SYMBIOTIC RATE CONTROL AND FOCAL AREA RESOLUTION CONTROL OF MPEG2 TRANSCODER

A thesis submitted
to Kent State University in partial
fulfillment of the requirements for
the degree of Master of Science

by

Qiong Gu

May 2004

Thesis written by
Qiong Gu
B.S., Shanghai University of Science and Technology, 1988
M.S., Kent State University, 2004

Approved by

____, Advisor

____, Chair, Department of Computer Science

____, Dean, College of Arts and Sciences

TABLE OF CONTENTS

APPENDIX A

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thanks to my family, without their love and support, I would not have done this thesis. Thanks to my advisor, Dr. Javed Khan, who guided my research work and share his creative ideas with me.

I am very thankful for having known and worked with people in the medianet lab, Computer Science Department. I had pleasant experience throughout my MS program.

Chapter 1

INTRODUCTION

1.1 Technology over view

In recent years, the Internet is popularly used in all fields all over the world. The abrupt growing of applications and businesses via Internet leads to heavy traffic in it. A statistic showed that the Internet traffic doubled once each year over the decade of the 1990's [1]. With the massive increase in demand for Internet services, related technologies, including software and hardware, have been improved quickly and efficiently within the past ten years.

One of the most remarkable improvements took place within the backbone network. Table 1-1 shows the development process of backbone technology [2]. The researchers are developing and using Internet2 [3], the next-generation network, operates at speeds up to one hundred times faster than the existing public Internet. The Abliene Network [4] is a high-performance backbone network used by the Internet2 community. It was scheduled for release in 2003 and included TAT-14 lines capable of transferring the amazing 640 Gigabits per second. Increased bandwidth is the key to providing enhanced web offerings, such as video and audio applications (including broadcast-quality television), and large-scale software distribution.

Bandwidth Rates	Backbone	Year
56Kbps	NSFNET	1986
1.54 Mbps	T1 or DS-1	1985
46.08 Mbps	T3 or DS-3	1992
51.08 Mbps	OC-1	1993
155 Mbps	OC-3	1995
600 Mbps	OC-12	1996
2.4 Gbps	OC-48	2001
9.6 Gbps	OC-192	2001
13.271 Gbps	OC-256	2002
39.813 Gbps	OC-768	2002
159.252 Gbps	OC-3072	2004
640 Gbps	TAT-14	2003

Table 1-1 Backbone Bandwidth Rates

According to the *Boardwatch Directory of Internet Service Providers,* between 1996 and 1998, the total U.S. Internet trunk doubled in capacity every seven months [5]. The reports from TeleGeography Inc. shows that international Internet bandwidth grew 174 percent between 2000 and 2001. Table 1-2 shows 1999-2001 Regional Internet bandwidth within the past 10 years [6].

Nation	2000 (Mbps)	2000 (Mbps)	Percent Growth
Africa	649.2	1,230.8	89.6%
Asia	22,965.1	52,661.9	129.3%
Europe	232,316.7	675,637.3	190.8%
Latin Amercica	2,785.2	16,132.5	479.2%
U.S. & Canada	112,222.0	274,184.9	144.3%

Table 1-2 International Internet Bandwidth by Region, 1999-2001

A network with these data transmission rates may be capable of delivering 100 to 150 Mbps to the desktop, which will support streaming capabilities. ‘Streaming’, in the context of communication, means that the audio/video file is transmitted while it is being created and converted a continuous video and sound file at the receiving point. It allows high quality multimedia material to be delivered in ‘real time’. The advantage of streaming, compared to downloading, is that a streamed asset can be played while it is being transferred to the client system. When a media asset is not streamed, it needs a while to be downloaded completely before it can be played. This advantage of streaming makes it possible to play multimedia data over a network with an unknown distance.

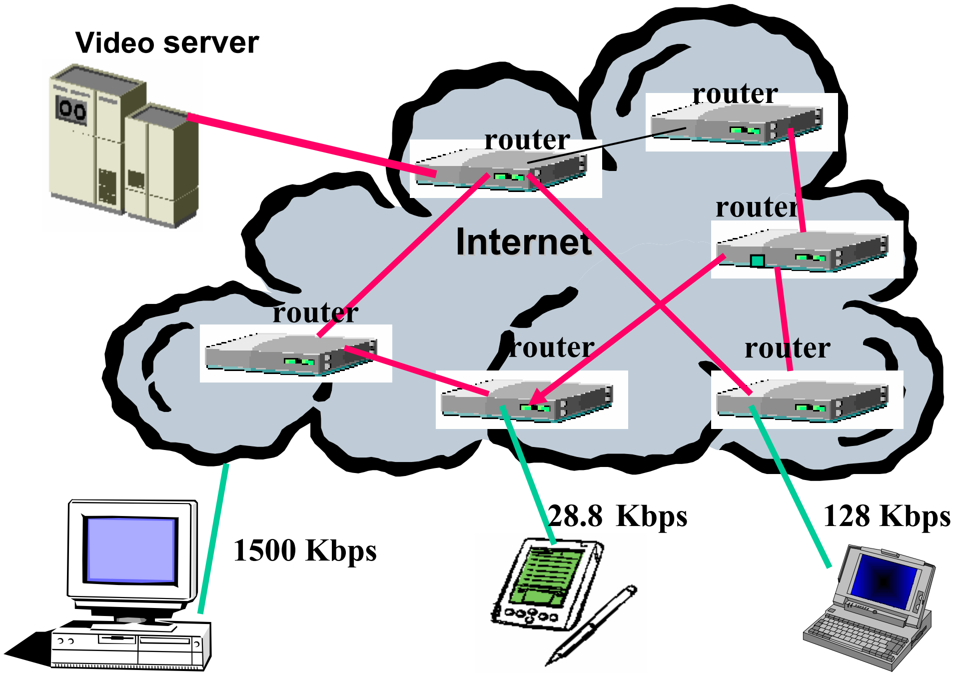


Figure 1-1 Diversity And Variability Of The Network

Since we seem to have enough bandwidth on the backbone, people are now focusing on streaming technologies. However, the backbone has not solved all problems of streaming technologies. The backbone bandwidth is the core bandwidth that has expanded at the speed of $O(2^n)$ every year. But the edge bandwidth expanded at the speed of $O(n)$. Table 1-3 shows the history of the edge bandwidth.

Standard	Rate	**Year**
V.32bis	14.4Kbps	1991
V.34	28.8Kbps	1995
V.90-v.92	56.6Kbps	1997~2001
Cable	300 Kbps to 1500 Kbps.	1995
DSL	1,544 Kbps to 512 Kbps	1998

Table 1-3 The history of the edge bandwidth

The unbalance expansion of the network bandwidth limited the quality of service of the Internet application, like network bottlenecks will cause the fail of data transfer. For example, like in figure 1.1, someone wants to watch TV via the Internet. A person first needs to hook his computer onto the Internet. He usually uses is a cable modem, a DSL modem, or a slow dial-up modem. Then, the modem needs to be connected to an Internet Service Provider (ISP) though the Local Area Network (LAN). Next, the LAN is connected to the Wide Area Network (WAN). After that, the WAN is connected to a high level WAN. After multi-levels of WAN, it is finally connected to the backbone. What is the speed of the WAN? T1. All in all WAN connects to another WAN, and finally, though all these network of networks, the computer connects to the backbone. On the other side of the system, the WebTV server also goes through such different levels of networks to reach the backbone. Remember, the backbone only has 11 Naps (Network Access Points). All levels of these networks have different speeds. For example, a typical cable or DSL modem has almost a rate of 1Mbps download speed a slow dial-up modem has a 56kbps download speed, and a typical WAN has a 1.54 Mbps bandwidth by using T1 or 48.08 Mbps by using T3. Since the different levels of networks exist with different bandwidths through the process, the effective transmission speed is determined by the lowest bandwidth period between the user's computer and the WebTV server. So the user is not guaranteed to see a good quality TV via the Internet, even if the backbone has reached 9.6 Gbps bandwidth. In fact, as the backbone and periphery technology are both pushing towards the upper limit of the network speed, the diversity in network capacity and the variability in the available quality of service among different parts of the Internet are also rapidly increasing, and the problems related to them concern researchers more. To solve these problems, a lot of research work has been focusing on increasing the bandwidth of whole networks. One example is how to run Ethernet at a blistering 10 gigabits per second. This Ethernet is not only 10 times faster, but is also more widespread than ever before. Companies like Nortel Networks have controlled releasing a 10G bit/sec Ethernet module. Others focus on changing network protocols. Some new protocols like RSVP [7] help reserve bandwidth for time-sensitive data being delivered in a timely fashion. All those works are focus on improving the traditional network. Unfortunately, even though we have the 10 gigabits Ethernet, the network bottleneck still exists. The protocol work is lengthy and difficult since it requires standardization and manual, backward compatible deployment.

On the other hand, some researchers focus on research to make systems operable across asymmetric network capacities. The DARPA research community created the concept of active network in 1994 [8,9]. Active network allows their users to inject customized programs into the nodes of the network. It is like replacing packets with “capsules” – program fragments that are executed at each network router/switch traverse.

Our research group tried some ways to allow the data stream be transferred on such an asymmetric network while avoiding the data to be blocked or broken. In this thesis, the research focuses on presenting a method to improve the performance of the network when video stream travels through the Internet. We present the case, by example, of a network aware adaptive video communication scheme. We use the active network concept to let the network node perform some computation on the video data flowing through them and let that video data meet the different network environment.

The video data communication is one of the Internet applications with the biggest demand. However, today’s existing video technology is ill-suited to cope with variations. For examples, in a video multicast distribution tree, if the receivers have various capacities, there are only three options for researchers to choose: 1) the server serving at a high resolution version which leads to the receivers with lower capacities being cut-off, 2) the server serving at a minimum resolution version focusing the high resolution client to be satisfied with the low resolution version despite their local capacities, or 3) the server serving with multiple versions of the stream resulting in the redundant information flow.

Our research group suggests a new scheme to achieve an optimum and intelligent result for difficult situation stated above. This scheme needs to solve two fundamental problems in video and network technology. One solution is to develop a video transmission technology that enables the dynamic adaptation. The other is to find a new model of networking where such adaptive units can be implanted right inside the network splice points where the networks and the links of varying characteristics meet in a global scale internet. Even though broadcasting video Internet is one of the first applications that require such adaptations, almost all the emerging network-based applications will require such adaptation ability.

Network researchers have developed the Active Network Backbone (ABone) [10] under DARPA (Defense Advanced Research Projects Agency) grants, which is suitable for adaptive units to be implanted and is being used for university research. Also, a video transmission technology, the Active Video Transcoder (AVT) technology, has been studied in our research group to enable implementing dynamic adaptations for a pre-encoded video stream to match the available bandwidth of heterogeneous networks. The operation of converting a pre-encoded video stream in a compressed format into another video stream, which is also in compressed format, is called Video Transcoding. Video Transcoding is an essential technique for video communications over heterogeneous networks and various client devices. A device that performs such an operation is called Video Transcoder. So, in turn, the Active Video Transcoder is such a device that works on the active network.

1.2 Research goal of this thesis

As discussed above, the research in this thesis is targeted towards the adaptive video communication schemes on the Internet. The Active Video Transcoder (AVT) technology is interpreted with MPEG2 [11] technology to develop intelligent schemes for efficient information communications. The Active Network is chosen to be our experiment platform. The research goal of this thesis is summarized as follows:

1. **Studying existing encode and decode schemes to build implanting methods for AVT employment**

   In this thesis, the MPEG2 encode and decode system is chosen for studying AVT technology. Because the MPEG2 technology has been popularly used in video information transfer on the Internet, developing AVT functions basis on MPEG2 system is not only a good way to prove the AVT idea, but also directly gives upgrading approaches to the current technology.

2. **Developing AVT schemes for the MPEG2 video**

   Bit rate reduction is a key technology to improve Internet video communications. Many approaches exist to achieve bit rate reduction:

   1) Reduce the number of the blocks in every macroblock [12]. This approach derives a new MPEG2 stream with half the spatial resolution from an original MPEG2 stream. In this approach, each macroblock (16 *16) of the original video is replaced by one block (8* 8) of the reduced-resolution video. This approach can reduce the bit rate, but can't scale it.

2) Frame Dropping [13]. In this approach, the transcoding has knowledge of the frame types (e.g. Intra- or Inter- frame coding type), and then dynamically drops frames according to the importance of the video frames. The dropping is either by the source (transcoding) or the network, depending on the level of network congestion.

3) Discarding high order DCT coefficients. [14] This approach is like a frequency filter. It performs operations on semi-uncompressed data. It operates in the frequency domain on the values of the DCT-coefficients. Semi-decompression/compression involves just entropy decoding/encoding. Low-pass filtering is where the higher frequency DCT-coefficients are discarded in recoding, leaving only the DC DCT-coefficient and a number of low-frequency components.

4) Suppressing color [14]. Color reduction operates on the chrominance information in the bit-stream. Switching color to monochrome removes all color information from a bit-stream. In a MPEG2 system, this approach is achieved by replacing each chrominance block by an empty block, and obviously, the color information is lost.

5) Re-quantization DCT [14]. In this approach, the re-quatization filter not only operates like a frequency filter on the DCT-coefficients but also dequantizes the coefficients. The coefficients are then re-quantized using a larger quantizer step. As quantization is the most inefficient process in the DCT based compression algorithms. The requantization may produce

some strange edge effects. However, the substantial bit-rate reduction can be achieved by reducing the spatial redundancy in this approach.

Because of the efficient bit rate reduction and obvious advantages of approach 5), two intelligent schemes, Symbiotic Rate Control technology and Focal Area Resolution Control technology, are proposed based on the re-quantization DCT in this thesis. They are expected to demonstrate the AVT technology in MPEG2 video communications on the Internet.

3. **Developing programs and performing numerical experiments to verify the Symbiotic Rate Control (SRC) scheme and the Focal Area Resolution Control (FARC) scheme for AVT.**

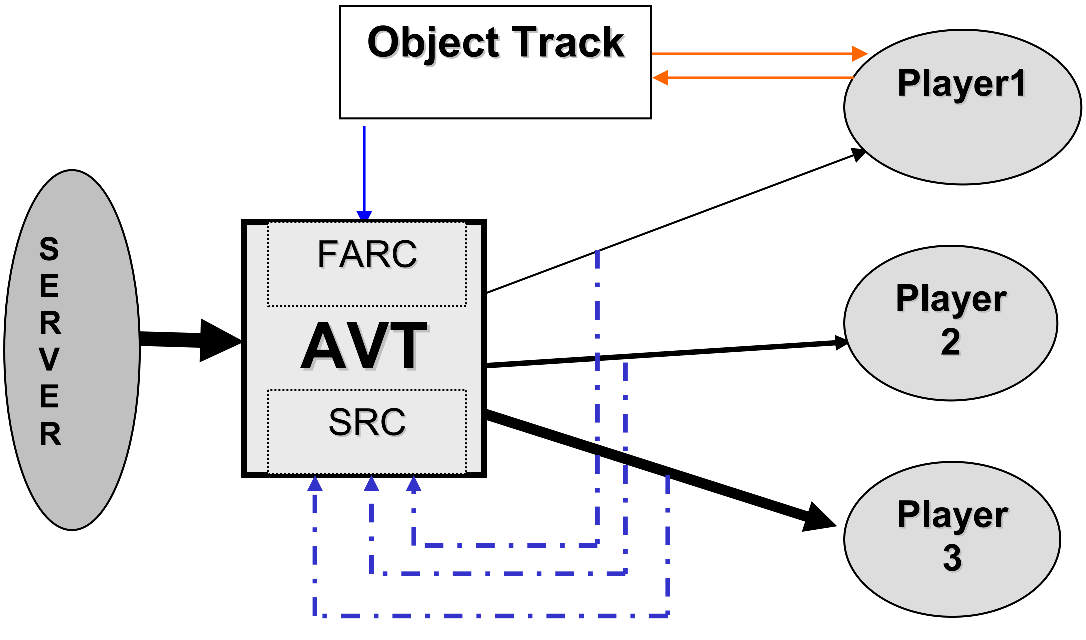


Figure 1-2 Diagram Of Video Transmission Through AVT

In this thesis, the author will focus on developing an AVT, which has the Symbiotic Rate Control and Focal Area Resolution Control. Symbiotic Rate Control (SRC) will detect the network available bandwidth and recompress the video stream if the video stream meets the network bottleneck. Since AVT recompress the video steam, it may sacrifice image quality. Focal Area Resolution Controller (FARC) is to enhance the image quality in an interesting area, such that, after recompressing, the video can be played at lower bit rate with good quality in focal areas.

Programs proposed in this thesis are to recognize the Symbiotic Rate Control scheme and the Focal Area Resolution Control scheme based on existing MPEG2 encode and decode programs. Numerical experiments are also proposed within these programs to test the bit rate reduction efficiency and also to verify the applicability of these technologies. These numerical experiments are also expected to give example approach of using the AVT technology and identify both advantages and drawbacks to give suggestions for future research work and commercial product development.

1.3 Organization of the thesis

In Chapter 2, MPEG2 systems and the idea of AVT technology is studied. After that, in Chapters 3 and 4, two of the most important functions of AVT, the Symbiotic Rate Control and the Focal Area Resolution Control schemes, are developed based on the MPEG2 system. The research result on the Symbiotic Rate Control (SRC) technology is presented in Chapter 3 with an intelligent scheme which can be used to adjust the output video bit rate according to the feedback of the available network bandwidth shift and buffer the video stream in case the network congestion occurs. In Chapter 4, a new and efficient scheme, the Focal Area Resolution Control (FARC), is studied to control the focal area resolution for the AVT system in which the bit resource is reallocated, and the video resolution of interesting areas enhances while that of another area is lowered. This scheme enables the user to get enough important video information in time even though the network bandwidth is limited. The numerical approaches for the SRC and the FARC are studied in Chapter 5, and the experiments and demonstrations of using these approaches are summarized in Chapter 6. These schemes prove to be very successful and useful. Lastly, he research work in this thesis is summarized in Chapter 7.

Chapter 2

ACTIVE VIDEO TRANSCODER

Traditional data networks passively transport data from the end of one system to another. Because the traditional networks are not sensitive to the data flowing, they are unable to be adjusted for the data transferring, according to feedback of the data flowing. The Active Networks studied in this thesis break the tradition by allowing the network node to perform customized computations on the user's data; then, it automatically adjusts the networks to transfer the information more efficiently with the same network limit capacity.

Because the MPEG2 is the most popular format for video communications, all theoretical and experimental results in this thesis are obtained specifically based on the MPEG2 format. However, similar schemes may be applicable to other video data formats.

2.1 Introduction of MPEG2

MPEG stands for Moving Picture Experts Groups [15,16]. This working group defines standards for digital compression of video and audio files. So the MPEG format is also a generic name of compactly representing digital video and audio signals for the consumer distribution. MPEG format is a DCT-based scheme with Huffman coding. The MPEG format uses the bit-stream to represent the video and audio signals, which takes less space, but retain good quality. An MPEG can achieve high quality videos with compression ratios over 100:1.

MPEG1 is the first standard for defining audio and video compression coding methods; it is also a multiplexing system for interleaving audio and video data so that they can be played back together. MPEG1 was designed to code progressively scanned videos at bit rates up to about 1.5 Mbit/s. It works for applications such as CD-i (compact disc interactive).

MPEG2 is an extension of MPEG1 and is directed at broadcast formats with higher data rates. MPEG2 defines the syntax for efficiently coding interlaced video, supports a wide range of bit rates, and provides for multi-channel surround sound coding.

MPEG2 is a generic video coding system supporting a diverse range of applications.

Here we outline the general structure of an encoder and decoder for MPEG2.

In the video system, MPEG uses luminance (Y) and two chrominance matrices (Cb and Cr) components to represent the red, green, and blue (RGB) signals in the color video. CCIR recommendation 601[17] defines how the component video signals can be sampled and digitized to form discrete *pixels*. There are three terms of formations, *4:4:4, 4:2:2* and *4:2:0.* They are often used to describe the sampling structure of the digital picture. For examples, *4:2:0* means the Cb and Cr matrices shall be one half size of the Y-matrix in both the horizontal and vertical dimensions.

Intra-frame Discrete Cosine Transform (DCT) coding and motion-compensated inter-frame prediction are used to remove the redundant information in the video, such as spatial and temporal redundancy and psycho-visual redundancy. Figure 2-1 shows the process of the MPEG2 video encoding. When a video is put in an encoder, the first thing done is Motion estimation. Motion Vector estimation is used to predict the frame to be coded from a reference frame. Predictions can be made both for forward and backward frames. The prediction is for the macroblock of the same spatial position from the reference picture. Most of the time, however, the motion has occurred between the macroblock being coded and the matching macroblock in the reference frame. A prediction is made to select the matching macroblock that has the minimum error with the macroblock currently being coded. The motion vector is a two-dimensional vector used for motion compensation that provides an offset from the coded macroblock to the matching macroblock. Each Predictive-coded picture (P picture) or bi-directionally predictive –coded picture (B picture) will generate a motion vector in each direction per macroblock.

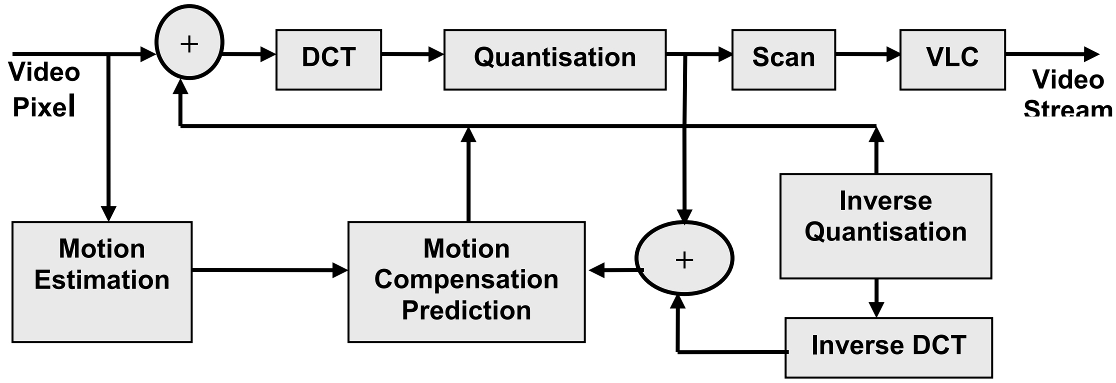


Figure 2-1 MPEG2 Video Encoder

Motion compensation uses the motion vectors to improve the efficiency of the prediction of the sample values. In the encoder, it subtracts the motion-vector prediction from the source picture to form a prediction error picture.

Then the prediction error picture is sent to the DCT. DCT in the MPEG2 coder is two-dimensional and work on 8 pixels by 8 lines block. DCT generates 64 coefficients with each of them having 11 bits. Since most of the high-frequency coefficients are near zero, the encoder will not transmit them. The encoder only quantizes and codes the remaining coefficients. Thus, the encoder accomplishes the bit rate reduction.

Quantization is used for reducing the number of bits for coefficients and reduces the bits to be transmitted. The quantizer degree used for each coefficient is decided by the quantization parameter, the quantizer_scale. Then, the coder goes through a scan. The “Zig-Zag scan” is used for both MPEG1 and MPEG2 to change the 8×8 matrix pattern coefficients to a list and ordered by the scan pattern. There is another “alternate scan” that can be used for MPEG2. In the Variable Length Code (VLC), the list of the values get from scanning will be entropy. Every VLC is a run of zeros followed by a non-zero coefficient. The VLC can be obtained from the Variable Length Code Table.

Now the output coded bit-stream, including the coded luminance, chrominance prediction error, and the support information such as the motion vectors and synchronizing information, are ready for transmission. Figure 2-2 shows the structure of the decoder.

In the decoder, when the video stream data arrived, they are first put into the Variable

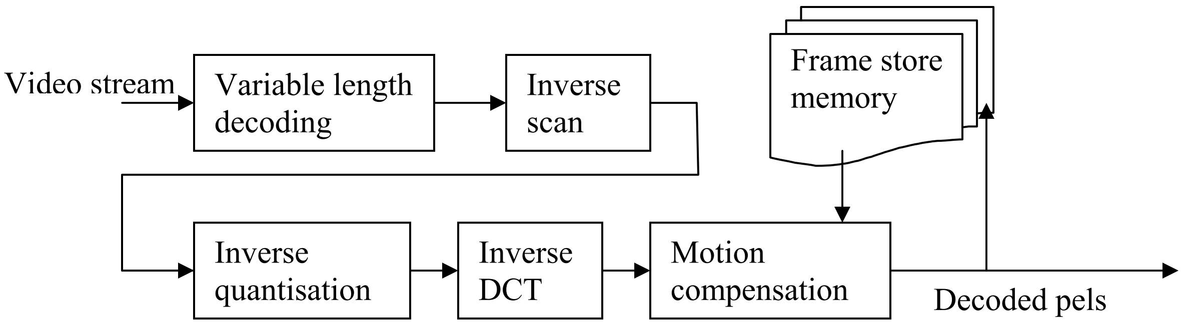


Figure 2-2 MPEG2 Video Decoder

Length Decoding (VLD). The supporting information and the list of the coefficients are then obtained. And then, the Inverse Scan and the Inverse Quantization are used to reconstruct the coefficients' matrix. The inverse DCT is used to do inverse transformation to produce the prediction error. The prediction error is added to the motion compensation prediction, which is obtained from the decoded pictures to reconstruct the decoded output picture. In such a process, the video bit-stream that is transferred across a constant rate channel is able to retain all the information of the video.

Figure 2-3 shows the detail structure of the MPEG2 video bit-stream. It includes the the information of the sequence, group of the pictures (GOP), slice, and Macroblock.

In MPEG2 format, the video information is coded in a bit-stream sequence. All the pictures are divided into groups, called the GOP (Group of picture). A structure of GOP can be controlled by two parameters which are N, the number of the pictures in a GOP, and M, the I/P frame distance. There are three types of pictures in a GOP:

1. I-picture (Intra-coded picture) is coded without reference to other pictures, and only uses information from itself.
2. P-picture (Predictive-coded picture) is coded using the previous I- or P-picture for motion compensation.

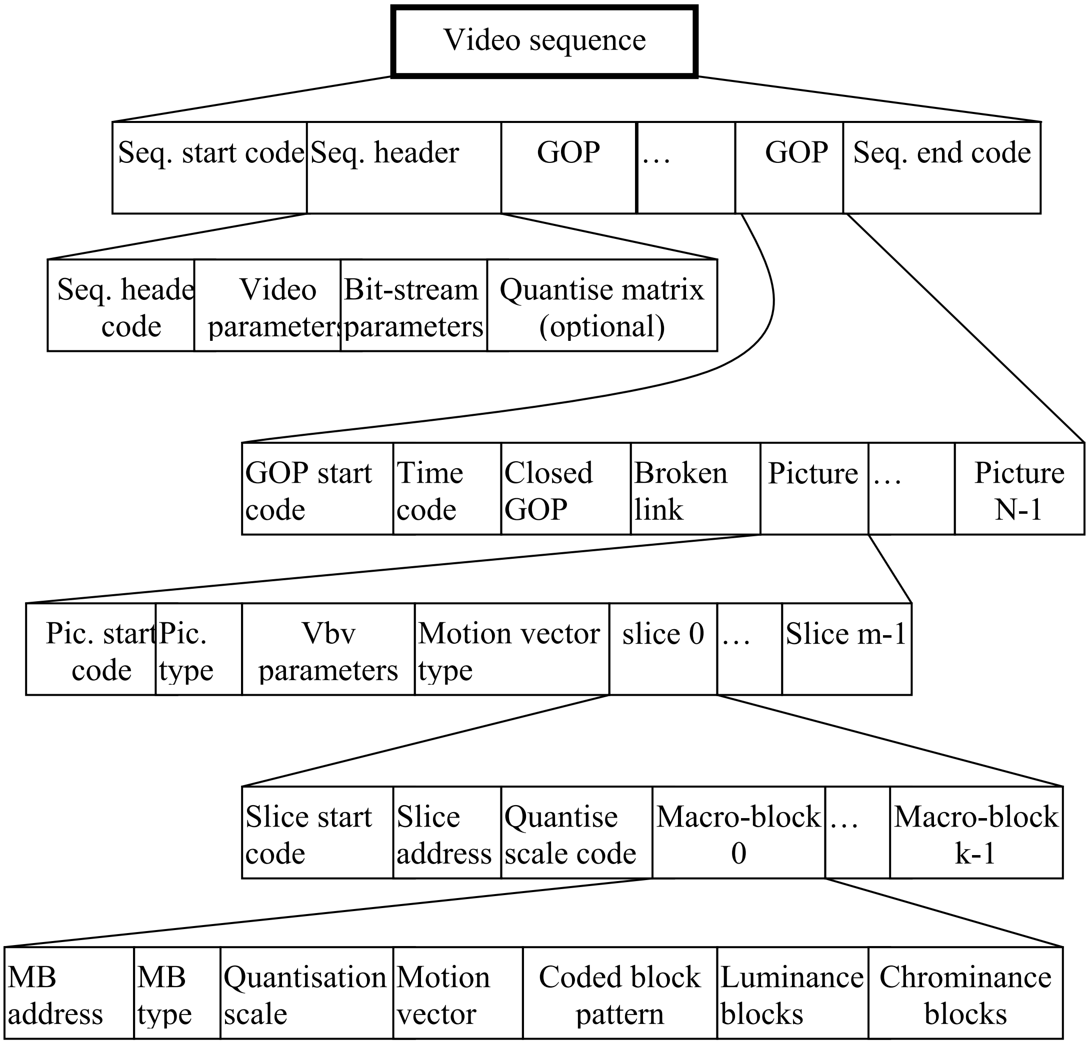


Figure 2-3 Structure Of The MPEG2 Video Sequence

3. B-picture (Bi-directionally predictive –coded picture) is coded using the previous or next I- or P-picture for motion compensation.

In a GOP, the picture order in the display is different from that in the bit-stream. Table 2-1 shows an example video where N=9 and M=3.

The picture order in display:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P

The picture order in bit-stream:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
I	P	B	B	P	B	B	I	B	B	P	B	B	P	B	B

Table 2-1 Display and Encoded Picture Order Detail

In the case that the reference picture is totally different from the picture being coded, the GOP structure may not be used. The picture will then be coded as an I-picture. In general, a B-picture needs the least bits in the bit-stream, a P-picture needs twice more bits than a B-picture, and an I-picture needs six times more than a B-picture.

2.2 Active Network

The concept of the active network emerged from DARPA (Defense Advanced Research Projects Agency) funded research projects. The main idea of the active network is to allow the individual or group users to inject customized programs into the nodes of the network. The "Active" architectures enable a massive increase of the complexity and customization of the computation that is performed within the network, e.g., that is interposed between the communicating end points [18].

Why is the Active Network needed? Many new network applications emerge rapidly and rely on the new network services that can accommodate their application modes better. The demand for these kinds of new services in the network is huge. For example, the demands for the following applications are obvious:

- Multimedia applications. As we have discussed in chapter 1, those applications need new services to overcome the network environment limitation.
- The wireless data transmission between laptops and networks. The mobile IP [19] needs a network-level routing service.

Furthermore, to increase the network performance for any kind of application, network node caching and load distribution are both required.

Either deploying the new network services at end-systems or implementing these services at the nodes interior, or at the network layer can offer better functionality and performance than ever before. However, in either case, to deploy most of these new services in the traditional network is difficult because changes are need in the existing network protocols. Active networks are studied to solve this problem. The active network is “active” in two ways:

- The Routers and switches can perform computations on the data flowing through them.
- The User can inject their programs into the nodes and enable the nodes to process specific functions for the user or the application.

Researchers have spent great effort and more than 10 years of time on the research of the active network architecture.

In this thesis, we discuss an approach called AVT that can be used in the active network node to control the MPEG2 video streaming rate. This approach is motivated by both a lead user application, which perform user-driven computations of nodes within the network, and the emergence of mobile code technologies that make dynamic network service innovation attainable.

2.3 Active Video Transcoder System

Video rate transcoding is one of the key technologies in implementing dynamic adaptation of the bit rate of a pre-encoded video stream to the available bandwidth over heterogeneous networks.

Active Video Transcoder is such software that can transmit an MPEG2 video stream over a heterogeneous network. Since this video transcoder is designed to sit in the active network node, it is called Active Video Transcoder (AVT). The AVT is the optimal solution of video stream transfer on heterogeneous networks.

For example, an MPEG2 video has a 9 Mbits/s output rate and store in a server. A user wants to watch this video on their desktop. The server should transmit this video stream to the user's desktop. The server transmits this video stream through an OC-192 line to a cable-head end. However, due to limited cable capacity, the cable-head end has to relay this incoming video onto a cable channel at a lower bit-rate, say 5Mbits/s, which is also in a compressed MPEG2 form. To achieve this, the head end uses AVT on the input video stream to lower its bit rate so that the video bit-stream can be transmitted via the cable channel. In fact, besides bit-rate adaptation, an AVT can dynamically change any

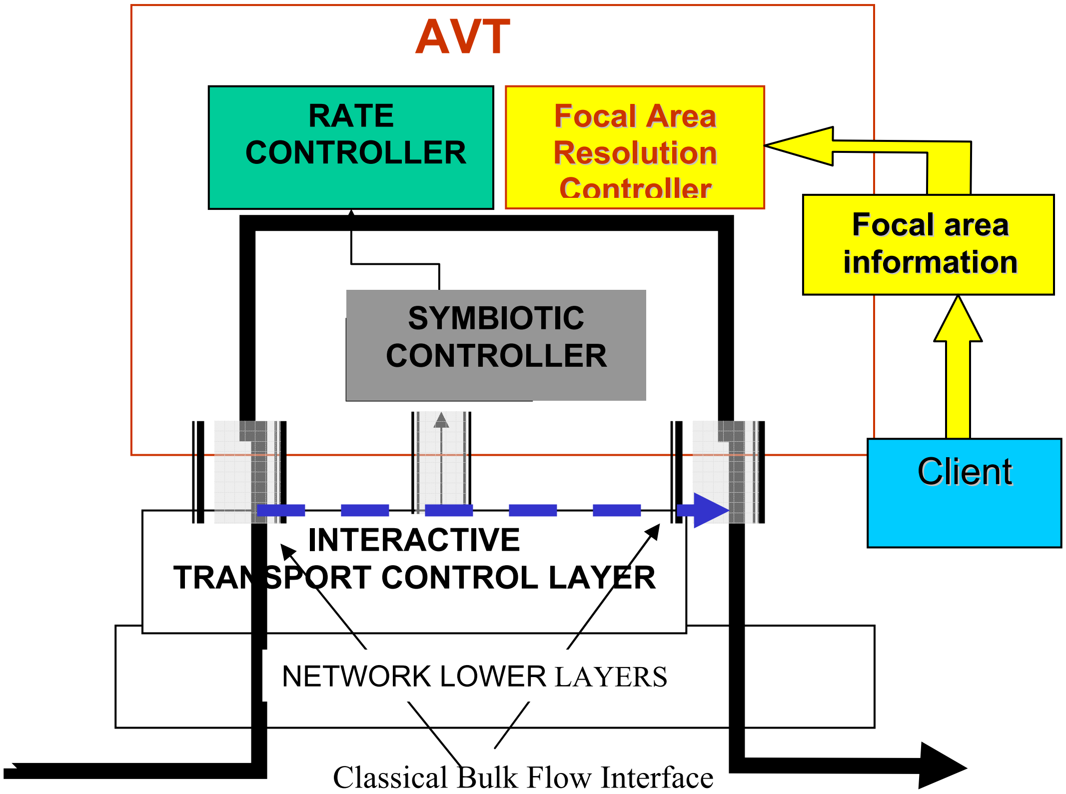


Figure 2-4 Concept Diagram Of An AVT MPEG2 Transcoder

coding parameters of a compressed video, including frame-rate, spatial resolution, video content and/or coding standards used. Figure 2-4 shows an example of aninteractive transport system and AVT. This interactive transport system can be placed in a suitable network junction point, which intercepts the video stream.

This System includes two parts:

1. Transport Control:

Even though the AVT is the main part of the system, it has to sit on top of an interactive transport control layer—TCP Interactive. Unlike conventional TCPs, this novel transport layer resizes an event to the application when there is an internal timer-out event, which is passing on the current window. The interface is almost identical to the TCP classic, except upon opening the socket, the application binds an interruption handler routine to the designated socket event. When the event occurs, the TCP triggers the handler. The binding is optional. If the application chooses not to bind any handler, the system defaults to silent mode identical and then to TCP classic.

2. Internal architecture of AVT

The AVT unit has a decoder and a re-encoder. The re-encoder uses a feedback rate control mechanism which is capable of working in two modes: normal mode and frugal mode. In frugal mode the rate can be controlled at frame level. The actual control signal that is being sent to the rate controller is generated by an application unit called *symbiosis controller*. The symbiosis controller accepts signal inputs from the transport layer to generate the symbiosis. The rate controller will re-compress the video stream with this symbiosis. (Details will be discussed in depth later in chapter 3.)

On the other side of this process, the client sends some feedback of focal area information to the AVT. The focal area resolution controller collects this information and reallocates the bits on a frame to let the focal area have the higher resolution than the rest area. Because symbiotic rate controller may sacrifice image quality, Focal Area Resolution Control is studied to enhance the image quality of an interesting area in a picture frame, like a people or a car, such that the video can be played at a lower bit rate with good quality in focal areas.(Details will be discussed later in depth in chapter 4.)

A system like this has several advantages compared to implementing the rate adaptation at the end-point (encoder). It subsumes the functionalities of a server-client model. In addition to this, it allows rate adaptation on a video stream that is already encoded and thus enables to transfer stored video at a dynamically selected rate. This decoupling also has the benefit that the AVT can sense local asymmetry network link capacities-rephrase and can be dynamically deployed inside a network. For example, it can sit at a node splicing fiber and a wireless network, and thus can downscale an incoming high-bandwidth video multicast stream for outgoing low-capacity wireless links. Additionally, there also exists the possibility of bring down AVT operation inside a network by emerging technology such as active networking.

With the capability of dynamically altering coding parameters of a compressed video, the AVT is expected to become core technology for universal multimedia access by

Internet users with different access links and devices. Figure 1-1 shows the role of an AVT in such environments.

The Internet is a collection of heterogeneous networks and different users may have different access to it. LAN access bandwidth usually is in the order of mega-bits/s. DSL and Cable provides access bandwidth from several hundred-kilo bits/s to a small number of mega-bits/s. ISDN users have 64k or 128k bits/s access to the Internet. Traditional telephone modem users have a bandwidth range from 14.4k to 56k bits/s. The access bit-rate for cellular phones currently is only a few kilo bits/s. In addition, these links have different channel characteristics. Also, devices used by people for Internet access are becoming more and more diversified. Presently, personal computers are the major Internet access devices. However, in the future, network appliances (or information appliances) including handheld computers, personal digital assistants (PDA’s), set-top boxes, screen telephones, smart cellular phones and network computers, are expected to replace personal computers to become the dominant access terminals for accessing the Internet. These network terminals (including PC and network appliances) have various computing power and display capabilities. Also, users’ interests in content may differ from one another. For multimedia data (including and not limited to compressed video) to be smartly delivered to users with different available resources, access links, and interests, the multimedia content must be adapted dynamically according to the different requirements of different users. AVT will be one of the key technologies to fulfill this challenging task.

Figure 2-5 shows the concept of a video transmission system for MPEG2. The input video goes through an encoder device (EL) becomes a MPEG2 stream that has a bit rate "R1". The AVT sits on the network node converting the video stream into streams that have different bit rate "R2". Then, the video stream arrives to the user's end with a decoder device (DL), and plays the video on the screen. Since the AVT is based on the combination of a decoder (D2) and encoder (E2), all the information of the video stream needs to pass from D2 to E2. Inner pipes exist to transmit the information between D2 and E1. The working condition can be obtained by the symbiotic controller, and then it will be sent to the re-encoder E2 in the form of the parameter file.

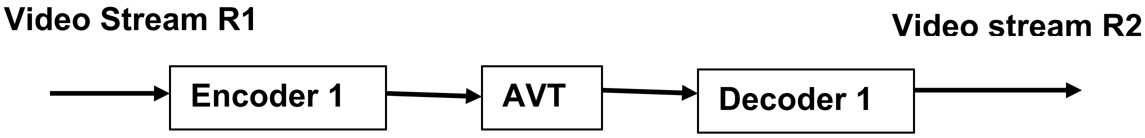


Figure 2-5 The Concept Of Video Transmission System

Chapter 3

SYMBIOTIC RATE CONTROL

In chapter two, the AVT technology has been discussed to resolve the network bottleneck or congestion . This chapter will focus on the symbiotic rate control, which is one of the most important components of the AVT, specifically used to solve the congestion problem via controlling the bit rate.

The symbiotic rate control is developed to dynamically and intelligently adjust the bit rate of video data transmission in the active network, which consists of two parts: the symbiosis controller and the rate controller.

The symbiosis controller uses a mechanism that controls the actual values of the rate dynamics for the symbiotic rate control. The main parameter of the rate controller -- target bit-rate is given by the symbiosis controller. The control obtains the target bit-rate by closely mimicking the rate provided by the underlying transport layer. The symbiotic network envisions an impairment management approach that encourages direct feedback from the transport layer on the impairment. The target bit-rate is then sent to the rate control dynamically.

The rate controller is the component that recompresses the video stream and controls the output stream bit rate.

In this chapter, a network video stream transmission system in the context of a MPEG2 video transcoder is studied. It can sense the change of the network environment and adapt to changes by adjusting the transmission bit rate. The model has been developed by closely following the general MPEG2 TM-5 (Test Model 5) [20]. The MPEG2 TM-5 is the latest important upgrade of the Test Model document, which was formed at Sydney, Australia meeting of the MPEG working group (WG11) in March of 1993. The MPEG2 Test Model (TM) has been serving as a cookbook for creating bit-streams during the collaborative co-experimental phase of the MPEG2 video. Each TM participant would follow the recipe given in the TM document to create an encoder. If the proposal met the criteria (coding gain, implementation complexity, robustness), then it survived. The mechanism of the MPEG2 symbiotic rate control is shown in figure 3-1.

When the MPEG2 video stream is transmitting on the active network, the transport layer will send the feedback to the symbiosis controller, which sits on the network node to tell the current network bandwidth situation. If the network is in its normal condition, the symbiosis controller will not turn on the rate controller and the MPEG2 video stream will directly go through the network node without any change. If the network bandwidth is less than this video stream needed, the transport layer will send the feedback signal to the symbiosis controller. The symbiosis controller will turn on the rate controller and generate a new target bit-rate to send to the rate controller. The symbiosis controller then redirects the video stream to the rate controller and the rate controller use the new target bit rate to recompress the video stream. The rate controller first decodes the video

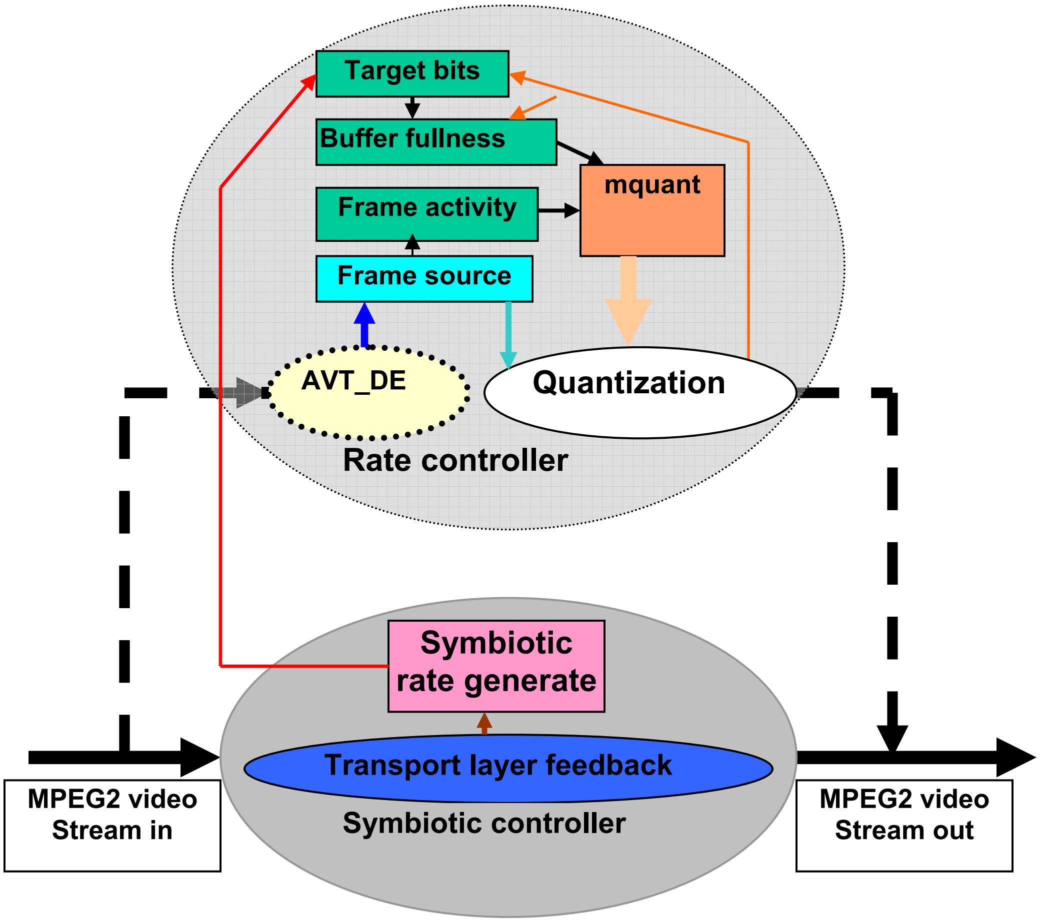


Figure 3-1 Diagram of Symbiotic Rate control

stream and then the new target bit-rate is used to adjust the target bits for GOP and every frame and buffer fullness. Finally, the quantization step parameter '*mquant*' will be changed. The output bits for the video stream will be reduced.

3.1 Symbiotic controller

In this thesis, the rate adaptation is focused on more, while the symbiosis controller is only in the schematic research. A simple symbiosis controller is illustrated in the following to show its functionality. This has been used in an experiment of the paper *Symbiotic Video Streaming by Transport feedback based Quality Rate Selection.*

In the current existing network, only end to end technology is implemented to sense the network congestion situation [21,22,23]. Within this technology, specific protocol, such as TCP sliding window protocol [24,25], is used to communicate between the sides of the clients and the server of the network. For example, when the client feels congestion occurring (time out), it sends a request to the server to reduce the bit rate by half. If it is still not working, it sends another request to reduce the bit rate by half more and keeps doing so until the traffic passes through.

In our approach, the symbiosis controller is designed to sit on the network node to obtain the feedback information from the clients, analyze the information, and dynamically produce bit rate adjustment information to the rate controller so that it will adjust the bit rate in the network node nearest to the client. By doing this, the local network congestion is resolved while the server side network is not affected.

In our AVT design, the only input of the rate controller is the *target bit rate* c_T; so the function of the symbiosis controller is to produce the *target bit rate* c_T according to the rate read from the video stream and the feedback bandwidth information from the TCP implementation. Though current versions of TCP implementations fine-tune many of the response characteristics, in our approach, the symbiosis is designed to respond primarily to a timeout event. The parameter ξ is defined to describe the time out event.

$\xi = 1$ time out event occur;
$\xi \neq 1$ normal situation.

If the target bit rate during normal mode generation is given by $c_{\max}$, when a time out event occurs ($\xi = 1$), bit rate adjustment is needed. The subscriber rate needs to retract to a smaller amount but yet retain a nonzero quantity c_{min}. This point is defined by a parameter *rate retraction ratio* ρ,

$$\rho = \frac{c_{\min}}{c_{\max}} \quad ,$$

where c_{min} is determined by the condition that with c_{min} . Also, based on the specific video instance and the tolerance of quality level, the system should still be able to generate a video, of course, with lesser quality.

For symbiosis with the underlying TCP, a running generation threshold function is defined in the following display:

$$\begin{aligned} c_T(t) &= \frac{1}{2}c(t-1)\,, && \text{when} \quad \xi = 1 \\ &= c_T(t-1), && \text{otherwise.} \end{aligned} \tag{3.1}$$

It retracts to half of its current size when a fault occurs. The running control function c(t) is then given by:

$$
\begin{aligned}
c(t) &= \rho \cdot c_{\max}\ , && \text{when } \xi = 1 \\
&= 2 \cdot c(t-1)\ , && \text{when } \xi \neq 1 \text{ and } c(t) \geq \frac{1}{2} c_T(t-1) \\
&= \min[C_{\max}, c(t-1)+1]\ , && \text{when } \xi \neq 1 \quad \text{and } c(t-1) \geq c_T(t-1)
\end{aligned}
\tag{3.2}
$$

The control function performs binary-exponential-backoff within the limits given by generation parameters ρ and normal mode target bit-rate c_{max}. When the system enters the frugal state $S(t) = 1$, the symbiosis keeps reducing the target bit-rate and ξ turns to a number other than 1 Then, a loss occurs and $\xi = 1$ again, but the system stays in the frugal state until the control (target bit-rate) recovers back to the normal target bit-rate. In the frugal state, the updated target bit-rate is sent to the rate controller dynamically.

3.2 Overview of the MPEG2 rate control

As discussed in chapter two, the output video stream bits that represent the video information is represented by parameter bit_rate, which is the bits per second that the encoder puts out and usually, a constant number within that encoder setup in the parameter file. The encoder uses this bit_rate to allocate the target bits for every frame. Since the encoder acquires all the parameters before it starts encoding the MPEG2 video and all the parameters cannot be changed in the period of encoding, in a conventional network system the output bit-stream will keep an unchanged constant rate--bit_rate. However, in our AVT design, the output video bit-stream rate has to be changed according to the networking condition.

How can the AVT change the output bit-stream to meet the networking bandwidth?

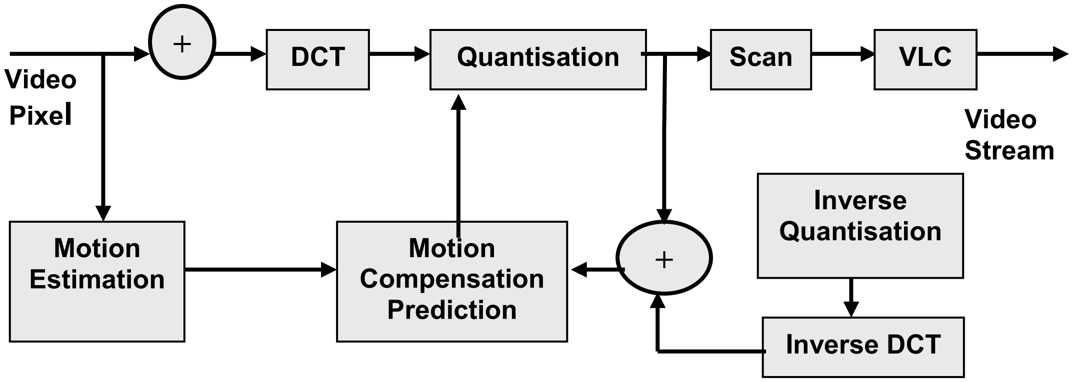


Figure 3-2 MPEG2 Video Encoder

From the structure of MPEG2 encoder shown in Figure 3-2, it is learned that the output bit-stream may be controlled through two modules: the *variable length coding* (VLC) and the *Quantization*. However, because of the following four reasons, the *Quantization* is chosen to control the output bit-stream. First, the frame orpicture is divided into macroblocks in an MPEG2. It is not possible to predict the exact amount of bits that will be produced from a macroblock for a given choice of coding parameters in the VLC step, although it is the step to generate bits for the video output. Second, the perceptual content and activity in a particular picture area also dictates the inherent amount of bits that may be required to encode it, but the VLC step does not know anything about that. Third, the bit requirement per macroblock depends on the picture type (I, B or P) as well other subjective factors; and all these factors are use within the *Quantization* step. Fourth, a proposed mechanism--- the feedback control mechanism- is set up at the *Quantization* step, where the output bit-rate is continually sensed to determine the overall piecewise constant rate with appropriate accounting for variations in a frame/picture type like TM-5.

How does the *quantization* step control the output bit-stream in the TM5? As shown in Figure 3-2, after *motion estimation* and *compensation*, the prediction errors for each 8x8 blocks are computed. These 64 pixel differences are then transformed into 64 DCT coefficients. Each DCT coefficient has its own quantization level due to the human visual system responding differently to the distortion in various DCT coefficients. To control the overall output bit rate, MPEG2 in its linear quantization mode uses a scale factor called *mquant* to determine the actual quantization levels that are applied to these DCT coefficients. The quantized outputs for intra and non-intra frames are respectively given by:

$$y = \frac{f(x, quant\_step) + .75 \times mquant}{2 \times mquant}, \qquad y = \frac{16 \times f(x, quant\_step)}{mquant}, \tag{3.3}$$

where x is the DCT coefficient, $y=f(x, quant\_step)$ is determined by ISO/IEC 13818-2 tables [7]. As *mquant* increases, the effective quantization steps become larger, more information is lost, encoding requires lower bits, and the quality of the picture degrades Also, the vice versa may occur. In this way, *mquant* is used as a major parameter to control the out put bit-rate.

3.3 Rate controller design basis on MPEG2 TM5 encoder

According the MPEG2 Test Model5 (TM), the output bit-rate is controlled by the quantization-step. There are three major steps to determine the *mquant*, which are shown in different colors in Figure 3-3.

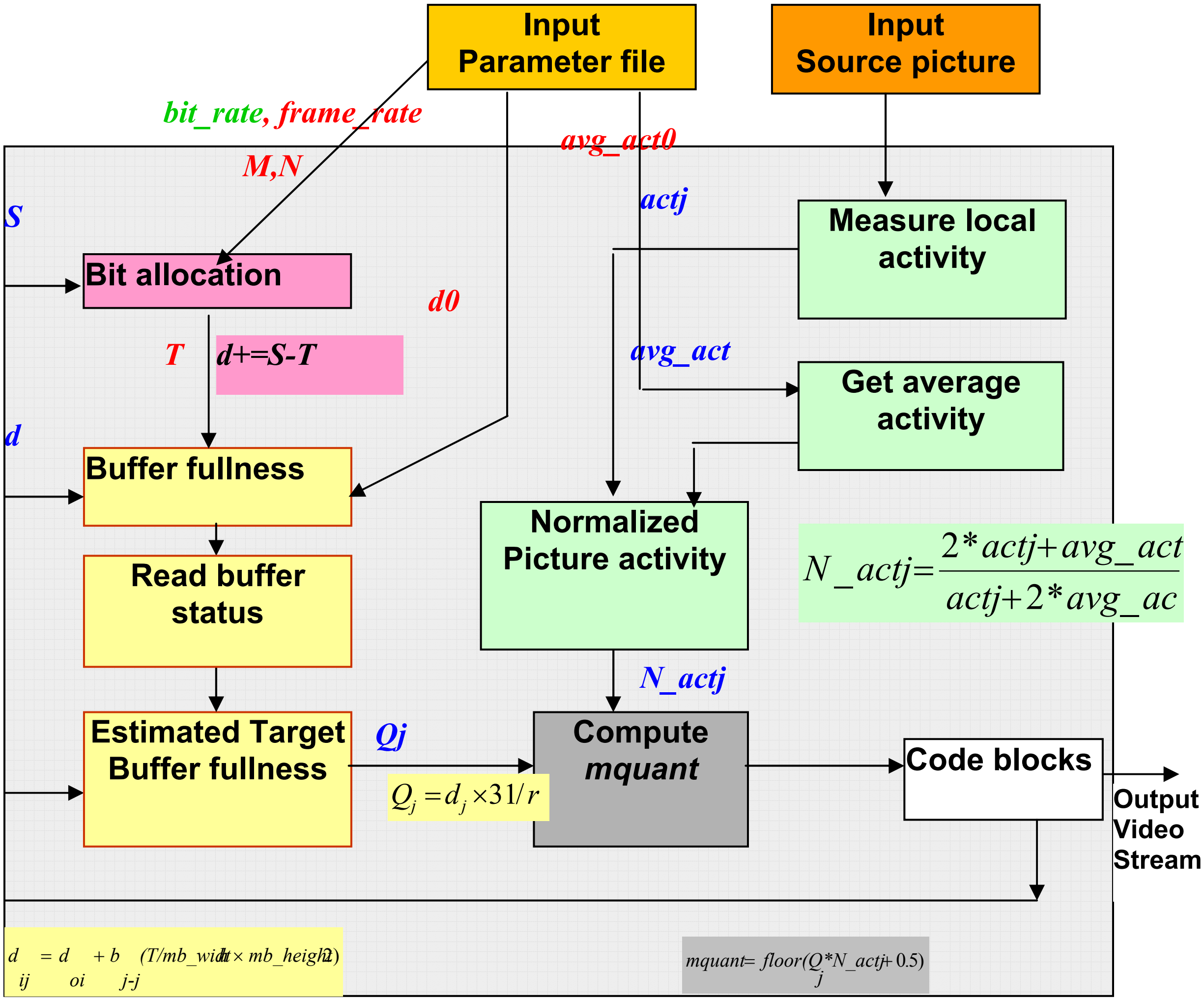


Figure 3-3 Rate Control Diagram

Step 1- Target bit allocation: this step estimates the number of bits available to code the next picture. It is performed before coding the picture.

When the encoder starts to encode a video stream, it first estimates the number of bits which will be allocated to a GOP (Group of Pictures). We assume there is a total of *R* bits which will be allocated to a GOP before encoding.

$$R = N \times bit\_rate / frame\_rate \tag{3.4}$$

Where *N* is the number of frames in the GOP, *bit_rate* is the rate at which bits output from the encoder per minute, *frame_rate* (the rate at which frames output from the encoder per minute). [These three parameters can get from parameter file – must rephrase because you use parameter twice. Be specific.]. The parameter file includes all the information about the video and MPEG2 stream.The encoder will then allocate these bits to every frame in this GOP. In the MPEG2 bit- stream, there are three type of pictures (I,P,B) in a GOP. As we mentioned in chapter two, of these three types of pictures, the I picture has the most information of the video, then it is followed by the P and B pictures, respectively. So, the MPEG2 has a *Complexity Estimation* for each of them. The global complexity measures assigned relative weights to each picture type. These weights (Xi, Xp, Xb) are reflected by the typical coded frame size of the I, P, and B pictures. I-pictures are assigned the largest weight since they have the greatest stability factor in an image sequence. B-pictures are assigned the smallest weight since

B data does not propagate into other frames through the prediction process. The weights are set in the parameter file, but they usually are computed by the encoder.

The initial values for the weights are:

$$X_i = (160 \times bit\_rate)/115.0, \tag{3.5}$$

$$X_p = (60 \times bit\_rate)/115.0, \tag{3.6}$$

$$X_b = (42 \times bit\_rate)/115.0, \tag{3.7}$$

So the target bits for these three types of pictures are:

$$T_i = \max\left\{ \frac{R}{1 + \frac{N_p \times X_p}{X_i \times K_p} + \frac{N_b \times X_b}{X_i \times K_b}}, \frac{bit\_rate}{8 \times framr\_rate} \right\}, \tag{3.8}$$

$$T_p = \max\left\{ \frac{R}{N_p + \frac{N_b \times K_p \times X_b}{X_p \times K_b}}, \frac{bit\_rate}{8 \times framr\_rate} \right\}, \tag{3.9}$$

$$T_b = \max\left\{ \frac{R}{N_b + \frac{N_p \times K_b \times X_p}{X_b \times K_p}}, \frac{bit\_rate}{8 \times framr\_rate} \right\}, \tag{3.10}$$

Where R is the number of the bits in GOP, T_i , T_p, T_b are the target bits for I, P, B pictures respectively. X_I, X_p, X_b are the global complexity measured weights for them. N_p *and* N_b are number of *P and B* pictures in the GOP. K_p *and* K_b are "universal" constants dependent on the quantization matrices. In the TM5, it is set to 1.0 and 1.4.

After a frame is encoded, **R** is updated to the remaining number of bits that have been allocated to the GOP:

$$R = R - S \tag{3.11}$$

S is the number of actual bits generated for the frame just encoded.

The global complexity weight for the picture is:

$$X = S * Q \tag{3.12}$$

Q is the average quantization for all macroblocks in the frame, and the computation of Q will be discussed in detail in next step.

The picture target settings allocate target bits for a frame based on the frame type, the remaining number of bits assigned to the GOP, and the remaining number of frames of that type in the Group of Pictures (GOP).

Step 2- Rate control via Buffer Monitoring: by means of a "virtual buffer", this step sets the reference value of the quantization parameter for each macroblock.

Rate control attempts to adjust bits allocation if there is a significant difference between the target bits (anticipated bits) and the actual coded bits for a block of data. The rate control in the encoder adjusts the bit allocation if the coded bits are different from the target bits. As shown in Figure 3-3, the virtual buffer status will affect the macroblock quantization step size to meet the target bits for the GOP.

The quantization parameter Q_j for macroblock j is given as:

$$Q_j = d_j \times 31/r \tag{3.13}$$

d_j is the appropriate virtual buffer fullness and$r$ is the reaction parameter.

And,

$$r = 2 \times \frac{bit\_rate}{frame\_rate}, \tag{3.14}$$

For macroblock j, the buffer [fullness-clarify] before being encoded is:

$$d_j = d_0 + B_{j-1} - \left(\frac{(j-1) \times T_j}{mb\_number} \right), \tag{3.15}$$

B_j is the total number of bits that the encoder has generated for the last macroblocks. T_j is the target bits for this frame. d_0 is the initial [fullness-rephrase] of the buffer. It has three values for the three types of frames. The initial value is:

$$d_0^i = 10 \times \frac{r}{31}, \tag{3.16}$$

$$d_0^p = K_p \times d_0^i, \tag{3.17}$$

$$d_0^b = K_b \times d_0^i, \tag{3.18}$$

Step 3- Adaptive Quantization: this step modulates the reference value of the quantization parameter according to the spatial activity in the macroblock to derive the value of the quantization parameter, mquant, that is used to quantize the macroblock.

The macroblock quantization is affected by the local activity measurement. If the local activity is high, it will increase *mquant* step size to decrease the bit allocation. It will adjust the bits between the same types of the pictures that have the different local activity measurements.

The computation formula for the normalized activity for the macroblock is given as:

$$N\_act_j = \frac{2 \cdot act_j + avg\_act}{act_j + 2 \cdot avg\_act} \quad , \tag{3.19}$$

act_j are pixel values, and they are given by:

$$act_j = 1 + \min(vblk_1, vblk_2, \ldots, vblk_8), \tag{3.20}$$

and

$$vblk_n = \frac{1}{64} \times \sum_{k=1}^{64} (P_k^n - P\_mean_n)^2 , \tag{3.21}$$

$$P\_mean_n = \frac{1}{64} \times \sum_{k=1}^{64} P_k^n , \tag{3.22}$$

P_k^n is the sample value in the *n*th original 8×8 block. *avg_act* is the average value of act_j of the last frame to be encoded The initial value of the first frame is 400.

Finally, it follows that:

$$mquant_j = Q_j \times N\_act_j , \tag{3.23}$$

where Q_j is the reference quantization obtained in Step 2.

The effect of this step is to roughly assign a constant number of bits per macroblock (this results in a more perceptually uniform picture quality).

Since N_act_j is determined by the video itself, Q_j is the parameter we use to determine how the allocation of frame-bits varies.

Chapter 4

FOCAL AREA RESOLUTION CONTROL

In Chapter three, the symbiotic rate control was studied. The symbiotic rate control can adapt with respect to the local network resources at the junction nodes and can downscale the MPEG2 stream rate to transmit video streams over heterogeneous networks. However, the symbiotic rate control has to sacrifice image quality. In order to using bandwidth efficiently, another AVT, the Focal Area Resolution Control (FARC), is studied in this chapter to enhance the image quality in the areas of interest, such that, the video can be transmitted and played at a lower bit rate but still preserve good image quality in focal areas.

4.1 Fundamentals of the Focal Area Resolution Control

Usually when people watch TV, they are not able to focus on the whole screen in an instant. Instead, they pay attention to some perceptually interesting spots. If you see a movie, you will be more interested in the actors than on the background. If you are a doctor and are checking an MRI video, you will pay more attention to the spot that has the specific pathological changes you are looking for. So, not all the areas on the video frame hold the same amount of interest and importance for each human. Usually, human eyes only focus on the parts of a frame that is more interesting to the individual.

This habit of human eyes, which is mentioned above, makes it possible to enhance the visual effect by increasing the resolution in the focal area dynamically and reducing the resolution in the other areas while keeping total bits of video unchanged. On the other hand, if an AVT has to reduce the bit-rate of a video stream to fit the networking situation, it can reduce the resolution of the unimportant areas to reduce the bit of the video stream in a particular range, and thus keep enough resolution for the important area. With this, the bit-rate of the video stream is reduced while the visual effect is maintained. In such an approach, the AVT usually can implement several different levels of the resolution for a frame according to the levels of importance of the areas.

4.2 Approach of the Focal Area Resolution Control for MPEP2 TM5

In chapter three, the rate control is done on the frame level. This means that when the AVT codes a frame, it will check the symbiosis controller and adjust the output bits for every frame to meet the bandwidth of the network. In this chapter, the Focal Area Resolution Control (FARC) is working on macroblock levels. It maps the focal area to corresponding macroblocks of the frames, and modifies the bit allocation between the focal areas and no focal areas according to the levels of importance to give more bits on the focal area while keeping more useful information of the video frame in the lower bit-rate output.

In the Focal Area Resolution Control, there are three main problems that need to be solved:

1. How do we obtain the focal area information?

2. How must we map the focal area onto macroblock levels?
3. How do we reallocate the bits from a different area?

To collect the focal area information, a research project called “Perceptual Focus Driven Digital Video Transmission” [26,27] has been created within our research group. One purpose of this project is to collect information about the focal area. Within that research, an eye-glance sensor will be placed at a viewer’s terminal. Some focus-tracking algorithm will be studied to incorporate automatic eye-glance tracing mechanism of a viewer’s perceptual focus into the transmission scheme. Other object tracking research project has been created detail in the publication “Flock-of-Bird Algorithm for Fast Motion Based Object Tracking and Transcoding in Video Streaming”.

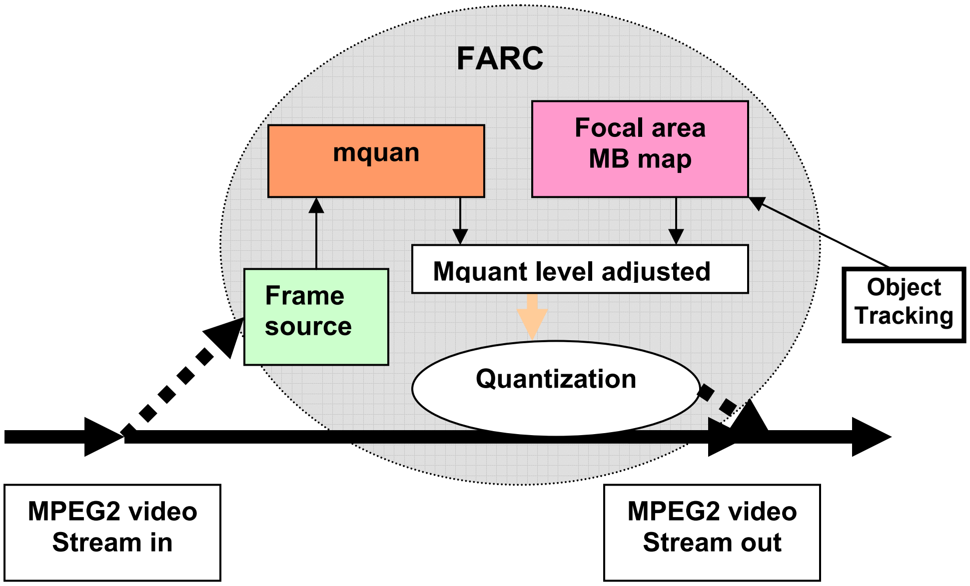


Figure 4-1 Diagram Of Focal Area Resolution Control

In FARC, the viewer's perceptual focus is assumed to be a rectangle. The viewer's perceptual focus information format is the position of the focal rectangle in the viewer's window frame and the reallocation level. The information format is as follows:

# of focal area, X_0, Y_0, X_d, Y_d, V

Since there may be more than one focal area in the window frame, the "***# of focal area***" denotes an individual area. The position of the focal area rectangle is set as the coordinate of the upper left-hand corner (X_0, Y_0). In the lower right-hand corner (X_d, Y_d) and the upper left-hand corner of the viewer's window frame, the origin is set up. And X, Y is in percentage format, which is the percentage of the total window frame value. For example, figure 4-2 shows a frame within the focal area. The focal area in the yellow rectangle. (X_0, Y_0) is set to (35, 10) and (X_d, Y_d) is set to (70,60). The X number 35,70 is the percentage of the frame width. And the Y is the percentage of the frame height. The reallocation level V is decided by the viewer. After the data is ready, the eye-glance sends the information to the FARC.

When the FARC receives the focal area information, the FARC analyzes the information, generates the *focal area MB map,* and calculates the reallocation levels for the normal areas.

Figure 4-2 Picture With A Focal Area

How is the *focal area MB map* generated? When a video stream goes through the FARC, the FARC detects the *display_horizontal_size* and the *display_vertical_size* of the video frame in pels, the *mb_width* (number of macroblocks in width), and the *mb_height* (number of macroblocks in height). Combining this information of ($\boldsymbol{X_0}$, $\boldsymbol{Y_0}$) and ($\boldsymbol{X_d}$, $\boldsymbol{Y_d}$), the FARC scans all the macroblocks to see if it is a focal area macroblock or not. Figure 4-3 shows the *focal area MB map* for the picture shown in Figure 4-2. All the blocks in shadow area are the focal area macroblocks and the others are the normal area macroblocks.

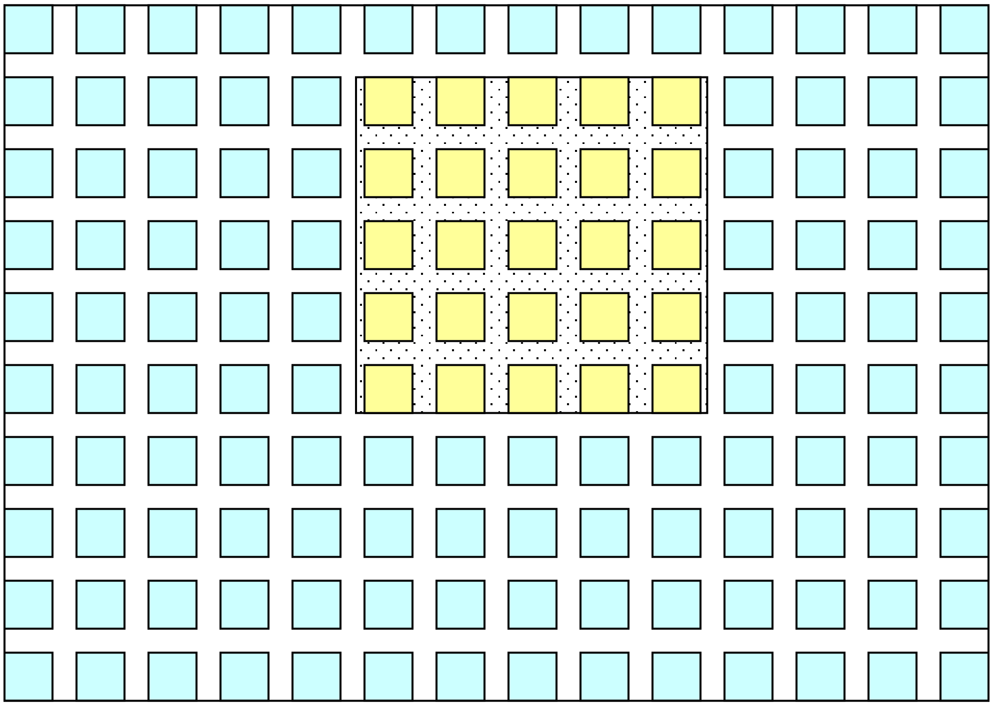


Figure 4-3 Focal Area MB Map

Since the reallocation level V_i of the focal area macroblock is in the feedback information, the FARC needs to determine the reallocation level $V_i^{'}$ for the normal areas.

As discussed in Chapter three, in an MPEG2 TM5 encoder, the output bit-rate is controlled by the *quantization* step. The DCT coefficients *x*s are first quantized by individual quantization factors *f(x, quant_step)*, which are determined by the ISO/IEC 13818-2 tables [16]. Then the quantizer scale parameter *mquant* is used to determine the actual quantization step sizes that are applied to these DCT coefficients. The quantized output for intra-and non-intra frames are respectively given by:

$$y = \frac{f(x, quant\_step) + .75 \times mquant}{2 \times mquant} \qquad y = \frac{16 \times f(x, quant\_step)}{mquant} \tag{4.1}$$

y is limited to the range [-2047….2047]. To count for a few of these factors, in the most top level, the value of $mquant_j$ for each macroblock is calculated as a product of two primary factors of the *buffer fullness* Q_j and the *macroblock activity* N_act_j.

$$mquant_j = Q_j \times N\_act_j \tag{4.2}$$

The final value of $mquant_j$ is coded either in the slice or in the macroblock header. The modulation parameter Q_j has been discussed in Chapter three is the reference quantization parameter obtained in bit allocation step 2 and is used to control the output bit-rate. In our FARC, the modulation parameter N_act_j is more useful. Normalized activity N_act_j is given as following for *macroblock j:*

$$N\_act_j = \frac{2 \cdot act_j + avg\_act}{act_j + 2 \cdot avg\_act} \quad , \tag{4.3}$$

As we discussed in chapter three, when the AVT downscales the output bit rate, video quality will be sacrificed. In FARC, we try to divide a video frame into different areas that are denoted by focal and normal areas according to levels of importance. N_act_j of the focal area is reduced to obtain more bit allocation so that the image quality in these areas can be enhanced. At the same time, N_act_j of the normal area is increased to reduce the bit allocation. In turn, the total bits to form this frame are the same as before and then the output bit rate remains.

The FARC sets a resolution level V_i for different areas. If the AVT allocates more bits for the *macroblocks* in the focal area according to the different V_i, then $V_i^{'}$.for the normal areas have to be calculated to balance the bit allocation. In regards to the considerations above, the equation (3.23) becomes:

$$mquant_j = Q_j * N\_act^{\mathrm{mod}ified}{}_i, \tag{4.4}$$

where,

$$N\_act^{\mathrm{mod}ified}{}_i = N\_act_i \cdot 2^{\sum_{all} \frac{-\log_2 V_i}{n}}, \tag{4.5}$$

Since the final output stream is in binary coding,, the *mquant* level is adjusted by a factor of $2^{\sum_{all} \frac{-\log_2 V_i}{n}}$. When *Vi* is bigger than 1, $2^{\sum_{all} \frac{-\log_2 V_i}{n}}$ is in the range of 0 ~ 1, and $N\_act^{\mathrm{mod}ified}{}_i$ will be less than N_act_j; so $mquant_j$ is reduced. According to equation

(4.1), the effective quantization step size becomes smaller and more bits are allocated to the macroblock. Thus, the resolution is increased.

In the normal area, *Vi* is set to a value less than 1 and $2^{\sum_{all} \frac{-\log_2 V_i}{n}}$ is bigger than 1 When $N\_act^{\text{mod}ified}{}_i$ is bigger than N_act_j, $mquant_j$ is increased. According to equation (4.1), the effective quantization step sizes become bigger, and fewer bits are allocated to the macroblock. The resolution is reduced.

When a macroblock's $mquant_j$ is adjusted, the output bits for this macroblock change. Does it affect output bits for the whole frame and the whole video?

Let us assume that after the management of the FARC, the total bits added in the focal area is:

$$B_f = \sum_{mb \;\; in \;\; \text{F-}area} \log_2 v_i, \tag{4.6}$$

The original bits for this frame are:

$$B_0 = \sum_{all \;\; \text{mb}} \log_2 v_0, \tag{4.7}$$

The difference for this frame is:

$$E_f = \sum_{\text{mb in F\_area}} \log_2 v_i \text{ - } \sum_{mb \text{ in F\_area}} \log_2 v_0, \tag{4.8}$$

If the normal area can reduce total bits *Ef*, then the total number bits of the frame will stay unchanged.

Assume that the normal area reduces bits by a number of E_n for this frame.

$$E_n = \sum_{mb\ in\ N-area} \log_2 v_0 - \sum_{mb\ in\ N-area} \log_2 v_i^{'} , \tag{4.9}$$

It follows the bits for this whole frame as:

$$E = E_f - E_n , \tag{4.10}$$

$$= \sum_{mb \inf ocal area} \log_2 v_i - \sum_{mb \inf ocal area} \log_2 v_0 - \left(\sum_{mb in N-area} \log_2 v_0 - \sum_{mb in N-area} \log_2 v_i^{'} \right), \tag{4.11}$$

Assume that the total number of macro - blocks in the focal area is and the rest of the areas is N.

Then,

$$E = M \times \log_2 v_i - M \times \log_2 v_0 - N \times \log_2 v_0 + N \times \log_2 v_i^{'}$$

$$= M \times \log_2 v_i - (M + N) \times \log_2 v_0 + N \times \log_2 v_i^{'}$$

let $E = 0$, it follows

$$V_i{'} = \frac{V_0^{\left(\frac{M}{N}+1\right)}}{V_i^{\frac{M}{N}}} \tag{4.12}$$

We set the normal area reallocation level by equation (4.12) to keep the total bit output for this frame unchanged. Repeat the same procedure for all the frames, and then the output bit rate of the video will not change.

For example, if a video frame has 9 macroblocks, the maximum value for every micro-block is 8.

8	8	8
8	8	8
8	8	8

Total bits for this frame is: $B = \sum \log_2 8 = 27$

Assume after SRC the maximum value for every micro-block is 4.

4	4	4
4	4	4
4	4	4

The total number of bits become: $B_0 = \sum \log_2 4 = 18$

Calculate through FARC with $v_i = 1024$, $\mathrm{M} = 1$ and $N = 8$ with equation (4.10). It is as follows:

$$v_i^{'} = \frac{v_0^{\left(\frac{m}{n}+1\right)}}{v_i^{\frac{M}{N}}} = \frac{4^{\frac{9}{8}}}{1024^{\frac{1}{8}}} = 2$$

2	2	2
2	1024	2
2	2	2

Total bits for this frame that has been reallocated by the FARC is:

$$B_0^{'} = 8 \times \log_2 2 + \log_2 1024 = 18$$

After $V_i^{'}$ has been determined, the FARC assigns the $V_i^{'}$ to every macroblock in the normal area and V_i to every focal area macro-blocks. The *focal area map (*Figure 4-3) will be *reallocation level MB map* shown in figure 4-4, which has the reallocation level for every macroblock.

When the FARC recompresses a macroblock, it checks the *reallocation level MB map* to obtain the corresponding reallocation level V_k. It adjusts the *quantization* step parameter *mquant* and increases or decreases the bit allocation for this macroblock.

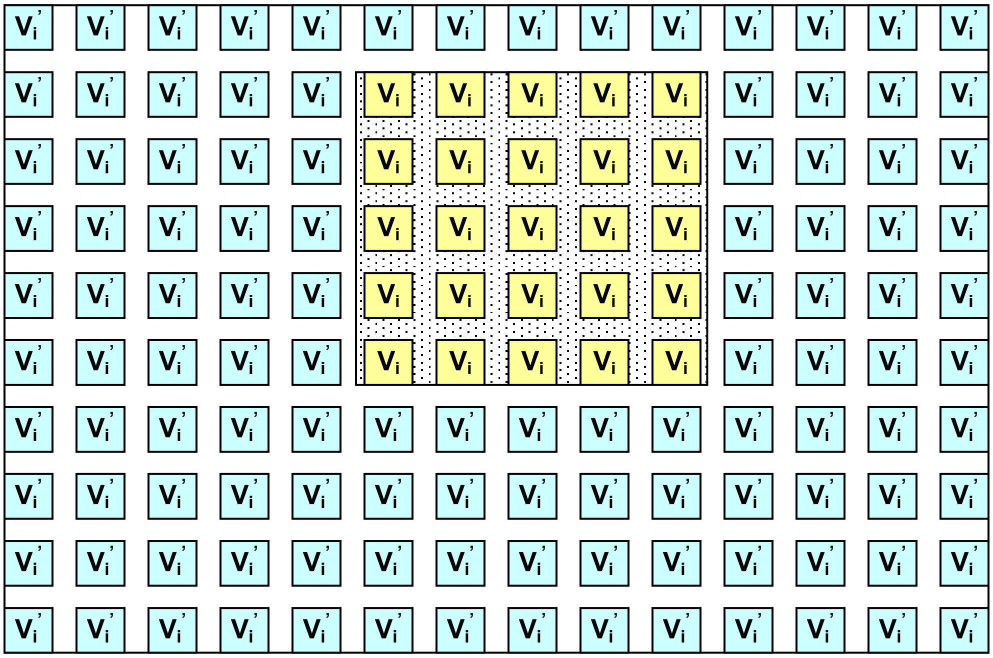


Figure 4-4 Reallocation Level MB Map

The detailed numerical process will be introduced in the next chapter.

Chapter 5

NUMERICAL APPROACHES

The ideas and schematic theorems for the Symbiotic Rate Control and the Focal Area Resolution Control have been discussed in previous chapters. The numerical approaches and experiments for these two AVT components are studied in this chapter to justify the ideas and give guidance for future research and applications.

5.1 The construction of the AVT system

In order to perform the numerical experiments, An AVT system has been constructed in a Linux system. In order to simplify the experiment process, we focused more on the network node where the AVT is embedded, and less emphasis on the server and client. A video stream file as the input of the experiments represents the server; the two other, which are the feedback information and the output video stream files, represent the client or the forward network. In simpler terms, the active network environments are simulated, and the experiment results are analyzed by comparing the output and input video stream files.

Figure 5-1 shows the AVT system. The AVT was constructed by combining the decoder, re-encoder and its functional components (SRC or FARC). The decoder and re-encoder in this AVT are called *xdec* and *xenc*, respectively. They have all the functions of the common decoder and encoder, but have been modified for an AVT. Four pipes are used to connect the decoder and re-encoder. Pipe Y, U, and V are employed to transfer the three types image files, luminance (Y) and chrominance (U,V), from *xdec* to *xenc*. The fourth pipe transfers the video stream information that *xdec* decoded from the video stream to *xenc*, like frame size, GOP size ex. When the AVT starts, the *xdec* will communicate with the xenc to establish the four pipes, then it begins to decode the input video stream. After the video sequence information is decoded in *xdec*, *xdec* sends all the video parameters to *xenc; xenc* starts to recompress the video picture from the Y,U, and V pipes and the new video stream output from AVT.

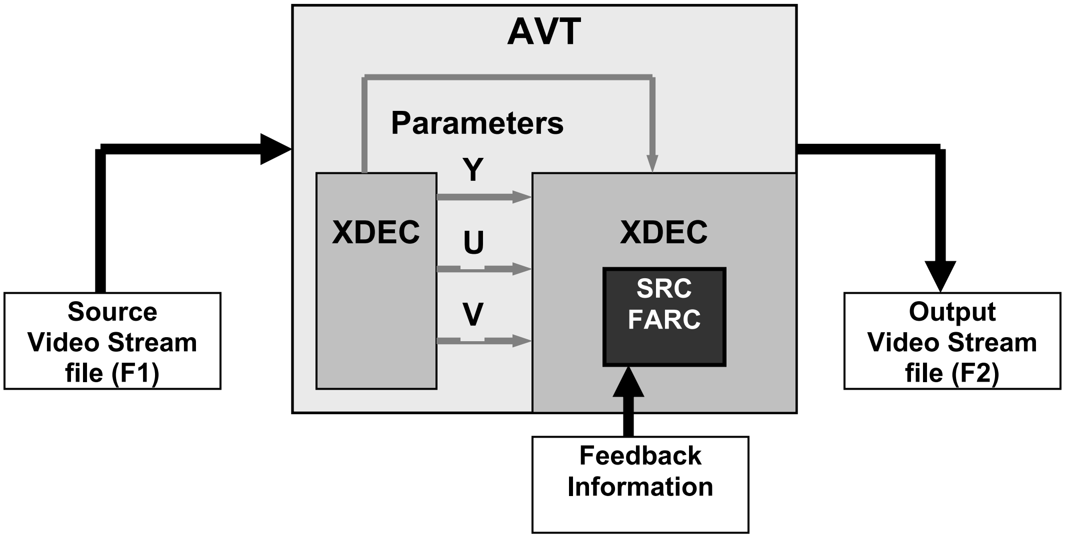


Figure 5-1 Construction of The Experimental AVT

The software used in the experiments have been developed based on an MPEG-2 Encoder/decoder, Version 1.2 (MPEG-2 TM 5) created by an MPEG software simulation group [20]. In these experiments, the main part source code of the encoder (*xenc)* and decoder (*xdec*) are from public sources [28].

In order to achieve basic AVT functions, several embedded points are chosen and modified to implant the functional codes. The program flowchart is shown in figure 5-2.

The main jobs for creating an AVT are:

1. Create an AVT program structure. All the AVT programs are collected in the directory AVT. The subdirectories *xdec, xnec* include all the programs for *xdec* and *xenc.* The subdirectory *test* includes the AVT's main program, all the parameter files and some sample video stream files.

2. Create the AVT main program to manage the AVT. This file is named as *mpeg2trc.c* and is saved in the directory *AVT/test/.* The major work for this program is to start an AVT main process and obtain the input and output video stream file names from then command line. Then, they generate children processes *xdec* and *xenc* and pass the input and output file names to them. The main process monitors the two child processors until they finish their jobs and then terminate the whole process altogether.

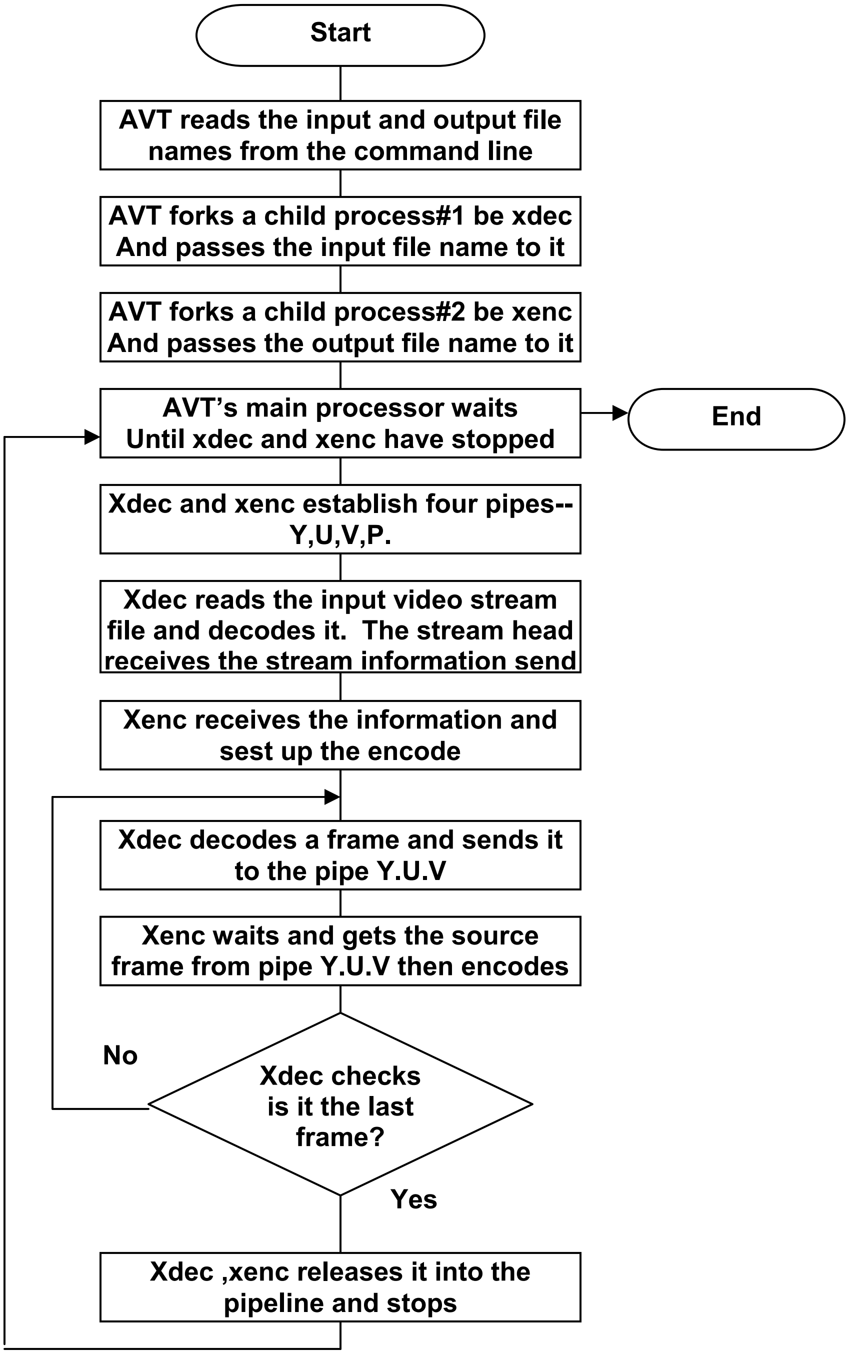


Figure 5-2 Flowchart Of The AVT

3. Modify the TM 5 decode to be an *xdec*. In TM5, the decoder is an independent program. It obtains the input video stream and output video file name from the command line and the output is a video file in the format of Y.U.V. It is saved in the local directory.

   In the AVT, the *xdec* is the subprogram of the AVT. It starts by using the AVT main program. The more important thing is when *xdec* decodes the video stream head. All the video parameters from the stream head are pushed to the pipe *P,* so that the *xenc* can fetch them from the pipe P and use them to recompress the video. After decoding the stream head information, the *xde*c starts to decode the video stream and all the output video image files are redirected to the three pipes (Y,U,V). After a while, the *xdec* coordinates with the *xenc* so that the *xenc* can fetch the video files from the Y,U,V pipes and start to re-encode them. As soon as the pipe is empty, the *xdec* decodes the next frame. Frame by frame, the xdec decodes the video until it reaches the last frame. After that, xdec sends *a last frame signal* to the *P* pipe and finishes decoding.

   The files related to *xdec* are all in the directory of AVT/xdec. Files that are modified from the original decode are:

   - global.h — this file defines all the global variables. All new flags for AVT and new functions used for the AVT were added to it.
   - mpegdec2.c — this is the main file of *xdec*. The code for constructing the four pipes,coordinating with the AVT, and xenc were added to this file.

- gethdr.c — this file is used to decode the video stream header. In AVT, the file has been added to the code to push all the decoded parameters that go into the program and into pipe *Y.*
- store.c — this file is used to store the decoded video file. In AVT, the output video files are pushed into the three pipes of Y, U, and V.
- getpic.c — this file is used to decode every frame. In AVT, the code to check if the current decode frame is the last frame and send the last frame information to the pipe Y, is added to it.

4. Modify the TM 5 encode to be a *xenc*. It is the same as the *xdec*. The original encode has been modified to be a sub-program of the AVT. All the programs of the *xenc* are stored in the directory of AVT/xenc. The difference between the original encoder and *xenc* is the following: the original encoder reads the video image file from the local directory and *xenc* receives them from the three pipes. The original encoder reads the video parameters which are used to encode the video from a parameter file; and the xenc receives all this information from the video stream of the *xdec* decode from the video stream and puts it into pipe *Y.* The original encode knows how many frames will be encoded. And the *xenc* doesn't have that information. It keeps encoding until the *xenc* receives the *last frame signal* that is sent by *xdec* from pipe *Y.*

Files that were modified from the original encode are:

- global.h — this file defines all the global variables. It contains all new flags for the AVT and new functions used for the AVT were added to it.
- mpegenc2.c — this is the main file for *xenc*. The code for constructing the four pipes and coordinating with the AVT and *xde*c were added to this file. And the codes are also modified to read the video parameters instead from the parameter file to get them from pipe Y.
- readpic.c — this file is used to read the video file. In the AVT, the codes are modified to read the video picture from the three pipes instead of a local file.
- putseq.c — this file is used to encode every frame. In the AVT, the codes are used to check if the current frame is the last one of the stream.

5. Create a method for the *xdec* to transfer all the video stream information to *xenc*. The detail has been shown in 3. 4.

With all this done, a basic AVT is created. The SRC and FARC functions are then able to be added to the basic AVT.

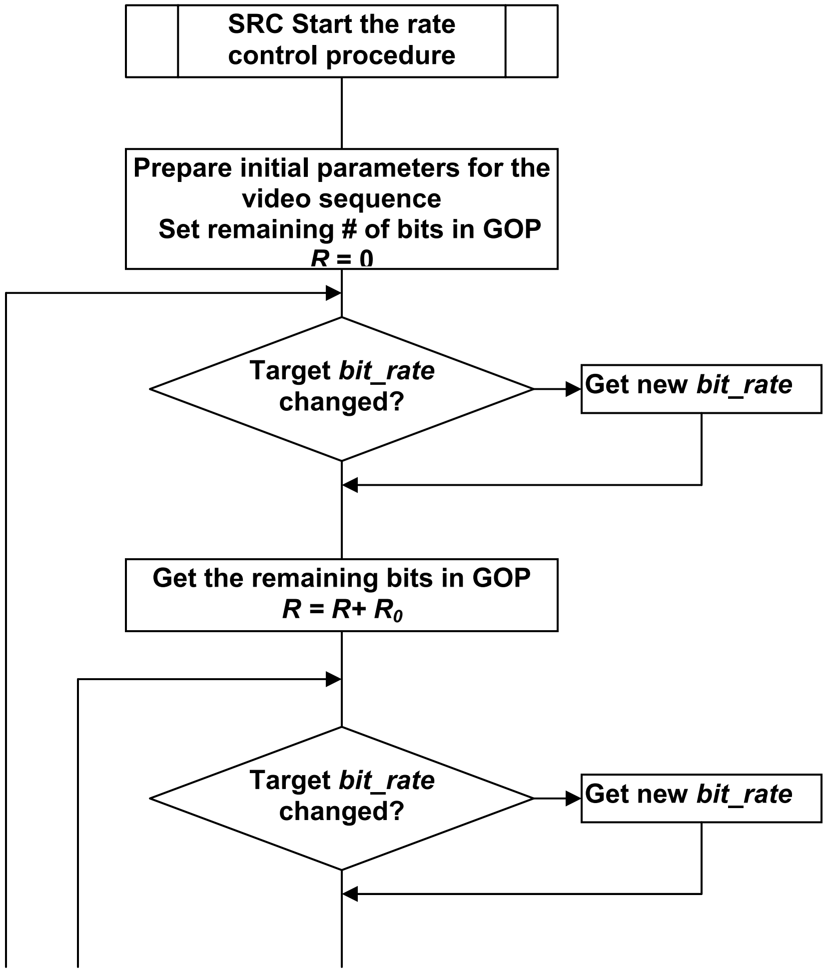
5.2 Numerical approach of the Symbiotic Rate Control (SRC)

As mentioned before, we assume that the symbiotic controller has detected all the information of the network and has stored them in a file--- the symbiotic file. In turn, the rate controller only needs to fetch all this information for re-encoding from this symbiotic file. As we discussed in chapter three, the main thing to control the output bit rate is to change the quantization step size parameter *mquant*, so the symbiotic rate control reacts to the network situation by changing the *mquant*. The embedding point for symbiotic rate control is in *xenc*'s quantization procedure. At the beginning of every picture being decoded, the SRC checks the symbiotic file to see if it needs to replace output bit_rate for the SRC. If the bit_rate has been changed, it re- calculates the target bit_rates for the GOP and also for the current picture. Then, the SRC uses the new target rate to calculate the *mquant*. The numerical program flowchart of the SRC is shown in figure 5-3.

The main jobs for the SRC are:

1. Create a symbiotic file *rate.par* and put it into the AVT/test/ directory. File rate.par has all the target bit-rate information for the video stream to adapt the network capability change.
2. Adapt the *AVT's* video stream output bits to the change needed. The main embedding point for the SRC is in the *AVT's xenc* quantization procedure. The code is included in the AVT/xenc/ratectl.c. Before the AVT re-encodes a GOP or picture, the SRC will check the *rate.p*ar file to adjust the target bit allocation for the current GOP or picture and reduces the out put bits rate for the AVT. Since the AVT has a huge time-consuming overhead, it is very hard to measure the output bit rate for the AVT. In the experiments of this thesis, we use the output file size to exemplify the results of the bit rate. For the same input video stream, if the AVT running time is the same but the output video stream file size is different, the out put bit rate is different.

3. Setup the statistics data collection system. The code is embedded in AVT/xenc/stats.c. All the statistics data will describe the quality of the video and the bits allocation for every video frame. In the SRC, SNR is used to represent the quality of the video. For every picture in the output video, SNR is separated among SNR_Y, SNR_U, *and* SNR_V for luminance file Y and chrominance files of U and V. The AVT running time for the current picture is T. All this information is stored in two files in theAVT/test called ETfile.out and DTfile.out.



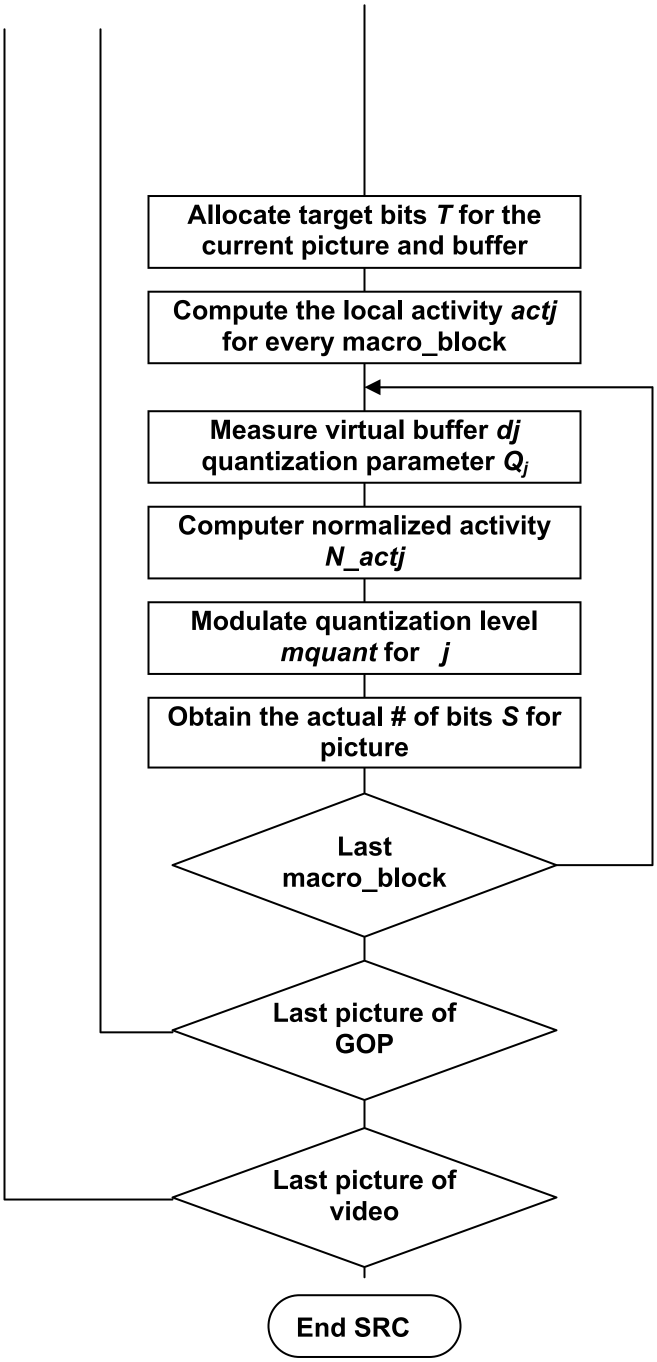


Figure 5-3 Symbiotic Rate Control Program Flowchart

5.3 Numerical approach of the Focal Area Resolution Control (FARC)

As mentioned before, we assume that the eye track has caught the focus spots, formed a focal information file, and has given feedback to the FARC. The main job of the FARC is to reallocate the bits between different macroblocks within the picture. As discussed in chapter four, the main thing to control the bits allocation is to modify the local activity for every macroblock, then to change the quantization step size parameter *mquant*. It is important to generate the relocation level map for the video picture. The numerical program flowchart of the FARC is shown in figure 5-4.

The main jobs for FARC are:

1. Create the focal area information file named *adjustq.* The file is put in AVT/test. The information included in this file is the location and size of the focal area.
2. Generate the relocation level map *adjmquant.* First get the focal area information and map it onto macroblocks, and then calculate the reallocation level for the normal area. After that, create the reallocation map *adjmquant.* The file is put into AVT/test. The code to achieve this is embedded in AVT/xenc/ratectl.c.
3. Setup the statistics data collection system. All the statistics data describe the quality of video and the bits allocation for every macroblock within the frames. This code is embedded into AVT/xenc/stats.c. The results file will be put into AVT/test/ directory.

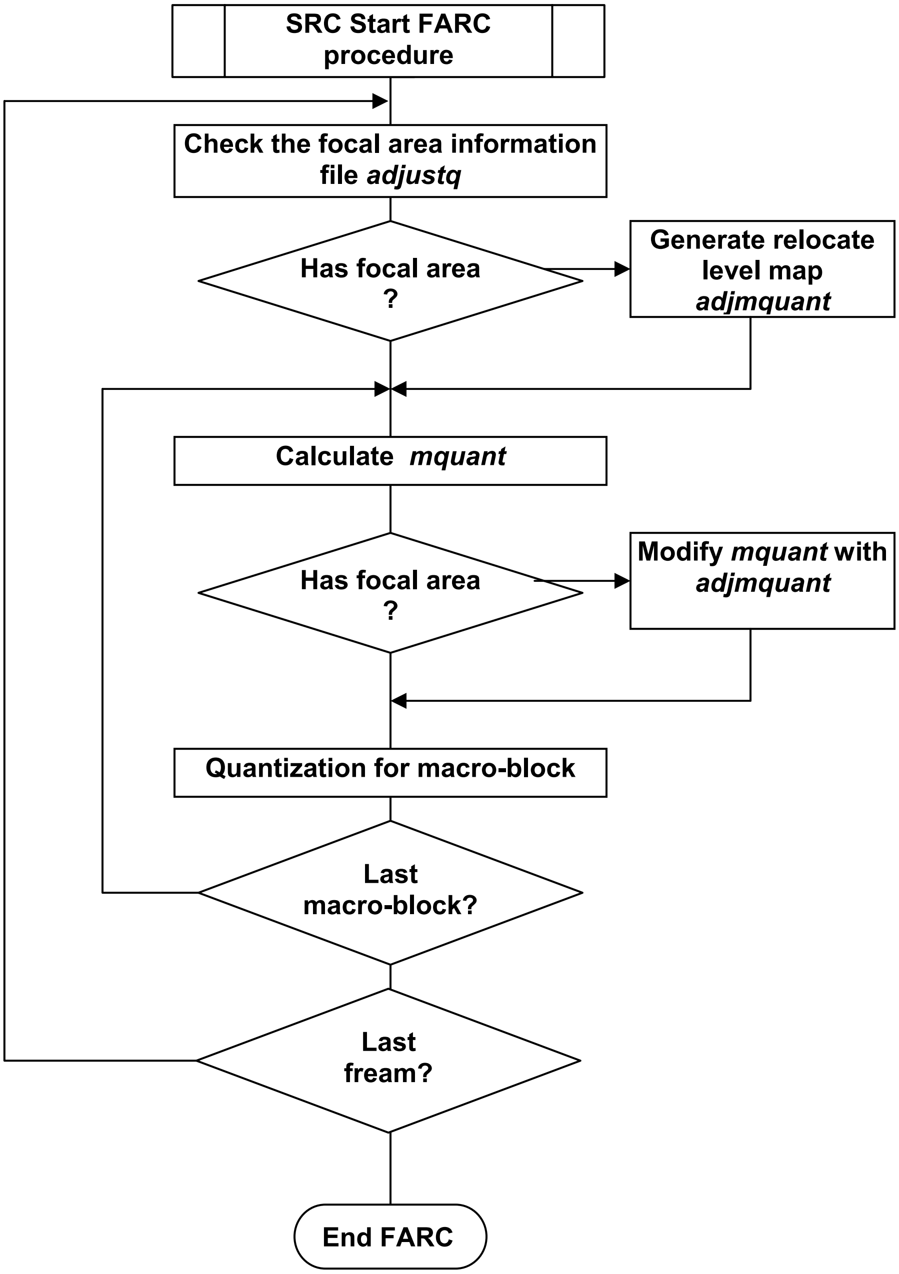


Figure 5-4 Focal Area Resolution Control Program Flowchart

Chapter 6

EXPERIMENTS

In order to study the functionality of the Symbiotic Rate Control (SRC) and the Focal Area Resolution Control (FARC), a series of experiments with different video stream samples were done based on the numerical approaches introduced in Chapter 5. All experiments have successfully proved to justify the idea of the SRC and the FARC.

6.1 Experiment arrangement for the SRC

The purpose of the experiments for the SRC is to prove the functionality of the SRC: the SRC can reduce the output bit rate of MPEG-2 video streams according to the target bit rate from the symbiotic file in all kinds of conditions.

A Linux machine was used to run the AVT in these experiments. The command for running the AVT program of the SRC is named *ftest*. The file *test.m2v* is used to input the sample MPEG2 video streams. The file *rate.par* is the symbiotic file, which is used to store the target bit rate information. The file *new.m2v* is used to store the video streams that have been treated by the SRC. The files *ETfile.out* and *DTfile.out* are used to store all the SNR and computing time information. An individual test runs by the following steps:

1. Rename the sample video stream to *test.m2v* as an input file;

2. Set target bit rate in the file *rate.par*;
3. Run the AVT by typing the command *ftest*;
4. After running the command, the output files *new.m2v*, *ETfile.out,* and *DTfile.out,* are overwritten with the new results;
5. Copy the data to Excel files with names indicating the specific experiment from the output files;
6. Analyze the results stored in the Excel files.

In order to test the AVT in different situations, ten different video streams with different bit rates have been chosen for these experiments. The summary of the ten video streams is shown in Table 6-1.

	File name	**Frame size**	**Bit rate**	**Number of frame**
1	*Earth.m2v*	256*176	4194.4kb/s	175
2	*garden.m2v*	704*408	3000kb/s	19
3	*Sking.m2v*	352*240	4194.4kb/s	175
4	*sa-704h-m_.m2v*	704*408	40960kb/s	491
5	*sa-704s-m_.m2v*	704*408	20480kb/s	49
6	*Sa-352s-h_.m2v*	352*240	10240kb/s	474
7	*Sa-352s-m_.m2v*	352*240	5120kb/s	474
8	*Balloom.m2v*	352*240	1638.4kbs/s	110
9	*Sa-176-h-.m2v*	704*408	2560.7kb/s	474
10	*Sa-176-m-.m2v*	704*408	1020.7kb/s	474

Table 6-1 List Of The Sample Video Stream Files

For each sample video stream, around ten tests were taken with different target rates. Because the different video stream have different original bit rates, the target rate series used for the tests may be different, but they are all return to the original bit rate.

The data to analyzed from the output Excel files included:

- Output file size vs. target bit rate. This is used to show how the SRC controls the output bit rate for the MPEG-2 video streams.
- Image quality parameter—SNR vs. control rate. This is used to monitor image quality changes after the SRC process.
- AVT computing time. This is used to monitor if the AVT computing times are related to the video stream size and bit rates.

6.2 Experimental results for the SRC

The experiments have been done for all the ten video streams indicated above. Although the sample video streams have different frame sizes, original bit rates or numbers of frames, they all have shown similar results for the SRC. Since the results are quite comparable, three sample analyses are chosen and shown in following.

Sample #1:

The test video stream is named *earth.m2v*. The display frame size is 256*176. The total number of frames of the video stream is 175. The original output bit rate of the video stream is 4194.4 kb/s. 12 tests have been taken with target bit rate from 0.005 Mb/s to 3 Mb/s. The results are shown in table 6-2,figure 6-1, and figure 6-2. In Table 6-2, the target bit rate column shows the series of the target bit rates used in the tests. SNRY is the average SNR over Y files of all frames of the video stream; SNRU, and SNRV are the average SNR over the U and V files, respectively. The output file size is the size of output stream. The Total AVT time is the AVT running time of the specific test.

Target bit rate (Mb/s)	**SNRY** (DB)	**SNRU** (DB)	**SNRV** (DB)	**Output file size** (Mb)	**Total AVT time** (ms)
3	48.300	33.495	32.050	3.905124	26.775
2.75	45.836	31.685	31.550	3.573165	26.699
2.5	43.653	30.404	30.750	3.24832	26.623
2.25	41.622	28.427	30.250	2.924221	26.539
2	39.959	27.069	29.900	2.599701	26.452
1.5	37.259	24.944	28.800	1.950096	26.274
1	34.210	22.698	27.100	1.300482	26.097
0.5	29.134	18.703	23.350	0.650642	25.907
0.1	18.674	6.798	9.335	0.155395	25.476
0.05	18.560	6.668	7.885	0.154427	25.482
0.01	18.428	6.586	6.885	0.154002	25.472
0.005	18.474	6.579	6.820	0.154016	25.468

Table 6-2 Sample #1 SRC Result

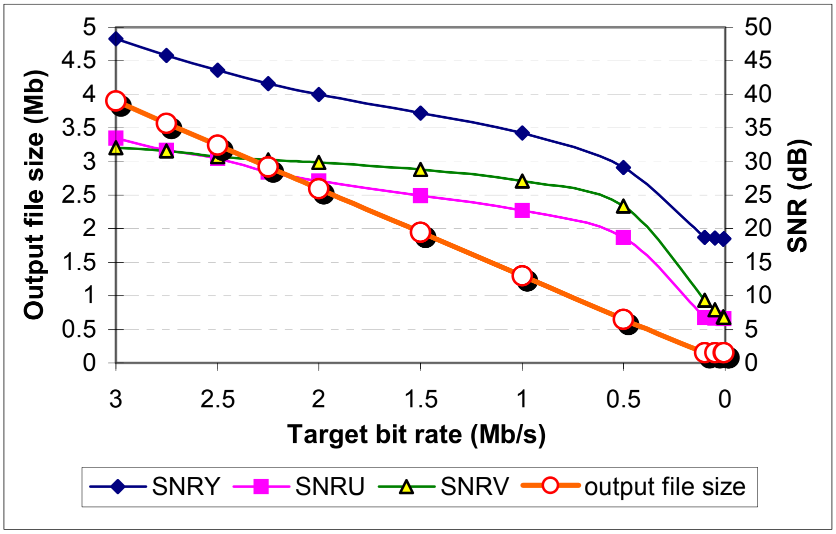


Figure 6-1 Sample #1 Target Bit Rate Vs. Output File Size and SNR Results

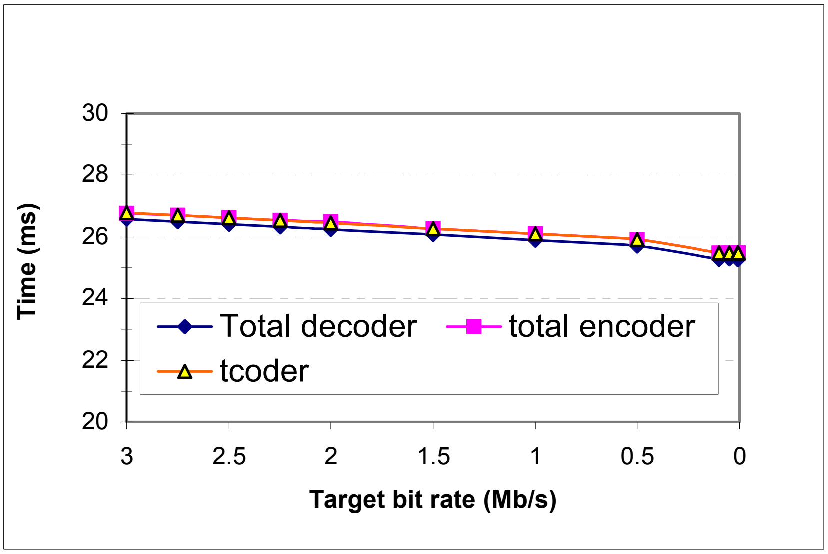


Figure 6-2 Sample #1 AVT Running Time Comparison

Sample #2:

The test video stream name is *garden.m2v*. The display frame size is 704*408. The total number of frames of the video stream is 19. The original out put bit rate of the video stream is 3000 kb/s. 12 tests have been taken with the target bit rates, ranging from 3000 kb/s to 5kb/s. The results are shown in table 6-3, figure 6-3, and figure 6-4.

Target bit rate (Mb/s)	**SNRY** (DB)	**SNRU** (DB)	**SNRV** (DB)	**Output file size** (Mb)	**Total AVT time** (ms)
3	28.263	27.384	25.121	0.301049	18.510
2.75	22.421	21.474	20.068	0.273978	18.382
2.5	20.032	18.558	16.763	0.250694	18.383
2.25	18.300	16.332	14.542	0.22711	18.373
2	16.937	14.726	12.795	0.203427	18.362
1.5	14.847	12.447	10.663	0.16026	18.315
1	13.653	11.521	9.915	0.135952	18.296
0.5	12.653	10.974	9.292	0.120599	18.260
0.1	12.253	10.830	9.084	0.11328	18.231
0.05	12.216	10.807	9.024	0.112423	18.227
0.01	12.179	10.795	8.982	0.111634	18.227
0.005	12.168	10.789	8.965	0.111531	18.224

Table 6-3 Sample #2 SRC Result

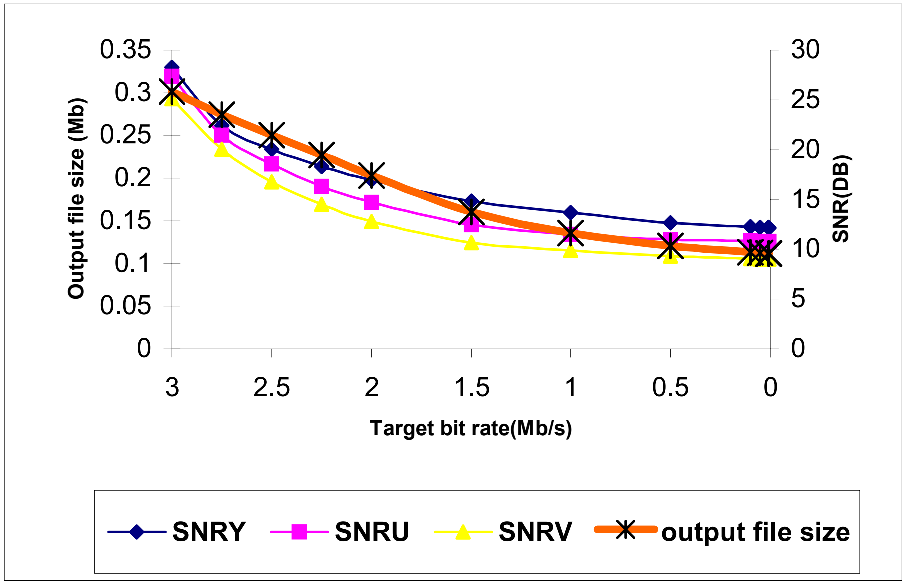


Figure 6-3 Sample #2 Target Bit Rate Vs. Output File Size and SNR Results

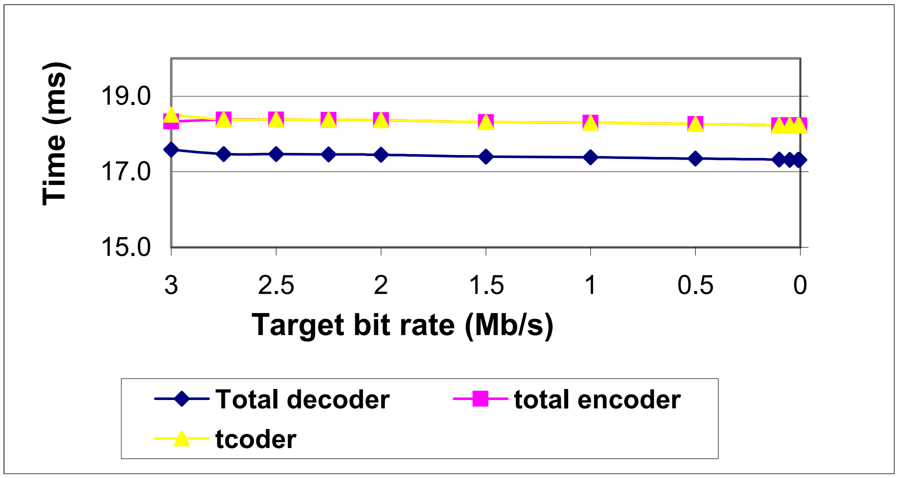


Figure 6-4 Sample #2 AVT Running Time Comparison

Sample #3:

The test video stream is named *Sking.m2v*. The display frame size is 352*240. The total number of frames of this video is 175. The original output bit rate of the video stream is 4194.4kb/s. Twelve tests have been taken with target bit rates varying from 4200 kb/s to 5 kb/s. The results are shown in table 6-4, figure 6-5, and figure 6-6.

Target bit rate (Mb/s)	**SNRY** (DB)	**SNRU** (DB)	**SNRV** (DB)	**Output file size** (Mb)	**Total AVT time** (ms)
4	32.198	24.737	18.414	3.573165	24.607
3.5	30.870	23.690	17.454	3.046831	24.484
3	29.567	22.641	16.530	2.612612	24.347
2.5	28.105	21.485	15.542	2.177043	24.216
2	26.397	20.239	14.481	1.742002	24.078
1.5	24.299	18.801	13.282	1.307059	23.939
1	21.508	16.958	11.795	0.872104	23.781
0.5	17.015	13.962	9.343	0.436531	23.589
0.1	13.169	10.705	6.698	0.258612	23.437
0.05	13.153	10.699	6.683	0.257577	23.441
0.01	13.146	10.696	6.682	0.256958	23.451
0.005	13.145	10.696	6.681	0.256906	23.453

Table 6-4 Sample #3 SRC Results

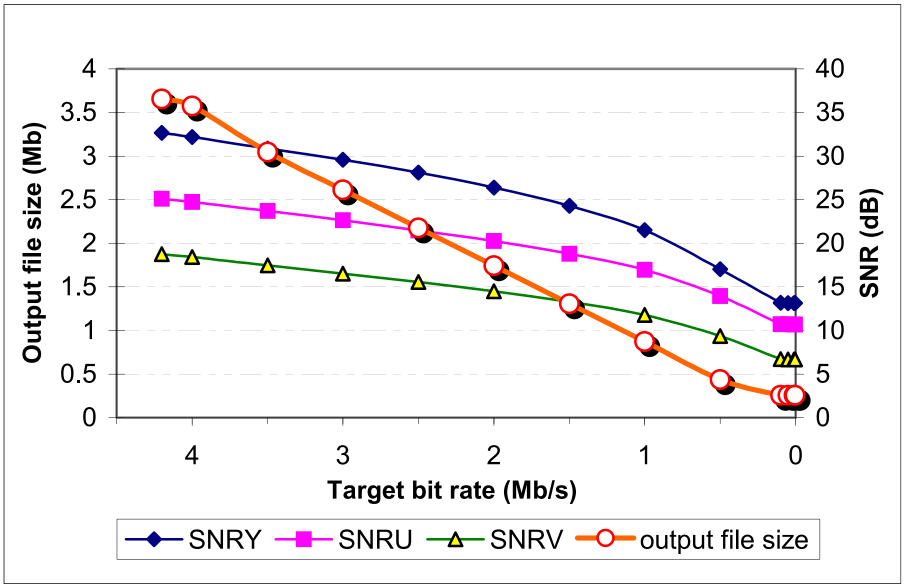


Figure 6-5 Sample #3 Target Bit Rate Vs. Output Bits and SNR Results

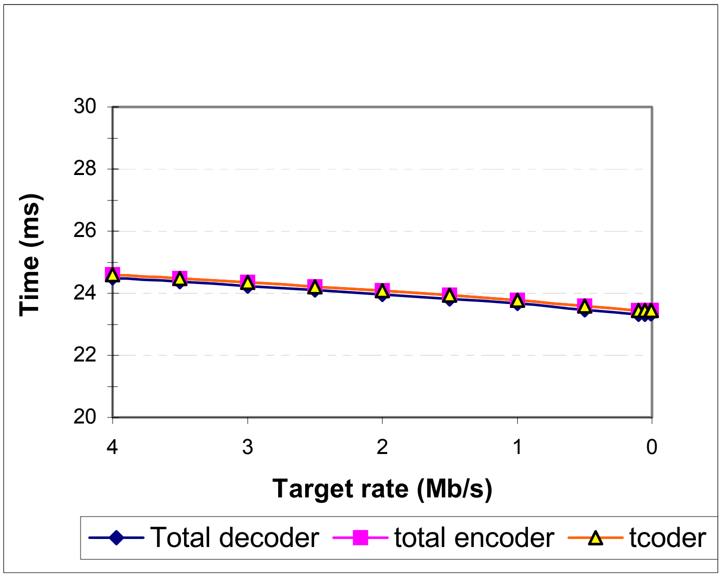


Figure 6-6 Sample #3 AVT Running Time Comparisons

The experiment results of the rest samples are similar to that of these 3 samples. The following figure shows the rest of the results (Target Bit Rate Vs. Output Bits) of the experiment.

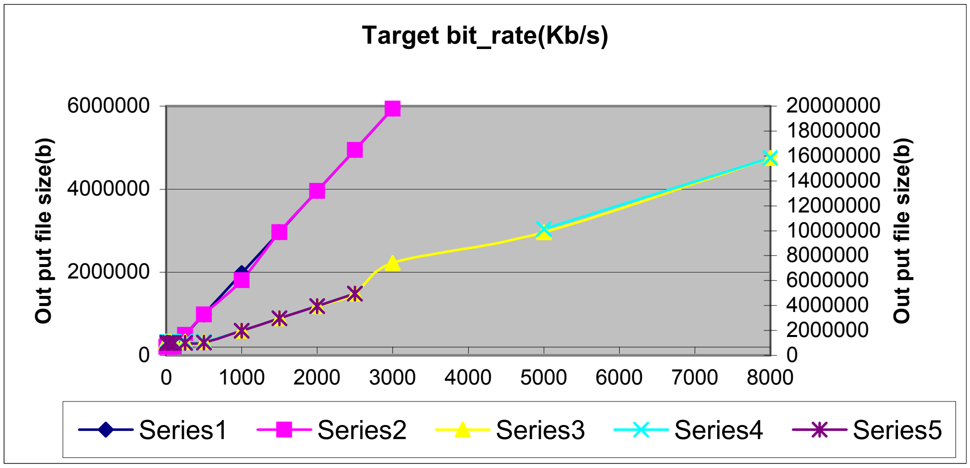


Figure 6-7 Target Bit Rate Vs. Output Bits for other experiment

6.3 Analysis and Conclusions of the experiments for the SRC

With the experiment results shown in the previous section, the following conclusions have been obtained:

1. The output bit rate is successfully controlled by the symbiotic rate controller. In figure 6-1, 6-3, and 6-5, the curve in orange shows the relation between the target bit rate and output bit rate. In a certain range of the target bit rates, such as 3 Mb/s to 0.1 Mb/s for sample #1, 3 Mb/s to 0.5 Mb/s for sample #2 and 4 Mb/s to 0.05 for sample #3, the output bit rates are successfully controlled by the target bit rate.

2. By decreasing of the target bit rate, the reduction ratio of output bit rate becomes smaller and smaller. Then it reaches a saturate point. Below this point, output bit rate will not be lowered any further. This point indicates the minimum bit rate required to keep the basic structure information of the video stream. Tables 6-2, 6-3, and 6-4 show that when the target bit rate decreases to 0.1 Mb/s and below, the output file sizes for SNRY, SNRV, and SNRV for change minimally. This means that a 0.1Mb/s target bit rate is approximately the minimum bit rate that can be controlled by the SRC.

3. The output bit rate is lowered by sacrificing the video image quality. In Figures 6-1, 6-3, and 6-5, the curves in yellow, rose, and dark blue show lowered SNRs after the target bit rate has been reduced. All three curves drop more than fifty percent. In addition to this, the visual comparison of the video streams before and after the SRC processing give the same conclusion. Figure 6-8 shows the picture of one frame in the original video stream of sample #8 while Figure 6-9 shows that of the video stream whose output bit rate has been reduced by the SRC with a target rate of 0.5 Mb/s. Obviously, the image shown in Figure 6-8 has much higher quality than that of Figure 6-9.

4. Bit rates can be controlled by the target bit rate no matter what frame size the video steam has. It is easy to obtain the result by comparing the frame sizes of samples 1, 2 and 3.

5. Bit rates can be controlled by the target bit rate no matter what length the video stream has.

Figure 6-8 Original Video Picture

Figure 6-9 Video Picture At SRC Target Rate 0.5Mb/S

6. Even though the computing time is not very fast, it depends only on the original video file size and is not affected very much by the target control rate. Figures 6-2, 6-4, and 6-6 show that the tests for a same sample at different target bit rates having pretty close computing time. Additionally, samples 1 and 3, which have almost the same original file size, have similar computing times while sample 2 has both a smaller original file size and smaller computing time.

6.4 Experiment arrangement for the FARC

The purpose of the experiments for the Focal Area Resolution Control (FARC) is to prove the functionality of the FARC, that is the FARC can reallocate the bits for macroblocks in a individual frame of a video stream according to the focal area information file to enhance the image quality in the focal area and reduce the bit rate the video streams under all kinds conditions.

Though the FARC can be functioned independently, in our experiment the FARC is used with the SRC in an AVT system. Since the SRC may lose the image quality of the video, the FARC is proposed to enhance the image quality in the focal area. So an individual experiment was arranged to run the AVT twice. First, the sample video stream (V0) is input the AVT and run the SRC function to reduce the bit rate in a certain ratio (RR) to obtain a new video (VS). Second, the same sample stream V0 is input to the AVT, switch on the FARC and run the AVT with both bit rate reduction by the SRC and bit reallocation by the FARC to obtain the second new out put video (VF) with the same rate reduction ratio (RR). After the experiment the original video (V0), the video after SRC (VS) and the video after FARC (VF) are compared to see the effects of the FARC. Obviously, the AVT in the experiments here has to be functioned with both the SRC and the FARC, and a switch is also required to turn on/off the function of FARC.

The test procedure to obtain VS or VF is similar to the procedure we used in the experiments for the SRC in section 6.1. The command *ftest*, the files *test.m2v*, *rate.par* and *new.m2v*are are used similarly in obtaining VS or VF, while the addition files *adjustq, BITfile and SNRfile* are added into the system. The file *adjustq* is used to set up the focal area information in obtaining VF. The file *BITfile* records all the bits information for every macroblocks of all over the video stream. The file *SNRfile* records all the SNR information for every macroblocks of all over the video stream. Theoretically, the focal area information for different frame of a video stream can be different and can also be adjusted dynamically, however, to make the case a little simpler we assume the focal area information is same for all frames in a same experiment. An individual test runs by the following steps:

A. Get the VS result.

1. rename the sample video stream to *test.m2v* as a input file;
2. set target bit rate in the file *rate.par* ;
3. set focal area information file *adjustq* and turn off the *FARC* in *adjuztq.*
4. run the AVT by type command *ftest*;
5. after running, the output files *new.m2v*, *BITfile* and *SNRfile* are overwritten with the new results;
6. copy data to Excel files with names indicating the specific experiment from the output files;

B. Get the VF result.

7. turn on the *FARC* in *adjuztq*;
8. run the AVT by type command *ftest*;
9. after running, the output files *new.m2v*, *BITfile* and *SNRfile* are overwritten with the new results;
10. copy data to Excel files with names indicating the specific experiment from the output files;

C. analyze the results stored in the Excel files.

The data to be analyzed from the experiments include

- the output video stream file size for VS and VF, which is used to show that the FARC increase the focal area image quality without increase the bit rate of the MPEG-2 video streams;
- the number of bits used for every macroblock in the same frame of VS and VF, which is used to show that the SFAC can reallocate the bits in the frame. The focal area get more bits allocated and the rest area get less;
- the SNR , which is used to monitor the image quality of the VS and VF,

6.5 Experiment result analysis for the FARC

Total four sample video streams have been tested in the way described in previous section. The summary of these sample video streams is shown in table 6-5.

	File name	**Frame size**	**Bit rate**	**Number of frame**
1	*Balloom.m2v*	352*240	1638.4 kb/s	110
2	*garden.m2v*	704*408	3000 kb/s	19
3	*Rose.m2v*	240*162	2092.7 kb/s	110
4	*Kai.m2v*	240*162	2092.7 kb/s	318

Table 6-5 Test Video Information For FARC

Because the experiment results for all 4 video streams are pretty similar, one of the results is analyzed in following.

The sample video stream is *bollam.m2v*, the frame size of this stream is 325*240. Total frames for this test video is 20. The focal area is (30,30 70,70) and the resolution level is 4. After the FARC process, the estimated resolution level for rest area is (-4). The resolution level in the focal area is set by the resolution control level which is defined in the file *adjustq*, the resolution level of the rest area is automatically assigned by the FARC. The resolution control *mquant* map is shown table 6-5:

-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	4	4	4	4	4	4	4	4	4	4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Table 6-6 Illustration Of Macroblock Level Focal Area Map For Sample #1

The results for a I-frame of the video stream are shown and analyzed in following:

1. The bits allocation comparison. In the experiments, the number of all bits allocated in every macroblock is collected for the video stream after different process. The following three tables show the bits allocations of I-frame for V0, VS and VF.

120	84	58	48	69	74	66	44	43	70	42	56	56	62	64	46	42	43	51	61	69	289
131	72	75	62	64	64	73	68	68	70	67	51	50	63	50	54	64	49	61	65	66	273
136	48	65	57	54	74	66	93	86	73	73	66	54	69	77	70	54	58	71	67	51	316
196	112	340	331	349	335	271	179	309	335	335	333	333	263	335	258	260	257	350	251	333	611
303	263	188	263	289	149	267	202	317	223	116	157	233	192	299	282	385	337	311	338	290	686
509	403	302	299	452	386	462	337	405	363	397	381	478	209	378	710	899	1021	662	684	1009	821
628	215	206	333	379	406	304	324	343	396	1172	1211	1472	388	327	410	1242	406	462	733	331	634
630	335	347	278	389	436	336	367	267	326	1137	1426	1135	486	425	460	349	411	362	521	307	709
516	353	272	264	304	364	407	641	485	465	1025	1717	855	738	619	409	326	410	585	372	831	1337
545	328	375	351	464	649	694	1762	1622	1728	1912	2070	1246	1864	1902	1845	1669	1925	1658	1549	1517	1658
515	369	449	584	1163	1454	1986	1837	1310	1389	1360	910	877	1038	1076	1598	1360	1252	1366	1514	1035	1273
557	416	476	462	1188	1119	1040	472	769	470	625	677	911	1052	829	500	641	847	376	291	1071	813
259	484	429	460	1159	1063	1042	622	875	820	753	555	662	518	558	522	766	1117	521	321	347	571
289	460	683	565	1127	1128	1200	990	830	807	550	636	922	326	268	377	394	638	517	660	234	708
816	881	966	1030	1067	1265	1009	819	549	878	848	624	694	689	814	653	454	803	424	404	528	935

Table 6-7 Bits Allocation Of I-Frame In Original Video Stream V0 Of Sample 1

116	74	56	48	65	69	60	44	43	66	48	54	54	58	59	52	42	43	49	57	59	232
124	48	57	50	58	52	49	50	50	53	56	45	50	51	44	44	50	49	45	49	52	194
120	48	49	49	49	50	56	57	50	46	46	45	45	51	50	50	47	44	43	52	44	166
112	48	56	50	52	54	47	43	50	51	51	49	49	50	51	50	52	50	50	43	49	155
114	51	50	50	54	50	48	45	46	43	44	43	47	48	52	55	49	49	48	48	49	128
109	45	46	47	53	51	53	51	57	51	52	53	50	50	51	69	63	68	59	51	79	128
125	53	46	51	49	52	49	48	49	51	88	89	100	63	49	49	76	57	57	66	50	113
116	52	46	47	49	50	49	49	48	51	93	94	79	64	51	51	50	50	50	45	53	100
115	47	46	47	49	48	51	49	48	48	72	103	66	46	48	47	48	48	47	50	71	108
120	45	53	51	50	52	54	97	121	105	95	123	87	117	102	93	107	110	101	88	91	92
108	49	50	55	75	81	118	110	75	60	63	51	51	54	59	62	66	61	63	91	70	101
100	50	53	51	115	93	131	54	50	51	55	46	61	67	55	46	57	74	59	53	66	70
103	47	49	49	83	82	135	58	66	59	55	52	54	55	50	48	47	95	49	48	52	48
107	48	55	54	58	76	136	48	63	66	50	48	60	51	48	52	47	154	57	52	46	43
120	87	82	87	72	86	131	57	55	70	62	51	49	52	55	51	55	138	62	52	59	75

Table 6-8 Bits Allocation Of I-Frame In The Video Stream VS Of Sample 1

106	52	53	48	55	58	46	44	43	56	42	50	56	50	46	46	42	43	51	49	51	125
108	48	51	50	52	58	49	50	50	49	53	51	44	51	44	44	50	49	45	49	52	118
105	48	49	49	55	50	50	53	50	46	46	45	45	51	50	50	47	50	43	46	44	93
110	48	56	56	58	54	47	43	50	51	51	49	49	50	51	50	52	50	50	43	49	90
111	51	44	50	60	50	72	52	64	59	44	51	55	56	68	69	55	49	48	48	49	88
113	45	46	47	53	51	75	67	71	69	68	65	62	50	63	136	55	51	49	51	54	84
113	47	46	45	49	52	71	56	61	59	209	152	195	74	65	61	51	57	57	55	50	69
113	52	46	47	49	50	59	57	60	59	175	186	149	84	67	63	56	50	50	45	53	72
117	53	46	47	49	48	57	72	56	56	130	303	109	83	61	55	54	48	47	50	66	73
114	45	53	51	50	52	88	247	256	219	234	318	181	277	283	241	76	71	87	63	63	73
106	49	50	55	53	70	324	250	158	150	151	97	87	114	106	164	62	58	56	77	58	66
102	50	53	51	76	69	83	54	50	51	55	46	57	59	55	46	57	54	59	53	62	56
96	47	49	49	70	69	89	58	58	55	55	52	54	55	50	48	47	53	49	48	48	48
101	48	55	54	58	66	89	48	59	60	50	48	57	51	48	52	47	50	57	52	46	43
108	73	72	76	58	58	77	50	55	51	47	51	49	52	51	51	55	58	58	52	55	59

Table 6-9 Bits Allocation Of I-Frame In The Video Stream VF Of Sample 1

In order to compare the bit allocation, the numbers of total bits allocation in the focal area, the rest area and the their sum are calculated respectively in Table 6-10.

Video stream after different process	Bit rate	Total bits of I-frame (bits)	Total bits in the focal area	Total bits in the rest area
	(bits/s)	(bits)	(bits)	(bits)
V0	2,150,000	182,917	56,507	126,410
VS	50,000	22927	5019	17908
VF	50,000	23136	8205	14931

Table 6-10 Comparison Of Bits Allocation Of I- Frame

Table 6-10 shows that the bit rates of both VS and VF are reduced comparing to V0, but VF have more bits allocated in the focal area than VS, while the bit rate or the total bit of I-frame are almost the same for both VS and VF. This proves that the bit reallocation is successfully executed in the FARC process.

2. Comparison of SNR for the video streams V0, VS and VF.

The SNR of focal area were calculated for the video streams after different process V0, VF and VF. The following three charts show the SNRs vs. position in a I-frame of V0, VS and VF.

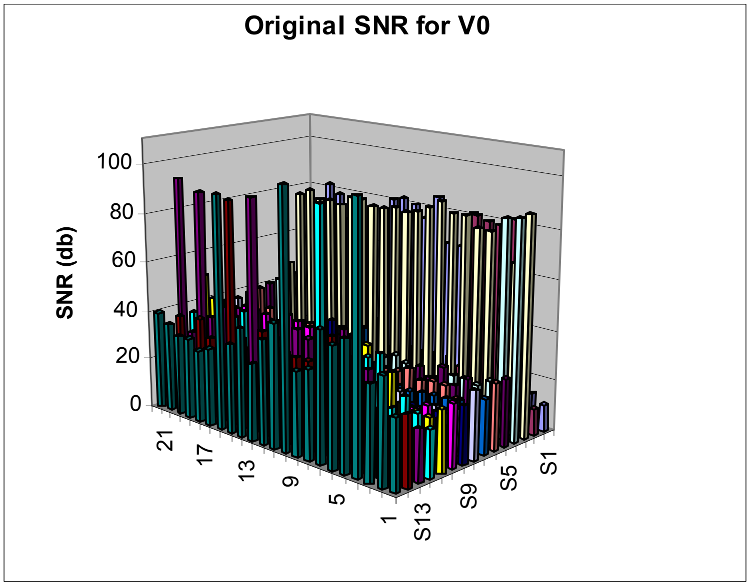


Figure 6-10 SNR In Original Video Stream V0

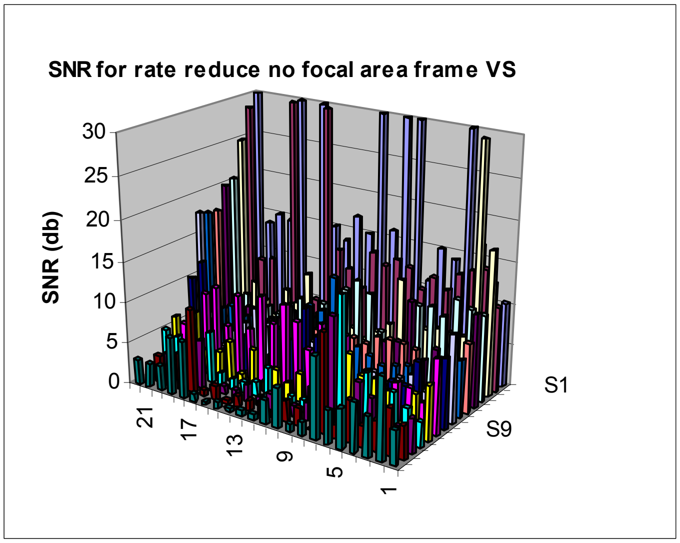


Figure 6-11 SNR In Video Stream VS

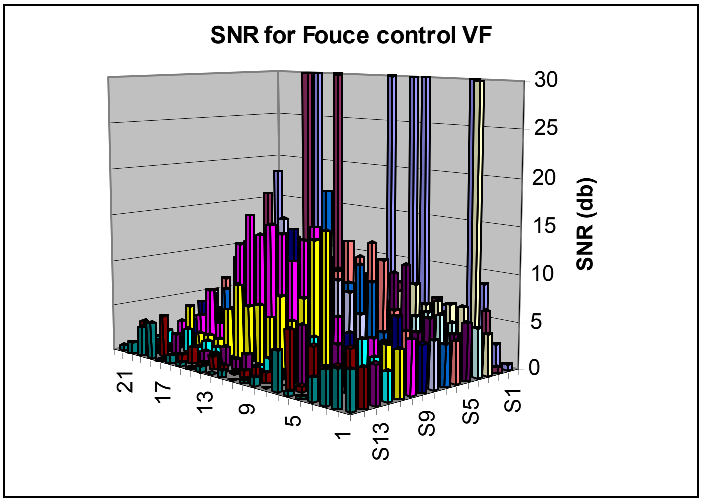


Figure 6-12 SNR In Video Stream VF

Comparing Figures 6-11 and 6-12, one can obviously see that the SNRs in focal area of VF are higher than that of VS. This has given the evidence that the image quality of the video stream after bit rate reduction has enhanced by the FARC process.

Table 6-11 shows that both VS and VF have lower average SNR over the whole picture frame comparing to the original video stream V0. This is because the SRC reduced the video bit rate by sacrificing the video quality. In the focal area, VF has higher average SNR than VS while in the rest area VF has lower average SNR than VS. This gives evidence that the FARC has reallocated the bits and enhanced the image quality in the focal area by sacrificing that in the rest area.

Video stream after different process	Average SNR over whole frame	Average SNR over the focal area	Average SNR over the rest area
V0	34.96	25.29	37.57
VS	7.77	5.25	8.45
VF	6.65	9.80	6.065

Table 6-11 Comparison Of The Average SNR In Different Area Of I-Frame For The Video Streams After Different AVT Processes

4. SNR versus the resolution control level.

Figure 6-13 shows the SNRs changing versus resolution control level, where the resolution control level is the parameter reflects the expected image quality enhancing degree of the focal area, which is defined in the file *adjuztq*, level 0 means the FARC doesn't turn on; *SNR FY, SNR FU* and *SNR FV* are the average SNRs over the focal area for Y-,U- and V- files of image respectively, while SNR *RY, SNR RU* and *SNR RV* are the average SNRs over the rest area. The curves in

Figure 6-13 shows that when the resolution control level increases from 0 to 8, the average SNRs for the focal area increase, while that for the rest area decreases.

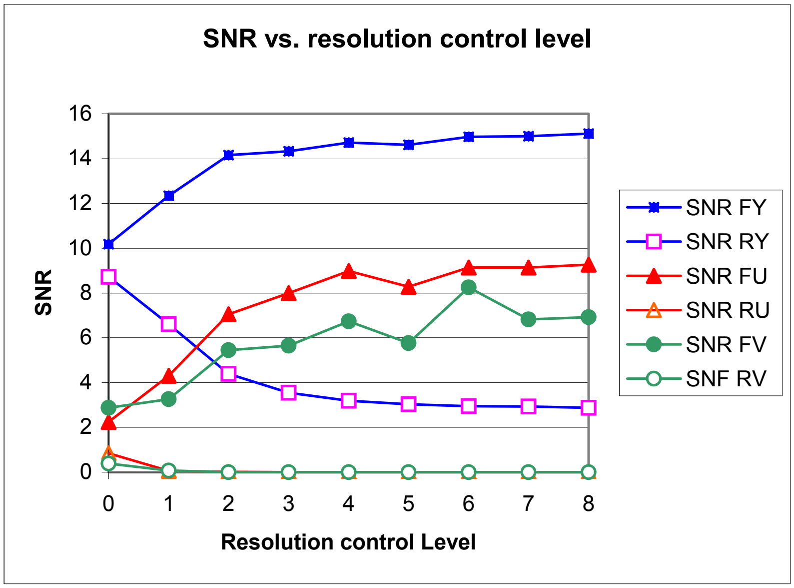


Figure 6-13SNR For Focal Area And Rest Area Depend On The Different Resolution Levels

5. Comparison of the visual effect of the video stream after different AVT processes.

After the AVT processes, the different video streams V0, VS and VF were created. The video streams were played by a same MPEG-2 play, and the image quality enhancing effect was observed by comparing playing VS and VF. The following two pictures were grasped from the video streams VS and VF. The image enhancing effect is obvious in the focal area in the picture for VF.

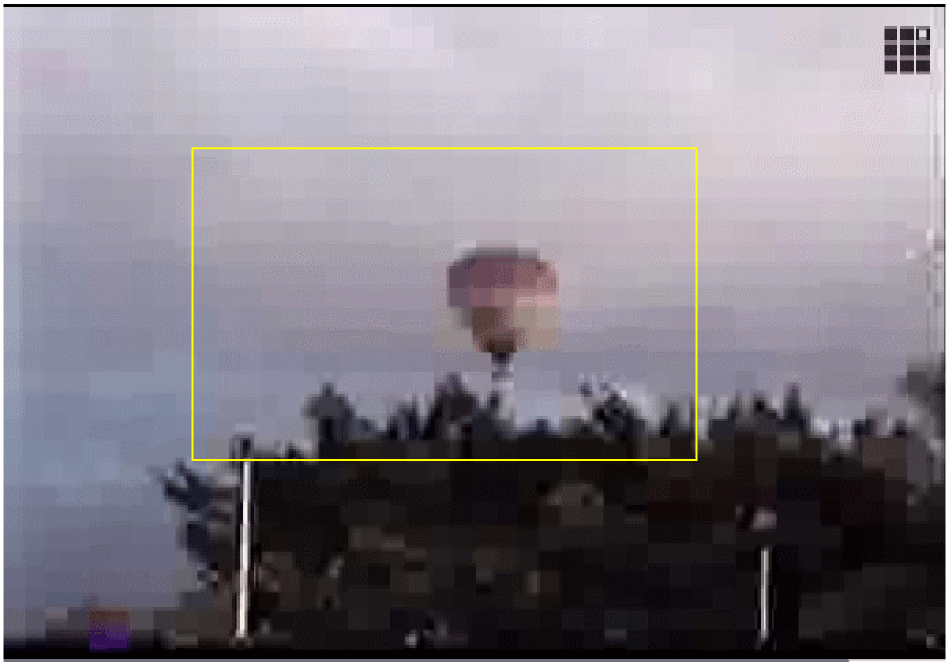


Figure 6-14 Picture Of A Video Frame Grasped From The Video Stream VS

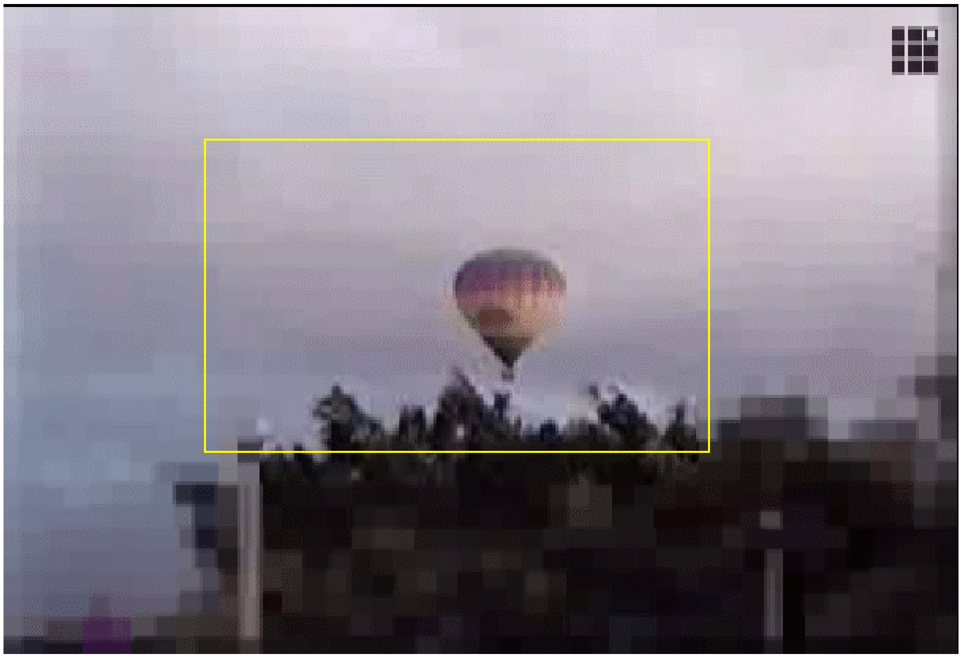


Figure 6-15 Picture Of The Same Frame Grasped From The Video Stream VF

6.6 Conclusions of experiments for the FARC

With the experiments results analysis in the previous section, the following conclusions are obtained for the FARC

1. The FARC only reallocate the bits over the macroblocks in every video frame, the total number of bits remains unchanged.
2. The FARC has successfully enhanced the image quality of the focal area by lowering that of the rest area
3. The image quality enhancing degree can be successfully controlled by the FARC via the resolution control level.
4. With appropriate parameters for the AVT, and using both the FARC and the SRC, a movie can be played at a every low bit rate but still have very good image quality at interesting focal areas

Chapter 7

SUMMARY

The adaptive video communication schemes on the Internet have been popularly studied worldwide to overcome network congestion problems. One smart solution, the Active Video Transcoding (AVT) technology, has been studied based on the idea of the Active Network being able to transfer data streams on asymmetric networks and avoid data being blocked or broken.

Because of their importance within AVT technology, the Symbiotic Rate Control (SRC) and the Focal Area Resolution Control (FARC) have been studied within this thesis for MPEG-2 video stream transferring. The SRC scheme has been developed to dynamically and intelligently adjust the bit rate of video data transmission in the active network in accordance to the network situation to avoid network congestion. However, an unfortunate consequence is that it may sacrifice video quality. As a solution to avoid sacrificing the video quality, the FARC scheme has been developed to enhance the image quality in the areas of interest, which means that the video can be transmitted and played at a lower bit rate, but still have good image quality in focal areas. In these two schemes, the quantization step size of MPEG-2 (*mquant*) has been chosen as the implant point of the AVT code, the SRC and FARC programs communicate with MPEG-2 via this point and adjust the bit rate to adapt the network situation dynamically and intelligently.

The sample programs for the SRC and FARC have been coded basis on the MPEG-2 TM5, the SRC, and FARC functions and have been successfully demonstrated in a simulated active network environment. A series of experiments have been done based on these programs. The experiments for the SRC have shown that:

- The output bit rate can be successfully controlled by the SRC according to the symbiotic control file (the target bit rate file) no matter what frame size and video length the input video streams have.
- By decreasing the target bit rate, the reduction ratio of output bit rate becomes smaller and smaller, and then reaches a saturate point. Below this point, output bit rate will not be lowered anymore. This point indicates the minimum bit rate required to keep the basic structure information of the video stream.
- The output bit rate is lowered by sacrificing video image quality.
- Even though the computing time is not very fast, it depends only on the original video file size and is not affected very much by the target control rate.

The experiments for the FARC have also shown that:

- The FARC only reallocate the bits over the macroblocks in every video frame, and the total numbers of bits remain unchanged.

- The FARC has successfully enhanced the image quality of the focal area by lowering that of the rest area.
- The image quality enhancing degree can be successfully controlled by the FARC via the resolution control level.
- With appropriate parameters for the AVT, and using both the FARC and the SRC, a movie can be played at a very low bit rate, but still have very good image quality at focal areas of interest.

In conclusion, the experiments have successfully proved and demonstrated the idea of how the SRC and the FARC work in an active network environment to adapt a network situation. The research experiment developed schemes and codes useful in both guiding the future research activities and implementing commercial AVTs in the next generation networks: The commercial Active Networks.

Based on the research in this thesis, the following research topics may be interesting and useful for future study:

1. Improve programs to lower the computing time.
2. Dynamically obtaining focal area resolution adjustment information from the client.

APPENDIX A

Publications Related To This Research Work

1. Javed I. Khan, Qiong Gu and Raid Zaghal, *Symbiotic Video Streaming by Transport feedback based Quality Rate Selection,* PThe 12th acket Video Workshop PV 2002, Pittsburgh, PA, April 24-26, 2002, http://www.pv2002.org (electronic proceedings).
2. Javed I. Khan, Seung S. Yang, Darsan Patel, Oleg Komogortsev, Wansik Oh, Zhong Guo, Q. Gu and Patrick Mail, *Resource Adaptive Netcentric Systems on Active Network: A Self-Organizing Video Stream that Auto Morphs Itself while in Transit via a Quasi-Active Network*, DARPA Active Network Research Conference and Exposition, DANCE 2002, Sun Francisco, May 28-31, 2002, pp.427-445.
3. Javed I. Khan, Raid Zaghal, and Qiong Gu, *Dynamic QoS Adaptation for Time Sensitive traffic with Transientware*, Proceedings of the 4th International Conference on Wireless and Optical Communication, WOC03, July, 2003, Banff, Canada. Editor: Lambertus Hesselink, ISBN: 0-88986-374-1 (383) (to appear).
4. Javed I. Khan, Raid Zaghal, and Qiong Gu, *Symbiotic Streaming of Time Sensitive Elastic Traffic on an Interactive Transpor*t, Proceedings of the 8th

IEEE Symposium on Computers and Communications - ISCC'2003, Antalya, Turkey, June 2003 June 2003, pp.1435-1440.

5. Javed I. Khan, Raid Zaghal, and Qiong Gu, *Dynamic Transport Enhancement for Time Elastic Traffic with Transientware,* Invited Panel Presentation on future NASA's Space Internet III, 2003, Cleveland, Ohio, June 2003. (also a panel presentation).

BIBLIOGRAPHY

[1] Andrew Odlyzko, *Internet growth: Myth and reality, use and abuse*, A. M. Odlyzko. iMP: Information Impacts Magazine, November 2000.

[2] *The Acceleration of Internet Transmissions --Backbone Bandwidth* {http://www.zdnet.com/devhead/stories/articles/0,4413,2216104,00.html}

[3] http://www.internet2.edu/about/aboutinternet2.html

[4] http://abilene.internet2.edu

[5] Jakob Nielsen, *Nielsen's Law of Internet Bandwidth*, 05 Apr 1998 issue Alertbox Articles

[6] Michael Pastore, *Internet Bandwidth Expands Around the Globe*, http://cyberatlas.internet.com/big_picture/hardware/article/

[7] R.Braden et al. Reso*Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification,* September 1997

[8] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, *A Survey of Active Network Research IEEE Communications Magazine*, Vol. 35, No. 1, pp80-86. January 1997.

[9] David J. Wetherall, Ulana Legedza, and John Guttag, *Introducing New Internet Services: Why and How. Appears in IEEE Network Magazine* July/August 1998

[10] http://www.isi.edu/abone/

[11] John Watkinson, *MPEG Handbook* Focal Press; 1st edition September 2001

[12] Video Transcoding By Reducing Spatial Resolution, *Published in Proceedings of the IEEE Int'l Conf. on Image Processing (ICIP)*, Vancouver, Canada, Sept. 2000

[13] B. Zheng and M. Atiquzzaman, *"TSFD: two stage frame dropping for scalable video transmission over data networks "*, IEEE Workshop on High Performance Switching and Routing, Dallas, TX, 29-31 May 2001, pp. 43-47

[14] Peng Peng Yin, Min Wu, and Bede Liu, *"VIDEO TRANSCODING BY REDUCING SPATIAL RESOLUTION"*

[15] *http://www.chiariglione.org/mpeg/index.htm*

[16] *ISO/IEC 13818-2:2000 Information technology -- Generic coding of moving pictures and associated audio information: Video*

[17] CCIR Recommendation 601: *Encoding Parameters of Digital Television for Studio*, *Green Book*, 1990.

[18] David L. Tennenhouse and David J. Wetherall, *Towards an Active Network Architecture*, Computer Communication Review, Vol. 26, No. 2, April 1996.

[19] J. Solomon, *Applicability Statement for IP Mobility Support*, October 1996

http://www.ietf.org/rfc/rfc2005.txt

[20] MPGE software simulation group, *http://www.mpeg.org/MPEG/MSSG/tm5/*

[21] Jacobson, *Congestion Avoidance and Control*, Proc. SIGCOMM' 88, Conf, ACM, pp-214-329, 1988.

[22] K. Ramakrishnan, *The Addition of Explicit Congestion Notification (ECN) to IP*, http://citeseer.nj.nec.com/ramakrishnan99proposal.html

[23] Brakmo, L.S., O'Malley, S.W., and Peterson, *L.L,TCP Vegas: New Technique for Cogestion Detection and Avoidance,* Proc. SIGCOMM' 94 Conf. ACM, pp-24-35, 1994.

[24] Floyd, S., and Jacobson, V., *Random Early Detection gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, V.1 N.4, August 1993

[25] Andrew S. Tanenbaum, *Computer Networks Third Edition*, Prentice Hall PTR

[26] Khan Javed I. & D**.** *Yun, Multi-resolution Perceptual Encoding for Interactive Image Sharing in Remote Tele-Diagnostics*, Manufacturing Agility and Hybrid Automation -I, Proceedings of the International Conference on Human Aspects of Advanced Manufacturing: Agility & Hybrid Automation, HAAMAHA'96, Maui, Hawaii, August 1996, pp183-187.

[27] Javed I. Khan & S. S. Yang, Resource Adaptive Nomadic Transcoder on Active Network, International Conference of Applied Informatics, AI 2001, February 19-22, 2001, Insbruck, Austria

[28] http://www.mpeg.org/MPEG/MSSG/#source