Joint Source-channel Multistream Coding and Optical Network Adapter Design for Video Over IP

Jie Chen, *Senior Member, IEEE,* and K. J. Ray Liu, *Senior Member, IEEE*

***Abstract*—In this paper, we present a panorama picture on how to achieve end-to-end video over IP service under the communication environments, consisting of backbone networks, hybrid access networks, and the end users. The paper consists of three equally weighted subtopics which cover some novel thoughts in designing and implementing the video over IP system in the different areas, namely, a Synchronous Optical NETwork (SONET) network adapter for backbone connections, the joint source-channel multistream coding in hybrid access networks, and the content-based source coding in the transform domain. We propose to link the three different problems associated with the hybrid networks, which have different characteristics and design requirements, to improve the critical performances in various areas of video over IP systems. The goal is to deliver video over IP networks in more cost-effective and reliable manner.**

IP/ATM over SONET is currently a commonly used backbone technique. In the first part of this paper, we present a flexible design and implementation of a SONET network adapter to carry IP traffic via optical fiber. Unlike many conventional designs, our single-chip implementation supports different data rates (OC-3, OC-12, and OC-48), carries IP traffic directly over fiber, achieves more flexible for multivendor interoperability, and provides bandwidth efficient designs at the lower system latency. Hybrid access-networks via wireline or wireless connections are most likely needed for last-mile services. In the second part of this paper, we propose a joint source-channel *multistream* video coding scheme to combat the transmission errors under the harsh network conditions. On top of traditional error control techniques, the simulation results demonstrate that our multistream design outperform the conventional approaches by up to 5–7 dB under the harsh network conditions. To support our multistream video coding scheme, we need to access and manipulate video objects rather than the frame of pixels. In the third part of this paper, we focus on the coding of arbitrary shape video fully in the transform domain.

***Index Terms*—Content-based video coding, error-resilient transmission, joint source-channel coding, optical network, SONET, streaming video, video over IP.**

I. Introduction

With the advances in digital compression technology and the steady deployment of broad-band networks such as fiber optics, cable, xDSL, and the third-generation wireless CDMA systems, multimedia services such as packets (data/voice/video over IP) through broad-band networks have been emerging as technologies for the new millennium. Those state-of-the-art technologies are changing our daily life and enabling not only e-commerce but also traditional businesses to enhance productivity, reduce costs, and increase business agility. However, before we can realize the full potential of these multimedia services, we have to address the challenge of how to deliver multimedia services over networks cost effectively, ubiquitously, and with sufficient quality. Due to the large variety of existing network technologies, it is most likely that hybrid networks are used to support multimedia services, as shown in Fig. 1. However, different networks have different characteristics. To optimize the performance of multimedia systems, we should consider some improvements in designing and implementing multimedia over IP system such as *jointly* considering video compression and delivery schemes based on the network alternatives, capacities, and characteristics. This paper consists of three subtopics which cover some novel thoughts in designing and implementing multimedia over IP system in the different areas; namely, a Synchronous Optical NETwork (SONET) network adapter for backbone connections, the joint source-channel multistream video coding in hybrid access networks, and the content-based source coding in the transform domain. Although these three pieces of designs can be totally separated, here we try to put them together as a system level paper. We do not intend to provide the solutions to solve the end-to-end problems in this paper. Instead, we propose the different ideas to improve the critical performances in the different areas of video over IP systems. The goal is to deliver video over IP networks in more cost-effective and reliable manner.

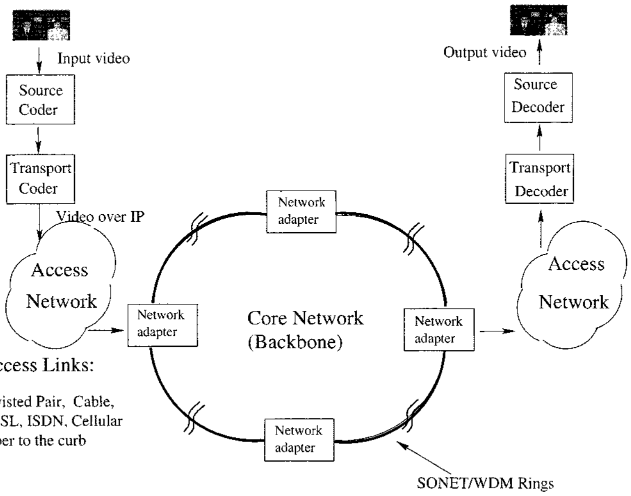


Fig. 1. Hybrid networks are used to support multimedia services.

Manuscript received September 4, 2001. The associate editor coordinating the review of this paper and approving it for publication was Dr. John A. Sorensen.

J. Chen was with the Electrical and Computer Engineering Department, Institute for Systems Research, University of Maryland, College Park, MD 20742 USA. He is now with Flarion Technologies, Bedminster, NJ 07921 USA.

K. J. R. Liu is with the Electrical and Computer Engineering Department, Institute for Systems Research, University of Maryland, College Park, MD 20742 USA.

Publisher Item Identifier S 1520-9210(02)01462-1.

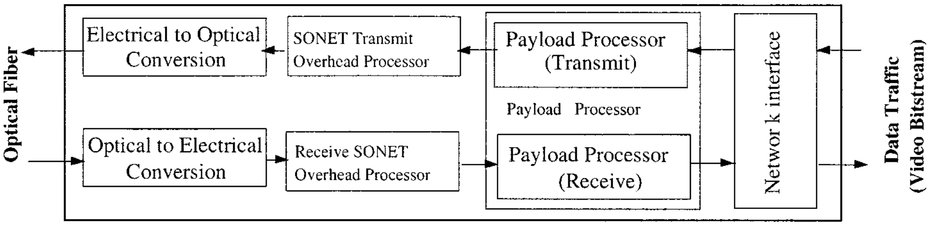

Optical Fiber
Electrical to Optical Conversion
SONET Transmit Overhead Processor
Payload Processor (Transmit)
Payload Processor
Optical to Electrical Conversion
Receive SONET Overhead Processor
Payload Processor (Receive)
Networ k interface
Data Traffic (Video Bitstream)

Fig. 2. Achitecture of a SONET network adapter served as the Layers 1 and 2 IP router to carry packets over SONET or directly over fiber.

TABLE I
DIFFERENT TECHNIQUES FOR ROBUST TRANSMISSION

Error control at encoder [4], [5], [6]	Encoder plays the primary role. Those methods utilize the source-coding and/or transport-control either to minimize the effect of transmission error without requiring any error concealment at decoder or to make the error-concealment task at the decoder more effective. Examples include FEC, joint source and channel coding, and layered coding.
Error control at decoder [7], [8], [9]	Decoder fulfills the task of error resilience. Those methods attempt to recover the lost information by estimation and interpolation without relying on additional information from the encoder. Examples include spatial and temporal smoothing, interpolation, and filtering
Error control by interaction [10], [11], [12]	Both encoder and decoder work cooperatively to minimize the impact of transmission errors. Examples include ARQ and selective predictive coding based on feedback from the decoder.

A. Improvements in Designing a SONET Network Adapter for Backbone Connections

Compared to speech communications, video communications always have enormous bandwidth requirements. When the traffic volume is low between IP routers, bandwidth partitions over a common interface made it attractive to carry IP packets over ISDN, X.25, or frame relay connections. As the traffic grows, it is becoming more desirable to carry packets over the SONET [1] or directly over optical fiber because of its reliability and broad bandwidth, at least in the backbone connections or core networks which connect different internet service providers (ISPs), as shown in Fig. 1.

In the first part of our paper, we propose a flexible design and implementation of a SONET network adapter served as the Layers 1 and 2 (the physical and data link layers) IP router, as shown in Fig. 2, which supports packets over SONET or directly over fiber. Because we emphasize on the design and implementation of video over IP system in this paper, we hereby refer *packet* to the IP packet of video or simply as video/IP. Unlike the conventional packets over SONET [2] or over ATM then over SONET [3] design, our single-chip design supports packets directly over fiber or over SONET at different transmission rates (OC-3, OC-12, and OC-48). In addition, our high-speed (2.4 Gb/s) SONET network adapter design provides more flexibilities for network security and multivendor interoperability. Moreover, we can reduce bandwidth requirements by applying the point-to-point protocol (PPP) header compression and the "strip PPP and CRC fields off" features.

B. Joint Source-Channel Multistream Coding in Hybrid Access Networks

Although the optical network is ideal for video transmission because of its reliability and broad bandwidth, it is now still too expensive for average users to have the fiber connected to the homes. As a result, access networks or last-mile services of either wireline or wireless connections are usually needed before we reach the core networks. Advanced digital communication technologies such as cable, xDSL, and the third-generation wireless CDMA are used to connect end users to core networks. One inherent problem of any access network connections is that information may be altered or lost during transmission due to channel noise/interference. The effect of such information loss can be devastating for the compressed video because any damage to the compressed bit stream may lead to objectionable visual distortion at the decoder; it also causes the commonly used predictive coding to propagate errors in the reconstructed video to future frames in the video sequence. Furthermore, in the commonly used variable length coding (VLC), the boundary between video code words is implicit. Transmission errors typically lead to an incorrect number of bits being used in VLC decoding, which causes loss of synchronization with the encoder. Many error control techniques therefore have been proposed for video communication, as summarized in Table I. On one hand, the traditional error control and recovery schemes in data communications have been extended for video transmission. On the other hand, the signal-reconstruction and error-concealment techniques have been proposed to strive to obtain a close approximation of the original signal or attempt to make the output signal at the decoder least objectionable to human eyes.

For data-oriented services, information reliability is more important than system delay/latency. Therefore, the missing packet retransmission is requested by the TCP layer at the end users at the cost of extra delay/latency. However, for real-time multimedia over IP services, the extra delay/latency is intolerable, and real-time protocol (RTP) user datagram protocol (UDP) instead of TCP is used. The real-time and interactivity requirements may exclude the deployment of some well-known error-recovery techniques such as the Automatic Repeat-reQuest (ARQ) retransmission. In addition, issues such as audio-visual synchronization and multipoint communications further complicate the problem of error recovery. As a result, the self-recovery transmission mechanism (or joint source-channel transmission mechanism) is more desirable. Our *multistream* video over IP in the second part of this paper is actually designed for that purpose. As the video segmentation techniques become more mature [13], it enables us to view, access, and manipulate video objects rather than the frame of pixels with great error robustness at a large range of bit-rates. On top of the traditional error control techniques such as forward error correction (FEC), we propose improvements in robust transmission in the second part of our paper by taking advantage of both content-based video coding and joint source-channel coding.

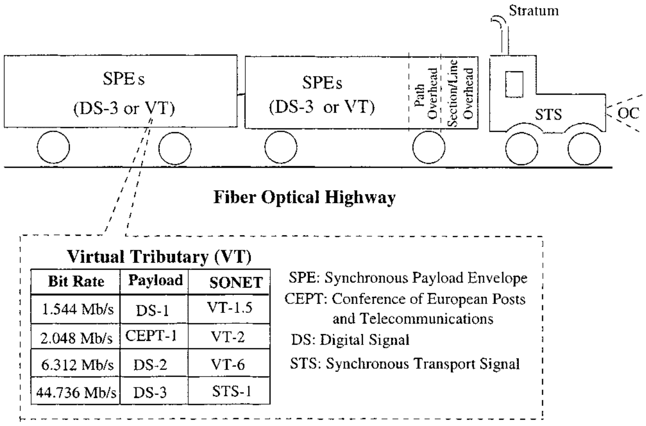
C. The Coding of Arbitrary Shape Video Fully in the Transform Domain

To support the content-based multistream video coding scheme proposed in the second part of this paper, we need to perform motion estimation and compensation for arbitrarily shaped video objects. Now the question becomes "What the content-based source coding scheme should we use?" Surely, we can adopt the MPEG-4 content-based video coding design to meet our requirements. However, in the third part of our paper, we propose an alternative coding of arbitrary shape video fully in transform domain instead. Unlike the conventional MPEG-4 modified block matching approaches [14], [15], the motion estimation and compensation in our coding scheme work fully in the discrete cosine transform (DCT) instead of the spatial domain. It is important to recognize that the DCT-based nature of our improved design enables the efficient combination of both DCT and motion estimation/compensation units, which consume most of computing power in a video coder [16]–[18], into one single unit. Therefore, our emphasis in this design is on optimizing the hardware complexity by minimizing the computational units along the data path. Simulation results demonstrate the comparable performances between our transform domain approach and the conventional MPEG-4 design [14], [15]. Notice that, if the original input image sequences are not decomposed into several video objects of arbitrary shapes, the proposed scheme simply degenerates into supporting the video coding of conventional image sequences with rectangular shapes. Therefore, this approach can be viewed as a logical extension of the DCT-based motion estimation/compensation schemes [19]–[22] toward arbitrarily shaped video sequences.

D. Paper Outline

In this paper, we present novel concepts in combining the design and implementation of video over IP systems from various aspects of core network design, data transmission in access network, and source coding. Our contributions include:

1) a flexible single-chip design and implementation of a SONET network adapter served as the Layers 1 and 2 IP router to deliver video over IP over SONET/SDH or directly over fiber, which provides reliable, high-speed, and multivendor compatible backbone connections between ISPs;
2) a joint source-channel multistream video coding scheme to combat the transmission errors in hybrid access networks under the harsh network conditions; and
3) a content-based video coding fully in DCT-domain to reduce the hardware complexity.



Bit Rate	Payload	SONET
1.544 Mb/s	DS-1	VT-1.5
2.048 Mb/s	CEPT-1	VT-2
6.312 Mb/s	DS-2	VT-6
44.736 Mb/s	DS-3	STS-1

Fig. 3. SONET can transport many different digital signals at many different data rates. Here, stratum is used as the reference clock for network synchronization.

The rest of paper is organized as follows. In Section II, we present our flexible design of the SONET network adapter to deliver video over IP over SONET or directly over fiber. Then we describe our joint source-channel multistream video coding scheme to combat the transmission errors in Section III. In Section IV, an efficient DCT-domain content-based video coding mechanism is proposed. Finally, we conclude our work in Section V.

II. A Flexible Design and Implementation of SONET Network Adapter

As we have pointed out in the introduction, video over optical network is a leading expedient solution to provide high-capacity backbone connections between ISPs. Although our design (as shown in Fig. 2) can support both IP and ATM traffic, here we only emphasize the IP design in this paper (please refer to [23] for ATM over SONET or directly over fiber design). Before we proceed with our discussion, we feel the need to answer some common questions. "Fiber optics provides abundant bandwidth, how does the video information be encapsulated in it?" SONET [or its counterpart, synchronous digital hierarchy (SDH)] [1] is a standard which provides different rates, formats, and optical parameter specifications for optical interfaces ranging from 51 Mb/s (OC-1) to 9.8 Gb/s (OC-192) capacities. The major attribute of SONET is its ability to transport many different (asynchronous or synchronous) digital signals using a standard synchronous transport signal (STS) format, as shown in Fig. 3. The video signals can vary from low bit-rate (DS0) in H.263, to medium bit-rate (DS-1, CEPT-1) in MPEG-1, up to high bit-rate (DS-2, DS-3)

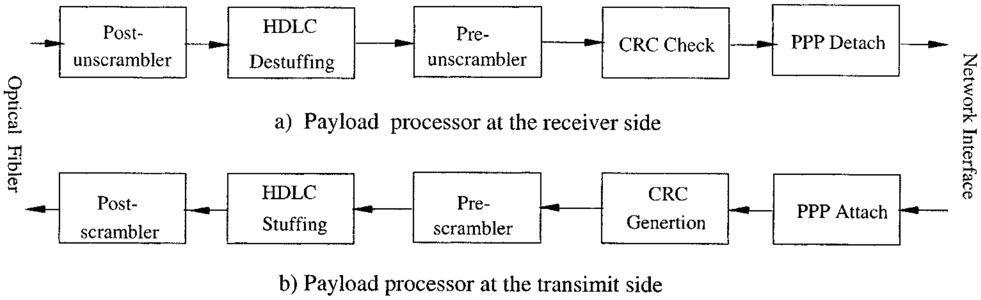


Fig. 4. Pipelined architecture of the payload processor: (a) payload processor at the receiver site and (b) payload processor at the transmit site.

in MPEG-2 and HDTV applications. The mapping of tributary signals (DS-1, CEPT-1, and DS-2) into an STS is accomplished through the use of virtual tributaries and payload pointers. The pointers allow the flexible alignment of payload within the transport signal by indicating where the asynchronous or synchronous payload begins [1]. DS0 signals at 64 kb/s are not addressed as a SONET format, but they are prepackaged in DS1s via switches, channel banks, etc. Another common question is: "What is novel in our design?" Unlike many conventional designs, we support the following:

1) different data rates (OC-3, OC-12, and OC-48) and different traffics (IP and ATM) on a *single* device;
2) IP traffic not only over SONET but also *directly* over fiber (there are many designs of IP over SONET then over fiber, but not IP directly over fiber);
3) more flexible design for multivendor interoperability and lower system latency (we provide many choices for ISPs to configure the device as we will discuss later);
4) bandwidth reduction by applying our novel PPP header compression and the "strip PPP header and CRC fields off" features;
5) two levels of scrambling to prevent malicious network attacks.

Packet over SONET, or more accurately Video/IP/PPP/HDLC/SONET, is described in [2]. Compared to the packet over SONET design, the packet directly over fiber design is more cost-effective and bandwidth efficient. Because the data engine or the core processor in our SONET network adapter design is the "payload processor," we derive the fully pipelined architecture in the payload processor, as shown in Fig. 4. The difference between the packet directly over fiber and over SONET design is whether or not to bypass the "SONET overhead processor" block, as shown in Fig. 2. In addition, the designs of the HDLC stuffing and destuffing blocks in "payload processor," as shown in Fig. 4 are different between these two modes. Here, we provide a provisionable *mode-control* register to control the operating modes of the device (0: packet over fiber, 1: packet over SONET). On the other hand, what is common between these two operating modes is that they share the same "payload processor" and "network interface" blocks [24]. In what follows, we will explain our SONET network adapter design and implementation in more detail. Furthermore, we will focus on the design of each block at the receive side of the payload processor, and the mirrored approach can be applied to the transmit side.

A. The Design of HDLC Stuffing and Destuffing Blocks

HDLC stuffing and destuffing blocks (shown in Fig. 4) work differently for the packet over SONET versus the packet directly over fiber mode. For the video/IP over SONET design, the "HDLC destuffing" block delineates a HDLC packet using the HDLC character (0x7E) at the transmit side, then the flags are removed at the receive side. The unescaped destuffing is performed (0x7D5D is unescaped to 0x7D and 0x7D5E is unescaped to 0x7E). However, for our novel packet directly over fiber design, the HDLC byte boundary is meaningless because the synchronization mechanism used in packet over SONET design is no longer valid. Therefore, the bit-synchronous HDLC has to be used. For the bit-synchronous HDLC, the "HDLC stuffing" block at transmit side inserts a zero bit after any run of five consecutive logic-one bits of transmitted packets. This distinguishes the real data from the HDLC end-of-packet mark, which is "01 111 110" (0x7E) in binary or a run of six consecutive logic-ones, which cannot by definition be a part of the encoded user data. The "HDLC destuffing" block at the receive side can easily decode this bit stream by counting consecutive logic-ones. If the counter reaches five and the next bit is a zero, then that zero is a stuffing bit and should be removed. If the bit is one, the end-of-packet is encountered. In addition, no escape characters, i.e., 0x7E is escaped to 0x7D5E, are needed in the bit-synchronous HDLC thus being more bandwidth efficient compared to the byte-synchronous HDLC. The optional CRC-32 or CRC-16 calculation on the whole received packet is performed after HDLC destuffing. The result after the CRC checking should be all zeros and a different value indicates an error.

B. The Design of PPP Detach

Creativity and flexibility have been built in our "PPP detach" design to improve bandwidth efficiency. The procedure of PPP detach is listed in Table II. Here, *PayloadType* [1:0] register is used to distinguish the different types of data traffics because our device is capable of handling both ATM and IP traffics. In the packet over SONET design, the *EOPMarker* indicates the end of packets (EOP). As we have mentioned in the introduction, any networks may experience information loss or distortion. It is better, from the performance point of view, not to discard the error packets in real-time multimedia services because the decoder may still be able to utilize those packets to recover the original signals or some useful information by applying the

TABLE II
PROCEDURE OF PPP DETACH. HERE, POS STANDS FOR PACKET OVER SONET OR DIRECTLY OVER FIBER

```
1. if (PayloadType $\neq$ POS) {
       do nothing and bypass the datagram;
   }
2. else if (Error Packet ending with AbortMarker and EOPMarker) {
       do nothing and bypass the datagram;
   }
3. else { /* valid POS */
   for (m=0; m $\leq$ 11; m++) {
       if (Protocol $\neq$ PPP-protocol[m]) {
           /* malformed packet */
           drop the packet;
           increase malformed packet counter by one;
       }
       else {
           strip or keep PPP header and CRC fields;
       }
   }
   }
```

error control schemes, as listed in Table I. Therefore, we introduce our novel *BadMarker* and *AbortMarker* to distinguish an error packet from a good packet ending with *EOPMarker*.

- *BadMarker:* This is used to mark the packet as a bad one once a checksum mismatch occurs, and the "CRC check" block at the receiver side can change EOPMarker to BadMarker.
- *AbortMarker:* This indicates where the packet gets aborted, i.e., due to a service disruption.

Notice that every packet can only end up with either EOPMarker, BadMarker, or AbortMarker, but not both. Next, we will discuss our novel design in the kernel (Step 3) of the PPP detach procedure listed in Table II.

1) Malformed Packets Handling: According to the Internet Engineering Task Force (IETF) 1619 Standard [2]:

> Any malformed packets must be silently dropped.

Here, the malformed packet stands for the packet that its PPP header fields mismatch the provisioned values. Provided that we only allow IP packet with the protocol number 0x0021 to pass, packets with different protocols other than 0x0021 are considered as malformed packets. As a result, we can silently drop the whole malformed packet and increase the malformed packet counter by one.

Many different protocols can be encapsulated in PPP such as RTP/IP/PPP or UDP/IP/PPP for real-time video/audio applications, and TCP/IP/PPP [25] for non-real-time data applications. Those protocols are numbered according to IETF 1700 [26] such as 0x0021 for IP, 0x8021 for the network negotiation and IP control protocol, and 0xC021 for the link control protocol (LCP). Because PPP can support multiprotocol encapsulation, we provide 12 programmable 16-bit registers, PPP-protocol $[m][n]$ $(m \leq 11, n \in 0\ldots15)$, to store the allowable PPP protocols which can be provisioned by different ISPs. (The two commonly used protocols: IP (protocol number: 0x0021) and IP control protocol (protocol number: 0x8021) are defined as default). The "PPP detach" block checks the PPP header to determine any malformed packet.

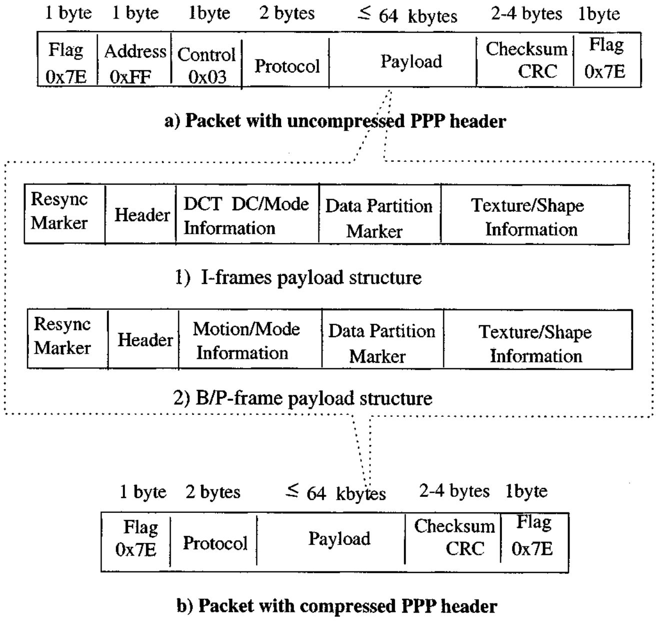


Fig. 5. Different PPP formats: (a) packet with uncompressed PPP header and (b) packet with compressed PPP header.

2) Optimize Bandwidth Usage: Due to the explosive growth of Internet traffics, bandwidth efficiency as a *de facto* is one of the most important design criteria of many network systems. There have been many approaches to achieve this design goal at many different levels ranging from application down to physical layer or from system down to chip design. Here, we provide our novel "PPP header compression" and "strip PPP header and CRC fields off" features to achieve bandwidth saving at the data link level based on the system setup and the characteristics of incoming traffics.

a) *PPP header compression*: Except for the commonly used uncompressed PPP format for data transmission, we propose our novel bandwidth efficient "PPP header compression" in our design. The uncompressed PPP formats is shown in Fig. 5(a). Here, the address field stands for the destination to accept the packet (default value: 0xFF is for all stations to accept the packet), and the control field stands for packet number (default value: 0x03 is for unnumbered packet). Since PPP supports multiprotocol encapsulation, the protocol field defines the types of data traffic in the payload such as IP, IPv6, AppleTalk, etc. For video over IP, the PPP payload field contains the IP paradigm for video service such as resynchronization marker, macroblock (MB) number, encoding mode, quantization step size, DCT and motion information, MB-based video texture information, etc. To accommodate the content-based video coding in the third part of our paper, we also include the shape information of the associated video objects in IP paradigm. The payload size of PPP is variable, up to some negotiated maximum size prior to a connection setup (default value: 1500 bytes). The checksum (CRC) is normally 2 bytes (CRC-16: $1+x^5+x^{12}+x^{16}$) but a 4-byte CRC (CRC-32: $1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}$) can be negotiated. Here, the CRC polynomial is generated on the whole packet including the PPP header but not HDLC Flag, 0x7E. Unlike the conventional PPP design, to reduce the bandwidth usage, we introduce "PPP header compression" or packets with *compressed* PPP header [as shown in Fig. 5(b)] in our design because the address and control fields of point-to-point network are normally set to be 0xFF03. Therefore, those fields can be omitted (two out of five PPP header bytes can be saved for each packet). We can check that the incoming packet is in the compressed or the uncompressed PPP header format based on the setting of control register, *PPP-header-check*.

b) *Strip PPP header and CRC fields off*: Because PPP supports multiprotocol encapsulation, we can classify the incoming data stream embedded in the PPP packets into two groups: *control signals* and *real datum*. Those two types of traffics can be distinguished based on the protocol numbers associated with the data packets.

— *Control signals*: Prior to real data traffic flowing through the networks, PPP has to go through three basic phases of negotiations: the LCP negotiation, the authentication and link quality management, and the network control protocol (NCP) negotiation [27]. Both transmitter and receiver need to negotiate and agree on certain network parameters such as maximum packet size, routing information (IP address), etc. Also, those control signals are embedded in PPP. Therefore, we are not allowed to strip those PPP header and CRC fields off to save bandwidth because they are critical for establishing the connections and not retrievable once they are discarded.

— *Real datum*: Once the connections established, the video bit-stream can be carried over PPP through the networks. The duration of network occupancy can last from minutes to hours depending upon the applications. Furthermore, the PPP packet sizes vary from medium to long (up to 64 kbytes) and are embedded in IP packets. Due to the reliability (BER below 10^{-10}) of optical fiber, both PPP header and CRC fields (4 up to 8 bytes) therefore can be stripped off from the IP packet to save bandwidth. Based on the recent studies of data traffics over Internet [28], nearly half of data streams have the packet size of 40 up to 44 bytes. With the average IP packet size of 40 bytes, we can save up to 20% of bandwidth.

C. The Scrambler Design for Network Security

As network security becomes increasingly more important, we include the $1 + x^{43}$ self-synchronous [29] scrambler/unscrambler in our SONET network adapter design. This self-synchronizing scrambler $(y(t) = x(t) \oplus y(t-43))$ works like an infinite impulse response (IIR) filter. The pseudorandom output of the 43th register $(y(t-43))$ is xored with the input $(x(t))$ and is transmitted $(y(t))$. The purpose of this scrambler is not only to ensure an adequate number of transitions (zeros to ones and ones to zeros) for SONET clock recovery but also to randomize the bits going out to the networks to prevent malicious attacks. The scrambler operates continuously throughout the bytes of the synchronous payload envelope (SPE) [1] but bypassing the SONET overhead bytes. On the top of including $1 + x^{43}$ self-synchronous in our design, there is another issue: "Where should we place the scrambler/unscrambler?" From the perspective of the SPE mapping, the placement of the scrambler before or after the HDLC stuffing block at the transmit side does not matter because the SPE mapping is transparent to the SONET network. The agreement in the IETF standard is to place the scrambler after the HDLC stuffing block for pragmatic reasons [2]. However, using this approach is not with its problems, because malicious users can transmit packets filled with either the HDLC flag pattern (0x7E) or the escape sequences, i.e., 0x7D5E so that the link bandwidth is halved. This could render sophisticated quality of service (QoS) control mechanisms useless [30]. Unlike the conventional design, we provide the great flexibility for ISPs to choose the location of the scrambler, which is indicated by "Payload-control register" at transmit side (00: No scrambler, 01: Post-scrambler, 10: Pre-scrambler, 11: Undefined). Our default setting Payload-control [1:0]="00" is for interworking with the old equipment design based on RFC 1619 [2].

Based on the previous discussion, in our improved SONET network design, there are two levels of scrambling: the SONET scrambling (x^{43}) and the packet scrambling (prescrambling and postscrambling). Note that only the prescrambling can avoid (not definitely) malicious network attacks by sending in alot of 7E/7Ds. As the counterpart of the scrambler, the unscrambler functioning as a FIR filter is placed in the mirrored position corresponding to the location of the scrambler.

D. The Performance of SONET Device

In this part of the paper, we propose many new ideas in designing and implementing the SONET network adapter in a more flexible, reliable, and bandwidth efficient way. The clock rate of our single-chip design, as shown in Fig. 2, is 78 MHz $((38\,880)/(4 * 125\ \mu\text{s}) = 77.76$ MHz) for single OC-48 link and four OC-12 links, or 20 MHz $((2430 * 4)/(4 * 125\ \mu\text{s}) = 19.44\ \text{MHz})$ for four OC-3 lines. The calculation is based on the following:

> each OC-48 frame, for instance, consists of 38 880 bytes (1440 bytes overhead + 37 440 bytes payload) which are processed in parallel via four different logical data channels in our design (the time span to process those bytes is 125 μs).

In terms of data throughput rate, our design can support up to 2.4 Gb/s.

The latency of each block in the payload processor, shown in Fig. 4, is listed in Table III. These result in negligible overall system latency, i.e., 324 ns in the transmit direction and 252 ns in the receive direction for each OC-48 connection.

In our loopback bit error rate (BER) test (Data stream → SONET network adapter at transmit side → fiber → SONET network adapter at receive side → Data stream), we observe that our design has negligible bit errors $(\text{BER} < 10^{-12})$. In addition, we use the same loopback setup to perform the network security test by maliciously sending alot of 7E/7Ds. With the two levels of scrambling, we can recover all sending bits.

TABLE III
LATENCY OF EACH BLOCK IN OUR DESIGN

Blocks at transmit side	Latency (Clock cycle)	Blocks at receive side	Latency (Clock cycle)
PPP attach	2	PPP detach	3
CRC generation	7	CRC check	4
HDLC stuffing	7	HDLC destuffing	7
Pre-scrambler	11	Pre-unscrambler	7
Post-scrambler	7	Post-unscrambler	6

Therefore, the optical network is the leading expedient solution for providing reliable, high-speed video over IP service.

III. JOINT SOURCE–CHANNEL ROBUST MULTISTREAM VIDEO TRANSMISSION

As we have pointed out in the introduction, although it is ideal to carry video over optical networks, the cost of providing the end-to-end optical connections is beyond the reach of the average users. Therefore, optical networks are most likely used in the core networks connecting different ISPs. In addition, the ubiquitous requirement of multimedia services excludes the use of optical networks. Therefore, some types of wireline or wireless access networks (last-mile services) are most likely needed to connect the end-users to the backbone ISPs, as shown in Fig. 1. However, any access networks may experience more or less transmission errors. Transmission errors can be roughly classified into two categories: 1) *random bit errors* and 2) *erasure errors*. How to provide an error-resilient transmission is a challenge for us, which leads to our work in the second part of the paper.

The random bit errors are caused by the imperfections of physical networks, which result in bit inversion, bit insertion, and bit deletion. Depending on the coding methods and the affected information contents, the impact of random bit errors can range from negligible to objectionable. When a fixed-length coding is used, a random bit error only affects one code word and the damage is generally acceptable. But if VLC i.e., Huffman coding, is used, random bit errors can cause the loss of synchronization so that many following bits are not decodeable until the next synchronization code word is received. In some cases, even after the synchronization is obtained, the decoded information can still be useless because there is no way for decoder to determine which spatial or temporal locations corresponding to the decoded information. Erasure errors, on the other hand, can be caused by the packet loss in packet networks, or system failures for a short time. Random bit errors in VLC can also cause erasure errors since a single bit error can lead to many following bits being undecodeable and hence useless. The effect of erasure errors is much more destructive than the random bit errors due to the loss or the damage of a contiguous segment of bits. Since almost all the state-of-the-art video compression techniques use VLC, there is no need to treat random bit errors and erasure errors separately. We therefore use the generic term "transmission errors" in this paper to refer to both random bit errors and erasure errors.

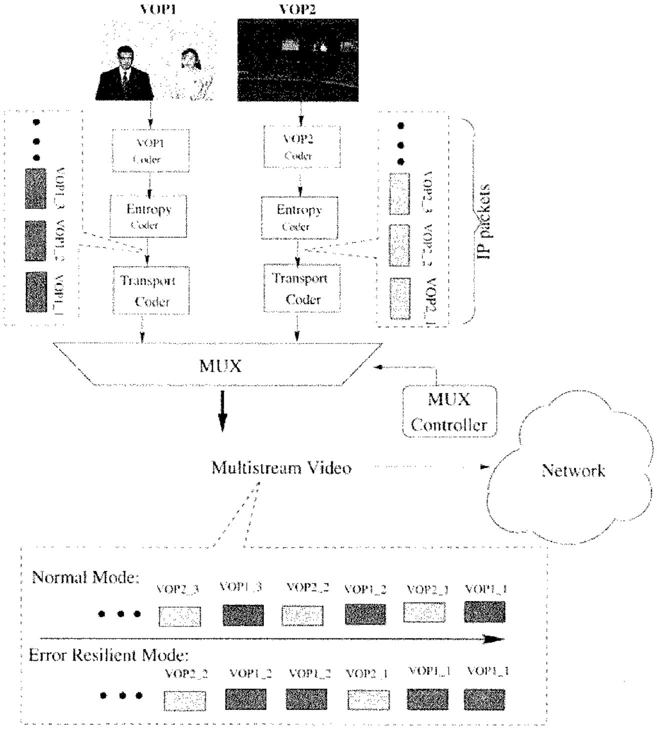


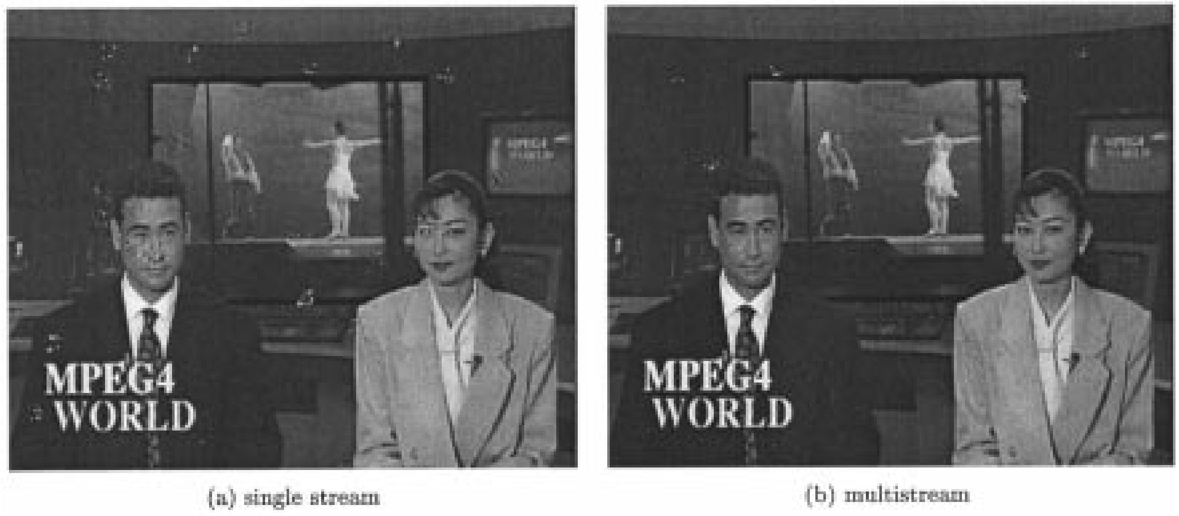
Fig. 6. Illustration of the transport coder with transport prioritization and its multistream video over IP design. Here, the drawing is not proportional to the real packet size.

With the development of video segmentation techniques [13], we can view, access, and manipulate video objects rather than frames of pixels, which is used in MPEG-1, MPEG-2, and H.263 video coding systems. By taking the advantage of content-based video coding, in this part of our paper, we propose a novel joint source-channel multistream video transmission scheme to combat transmission errors in access networks.

A. What is Unique in the Multistream Video Coding?

In our design, the video objects are encoded into different video streams, as shown in Fig. 6. Each video object is encoded differently based on its perceptual importance. The perceptual importance of the video streams can be determined based on the outputs of segmentation.

- *Easy case*: For the specific applications where the input signals come from a controlled environment or where the features to be analyzed are simple, the perceptual importance of video streams can be easily determined. For example, the video sequences are created by combining a foreground object that has been filmed over a blue screen with a background sequence that has been taken independently such as "Akiyo" test sequence. In the sequence, we simply refer the*primary video stream* to the video stream of the foreground video object, i.e., the person "Akiyo," and refer the *secondary video stream* to the stream of the background video object.
- *Difficult case*: The determination of perceptual importance can be difficult for generic images or video sequences because the perceptual importance of different objects dynamically varies at different occasions. Therefore, the determination depends very much on the segmentation techniques used, which still have to be improved [13]. Our multistream scheme works based on the



(a) single stream (b) multistream

Fig. 7. Video quality of different coding schemes: (a) single stream and (b) multistream.

assumption that we can segment the video sequence and distinguish the significance of different video streams.

On the top of traditional error control techniques, we propose a joint source-channel multistream video coding scheme to combat the transmission errors under the harsh network conditions. The basic idea behind this multistream scheme is to send the same packets of the primary video streams *twice* in the consecutive order under the harsh network conditions. Most likely, these dual description packets end up going through the different routes/paths of the networks. Because the chance that all paths simultaneously experience information losses is small, the decoder has better chance to reconstruct the original video or to recover some useful information based on the *dual* descriptions of primary video stream than the single description approach, as shown in Fig. 7. In this figure, all errors are isolated as MB-errors because we adopt MB-based scheme for the multistream design. In other words, instead of grouping consecutive MBs in the scanning order, an MB belongs to exactly one slice in our design (this type of design is quite common [31]). One reason why we adopt the MB-based instead of the slice-based scheme is that we try to minimize the impact of transmission error propagation due to the harsh network condition and isolate the error only to the affected MB. For example, in wireless communication, signal fading due to time-variant multipath propagation often causes the signal to fall below the noise level, thus resulting in a large number of errors. As a result, the bursty transmission error may render the consecutive video MB useless if we group them together. Therefore, we use MB-based coding and adopt the MB-based IP packet instead of the slice-based IP packet for transmission. The same argument also holds true for the wireline channels. For the wireline packet-switch networks, the routers i.e., CISCO routers are designed and configured based on the open shortest path first (OSPF) protocol for equal cost load balance. It routes the consecutive IP packets from the same source via different paths to the destination. As a result, we try to keep the IP payload as small as possible (MB-based versus slice-based). The other more important reason is that we try to keep the repeated information as small as possible and use the MB-based IP packet instead of the slice-based IP packet so that it is easier for us to keep the constant bandwidth in our multistream design. Furthermore, our multistream is designed to carry not only just bursty traffic, but also constant-bit-rate real-time video traffic. A long IP payload would have a negative impact on this type of traffic flow. Therefore, our MB-based design is also chosen for minimizing the system delay.

A common question is: "The idea of using object-based coding for error-resilience is not new. It was one of the main reasons that MPEG-4 develops the object-based coding scheme. What is new in our design?" The conventional content-based design codes the video into different video streams and transmits each video stream only *once* no matter whether the channel condition is good or bad. However, our multistream design distinguishes the primary video streams among the other video streams (the secondary video streams). We transmit the video streams according to the significance of the video streams under the adverse channel conditions. Sometimes, we may stop transmitting those secondary video streams and allocate their bandwidth resource to transmit the primary video streams *again*. It is important to point out that, under the normal channel condition, we still use the same approach as the conventional single stream video approach by transmitting all video streams *once* regardless of the primary or secondary video streams. Another common concern is: "In the case of adverse network, it may increase the bandwidth requirement and worsen the network traffic condition without improving the QoS by throwing in more packets in the multistream design." Instead of simply throwing in more packets in the communication pipes and dynamically increasing the bandwidth requirement, our multistream scheme keeps the *same* bandwidth requirement by adjusting the quantization step, adopting transport prioritization and sometimes transmitting only the primary video streams. The term transport prioritization here refers to various mechanisms to provide different quality of different video streams in transport, including using unequal error protection [31], [32] and assigning different priorities to different video streams. Our goal is to ensure that the critical information can reach the decoder while still keeping the same bandwidth requirement throughout the service because the bandwidth requirement is normally negotiated before the communication pipe is setup. During the multimedia service, it is impractical to expand the network bandwidth dynamically to accommodate the transmission of multiple copies of the video streams because it may affect the other users and only worsen the network traffic condition without improving the QoS. Overall, the multistream scheme requires the same bandwidth as the conventional singlestream approach. Next, we will explain our design in more detail.

B. The Design of Multistream Video Coding

The video objects are lossily encoded so that it reduces the bit-rate by representing the original video using some transformed variables, then applying quantization. In order to facilitate the rate-control transmission, we apply tight control on the encoded bit-rate so that different video objects may have different quantization step sizes [33], [34]. At the decoder side, the quality of the reconstructed video object is highly correlated with the quantizer step size used by the encoder. For instance, the larger the step sizes, the lower the bit-rate thus the lower the quality of the secondary video streams. The commonly used variable length entropy coder, i.e., Huffman coder is then employed after quantization. After the entropy coding, the overhead information such as resynchronization markers, data partitioning markers, etc., as shown in Fig. 5, are attached to the video streams. It leads to a common question: “MPEG-2 also has video scalability mechanism built for error-resilience and rate-control transmission, why don’t we consider it?” The difference between our approach and the MPEG-2 design is that we manipulate video objects rather than the frame of pixels. In MPEG-2, all video objects within a frame are treated equally important and the bandwidth is allocated evenly to transmit the coded information. To keep the same bandwidth requirement, our multistream design can not only adjust the quantization step like the MPEG-2 approach, but also adopt transport prioritization by sacrificing the secondary video information for the primary video information. The transport coder with transport prioritization refers to as an ensemble of devices performing channel coding, packetization and/or modulation, as shown in Fig. 6. For instance, the router will make the best effort to deliver those high priority packets associated with the primary video stream in congested networks. It results in less objectionable visual distortion than MPEG-2 design especially when the system throughput rate is low. The “MUX controller” in Fig. 6 controls whether the encoded IP packets are transmitted in either the “normal mode” or “adverse network mode” based on the network conditions. Here, the feedback information about the network condition can be obtained by using the delay and loss-rate statistics at the decoder [11], [35]. Under the *normal mode*, those encoded IP packets are multiplexed and sent out alternatively. In other words, we transmit all video streams regardless of the primary or secondary video streams similar to the single stream design. When the network condition deteriorates then “the MUX controller” switches to “adverse network mode”, and the IP packets of the primary video stream are repeated once and then sent out, as shown in Fig. 6. Compared to the processing of the primary video stream, the rest of secondary video streams are more coarsely quantized, less error protected coded and transmitted at low priority. Providing that the motion of those secondary video objects is zero or low at that moment such as in the “Akiyo” test sequence, we can even stop transmitting those background secondary video streams. The other common question is: “The retransmission is usually taken care of by the ARQ and TCP. Why should we design the multistream video transmission?” The reason to introduce the multistream video transmission is because sometimes it may not be feasible or cost effective in the certain applications to guarantee the lossless video transmission. For instance, the real-time and interactivity requirements exclude the deployment of some well-known error-recovery techniques such as the ARQ retransmission. In addition, issues such as audio-visual synchronization and multipoint communications further complicate the problem of error recovery. As a result, the self-recovery transmission mechanism (or joint source-channel transmission mechanism) is more desirable. If the system delay is not an issue such as in non-real-time data-oriented services, then the “resend” is taken care of by the TCP layer, and ARQ in the link layer or the media access control (MAC) layer. Under such a circumstance, we can send the multiple copies of the video streams regardless of the primary or secondary video streams to the receiver to reduce the transmission errors under the harsh network conditions. However, it is not the case for the real-time video services because dynamically changing the overall system delay is not tolerant.

Our multistream transmission scheme works based on the fact that there are several parallel paths/routes between the source and destination. Each path may be temporarily down or suffer from bursty errors. However, the error events of different paths are independent so that the probability that all paths simultaneously experiencing losses is small. These wireline/wireless paths can be physically distinct between the source and destination. For example, in wireline packet-switch networks, the routers (like CISCO routers) are designed and configured based on the OSPF protocol for equal cost load balance. It routes the IP packets from the same source via different paths to the destination. In addition, it intelligently redirects the traffic flow once adverse channel condition encountered instead of keeping throw the packets in the already congested path. Therefore, our multistream design outperforms the conventional design under the harsh network condition because the chance of the dual descriptions of the video streams via different paths simultaneously experiencing information loss is small. As for the wireless communication, the mobile is moving during the transmission. Therefore, the channel condition at time t may very well be different than that at time $t + 1$. Furthermore, the signals can be deflected by surrounding infrastructures. Under the multipath fading condition, there are many paths between transmitter and receiver. In our multistream design, the dual descriptions of video streams at time t and $t + 1$ independently travel through the different paths thus introducing the diversity. Because the chance of all paths simultaneously experiencing information loss is small, the sum of signal level at the receiver antenna in our multistream design is stronger than that in the single stream design; thus achieving better picture quality. Even when only one single physical path exists between the source and destination, the path can be divided into several virtual

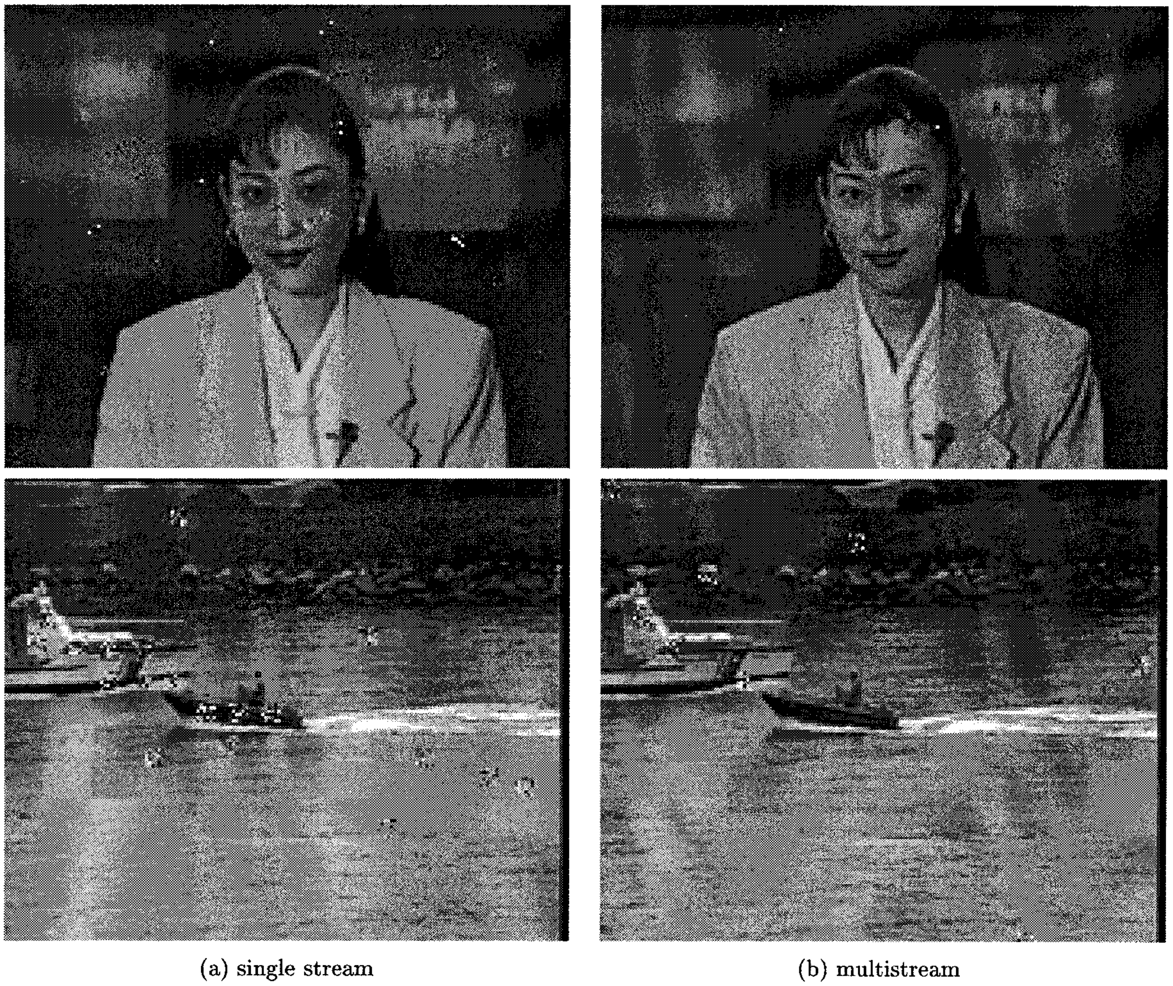


Fig. 8. Transmission errors affect the video quality by using different coding schemes.

channels by using the time interleaving scheme, as shown in Fig. 6. Overall, under the adverse channel conditions, the decoder can still reconstruct the original video sequence or recover some useful information depending on the stream which is received correctly or least distorted. The resolution of the reconstructed video at the decoder side is gracefully improved depending on the number of different descriptions received correctly. The other question is: “What is the difference between our design and the conventional multiple-description coding?” Unlike the conventional multiple-description coding [36], [37], we apply the content-based video coding instead of the layered coding. In the layered video coding approach, all video objects within a frame are treated equally important and the bandwidth is allocated evenly to transmit those coded information. On the other hand, in content-based video coding, video information is partitioned into more than one video object and encoded into multiple video streams. Combining different video streams with different transport priorities, our emphasis is on the robust transmission of the primary video streams. Under the harsh network condition, we allocate more resource to error protect those important video information. In principle, we trade the quality of the least significant video objects for the quality of the most significant ones. As a result, we can reconstruct a close approximation of the original video (at least for the primary video object) and make the output video at the decoder least objectionable to human eyes.

C. Simulation Results

To test the performance of our proposed multistream technique to combat the transmission errors in the hybrid access networks, we performed several experiments by taking some MPEG-4 test sequences in CIF or QCIF formats as input (Each sequence contains 300 video frames and we assume that the significance of different video objects is known in advance). The segmented video objects in each test sequence are differently coded into different video streams using our novel compressed domain content-based video coding scheme proposed in Section IV. The video streams are then quantized and transported with transport prioritization to achieve the heterogeneous video qualities for different video objects (the primary and secondary video streams). The syntax of both the YUV MB and its associated binary alpha block (BAB) supports the communication of information relating to the decoded video quality. Furthermore, the same quantization functionality for the YUV block is also applied to the compressed BAB. In our experiments, the quantization parameters and channel coding rates are chosen so that the combined output rates of video streams remain *constant* thus the same required bandwidth throughout the service.

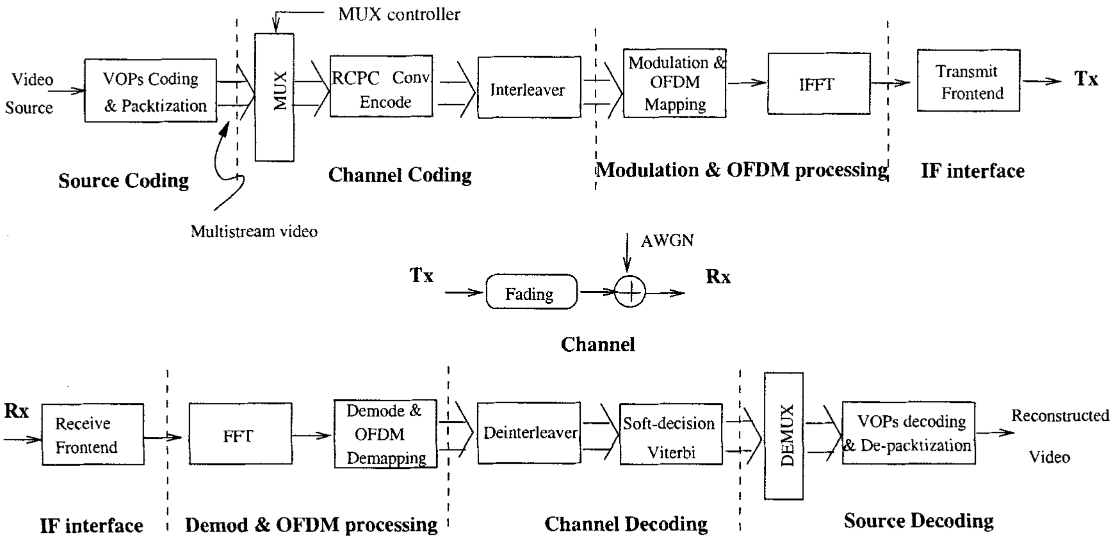
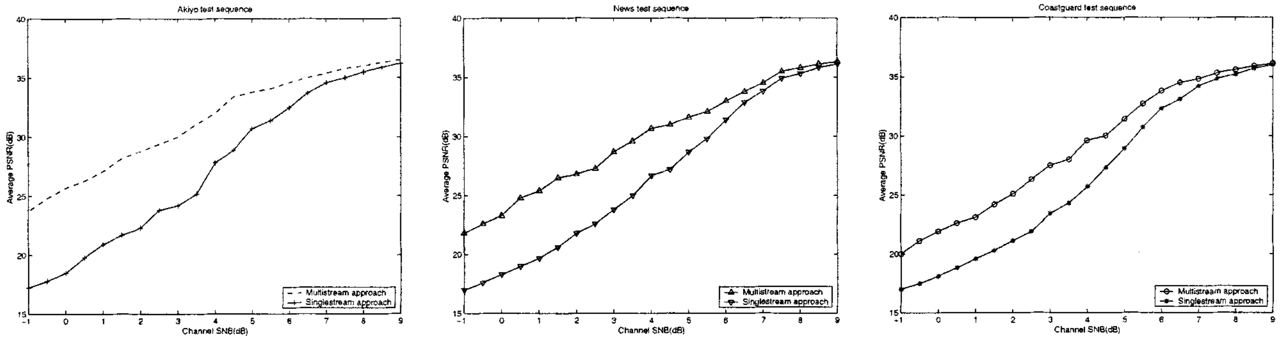


Fig. 9. Simulation setup for testing the performance of our multistream video coding scheme under the wireless channel conditions.



(a) “Akiyo” (b) “News” (c) “Coastguard”

Fig. 10. Performance comparison between our proposed multistream design and the conventional single stream design.

Due to our joint source-channel coding design, the rate-compatible punctured convolutional (RCPC) [38] code is employed for channel coding in our simulations. It provides an efficient means in implementing a variable-rate error control for different video streams so that only a single encoder/decoder pair is needed. In addition to that, the orthogonal frequency division multiplexing (OFDM) is used for modulation because it increases the robustness against frequency selective fading or narrow-band interference (For detailed information about OFDM design, please refer to [39]). At the decoder side, the soft-decision Viterbi decoding scheme is utilized to improve the error-correction capabilities because it outperforms the hard-decision one by 2 dB. We test the performance of our proposed scheme under both the wireline and wireless access channels. Our simulations are performed on the added white Gaussian noise (AWGN) channel. In addition, for our wireless channel simulation, we adopt the Rayleigh fading channel to model the bursty channel condition caused by multipath fading [40]–[42] (Under the urban transmission environment, there are up to six different paths/rays in our multipath fading model, which is the same as the commonly used urban channel models).

In our wireline simulation, we assume that there are two distinct routes between the source and destination. The router, which acts like the CISCO router, at the encoder side distributes the video IP packets via the two paths independently. The transmission errors affect the video quality by using different coding schemes, as shown in Fig. 8. For the simple cases such as “Akiyo” and “Mother and Daughter” test sequences, where the foreground scenes change while the background scenes remain the same, we can allocate all bandwidth under the harsh network condition to transmit the primary video object, because it dominates the whole video scene. Our multistream works extremely well for those simple test cases. For the complicate cases such as the “Coastguard” test sequence in which both the foreground and background scenes change, we allocate more bandwidth to transmit the primary video streams than the secondary ones when the network condition deteriorates. Compared to the conventional layered coding mechanism, our multistream works better in terms of error resilience. From the simulation results, we observe that the multistream approach outperforms the single stream approach by 3–5 dB under the noisy channel conditions.

Under the wireless multipath fading condition (we use the carrier frequency of 900 MHz, the mobile velocity of 10–20 mi/h with up to six propagation paths and the system throughput rate is around 128 kb/s), our simulation with the setup, as shown in Fig. 9, demonstrates that our multistream works better than the single stream design by up to 5–7 dB, as shown in Fig. 10.

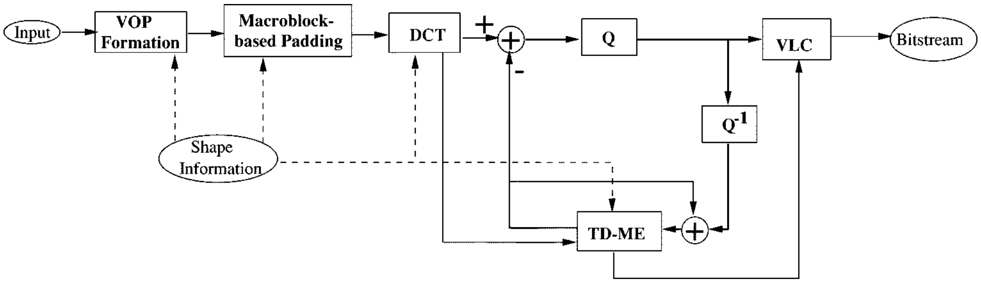


Fig. 11. Our fully DCT-based coder structure. Here, TD-ME stands for estimates motion in the transform domain.

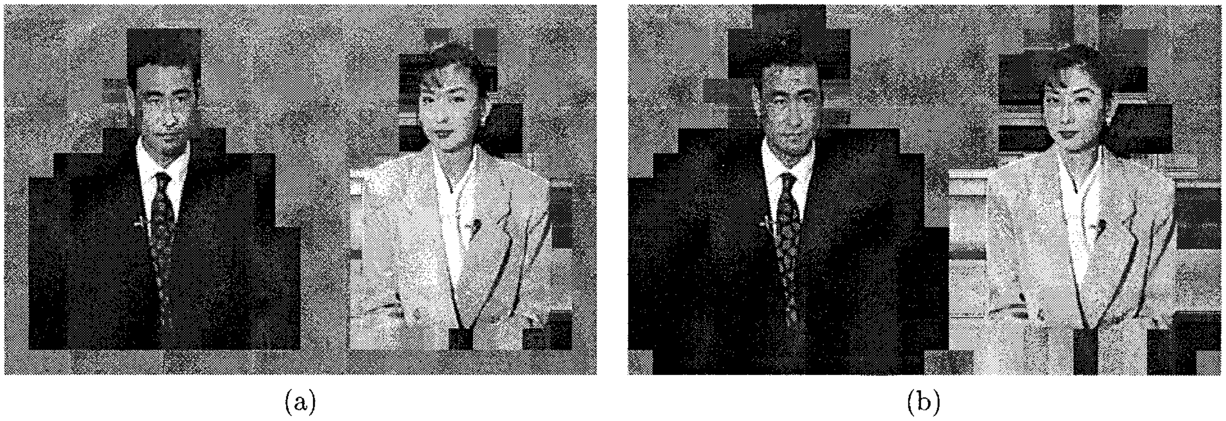


Fig. 12. Padding technique is applied to the VOP of "News" sequence: (a) repetitive padding of VOP and (b) extended padding of VOP.

The reason is that the IP packets of primary video streams are repeated once and then sent out, thus introducing the signal diversity under the harsh network condition. As a result, the dual descriptions of primary video stream independently travel through the different fading paths. Because the chance that all paths simultaneously experience information loss is small, the sum of the signal level at the receiver antenna in our multistream design is stronger than that in the single stream design (In our design, the OFDM is used to increase the robustness against the frequency selective fading or the narrow-band interference). As a result, the decoder can reconstruct a better original video sequence depending on the stream which is received correctly or least distorted by taking advantage of the multiple paths between source and destination.

We also perform tests to compare our multistream design with the MPEG-2 and multidescription rate-control, error-resilient coding schemes. Under the adverse channel condition, we transmit every frame instead of the primary video streams twice. Provided that the system throughput is low ≤ 64 kb/s as in video phone applications, we can only use the coarse quantization step to keep the constant bandwidth during the service. Based on the simulation results, the coarse quantization causes more objectionable visual distortion. Therefore, to support our multistream video coding scheme, it is better to adopt the transport prioritization and to access and manipulate video objects rather than the frame of pixels.

IV. Coding of Arbitrary Shape Video Fully in the Transform Domain

In the second part of this paper, we have discussed the joint source-channel multistream coding. Now, the question becomes "What content-based source coding scheme should we use?" Surely, we can adopt the MPEG-4 content-based video coding design to meet our multistream design requirement. However, in this part of our paper, we propose "Arbitrarily shaped video coding in the DCT-domain," instead. Instead of treating the moving picture as a single entity in the MPEG-1 and MPEG-2 design, our multistream design treats the moving picture as an organized collection of visual objects and encodes them into different video streams. Furthermore, an image or a video sequence may be constructed by the composition of one or more independently coded visual objects. Unlike the conventional MPEG-4 arbitrarily shaped visual coding approaches performing the compression in DCT-domain and the motion estimation/compensation in the spatial domain, it is worth pointing out that we achieve our content-based video compression, motion estimation, and compensation *all* in the DCT-domain. The advantage of such an approach is that we can optimize the hardware complexity by combining both DCT and motion estimation/compensation units into a single unit, thus reducing the computational units along the data path.

A. Compressed Domain Content-Based Video Coding

Besides the spatial domain motion estimation, manipulating video data in the DCT-domain has actually long been recognized as more efficient in many advanced video applications [43]. In [43], Chang and Messerschmitt derive a complete set of algorithms for the manipulations of compressed video signals such as overlap translation, scaling, linear filtering, rotation, and pixel multiplication in the DCT-domain. In addition, they propose the idea of performing compression using DCT without motion compensation (MC) in the spatial domain. However,

TABLE IV
MACROBLOCK-BASED REPETITIVE PADDING PROCEDURES

Steps	Detail Procedures
1	Consider each undefined pixel outside the object boundary a zero pixel.
2	Scan each horizontal line of a block (a block could be 16 × 16 or 8 × 8). Each scan line is possibly composed of two kinds of line segments: • zero segments that have all zero pixels within each segment • non-zero segments that have all non-zero pixels within each segment. – If there are no non-zero segments, do nothing. – Otherwise, there are two situations for a particular zero segment: • it can be positioned between an end point of the scan line and the end point of a non-zero segment. Then, fill all of the pixels in the zero segments with the pixel value of the end point of the non-zero segment. • it can be positioned between the end points of two different non-zero segments. Then, fill all of the pixels in the zero segments with the average pixel value of the two end points.
3	Scan each vertical line of the block and perform the identical procedure as described in Step 1 to each vertical line.
4	If a zero pixel can be filled in by both Steps 2 and 3, the final value takes the average of the two possible values.
5	Consider the rest of zero pixels: • scan any one of them horizontally to find the closest non-zero pixel on the same horizontal scan (if there is a tie, the non-zero pixel to the left of the current pixel is selected) • scan any one of them vertically to find the closest non-zero pixels on the same vertical scan (if there is a tie, the non-zero pixel on the top of the current pixel is selected). Replace the zero pixel by the average of these two horizontally and vertically closest non-zero pixels.

TABLE V
COMPUTATIONAL COMPLEXITY OF STEP 2.1–2.5 IN OUR DESIGN FOR A MACROBLOCK SIZE OF $N \times N$ (HERE WE USE $N = 16$ AS AN EXAMPLE)

Step	Operation	Multiplication	Additions/ Subtractions	Computational complexity
2.1	2D-DCT (type-II) computation	48	200	$O(N^2)$
2.1	Rotation (type-I DCT)	64	0	$O(N^2)$
2.2	Pseudo phases calculation	160	32	$O(N^2)$
2.3	$F(\cdot,\cdot), G(\cdot,\cdot)$ computation	48	200	$O(N^2)$
2.4	Half-pel motion estimation	32	64	$O(N^2)$
2.5	Prediction errors computation	0	16	$O(N)$
	Total	322	512	$O(N^2)$

nothing is mentioned about how to achieve the combined design of DCT and motion estimation units, which consume most of the computing power in a video coder. In [19], Koc and Ray Liu propose a detailed framework concerning how to perform motion estimation in the DCT-domain. The scheme is proposed as an alternative cost-effective solution to outperform the then state-of-the-art coding standards such as H.261, MPEG-1, and MPEG-2 for *rectangular* frame video coding. Nothing about the arbitrarily shaped video coding is mentioned.

In recent years, many communication/interactive applications have appeared on the Internet. Unlike MPEG-1 and MPEG-2, wherein the emphasis is primarily on coding efficiency, the content-based video coding is the main theme of MPEG-4. Although the new or modified techniques such as the shape coding are introduced in MPEG-4, the DCT and block-based motion estimation are still the fundamental techniques in the standard. The difference between our work in this part of paper and that in [19] bears some analogy to the difference between video coding scheme in MPEG-4 and that in MPEG-1 and MPEG-2. In other words, the DCT pseudophase technique is still the kernel of our design. Our fully DCT-based coder structure is shown in Fig. 11. We combine both the DCT coder and the motion estimation/compensation processors into one unit, and remove the inverse IDCT unit from the feedback loop. All cost-effective, high throughput advantages (motion estimation/compensation in the DCT-domain versus the MPEG approaches) mentioned in [19] are also applied in our design. In fact, we can view the motion compensated video coding of rectangular frames in the DCT-domain as the special case of our compressed domain arbitrarily shaped video coding. In what follows, we will focus on the difference between our newly proposed scheme and that in [19].

1) MB-Based Repetitive Padding: The MB-based repetitive padding is required to estimate motion for the *contour* MBs which reside on the boundary of the video object and contain partial video information, as shown in Fig. 12. The procedures of padding are listed in Table IV.

2) Binary Shape Coding: The arbitrary shape of the video objects can be described in either the binary or gray mode. In this paper, we are only interested in the binary mode because it forms the simplest class of objects. The video objects and shapes are separately coded and the suitable shape coding methods include:

1) *vertex-based coding* [44] and *chain coding* [45] in contour-based approaches;
2) *modified-READ (MR) method* [46] and *context-based arithmetic encoding (CAE)* [47] in bitmap-based approaches;
3) *Chroma-keying* [48].

TABLE VI
SUMMARY OF ARBITRARY SHAPE VIDEO CODING IN THE DCT-DOMAIN

Input: The video object planes (VOPs).
Output: The motion vectors and prediction errors.

1. **VOP formation and padding**: Based on the shape information of the VOP, generate the tightest rectangle – bounded VOP window that contains the video object to achieve high coding efficiency. The bounded window has the minimum number of macroblocks with each of size 16 × 16 pels. A shift parameter is hereafter encoded as "hor/ver_spat_ref" in MPEG-4 [15] to indicate the location of the bounded VOP window with respect to the borders of a reference VOP. The VOP is then macroblock-based repetitive padded.

2. **Content-based motion estimation/compensation**: The motion vector is computed only for each of the macroblock (or the 8 × 8 block for advanced motion compensation) which contains the video object. Otherwise jump to Step 3.

2.1 Compute the 2-D DCT coefficients of second kind (type-II), $X_t^{cc}(k,l)$, $X_t^{cs}(k,l)$, $X_t^{sc}(k,l)$ and $X_t^{ss}(k,l)$, of a macroblock of pixels in the current VOP, x_t, such as:

$$X_t^{sc}(k,l) = \frac{4}{N^2}C(k)C(l)\sum_{m,n=0}^{N-1} x_t(m,n)\sin[\frac{k\pi}{N}(m+0.5)]\cos[\frac{l\pi}{N}(n+0.5)], \quad \text{for} \quad k\in\{1,\ldots,N\},\quad l\in\{0,\ldots,N-1\}. \tag{1}$$

Meanwhile, the corresponding macroblock of pixels in the reference VOP, x_{t-1}, are converted to 2-D DCT coefficients of first kind (type-I), $Z_{t-1}^{cc}(k,l)$, $Z_{t-1}^{cs}(k,l)$, $Z_{t-1}^{sc}(k,l)$ and $Z_{t-1}^{ss}(k,l)$ such as:

$$Z_{t-1}^{cs}(k,l) = \frac{4}{N^2}C(k)C(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x_{t-1}(m,n)\cos[\frac{k\pi}{N}(m)]\sin[\frac{l\pi}{N}(n)], \quad k\in\{0,\ldots,N\}, l\in\{1,\ldots,N-1\}. \tag{2}$$

2.2 Determine the pseudo phases $f(k,l)$ and $g(k,l)$ from *system equation*:

$$\vec{\mathbf{Z}}_{t-1}(k,l)\,\vec{\theta}_{m,n}(k,l) = \vec{\mathbf{X}}_t(k,l) \qquad \text{for } k,l\in\{1,\ldots,N-1\}. \tag{3}$$

where

$$\vec{\mathbf{Z}}_{t-1}(k,l) = \begin{bmatrix} Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) & -Z_{t-1}^{sc}(k,l) & Z_{t-1}^{ss}(k,l) \\ Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{ss}(k,l) & -Z_{t-1}^{sc}(k,l) \\ Z_{t-1}^{sc}(k,l) & -Z_{t-1}^{ss}(k,l) & Z_{t-1}^{cc}(k,l) & -Z_{t-1}^{cs}(k,l) \\ Z_{t-1}^{ss}(k,l) & Z_{t-1}^{sc}(k,l) & Z_{t-1}^{cs}(k,l) & Z_{t-1}^{cc}(k,l) \end{bmatrix},$$

which contains those *type-I* DCT coefficients obtained from Step 2.1. And, $\vec{\mathbf{X}}_t(k,l) = [X_t^{cc}(k,l), X_t^{cs}(k,l), X_t^{sc}(k,l), X_t^{ss}(k,l)]^{\mathbf{T}}$ contains those *type-II* DCT coefficients. $\vec{\theta}_{m,n}(k,l) = [\star, f(k,l), g(k,l), \star]^{\mathbf{T}}$, here $\star$ stands for don't care.

2.3 Find the inverse DCT of $f(k,l)$ and $g(k,l)$, respectively.

$$\begin{aligned} F(m,n) &= \frac{4}{N^2}\sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l)\cos\frac{k\pi}{N}(m+\frac{1}{2})\sin\frac{l\pi}{N}(n+\frac{1}{2}) \\ &= [\delta(m-m_u)+\delta(m+m_u+1)]\cdot[\delta(n-m_v)-\delta(n+m_v+1)], \end{aligned} \tag{4}$$

$$\begin{aligned} G(m,n) &= \frac{4}{N^2}\sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l)\sin\frac{k\pi}{N}(m+\frac{1}{2})\cos\frac{l\pi}{N}(n+\frac{1}{2}) \\ &= [\delta(m-m_u)-\delta(m+m_u+1)]\cdot[\delta(n-m_v)+\delta(n+m_v+1)], \end{aligned} \tag{5}$$

which are composed of impulse functions whose peak positions indicate the integer-pel motion vector (m_u, m_v) and peak signs reveal the direction of the movement.

In our design, we select the block-based methods of CAE for our shape coding. The adopted block-based syntax has allowed the compressed BABs to be blended seamlessly into the traditional video syntax structures, as shown in Fig. 5. This in turn eases the task of supporting the important features such as the error-resilient, bit-allocation, and rate-control operations [33], [34]. Just as with the YUV encoding, a BAB may be intra-coded using the CAE. Furthermore, it may be inter-coded using MC and CAE, or it may merely be reconstructed by the MC without CAE which analogous to the "not-coded" MB mode in the MPEG-4 video standard. Both the YUV and binary alpha decoding require the use of motion estimation/compensation to exploit the spatial redundancy.

3) Arbitrary Shape Video Coding Algorithm: In terms of arbitrary shape motion estimation, we can treat the contour MBs same as the regular ones except for pixels outside are padded based on the video content inside the video boundary by following the procedures in Table IV. In what follows, the video object plane (VOP) refers to the instances of video objects at a given time. Because our content-based video coding is the extension of the work in [19], please refer to [19] for the detailed explanation of the equations and definitions such as the "Type I" and "Type II" DCT, $\vec{\mathbf{X}}$, $\vec{\mathbf{Z}}$, etc. Our proposed scheme is summarized in Tables VI and VII.

Now let us take a look at the overall computational complexity. To process each VOP, Step 1 of the proposed approach (VOP formation and padding) is only needed to be executed once. Therefore, the overall computational complexity of our design is determined by the complexity of Steps 2.1–2.5, which serves as the computing engine of the whole design. The com-

TABLE VII
SUMMARY OF ARBITRARY SHAPE VIDEO CODING IN THE DCT-DOMAIN (CONTINUATION)

2.4 To obtain an estimation at half-pel accuracy, we can have better flexibility and scalability by first getting the integer-pel motion vectors (m_u, m_v) per macroblock. The half-pel motion vector is then determined by computing $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ for $u \in \{m_u - 0.5, m_u, m_u + 0.5\}$ and $v \in \{m_v - 0.5, m_v, m_v + 0.5\}$.

$$\overline{DCS}(u,v) = \sum_{k=0}^{N-1}\sum_{l=1}^{N} C(k)C(l)f(k,l)\cos\frac{k\pi}{N}(u+\frac{1}{2})\sin\frac{l\pi}{N}(v+\frac{1}{2}) \tag{6}$$

$$\overline{DSC}(u,v) = \sum_{k=1}^{N}\sum_{l=0}^{N-1} C(k)C(l)g(k,l)\sin\frac{k\pi}{N}(u+\frac{1}{2})\cos\frac{l\pi}{N}(v+\frac{1}{2}) \tag{7}$$

The peak value of $\overline{DCS}(u,v)$ and $\overline{DSC}(u,v)$ indicate the half-pel motion and its direction. Therefore, the half-pel motion vector is determined by only considering the nine possible positions around the integer-pel displacement without interpolation.

2.5 Based on the previously derived motion estimation, DCT of the motion-compensated residual DBD (Displaced Block Difference between current block $\mathbf{B}_{curr}$ and displaced reference block $\mathbf{B}_{ref}$) is computed as:

$$DCT\{\mathbf{DBD}\} = DCT\{\mathbf{B}_{ref}\} - DCT\{\mathbf{B}_{curr}\}.$$

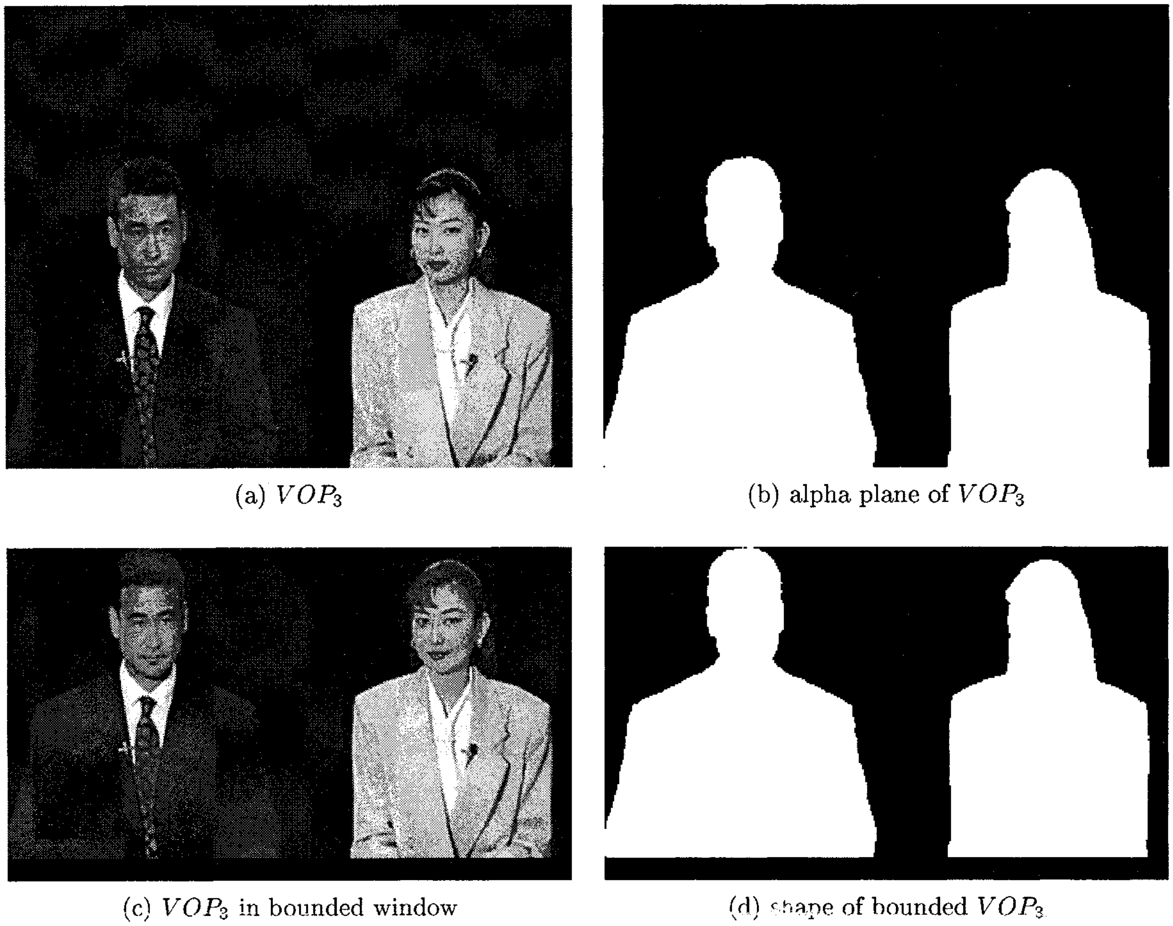
The prediction errors are then quantized and send to the receiver along with coded macroblock motion vectors.

3. **While loop**: Go to Step 2 until the whole video object is estimated/compensated. The process starts from the top left macroblock in the bounded VOP window to the top right one, and then to the next row, and so on for every macroblock in the bounded VOP window.

plexity of each step is listed in Table V. The reason we call our design the *content-based* video coding is that the motion estimation/compensation is performed only for those MBs containing the video information. Overall, the scheme requires the computational complexity of $O(N^2)$. Here, N stands for the MB size. As for our fully DCT-based design versus the conventional design in terms of computation saving, we indeed have performed some measurements. Compared to the fully search block matching scheme, our DCT-based design can save up to 60–75% computation depending upon the characteristic of incoming video sequence and other factors. (Although there are many fast search motion estimation schemes, we are not in the position to compare our design against all other approaches because it is not the main theme of this paper). For the large motions going beyond the block boundary, we will use motion vector (0,0), instead.

To facilitate the explanation of our proposed scheme, let us use "News" in CIF format with the frame size of 352×288 as input video sequence. The panorama scene is shown in Fig. 13. The "News" sequence consists of four VOPs and three corresponding binary alpha planes (the background VOP has no alpha plane). Here, we apply our design to VOP_3, the third VOP in Fig. 14(a), as an example to illustrate our design because it is the foreground VOP and most importantly the location and shape of VOP_3, as shown in Fig. 14(b), vary with time. The reason to introduce the VOP formation, which is the first step of our design, is to achieve high data compression rate because we do not need to estimate motions for those MBs containing no video information. After the VOP formation, the VOP_3 is now bounded by the tightest rectangle containing the video object. However, this tightest rectangle may not consist of multiples of MB of size 16×16. Therefore, we need to extend the bottom right coordinate of VOP_3 window in Fig. 14(a) to satisfy that requirement. The final bounded VOP_3 with the window size of 352×208 and its corresponding alpha plane are shown in Fig. 14(c) and 14(d), respectively. Then, the bounded VOP is padded, as shown in Fig. 12. The bounded VOP window is further divided into nonoverlapped MBs. The motion estimation process starts from the top left MB in the bounded VOP window to the top right one, and then to the next row, and so on for every MB in the bounded VOP window. The transform domain nature of the approach enables us to directly extract motion vectors out of the consecutive VOPs of a moving object. To envision how it works, we present an example to estimate motion of a contour MB by following Steps 2.1–2.4 in our design, as shown in Fig. 15. The peak position among $F(m,n)$ and $G(m,n)$ indicates the integer-pel motion vector of (3, 2). The peak position among $\mathrm{DSC}(u,v)$ and $\mathrm{DCS}(u,v)$ implies the half-pel motion vector of (3.5, 2.5).

Fig. 13. Panorama scene of "News" in CIF format.


(a) VOP_3

(b) alpha plane of VOP_3

(c) VOP_3 in bounded window

(d) shape of bounded VOP_3

Fig. 14. VOP formation on VOP_3 of "News": (a) VOP_3; (b) alpha plane of VOP_3; (c) VOP_3 in bounded window; and (d) shape of bounded VOP_3.

After motion estimation, the current block $\mathbf{B}_{\mathrm{curr}}$ of size $N \times N$ in the current frame can be best predicted by the block $\mathbf{B}_{\mathrm{ref}}$ displaced from the previous block position with the estimated motion vector (m_u, m_v). Based on the derivation in [21], [43], the DCT of the motion-compensated residual in terms of the displaced block difference (DBD) is given by

$$\begin{aligned}\mathrm{DCT}\{\mathbf{DBD}\} =& \mathrm{DCT}\{\mathbf{B}_{\mathrm{ref}} - \mathbf{B}_{\mathrm{curr}}\}\\ =& \mathrm{DCT}\{\mathbf{B}_{\mathrm{ref}}\} - \mathrm{DCT}\{\mathbf{B}_{\mathrm{curr}}\}.\end{aligned}$$

In other words, the DCT of the motion-compensated residual can be expressed as the difference between the DCT of the displaced block and the DCT of the current block. As a result, we can perform MC in the DCT-domain, as shown in Fig. 11, which serves the purpose of building a fully DCT-based motion compensated video without converting back to the spatial domain. Most importantly, we can efficiently combine the design of the DCT coder and the DCT-domain motion estimation/compensation unit into one unit thus significantly reducing the hard complexity.

Now the question becomes: "How do we extract the displaced DCT block in the DCT-domain or how to compute $\mathrm{DCT}\{\mathbf{B}_{\mathrm{ref}}\}$?" For the integer pixel and subpixel MC in DCT-domain, we adopt the bilinear interpolation method proposed in [21], [43].

B. Simulation Results

Simulations have been performed on the "News" sequence in the CIF format. The bounded previous and current VOPs are shown in Fig. 16(a) and (b), respectively. The reconstructed VOP using our proposed compressed domain coding scheme is shown in Fig. 16(d). The simulation results demonstrate the comparable video quality between the reconstructed and current VOPs. Considering that the motion compensated video coding of the rectangular frame is the special case of our arbitrarily shaped video coding, it is easy to see that our proposed design backward compatible to code regular images. In other words, we can view our approach as a logical extension of those DCT-based motion estimation schemes [19], [20] toward coding video sequences of arbitrary shape.

Due to its lower computational complexity as compared to other difference measures, the sum of absolute difference (SAD) is adopted in the MPEG-4 standards to measure the prediction errors [14], [15]. Simulations have also been performed to compare our design with the modified block matching (or polygon matching) method used in MPEG-4 in terms of prediction errors. Here, the MPEG-4 video reference software (MoMuSys) is used as reference in simulating the performance of modified block matching approach. The results are shown in Fig. 17. Furthermore, the simulation results demonstrate that the comparable performance of both our de-

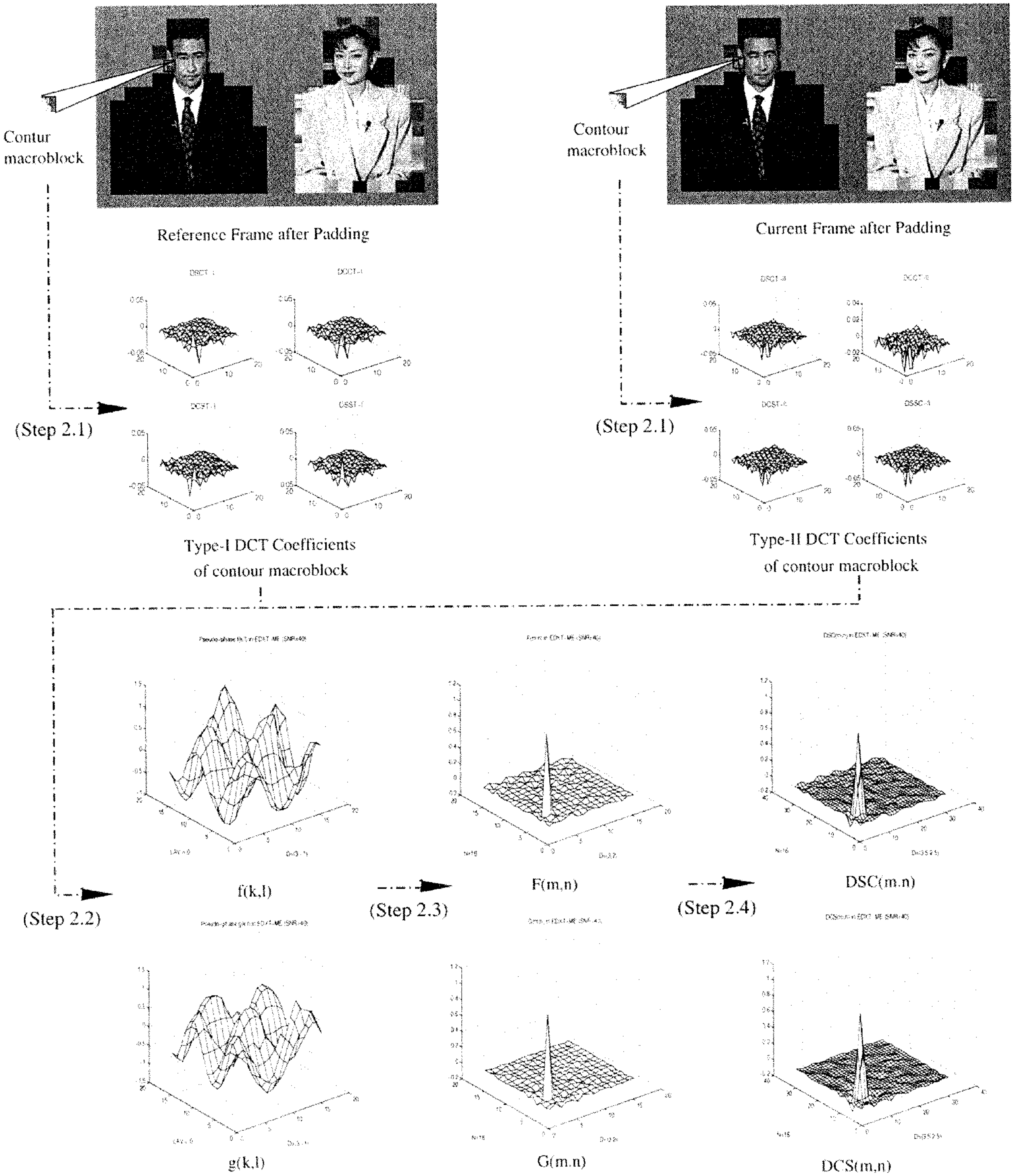

Contur macroblock
Reference Frame after Padding
Contour macroblock
Current Frame after Padding
(Step 2.1)
(Step 2.1)
Type-I DCT Coefficients of contour macroblock
Type-II DCT Coefficients of contour macroblock
(Step 2.2)
f(k,l)
(Step 2.3)
F(m,n)
(Step 2.4)
DSC(m,n)
g(k,l)
G(m,n)
DCS(m,n)

Fig. 15. Motion estimation of a contour MB by following Steps 2.1–2.4 in our compressed domain design.

sign and the one used in MPEG-4 in terms of prediction errors. Compared to the conventional arbitrarily shaped video coding design, we optimize the hardware complexity by minimizing the computational units along the data path thus more cost effective.

Other than the "News" test sequence, the simulations are also performed for "Foreman" and "Mother and Daughter" sequences, etc. In order to show that our design is also backward compatible to handle the rectangular frame of video, here we treat "Mother and Daughter" sequence as the regular frame of pixels. The simulation results, as shown in Fig. 18, demonstrate the comparable video quality between our compressed domain design and the conventional approach used in video standards.

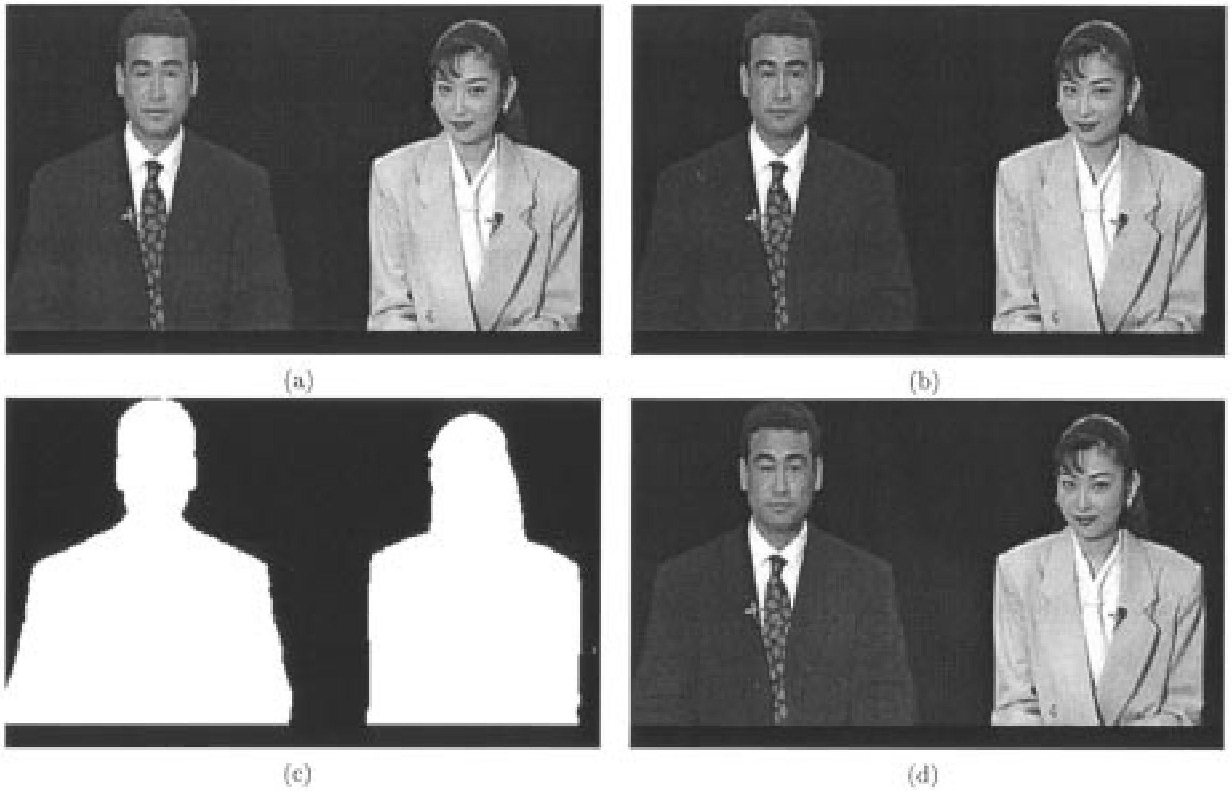


Fig. 16. Illustration of the performance of our content-based video coding: (a) bounded previous VOP; (b) bounded current VOP; (c) bounded alpha plane for previous VOP; and (d) reconstructed VOP using our proposed design.

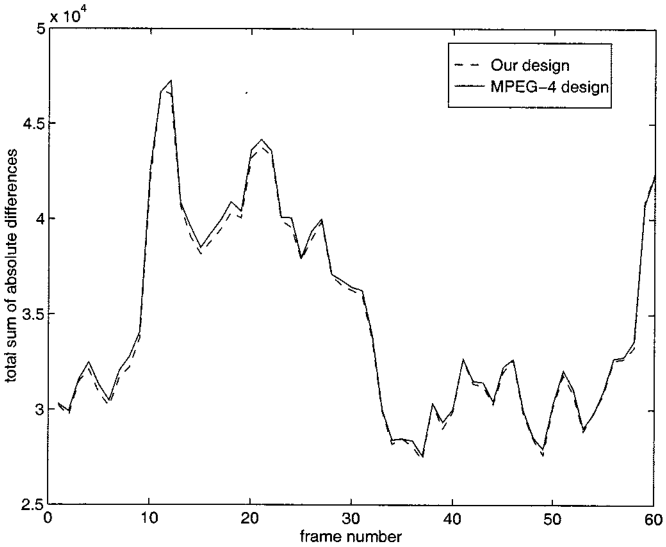


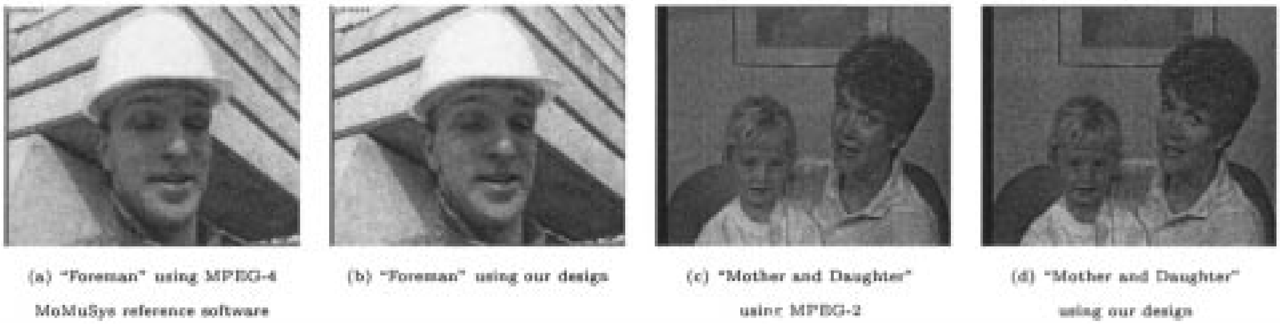
Fig. 17. Comparing the performance between MPEG-4 and our video coding approaches in terms of prediction errors using "News" testing sequence. Here, total sum of absolute differences is the summation of SAD for all MBs within each frame: (a) "Foreman" using MPEG-4 MoMuSys reference software; (b) "Foreman" using our design,; (c) "Mother and Daughter" using MPEG-2; and (d) "Mother and Daughter" using our design.

V. Conclusion

To optimize the performance of the multimedia over IP systems with the given QoS requirements, we should *jointly* consider video compression and delivery schemes based on the network alternatives, capacities, and characteristics. This paper discusses three subtopics.

1) A SONET network adapter design served as the Layers 1 and 2 (the physical and data link layers) IP router to deliver packet video over SONET/SDH or directly over fiber to provide reliable, high-capacity backbone connections.
2) A joint source-channel multistream video coding scheme to combat the transmission errors in hybrid access networks under the harsh network conditions.
3) Compressed domain content-based video coding to reduce the hardware complexity and to improve the encoder performance.

Although these three subtopics can be separately presented, we try to put them together as a system level paper. The reason is that, under the current communication environment, the video over IP services most likely go through hybrid networks (core networks, access networks) with different characteristics. Our design targets at the different problems associated with hybrid networks, i.e., robustness transmission in access networks, multivendor interoperability in the design of backbone devices, etc. Also, we present some novel thoughts in designing and implementing video over IP systems. The simulation results have confirmed our ideas. Overall, our goal is to improve the quality of video over IP service under the current communication environment.



(a) "Foreman" using MPEG-4 MoMuSys reference software

(b) "Foreman" using our design

(c) "Mother and Daughter" usinr MPEG-2

(d) "Mother and Daughter" using our design

Fig. 18. Comparing the video quality of MPEG-2 and our video coding approaches in terms of video quality.

REFERENCES

[1] M.-C. Chow, *Understanding SONET/SDH: Standards and Applications*: Andan Publisher, 1995.

[2] "PPP over SONET/SDH,", IETF RFC 1619, 1994.

[3] T. C. Kwok, "PPP over ATM," *IEEE Commun. Mag.*, vol. 37, pp. 84–89, May 1999.

[4] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 801–806, June 1989.

[5] Y.-Q. Zhang, Y. J. Liu, and R. L. Pickholtz, "Layered image transmission over cellular radio channels," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 786–796, Aug. 1994.

[6] M. Khansari and M. Vitterli, "Layered transmission of signals over power-constrained wireless channels," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, Oct. 1995, pp. 380–383.

[7] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and cell loss recovery for DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, June 1993.

[8] S. S. Hemami and T. H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Processing*, vol. 4, pp. 1023–1027, July 1995.

[9] P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, San Francisco, CA, Mar. 1992, pp. 545–548.

[10] W. Wada, "Selective recovery of video packet loss using error concealment," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 807–814, June 1989.

[11] S. Cen, C. Pu, R. Staehli, C. Cowan, and J. Walpole, "A distributed real-time MPEG video audio player," in *Proc. 5th Int. Workshop Network and Operating System Support for Digital Audio and Video*, Durham, NH, Apr. 1995, pp. 151–162.

[12] M. Ghanbari, "Postprocessing of late cells for packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 669–678, Dec. 1996.

[13] P. Salembier and F. Marques, "Region-based representations of image and video: Segmentation tools for multimedia services," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1147–1169, Dec. 1999.

[14] "Special issue on MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, Feb. 1997.

[15] *MPEG-4 Video Verification Model Ver. 7.0*, ISO/IEC JTC1/SC29/WG11, 1997.

[16] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression—a survey," *Proc. IEEE*, vol. 83, pp. 220–245, Feb. 1995.

[17] K. K. Chan and C.-Y. Tsui, "Exploring the power consumption of different motion estimation architectures for video compression," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1217–1220, July 1997.

[18] T. Xanthopoulos and A. P. Chandrakasan, "A low-power IDCT macrocell for MPEG-2 MP@ML exploiting data distribution properties for minimal activity," *IEEE J. Solid-State Circuits*, vol. 34, pp. 693–703, May 1999.

[19] U.-V. Koc and K. J. R. Liu, "Discrete-cosine/sine-transform based motion estimation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Austin, TX, Nov. 1994, pp. 771–775.

[20] U. V. Koc and K. J. R. Liu, "DCT-based subpixel motion estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, 1996, pp. 1930–1933.

[21] U.-V. Koc, "Low complexity and high throughput fully DCT-based motion compensated video coders," Ph.D. dissertation, Univ. of Maryland, College Park.

[22] J. Chen and K. J. R. Liu, "A complete pipelined parallel CORDIC architecture for motion estimation," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 653–660, May 1998.

[23] J. Chen, "SONET network adapter design for ATM over SONET or directly over fiber," in *Proc. IEEE Int. Symp. Circuits and Systems*, Sydney, Australia, 2001.

[24] R. Handel, M. N. Huber, and S. Schroder, *ATM Networks: Concepts, Protocols, Applications*. Reading, MA: Addison-Wesley, 1998.

[25] U. Black, *Advanced Internet Technologies*, ser. Advanced communication technologies. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[26] "Assigned Numbers,", IETF RFC 1700, 1994.

[27] J. Carlson, *PPP Design and Debugging*. Reading, MA: Addison-Wesley, 1998.

[28] "Packet Size Distributions," National Lab for Application Network Research, 1998.

[29] "PPP over SONET/SDH,", IETF RFC 2615, 1994.

[30] B. D. J. Manchester, J. Anderson, and S. Drvida, "IP over SONET," *IEEE Commun. Mag.*, vol. 36, pp. 136–142, May 1998.

[31] M. Budagavi, W. R. Heinzelman, J. Webb, and R. Talluri, "Wireless MPEG-4 video communication on DSP chips," *IEEE Signal Processing Mag.*, vol. 17, pp. 36–53, Jan. 2000.

[32] "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video,", ISO/IEC DIS 13 818-2, 1994.

[33] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[34] J. I. Ronda, M. Eckert, F. Jaureguizar, and N. Garcia, "Rate control and bit allocation for MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1243–1258, Dec. 1999.

[35] Z. Chen, S.-M. Tan, R. H. Campbell, and Y. Li, "Real time video and audio in the world wide web," in *Proc. 4th World Wide Web Conf.*, Oct. 1995, pp. 326–329.

[36] V. A. Vaishampayan, "Application of multiple description codes to image and video transmission over lossy networks," in *Proc. 7th Int. Workship Packet Video*, Brisban, Australia, Mar. 1996, pp. 55–60.

[37] Y. Wang, M. T. Orchard, and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," in *Proc. IEEE 1st Workshop Multimedia Signal Processing*, Princeton, NJ, Mar. 1997, pp. 419–424.

[38] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1988.

[39] R. V. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Norwood, MA: Artech House, 1994.

[40] W. C. Jakes, *Microwave Mobile Communications*. Piscataway, NJ: IEEE Press, 1994.

[41] R. Steele, *Mobile Radio Communications*: Pentech, 1992.

[42] T. S. Rappaport, *Wireless Communications: Principle and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[43] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1–6, Jan. 1995.

[44] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 228–235, June 1994.

[45] T. Kaneko and M. Okudaira, "Encoding of arbitrary curves based on the chain code representation," *IEEE Trans. Commun.*, vol. COM-33, pp. 807–814, July 1985.

[46] F. McConnell, D. Bodson, and R. Schaphorst, *FAX: Digital Facsimile Technology and Applications*. Norwood, MA: Artech House, 1992.

[47] G. Langdon and J. Rissanen, "Compression of black-white images with arithmetic coding," *IEEE Trans. Commun.*, vol. COM-6, pp. 158–67, June 1981.

[48] T. Chen, C. Swain, and B. Haskell, "Coding of subregions for content-based scalable video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 256–260, Feb. 1997.

Jie Chen (S'95–M'97–SM'02) received the B.S. degree in electrical engineering and the M.S. degree in physics from Fudan University, Shanghai, China, in 1987 and 1990, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1992 and 1998, respectively.

He is currently a Principle System Engineer of Flarion Technologies, a Bell Lab spin-off start-up company, working on digital audio broadcast and wireless broadband network. He was with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, in research and development of optical network integrated circuits. In addition, he also served as a member of the ITU-T working group for developing the third-generation CDMA2000 standard. His research interests span the broad aspects of computer engineering and microelectronics (with emphasis on high-performance, low-power architecture, CAD and VLSI design) with the applications for wireless communication, networking (with emphasis on computer and optical networking, and network management) and multimedia signal processing (with emphasis on image and video processing). He is the author of book *Design of Digital Video Coding Systems: A Complete Compressed Domain Approach* (New York: Marcel Dekker 2001). He has published over 20 refereed journal and conference papers and holds four U.S. patents.

Dr. Chen has received numerous awards, including the Bell Lab division award for contributions in the optical network device design, the Hughes division award for contributions in multimedia over wireless local-loop system design, and the student paper award of the International Conference on Circuits and Systems'97 (ISCAS'97). He serves as an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA, an Editor for *EURASIP Journal of Applied Signal Processing*, and was a Guest Editor for a Special Issue of Multimedia over IP of IEEE TRANSACTIONS ON MULTIMEDIA, chaired several special sessions in the international conferences, and held tutorial session of "Understanding Emerging and Future Communication Technologies" at ISCAS'2001. He currently serves as the member of both multimedia and video technologies committees for IEEE Circuits and Systems Society.

K. J. Ray Liu (S'86–M'86–SM'93) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., and the Ph.D. degree from the University of California, Los Angeles, both in electrical engineering.

He is Professor of the Electrical and Computer Engineering Department and Institute for Systems Research, University of Maryland, College Park. He was a Visiting Professor at Stanford University, Stanford, CA; IT University, Copenhagen, Denmark; and Hong Kong Polytechnic University, Hong Kong. His research interests span broad aspects of signal processing architectures; multimedia signal processing; wireless communications and networking; information security; and bioinformatics, in which he has published over 230 refereed papers, of which over 70 are in archival journals.

Dr. Liu is the recipient of numerous awards, including the 1994 National Science Foundation Young Investigator, the IEEE Signal Processing Society's 1993 Senior Award (Best Paper Award), the IEEE Vehicular Technology Conference Best Paper Award, Amsterdam, 1999. He also received the George Corcoran Award in 1994 for outstanding contributions to electrical engineering education and the Outstanding Systems Engineering Faculty Award in 1996 in recognition of outstanding contributions in interdisciplinary research, both from the University of Maryland. He is Editor-in-Chief of *EURASIP Journal on Applied Signal Processing*, and has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of Special Issues on Multimedia Signal Processing of PROCEEDINGS OF THE IEEE, a Guest Editor of a Special Issue on Signal Processing for Wireless Communications of IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, a Guest Editor of a Special Issue on Multimedia Communications over Networks of IEEE SIGNAL PROCESSING MAGAZINE, a Guest Editor of a Special Issue on Multimedia over IP of IEEE TRANSACTIONS ON MULTIMEDIA, and an Editor of the *Journal of VLSI Signal Processing Systems*. He currently serves as the Chair of Multimedia Signal Processing Technical Committee of IEEE Signal Processing Society and the Series Editor of the Marcel Dekker series on Signal Processing and Communications.