RMI System: Internet Meets the Future Home Theater

Roger Zimmermann, Chris Kyriakakis, Cyrus Shahabi, Christos Papadopoulos, Alexander A. Sawchuk, and Ulrich Neumann
University of Southern California

The Remote Media Immersion (RMI) system blends multiple cutting-edge media technologies to create the ultimate digital media delivery platform. Its streaming media server delivers multiple high-bandwidth streams, transmission resilience and flow-control protocols ensure data integrity, and high-definition video combined with immersive audio provide the highest quality rendering.

The charter of the Integrated Media Systems Center (IMSC) at the University of Southern California (USC) is to investigate new methods and technologies that combine multiple modalities into highly effective, immersive technologies, applications, and environments. One result of these research efforts is the Remote Media Immersion (RMI) system. RMI's goal is to create and develop a complete aural and visual environment that places participants in a virtual space where they can experience events that occurred elsewhere. RMI technology realistically recreates, on demand, the visual and aural cues recorded in widely separated locations.[1]

The RMI system is the next step in audio-visual fidelity for streaming media delivered on demand over the Internet. The RMI system pushes the boundaries beyond what's currently available in any commercial system or other research prototype. Its emphasis is on the highest quality of audio-visual experiences and realistic, immersive rendering. To achieve our goal, we were faced with numerous challenges and had to design novel techniques to make RMI a reality.

In this article, we detail some of the RMI components and the techniques that we employ within each (focusing mainly on the transmission and rendering aspects). The hope is that our advances in digital media delivery will enable new applications in the future in the entertainment sector (sports bars, digital cinemas, and eventually the home theater), distance education, or elsewhere.

The RMI effort

The focus of the RMI effort is to enable the most realistic recreation of an event possible while streaming the data over the Internet. Therefore, we push the technological boundaries beyond what current video-on-demand or streaming media systems can deliver. As a consequence, the system requires high-end rendering equipment and significant transmission bandwidth. However, we trust that advances in electronics, compression, and residential broadband technologies will make such a system financially feasible first in commercial settings and later at home in the not-too-distant future. Some of the indicators that support this assumption are, for example, that the DVD specification's next generation calls for network access of DVD players.[2] Furthermore, Forrester Research forecasts that almost 15 percent of films will be viewed by on-demand services rather than by DVD or video by 2005.[3] The infrastructure necessary for these services is gradually being built. For instance, in Utah, 17 cities are planning to construct an ultra-high-speed network for businesses and residents.[4]

The RMI project integrates several technologies that are the result of research efforts at IMSC. The current operational version is based on four major components that are responsible for acquiring, storing, transmitting, and rendering high-quality media.

- *Acquiring high-quality media streams*. This authoring component is an important part of the overall chain to ensure users experience high-quality rendering. As the saying "garbage in, garbage out" implies, no amount of quality control in later stages of the delivery chain can make up for poorly acquired media. In the current RMI version, authoring is an offline process and involves its own set of technologies. Due to space constraints, this article doesn't focus on this component.

- *Real-time digital storage and playback of multiple independent streams*. The Yima[5] Scalable Streaming Media Architecture provides real-time storage, retrieval, and transmission capabilities. The Yima server is based on a scalable cluster design. Each cluster node is an off-the-

Published by the IEEE Computer Society

shelf PC with attached storage devices and, for example, a fast or gigabit Ethernet connection. The Yima server software manages the storage and network resources to provide real-time service to multiple clients requesting media streams. Media types include, but aren't limited to, MPEG-2 at NTSC and HDTV resolutions, multichannel audio (such as 10.2 channel immersive audio), and MPEG-4.

- *Protocols for synchronized, efficient real-time transmission of multiple media streams*. A selective data retransmission scheme improves playback quality while maintaining real-time properties. A flow-control component reduces network traffic variability and enables streams of various characteristics to be synchronized at the rendering location. Industry standard networking protocols such as the Real-Time Transport Protocol (RTP) and Real-Time Streaming Protocol (RTSP) provide compatibility with commercial systems.

- *Rendering immersive audio and high-resolution video*. Immersive audio is a technique developed at IMSC for capturing the audio environment at a remote site and accurately reproducing the complete audio sensation and ambience at the client location with full fidelity, dynamic range, and directionality for a group of listeners (16 channels of uncompressed linear PCM at a data rate of up to 17.6 Mbits per second [Mbps]). The RMI video is rendered in HDTV resolutions (1080i or 720p format) and transmitted at a rate of up to 45 Mbps.

Overview of components

RMI's group sizes can range from one person or family at home to a large venue seating hundreds. For the visual streams, we decided that we required at least high-definition resolution as defined by the Advanced Television Systems Committee (ATSC).[1] The highest quality ATSC modes are either 1920 × 1080 pixels at an interlaced frame rate of 29.97 per second, or 1280 × 720 pixels at a progressive frame rate of 59.94 per second. (We discuss video playback details later.) For the audio rendering, we rely on the immersive audio technology developed at IMSC, which uses a 10.2-channel playback system. Immersive audio's rendering capabilities go beyond current stereo and 5.1-channel systems, which we also discuss in more detail later.

Each presentation session retrieves and plays back at least one high-definition visual and one immersive aural stream in synchronization. The available media content imposed this choice on the RMI system, so it isn't an inherent limitation in the Yima design. We store the streams separately on the server for two reasons. First, we designed the RMI system to be extensible so that additional video or other streams can become part of a presentation in the future. Second, letting streams be separately stored lets RMI retrieve different components of a presentation from different server locations. The final, fine-grained synchronization is achieved at the client side.

We designed the RMI system to be extensible so that additional video or other streams can become part of a presentation in the future.

Delivering high-resolution media

An important component of delivering isochronous multimedia over IP networks to end users and applications is the careful design of a multimedia storage server. Such a server must efficiently store the data and schedule the data's retrieval and delivery precisely before it is transmitted over the network. RMI relies on our Yima streaming media architecture. IMSC researchers designed Yima to be a scalable media delivery platform that can support multiple, high-bandwidth streams. For space reasons, this article focuses on Yima's features and techniques that are most relevant for the RMI system. (See related research for additional information.[5])

Figure 1 (next page) shows the server cluster architecture, which can harness the resources of many nodes and many disk drives per node concurrently. In our current implementation, the server consists of a four-way cluster of rack-mountable PCs running Red Hat Linux. However, larger configurations are possible to increase the number of concurrent RMI sessions supported. Each cluster node is attached to a

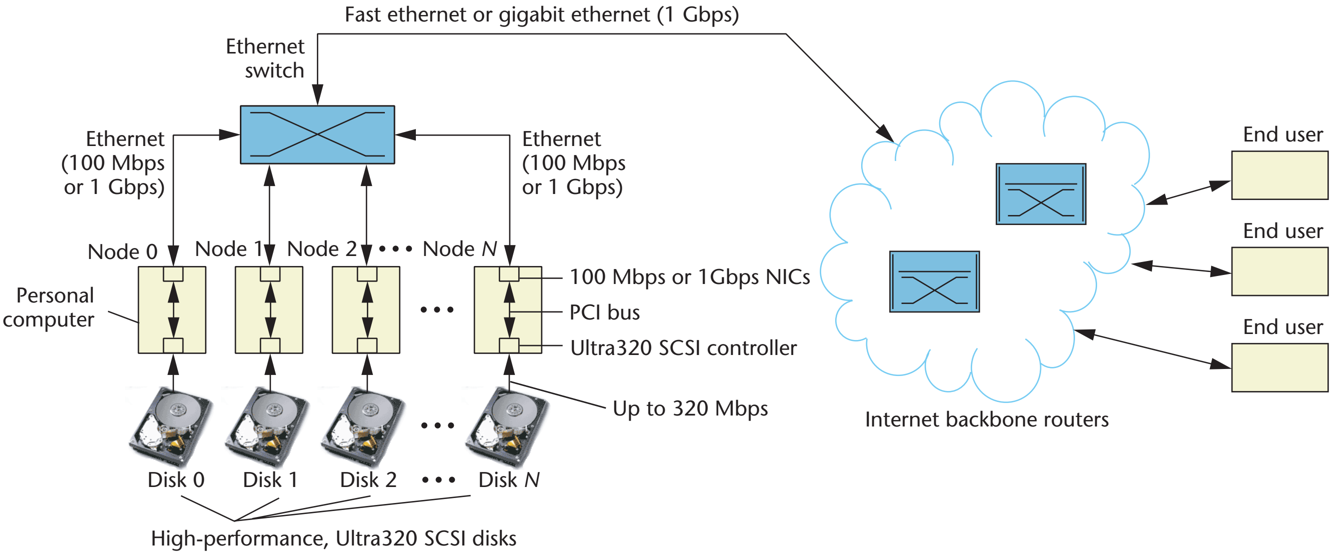

Figure 1. The Yima server cluster architecture.

local network switch with a fast or Gigabit Ethernet link. The nodes communicate with each other and send the media data via these network connections. We connected the local switch to both a wide area network (WAN) backbone (to serve distant clients) and a local area network (LAN) environment with local clients. Choosing an IP-based network keeps the per-port equipment cost low and is immediately compatible with the public Internet.

The RMI system stores the media data in segments (called blocks) across multiple, say four, high-performance hard disk drives. We can assign a media object's data blocks (in a load-balanced manner) to the magnetic disk drives that form the storage system in a round-robin sequence[6] or randomly.[7] Yima uses a pseudorandom data placement combined with a deadline-driven scheduling approach. This combination enables short startup latencies and can easily support multimedia applications with nonsequential data access patterns including variable bit rate (VBR) video or audio as well as interactive operations such as pause and resume. It also enables efficient data reorganization during system and storage scaling.[8]

Error control with selective retransmissions from multiple server nodes

One characteristic of continuous media streams is that they require data to be delivered from the server to a client location at a predetermined rate. This rate can vary over time for streams that have been compressed with a VBR media encoder. VBR streams enhance the rendering quality, but they generate bursty traffic on a packet-switched network such as the Internet. In turn, this can easily lead to packet loss due to congestion. Such data loss adversely affects compressed audio and video streams because the compression algorithm has already removed much of the data's temporal or spatial redundancy. Furthermore, important data such as audio–video synchronization information might get lost that will introduce artifacts in a stream for longer than a single frame. As a result, it's imperative that as little as possible of a stream's data is lost during the transmission between the server and a rendering location.

The Yima cluster architecture takes advantage of the distributed storage resources among the multiple nodes and the multiple network connections that link all the nodes together. Media data is transmitted via RTP encapsulated in connectionless User Datagram Protocol (UDP) packets. To avoid traffic bottlenecks, each node transmits the data blocks that it holds directly to the clients via RTP. Hence, each client will receive RTP data packets from each server node within the cluster.

The current Internet infrastructure wasn't designed for streaming media and provides only best-effort packet delivery. Therefore, RTP/UDP datagrams aren't guaranteed to arrive in order or at all. We can easily achieve reordering with the help of a global sequence number across all packets, but information loss requires special provisions if the rendered streams' quality at the receiving side should be acceptable.

One possible solution is using forward error correction (FEC). With this method, the server continuously adds redundant information to the stream that aids the receiver in reconstructing the original information if data is corrupted or lost during transmission. Because of its preemptive nature, FEC can add significant overhead that consumes additional bandwidth even when it isn't needed. With RMI, we're transmitting some streams that require in excess of 45 Mbps bandwidth.[9] In that case, retransmission-based error control (RBEC) is an attractive option. RBEC can be an effective solution for streaming media applications that use a play-out buffer at the client side.[10]

A central question arises when data is randomly stored across multiple server nodes and RBEC is used: When multiple servers deliver packets that are part of a single stream, and a packet doesn't arrive, how does the client know which server node attempted to send it? In other words, it isn't obvious where the client should send its request for retransmission of the packet. There are two general solutions to this problem. First, the client can broadcast the retransmission request to all server nodes, or second, it can compute the server node to which it issues the retransmission request.

Broadcast retransmissions. With the broadcast approach, all server nodes receive a packet retransmission request. The request broadcasting in this scenario can be well targeted by the client to include all the server nodes but no other computers. By observing the RTP/UDP packet header source IP address, the client can easily establish the complete set of server nodes. Once a server receives a request, it checks whether it holds the packet, and either ignores the request or retransmits. Consequently, this approach wastes network bandwidth and increases server load.

Unicast retransmissions. The second, more efficient and scalable method of sending retransmission requests requires that we identify the unique server node that holds the missing packet. To accomplish this, the client could reproduce the pseudorandom number sequence that was originally used to place the data across multiple server nodes. This approach has several drawbacks. First, identical algorithms on both the clients and servers must be used at all times. If we upgrade the server software, then we must upgrade all clients immediately, too. The logistics of such an undertaking can be daunting if the clients are distributed among thousands of end users. Second, during scaling operations the number of server nodes or disk drives changes, and hence new parameters need to be propagated to the clients immediately. Otherwise, the server nodes will be misidentified. Third, if for any reason the client computation is ahead, or behind the server computation (for example, the number of packets received doesn't match the number of packets sent), then any future computations will be wrong. This could potentially happen if the client has only a limited memory and packets arrive sufficiently out of sequence.

A more robust approach exists. The client can determine the server node from which a lost RTP packet was intended to be delivered by detecting gaps in node-specific packet-sequence numbers. We term these *local sequence numbers* (LSN) as opposed to the global sequence number (GSN) that orders all packets. Although this approach requires packets to contain a node-specific sequence number along with a GSN, the clients require little computation to identify and locate missing packets.

We implemented this technique and evaluated it with an extensive set of experiments across LAN and WAN environments. The results show that the method is feasible and effective. (See our related research for more details.[11])

Client-server adaptive flow control

RMI relies on the efficient transfer of multimedia streams between a server and a client. These media streams are captured and displayed at a predetermined rate. For example, video streams might require a rate of 24, 29.97, 30, 59.94, or 60 frames per second. Audio streams can require 44,100 or 48,000 samples per second. An important measure of quality for such multimedia communications is the precisely timed playback of the streams at the client location.

Achieving this precise playback is complicated by the popular use of VBR media stream compression. VBR encoding algorithms allocate more bits per time to complex parts of a stream and fewer bits to simple parts to keep the visual and aural quality at near constant levels. For example, a movie's action sequence might require more bits per second than its credits. As a result, different transmission rates might be necessary over the length of a media stream to avoid starvation or overflow of the client buffer. As a contradictory requirement, we want to minimize the variability of the data transmitted through a network. High variability produces uneven resource

Unlike previous approaches, we consider multiple signaling thresholds and adaptively predict the future bandwidth requirements.

utilization and might lead to congestion and exacerbate display disruptions.

The RMI flow-control mechanism's focus is on achieving high-quality media playback by reducing the variability of the transmitted data and hence avoiding display disruptions due to data starvation or overflow at the client. We identified the following desirable characteristics for the algorithm:

- *Online operation*. This is necessary for live streaming and desirable for stored streams.

- *Content independence*. An algorithm that isn't tied to any particular encoding technique will continue to work when new compression algorithms are introduced.

- *Minimizing feedback control signaling*. The overhead of online signaling should be negligible to compete with offline methods that don't need any signaling.

- *Rate smoothing*. The peak data rate and the number of rate changes should be lowered compared with the original, unsmoothed stream. This will greatly simplify the design of efficient real-time storage, retrieval, and transport mechanisms to achieve high-resource utilization.

We designed a high-performance rate control algorithm that adjusts the multimedia traffic based on an end-to-end rate control mechanism in conjunction with an intelligent buffer management scheme. Unlike previous approaches, we consider multiple signaling thresholds and adaptively predict the future bandwidth requirements. With this multithreshold flow-control (MTFC) scheme, we accommodate VBR streams without a priori knowledge of the stream bit rate. Furthermore, because the MTFC algorithm encompasses server, network, and clients, it adapts itself to changing network conditions. Display disruptions are minimized even with few client resources (such as a small buffer size). MTFC uses a modular rate change computation framework in which we can easily incorporate new consumption prediction algorithms and feedback-delay estimation algorithms. It is currently implemented in our Yima[5,12] streaming media server and clients. This lets us measure and verify its effectiveness in an end-to-end application such as RMI.

The client play-out buffer is a crucial component of any feedback-control paradigm. The server is the data producer that places data into the buffer, and the media decoder is the consumer that retrieves data. If the production and consumption rates are exactly the same, then the amount of data in the buffer doesn't change. If there is a mismatch, however, data will either accumulate or drain. If the buffer overflows or underflows then display disruptions will appear. Hence, the goal of managing the buffer is to keep the data level approximately at half the buffer size so that fluctuations in either direction can be absorbed. In an online feedback scheme, when the data level sufficiently deviates from the buffer midpoint, a correction message is sent to the server to adjust the sending rate.

We can configure the MTFC buffer manager with a variable number of data level watermarks or thresholds. For example, we can mark the buffer space with five, nine, or 17 thresholds (creating six, 10, or 18 logical data partitions). If the buffer data level crosses one of the thresholds, then a speed-up or slow-down message is sent to the server. Note that the thresholds trigger a message only if the data increase or drain happens in the correct direction—thresholds above the midlevel send a slow-down message for data increases and thresholds below midlevel trigger a speed-up message when data continues to drain. The desired rate at which the server should speed up or slow down the transmission is calculated based on the current buffer level, its rate of increase or decrease, and the predicted future transmission. This design provides soft and infrequent rate changes during normal operation, while it takes corrective action aggressively when a buffer overflow or underflow looms. MTFC also

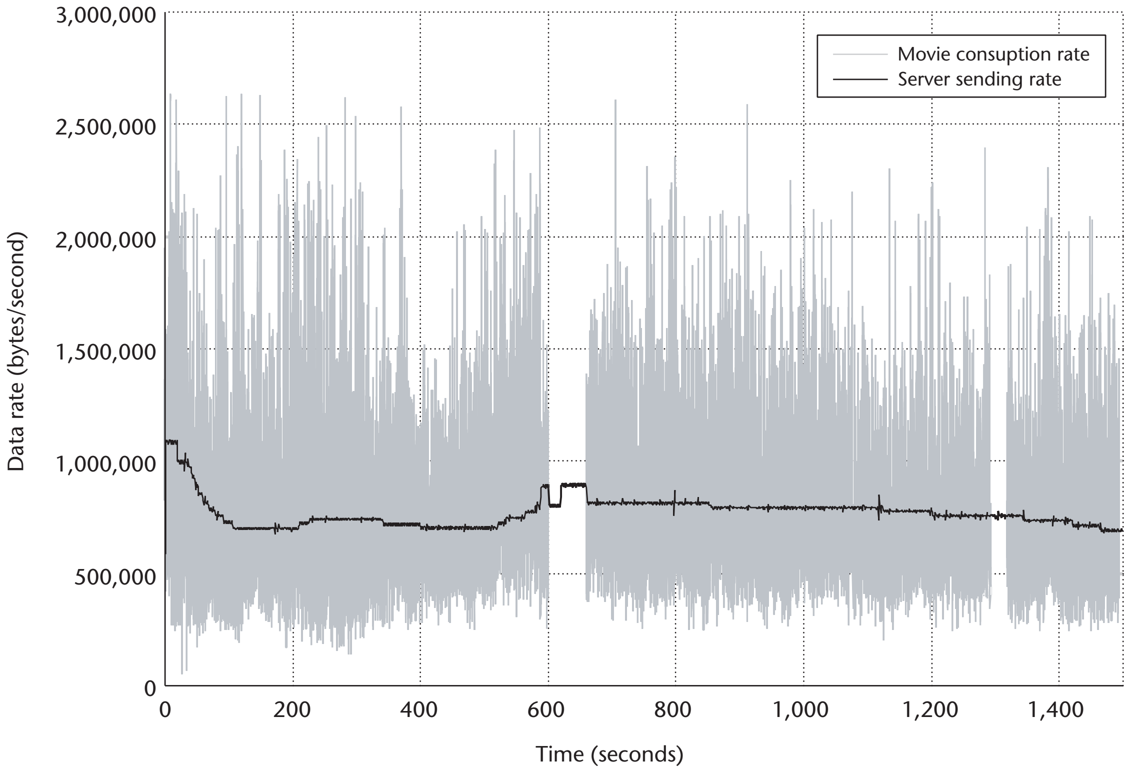

Figure 2. An example of an unsmoothed versus smoothed movie transmission profile.

has different threshold spacings at its disposal to aid in its operation. We can space thresholds at an equal distance, geometrically, or logarithmically to provide different control characteristics.

From our real-world experiments, we conclude that more than three buffer thresholds reduces the variability of the data transmission and the feedback overhead. At the same time, a consumption rate prediction algorithm smoothes streams with no prior knowledge of their transmission schedule. Therefore, our technique is well suited for highly dynamic environments that need to adapt to changing network and load conditions. Furthermore, the achieved smoothing allows for improved resource use by reducing peak data rates. The experimental results further show that our scheme outperforms existing algorithms in terms of traffic smoothness with similar or less signaling frequency. Figure 2 shows an example of an original VBR movie transmission profile and the resulting schedule when using MTFC.

Rendering

The rendering side of the RMI system consists of several parts. The video and audio streams are received over a sufficiently fast IP network connection on a PC running two instances of the Yima playback software. We structured this media player into several components and only one of them interfaces with the actual media decoder. This lets us plug in multiple software and hardware decoders and hence support various media types. For RMI, one of the players interfaces with a CineCast HD MPEG-2 decompression board manufactured by Vela Research. This decoder accepts MPEG-2 compressed high-definition video at data rates in excess of 50 Mbps and in both 1080-interlaced and 720-progressive formats. An HD serial digital interface (SDI) connection transports the video to a high-resolution front projector. Depending on the venue's size, we can use different projector models. MPEG-2 video at a data rate of 40 to 50 Mbps is referred to as having contribution quality. It's often the format of choice between production facilities and provides high visual quality. Consequently, extending the visual field requires that we improve the aural presentation as well.

Immersive audio

Audio can play a key role in creating a fully immersive experience. Achieving this requires that we exceed the spatial limitations of traditional two-channel stereo. Researchers have pro-

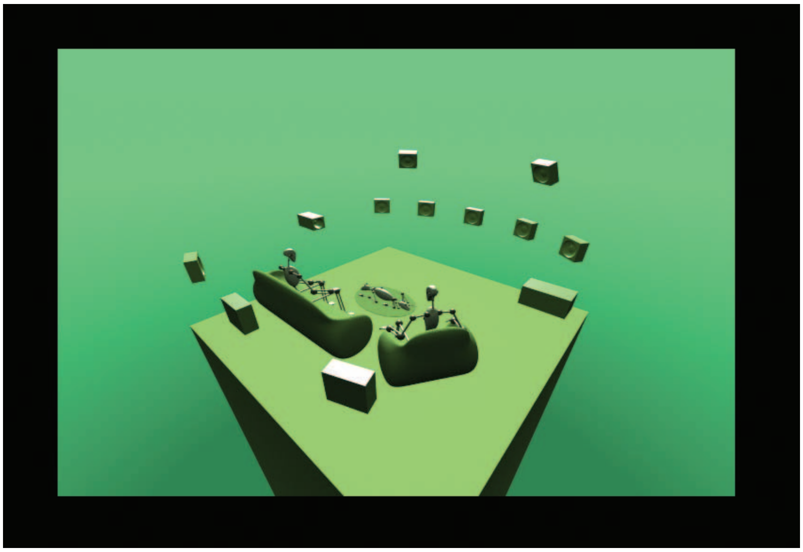

Figure 3. An immersive 10.2-channel audio setup illustrating the 12 speaker locations.

posed several rendering methods that use digital filters to represent the spectral characteristics of sound sources from different directions in 3D space.[13] These methods rely on accurate representation of head-related transfer functions (HRTFs) that represent the modifications imparted on sound by the head and pinnae. To deliver sound over loudspeakers, these methods also require precise cancellation of the crosstalk signals resulting from the opposite side loudspeaker to deliver the desired sound to each ear. As a result, they work well only for a single listener in a precisely defined position.

Multichannel surround-sound systems exist that use three front channels and two surround channels to provide a sense of envelopment for multiple listeners. Although these systems work well in movies, they aren't suitable for immersive systems. They leave significant spatial gaps in the azimuth plane (for example, at 90 degrees to the side of the listeners) and provide no coverage in the median plane (no elevation cues).

To minimize localization errors, the number of loudspeakers must increase linearly with the listening area's width.[14] A listener that moves just a few centimeters from the designated listening spot is subjected to high imaging distortion and no longer experiences the correct localization cues. Increasing the number of loudspeakers addresses this problem.

We designed and implemented a multichannel rendering system that addresses some of these limitations. The psychoacoustics literature explains that human listeners can localize sounds precisely in the front hemisphere and less precisely to the sides and in the rear hemisphere.[15] Therefore, localization errors from the front channels that arise when the listener's position is offset from the desired center location will be particularly evident.

In our implementation, we allocate five channels to the front horizontal plane by augmenting the traditional three front loudspeakers with two additional wide channels. This reduces localization errors in the front hemisphere and provides the wide channels with simulated side-wall reflection cues that increase the sense of spatial envelopment.[16] In addition to the five front channels, we added a rear surround channel to fill the gap directly behind the listener. We also reproduce elevation information by placing two height channels above the front left and right loudspeakers. Early experiments we performed with this configuration show that these 10 channels significantly increase listeners' sense of localization and envelopment.

The RMI implementation uses a second software player to interface with a 16/24-channel sound card model RME 9652 Hammerfall. The audio data is received via 16 channels in either 16- or 24-bit uncompressed linear PCM format at a data rate of up to 17 Mbps. The audio is transported via Alesis Digital Audio Tape (ADAT) lightpipes to a Digidesign ProTools system and rendered via individual equalizers and powered speakers. Figure 3 shows the speaker locations of a typical 10.2-channel setup.

A major challenge in multilistener environments arises from room acoustical modes, particularly in small rooms, that cause a significant variation in the responses measured at different listener locations. Responses measured only a few centimeters apart can vary by up to 15 to 20 decibels (dB) at certain frequencies. This makes it difficult to equalize an audio system for multiple simultaneous listeners. Furthermore, it makes it impossible to render a remote performance for a large audience and have them experience the sound exactly as it is being produced at the remote event. Previous methods for room equalization have used multiple microphones and spatial averaging with equal weighting. This approach tends to smooth out large variations due to standing waves, but it doesn't account for the effects of room modes that for some frequencies can be more concentrated in one region than another.

We developed a new method that derives a representative room response from several room responses that share similar characteristics.[17,18] We can use this response to equalize the entire class of responses it represents. Our approach is based on a fuzzy unsupervised technique (c-means clustering) for finding similarities, clustering room responses, and determining their representatives. Our results show a significant improvement in equalization performance over single point equalization methods, and we're currently implementing a real-time version that can perform the necessary corrections on the incoming audio channels.

Applications and demonstrations

We tested and showcased the RMI system in several configurations and locations. We deployed Yima servers in our laboratory on the USC campus and remotely at the Information Sciences Institute/East, Arlington, Virginia. The connectivity was provided either through Internet2 or via a SuperNet link.

We demonstrated the client and rendering setup at the following venues:

- The demonstration room next to our laboratory on the USC campus seats 10 people for an intimate, living room type of experience. This room contains a permanent setup and is our primary test location. Our first public demonstration of the RMI system took place there in June 2002. We routinely update the equipment with the latest software and hardware versions and continually demonstrate the system to IMSC visitors.

- In October 2002, we presented a recorded performance by the New World Symphony conducted by Alasdair Neale in USC's Bing Theater. This venue seats approximately 500 people and was specially equipped with a 30 × 17-foot screen and a 10.2-channel audio system for the RMI performance, which was part of an Internet2 Consortium meeting hosted by USC.

- In March 2003, we demonstrated the RMI system as part of the Internet2 Performance Production Workshop and the Internet2 Music Education Symposium at the New World Symphony's Lincoln Theater in Miami Beach, Florida. Again, the venue was specifically equipped with a large screen and a 10.2-channel audio system.

In the more distant future, we can envision a portable device with a high-resolution screen and a personal immersive audio system.

- In September 2003 the Remote Media Immersion system was demonstrated at a library dedication at Inha University, South Korea. The library was dedicated by Y.H. Cho, the Chairman of Korean Airlines, in memory of his father. This milestone marked the first international demonstration of RMI.

We've shown that RMI is adaptable to different situations. It also has proven reliable enough to be moved or recreated in different locations and for large audiences. Work to simplify the rendering setup (speaker calibrations and so forth) is currently under way at IMSC.

Discussion and conclusions

The current RMI setup is out of reach for most home users. For widespread adoption, numerous technological advances will be necessary, which subsequently will lead to more affordable prices and make the RMI system feasible for high-end home use. For example, we're currently using the MPEG-2 algorithm at a low-compression ratio to achieve our target visual quality. An improvement in compression algorithms and affordable hardware availability will most certainly make the same quality available at lower bit rates in the future—MPEG-4 is a candidate here. Hence, we envision a cross over point in the next few years, when the bandwidth required for RMI is below the bit rates that new high-speed, residential broadband technologies can deliver—for example, very high-speed DSL (VDSL).

Additionally, we're working toward simplifying and automating the setup and calibration procedures currently necessary. Subsequently, we can incorporate these signal-processing algorithms into multichannel home receivers at moderate cost.

In the more distant future, we can envision a portable device with a high-resolution screen and a personal immersive audio system. For a single listener, it's possible to achieve enveloping audio rendered with only two speakers, as long as we know the position and shape of the listener's ears. The PDA of the future will at some point have enough processing capabilities to combine visual tracking with adaptive rendering of both audio and video. **MM**

References

1. D. McLeod et al., "Integrated Media Systems," *IEEE Signal Processing*, vol. 16, Jan. 1999, pp. 33-76.
2. T. Smith, "Next DVD Spec. to Offer Net Access, Not More Capacity," *The Register*, 27 Oct. 2003.
3. T. Richardson, "CDs and DVDs Are 'Doomed'," *The Register*, 2 Sept. 2003, pp. C1, C2.
4. M. Richtel, "In Utah, Public Works Project in Digital," *The New York Times*, 17 Nov. 2003, Section C, pp. 1; http://www.nytimes.com/2003/11/17/technology/17utopia.html?8hpib.
5. C. Shahabi et al., "Yima: A Second Generation Continuous Media Server," *Computer*, vol. 35, no. 6, June 2002, pp. 56-64.
6. S. Berson et al., "Staggered Striping in Multimedia Information Systems," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, 1994, ACM Press, pp. 79-90.
7. J.R. Santos and R.R. Muntz, "Performance Analysis of the RIO Multimedia Storage System with Heterogeneous Disk Configurations," *ACM Multimedia Conf.*, 1998, ACM Press, pp. 303-308.
8. J.R. Santos, R.R. Muntz, and B. Ribeiro- Neto, "Comparing Random Data Allocation and Data Striping in Multimedia Servers," *Proc. SIGMETRICS Conf.*, 2000, ACM Press, pp. 44-55.
9. E.A. Taub, "On Internet of the Future, Surfers May Almost Feel the Spray," *The New York Times*, Circuits section, 9 May 2002, p. C4; http://dmrl.usc.edu/pubs/NYTimes- RMI.pdf.
10. C. Papadopoulos and G.M. Parulkar, "Retransmission-Based Error Control for Continuous Media Applications," *Proc. 6th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video* (NOSSDAV 96), LNCS 1080, Springer-Verlag, 1996, pp. 5-12.
11. R. Zimmermann et al., "Retransmission-Based Error Control in a Many-to-Many Client-Server Environment," *Proc. SPIE Conf. Multimedia Computing and Networking* (MMCN), SPIE Press, 2003, pp. 34-44.
12. R. Zimmermann et al., "Yima: Design and Evaluation of a Streaming Media System for Residential Broadband Services," *Proc. VLDB 2001 Workshop Databases in Telecommunications* (DBTel 2001), LNCS 2209, Springer-Verlag, 2001, pp. 116-125.
13. C. Kyriakakis, "Fundamental and Technological Limitations of Immersive Audio Systems," *IEEE Proc.*, vol. 86, 1998, pp. 941-951.
14. R. Rebscher and G. Theile, "Enlarging the Listening Area by Increasing the Number of Loudspeakers," 88th AES (Audio Engineering Society) Convention, 1990, Preprint No. 2932.
15. J.C. Makous and J.C. Middlebrooks, "Two Dimensional Sound Localization by Human Listeners," *J. Acoustical Soc. of America*, vol. 87, no. 5, May 1990, pp. 2188-2200.
16. Y. Ando, *Concert Hall Acoustics*, Springer-Verlag, 1985.
17. S. Bharitkar and C. Kyriakakis, "Perceptual Multiple Location Equalization with Clustering," *Proc. 36th IEEE Asilomar Conf. Signals, Systems, and Computers*, IEEE Press, 2002, pp. 179-183.
18. S. Bharitkar, P. Hilmes, and C. Kyriakakis, "The Influence of Reverberation on Multichannel Equalization: An Experimental Comparison between Methods," *Proc. 37th Conf. Signals, Systems, and Computers*, IEEE Press, 2003, pp. 155-159.

Christos Papadopoulos is an assistant professor at the University of Southern California (USC). His research interests include network security, multimedia communications, and operating systems. Papadopoulos has a PhD in computer science from Washington University in St. Louis, Missouri. He is affiliated with the Integrated Media Systems Center (IMSC) and Information Sciences Institute (ISI).

Chris Kyriakakis is the director of the IMSC Immersive Audio Laboratory at USC. His research interests include multichannel audio signal processing, automatic room equalization, microphone arrays for sound acquisition, stereoscopic high-definition video capture and rendering, virtual microphones for multichannel audio synthesis, and streaming of multichannel audio over high-bandwidth networks. Kyriakakis has a PhD in electrical engineering from the University of Southern California.

Ulrich Neumann is an associate professor of computer science at USC. He is also the director of the Integrated Media Systems Center (IMSC), an NSF Engineering Research Center (ERC). Neumann holds an MSEE from SUNY at Buffalo and a PhD in computer science from the University of North Carolina at Chapel Hill. His research relates to immersive environments and virtual humans.

Alexander A. Sawchuk is a professor in the Department of Electrical Engineering and deputy director of the Integrated Media Systems Center at USC. His research includes image processing, acquisition and display, and optoelectronic data storage and network systems. Sawchuk has a BS from the Massachusetts Institute of Technology, and an MS and PhD from Stanford University, all in electrical engineering.

Cyrus Shahabi is an assistant professor, the director of the Information Laboratory (InfoLAB) at the Computer Science Department, and a research area director at the Integrated Media Systems Center (IMSC) at USC. His current research interests include streaming architectures and multidimensional databases. Shahabi has a PhD in computer science from the University of Southern California.

Roger Zimmermann is a research assistant professor of computer science, a research area director with the Integrated Media Systems Center (IMSC), and the director of the Data Management Research Laboratory (DMRL). His research interests are in streaming media architectures and distributed database integration. Zimmermann has a PhD in computer science from the University of Southern California.

Readers may contact Zimmermann at the Univ. of Southern California, 3740 McClintock Ave., Suite 131, Los Angeles, CA 90089; rzimmerm@usc.edu.