Scalable Multimedia Information Distribution over Heterogeneous Network

Wang Pei Jie, Shang Fuchun, Ding Ping, Tham Jo Yew, Lee Seng Luan

Abstract-- In heterogeneous network environment, the hardware capabilities of network devices are very different. For efficient utilization of system resources, it is important to transmit just enough information according to the hardware capabilities of different devices. We present a multimedia communication system that can provide different multimedia quality streams for different users at both the codec and system level. In our system, a mechanism for media quality negotiation and control is provided. The advantage of our system is that different sub-streams of different quality can be extracted from a same compressed stream, so that only one version of compressed media needs to be maintained in the server. This will save storage space and simplify media stream management. Scalable codec plays an important role in our system, and we shall describe how the scalability is achieved at the codec level.

Index Terms—Scalable, Multimedia, System

I. INTRODUCTION

Multimedia communication needs much more bandwidth and computing power. Even in very fast network environment, multimedia information needs to be compressed for transmission or storage. Therefore for a system to support multimedia communication, bandwidth and computing power are important resources. Efficient utilization of these resources is crucial to system performance, especially in heterogeneous network environment.

In a heterogeneous network environment, the computing power, display capabilities, display sizes and bandwidth available of network devices are very diverse. The difference in bandwidth can vary from less than 10kbits/s to more than 100Mbits/s. The difference in display resolution can be from 160 x 160 of PDA to 1600 x 1200 of a high end PC. The difference in display capabilities can be from black and white for a PDA to 32 bits true color for a PC. The computing power, display size and bandwidth for handphones are much lower than those for the PDA's. With the rapid development of Internet and wireless communication, heterogeneity will become the key feature of the multimedia communication environment.

In such a heterogeneous network environment, providing one level of media quality to all devices in the network is very inefficient, if not impossible. For example, MPEG1 video is suitable for a PC in high-speed network, but it cannot be delivered and rendered on a PDA's screen with acceptable quality. This is because of the following reasons. (i) The bandwidth for PDA is not enough for transferring so much information, and most of the packets including those that contain important information are dropped during transmission. This results in a waste of bandwidth in the network and causes unnecessary network congestion. (ii) Even if the bandwidth is enough, the PDA is not fast enough to decode the video stream in real time. This results in packet loss, and again a waist of bandwidth. (iii) Even if the bandwidth and computing power are sufficient, the display area of a PDA is not big enough for rendering full resolution MPEG1 video. Therefore some information is dropped, resulting in the waste both bandwidth and computing power. The ideal situation is one that provides different quality of media streams to different devices according to the user requirements or hardware capabilities. This would satisfy the user requirements and and at the same time reduce the load of network.

Some solutions have been proposed to solve the problem [9], [10]. For example, Bochmann described a system that has the ability to let users define the quality they want [9]. But in this system, the multimedia source is compressed into several versions to provide scalability. This will increase the storage space in the server. It is also difficult to define how many versions to be compressed, because the user's quality requirements are so diverse. Another system PADMA [10] provides similar functions. In this system, client and server can negotiate media quality according to the system and hardware capabilities. But

Center for Wavelets, Approximation and Information processing, Department of Mathematics, National University of Singapore, Singapore 119260
Email: matwpj@nus.edu.sg

.

this system can only support live transmission, that is, the change of quality is achieved by adjust the parameter of encoder. For the compressed multimedia stream, this flexibility is not available.

Our approach is to provide scalability on both the codec and system level. Here scalability means the ability to meet the diverse quality requirements simultaneously. The codec scalability will guarantee the extraction of appropriate subsets from the same compressed bit stream to reconstruct a version of the same video that conforms to certain specifications.

The architecture of the system is illustrated in Figure 1. The system can be divided roughly into server side and client side. The server will provide multimedia streams with the desired quality. The source can be from a media database where the pre-compressed multimedia streams resides, or from a video camera and/or microphone. If the source is from a video camera and/or microphone, the media streams need to be compressed in real time. On the client side user interface will be provided for users to choose the media items of required quality, including frame rate, bit rate, resolution and color depth. The audio and

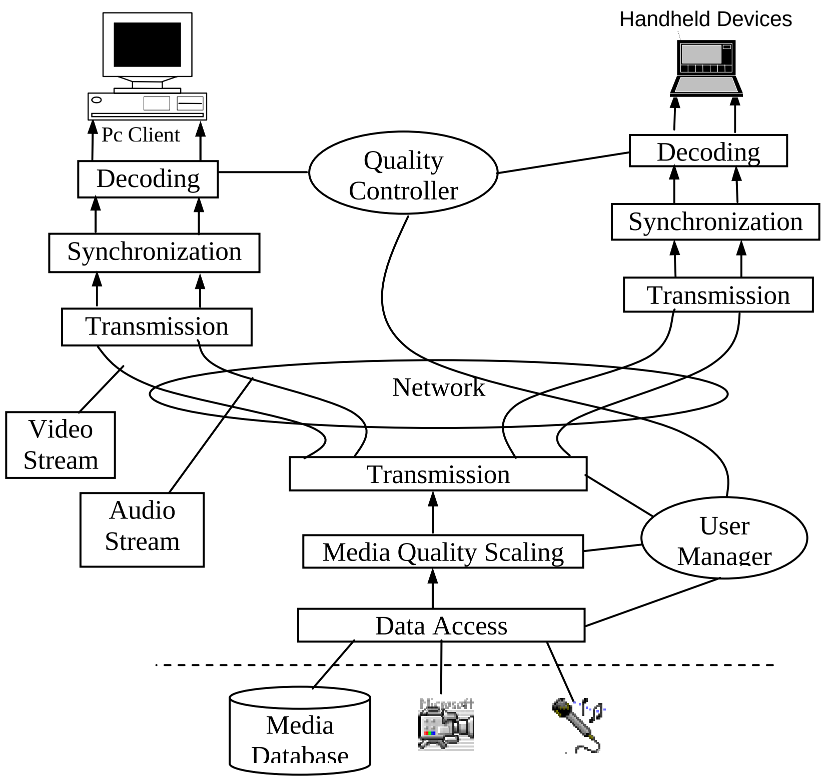

Figure 1. The Architecture of Scalable Multimedia Communication System

The system level will provide an infrastructure to support media quality negotiation, quality adjustment, sub-stream extraction, transmission and user profile management. This approach supports both live and pre-compressed transmission. In both cases, one server can simultaneously support several clients with different media quality requirements. For example, a PDA user can request the server to send a black and white QCIF resolution version of the video from a compressed video stream, while a PC user can request the server for a copy of full color CIF resolution video from the same compressed stream at same time. Moreover, only one version of the compressed multimedia stream needs to be maintained in server.

II. THE ARCHITECTURE OF THE SYSTEM

video are processed and transmitted separately and synchronized before decompressing at the client side. In his way users can choose only audio or video or both. At any one time, one server can satisfy the request of several clients with different quality requirements. In this case, different subsets of the same multimedia stream can be extracted to satisfy the user quality requirements.

Users need to register themselves by specifying the media quality required when they access the server the first time. The system will send a suitable media stream to each user according to the requirements. The system provides persistent mechanism to save this requirement, so that users need not specify the media quality in future access. The system will allow users to change the quality requirements any time.

The system is composed of several modules, including data access, quality scaling, user manager, quality controller, memory manager, synchronizer, decoder and transmission part. Each part is composed of one or several middleware objects that can be loaded dynamically. For specific applications users can choose the components needed.

The data access module is used to access data source. It is composed of several components that can be used for different situations, including different kinds of file reader, video and audio capturing, and video and audio encoding. If it is used to read multimedia files from the media database, a file reader component is used. If it is used to capture video and audio from a camera and microphone for live transmission, video/audio capturing and video/audio encoder are needed.

Due to the streaming feature of multimedia transmission, each object in the system will process the stream sequentially. Each object has defined a special interface for connecting to one another. After loading, all the objects will negotiate the media type that the system will support during connection time. Once connection is complete, the system is ready for multimedia information transmission.

III. MEDIA QUALITY NEGOTIATION AND CONTROL

The purpose of quality negotiation and control is to guarantee that users can get the media quality they required without burden to the network. To ensure this, users should be able to specify the media quality, the server should provide media stream according to the user requirements, and finally user specified quality should be least affected when system resources change. Coordinating the user manager, quality controller, and scaling module with the help of other modules can achieve this goal.

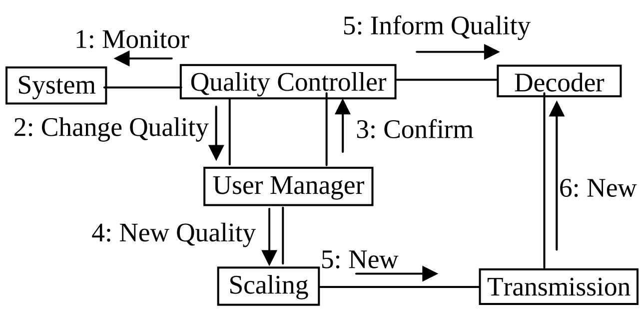

Figure 3. The interactions among objects when quality

The user manager provides user interface and management functions for users to specify their quality requirements. The quality controller provides functions to monitor and detect system quality capabilities. The scaling module is used to extract subsets of the multimedia stream with different media quality according to the user media quality requirements. When a user registers, the user manager will consult the quality controller whether the quality can be satisfied both on the server and client side. A successful registration means that the system has the ability to satisfy the user requirements. Otherwise the user should adjust the requirements. This will guarantee the user quality requirements at beginning. The interaction among modules at this stage is illustrated in Figure 2.

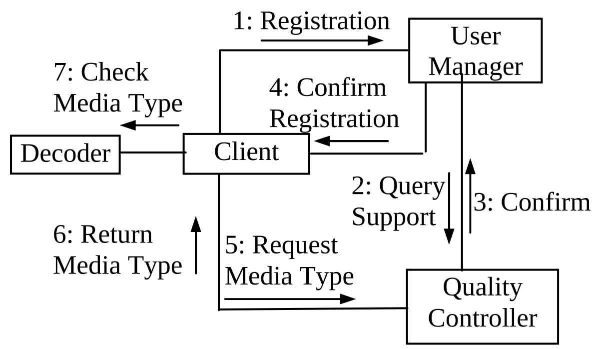

Figure 2. The interaction among objects when client side starts up

The user manager and scaling module will be responsible for providing suitable media stream according to user requirements after successful registration. The user manager can also communicate to the data access module the quality of the multimedia content they should provide in some cases. For example, in real-time communication, a video camera need only to capture a video sequence with the minimum quality that can satisfy all the user requirements at one time, and need not provide the best quality it can capture. This will enable a more efficient use of system resources. The scaling module may produce several new media streams with different quality from same original media stream according to the user requirements specified by the user manager. The sub-streams will be sent to destination by the transmission module according to the user network address in the user manager.

The quality controller is responsible to monitor whether the quality required by the user can be achieved. Some times the quality needs to be adjusted. One possibility is that the user altered the quality requirements. Another possibility is for the quality controller to adjust dynamically the quality according to the user preference to adapt the fluctuation of the available bandwidth. In the case of user adjusted quality, the user manager will inform the quality controller and scaling module the quality changes as illustrated in Figure 2. In the case of quality controller adjusted quality, the quality controller will be responsible for informing the user manager and decoder the quality changes. The interaction among objects is illustrated in Figure 3.

IV. SCALABLE CODECS

A. *Wavelet-based Scalable Video Codec*

Our video codec is a highly scalable codec that provides full spectrum of scalability, including bandwidth, distortion, spatial resolution, temporal resolution and colour depth scalability. Bandwidth scalability, also referred to as bit rate scalability, allows a party (either a source producer or consumer) to gracefully scale for a wide range of bit rates from the same scalable video source. Distortion scalability also refers as signal-to-noise ratio (SNR) scalability, allows a party to trade for a different level of quality (or fidelity) of the video from a common compressed video bit stream. Video spatial resolution scalability refers to the flexibility to support different display resolutions (or picture sizes) by means of selecting different pertinent subsets of a common compressed video bit stream. Video temporal resolution scalability defines the flexibility to choose different picture frame rates for playback from a common compressed video source.

Figure 4 shows a general overview of the wavelet-based scalable video codec. The stream of incoming video frames are first partitioned into distinct groups of frames (GOF) of *F* frames each, where *F* = 16 as an example. It is also noted that each color video frame is first transformed from the RGB to the YUV 4:1:1 color domain prior to any processing.

Stage 1 of the encoder aims to exploit the temporal correlations of the frames within a GOF. The first frame in a GOF is used as a reference frame, where succeeding frames in the GOF are "matched" with respect to the reference frame by estimating a set of block motion vectors for each frame using a fast block-based motion estimation algorithm (e.g., [8]). In Stage 2, a 3-D dyadic discrete wavelet transform is performed on the motion-compensated frames. Figure 5 shows the resulting wavelet structure of one of the temporally-transformed frames with three spatial scales, where subband 0 has the lowest spatial resolution. We employ the biorthogonal 9-7 wavelets [1], [2], as this filter bank gives comparatively less ringing artifacts at low bit rates [1]. Finally in Stage 3, a new scalable data structure called *tri-zerotrees* (TRI-ZTR) is proposed to effectively encode the GOF into an embedded and scalable compressed video bit stream. A TRI-ZTR interband relationship is formed [3] and a successive refinement/layered coding

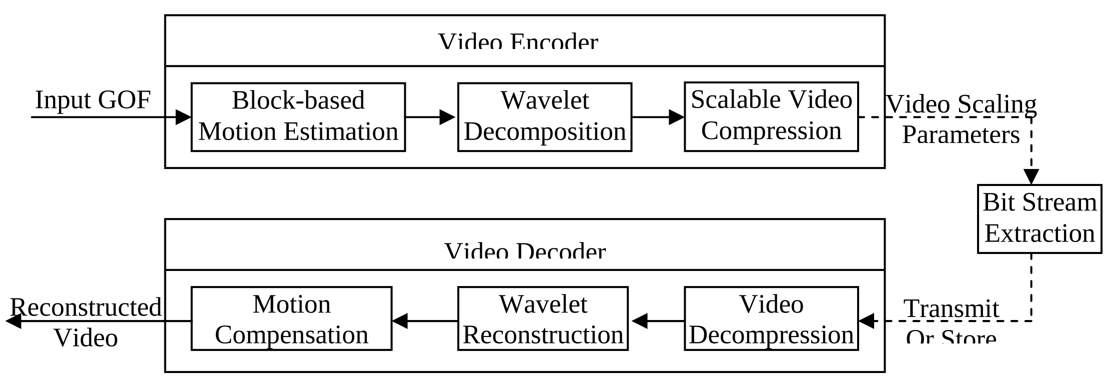

Figure 4: Overview of the wavelet-based scalable video encoder

strategy [4,5,6,7] is used, where the more important wavelet coefficients are selected first and encoded in multiple embedded layers --- each coding layer adds another bit of precision to their magnitudes. To achieve multiresolution scalability, we introduce the idea of *resolution blocking* by means of encoding certain partitioning information (called *resolution flags*) into the bit stream. Partial bit stream extraction can be carried out based on a given set of video scaling parameters. The new downscaled bit stream is then transmitted to the decoder for reconstruction. It is clear that the decoder essentially performs the inverse processes of the three main stages of the encoding part, but in the reverse order of processing. It is noted that the received video bit stream can also be stored at the decoder, and then be further downscaled, if necessary.

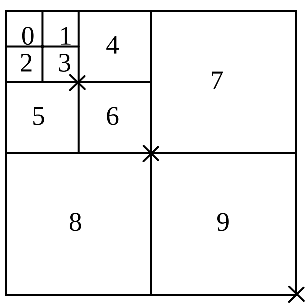

Figure 5. Pyramid wavelet decomposition structure (cross mark indicating where the resolution flags are inserted for multiresolution scalability)

The two main concepts of the proposed highly scalable video codec are based on the ideas of *embedded coding* and *resolution block coding* (comprising both spatial resolution and temporal frame blocks). Figure 6 illustrates a typical scalable compressed bit stream where multiple resolution/frame blocks constitute subsets of the bit stream that are arranged in an embedded manner according to multiple coding layers.

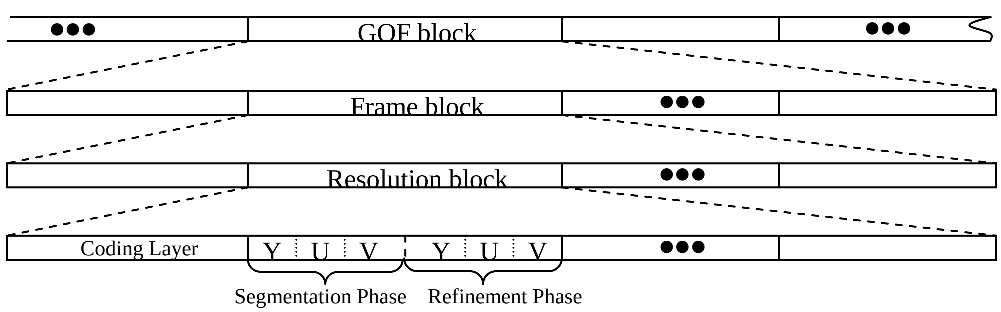

Figure 6. A typical scalable compressed bit stream comprising embedded resolution/frame blocks.

B. Scalability and Re-scalability

Bandwidth scalability is achieved by extracting only the pertinent subsets of the bit stream corresponding to the first few coding layers. Since the coding layers are encoded in successive segmentation and refinement phases, the bit stream can be truncated at any arbitrary point so as to meet a specific bit rate. Also, distortion scalability can be achieved easily. This is possible because the embedded coding strategy has indirectly arranged the information of the video in a manner such that more important information are encoded in earlier coding layers and subsequent coding layers act as enhancement layers. By choosing subsets of resolution blocks from more coding layers, we can gracefully scale for higher quality videos. Spatial resolution scaling can also be easily supported. As pointed out in Figure 5, resolution flags are inserted into the bit stream to demarcate resolution blocks that belong to different spatial resolutions/scales. With this explicit partitioning information, we can now choose to decode only those subsets of resolution blocks that belong to a desired spatial resolution. Temporal resolution scaling is also possible. A lower playback frame rate can be achieved by discarding frame blocks that correspond to higher temporal scales within a temporally-transformed GOF. This is possible because each frame block only comprises information that corresponds a particular temporal scale within the transformed GOF. Clearly, colour depth scalability is also possible. A grayscale video can be chosen from the same compressed bit stream by decoding only those resolution blocks that correspond to the "Y" component of the video. Finally, it is worth emphasizing again that all the above video scaling parameters were achieved from the same compressed video *without* having prior knowledge of the actual video playback specifications that would later be chosen while the scalable video bit stream is being generated.

C. Scalable Audio Codec

Compared with video stream, an audio stream takes up only a small part of the bandwidth. In normal cases, the scalability in video codecs is more effective for improving system performance. But in extremely low bit-rate situation, scalability in audio becomes useful. Our speech codec provides two levels of scalability.

Speech can be separated apart as voiced and unvoiced, and some parts of a speech are mixtures of the two. The voiced speech has relatively high-energy content and periodicity, while the unvoiced speech looks like random noise without periodicity. Linear Predictive Coding (LPC) [11], [12] is an efficient coding method to quantize the coefficients of the all-pole model. We combine LPC and frequency-domain harmoniccoder for voiced speech and time-domain waveform coder for “noise like” unvoiced speech.

The compressed speech stream can be disparted as two parts: the first 2k bits and the following 2k bits. The first 2k bits can satisfy the low bandwidth requirements. By adding the following 2k bits, a higher quality version of speech can be reverted for higher bandwidth. The following two paragraphs explain how this scalability is achieved.

The most important parameters are 10 LPC parameters. Usually every frame needs 34 bits to represent 10 LPC parameters. To decrease the bit rates, the coefficients are quantized twice, which results in representing 10 LPC parameters with 18 bits/frame. The result is put to the first 2k bits. LPC parameters are decided by matching using a quantization table. So the 10 LPC parameters can be reverted by looking up the same quantization table during decompression. Some errors are caused because the 10 LPC parameters have been quantized twice. In order to produce scalable stream, the errors are also coded and put to the second part of the stream as a supplement of the first part.

The voiced speech is quantized after removing the correlation, which can be reverted by adding pitch during decompression. For unvoiced speech, it can be reverted only by looking up its quantization table because unvoiced speech bears little correlation. Both processes cause errors. For the same reason, the errors are coded and put to the second part.

V. SUMMARY

We have presented a multimedia communication system that can provide different multimedia quality streams for different users by providing scalability at both the codec and system level. In our system, a mechanism for media quality negotiation and control is provided. The advantage of our system is that different sub-stream different qualities can be extracted from a same compressed stream, so that only one version of the compressed media needs to be maintained in the server. This will save storage space and simplify media stream management. Scalable codec plays an important role in our system, and a brief description of the code is presented..

VI. REFERENCE:

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” IEEE Trans. on Image Proc., vol. 1, pp. 205-220, Apr. 1992 .

[2] A. Cohen, I. Daubechies, and J. C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” Comm. Pure Appl. Math., vol. 45, pp. 485-560, 1992.

[3] J. Y. Tham, S. Ranganath, A. K. Kassim, "Highly scalable wavelet-based video codec for very low bit rate environment," IEEE Journal on Selected Areas in Communications -- Special Issue on Very Low Bit-rate Video Coding, vol. 16, no. 1, pp. 12-27, Jan. 1998

[4] J. M. Shapiro, “An embedded wavelet hierarchical image coder,” Proc. IEEE Intl. Conf. on ASSP, vol. 4, pp. 657-660, Mar. 1992.

[5] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” IEEE Trans. Sig. Proc., vol. 41, no. 12, pp. 3445-3462, 1993.

[6] A. Said and W. A. Pearlman, “Image compression using the spatial-orientation trees,” IEEE Intl. Symp. Circts. and Systs., Chicago, pp. 279-282, May 1993.

[7] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” IEEE Trans. on Circts. Systs. Video Tech., vol. 6, no. 3, pp. 243-250, June 1996.

[8] J. Y. Tham, S. Ranganath, M. Ranganath, A. K. Kassim, “A novel unrestricted center-biased diamond search for block motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 4, pp. 369-377, Aug. 1998.

[9] V. Bochmann, G.; Kerherve, B.; Hafid, A.; Dini, P.; Pons, A., “Architectural design of adaptive distributed multimedia systems” Proceedings., International Workshop on Multimedia Software Development, 1996 , Page: 31 -40

[10] Khan, S.; Li, K.F.; Manning, E.G. “Padma: an architecture for adaptive multimedia systems”, 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Page(s): 105 -108 vol.

[11] F.Soong and B. H. Juang. “Line spectrum pairs and speech data compression” Prop. Of ICASSP, 1.10.1-1.10.4, 1984.

[12] B. Atal, R. B. Cox, and P. Kroon. “Spectral quantization and interpolation for CELP coders”. Prop. Of ICASSP, pp69-072, May 1989.