A Multi-resolution Video Scheme for Multimedia Information Servers in Mobile Computing Environment

Jinsung Cho and Heonshik Shin

Department of Computer Engineering
Seoul National University
Seoul 151-742, Korea
Tel: +82-2-880-7295, email: {cjs,shinhs}@comp.snu.ac.kr

ABSTRACT

Nowadays we envisage a large number of clients carrying portable multimedia terminal to access multimedia information over wireless links in the near future. The realization of such services requires the development of multimedia information servers that provide multimedia streams for mobile clients. This paper aims at identifying the server requirements in mobile computing environment and present a novel multi-resolution video layout scheme for the requirements. The proposed scheme is based on the multi-resolution video stream model which can be created by various scalable compression algorithms. Simulation studies show its optimality in terms of disk load balancing and its adaptability to the low and often fluctuant network bandwidth, the QoS change of video streams, and the request for interactive operations.

I. INTRODUCTION

Recent advances in computer technology and demands of video, audio, and text integration services have been the driving forces behind the emergence of various multimedia information services. The realization of such services requires the development of *multimedia information servers* that support efficient mechanisms for storing and delivering multimedia streams. The fundamental problem in developing multimedia servers[1] is that the delivery and playback of multimedia streams must be performed at real-time rates[1]. Recent researches on multimedia servers are focused on the guaranteed retrieval and delivery of continuous media such as video and audio[2], data distribution and disk scheduling schemes for the high throughput of storage subsystem[3], efficient buffer management[4], admission control and QoS management techniques for the deterministic service guarantee[5], and the scalable server architecture and scheduling algorithms in parallel or distributed environment[6], [7]. However, they assume that clients are connected to the server by fixed high-speed network capable of delivering the required bandwidth of multimedia streams.

Further advance in personal computer hardware and wireless communication technologies has created new computing environment, or *mobile computing environment*[8], [9]. In the future mobile computing environment, a large number of clients carrying a low-power palm-top machine will access multimedia information over wireless links anywhere and at any time[10]. The anticipated applications and services in mobile computing environment will include personalized information, travel information, electronic magazines, shopping, and electronic mail services. It is taken for granted that a challenging area of mobile computing is the design of a server that provides mobile clients with necessary multimedia services.

In this paper, we design a multimedia information server in mobile computing environment. The remainder of this paper is organized as follows: Section II describes the characteristics of mobile computing environment and specifies their impact on multimedia servers. In Section III, we introduce our server architecture and propose the multi-resolution video scheme. Section IV analyzes the scheme for its properties and optimality in terms of disk load balance. The empirical evaluation for the scheme is given in Section V, and Section VI describes our implementation work for the realization of the proposed scheme. Section VII concludes this paper.

II. IMPACT OF MOBILITY ON MULTIMEDIA INFORMATION SERVERS

Characteristics of mobile computing environment are due to the wireless medium and mobile hosts. The former is characterized by low bandwidth, frequent disconnections, and high bandwidth variability and the latter by limited computing power, small screen and size, and limited battery life[8], [9]. Considering these characteristics, we must cope with many challenging problems when we attempt to design multimedia servers for use in mobile computing. Since most existing multimedia servers have been designed for fixed clients, providing multimedia services for mobile clients involves modifying the server architecture to accommodate the constraints induced by mobile clients.

First, we begin by introducing a system model of typical mobile information services. In Figure 1, *mobile support stations* (MSSs) are fixed hosts and equipped with wireless interface. An MSS serves as a gateway between wired and wireless medium and covers the designated *wireless cell* area. A *mobile client host* (MCH) is a portable multimedia terminal that can move while retaining its network connection through wireless communication. Via MSSs, MCHs access services located at the *mobile multimedia information server* (MMIS) over the high-speed communication backbone. MMIS can be distributed or replicated geographically. As the under-

[1] In this paper, the terms multimedia information server and multimedia server are used interchangeably.

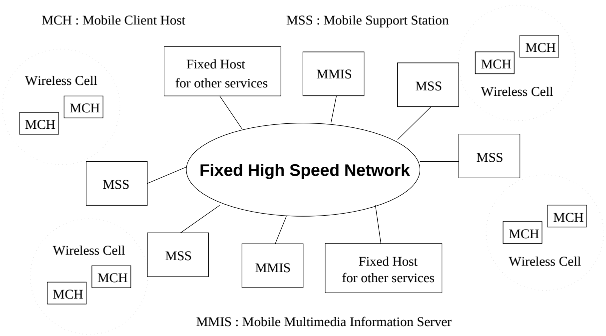


Fig. 1. The system model of typical mobile information services

lying communications system is responsible for the data delivery over wired and wireless network and the location management of mobile clients, we will focus on the design of MMIS in this paper.

In multimedia information systems, continuous media (especially video) is probably the most demanding data type because of its large data size and rather strict timing constraints. Although network bandwidth is relatively low in mobile computing environment, multimedia streams must be continuously presented and synchronized. Besides continuous and synchronous playback of multimedia streams, we identify the following requirements that should be met while designing a MMIS:

- *QoS management of multimedia streams.* Each MCH is connected to the system with available network bandwidth, depending on whether it is plugged in or using wireless access and on the type of connection at its current cell. In addition, MCHs have different capabilities in manipulating video streams, such as color depth, window size, and frame rate. Hence, clients are likely to request various qualities even for the same video stream. MMIS should effectively manage the various QoS level of multimedia streams.
- *Adaptability to high bandwidth variability.* Fluctuation in network bandwidth is one of the fundamental problems in the field of mobile computing. Furthermore, wireless networks have higher error rates and are prone to frequent disconnections which result from communication failure, battery limitations, and voluntary disconnections. Thus MMIS should be able to adapt itself to currently available resources, providing clients with gracefully degraded quality.
- *Support for interactive operations.* In mobile information services, interactive operations (e.g., pause, resume, reverse, rewind, and fastforward) are far more significant than static services like VOD. Clients are likely to pause or rewind for more detailed information and to search streams for required information via fastforward. MMIS should support these operations without additional costs such as extra disk and network bandwidth.

III. THE PROPOSED SCHEME

The most demanding resources in traditional multimedia servers are I/O (disk and network) bandwidth and storage. So far most studies on multimedia servers have concentrated on the distribution of video streams across multiple disks for high degree of parallelism and the effective management of disk bandwidth. This is because the bandwidth of disk subsystem is relatively lower than that of network in static environment. In the mobile computing environment, however, such an imbalance is not true any more: A network subsystem has much lower bandwidth. Therefore the design of MMIS should depend on the requirements described in Section II rather than the effective management of disk bandwidth.

A. Architecture of the server

As mentioned earlier, most multimedia servers adopt a disk array-based architecture for large bandwidth and load balancing of disks[1]. We also assume a disk array model for the server architecture. The major concern is to place disk blocks across disks, satisfying the new requirements given in Section II while keeping the large bandwidth and load balancing of disks. Considering the scale of the system, however, our disk array model can be implemented either in a single computer equipped with multiple disks[11] or in a distributed[7] or parallel[6] system which consists of multiple nodes.

B. Multi-resolution video stream model

In general the notion of video resolution is defined in three dimensions: chroma, spatial, and temporal. In these dimensions, video streams can be compressed into multi-resolution format by various scalable compression algorithms, however, any of which may be used for the purposes of our argument. A multi-resolution video stream is a video sequence encoded such that subsets of the full resolution video bit stream can be decoded to recreate lower resolution video streams. For modeling multi-resolution video streams, we propose a z-level multi-resolution video stream model. For a video stream S_i,

$$S_i = \{G_s^i \mid G_s^i \text{ is a segment}, \ 0 \le s < m\}$$

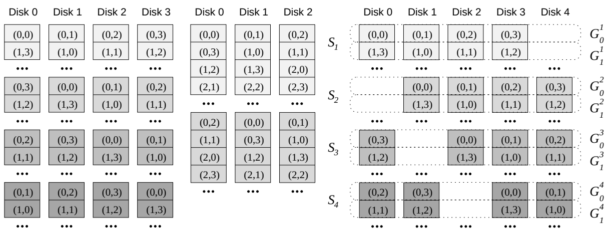
$$G_s^i = \{(s,c) \mid (s,c) \text{ is a component}, \ 0 \le c < z\},$$

where s and c denote the segment number and the component number, respectively. m is the number of segments, or the length of the stream. The k-level resolution can be obtained by integrating k components from the lowest one; $S = \{(s,c) \mid 0 \le s < m, \ 0 \le c < k\}$ are serviced. In our multi-resolution stream model, each video stream can be provided with z levels of quality and the QoS parameter is represented by the number of components in a segment, or k. All the components constitute the full resolution quality.

The multi-resolution video stream given above can be implemented by various coding technologies. The current multi-resolution or scalable video compression techniques include DCT-based schemes, subband(wavelet) schemes, fractal-based schemes, and object-based schemes[13]. All the schemes well match our multi-resolution video model. Consequently, we can utilize 'off-the-shelf' technology to implement our multi-resolution video stream model.

C. Multi-resolution video layout

We now intend to distribute multi-resolution video streams across disks while keeping high bandwidth and



(a) $z = d$ (b) $z > d$ (c) $z < d$

Fig. 2. An example of multi-resolution video layout

load balancing of disks regardless of the QoS of requested streams. We begin by introducing an example of multi-resolution video layout in Figure 2. Three cases can be identified according to the resolution of stream, z, and the number of disks, d.

First, in case of $z = d$, where the number of disks is equal to the QoS level, components in a segment are distributed across all the disks and successive components are placed on consecutive disks[2]. Afterwards, for the disk load balancing, the first segments of successive segments G_i and G_{i+1} (i.e., $(i, 0)$ and $(i+1, 0)$) are assigned on consecutive disks(disk j and disk $j + 1$), as shown in Figure 2(a). In addition, we distribute the starting point (the first component in the first segment, or $(0, 0)$) of each stream for the load balancing of disks when multiple streams are requested concurrently. Second, when $z > d$, multiple components in a segment may be placed on a disk. However, the strategy is similar to the case of $z = d$. That is, successive components in a segment are placed on consecutive disks and the first components of successive segments are assigned on consecutive disks as depicted in Figure 2(b). Third, when $z < d$, a video stream is allocated to only z disks and idle disks are evenly distributed, as shown in Figure 2(c). Then, this case is logically equivalent to the case of $z = d$.

The proposed multi-resolution video layout allows deterministic access to disks, which is the most significant characteristic for the continuity requirement of video streams, and achieves the load balance of disks while providing various levels of QoS. We prove the optimality in terms of the disk load balancing in the next section.

We now formulate our layout scheme. For a video stream $S = \{(s, c) \mid 0 \le s < m,\ 0 \le c < z\}$, and the starting point of S being disk j, the components stored in disk i are as follows:

$$D_i = \{(s, c) \mid 0 \le s < m,\ c = (i + j - s) \bmod d + l \cdot d,\ 0 \le l < \left\lceil \frac{z}{d} \right\rceil\} \quad \text{if } z \ge d. \tag{1}$$

Since the layout for $z < d$ is logically equivalent to that for $z = d$, we will consider the case where $z \ge d$ for the analysis of our scheme.

IV. ANALYSIS

A. Properties of the proposed scheme

In this subsection, we describe how the proposed scheme satisfies the requirements of MMIS identified in Section II.

- *QoS management of multimedia streams.* Our scheme handles various levels of QoS for each video stream. For each QoS request for video streams, our MMIS only calculates the required QoS parameter in our multi-resolution video model. Based on the QoS value, the requested resolution of service proceeds.
- *Adaptability to high bandwidth variability.* First of all, MSS should provide MMIS with some information like changes in bandwidth of connection with MCH. Furthermore, if necessary, MMIS notifies the changed information to MCH. Our MMIS can adapt to the network bandwidth variability by recalculating QoS parameters according to the currently available resources. QoS changes can also be accomplished through the negotiation with clients without any overhead.
- *Support for interactive operations.* We are not concerned with operations like pause and resume because they do not require additional disk and network bandwidths. MMIS should be able to support rewind and fast-forward operations within the given bandwidth. In our layout scheme, such operations can be achieved by degrading the QoS of the requested stream and retaining the transmission rate. For example, let us assume that a fast-forward operation is requested for a video stream with the fourth level resolution. If we lower the QoS level to the second level and the first level, the two-times-fastforward and the four-times-fastforward can be achieved, respectively, without any additional disk and network bandwidth. For the case of low level QoS where it is impossible to degrade the level, a segment skipping scheme[14] can be integrated with our scheme.

In summary, by simply adjusting the QoS parameter, the proposed scheme meets the requirements of MMIS. Besides the properties described above, our scheme can provide services with more clients by the flexible admission control. A new request rejected by the admission control strategy may be serviced by gracefully degrading the quality of existing requests.

B. Optimality of the proposed scheme

Workloads for client requests should be evenly distributed across all disks, so that the server is able to fully utilize the bandwidth of disk subsystem. The proposed layout scheme in this paper is optimal in the sense that, regardless of the requested QoS, the disk workload generated for a video stream is evenly distributed.

In the proposed scheme, video streams are stored by the component, of which the size may be different from each other. Let $x_{s,c}$ be the number of disk blocks for (s, c) component (the cth component of the sth segment). For the service of a video stream $S = \{(s, c) \mid 0 \le s < m,\ 0 \le c < z\}$ with k-level QoS ($k \le z$), the server retrieves a set of components $\{(s, c) \mid 0 \le s < m,\ 0 \le c < k\}$, which requires $\sum_{s=0}^{m} \sum_{c=0}^{k-1} x_{s,c}$ disk blocks.

Current compression techniques yield variable bit rate streams, so that it is impractical to let the size of all components be the same (i.e., $x_{s,c} = x$ for $0 \le s < m, 0 \le c < z$). In [15], Shenoy and Vin suggest a scheme in which the block size can vary across sub-streams(lower resolution streams) but is fixed for a given sub-stream

[2]The consecutive disk of disk $d - 1$ is disk 0.

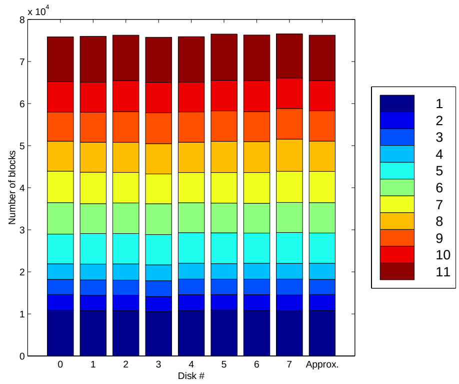


Fig. 3. Optimality in disk load balancing $(m = 1800, d = 8)$

to maximize performance. This can be realized in our multi-resolution video stream model by fixing the number of blocks in the same component level, or $x_{s,c} = x_c$, $0 \leq s < m$.

Intuitively, in our scheme, the number of the ith components retrieved from each disk is m/d, since we evenly distribute components in a segment across the disks. Considering the number of segments is far greater than the number of disks or $m \gg d$, we assume that $m = \alpha \cdot d$. For the service of S with k-level QoS, then, the number of disk blocks retrieved from disk i is

$$\sum_{c=0}^{k-1} \left(\sum_{s\ of\ (s,c) \in D_i} x_{s,c} \right) = \sum_{c=0}^{k-1} \alpha x_c \qquad (2)$$

where D_i is given in Eq. (1). Hence, the proposed scheme uniformly distributes the disk workload generated for a video stream regardless of the requested QoS.

Even when $x_{s,c}$ $(0 \leq s < m)$ is variable in the same component level, it is expected that the number of disk blocks retrieved from each disk converges to $\sum_{c=0}^{k-1} \alpha x_{avg,c}$, where $x_{avg,c} = \frac{1}{m} \sum_{s=0}^{m-1} x_{s,c}$. Figure 3 represents the experimental results. For this experiment we employ trace data from the multi-resolution video codec given in [16]. The codec is based on 3-D subband coding and multi-rate quantization of subband coefficients, followed by arithmetic coding. The trace data has 11 levels of resolution. We can find from Figure 3 that the proposed scheme evenly distribute the disk workload even for the variable bit rate streams.

V. SIMULATION

For the empirical evaluation of the proposed scheme, we have created an event-driven simulator written in C with SMPL[17] libraries. This simulator models the multimedia server including the disks, data layout, video streams, and the server software such as schedulers. The server is assumed to store 6 types of 60 video streams of which the stream rates are between 0.5 Mbps and 1.5 Mbps. Clients randomly choose the video streams which are laid out by the multi-resolution video layout scheme proposed in this paper. We assume that the resolution level of video streams is equal to the number of disks, or $z = d$, since the other cases ($z < d$ and $z > d$) shows the same properties and optimality in terms of disk load balance as described in Section IV. Each client gives a QoS parameter and may change it during the service of video streams.

TABLE I
Parameters used in the simulation

Description	Symbol	Value
Stream rate	R_{stream}	0.5 ∼ 1.5 Mbps
Disk transfer rate	R_{disk}	10 MB/s
Disk block size	S_{block}	1 KB
Disk access overhead	$T_{overhead}$	15 ms
Number of disks	d	8
Resolution level	z	8

Considering the scalable server architecture in parallel or distributed environment, we model the disk access time for the retrieval of (s, c) component, $T_{disk}(s, c)$, simply as follows:

$$T_{disk}(s, c) = T_{overhead} + \frac{S_{block}}{R_{disk}} \times x_{s,c}, \qquad (3)$$

where $T_{overhead}$, S_{block}, and R_{disk} denote the overhead for disk access, the disk block size, and the transfer rate of a disk, respectively. $T_{overhead}$ includes disk seek time, scheduling overhead, etc., and is assumed to have the exponential distribution with the mean value of 15 ms. Although we use 10 MB/s for R_{disk} in this simulation, it may be increased in parallel or distributed servers where each node is equipped with a high performance disk subsystem such as RAID. We summarize the simulation parameters and their values in Table I.

Figure 4 illustrates the adaptation of the proposed scheme to the variation produced by mobile computing environment including the fluctuation of network bandwidth, the QoS change of video streams, and the request for interactive operations. For this experiment, we adopt the following scenario:

(a) Initially, three clients, A, B, and C, request video streams S_A, S_B, and S_C, of which the stream rates are 0.5 Mbps, 1.5 Mbps, and 0.9 Mbps, respectively. A and C request the full resolution quality and B does the half resolution quality.
(b) After 5 minutes, the network bandwidth with A decreases to 0.3 Mbps.
(c) After 5 minutes, the network bandwidth with A decreases to 0.1 Mbps.
(d) After 2 minutes, the network bandwidth with A increases to the original value, or 0.5 Mbps.
(e) After 3 minutes, B raises the quality of S_B to the full resolution.
(f) After 5 minutes, B lowers the quality of S_B to the half resolution.
(g) After 5 minutes, C requests a two-times-fastforward operation.
(h) After 1 minutes, C requests a four-times-fastforward operation.
(i) After 1 minutes, C requests the normal playback.
(j) After 1 minutes, a new client D requests S_D with the half resolution quality.

Figure 4 shows that our scheme fulfills the requirements of mobile computing environment simply by re-

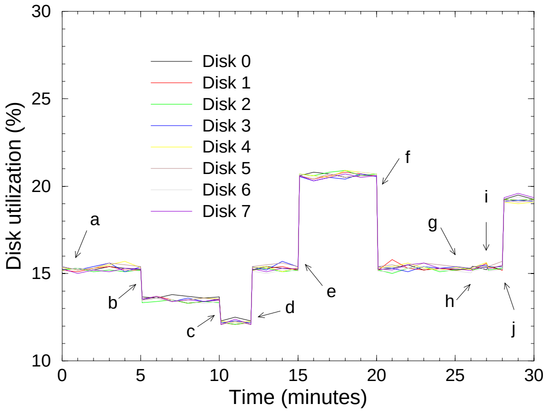


Fig. 4. Adaptive behavior of the proposed scheme. *Our scheme well adapts to the fluctuation of network bandwidth, the QoS change of video streams, and the request for interactive operations. The disk utilization is obtained from dividing the disk service time by the measured period of time.*

calculating the required QoS parameters, as described in Subsection IV-A.

VI. IMPLEMENTATION

For the purpose of realizing the proposed technique, we have implemented the prototype of multi-resolution VOD services, although it does not have mobile connections. The multi-resolution video manager is implemented as a server process in QNX real-time microkernel operating systems[18] on a Pentium PC with a AHA-1540CP SCSI adapter and 4 Quantum 850MB SCSI disks.

To employ existing codecs, we have realized the multi-resolution video model using MPEG-1 video in temporal dimension. In the first prototype of the server, the QoS level of a video stream is provided with a high, medium, or low resolution. For our multi-resolution video model, MPEG-1 video streams are parsed and separated by the frame type, and then, a segment is made up of a GOP and a set of the same type frames in the GOP constructs each component in a segment, for example, (I_1), (P_1, P_2, P_3, P_4), $(B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10})$. By the multi-resolution video layout scheme, the reconstructed MPEG-1 video is placed on disks. The multi-resolution video manager performs this work and provides the retrieval service with a given QoS parameter. In the retrieval, when necessary, the frame sequence is reorganized into the original, so that the existing MPEG hardware/software decoder can work.

Based on the multi-resolution video manager, a VOD application has been developed. Our prototype exhibits that the multi-resolution playback and fastforward playback can be achieved, although the client program runs with fixed connections. This gives us more insights into our technique when extended to a practical environment with mobile connections.

VII. CONCLUSIONS

In this paper, we have specified the additional requirements of multimedia information servers in mobile computing environment: *QoS management of multimedia streams*, *adaptability to high bandwidth variability*, and *support for interactive operations*. For the requirements, we have proposed a multi-resolution video stream model and disk layout scheme. The video stream model can be implemented by various scalable compression algorithms.

Simulation results have shown that the proposed scheme effectively adapts to the variation induced in mobile computing environment by adjusting the QoS parameters of the multi-resolution video streams and effectively utilizes the disk bandwidth. In addition, we have implemented a prototype to present the practicality of our scheme. In the prototype, multi-resolution video streams are constructed from MPEG-1 video streams and can be played back with a high, medium, or low resolution. Fastforward playback can be also achieved. Finally, our scheme may service a new request rejected by the admission control strategy, by gracefully degrading the quality of existing requests. The flexible admission control remains to be addressed in the future.

REFERENCES

[1] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe, “Multimedia storage servers: A tutorial,” *IEEE Computer*, vol. 28, no. 5, 1995, pp. 40–49.

[2] D. P. Anderson, Y. Osawa, R. Govindan, “A file system for continuous media,” *ACM Transactions on Computer Systems*, vol. 10, no. 4, 1992, pp. 311–337.

[3] A. L. N. Reddy, “Disk scheduling in a multimedia I/O system,” *Proc. of ACM Multimedia 93*, 1993, pp. 225–233.

[4] K.-L. Wu and P. S. Yu, “Consumption-based buffer management for maximizing system throughput of a multimedia system,” *Proc. of International Conference on Multimedia Computing and Systems*, 1996, pp. 164–171.

[5] H. M. Vin, P. Goyal, A. Goyal, and A. Goyal, “A statistical admission control algorithm for multimedia servers,” *Proc. of ACM Multimedia 94*, 1994, pp. 33–40.

[6] J. Cho and H. Shin, “Scheduling video streams in a large-scale video-on-demand server,” *Parallel Computing*, vol. 23, no. 12, 1997, pp. 1743-1755.

[7] C. S. Freedman and D. J. DeWitt, “The SPIFFI scalable video-on-demand system,” *Proc. of 1995 ACM SIGMOD*, 1995, pp. 352–363.

[8] T. Imielinski and B. R. Badrinath, “Mobile wireless computing: Challenges in data management,” *Communications of ACM*, vol. 37, no. 10, 1994, pp. 18–28.

[9] G. H. Forman and J. Zahorjan, “The challenges of mobile computing,” *IEEE Computer*, vol. 27, no. 4, 1994, pp. 38–47.

[10] S. Sheng, A. Chandrakasan, and R. W. Brodersen, “A portable multimedia terminal,” *IEEE Communications Magazine*, vol. 30, no. 12, 1992, pp. 64–75.

[11] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, “Streaming RAID^{TM} – A disk array management system for video files,” *Proc. of ACM Multimedia 93*, 1993, pp. 393–400.

[12] ISO/IEC 13818, “Information technology - generic coding of moving pictures and associated audio -”.

[13] J. Hunter, V. Witana, and M. Antoniades, “A review of video streaming over the internet,” White paper, `http://www.dstc.edu.au/RDU/staff/jane-hunter/video-streaming.html`.

[14] M.-S. Chen, D. D. Kandlur, and P. S. Yu, “Support for fully interactive playout in a disk-array-based video server,” *Proc. of ACM Multimedia 94*, 1994, pp. 391–398.

[15] P. J. Shenoy and H. M. Vin, “Efficient support for interactive operations in multi-resolution video servers,” *Multimedia Systems*, accepted for publication, 1999.

[16] D. Taubman and A. Zakhor, “Multirate 3-D subband coding of video,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, 1994, pp. 572–588.

[17] M. H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987.

[18] “QNX - System architecture,” *QNX Operating Systems Manual*, QNX Software Systems Ltd., 1993.