig. 8, if available server resources are high, then a Tier 3 user could request a higher frame rate up to 15 fps. Since other permitted parameters are the same between the tier 2 and 3 agents, those parameters would not be able to be upgraded. If resources become strained after a request has been granted then all agents exceeding priority are dropped back down to default levels. If an agent wishes to reduce their parameters they may do so whenever they'd like.

Tier 2 Agent	Tier 3 Agent
Default Parameters: Resolution Quality: 65% Frame Type: Grayscale Frame Rate: 15	Default Parameters: Resolution Quality: 65% Frame Type: Grayscale Frame Rate: 5

Figure 8. Priority Policy Tier Upgrade Example.

3.4.2.2 Contention Management

The contention management of the system is handled by setting a max "number" of agents which can be accessing the server at any time. This number corresponds to the total priority value of all of the agents connected. Tier 1 agents since they have the highest default settings have a priority value of 4, with each successive tier having a decremented priority value (i.e., tier 2 has value of 3, etc.). Any combination of agents may utilize the server provided the sum of their priority values stays under the max permitted. Once the max permitted number of agents is reached any additional users connecting to the server will either be queued if their priority is equal or less than all others connected to the server, or allowed to connect while another lower priority agent's transmission is temporarily suspended until space frees up. Once additional space becomes available (e.g., an agents finishes receiving their transmission), then the highest priority agent queued will be allowed to begin receipt of their requested video feed. In cases where multiple agents of the same priority were queued then the one queued the longest will be de-queued first. An outline of the F_p system and how it manages agents can be seen in Fig. 9 below.

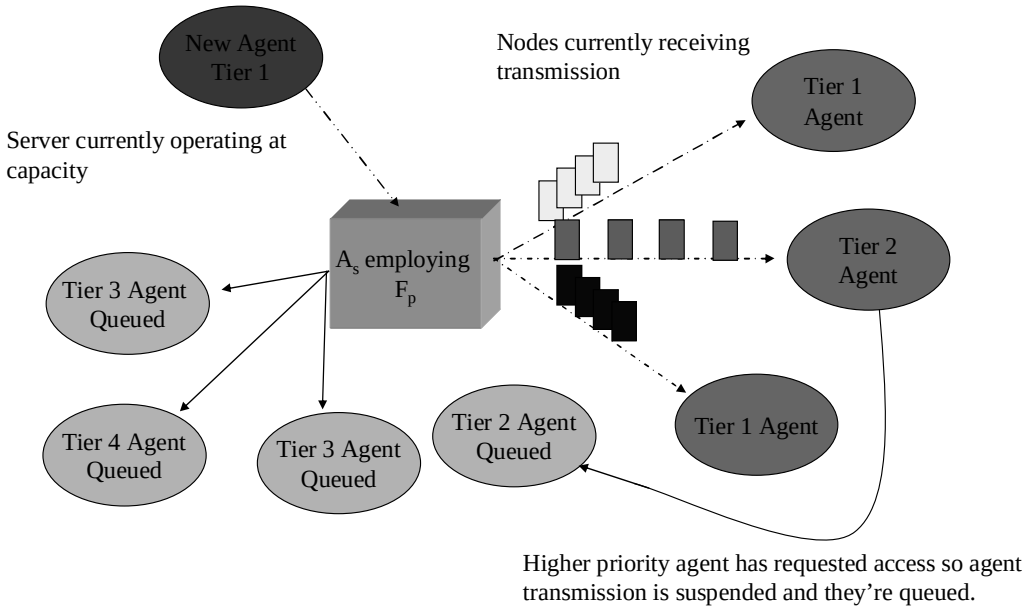

Figure 9. An Example of a Possible System State when Using F_p.

3.5 Overhead

This section covers the order complexity and operational cost of both F_r and F_t. Specifically section 3.5.1 covers the order complexity of F_r and F_t, and 3.5.2 goes over the operational cost of F_r and F_t. Order complexity and operational cost for F_p are not included. As mentioned at the beginning of this chapter although this chapter offers a complete strategy for setting up Knowledge-based Compression, for implementation it only covers F_r and F_t.

3.5.1 Order Complexity

The order complexity for F_r and F_t was calculated using Big O notation in order to describe the asymptotic behavior of the filters. The purpose behind understanding the order complexity of F_r and F_t is to give a simple but effective way of comparing the filters to other possible alternatives, which may be the case if a redesign of one of the filters is attempted in future work. Because all of the image formatting and manipulation for F_r and F_t is performed in the spatial domain, all of the functions within the filters have an order of complexity of at most O(height*width), where height and width refer to the dimensions of the image being formatted or manipulated. The spatial domain refers to the normal image space; therefore a change to a pixel in image I_1 has a direct mapping to the resulting pixel in R_1. Therefore the overall complexity for both F_r and F_t is O(height*width). The Big O complexities for the algorithms and the functions within the algorithms are given in Table 1 on the following page.

Table 1. Order Complexity for Fr, Ft, and Interface Class.

	Complexity	Comment
Overall Complexity	O(h*w)	Overall complexity for filters
* Fr	O(h*w)	Redundancy Filter.
convertImageToRendered	O(h*w)	Converts image to format usable by color conversion tools.
colorConversion	O(h*w)	Converts RGB image to Grayscale.
bufferedImageToPlanarImage	O(h*w)	Converts image to format usable by formatting tools.
differencing	O(h*w)	Subtracts new image from previous image
thresholding	O(h*w)	Determines which pixels are outside of threshold.
noiseRemoval	O(h*w)	Smoothes the amount of noise within the image.
changePrediction	O(1)	Identifies whether context or no context change has occurred.
* Ft	O(h*w)	Task Filter
convertImagetoRendered	O(h*w)	Coverts image to format usable by formatting tools.
colorConversion	O(h*w)	Coverts RGB image to Grayscale.
changeBrightness	O(h*w)	Increases brightness of image.
changeResolution	O(h*w)	Changes resolution of image.
* Interface Class	O(h*w)	Used to interface Fr and Ft with host system.
getImage	O(h*w)	Calls Ft
getFrameRate	O(h*w)	Calls Fr
Gets/Sets	O(1)	Multiple get / set functions which set parameters for Fr and Ft
* Denotes class name.		

3.5.2 Operational Cost

Understanding the operational cost for using both F_r and F_t is important when determining whether or not Knowledge-based Compression will be suited for a specific system setup. As with all compression techniques, Knowledge-based compression does involve some overhead in order to achieve a reduction in the amount of bandwidth needed for video transmission. Operational cost for F_r and F_t is defined in terms of computer utilization and the amount of processing delay to be expected when utilizing the filters. Computer utilization is simply how much of the total processing power is being used to run a specific task, which in this case is F_r, F_t, or a combination of the two. The processing delay refers to how much time is expected to lapse between the start of the filter call and the end of the filter call. The system setup used for testing was a 2.20 GHz Pentium IV with 1.00GB of PC1066 RDRAM running Windows XP.

3.5.2.1 Computer Utilization

The methodology and results for the computer utilization benchmark are described in detail in the following section. The benchmark demonstrated that on average F_r had an estimated computer utilization of 10.34% when corrected for the base line and test bench. Ft had an average estimated computer utilization of 23.74% when corrected for the base line and test bench. As well as demonstrating that when used in conjunction F_r and F_t have an average estimated computer utilization of 15.26%. The average computer utilization when F_r and F_t are combined is lower then when F_t is used standalone due to the fact that less processing power is needed when the frame rate has been reduced by F_r.

Computer utilization was monitored using Windows XP Performance Monitor. The performance monitor was set to record data in 1 second intervals, which was the minimum interval supported by the monitor. The dataset used to examine the utilization consisted of approximately 1 minute of footage to give the filters adequate time to gather enough information to compute the min, max, mean, median, and mode utilization for the filters. The utilization for both the baseline and test bench was also calculated so that the filters' utilization could be properly corrected to account for them. The baseline refers to the computer's utilization during its idle state. It's important to correct for the baseline because it accounts for background processes which can add to the overall CPU utilization. The test bench refers to the software running the filters. The same dataset used to monitor CPU utilization for the filters was also used on the test bench, but run without the use of any filters to understand how much CPU utilization is taken up by just the test bench.

The results of the computer utilization benchmark are summarized in Table 2 and Table 3. Table 2 details the uncorrected utilization values for F_r, F_t, and combined F_r and F_t, as well as giving the utilization values for the baseline and test bench. Table 3 displays the corrected utilization values for Fr, Ft, and combined Fr and Ft after they have been adjusted to take the baseline and test bench computer utilization into account. The

adjustment is calculated by taking the utilization measured in one of the filter columns and subtracting from it both the baseline and test bench utilization for that measurement.

Table 2. Computer Utilization for Baseline, Test Bench, F_r, F_t, and Combined F_r and F_t.

CPU Utilization (%)	Baseline	TestBench	Fr	Ft	Fr & Ft
Min Util:	1.56	21.88	17.19	40.63	15.63
Max Util:	15.63	78.13	100.00	89.06	96.88
Mean Util:	4.49	40.60	55.43	68.82	60.34
Median Util:	1.56	39.84	63.08	67.44	70.31
Mode Util:	1.56	26.56	64.06	76.56	76.56

Table 3. Estimated Computer Utilization for F_r, F_t, and Combined F_r and F_t. Corrected for Baseline and Test Bench Utilization.

CPU Utilization (%)	Fr	Ft	Fr & Ft
Mean Util:	10.34	23.74	15.26
Median Util:	21.67	26.03	28.91
Mode Util:	35.94	48.44	48.44
Corrected for Baseline + Test Bench			

3.5.2.2 Time Delay

The following section details the expected time delay when using either F_r or F_t. As mentioned prior, the time delay refers to the amount of time elapsed between the start of the filter call and then end of the filter call. Knowing the expected time delay is especially important in the case of F_t since how quickly the images can be transmitted depends in part on the amount of time F_t takes to format the image. For F_r the effect of time delay may have little effect on the actual system depending on whether or not a new thread is spawned when calling the filter. Since F_r only outputs the suggested frame rate, halting the video stream while waiting for a response isn't necessary. The average time delay for F_r is 88.59ms, and the average time delay for F_t is 57.6ms.

The averages for both F_r and F_t do not factor in 0ms run times in order to give a more accurate representation of how long to expect when actual processing is needed. As

described in the implementation both F_r and F_t have situations in which no processing is performed which results in a time delay of approximately 0ms. In the case of F_r, this occurs when the robot is moving since the filter checks for that at the beginning and exits if true. F_t has a time delay of approximately 0ms when the robot is searching since no formatting is performed on the image.

The run time of the two filters was calculated by reading a timer object right before and after each of the filter calls and then recording that value. The dataset used for calculating the run times consisted of 837 image pairs taken from one of the datasets used in the experimental setup described later in the thesis. The images consist of consecutive frames which are 500ms apart from each other. The gap simulates the expected frame which would be tested normally with F_r since its set to run in 500ms intervals. The gap in time does not affect F_t since the actual content of the image isn't important just the current task. The task list used for F_t was taken from a predetermined list of operation bits generated by a trained robot operator observing the dataset. The results of the time delay test can be seen in Table 4.

Table 4. Expected Run Time for both F_r and F_t.

Run Time (ms)	Fr	Ft
Mean	88.59	57.6
Median	78	62
Mode	78	62
Total frames tested 837		

Chapter Four

Experiments and Results

Experiments were conducted to determine the reduction in bandwidth for canonical imagery sequences when compared to the bandwidth used for a raw unmodified video stream. No testing research was conducted on quality of performance or user preference; the impact of frame rate and image quality has been sufficiently established by prior work and that level of analysis was beyond the scope of this work. This chapter substantiates the validity of the approach in addressing the research question from chapter one.

As stated in chapter one, the underlying research question this thesis attempts to address is as follows: How can multi-agent video bandwidth consumption be reduced while not affecting post processing and meeting the requirements of the consumers? Chapter three explained how bandwidth consumption can be reduced while preserving post processing capabilities and meeting consumer requirements. This chapter provides support for three claims about Knowledge-based Compression's bandwidth reduction. The three claims which are supported by the results described in this chapter are as follows:

1. The F_r portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 24.07% to 33.42% in a realistic search and rescue scenario.
2. The F_t portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 32.95% to 33.78% in a realistic search and rescue scenario.
3. The combined F_r and F_t portions of the system have the capability to reduce bandwidth consumption by at least 59.08% to 67.83% in a realistic search and rescue scenario.

The results are composed of three different tests which will be discussed in detail below.

Section 4.1 explains the experimental methodology used in testing. Section 4.2 describes the performance metrics used. Section 4.3 describes the results for the three tests. Finally, section 4.4 summarizes the findings.

4.1 Experimental Methodology

Section 4.1 covers the experimental setup for the three tests carried out on the various portions of Knowledge-based Compression. Specifically, section 4.1.1 describes the experimental setup for the F_r reduction test. Section 4.1.2 describes the experimental setup for the F_t reduction test. Finally, section 4.1.3 describes the experimental setup for the comparison of both F_r and F_t versus an MPEG encoded and unmodified video stream.

4.1.1 F_r Reduction Test

The purpose of the F_r reduction test was to determine a reasonable amount of expected reduction when using F_r. The experiment makeup of the F_r reduction test consisted of a control and test video, which were used to compare the effectiveness of F_r at bandwidth reduction. The test was conducted outdoors on a rubble pile test bed, and the final feed used for testing was approximately 1 minute in length.

The experiment scenario consisted of a robot moving through a rubble pile looking for survivors; once a survivor was located the robot monitored that area and its surroundings. The robot was teleoperated by a search and rescue expert and used a front camera mounted for capturing the video.

F_r was run with three sets of *tolerance* and *threshold* values to compare bandwidth savings with accuracy. F_r was set to run every 400 ms, and the *tolerance/ threshold* values used for testing were 5/10, 5/20, and 10/20. The Fr run time delay of 400 ms was chosen because it was between the range of 300 and 500 ms specified in the design and implementation of chapter 3. The threshold and tolerance parameters were chosen using the range of values determined through previous empirical testing which didn't result in

any missed context changes (i.e., the parameters used were more likely to detect small amounts of motion). The parameters were kept sensitive because the experiment was simulating a search and rescue scenario; therefore the accuracy of the algorithm in detecting any context changes was important. An example set of test images used by F_r can be seen in Fig. 10.

Figure 10. Image Pairs Used for Determination of Redundancy in the F_r Portion of Knowledge-based Compression.

4.1.2 F_t Reduction Test

The goal of the F_t reduction test was to compare the reduction savings of the F_t portion of Knowledge-based Compression with that of an unmodified video transmission. The F_t reduction test consisted of two similar datasets taken from archived footage of a robot being teleoperated through a collapsed tunnel system. The two datasets were composed of archived video feeds of a robot navigating through a partially collapsed pipe system. The total length of the two tests was 7:32min and 6:59min. The two new datasets were chosen for the comparison over the original dataset used in 4.1.1 because the initial dataset was not long enough to adequately categorize distinctions in robot operation. Since the datasets used were taken from archived footage, the technology was not in place at the time to monitor changes in operation; therefore a trained robot operator was used to generate a predetermined list of operation bits to satisfy the requirements of the F_t portion.

The operation bit had three possible values: *searching*, *navigating slow,* and *navigating fast*.

4.1.3 F_r and F_t Combined Reduction Test

When Knowledge-based Compression is implemented in the field it's expected that both F_r and F_t will be used in conjunction, it is therefore important to determine the amount of reduction which can be expected. The F_r and F_t combined reduction test consisted of the same two datasets used in the standalone F_t reduction test and compares the bandwidth throughput of the combined F_r & F_t portion of Knowledge-based Compression with that of an unmodified video transmission and an MPEG transmission. The MPEG stream was encoded at 375Kb/sec to simulate the amount of encoding needed if 4 concurrent MPEG streams were to be run on a 1.5Mb connection, which would be the case if the system was experiencing somewhat heavy congestion. The F_r portion of the test was set with a *threshold* of 10 and *tolerance* of 5% to demonstrate how the system performs using the strictest set of values used in the first test. Since the datasets used in this test were identical to the datasets used in the F_t reduction test the same constraints were used to set the operation bits for the F_t portion of Knowledge-based Compression.

4.2 Performance Metrics

This section describes the performance metrics used to examine the performance of Knowledge-based Compression at reducing bandwidth. The F_r reduction test was run to test how the filter manages bandwidth during a real world application (e.g., search and rescue operation), and therefore the *average throughput*, *accuracy*, and *total bandwidth difference* were tested. Average throughput was calculated by monitoring the average amount of bandwidth used in transmission over time and was used to demonstrate the difference in throughput between the system using F_r and a standard video transmission stream. The accuracy of each of the tests was monitored by noting whether or not that

parameter set missed a motion event. The *total bandwidth difference* was the percentage change in total bandwidth used between the standard method of transmission and the system using F_r. The metrics were chosen to substantiate that an acceptable balance between bandwidth savings and accuracy must be decided on for each situation.

The F_t reduction test was measured using the *total bandwidth difference* to demonstrate the capabilities of F_t when used standalone.

The metrics for the F_r and F_t combined reduction test consisted of *average overall throughput over time* of the feeds, *throughput over time*, and *total bandwidth difference*. The metrics were measured using the same methods used in the F_r reduction test. The metrics were chosen to best demonstrate the range in the bandwidth savings of Knowledge-based Compression in comparison to both an unmodified video stream and an MPEG encoded video stream.

4.3 Results and Analysis

Section 4.3 covers the results and analysis for the three tests carried out on the various portions of Knowledge-based Compression. Specifically, section 4.3.1 covers the results and analysis for the F_r reduction test. Section 4.3.2 describes the results and analysis for the F_t reduction test. Finally, section 4.3.3 describes the results and analysis for the F_r and F_t combined reduction test.

4.3.1 F_r Reduction Test

As expected, as the sensitivity of the parameters increased, the amount of bandwidth savings decreased. Therefore it is important to find an acceptable balance between bandwidth savings and accuracy. The sensitivity of F_r should be adjusted based on the current mode of use. The sensitivity is adjusted based on *tolerance* and *threshold* values. By using *tolerance* and *threshold* values that are within the acceptable range found in this test, F_r is able to reduce bandwidth consumption by at least 24.07% to 33.45% in a

realistic search and rescue scenario. Section 4.3.1.1 outlines the results from the F_r reduction test.

4.3.1.1 Comparison of Average Throughput Over Time

The F_r reduction test was carried out to realistically determine how the system performs when used in a non-lab environment. The initial test was run using the optimal *tolerance*/*threshold* values found in the initial empirical testing of the system which were a *tolerance* of 10% and a *threshold* of 20 pixels; however those values were found to be too lenient for accurately characterizing changes in the environment. Specifically, once the survivor was found there were a few small movements that went undetected by F_r. While a vast majority of the movement was detected, the nature of the scenario proved that a more sensitive set of values were preferable. The results of the parameters used can be seen in Fig. 11 and Table 5, and show that the F_r portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 24.07% to 33.45% when compared to an unmodified video stream. As expected, as the sensitivity of the parameters increased, the amount of bandwidth savings decreased. Therefore it is important to find an acceptable balance between bandwidth savings and accuracy. The sensitivity of F_r should be adjusted based on the current mode of use. If the current situation is non-critical then the sensitivity of F_r can be reduced to increase bandwidth savings, however once something of interest is noted then the sensitivity should be increased in order to ensure accuracy.

Table 5. Total Bandwidth Difference Between F_r Parameter Sets and Standard Transmission.

Parameter Set Tol = Tolerance Thresh = Threshold	Total Mbits Sent	Reduction (%)
Tol 5 Thresh 10	82.22	24.07%
Tol 5 Thresh 20	76.90	29.87%
Tol 10 Thresh 20	71.95	33.45%
Standard Transmission total 108.11 Mbits		

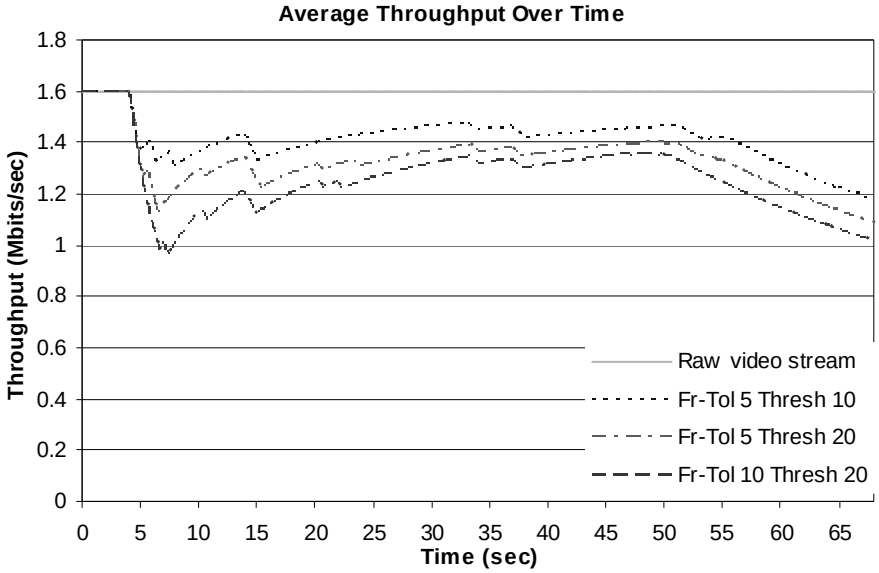

Figure 11. F_r Performs Better Regardless of Sensitivity but Bandwidth Savings are Improved with Relaxed Sensitivity Settings.

4.3.2 F_t Reduction Test

This section outlines the performance benefits of the F_t portion of Knowledge-based Compression. Specifically, section 4.3.2.1 demonstrates that F_t is capable of reducing bandwidth by at least 32.95% to 33.78%.

4.3.2.1 F_t vs. Unmodified Reduction

The F_t reduction test supports that F_t has a favorable result on reducing bandwidth when compared with an unmodified video stream. The first dataset resulted in 33 distinct operation changes (*Search*, *Navigating-Slow*, and *Navigating-Fast*) with a total throughput of 1154.35Mbits. When compared with the unmodified video stream which transmitted 1721.52Mbits, F_t resulted in a total reduction of 32.95%. The second dataset resulted in 46 distinct operation changes with a total throughput of 950.56Mbits. When compared with the unmodified video stream which transmitted 1435.41Mbits, F_t resulted in a total reduction of 33.78%. A summary of the results can be seen in Table 6. The column labeled operation changes refers to the amount of switches between *navigating* and *searching* which took place during that transmission period.

Table 6. Total Bandwidth Difference Between F_t Datasets and Standard Transmission.

	Total Mbits Sent	Standard Transmission Mbits Sent	Reduction (%)	Operation Changes
Dataset 1	1154.35	1721.52	32.95	33
Dataset 2	950.56	1435.41	33.78	46

4.3.3 F_r & F_t Combined Reduction Test

Knowledge-based Compression vs. MPEG Comparison test performed as expected, and showed that the F_r and F_t portions of the system have the capability to reduce bandwidth consumption by at least 59.08% to 67.83%, when compared to an unmodified video stream. However, Knowledge-based Compression does not have the bandwidth savings of an MPEG encoded video stream but does allow for more functionality in regulating the stream. Section 4.3.3.1 describes the reduction capabilities of the combined F_r and F_t portions. Section 4.3.3.2 covers the throughput over time of the three streams. Finally,

section 4.3.3.3 goes over the average throughput over time of the three streams which gives a more abstract view of the throughput for the three streams.

4.3.3.1 F_r and F_t vs. MPEG vs. Unmodified Reduction

The combined F_r and F_t comparison test resulted in much better bandwidth savings than the two standalone tests for F_r and F_t. Since the datasets used were identical to the standalone F_t test the number of changes in operation registered by F_t were the same. The results of the bandwidth difference comparison can be seen in Tables 7 and 8.

Table 7. Total Bandwidth Difference Between Video Feed Transmissions for First Dataset.

Video Feed	Total Mbits Sent	Reduction (%)
F_r & F_t	553.82	67.83%
MPEG	164.25	90.46%
Standard Transmission total 1721.52 Mbits		

Table 8. Total Bandwidth Difference Between Video Feed Transmissions for Second Dataset.

Video Feed	Total Mbits Sent	Reduction (%)
F_r & F_t	587.43	59.08%
MPEG	151.91	89.42%
Standard Transmission total 1435.41 Mbits		

4.3.3.2 Comparison of Throughput Over Time

The results demonstrate that although Knowledge-based Compression mostly transmits at a rate in between an unmodified and MPEG encoded stream, Knowledge-based Compression does occasionally offer a better rate than MPEG. The spikes in throughput seen in the results, Fig. 12 and Fig. 13, are brought about by changes in scene context; during longer periods of no context change the rate offered by Knowledge-based Compression is often better than that of MPEG.

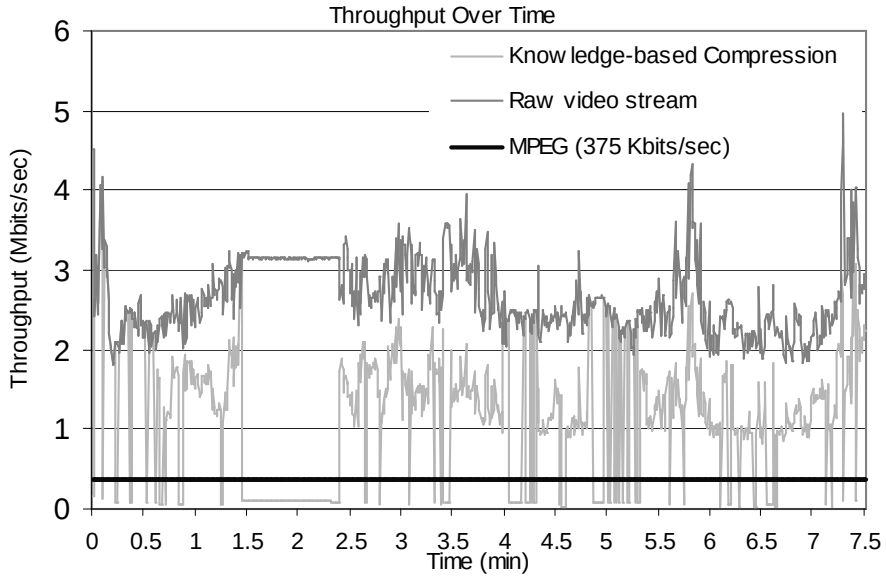

Figure 12. Displays Spikes in Knowledge-based Compression Performance Resulting from Context Changes for Dataset 1. The Figure Also Shows that Knowledge-based Compression Occasionally Performs Better Than MPEG.

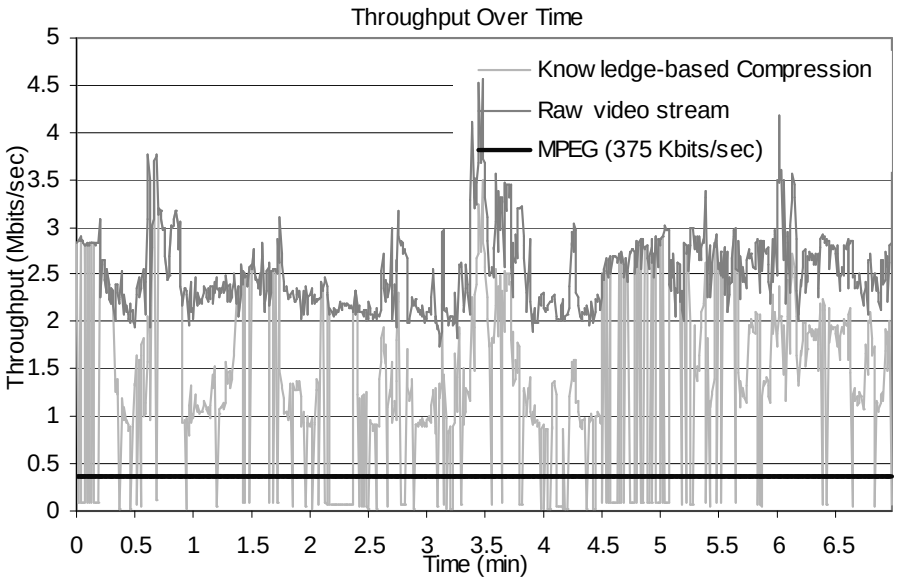

Figure 13. Displays Spikes in Knowledge-based Compression Performance Resulting from Context Changes for Dataset 2. The Figure Also Shows that Knowledge-based Compression Occasionally Performs Better Than MPEG.

4.3.3.3 Comparison of Average Throughput Over Time

The figures on the average throughput over time used by the three methods of video transmission can be seen in Fig. 14 and Fig. 16 below. The results from the comparison of average throughput over time show that when averaged out the spikes in Knowledge-based Compression performance disappear and it transmits at a rate much lower than an unmodified video stream but above that of an MPEG encoded stream.

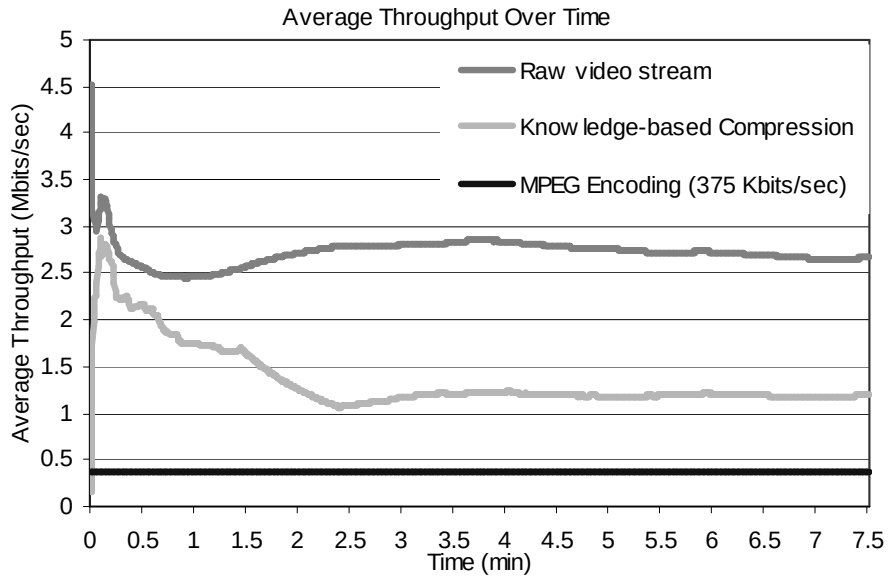

Figure 14. Knowledge-based Compression Performance is Between that of an Unmodified Stream and an MPEG Encoded Stream in the First Dataset.

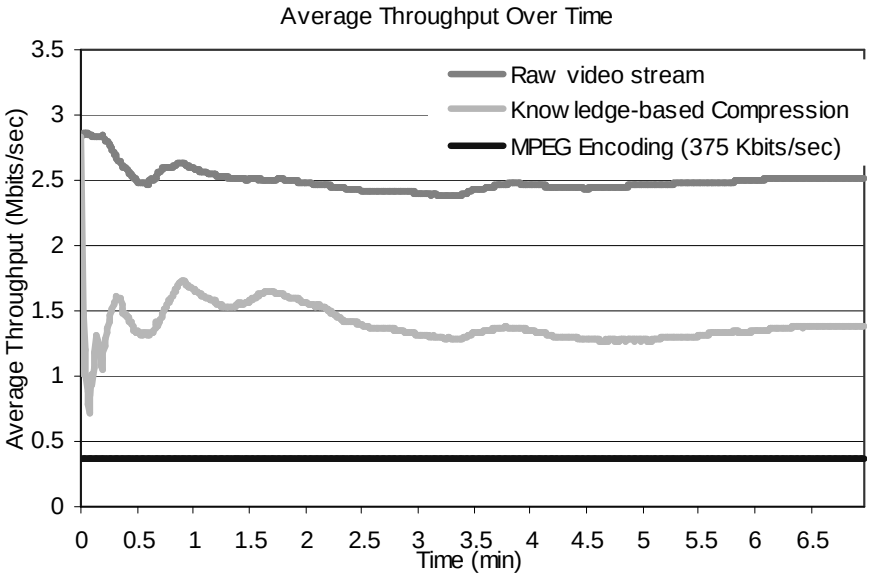

Figure 15. Knowledge-based Compression Performance is Between that of an Unmodified Stream and an MPEG Encoded Stream in the Second Dataset.

4.4 Summary

Three claims about Knowledge-based Compression are supported by the results described in this section. First, that the F_r portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 24.07% to 33.42% in a realistic search and rescue scenario. Second, that the F_t portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 32.95% to 33.78% in a realistic search and rescue scenario. Lastly, that the combined F_r and F_t portions of the system have the capability to reduce bandwidth consumption by at least 59.08% to 67.83% in a realistic search and rescue scenario.

The results demonstrated that in all three tests Knowledge-based Compression resulted in a savings in bandwidth when compared with an unmodified video stream. Although Knowledge-based Compression does not offer the bandwidth savings of MPEG, it does offer a viable alternative if post processing is needed or if the system can not afford the overhead involved in encoding a video stream using MPEG. As stated previously Knowledge-based Compression was designed to ensure that complete information obtained from the robot or sensor is able to be preserved while reducing the overall bandwidth used in servicing requests for that data. Since Knowledge-based Compression was designed for an application in which MPEG compression is not suitable, e.g. post processing, it is still a viable alternative for video transmission.

Chapter Five

Conclusions

This thesis has addressed the issue of finding a suitable strategy for regulating bandwidth transmission during safety, security, and rescue operations while satisfying the constraints imposed on it due to the nature of the situation. From chapter one, the constraints imposed on the system are as follows:

1. Each user application may require its own post-processing of the imagery. Post-processing computer vision algorithms often cannot be performed on images which have undergone lossy compression.
2. The imagery may need to be used for forensic assessment or evidence at a later date. Therefore, the complete video stream must be stored and any video compression must be reversible. Off-board storage of video from a robot or sensor is highly desirable since the field device may be destroyed as the incident unfolds or may simply fail at an inopportune moment.

With the system in place the following claims can be substantiated as demonstrated in this thesis, they address the issues posed by the constraints as well as addressing the underlying motivation to find an alternative method of bandwidth transmission for safety, security, and rescue applications:

1. With the system as a whole, consisting of Information, Processing, and Server agents, service is reversible if the following constraints are met: The redundancy filter is applied before the task filter for the Immediate-Processing agent, or the redundancy filter is applied before the priority filter for the Post-Processing agent.
2. The redundancy filter will reduce bandwidth consumption while maintaining complete information of the images and be completely reversible.

3. Each of the three filters will reduce bandwidth consumption.
4. The system as a whole will maintain complete information, while supporting multiple agents.

Chapter two presented a review of work related to Knowledge-based Compression. The related work established that although there are many strategies available for video transmission, there is currently not an existing system in place which is designed to meet the needs of safety, security, and rescue operations. The work also established that the modifications performed on the video feed will not impact the performance of human operators since the regulation follows in line with the results in human factors studies.

Chapter three presented the approach and design used to implement Knowledge-based Compression. The design showed that provided the filters are applied in the correct order overall bandwidth consumption can be reduced while still allowing the system to maintain *complete* information.

Chapter four presented the experimental setup, the results, and the conclusions which can be drawn from the results for Knowledge-based Compression. The results demonstrated that Knowledge-based Compression always performs better than an unmodified video stream however seldomly better than when the video is encoded using MPEG. However, due to the criteria needed by safety, security, and rescue operations, Knowledge-based Compression is an optimal choice for video transmission since it is the only system currently available to meet all of the criteria and still reduce bandwidth consumption.

This chapter provides a summary of the findings in section 5.1. Section 5.2 presents a discussion on the implications of the findings. Section 5.3 concludes with an overview of possible avenues for future work.

5.1 Findings

How can multi-agent video bandwidth consumption be reduced while not affecting post processing and meeting the requirements of the consumers? The above question was based off the assumptions at the start of the research. The assumptions going into the research were that the system must be able to reduce the overall bandwidth transmission between robots, sensors, and operators, while following a strict set of constraints imposed on the system. The constraints on the system were as follows:

1. Each user application may require its own post-processing of the imagery. Post-processing computer vision algorithms often cannot be performed on images which have undergone lossy compression.
2. The imagery may need to be used for forensic assessment or evidence at a later date. Therefore, the complete video stream must be stored and any video compression must be reversible. Off-board storage of video from a robot or sensor is highly desirable since the field device may be destroyed as the incident unfolds or may simply fail at an inopportune moment.

Knowledge-based Compression was found to be the solution to the aforementioned question as shown in the explanation and results covered in this thesis. Because Knowledge-based Compression utilizes filters which either perform lossless compression on the video transmission or backup video prior to performing lossy compression, it ensures that complete forensics data is maintained while suiting the needs of specific users. Knowledge-based Compression was also shown to reduce video transmission and the following results were obtained. First, that the F_r portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 24.07% to 33.42% in a realistic search and rescue scenario. Second, that the F_t portion of Knowledge-based Compression has the capability to reduce bandwidth consumption by at least 32.95% to 33.78% in a realistic search and rescue scenario. Lastly, that the combined F_r and F_t portions of the system have the capability to reduce bandwidth consumption by at least 59.08% to 67.83% in a realistic search and rescue scenario.

The results from both the simulated rescue operation and the archived footage of the collapsed pipe system show that Knowledge-based Compression reduces transmission of video frames when compared to that of an unmodified video transmission; moreover when certain conditions are met (i.e., very little context change), Knowledge-based Compression can result in better performance than that of an MPEG encoded stream. However, typically MPEG will outperform Knowledge-based Compression since the throughput can be set at any value without regard to the quality of the encoded video stream. Since Knowledge-based Compression was designed for an application in which MPEG compression is not suitable (e.g., post processing), it is still a viable alternative for video transmission.

5.2 Discussion

This thesis has demonstrated a strategy for reducing the amount of bandwidth needed for video transmission over wireless networks. This strategy satisfies the needs of tactical and downstream users and allows complete video to be archived for forensics or evidence. The strategy focuses on knowledge at the application layer of a wireless network to meet the constraints imposed on it by the system, as opposed to MPEG or H.26x encoding which operates at the presentation layer of the OSI reference model and offer a less dynamic compression.

As mentioned in the earlier chapters Knowledge-based Compression is composed of three different types of filters: F_r, F_t, and F_p. When used in conjunction they allow the system to reduce bandwidth consumption as well as preserve data integrity for later post processing. The system is not without its own limitations however. Because Knowledge-based Compression is constrained to preserve complete image data the savings will generally not be as good as that of a strictly lossy encoded stream such as MPEG. The filters are subject to some limitations as well, which will be discussed in detail below.

The redundancy filter was designed to regulate frame rate transmission based on context change within the robot or sensors field of view. Although fairly accurate, the

parameters for the filter may have to be modified depending on the type of scenario the filter is being used in. A more dynamic approach to adjusting filter parameter values may be better suited to the task, however was outside the scope of this thesis. The issue with the current parameter setup is that although the default settings for the parameters will be fairly accurate across all types of scenarios, they must be adjusted before hand at the start of each scenario to ensure the best performance. This required adjustment is due to the fact that conditions may change, such as poor lighting, or other issues which could disrupt the filters ability to accurately categorize context change.

The task filter is much less prone to issues resulting from scenario changes but does have some limitations stemming from the robot itself. As mentioned in chapter 3, the task filter modifies the video stream based on the current task at hand, and categorizes tasks as *searching*, *navigating-fast*, and *navigating-slow*. The task filter uses voltage thresholds taken from the robot's motor controls to categorize which task the robot is currently in. The limitation with the robot comes from the fact that the robot doesn't accurately categorize its voltage levels, and rather then having a gradient of voltage more often then not it's either at min or max voltage depending upon whether or not the robot is stopped or moving. This problem with the voltage causes the task filter to effectively only use two different types of tasks, *searching* and *navigating-fast*. Although the bandwidth savings are still present and in fact better since navigating slow has less bandwidth savings then navigating fast, the user does receive a reduced quality of perception from the formatting.

The priority filter regulates access and formatting based on the user type and current server congestion. The limitations present within the priority filter arise from the method of identifying server congestion. Unfortunately an accurate low cost method for identifying the available bandwidth within a network is not readily available or easily implemented within the system, and implementing one would be outside the scope of this thesis. Instead of using an *ABET*, the priority filter estimates server load based on the type of user connected and formatting requested. Although this technique allows a fairly good

estimation of server load, the server threshold must be adjusted whenever the server's maximum bandwidth changes.

Although there are some limitations present within the system, Knowledge-based Compression still offers a better alternative to regulating video transmission then the standard method currently used to transmit video streams in safety, security, and rescue operations. The limitations mentioned above offer avenues for possible future work, which will be discussed in the following section.

5.3 Future Work

As mentioned in the discussion some possible options for future work within the individual filters include: implementing a more dynamic method for setting sensitivity in the F_r portion of the system, identifying what is causing the robot to misreport voltage, as well as adding more intelligence to the detection of server load in the F_p portion. The future work related to the sensitivity of the F_r portion has to do with the fact that sometimes one set of settings does not necessarily address the needs of all situations. Although there are already methods currently in place to manually modify the sensitivity of the F_r portion, a more dynamic approach would be preferred since it is likely that the operator will not have the time to manually tune the sensitivity. The issue with the robot misreporting voltage stems from the fact that regardless of how fast the robot is moving, the voltage maintains fairly constantly except when the robot is stationary at which point it drops. Once the voltage is accurately reported the task filter can then perform a more gradual formatting of the image feed rather then changing it from one extreme to the other. Finally, the current method of detection of server load uses knowledge about the typical load used by different agent types. However, the server load settings are tailored with a specific network in mind and therefore must be changed if the size of the maximum available bandwidth changes. A better method of monitoring the server load would use bandwidth estimation techniques which are currently being researched academically, and unfortunately were not able to be utilized in the current system.

Other broader avenues of future work could involve expanding some of the concepts used in the filters to meet other applications which may not fall under the same constraints in which Knowledge-based Compression was designed for. For instance in cases where there are obvious differences in tasks being monitored, the task based filter could be adapted to fit. Although Knowledge-based Compression is fully functional in its current form its concept can easily be adapted to fit other applications by adjusting the various filters to fit the needs of the application.

References

[1] R. Murphy, J.L. Burke, "Human-Robot Interaction in USAR Technical Search: Two Heads are Better Than One," also appears in *Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (RO_MAN 2004)*.

[2] H. Anderson, M. Jackson, A., R. McEwan, and J. Mullin, “Impact of video frame rate on communicative behaviour in two and four party groups,” in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000. pp. 11-20.

[3] R. Murphy, J.L. Burke, "Up from the Rubble: Lessons Learned about HRI from Search and Rescue," also to appear in "Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society".

[4] G. Ghinea, S. R. Gulliver, “Stars in their eyes: what eye- tracking reveals about multimedia perceptual quality,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol.34, Iss.4, pp. 472-482. July 2004.

[5] J. D. McCarthy, D. Miras, M. A. Sasse, “Sharp or Smooth?: comparing the effects of quantization vs. frame rate for streamed video,” in *Proceedings of the SIGCHI conference on Human factors in computing systems,* 2004. pp. 535-542.

[6] A.F.T. Winfield, “Wireless video tele-operation using internet protocols,” in *Proceedings of the Fourteenth International Conference on Unmanned Air Vehicle Systems*, Bristol, April 1999.

[7] F. Kai-Tat, S. Wan-Chi, C. Yui-Lam, “Dynamic frame skipping for high-performance transcoding,” in *International Conference on Image Processing*, , 7-10 Oct 2001, Vol.1, pp. 425-429.

[8] L. Chia-Wen, H. Jenq-Neng, W. Tzong-Der, “Dynamic frame-skipping in video transcoding,” in *IEEE Second Workshop on Multimedia Signal Processing*, 7-9 Dec. 1998, pp. 616-621.

[9] P. Chih-Wei, C. Mei-Juan, C. Ming-Chung, “Efficient Motion-Estimation Algorithm for Reduced Frame-Rate Video Transcoder,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2002. Vol 12, No. 4. pp. 269-275. April 2002.

[10] B. Keepence, "Turning to MPEG-4," *Security*, Vol.41, Iss.3, pp. 49, Mar 2004.

[11] P. Korzeniowski, "As Wireless Networks Spread, Video Transmissions Taking Root," *Investors Business Daily*, pp. 7, 2004.

[12] J. Friedrick, "Industry has multiple options at hand for video transmission," *Security Systems News*, Vol.6, Iss. 11, Criminal Justice Periodicals, pp. 35. Nov. 2003.

[13] T. I. Page, "Incorporating Scene Mosaics as Visual Indexes into UAV Video Imagery Databases," M.S. thesis, Air Force Institute of Technology. OH. USA. 1999.

[14] D. Whitney, "Cellphone delivered video becomes eye of the storm," *Electronic Media*, Chicago, Vol.21, Iss.2, pp. 12, Jan 14, 2002.

[15] J. Friedrick, "Video transmission moves to next level of sophistication with expanded options," *Security Systems News*, Dec 2002, Vol.5, Iss.12, Criminal Justice Periodicals, pp. 28.

[16] K. Chen, K. Nahrstedt, S. H. Shah, "Dynamic bandwidth management for single-hop ad hoc wireless networks," in *Pervasive Computing and Communications*, IEEE, 2003. pp. 195-203. 23-26 March 2003.

[17] K. Sungwook, P. K. Varshney, "An integrated adaptive bandwidth-management framework for QoS-sensitive multimedia cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 835-846, May 2004.

[18] Y. Horibe, Y. Zhang, "Adaptive QoS-guaranteed channel reservation in multimedia wireless networks," in *Circuits and Systems and West Sino Expositions*, IEEE, 2002, Vol.1, pp. 404-408, 29 June-1 July 2002.

[19] J. Kim, C. Oliveira, T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE Journal on Selected Areas in Communications*, Vol.16, Iss.6, pp. 858-874, Aug 1998.

[20] M. Forshaw, M. Hodgetts, S. Young, "Image comparison methods for perimeter surveillance," in *Image Processing and Its Applications*, 1999, Vol.2, pp.799-802, 1999.

[21] P. Bouthemy, N. Peyrard, "Motion-based selection of relevant video segments for video summarization," in *International Conference on Multimedia and Expo*, ICME, 2003. Vol.2, pp. 409-412, 6-9 July 2003.

[22] L. Bruzzone, D. F. Prieto, "An adaptive semi parametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *IEEE Transactions on Image Processing*, Vol.11, Iss.4, pp. 452-466, Apr 2002.

[23] L. Bruzzone, R. Cossu, "An unsupervised change detection technique robust to registration noise," in *IEEE International Geoscience and Remote Sensing Symposium*, IGARSS, 2002. Vol.1, pp. 306-308.

[24] P. Rosin, "Thresholding for change detection," in *Sixth International Conference on Computer Vision*, 1998. pp. 274-279, 4-7 Jan 1998.

[25] Caruso, A., F. Melgani, G. Moser, S. B. Serpico, "Partially supervised detection of changes from remote sensing images," in *IEEE International Geoscience and Remote Sensing Symposium*, IGARSS, 2002. Vol.1, pp. 299-301.

[26] M. K. H. Leung, L. Liyuan, "Integrating intensity and texture differences for robust change detection," *IEEE Transactions on Image Processing*, Vol.11, Iss.2, pp. 105-112, Feb 2002.

[27] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image Change Detection Algorithms: A Systematic Survey," *IEEE Transactions on Image Processing*, January 2005.

[28] M. Black, D. Fleet, Y. Yacoob, "Robustly Estimating Changes in Image Appearance," *Computer Vision and Image Understanding*, Vol. 78, pp. 8-31, 2000.

[29] E. H. Mamdani, "Twenty years of fuzzy control: experiences gained and lessons learnt," in *Second IEEE International Conference on Fuzzy Systems*, 1993. pp. 339-344.

[30] A. Florescu, V. Alexandru, D. A. Stoichescu, "Fuzzy control of AC-DC converters supplying high precision DC drive systems," in *24th International Spring Seminar on Electronics Technology: Concurrent Engineering in Electronic Packaging*, 2001. pp. 193-196.

Appendices

Appendix A:
Definitions of Knowledge-based Compression Related Terms

The following list of definitions covers the terminology used in the paper and is provided as a reference.

1. *ABET.* Available Bandwidth Estimation Tool.

2. *Agent.* A human, robot, or computer which interacts with the world to make changes or to sense what is happening.

3. *Band Reduction.* The process by which the number of image bands are reduced. For instance, RGB possesses 3 bands whereas grayscale has 1 band.

4. *Differencing.* The process where image I_1 is subtracted from image I_2 on a 1 to 1 pixel basis.

5. *Complete.* The state of an image frame which allows the same result to be obtained from an algorithm, filter, or some other form of processing regardless of the images' locality in relation to the original source.

6. *DWFS.* In DWFS, each flow has a weight which defines its bandwidth requirement relative to that of other flows. A scheduler combined with the link level IEEE 802.11 protocol then schedules the flows so their received throughput is proportional to their weights.

7. *Knowledge-based Compression.* A collection of filters which use dynamic information to regulate bandwidth, and when used in conjunction with each other can guarantee reversible server as well as a reduction in bandwidth transmission.

8. *Median Filter.* A non-linear digital filtering technique often used to remove noise from images or other signals.

9. *Processing Agent.* The role of a processing agent is to process data received from another source or agent to autonomously effect change within its scope of existence.

10. *Reversible.* Processing performed on an image is reversible if further processing can be carried out to return the image back to a complete state.

11. *Server Agent.* The role of a server agent is to maintain state data of data received to ensure that later change to the data will not affect other agents' ability to obtain complete information.

Appendix A: (Continued)

12. *Threshold.* In the case of F_r, threshold refers to the range of difference between a one to one mapping of pixels between images I_1 and I_2 where change is acceptable.

13. *Thresholding.* Thresholding occurs after differencing has taken place and involves clumping together similar groups of pixels. In the case of F_r pixels are assigned as either black or white pixels depending on whether they are below or above a set value.

14. *Tolerance.* In the case of F_r, tolerance refers to the percentage of change between the two images I_1 and I_2 that is allowed before they are categorized as non-redundant.