IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY

A PUBLICATION OF THE IEEE CIRCUITS AND SYSTEMS SOCIETY

WWW.IEEE-CAS.ORG

DECEMBER 2004 VOLUME 14 NUMBER 12 ITCTEM (ISSN 1051-8215)

TRANSACTIONS PAPERS

IEEE CIRCUITS AND SYSTEMS SOCIETY

The Circuits and Systems Society is an association of IEEE members with professional interest in the field of circuits and systems theory. All members of the IEEE are eligible for membership in the Society and will receive this TRANSACTIONS upon payment of the annual Society membership fee of $18.00, plus an annual subscription fee of $28.00. For information on joining, write to the IEEE at the address below. *Member copies of Transactions/Journals are for personal use only.*

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (ISSN 1051-8215) is published monthly by the Institute of Electrical and Electronics Engineers, Inc. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society/Council, or its members. **IEEE Corporate Office:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997. **IEEE Operations Center:** 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. **NJ Telephone:** +1 732 981 0060. **Price/Publication Information:** Individual copies: IEEE Members $20.00 (first copy only), nonmembers $41.00 per copy. (Note: Postage and handling charge not included.) Member and nonmember subscription prices available upon request. Available in microfiche and microfilm.

Periodicals Postage Paid at New York, NY and at additional mailing offices. **Postmaster:** Send address changes to IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331. GST Registration No. 125634188. Printed in U.S.A.

Digital Object Identifier 10.1109/TCSVT.2004.839482

Algorithms for Multiplex Scheduling of Object-Based Audio–Visual Presentations

Hari Kalva, *Member, IEEE,* and Alexandros Eleftheriadis, *Senior Member, IEEE*

***Abstract*—Object-based representation of audio–visual (AV) presentations provides a flexible scheme to create interactive content that lends itself to resource-driven adaptation. The content adaptation needs of mobile devices can be met well with the use of object-based AV presentations. The main distinguishing feature of object-based AV presentations is the scene composition at the user terminal. In this paper, we discuss the problem of scheduling the delivery of object-based AV presentations under resource constraints. We explore the similarities with the problem of job sequencing on a single machine. We present a family of algorithms to determine the schedulability of AV presentations, and for unschedulable presentations, we present algorithms to compute a schedule that minimizes the additionally acquired resources. We present algorithms for computing incremental schedules for applications such as content authoring that require immediate feedback on resource consumption. The algorithms can be used to schedule object-based MPEG-4 presentations. We discuss the algorithms and results by considering a relatively complex MPEG-4 presentation with 16 objects, including audio, video, and images.**

***Index Terms*—Delivery scheduling, MPEG-4, object-based content, scheduling algorithms.**

NOMENCLATURE

N	Set of objects to be scheduled.
N	Number of objects to be scheduled.
n_i	Number of access units (AUs) per object ($1 \leq i \leq N$).
$A_j(k)$	Access unit k of object j.
A	$\equiv \bigcup_{j,k} A_j(k)$. Set of all AUs in the presentation.
$T_j^d(k)$	Decoding time of $A_j(k)$.
$T_j^s(k)$	Send time of $A_j(k)$.
σ	$\equiv \bigcup_{j,k} T_j^s(k)$. Send-time schedule.
C	Transmission channel of capacity C.
$s_j(k)$	Size in bytes of $A_j(k)$.
$d_j(k)$	Duration (channel occupancy) of AU k on the wire; $d_j(k) = C \div s_j(k)$.
T_s	Startup delay.
$T_s^{\max}$	Maximum startup delay.
$T_s^{\min}$	$= \sum_k d_k(0) \ni T_k(0) = 0$. Time to transmit AUs of all objects with DTS/CTS of zero.

Manuscript received April 10, 2000; revised July 27, 2003. This paper was recommended by Associate Editor A. Puri.

H. Kalva is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: hari@cse.fau.edu).

A. Eleftheriadis is with the Department of Electrical Engineering, Columbia University, Columbia, NY 10027 USA (e-mail: eleft@ee.columbia.edu).

Digital Object Identifier 10.1109/TCSVT.2004.837020

I. INTRODUCTION

IMAGE and video encoding has been totally transformed with the advent of new coding and representation techniques [28]. These next-generation coding techniques have made possible encoding and representation of audio–visual (AV) scenes with semantically or structurally meaningful objects. Such a new paradigm of object-based representation of AV scenes and presentations will change the way AV applications are created. MPEG-4 is a standardization activity, under the auspices of the International Standards Organization, specifying tools to enable object-based AV presentations [25]. These include tools to encode individual objects, compose presentations with objects, store these object-based presentations, and access these presentations in a distributed manner over networks. The main distinguishing feature of object-based AV presentations is the scene composition at the user terminal. The objects that are part of a scene are composed and displayed at the user end as opposed to encoding the composed scenes as is done in the case of MPEG-2 systems. Such object-based representation and presentation has several benefits including compression efficiency and the capability to interact with individual objects.

The MPEG-4 Systems specification [1], [12], [13], [26] defines an architecture and tools to create AV scenes from individual objects. The scene description and synchronization tools are at the core of the systems specification. The MPEG-4 scene description, also referred to as binary format for scenes (BIFS), is based on the virtual reality modeling language (VRML) and specifies the spatiotemporal composition of objects in a scene [26]. MPEG-4 also specifies the delivery multimedia integration framework (DMIF), a general application and transport delivery framework [14]. In order to keep the user unaware of underlying transport details, MPEG-4 defined an interface between user level applications and the underlying transport protocol called the DMIF application interface (DAI). The DAI provides the required functionality for realizing multimedia applications with quality-of-service (QoS) support. This architecture allows creation of complex presentations with wide-ranging applications. As the complexity of the content increases, so does the complexity of the servers and user-terminals involved. The servers now have to manage multiple streams (objects) to deliver a single presentation.

The flexibility of MPEG-4 enables complex interactive presentations but makes the content creation process nontrivial. Unlike MPEG-2, the content creation process involves much more than multiplexing the media streams. Determining the schedulability, i.e., whether objects in a presentation can be delivered in realtime, of a presentation is also important during the content creation process to determine if the presentation being designed

can be scheduled for specific channel rates and client buffer capacity. It may not be possible to schedule a presentation with a given set of resources. In order to create a schedulable presentation, some constraints may be relaxed. In the case of scheduling objects, relaxing a constraint may involve increasing the buffer capacity, increasing the channel capacity, not scheduling some object instances, or removing some objects from a presentation.

In this paper, we discuss the problem of scheduling AV objects and present algorithms for optimal scheduling of AV objects. We present new algorithms, based on job sequencing on a single machine proposed by Carlier [3], for scheduling objects in a presentation. The original contributions of this paper include: algorithm to determine the schedulability of AV presentations, algorithm to compute startup delay optimal schedules, algorithm to compute schedules that minimize the required channel capacity, and algorithms to compute incremental schedules. A more detailed discussion including the issues in scheduling interactive presentation can be found in [17]. This paper is organized as follows. The general problem of scheduling AV objects and related earlier work is presented in Section II. The characteristics of startup delay and terminal buffer are discussed in Section III. In Section IV, we present several algorithms to schedule AV presentations. Discussions and results are presented in Section V. We conclude the paper in Section VI.

II. Scheduling AV Objects

Scheduling and multiplexing of AV objects in a presentation is a complex problem. Scheduling of AV objects has been the subject of study in [1], [23], and [27]. In [23], Little and Ghafoor present synchronization of multiobject presentations using Petri-net models to describe timing relations in multimedia presentations. They present network-level and application-level synchronization protocols for multiobject presentations. The problem considered is delivering objects from multiple sources to a single destination. The problem we are considering is the network-independent scheduling of interactive AV objects on the server side. We assume the use of underlying network services for establishing connections for data transport. We also show that scheduling objects jointly results in bandwidth savings. In the Firefly system [2], the issue addressed was scheduling a set of local objects to ensure synchronization by adjusting the duration of the media objects involved. The authors address the issue of meeting specified timing constraints on the presentation by adjusting the playrate of objects (speeding up or slowing down playback) but do not consider network delivery issues. We address the problem of synchronization by assuming a constant delay network, a terminal buffer, and time stamps associated with the objects in the presentations. In [27], Song *et al.* describe the JINSEL system that uses bandwidth profiles (BPs) to reserve bandwidth for media objects on a delivery path. The JINSEL system computes the bandwidth required on the network segments on the delivery path using the amount of buffer available on the switch/component. This approach to delivering object-based presentations is not practical because of the processing required at the intermediary nodes. The approach reserves a constant bit rate (CBR) channel assuming only CBR sources while it has been shown that multiple CBR sources, when combined, result in a variable bit rate (VBR) source [9]. We consider a generic MPEG-4 presentation and present algorithms for determining a delivery schedule.

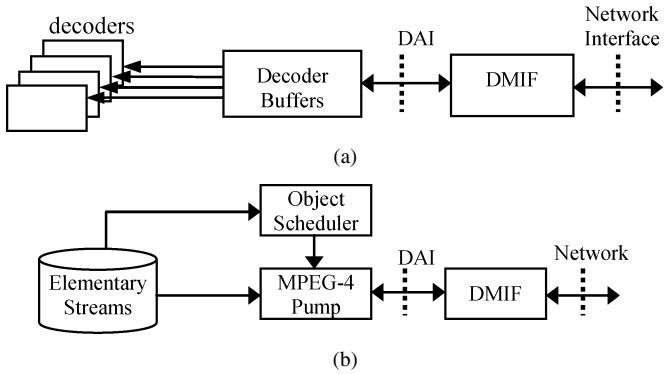

Fig. 1(a). Terminal model. (b) Server model.

In the following, the problem is explained in the context of MPEG-4 Systems. MPEG-4 Systems specifies an architecture to describe scenes and communicate AV data that corresponds to the objects in a scene [1]. A scene consists of one or more AV objects with each of these objects associated with an elementary stream that carries the corresponding data. All the elementary streams are typically multiplexed in a transport multiplex. A server that is transmitting objects (elementary streams) should make sure that an *access unit* (AU) (an AU is the smallest data entity to which timing information can be attributed, e.g., frames in an elementary stream) arrives at the terminal before its decoding time. The constraints on the server transmission are the channel capacity and buffer capacity at the receiving terminal. This problem has similarities with VBR scheduling [24] where the goal is to maximize the number of streams supported by a server. One of the main differences is that in VBR scheduling discussed in [24] and references therein, the assumption is that the video data being handled is periodic (e.g., 30 f/s). In a general architecture such as MPEG-4, such an assumption is not valid as the data may consist of only still images and associated audio. Furthermore, the multiple streams in MPEG-4 presentations are synchronized at the same end-user terminal using a single clock or possibly multiple clocks whereas there are no interdependencies when scheduling multiple VBR video streams. This puts tighter restrictions on the scheduling of an AV presentation. In such cases, the decoding times of individual AUs have to be considered for efficient scheduling. Furthermore, the delay tolerances and relative priorities of objects in an AV presentation can be used to schedule objects for delivery. To make a presentation schedulable, objects of lower priority could be dropped. Even different instances of an object may be assigned different priorities (e.g, higher priority for I and P frames and a lower priority for B frames in an MPEG video stream). These characteristics of the AV services can be used to efficiently schedule a presentation with minimal resource consumption.

A. System Model and Assumptions

We discuss the scheduling of AV objects in the context of a system consisting of client (end-user) server, and network components as shown in Fig. 1(a). Fig. 1(b) shows the server model. The server delivers objects in a presentation as scheduled by

the scheduler. The scheduler uses the decoding timestamps to schedule the delivery of AUs. A decoder is assumed at the far end that decodes the objects for real-time playback. On the client side, data is retrieved from the network and provided to the decoders at decoding time of that AU. Any data that arrives before its decoding time is buffered at the terminal. The terminal buffer model is not considered to keep the schedule independent of terminal designs. However we need the minimum buffer size for a class of terminals to compute object schedules. The data delivered from the server is transported on the channel established between the client and the server. The following assumptions are made about the content, decoders, network, and the server.

Content:
- An AV presentation is composed of one or more objects (AV Objects).
- An AU is the smallest piece of data that can be associated with a decoding time.
- An AV object contains one or more AUs.
- Objects and their AUs may be assigned relative priorities.

Terminal/Decoders:
- The decoders have given, limited memory for receiving and decoder buffers.
- The object data is removed instantaneously from the buffer at the decoding time given by the object's decoding timestamp.
- An object/instance that is received before the decoding time is buffered in the decoder-input buffers until its decoding time.
- More than one object instance may be present in the decoder-input buffers.

Channel/Network:
- End-to-end delays from the server to the player (including the transmission delay) are assumed to be constant.
- The capacity required for the signaling channel is assumed to be negligibly small.
- The transport layer is work conserving, and delivers the packets to the network instantaneously.

Server:
- Audio-visual objects are available at the server in the form of time-stamped AUs.
- All the AUs of an object are delivered in their decoding order.
- A server presents an AU to the transport layer at the send time determined by the scheduler.

B. Problem Formulation

Given a set of N objects that comprise an AV presentation, with each object containing n_i AUs each with a decoding time T_j^d, of kth AU of object j, a transmission channel of capacity C, terminal buffer of size B, allowed startup delay of $T_s^{\max}$, and duration (channel occupancy) of each AU on the channel, $d_j(k)$, is there a schedule σ that satisfies the following constraints:

$$T_j^s(k) \le T_j^d(k) - d_j(k) \tag{1}$$

$$T_j^s(k+1) \ge T_j^s(k) + d_j(k) \tag{2}$$

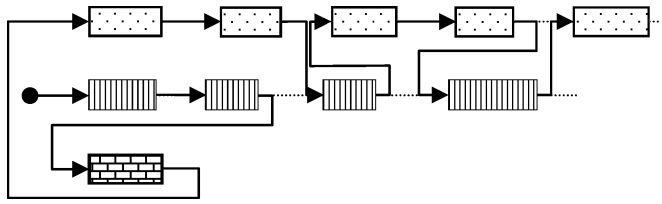

Fig. 2. Sequencing of AUs in a 3-object presentation.

$$\begin{aligned} &\text{if } A_i(k) \in A - A_j(l), \text{ then} \\ &\quad \text{either } T_i^s(k) + d_i(k) \le T_j^s(l) \\ &\qquad \text{or } T_i^s(k) \ge T_j^s(l) + d_j(l) \end{aligned} \tag{3}$$

$$B(t) = \sum_{j,k} C * d_j(k), \quad j \ni T_j^s(k) + d_j(k) \le t; T_j^d(k) \ge t \tag{4}$$

$$T_s \le T_s^{\max}? \tag{5}$$

Constraints (1)–(5) represent the conditions for transmission and playback of object based AV presentations. Constraint (1) enforces the on-time delivery of AUs. Ignoring the constant end-to-end delays, (1) gives the latest time an AU can be transmitted. Constraint (2) imposes intraobject synchronization by enforcing precedence constraints among the AUs. Access units are never transmitted out of order; they are transmitted in their decoding order. Since a single channel is used for transmission, channel occupancy of any two AUs cannot overlap. Constraint (3) ensures that data is delivered on a single channel between a server and a client, i.e., if we consider any two AUs in the presentation, only one of them can be transmitted at any time. Constraint (4) gives the buffer occupancy at the end-user terminal at time t. Constraint (5) gives a bound on the startup delay for the given presentation. If the problem cannot be solved, i.e., a schedule that satisfies the given resource constraints cannot be found, some of the constraints could be relaxed in order to find a schedule. The constraints can be relaxed by reducing the number of objects, increasing the startup delay, or increasing the channel capacity.

Example: Consider the scheduling of a presentation with three objects as shown in Fig. 2. Object 1 has five AUs, object 2 has three AUs, and object 3 has one AU. The AUs are shown with increasing decoding time stamps from left to right. We have to find a schedule, if one exists, that sequences the AUs starting with the first AU of one of the three objects and satisfying the constraints. Fig. 2 shows one such sequence. The general problem of determining the existence of such a sequence is NP-complete. We prove that in *Theorem 1*.

Scheduling is a complex problem and has been widely studied [4]–[6], [8]. Many of the scheduling problems are NP-complete and a number of approximation algorithms are developed trading off optimality for tractability [10], [30]. The scheduling problem closest to the AV object scheduling is job-shop scheduling on a single machine. There has been earlier work on scheduling on single machines. Complexity of machine scheduling problems is studied in [21]. Carlier proved the NP-hardness of one-machine sequencing problem in [3] and some approximation algorithms are discussed in [10]. Another problem with similarities to AV scheduling is job

2004 Index
IEEE Transactions on Circuits and Systems for Video Technology
Vol. 14

This index covers all technical items — papers, correspondence, reviews, etc. — that appeared in this periodical during 2004, and items from previous years that were commented upon or corrected in 2004. Departments and other items may also be covered if they have been judged to have archival value.

The Author Index contains the primary entry for each item, listed under the first author's name. The primary entry includes the coauthors' names, the title of the paper or other item, and its location, specified by the publication abbreviation, year, month, and inclusive pagination. The Subject Index contains entries describing the item under all appropriate subject headings, plus the first author's name, the publication abbreviation, month, and year, and inclusive pages. Subject cross-references are included to assist in finding items of interest. Note that the item title is found only under the primary entry in the Author Index.

AUTHOR INDEX

+ Check author entry for coauthors

+ Check author entry for coauthors

SUBJECT INDEX

3G

A

B

+ Check author entry for coauthors

C

D

E

+ Check author entry for coauthors

S

T

V

+ Check author entry for coauthors

W

+ Check author entry for coauthors

INFORMATION FOR AUTHORS

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY covers all aspects of visual information technology, including but not limited to video A/D and D/A, display technology, image analysis and processing, video signal characterization and representation, image and video compression techniques, signal processing, multidimensional filters and transforms, analog video signal processing, neural networks for video applications, nonlinear video signal processing, video archiving and retrieval, computer vision, video transport, network video, streaming video, high-speed real-time circuits, synthetic video, object- based representation, MPEG technologies, VLSI architecture and implementation for video technology, multiprocessor systems—hardware and software, video systems architecture, video quality assessment, and error resilient video coding. The TRANSACTIONS consists of two sections.

1) *Transactions Papers*—This section is intended for self-contained articles that provide new and significant contributions to the field covered by the TRANSACTIONS, with coherent coverage of theoretical background, technical contributions, practical applications, and experimental results.
2) *Transactions Letters*—This section is intended for short articles that describe new and significant contributions to the field covered by the TRANSACTIONS, to be published in the most expeditious manner. This category includes: i) disclosure of potentially important ideas with convincing and encouraging results; ii) a condensed version of a submitted Transactions Paper; and iii) updates and correspondence on previously published papers.

Submission of a contribution is taken to manifest the fact that the submission has not been submitted, accepted, published, or copyrighted elsewhere. However, manuscripts submitted to or accepted by this Transactions may also be submitted for presentation at the IEEE International Symposium on Circuits and Systems or other conferences sponsored by the IEEE Circuits and Systems Society.

A. Submission of a Transactions Paper

1) Only electronic submissions of Transactions Papers via e-mail will be accepted. To submit a Transactions Paper, send an e-mail with subject "Transactions Paper" to the address: submission_tcsvt@hhi.de with the manuscript contained as an attachment strictly in PDF format, readable using Adobe Acrobat Reader. The body of the e-mail as well as the manuscript itself should contain the following information: Manuscript Type, Transactions Paper, Color illustrations: Yes/No, Author Name(s), Author Institution(s), Author E-mail(s), Paper Title, Abstract, Key Words, and any comments to the Editor-in-Chief.
2) Manuscript should state the significance of the problem in the Introduction.
3) The total length of a manuscript should not exceed ten (10) pages when printed using IEEE two-column format including figures (approximately 30 double-spaced typewritten pages of 8 1/2 × 11). The style files for LaTeX and Word can be downloaded from http://www.ieee.org/organizations/pubs/authors.html. The final submission format must be PDF.
4) Originals for the illustrations should be ready to be submitted immediately upon acceptance of the Transactions Paper.
5) See instructions on the web site for completing the IEEE copyright form for the manuscript.
6) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor-in-Chief in the comment form. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit which have appeared or will appear before other audiences.

B. Submission of a Transactions Letter

1) Only electronic submissions of Transactions Letters via e-mail will be accepted. To submit a Transactions Letter, send an e-mail with subject "Transactions Letter" to the address: submission_tcsvt@hhi.de with the manuscript contained as an attachment strictly in PDF format, readable using Adobe Acrobat Reader. The body of the e-mail as well as the manuscript itself should contain the following information: Manuscript Type, Transactions Letter, Color illustrations: Yes/No, Author Name(s), Author Institution(s), Author E-mail(s), Paper Title, Abstract, Key Words, and any comments to the Editor-in-Chief.
2) Manuscript should state the significance of the problem in the Introduction.
3) The total length of a manuscript should not exceed five (5) pages when printed using IEEE two-column format including figures (approximately 15 double-spaced typewritten pages of 8 1/2 x 11). The style files for LaTeX and Words can be downloaded from http://www.ieee.org/organizations/pubs/authors.html. The final submission format must be PDF.
4) Originals for the illustrations should be ready to be submitted immediately upon acceptance of the Transactions Letter.
5) See instructions on the web site for completing the IEEE copyright form for the manuscript.
6) If the manuscript has been presented, published, or submitted for publication elsewhere, please so inform the Editor-in-Chief in the comment form. Our primary objective is to publish technical material not available elsewhere, but on occasion we publish papers of unusual merit which have appeared or will appear before other audiences.

C. Style for Manuscript

1) The manuscript should be typewritten using double space.
2) Provide an informative abstract of 100 to 250 words at the beginning of the manuscript.
3) Provide a separate double-spaced page listing all footnotes. Acknowledgment of financial support is often placed at the end of the footnote.
4) References should appear in a separate reference section at the end of the paper, with items referred to by numerals in square brackets, e.g., [12]. References should be complete and in the IEEE style. Style for papers: author, first initial followed by last name, title, volume, inclusive page numbers, month, and year. Style for books: author, title, location and publisher, year, chapter or page numbers (if desired). See any recent issues for examples.
5) Provide a separate page listing all figure captions, in proper style for the typesetter, e.g., "Fig. 6. The final wide-band amplifier design."
6) For further information, consult *Information for IEEE Transactions and Journal Authors*, available online at http://www.ieee.org/organizations/pubs/ transactions/information.htm, or by e-mail to trans@ieee.org.

D. Style for Illustrations

1) Originals for illustrations should be sharp, noise-free, and of good contrast. We regret that we cannot provide drafting or art service.
2) Line drawings should be in India ink on drafting cloth, paper, or board. Use only 8 1/2 × 11 size sheets to simplify handling of the manuscript.
3) On graphs, show only the coordinate axes, or at most the major grid lines, to avoid a dense hard-thread result.
4) All lettering should be large enough to permit legible reduction of the figure to column width, perhaps as much as 4:1. Typing on figures is not acceptable.
5) Photographs should be glossy prints, of good contrast and gradation, and any reasonable size.
6) Number each original on the back, or at the bottom of the front, and also indicate the author's name.
7) Note C-5 above. Captions lettered on figures will be blocked out in production, in favor of typeset captions.
8) Electronic art, submited by disk, email, or CD, must be in tif, ps, or eps format. (This includes author photos.)

E. Page and Color Photo Charges

After a manuscript has been accepted for publication, the author's institution will be approached with a request to pay a voluntary charge of $110 per page to cover part of the cost of publication. If the voluntary page charge is not honored, IEEE will charge a mandatory page charge of $175 per page for each page in excess of ten (10) for a TRANSACTIONS PAPER or in excess of five (5) for a TRANSACTIONS LETTER. The author will receive 100 free reprints (without cover) only if the voluntary page charge is honored. Detailed instructions will accompany the page proofs. Color photos judged to be essential to show the results of a paper may be printed in color. The cost of printing color photos will be shared by the author(s) and the Society.

F. Copyright

It is the policy of the IEEE to own the copyright to the technical contributions it publishes on behalf of the interests of the IEEE, its authors, and their employers and to facilitate the appropriate reuse of this material by others. To comply with the U.S. Copyright Law, authors are required to sign an IEEE Copyright Form before publication. This form returns to authors and their employers full rights to reuse their material for their own purposes. Authors must submit a signed copy of this form with their manuscript.

G. Special Issues

Special issues of this TRANSACTIONS may be proposed to the Editor-in-Chief by any individual when a new, unique, exciting, and/or timely subject area is identified.

Digital Object Identifier 10.1109/TCSVT.2004.839484

IEEE CIRCUITS AND SYSTEMS SOCIETY

Officers

M. N. S. SWAMY, *President*
Dept. Elect. Comput. Eng.
Concordia Univ.
Montreal, QC H3G 1M8, Canada

G. GIELEN, *President-Elect*
Dept. Elektrotech. Esat-Micas.
Katholieke Univ. Leuven
Leuven, Belgium

G. DE MICHELI, *Past President*
Gates Computer Science
Stanford Univ.
Stanford, CA 94305

R. J. MARKS II, *VP—Administrative*
Rogers Eng. & Comput. Sci. Bldg.
Baylor Univ.
Waco, TX 76798

P. (C.-Y.) WU, *VP—Conferences*
Dept. Electron. Eng.
Nat. Chiao Tung Univ.
Hsinchu 30050, Taiwan, R.O.C.

P. PIRSCH, *VP—Publications*
Inst. for Microelectronic Syst.
Univ. Hannover
D-30167 Hannover, Germany

M. HASLER, *VP—Technical Activities*
Lab. of Nonlinear Syst.
Dept. Communication Syst.
Swiss Federal Inst. Technol.
CH-1015 Lausanne, Switzerland

E. J. YOFFA, *VP—Regions 1–7*
IBM Corp.
T. J. Watson Res. Ctr.
Yorktown Heights, NY 10598

M. J. OGORZALEK, *VP—Region 8*
Dept. Elect. Eng.
Univ. Mining & Metallurgy
30-059 Krakow, Poland

P. M. JULIAN, *VP—Region 9*
DIEC
Univ. Nacional Del Sur
Buenos Aires, Argentina

H. YASUURA, *VP—Region 10*
Dept. Comput. Sci. & Commun. Eng.
Kyushu Univ.
Fukuoka 816-8580, Japan

Board of Governors

Term Ends December 31, 2004
J. CONG
M. FLOMENHOFT
E. G. FRIEDMAN
E. MACII
R. W. NEWCOMB

Term Ends December 31, 2005
A. E. DUNLOP
R. ETIENNE-CUMMINGS
F. MALOBERTI
A. NISHIHARA
H. SCHMID

Term Ends December 31, 2006
G. de VEIRMAN
W.-C. FANG
T. ROSKA
R. C. SMITH
L. TRAJKOVIC

Representatives

IEEE Solid-State Circuits Society
G. GIELEN
Dept. Electrotechniek.
Katholieke Univ. Leuven
Leuven, Belgium

IEEE Neural Networks Society
C. Y. WU
Dept. Electron. Eng.
National Chiao Tung Univ.
Hsinchu 30050, Taiwan, R.O.C.

IEEE Sensors Council
M. E. ZAGHLOUL
Dept. Elect. Comput. Eng.
George Washington Univ.
Washington, DC 20052

IEEE Press
G. R. CHEN
Dept. Electron. Eng.
City Univ. of Hong Kong
Hong Kong

IEEE Nanotechnology Council
B. J. SHEU
Nassda Corp.
2650 San Tomas Expwy.
Santa Clara, CA 95054

H. C. REDDY
Dept. Elect. Eng.
California State Univ.
Long Beach, CA 90840

Society on Social Implications of Technology
R. W. NEWCOMB
Dept. Elect. Comput. Eng.
Univ. Maryland
College Park, MD 20742

Standing Committees

Awards
G. DE MICHELI
Gates Computer Sci.
Stanford Univ.
Stanford, CA 94305

Constitution & Bylaws
B. J. SHEU
Nassda Corp.
2650 San Tomas Expwy.
Santa Clara, CA 95054

Fellows
H. C. REDDY
Dept. Elect. Eng.
California State Univ.
Long Beach, CA 90840

Nominations
J. NOSSEK
Inst. for Circuit Theory
& Signal Processing
Munich Univ. Technology
D-80290, Munich, Germany

Dist. Lecturer Program
M. PEDRAM
Dept. Elect. Eng.-Systems
Univ. Southern California
Los Angeles, CA 90089

Technical Committees

Analog Signal Processing
W. SERDIJN
Electronics Research Lab.
Delft Univ. of Technol.
Delft, The Netherlands

Biomedical Circuits & Systems
T. LANDE
Dept. of Informatics
Univ. of Oslo
Oslo N-0316, Norway

Blind Signal Processing
R. W. LIU
Dept. Elect. Eng.
Univ. Notre Dame
Notre Dame, IN 46556

Cellular Neural Networks & Array Computing
B. SHI
Dept. Elect. & Electron. Eng.
Hong Kong Univ. of Sci. & Technol.
Kowloon, Hong Kong

Circuits & Systems for Communications
M. A. BAYOUMI
Ctr. Advanced Comput. Studies
Univ. Louisiana
Lafayette, LA 70504

Computer-Aided Network Design
M. SANTORO
Sunnyvale, CA 94087

Digital Signal Processing
T. SARAMAKI
Institute of
Signal Processing
Tampere Univ. Technol.
FIN-33101 Tampere
Finland

Graph Theory & Computing
K. THULASIRAMAN
School of Comput. Sci.
Univ. Oklahoma
Norman, OK 73019

Life-Science Syst. & Applications
C. LI
IBM T. J. Watson Res. Ctr.
Yorktown Heights, NY 10598

Multimedia Syst. & Appl.
L.-G. CHEN
Dept. Elect. Eng.
National Taiwan Univ.
Taipei 106, Taiwan, R.O.C.

Nanoelectronics & Gigascale Syst.
A. ANDREOU
Dept. Elect. & Comput. Eng.
Johns Hopkins Univ.
Baltimore, MD 21218

Neural Syst. & Appl.
H. KWAN
Dept. Elect. & Comput. Eng.
Univ. of Windsor
Windsor, ON, Canada

Nonlinear Circuits & Syst.
G. MAGGIO
Ctr. for Wireless
Communications
Univ. of California, San Diego
La Jalla, CA 92093

Power Syst. & Power Electron. Circuits
M. K. KAZIMIERCZUK
Dept. Elect. Eng.
Wright State Univ.
Dayton, OH 45435

Sensory Systems
O. YADID-PECHT
School. of Elect. Comput. Eng.
Ben-Gurion Univ.
Beer Sheva 84105
Israel

Visual Signal Proc. & Commun.
H. SUN
Mitsubishi Elect.
Research Labs.
558 Central Ave.
Murray Hill, NJ 07947

VLSI Syst. & Appl.
K. K. PARHI
Dept. Elect. Eng.
Univ. Minnesota
Minneapolis, MN 55455

IEEE CAD Trans.
K. MAYARAM, *Editor*
Dept. Elect. Comput. Eng.
Oregon State Univ.
Corvallis, OR 97331

IEEE CAS I Trans.
K. K. PARHI, *Editor*
Dept. Elect. Comput. Eng.
Univ. Minnesota
Minneapolis, MN 55455

IEEE CAS II Trans.
S. BASU, *Editor*
Comput. Commun. Res. Div.
National Science Foundation
Arlington, VA 22230

IEEE CSVT Trans.
T. SIKORA, *Editor*
Inst. for Telecommunications
Technical Univ. Berlin
10587 Berlin, Germany

IEEE Multimedia Trans.
T. CHEN, *Editor*
ECE Dept.
Carnegie Mellon Univ.
Pittsburgh, PA 15213

IEEE VLSI Syst. Trans.
H. RANGANATHAN, *Editor*
Dept. Comput. Sci. & Eng.
Univ. South Florida
Tampa, FL 33620

IEEE CAS Magazine
M. J. OGORZALEK, *Editor*
Dept. Elect. Eng.
Univ. Mining & Metallurgy
30-059 Krakow, Poland

IEEE Circuits and Devices Mag.
C. Y. WU, *Editor for CAS Society*
Dept. Electron. Eng.
National Chiao Tung Univ.
Hsinchu 30050, Taiwan, R.O.C.

APCCAS Gen. Chair
B. LIU
Dept. Elect. Eng.
Nat. Cheng Kung Univ.
Tainan 70101, Taiwan
R.O.C.

DAC Gen. Chair
S. MALIK
Princeton Univ.
Princeton, NJ

ICCAD Gen. Chair
H. ONODERA
Graduate Sch. Inform.
Kyoto Univ.
Kyoto, Japan

ICCD Gen. Chair
K. SHEPARD
Dept. Elect. Eng.
Columbia Univ.
New York, NY 10027

ICECS Gen. Chair
O. YADID-PECHT
Dept. Electro-Optics
Ben-Gurion Univ.
of the Negev
84105 Beer Sheva,
Israel

ISCAS Gen. Chair
A. ANTONIOU
Univ. Victoria
Victoria
BC V8P 5C2, Canada

MWSCAS Gen. Chair
T. HINAMOTO
Graduate Sch. of Eng.
Hiroshima Univ.
Hiroshima, Japan

Digital Object Identifier 10.1109/TCSVT.2004.839483

Optimal Watermark Detection Under Quantization in the Transform Domain

Alexia Briassouli and Michael G. Strintzis, *Fellow, IEEE*

***Abstract*—The widespread use of digital multimedia data has increased the need for effective means of copyright protection. Watermarking has attracted much attention, as it allows the embedding of a signature in a digital document in an imperceptible manner. In practice, watermarking is subject to various attacks, intentional or unintentional, which degrade the embedded information, rendering it more difficult to detect. A very common, but not malicious, attack is quantization, which is unavoidable for the compression and transmission of digital data. The effect of quantization attacks on the nearly optimal Cauchy watermark detector is examined in this paper. The quantization effects on this detection scheme are examined theoretically, by treating the watermark as a dither signal. The theory of dithered quantizers is used in order to correctly analyze the effect of quantization attacks on the Cauchy watermark detector. The theoretical results are verified by experiments that demonstrate the quantization effects on the detection and error probabilities of the Cauchy detection scheme.**

***Index Terms*—Cauchy detector, dither theory, quantization attacks, watermarking.**

I. Introduction

THE advances in multimedia processing allow text, audio, image, and video data to be represented, stored and distributed in digital format. However, the facility with which digital information can be reproduced at a high quality has led to numerous violations of copyright laws and intellectual property rights. This issue has attracted great attention, leading to the development of various copyright protection systems. A method that has gained considerable popularity is the watermarking of data. Watermarking is essentially the embedding of information directly into the data to be protected, which in turn serves as a host or a cover for that information. There exists both visible and invisible watermarks, depending on the intended use of the digital data. The former are usually used in preview images in the World Wide Web or in image data bases to prevent their commercial use. On the other hand, digital copyright systems employ invisible watermarks [1], which are the concern of the present paper.

The watermark is a digital signal, usually containing information about the data origin, destination, and owner, or additional information concerning transaction dates, serial numbers, etc., which can be useful when tracking illegal use of digital data [2], [3]. This information is contained in a secret key, which is known by the legal owner of the watermarked data. Watermarking is different from other copyright protection methods, like steganography, as it needs to be robust against malicious attacks by unauthorized users, who try to remove or render the digital signature undetectable. To ensure maximum robustness, the watermark should be resilient against attacks even if the attacker knows the exact embedding process, but not the cryptographically secure key. In cryptography, this is known as *Kerkhoff's Law* [3].

Digital data is subject to nonmalicious attacks, such as geometric transformations, analog-to-digital conversion, compression, and quantization [4]. In addition to these benign attacks, which are simply caused by the processing and transmission of the data, many attacks are intentional, and aim at the removal of the watermark without destroying the host data, to achieve its illegal use. In this paper, we focus on quantization attacks, which are unavoidable, since quantization is essential for the compression and transmission of digital data [4], [5]. Quantization partially destroys the watermark, making it more difficult to detect. Thus, it is of particular interest to study the effects of quantization on watermarked data and examine the limits of performance of watermark detection schemes under quantization.

The effects of quantization attacks on watermarking schemes have also been examined in [6] and [7], where the analysis of the detection robustness was based on the traditionally used correlator. This detector is widely used, although its performance is optimal only in the case of Gaussian data [8], [9]. In [6], the data used are discrete cosine transform (DCT) coefficients of still images, which do not follow a Gaussian but a much more heavy tailed distribution [10]. In our analysis, we will also consider still image or video DCT coefficients, since watermarks are often embedded in this domain. These coefficients have been modeled more accurately with the generalized Gaussian distribution by Hernandez [9], who also designed an improved detector based on this model. However, the most important image (or video) DCT coefficients can be even more accurately modeled by the symmetric alpha-stable ($S\alpha S$) distribution [11], [12]. Thus, we have preferred to focus our analysis on a detector that is based on this statistical model [13].

It must be noted that the Cauchy detector is used and not a general $S\alpha S$ detector, as the Cauchy and Gaussian distributions

Manuscript received March 10, 2001; revised March 2, 2004. This work was supported in part by the European Program ASPIS: An Authentication and Protection Innovative Software System for DVDROM and Internet, Information Society Technologies (IST). This paper was recommended by Associate Editor R. L. Stevenson.

A. Briassouli was with the Informatics and Telematics Institute (ITI/CERTH), Thessaloniki, Greece. She is now with the Beckman Institute, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: briassou@vision.ai.uiuc.edu).

M. G. Strintzis is with the Informatics and Telematics Institute (ITI/CERTH), Thessaloniki, Greece, and also with the Department of Electrical and Computer Engineering, University of Thessaloniki, Thessaloniki, Greece (e-mail: strintzi@eng.auth.gr).

Digital Object Identifier 10.1109/TCSVT.2004.836753

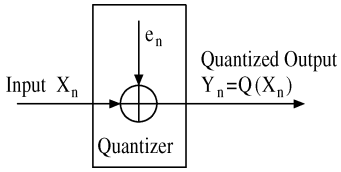

Fig. 1. Additive noise model of a quantizer.

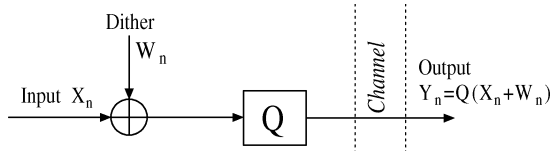

Fig. 2. NSD quantizer.

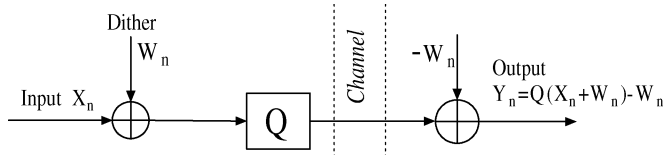

Fig. 3. SD quantizer.

are the only members of the $S\alpha S$ family with a closed from expression. Although the Cauchy detector is not fully optimal for general $S\alpha S$ data, it has proven to be particularly robust, even when the given data deviates from the strictly Cauchy model. In particular, in [14] and [15], it is shown that this detector continues to perform well even with non-Cauchy data, whereas the performance of the Gaussian detector deteriorates visibly in a non-Gaussian environment. As a consequence, the Cauchy watermark detector is expected to remain nearly optimal even for general $S\alpha S$ data. The robustness of this detector and the accuracy of the $S\alpha S$ data modeling motivated us to examine the performance of Cauchy watermark detection under quantization attacks.

The embedding of an additive watermark in the data followed by quantization can be considered as a case of *dithered quantization* [16]. Dithered quantization is a technique in which a low-power random or pseudorandom signal is purposefully added to a host signal prior to quantization. This intentional distortion leads to a subjectively more pleasing reproduction of the host signal after quantization and, in some cases, can actually cause the quantization error to behave in a statistically desirable way. A watermark and a dither signal do not serve the same purpose, but their effect on the host data is the same. Consequently, known results for dithered quantization can be used to study watermarking, and especially the robustness of watermark detection, under quantization.

This paper is organized as follows. Section II presents the theory of dithered quantizers, which is used in the analysis of the quantization effects on watermarking. In Section III, spread spectrum watermarking in the DCT-domain is described. Section IV presents the method of statistical watermark detection in the transform domain. In particular, the nearly optimal Cauchy detector is presented and its performance under quantization is analyzed. Section V contains experimental results and, finally, conclusions are drawn in Section VI.

II. Dithered Quantization

A very basic step in lossy digital data compression and analog-to-digital conversion of data is quantization. The simplest type of quantizer (Fig. 1) is the uniform quantizer, which maps the input X_n to a collection of equally spaced output levels, giving the quantized output $Q(X_n)$. Despite the simplicity of the uniform quantizer, it has proven difficult to analyze theoretically, because of its inherent nonlinearity [17]. A common misconception is that the quantization error $e_n = Q(X_n) - X_n$ consists of a sequence of independent identically distributed (i.i.d.) random variables, uncorrelated with each other and with the input signal. Actually, this approximation, known as the Bennett approximation [18], approaches accuracy only when the the number of quantizer levels is very large, or the quantization step size Δ is very small. In many modern applications, this condition is definitely not met. A way to force the Bennett conditions and, thus, temper the statistical properties of the error signal, is to use a dither signal, which is added to the host signal before quantization to "break up" the pattern of the quantization noise and render the reconstructed signal perceptually more pleasing. In this way, the output signal becomes more smooth, leading to a smoother quantization error. More importantly, the resulting quantization error is independent of the input signal and consequently becomes less audible or visible [16]. The addition of a watermark to digital multimedia data before quantization is very similar to this procedure [7], so the quantization of watermarked data can indeed be considered as a case of dithered quantization.

Dithered quantizing systems are shown in Figs. 2 and 3. In both schemes, the system input is the original signal X_n while the quantizer input is $X_n + W_n$. The dither signal W_n is considered to be a strict-sense stationary random process [17], independent of X_n. Fig. 2 represents a nonsubtractively dithered (NSD) system, while a subtractively dithered (SD) system is presented in Fig. 3, in which the dither signal is subtracted from the output of the dithered quantizer. Naturally, in the second case it is assumed that the dither signal must be available at both ends of the channel (Fig. 3). The *total error* ε_n of these schemes is defined as the difference between the original system input and the final output, while the *quantization error* e_n is the difference between the quantizer input and output. These errors are not the same for subtractively and NSDs. In particular, the *total error* of a SD, also referred to as *quantization noise* in [16], is the difference of the original input and the final output

$$\varepsilon_n^{\mathrm{SD}} = Q(X_n + W_n) - (X_n + W_n). \tag{1}$$

In this case, the total error of the SD system happens to be equal to the *quantization error* of a NSD, a very interesting and useful result, as we will show later on. The total error for a NSD is

$$\varepsilon_n^{\mathrm{NSD}} = Q(X_n + W_n) - X_n = \varepsilon_n^{\mathrm{SD}} + W_n \tag{2}$$

which is obviously different from the total error of an SD system.

The purpose of the dither signal is to render the quantization error uniform and independent of the input signal, which can be achieved if certain conditions, developed by Schuchman [19],

are met. Specifically, Schuchman's conditions require that the quantizer does not overload, meaning that the quantization error never exceeds $\Delta/2$, and the characteristic function of the dither signal, defined as

$$M_W(ju) = E(e^{juW}) \tag{3}$$

has the following property:

$$M_W\left(\frac{j2\pi l}{\Delta}\right) = 0, \quad l \neq 0. \tag{4}$$

If these conditions are met, Schuchman proved that the total error $\varepsilon_n^{\mathrm{SD}}$ of a SD quantizing system is uniformly distributed on $(-\Delta/2, \Delta/2]$, and statistically independent of the original input signal X_n.

Gray and Stockham [16] derived the following necessary and sufficient condition for the kth moment of the nonsubtractive quantization error e_n^{NSD} to be independent of the input signal x:

$$\frac{d^k}{du^k}\left(M_W(ju)M_v(ju)\right)\Big|_{u=2\pi b/\Delta} = 0, \quad b \in \mathbf{Z},\ b \neq 0 \tag{5}$$

where v is a random variable uniformly distributed on $(-\Delta/2, \Delta/2]$, and independent of x. This random variable has no special meaning, but it is necessary to formulate condition (5). As mentioned in [6], these properties of the quantization and the total errors are valid for uniform scalar quantizers where overload does not occur, which is the case examined here.

These results are particularly important for the analysis in Section IV, where the effect of the quantization errors on the watermark detector are examined. The quantization error of a watermarking system under quantization is identical to the quantization error of a NSD quantizer, since the dither signal, which is the watermark, is not subtracted from the system output. Thus, the quantization error of the watermarked signal is

$$e_n^{\mathrm{NSD}} = Q(X_n + W_n) - (X_n + W_n). \tag{6}$$

However, as (6) shows, the *quantization error* of an NSD system is identical to the *total error* of an SD quantizer given in (1). Consequently, Schuchman's Conditions hold for the watermark quantization error that interests us (the quantization error of a NSD quantizer), since it is equal to the total error of an SD system. Therefore, the watermarking quantization error will be validly considered in the sequel as a sequence of i.i.d. uniformly distributed random variables in $(-\Delta/2, \Delta/2]$, independent of the input signal and white.

III. DCT-Domain Watermarking

In this section, we briefly describe the watermark generation and embedding process used for image or video data in the DCT domain [20]. Most watermarking systems are inspired by the spread spectrum modulation schemes used in digital communications in jamming environments [21]–[23]. The role of the jammer in the watermarking problem is assumed by the host signal, which is equivalent to additive noise, and by the attacker who unintentionally or intentionally tries to destroy or extract the embedded watermark [2], while the watermark is the hidden information signal. Spread spectrum techniques are so popular for watermarking because they exhibit low probability of intercept (LPI), which in the case of watermarking translates to high imperceptibility [8]. Increased robustness of the watermarking system is ensured by the anti-jamming properties of spread spectrum systems. The pseudorandom modulation of the watermark, described analytically in Section III-A, increases the achieved security, since knowledge of this sequence is required for successful watermark retrieval [2].

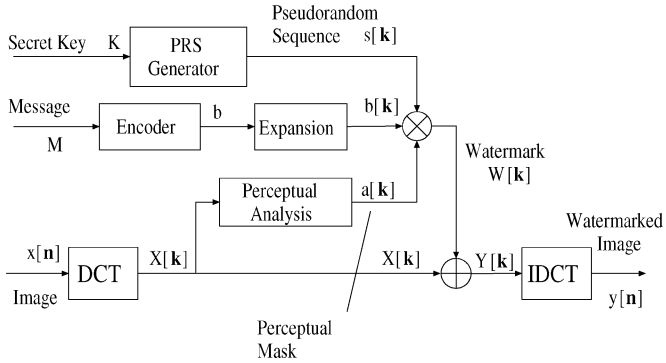

Fig. 4. Watermark embedding process.

A. Watermark Generation and Embedding

If an image is represented as a discrete two-dimensional (2-D) sequence $x[\mathbf{n}]$ of its luminance component with $N_1 \times N_2$ pixels, the watermark can be considered as a 2-D DCT signal $W[\mathbf{k}]$, which is added to the DCT coefficients. Indexes in boldface typesetting represent the corresponding 2-D indexes, so $\mathbf{k} = [k_1, k_2]$ and $\mathbf{n} = [n_1, n_2]$. The DCT can either be applied to the entire image or in blocks. We use the blockwise DCT because it is used much more widely in practice and because the blockwise application of the DCT is faster than the full-image DCT. In our case, it also happens to allow the statistical characterization of each DCT coefficient.

The watermark may or may not contain a message M, which is mapped by an encoder to an N-dimensional codeword vector $\mathbf{b}$ (Fig. 4). In the frequently considered case where the watermark does not contain a message the codeword vector, $\mathbf{b}$ reduces to $b = 1$ [9]. Every element of the codeword vector b_i is then repeated in a different set S_i of image or video pixels in the DCT domain, in an *expansion process*, which spreads it over all the pixels. It must be noted that this repetition of the information bits b_i introduces a certain degree of redundancy in the watermark, which is desirable, as it increases its robustness. In the case of watermark detection, there is no hidden message, so the element repeated over all the pixels is simply $b = 1$.

A direct spread spectrum modulated watermark is then created from the signal resulting from the above expansion process, by multiplying it in a pixel-by-pixel manner with an appropriate 2-D pseudorandom sequence $s[\mathbf{k}]$. The security of the entire system is ensured by the initializing seed to the pseudorandom noise generator: this key is cryptographically secure and is known only by the legal owner of the watermarked document. Without knowledge of the secret key, the generation of the watermark at the receiver is impossible.

The *spreading sequence* $s[\mathbf{k}]$ must have noise-like properties, in order to spread the spectrum of the input signal $b[\mathbf{k}]$,

so its mean should be approximately zero and its auto-correlation should approach the delta function. For the system considered here, the pseudorandom sequence is bipolar, i.e. it takes the values $\{+1, -1\}$, with relative frequencies 1/2 each. The multiplication of the signal with this sequence spreads its power over a wide bandwidth, increasing its invisibility and robustness.

In order to guarantee the invisibility of the alterations introduced by the watermark, the spread signal is multiplied by a perceptual mask $a[\mathbf{k}]$, that takes into account the properties of the human visual system [24], [25]. In order to design it, one must estimate the visibility threshold $T(i,j)$ for every (i,j) DCT coefficient of each 8×8 block, which is approximated in logarithmic units by the following function

$$\log T(i,j) = \log\left(\frac{T_{\min}\left(f_{i,0}^2 + f_{0,j}^2\right)^2}{\left(f_{i,0}^2 + f_{0,j}^2\right)^2 - 4(1-r)f_{i,0}^2 f_{0,j}^2}\right) + K\left(\log\sqrt{f_{i,0}^2 + f_{0,j}^2} - \log f_{\min}\right)^2 \quad (7)$$

where $f_{i,0}$ and $f_{0,j}$ are the vertical and horizontal spatial frequencies, respectively, (in cycles/degree) of the DCT basis functions and $T_{\min}$ is the minimum value of the quadratic function $T(i,j)$ associated with $f_{\min}$. It must be noted that this model is valid for the ac frequencies only and not for the dc coefficient. The threshold $T(i,j)$ is corrected for every block through the following equation:

$$T'(i,j) = T(i,j)\left(\frac{X_{0,0}}{\overline{X}_{0,0}}\right)^{\alpha_T} \quad (8)$$

where $X_{0,0}$ is the dc coefficient for each block and $\overline{X}_{0,0}$ is the average luminance of the screen. These two functions contain parameters which are set to $r = 0.7$, $T_{\min} = 1.1548$, $K = 1.728$, $f_{\min} = 3.68$ cycles/degrees and $\alpha_T = 0.649$, following [25] and the perceptual mask is obtained from the visibility threshold according to the following equation:

$$\alpha[k_1,k_2] = 4 \cdot \left(1 + (\sqrt{2}-1)\delta(l_1)\right) \cdot \left(1 + (\sqrt{2}-1)\delta(l_2)\right) \cdot \gamma \cdot T'(l_1,l_2) \quad (9)$$

where $l_1 = k_1 \bmod 8$, $l_2 = k_2 \bmod 8$, $\delta(\cdot)$ is the Kronecker function and $\gamma < 1$ is a scaling factor. This mask ensures that the watermark will remain imperceptible to the human eye, but at the same time it will alter the pixel values as much as possible in order to achieve maximum robustness.

The resulting watermark, for the case where $b = 1$, is $W[\mathbf{k}] = a[\mathbf{k}]s[\mathbf{k}]$ and is added to the original DCT coefficients $X[\mathbf{k}]$, giving the watermarked signal $Y[\mathbf{k}] = X[\mathbf{k}] + W[\mathbf{k}]$ (Fig. 4). The only way for the attacker to retrieve the hidden signal is to have a copy of the pseudorandom sequence, in other words its seed and the procedure through which it is generated. This way, the attacker cannot extract the watermark without knowledge of the secret key (the pseudorandom sequence seed), even if the entire watermark generation and embedding process is known. Thus, the dependence of the generated watermark on the secret cryptographic key and the spread spectrum embedding procedure result in a robust watermark, which is resilient against standard signal processing and manipulation as well as intentional attacks.

IV. Watermark Detection After Quantization

A. Statistical Watermark Detection

Traditionally, detectors correlate the given data with the known watermark. Alternatively, the detection problem can be formulated as a binary hypothesis test [26]. Although the given data for this test possibly contains a watermark, the actual test is performed without knowledge of the original, unwatermarked data. Thus, this technique actually is a *blind* watermark detection method. The analysis that follows can be applied to any kind of data, but we will focus on image and video DCT coefficients, in which watermarks are often embedded. The two hypotheses for the test are formulated as follows:

$$\begin{aligned} H_1: &\quad Y[\mathbf{k}] = X[\mathbf{k}] + W[\mathbf{k}] \\ H_0: &\quad Y[\mathbf{k}] = X[\mathbf{k}]. \end{aligned} \quad (10)$$

In this detection problem, the watermark $W[\mathbf{k}]$ is the desired signal or information, while the DCT image or video coefficients $X[\mathbf{k}]$, i.e. host signal, play the role of unknown additive noise with known statistical properties. The goal of the watermark detector is to verify whether or not a watermark exists in the given data, based on the statistical properties of the (possibly) watermarked signal that is received, and the known watermark. A basic assumption is that the statistical distribution of the DCT coefficients is not significantly altered after the insertion of a watermark. This assumption is intuitively justified by the fact that the watermarked signal should follow a statistical distribution similar to that of the original signal, if the embedded signal is imperceptible.

We assume that the data has been modeled by an appropriate statistical distribution, so the probability density functions (pdfs) of the host signal and its watermark are known or can be very accurately approximated from the given data. In that case, the detection rule is

$$\Lambda(Y) \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \quad (11)$$

where the likelihood ratio $\Lambda(Y)$ is defined as

$$\Lambda(Y) = \frac{f(Y|H_1)}{f(Y|H_0)}. \quad (12)$$

In practice, the log-likelihood ratio is usually preferred to perform hypothesis testing. The log-likelihood ratio is simply defined as the natural logarithm of the likelihood ratio, $l(Y) = \ln(\Lambda(Y))$, given by

$$l(Y) = \ln\left(\frac{f(Y|H_1)}{f(Y|H_0)}\right). \quad (13)$$

Neyman–Pearson testing is performed, where $l(Y)$ is compared to a threshold η, which is determined by the probability of false-alarm P_{fa}, i.e. the probability of detecting a watermark in unwatermarked data (Fig.5). The exact relation of this threshold to the original data can be formulated analytically and is presented in the following section.

B. Cauchy Detector

The hypothesis test described above depends on the statistical model used for the data, so it is necessary to model the data

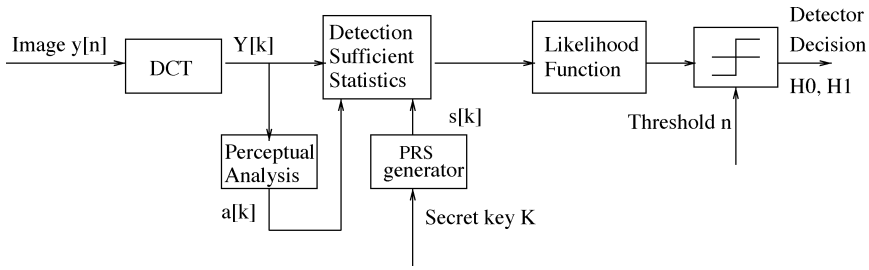

Fig. 5. Watermark detection process.

by a suitable pdf to be used in the design of the detector. The parameters of this model can be estimated with accuracy from the given data [27], as it will become clear later on.

It is well known that DCT image or video coefficients, are not accurately modeled by the Gaussian distribution, since they display heavy tails [28], [29]. The use of more appropriate models, such as the generalized Gaussian distribution [30] has been suggested in the literature. However, other heavy tailed distributions, like the members of the alpha-stable family, have been found to be more suitable. The $S\alpha S$ family is often used to describe non-Gaussian signals, which are characterized by an impulsive nature [11]. Because of their impulsiveness, these signals exhibit distributions with heavy tails, like the low-frequency DCT coefficients.

The stable distribution can be best described by its characteristic function

$$\varphi(\omega) = \exp\left(j\delta\omega - \gamma|\omega|^{\alpha}\left[1 + j\beta\,\mathrm{sign}(\omega)\phi(\omega,\alpha)\right]\right) \quad (14)$$

where

$$\phi(\omega,\alpha) = \begin{cases} \tan\frac{\alpha\pi}{2}, & \text{for } \alpha \neq 1 \\ \frac{2}{\pi}\log|\omega|, & \text{for } \alpha = 1. \end{cases} \quad (15)$$

The stable distribution is completely described by the following four parameters [31]: the location parameter δ $(-\infty < \delta < \infty)$, the scale parameter γ $(\gamma > 0)$, which is known as the *dispersion*, the index of skewness β $(-1 \leq \beta \leq 1)$, and the *characteristic exponent* α $(0 < \alpha \leq 2)$. We will examine the $(S\alpha S)$ distribution, which has $\beta = 0$. The location parameter δ gives the distribution mean for $1 < \alpha \leq 2$, while for $0 < \alpha \leq 1$ it gives the median. The dispersion γ can be any positive number and corresponds to the spread around the location parameter δ, in other words it behaves like the variance of the Gaussian distribution [32].

The characteristic exponent α determines the shape of the distribution as it measures the "thickness" of the tails of the pdf. For smaller values of α, the tails of the distribution are heavier and the corresponding random process displays high impulsiveness, while increasing values of α correspond to distributions with rapidly decaying tails, that approach the Gaussian case. There are closed-form expressions for the pdf of $S\alpha S$ random variables only for $\alpha = 2$ and $\alpha = 1$, which correspond to the Gaussian and Cauchy distributions, respectively [11]. A Cauchy detector has been designed for the case that $\alpha = 1$ in [13]. Naturally, this detector is not optimal for all cases, since the data is generally $S\alpha S$ and not specifically Cauchy. However, the Cauchy detector has proven to be particularly robust even if the input data has a shape factor that deviates from $\alpha = 1$ [14], [15]. It is certainly better than the traditional correlator, as it correctly assumes a heavy tailed model for the data.

The Cauchy model has the following pdf:

$$f_X(x) = \frac{1}{\pi}\frac{\gamma}{\gamma^2 + (x-\delta)^2} \quad (16)$$

where γ is the data dispersion and δ is the location parameter. The corresponding Cauchy detector can be derived by replacing (16) in (13). In this case, the hypothesis test presented in (10) uses the pdfs

$$\begin{aligned} f\left(Y(\mathbf{k})|H_0\right) &= f_X\left(Y(\mathbf{k})\right) \\ f\left(Y(\mathbf{k})|H_1\right) &= f_X\left(Y(\mathbf{k}) - W(\mathbf{k})\right). \end{aligned} \quad (17)$$

In the sequel, the 2-D index $\mathbf{k}$ will be omitted for notational simplicity. If the set of DCT coefficients that contain a watermark is denoted by S, the likelihood ratio becomes

$$\begin{aligned} \frac{f(Y|H_1)}{f(Y|H_0)} &= \prod_S \frac{f_X(Y-W)}{f_X(Y)} \\ &= \prod_S \frac{\gamma^2 + (Y-\delta)^2}{\gamma^2 + (Y-W-\delta)^2} \end{aligned}$$

and the corresponding log-likelihood ratio is

$$l(Y) = \sum_S \ln\frac{\gamma^2 + (Y-\delta)^2}{\gamma^2 + (Y-W-\delta)^2}. \quad (18)$$

This ratio will be compared against a suitably chosen threshold in order to determine the existence of a watermark. The low- and mid-frequency DCT coefficients are very accurately modeled by $S\alpha S$ distributions which may deviate slightly from the Cauchy model. Despite this, it has been shown [14] that the performance of this detector is not significantly affected by such deviations, so its results remain very accurate.

C. Effect of Quantization on the Cauchy Detector

Quantized data is corrupted by quantization errors e_{H_1} and e_{H_0}, corresponding to watermarked and unwatermarked data, respectively. These errors are defined as the difference between the quantizer input and output

$$e_n = Q(X_n + W_n) - (X_n + W_n) \quad (19)$$

which satisfy Schuchman's conditions, as seen in Section II. Consequently, the watermarking quantization error e_n of (19) will be statistically independent of the input signal, i.i.d., and uniformly distributed on $(-\Delta/2, \Delta/2]$. These properties of the

quantization error are of essential importance for the analysis that follows.

Taking the quantization errors into account, the binary hypothesis test of (10) becomes

$$\begin{aligned} H_1: &\quad Y[\mathbf{k}] = X[\mathbf{k}] + W[\mathbf{k}] + e[\mathbf{k}] \\ H_0: &\quad Y[\mathbf{k}] = X[\mathbf{k}] + e[\mathbf{k}]. \end{aligned} \tag{20}$$

Thus, the corresponding log-likelihood ratio under H_0 is

$$l(Y|H_0) = \sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+e-W-\delta)^2} \tag{21}$$

and under H_1

$$l(Y|H_1) = \sum_S \ln \frac{\gamma^2 + (X+W+e-\delta)^2}{\gamma^2 + (X+e-\delta)^2}. \tag{22}$$

The likelihood functions (21) and (22) are sums of many i.i.d. random variables, so according to the central limit theorem, they can be approximated by a Gaussian variable of mean and variance that can be estimated from the data. This approximation is accurate in practice, since a large number of coefficients are summed in the log-likelihood ratio test.

As already detailed in Section III-A, the watermark is equal to the value of the perceptual mask $a[\mathbf{k}]$ multiplied by the pseudorandom sequence $s[\mathbf{k}]$ ($W[\mathbf{k}] = a[\mathbf{k}]s[\mathbf{k}]$). The pseudorandom sequence used is modeled as an i.i.d. two—dimensional random process with a two-level equiprobable discrete marginal distribution, $s[\mathbf{k}] \in \{-1, 1\}$. The perceptual mask, which defines the watermark amplitude $a[\mathbf{k}]$, is designed according to the properties of the human visual system as described in Section III-A. Thus, the expected value of the watermark at every pixel $\mathbf{k}$ is $a[\mathbf{k}]$ or $-a[\mathbf{k}]$, with probability 1/2 each. It must also be noted that the watermark is supposed to be embedded in a set S of the DCT coefficients, so only these coefficients are taken into account.

The mean of (21) can then be found from

$$E[f(\mathbf{y})] = \int f(\mathbf{y}) p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y} \tag{23}$$

where $p_{\mathbf{y}}(\mathbf{y})$ is the joint probability density function of a multidimensional random variable $\mathbf{y}$. The above should be understood as a short notation for a multivariable integral and $d\mathbf{y}$ as a short notation for $dy_1, \ldots, dy_n$. In our case, there are two random variables, the watermark and the quantization error. The pdf of the latter is $p_e(e) = 1/\Delta$ since the quantization error has been considered to be uniformly distributed in $(-\Delta/2, \Delta/2]$. The watermark is $-a[\mathbf{k}]$ or $a[\mathbf{k}]$ for every pixel $\mathbf{k}$, with probability 1/2 each. Incorporating this in (21) and omitting the index $\mathbf{k}$ gives

$$\begin{aligned} m_0 = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{1}{\Delta} \Bigg[& \frac{1}{2} \sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+e-a-\delta)^2} \\ & + \frac{1}{2} \sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+e+a-\delta)^2} \Bigg] de. \end{aligned} \tag{24}$$

It can be found that

$$\begin{aligned} \int \ln\left[(Y+e)^2 + \gamma^2\right] de = & -2e + 2\gamma \tan^{-1}\left(\frac{Y+e}{\gamma}\right) \\ & + (Y+e) \ln\left[\gamma^2 + (Y+e)^2\right]. \end{aligned} \tag{25}$$

In the interval $(-\Delta/2, \Delta/2]$, the above integral becomes

$$\begin{aligned} Int(Y) = & -2\Delta + 2\gamma \left[\tan^{-1}\left(\frac{Y+\frac{\Delta}{2}}{\gamma}\right) - \tan^{-1}\left(\frac{Y-\frac{\Delta}{2}}{\gamma}\right) \right] \\ & - \left(Y - \frac{\Delta}{2}\right) \ln\left[\gamma^2 + \left(Y - \frac{\Delta}{2}\right)^2\right] \\ & + \left(Y + \frac{\Delta}{2}\right) \ln\left[\gamma^2 + \left(Y + \frac{\Delta}{2}\right)^2\right]. \end{aligned} \tag{26}$$

In view of (26), with Y replaced by $(X+a-\delta)$, $(X-a-\delta)$, or $(X-\delta)$ as appropriate, (24) becomes

$$m_0 = \frac{1}{2\Delta} \sum_S [2Int(X-\delta) - Int(X+a-\delta) - Int(X-a-\delta)]. \tag{27}$$

From this equation, the mean of the log-likelihood ratio can be computed under hypothesis H_0. The variance $\sigma_0^2 = Var[l(Y|H_0)]$ can be similarly computed from

$$\sigma_0^2 = E\left[(l(Y|H_0) - m_0)^2\right] = E\left[l^2(Y|H_0)\right] - m_0^2 \tag{28}$$

where

$$E\left[l^2(X)|H_0\right] = E\left[\sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+e-W-\delta)^2}\right]^2. \tag{29}$$

Considering, as before, that the watermark is a sequence of i.i.d. random variables with the equiprobable values $\{-\alpha[\mathbf{k}], \alpha[\mathbf{k}]\}$ for every pixel $\mathbf{k}$, and also that the quantization error e is a uniformly distributed random variable on $(-\Delta/2, \Delta/2]$, we obtain

$$\begin{aligned} E\left[l^2(X)|H_0\right] = & \frac{1}{2} E_e \left[\sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X-a+e-\delta)^2}\right]^2 \\ & + \frac{1}{2} E_e \left[\sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+a+e-\delta)^2}\right]^2. \end{aligned} \tag{30}$$

Since the quantization error is uniformly distributed on $(-\Delta/2, \Delta/2]$, the following integral will need to be computed:

$$E_e\left[l^2(X)\right] = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \left[\sum_S \ln \frac{\gamma^2 + (Y_1+e)^2}{\gamma^2 + (Y_2+e)^2}\right]^2 de. \tag{31}$$

This integral is computed numerically in the experiments and is left as is in the equations that follow. Under the same assumptions as before for the watermark, the variance is finally given by

$$\begin{aligned} \sigma_0^2 = & \frac{1}{2} E_e \left[\sum_S \ln \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X-a+e-\delta)^2}\right]^2 \\ & + \frac{1}{2} E_e \left[\sum_S \ln^2 \frac{\gamma^2 + (X+e-\delta)^2}{\gamma^2 + (X+a+e-\delta)^2}\right]^2 - m_0^2. \end{aligned} \tag{32}$$

With these quantities known and since $l(Y)$ follows a Gaussian distribution, the probabilities of detection and false alarm for the ratio $l(Y)$ are given by

$$P_{\rm fa}(t) = Q\left(\frac{t-m_0}{\sigma_0}\right) \tag{33}$$

$$P_{\rm det}(t) = Q\left(\frac{t-m_1}{\sigma_1}\right) \tag{34}$$

where t is the threshold against which the data are compared and $Q(x)$ is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt. \tag{35}$$

For a given probability of false-alarm $P_{\rm fa}$, we can compute the required threshold for a watermark to be detected

$$\begin{aligned} P_{\rm fa}(t) &= Q\left(\frac{t-m_0}{\sigma_0}\right) \Rightarrow \\ \frac{t-m_0}{\sigma_0} &= Q^{-1}(P_{\rm fa}) \Rightarrow \\ t &= m_0 + \sigma_0 Q^{-1}(P_{\rm fa}). \end{aligned} \tag{36}$$

By using this known threshold, we can find the relation between $P_{\rm fa}$ and $P_{\rm det}$, which gives the receiver operating characteristic (ROC) curves

$$P_{\rm det} = Q\left(Q^{-1}(P_{\rm fa}) - 2\sqrt{\rm SNR}\right). \tag{37}$$

These curves give a measure of the performance of the proposed detector, both in the absence and in the presence of quantization. The probability of error for the detector can also be defined as the probability of false alarm and the probability of not detecting an existing watermark [7]. Using this error metric, one can measure the performance of the detector under quantization attacks as the probability of error versus the embedded watermark strength or the quantizer step size. Thus, one can examine the effect of quantization of varying strength on the watermarked data for a given probability of false alarm in terms of detection or error probability.

V. Experimental Results

The theoretical analysis of the previous section can be used to examine the influence of quantization on watermarked image or video data. It is possible to measure the performance of the detector in the presence of quantization for watermarks of increasing strength. The robustness of the Cauchy detector in the presence of quantization attacks can be predicted and the minimum watermark strength needed to overcome the quantization error can be estimated. Additionally, in the experiments that will be described, the theoretical results of Section IV-C will be verified. This is very important, as these theoretical results allow the accurate prediction of the performance of watermarking schemes under quantization attacks.

The analysis of the previous sections can be applied to any kind of host and watermark signals, characterized by the heavy-tailed Cauchy or $S\alpha S$ distribution described in Section IV-B. The blockwise DCT transform is applied to the host data in 8×8 blocks. The coefficients corresponding to every frequency are arranged in order of importance using the zig-zag scan and are examined separately. In particular, the DCT coefficients for the same frequency over all the image blocks are grouped in a sample sequence, which can be considered a *subsignal*. This allows the focus of the experimental analysis on the most important coefficients, which are the low- and mid-frequency ones. For our experiments we use low-frequency DCT coefficients of an MPEG-4 video object, and in particular the fifth DCT coefficients of the foreground object from an I-frame of the Akiyo video sequence. These coefficients, or subchannels, follow a heavy-tailed distribution for which the detector presented in Section IV-B is nearly optimal.

The analysis of the quantization attacks on the host data is carried out assuming uniform scalar quantization of each data set. Such quantization is common in practice and is in fact used for the JPEG standard for still images, as well as the MPEG standards for video sequences, which assign different but equal step sizes to every DCT coefficient. The performance of an actual detector for all the DCT coefficients is expected to be very similar to that of the detector for each subchannel, and especially the low and mid frequency ones examined here, since they are the most important ones.

A. Detector Robustness for Varying Quantization Attacks

Initially, the expected mean and variance of the log-likelihood ratio (18) under quantization are computed from (27) and (32). The probability of false alarm is set to $P_{\rm fa} = 10^{-3}$ to reduce the number of free parameters. Using the estimated values of m_0, σ_0^2, and the $P_{\rm fa}$, the detection threshold is computed via (36). Finally, the probability of detection is computed from (34). This is repeated for increasing quantization step sizes and for a particular watermark. The watermark-to-document ratio (WDR) is defined by

$$\text{WDR} = 10\,\log_{10}\left(\frac{\sigma_w^2}{\sigma_x^2}\right) \tag{38}$$

where σ_w^2 and σ_x^2 represent the variance or energy of the watermark and the host document respectively. This ratio is set to $\text{WDR} = -23.6163$ dB, so that the watermark power is low relative to the document energy [6].

The calculation of the detection probability for decreasing quantization step sizes leads to the curves depicted in Fig. 6(a) and (b) on a linear and logarithmic scale respectively. As these figures show, the detection probability indeed decreases when the data is very coarsely quantized, but increases as the quantization step becomes smaller. Thus the detection probability increases for higher values of σ_w/Δ, which correspond to smaller quantization step sizes, and decreases in the opposite case. As these figures show, the probability of detection is above 0.90 for values of σ_w/Δ above 0.75. The detector can consequently be considered very reliable for these quantization step sizes, since the detection probability is very high, although the probability of false-alarm $P_{\rm fa} = 10^{-3}$ is very low.

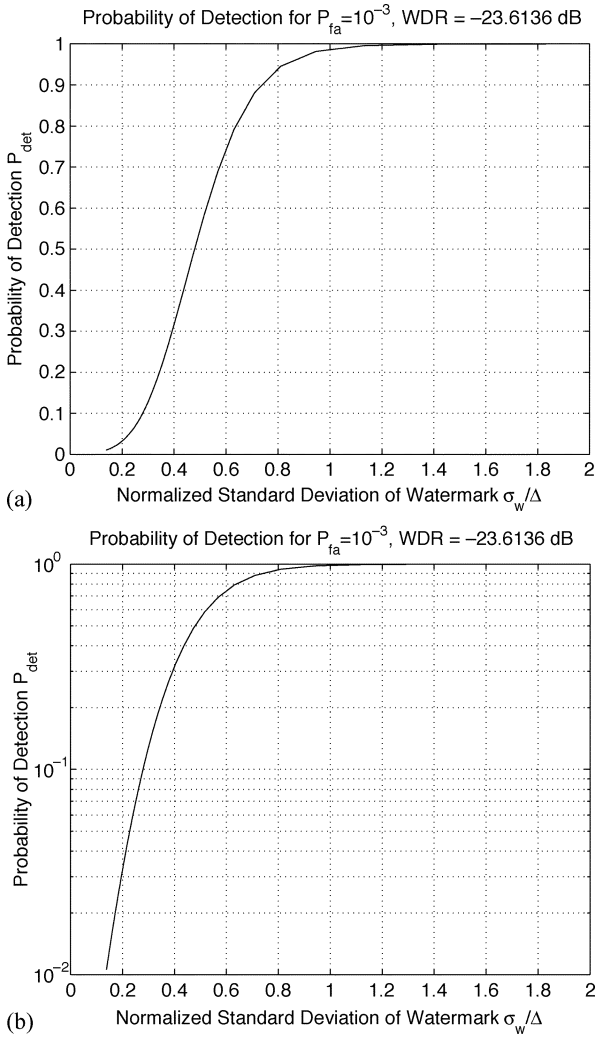

Fig. 6. Probability of detection for varying quantization step sizes, fixed $P_{\rm fa}$, and WDR for the fifth DCT coefficient of Akiyo.

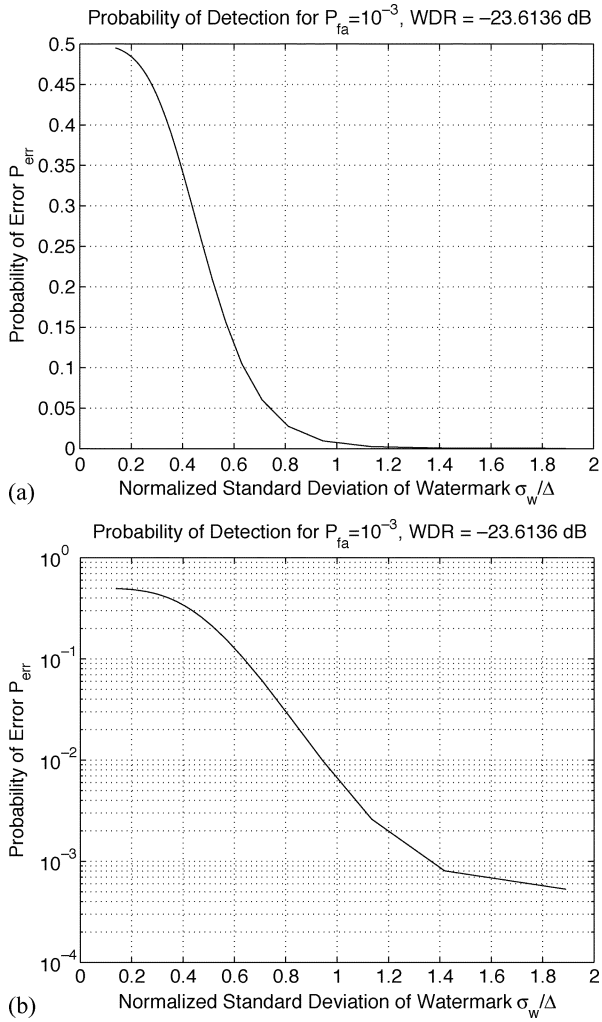

Fig. 7. Probability of error for varying quantization step sizes, fixed $P_{\rm fa}$, and WDR for the fifth DCT coefficient of Akiyo.

Another way to examine the detector performance is through the evaluation of the error probability. The error probability is defined as the probability of detecting a nonexistent watermark $P_{\rm fa}$, known as the probability of false alarm plus the probability of "missing" a watermark, $P_{\rm ndet}$ multiplied by the respective a priori probabilities for H_0 and H_1. In the case examined here, both hypotheses have equal a priori probabilities 1/2, so the error probability is

$$P_e = \frac{1}{2}(P_{\rm ndet} + P_{\rm fa}) = \frac{1}{2}(1 - P_{\rm det} + P_{\rm fa}). \quad (39)$$

This quantity is computed using (33) and (34) and plotted against decreasing quantization step sizes, or increasing σ_w/Δ, in Fig. 7(a) and (b). The false-alarm probability is set to $P_{\rm fa} = 10^{-3}$ and the same watermark, with $\mathrm{WDR} = -23.6163$ dB, is embedded in the host signal. In this case, the error probability is very high for high quantization step sizes or low normalized watermark standard deviation. As before, the error probability becomes very small when σ_w/Δ is above 0.75. Fig. 7(a) and (b) shows that the probability of error becomes close to 0 when the quantization step size decreases and the corresponding normalized watermark standard deviation increases.

B. Detector Robustness for Watermarks of Varying Strength

The detector performance is also examined for a specific quantization attack and watermarks of varying strength. Fig. 8(a) and (b) depicts the detection probabilities for values of WDR ranging from about -35 to -10 dB and for $P_{\rm fa}$ fixed to 10^{-3}. The probability of detection is quite low for small values of WDR, which is logical since the embedded watermark is very weak in these cases. However, for WDR above about -18 dB the detection probability is very high, above 0.90. These values of WDR are quite usual, as watermarks with similar values of WDR have been used in [6]. Thus, the detector can be considered reliable in the presence of quantization since its performance is very good for watermarks often used in practice. This is also evident in Fig. 9(a) and (b), where the error probability of (39) is plotted for the same simulation settings. The error probability is very high, close to 0.50, for very weak watermarks, but it becomes very low, below 0.10, when the WDR is above -18 dB, as expected from the previous results.

C. Experimental Values of m_0 and σ_0^2

The results of (27) and (32) can be verified experimentally by calculating the mean and variance of the log-likelihood ratio ex-

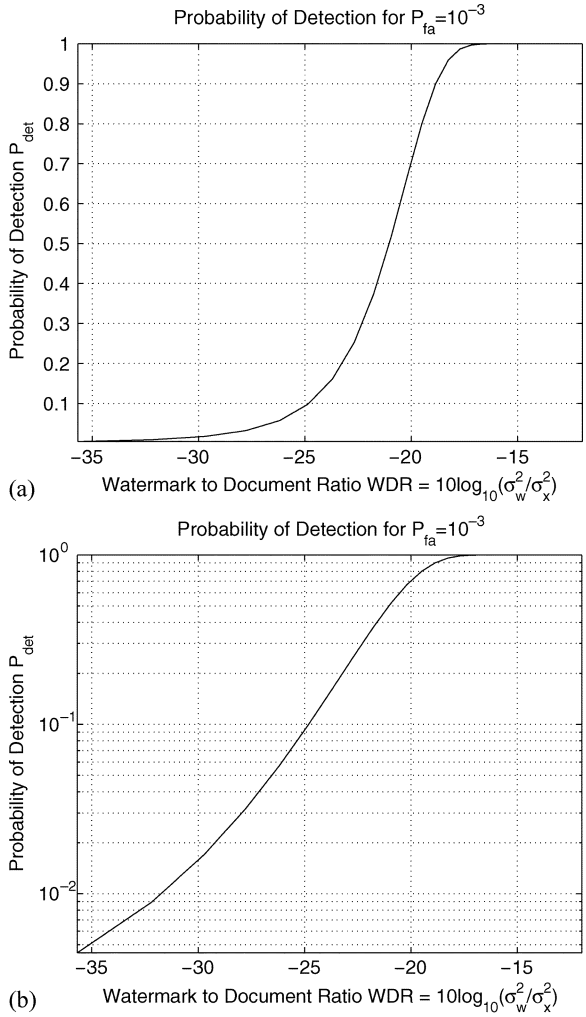

Fig. 8. Probability of detection for watermarks of varying strength and fixed $P_{\rm fa}$ for the fifth DCT coefficient of Akiyo.

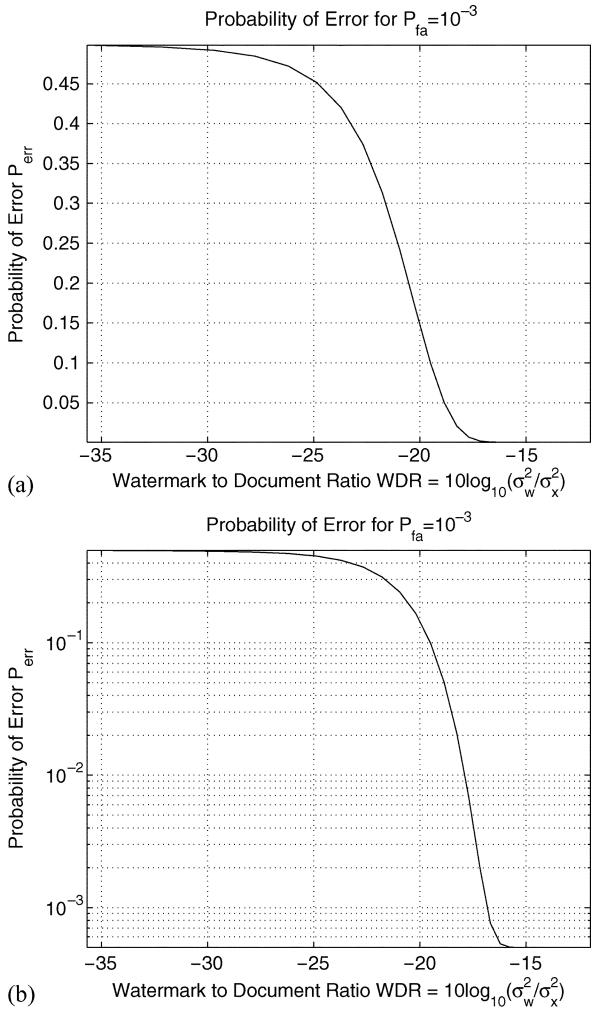

Fig. 9. Probability of error for watermarks of varying strength and fixed $P_{\rm fa}$ for the fifth DCT coefficient of Akiyo.

perimentally as well as theoretically. In particular, Monte Carlo experiments are carried out by embedding a large number of pseudorandomly generated watermarks in the host data with the procedure described in Section III. For the Monte Carlo tests, 200 watermarks are generated from the corresponding pseudorandom sequences. The log-likelihood ratio is then calculated for every one of these experiments and its mean and variance are finally computed and compared against the theoretically estimated values. To verify the robustness of the results and to form a better picture of the accuracy of the proposed estimators, we conduct such Monte Carlo experiments for watermarks of varying strength. In particular, different watermarks are embedded in the host data prior to quantization and the log-likelihood ratio is computed with (18). The mean and variance of the log-likelihood ratios of every watermark are found experimentally from the Monte Carlo runs, as well as theoretically from (27) and (32). In Fig. 10(a) and (b), it can be seen that the theoretical estimates of the log-likelihood ratio mean and variance are accurate for watermarks of varying strength. This can also be seen in Table I, which shows the values represented graphically in Fig. 10. These results show that the estimator is almost always very accurate, since very small differences between the theoretical and experimental estimates are observed.

The accuracy of the estimates of m_0 and σ_0^2 is especially important for the verification of the curves of Figs. 6–9 and 11. The probabilities of false-alarm, detection and consequently the error probability too, all depend on the values of these parameters, as can be seen from (33), (34), and (37). Thus, if the number of free parameters is reduced, by predefining the probability of false alarm for example, these probabilities depend solely on the ratio mean and variance. Since the experimentally computed means and variances practically coincide with their theoretically computed values, it is obvious that the experimental error and detection probabilities will be very close to their theoretic values. Consequently, the accuracy of (27) and (32) permits the theoretic investigation of the detector performance under quantization and the comparison of various watermarking schemes.

D. ROC Curves Under Quantization

The relation of the detection probability with the probability of false alarm is depicted in the ROC curves of Fig. 11(a) and (b). The WDR is set to $\mathrm{WDR} = -28.689$ dB in this case, while the $P_{\rm fa}$ takes values from 10^{-4} to 10^{-1}. For the very small values of $P_{\rm fa}$, close to 10^{-4}, the detection probability for unquantized data is relatively high, remaining above 0.75–0.80, but the effects of quantization are quite serious, as the values of

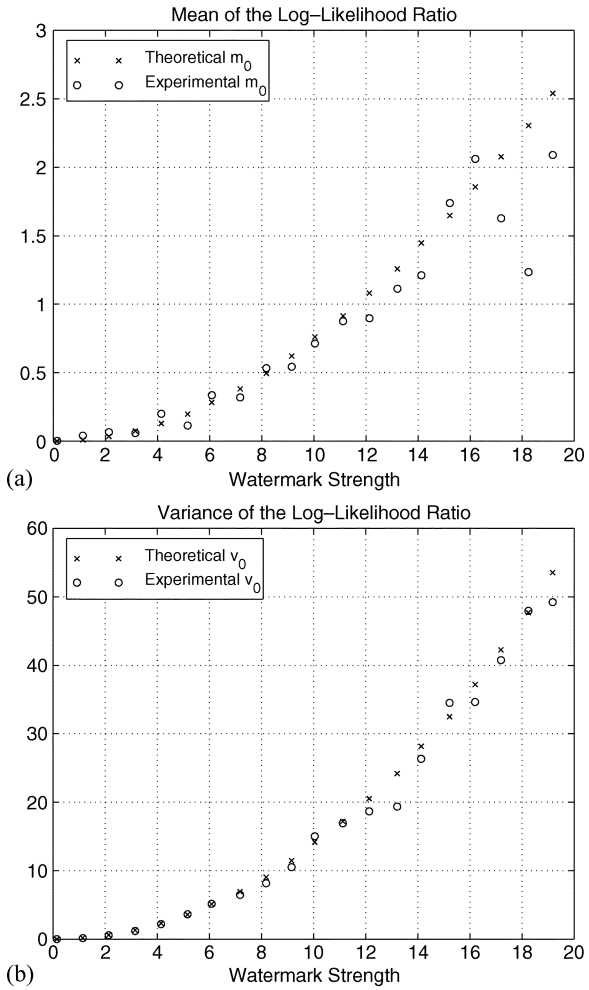

Fig. 10. Theoretical and experimental values of the log-likelihood ratio mean and variance.

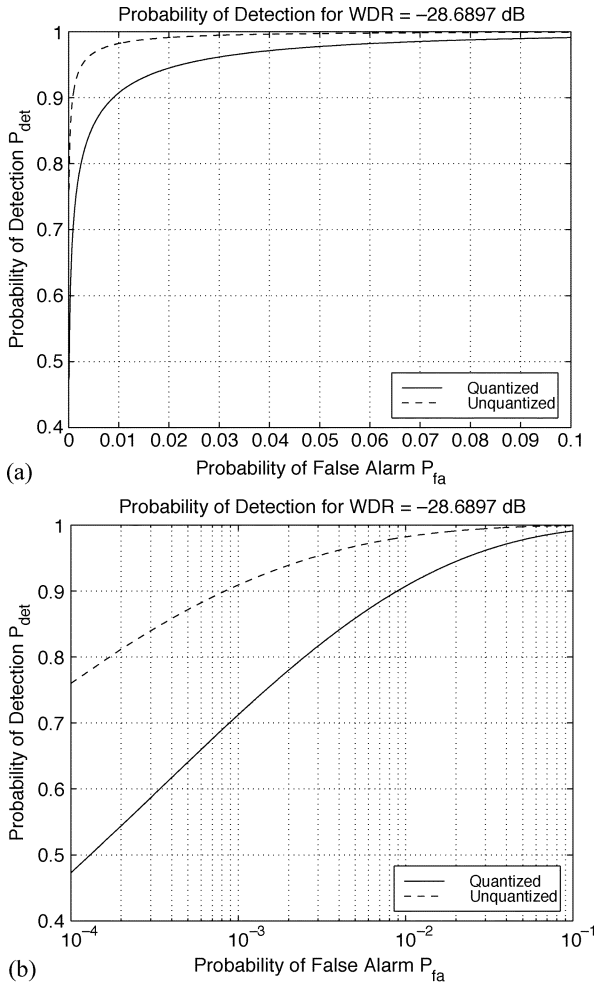

Fig. 11. ROC curves: probability of detection versus probability of false alarm for quantized and unquantized data from the fifth DCT coefficient of Akiyo.

TABLE I
THEORETICAL AND EXPERIMENTAL VALUES OF THE LOG-LIKELIHOOD RATIO MEAN AND VARIANCE

SNR			
Theoretical m_0	Experimental m_0	Theoretical v_0	Experimental v_0
0.000062	0.003166	0.002114	0.002101
0.009515	0.040358	0.171298	0.157676
0.034222	0.065259	0.611581	0.561945
0.073989	0.058685	1.324060	1.189549
0.128752	0.199737	2.310508	2.198796
0.198442	0.113571	3.573362	3.631999
0.282716	0.335679	5.115853	5.150009
0.381381	0.318942	6.941894	6.459357
0.494315	0.532581	9.056030	8.164082
0.621062	0.543422	11.463795	10.527803
0.761264	0.712844	14.171459	15.008780
0.914510	0.875945	17.186144	16.924128
1.080292	0.896968	20.515959	18.660339
1.258161	1.113276	24.169739	19.350195
1.447444	1.211168	28.157642	26.336855
1.647435	1.739450	32.490906	34.510052
1.857520	2.060917	37.181362	34.637211
2.076785	1.627385	42.242653	40.753242
2.304234	1.234108	47.689831	47.950150
2.539003	2.089883	53.538097	49.216347

$P_{\rm det}$ for quantized data are below 0.70. However, this is the case only for very small values of $P_{\rm fa}$. As Fig. 11(a) and (b) shows, the detection probability is above 0.80 for $P_{\rm fa}$ above 3×10^{-3}, where the corresponding detection probability without quantization is above 0.90. Very accurate values of $P_{\rm det}$, even in the presence of quantization, are finally obtained for $P_{\rm fa}$ above 8×10^{-3}, where the probability of false alarm for quantized data is above 0.90.

From these results one can conclude that quantization does affect the detector performance, as there is a decrease of the detection probability in comparison to the detection probability for unquantized data. However, as it is obvious from the previous discussion and Fig. 11(a) and (b), the detector is not rendered useless in the presence of quantization. On the contrary, very high detection probabilities can still be obtained, as long as the probability of false alarm is not very low. As seen in the previous experiments, the performance of the detector is very good even for values of $P_{\rm fa}$ as low as 10^{-3}. Thus, its results can be considered reliable in the presence of quantization, even for quite restrictive probabilities of false alarm. Naturally, this increases the detector's usefulness and reliability, as its performance does not deteriorate under quantization. This analysis finally permits us to estimate the detector performance *a priori* and predict the detection results under various restrictions.

In order to verify the ROC curves, we conduct experiments with 200 pseudorandomly generated watermarks and compare

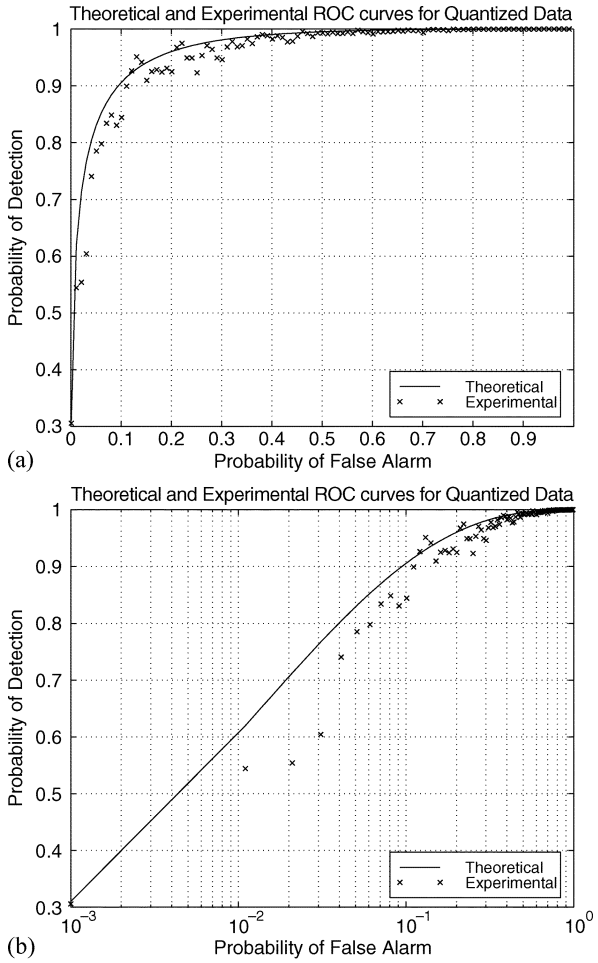

Fig. 12. Experimental and theoretical ROC curves for quantized data from the fifth DCT coefficient of Akiyo.

the actual detection probabilities with their theoretical values. This number of Monte Carlo runs actually permits the verification of probabilities down to 5×10^{-3}, which is why there are fewer experimental results for low values of P_{fa}. Fig. 12(a) and (b) depicts the theoretically estimated ROC curves for quantized data as well as the corresponding experimental results. The theoretical curves are derived from (37) using (27) and (32) for m_0 and σ_0^2, respectively. The empirical ROC curves are computed using the experimental values of the likelihood ratio mean and variance found from the Monte Carlo runs. As it is obvious from Fig. 12(a) and (b), the experimental results follow the theoretical ROC curves quite closely. Thus, the theoretically predicted ROC curves and in general the prediction of error probabilities in the presence of quantization can be considered reliable, since they coincide with the actual empirical values. Finally, these results confirm the theoretical analysis of the effects of quantization on improved watermark detection of Section IV.

VI. Conclusion

In this paper, we examined the effects of the very common and nonmalicious quantization attack on watermarking schemes. We focused on the effects of the often used uniform scalar quantization on the detection of such a watermark. Specifically, we analyzed the effects of quantization on the nearly optimal Cauchy watermark detector, which is more accurate than the traditionally used correlation detector. For our analysis, we also used certain results from the theory of dithered quantizers and, thus, derived theoretical expressions for the effect of quantization on the improved watermark detector. These theoretical expressions of the dependencies between the quantization strength and the error and detection probabilities were used to examine the effects of a specific quantization to watermarks of varying embedding strength and the effects of varying quantization step sizes. It was shown that in the presence of quantization there is a slight degradation of the detector performance, which depends on the watermark strength and the quantization step size. Actually, it was found that if the watermark is not very weak and the quantization is not very coarse, the detector remains reliable, giving quite high detection probabilities. The theoretical analysis was also verified experimentally, providing empirical measures of the Cauchy detector performance in the presence of quantization.

Consequently, one can accurately predict the detector performance in the presence of quantization, even before actual experiments are conducted. Since the theoretical results are reliable, they can be used to predict the expected detection and error probabilities for a given watermark. Finally, it is also possible to know *a priori* the watermark strength necessary to achieve a certain probability of detection, false alarm and/or error. Such reliable theoretical results are very important, as the performance of most watermarking schemes under various attacks, such as the quantization attack examined here, is usually evaluated only empirically.

References

[1] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, 1996, pp. 211–214.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, pp. 1079–1107, July 1999.

[3] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images," *Proc. IEEE*, vol. 87, pp. 1108–1126, July 1999.

[4] M. G. Strintzis and D. Tzovaras, "Optimal construction of subband coders using lloyd-max quantizers," *IEEE Trans. Image Processing*, vol. 7, pp. 649–668, May 1998.

[5] ——, "Optimal pyramidal decomposition for progressive multi-dimensional signal coding using optimal quantizers," *IEEE Trans. Signal Processing*, vol. 46, pp. 1054–1069, Apr. 1998.

[6] J. J. Eggers and B. Girod, "Quantization effects on digital watermarks," *Signal Processing*, vol. 81, no. 3, 2001.

[7] ——, "Watermark detection after quantization attacks," presented at the 3rd Workshop on Information Hiding, Dresden, Germany, Sept./Oct. 1999.

[8] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1995.

[9] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55–68, Jan. 2000.

[10] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COMM-31, pp. 835–839, 1983.

[11] M. Shao and C. L. Nikias, "On Symmetric Stable Models for Impulsive Noise," Univ. Southern California, Los Angeles, Tech. Rep. USC-SIPI-231, Feb. 1993.

[12] G. Samorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic Models With Infinite Variance*. New York: Chapman and Hall, 1994.

[13] A. Briassouli, P. Tsakalides, and A. Stouraitis, "Hidden messages in heavy-tails: DCT-domain watermark detection using alpha-stable models," IEEE Trans. Multimedia, to be published.
[14] P. Tsakalides, "Array Signal Processing With Alpha-Stable Distributions," Ph.D. dissertation, Univ. Southern California, Los Angeles, CA, Dec. 1995.
[15] G. A. Tsihrintzis and C. L. Nikias, "Data adaptive algorithms for signal detection in subgaussian impulsive interference," *IEEE Trans. Signal Processing*, vol. 45, pp. 1873–1878, July 1997.
[16] R. M. Gray, "Dithered quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 805–812, Mar. 1993.
[17] R. A. Wannamaker, S. P. Lipshitz, J. Vanderkooy, and J. N. Wright, "A theory of nonsubtractive dither," *IEEE Trans. Signal Processing*, vol. 48, pp. 499–516, Feb. 2000.
[18] W. R. Bennett, "Spectra of quantized signals," *Bell Syst. Tech. J.*, vol. 27, pp. 446–472, July 1948.
[19] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Trans. Commun. Technol.*, vol. COMM-12, pp. 162–165, 1964.
[20] D. Tzovaras, N. Karagiannis, and M. G. Strintzis, "Robust image watermarking in the subband or discrete cosine transform domain," in *Proc. Eur. Signal Processing Conf. (EUSIPCO)*, Greece, Sept. 1998, pp. 2285–2288.
[21] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum perceptual watermarking for images, audio and video," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.
[22] I. J. Cox, M. L. Miller, and A. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, pp. 1127–1141, July 1999.
[23] J. R. Hernandez, F. Perez-Gonzalez, J. M. Rodriguez, and G. Nieto, "Performance analysis of a 2D-multipulse modulation scheme for data hiding and watermarking of still images," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 510–524, May 1998.
[24] J. A. Solomon, A. B. Watson, and A. J. Ahumada, "Visibility of DCT basis functions: Effects of contrast masking," in *Proc. Data Compression Conf.*, Snowbird, UT, 1994, pp. 361–370.
[25] A. J. Ahumada and H. A. Peterson, "Luminance-model-based DCT quantization for image compression," *Proc. SPIE*, vol. 1666, pp. 365–374, 1992.
[26] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, 1987.
[27] J. P. Nolan, "Maximum likelihood estimation and diagnostics for stable distributions," Dept. of Math. and Stat., American Univ., Washingon, DC, June 1999.
[28] J. R. Hernandez and F. Perez-Gonzalez, "Statistical analysis of watermarking schemes for copyright protection of still images," *Proc. IEEE*, vol. 87, pp. 1142–1166, July 1999.
[29] K. A. Birney and T. R. Fischer, "On the modeling of DCT and subband image data for compression," *IEEE Trans. Image Processing*, vol. 4, pp. 186–193, Feb. 1995.
[30] F. Müller, "Distribution shape of two-dimensional DCT coefficients of natural images," *Electron. Lett.*, vol. 29, pp. 1935–1936, Oct. 1993.
[31] E. F. Fama and R. Roll, "Some properties of symmetric stable distributions," *J. Amer. Statist. Assoc.*, vol. 63, pp. 817–836, 1968.
[32] S. Cambanis, G. Samorodnitsky, and M. S. Taqqu, Eds., *Stable Processes and Related Topics*. Boston, MA: Birkhauser, 1991.

Alexia Briassouli recevied the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1999 and the M.S. degree in image and signal processing from the University of Patras, Patras, Greece, in 2000. She is currently working toward the Ph.D. degree in electrical engineering at the University of Illinois at Urbana-Champaign, Urbana.

From 2000 to 2001, she worked as a Research Assistant at the Informatics and Telematics Institute, Thessaloniki [Center of Research and Technology Hellas (CERTH)], Greece, participating in a European-funded research project. Her research interests lie in the fields of statistical signal processing and image processing. She has worked on the design of optimal watermark embedding and detection systems for images and video that are robust to various attacks. Her current research interests include statistical image processing and computer vision, and problems such as motion estimation and segmentation for video.

Michael G. Strintzis (M'70–SM'80–F'04) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively.

He then joined the Electrical Engineering Department, University of Pittsburgh, Pittsburgh, PA, where he served as an Assistant Professor (1970–1976) and an Associate Professor (1976–1980). Since 1980, he has been a Professor of electrical and computer engineering with the University of Thessaloniki, Thessaloniki, Greece, and, since 1999, Director of the Informatics and Telematics Research Institute, Thessaloniki. His current research interests include 2-D and 3-D image coding, image processing, biomedical signal and image processing, and DVD and Internet data authentication and copy protection.

Dr. Strintzis has been serving as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since 1999. In 1984, he was the recipient of a Centennial Medal of the IEEE.

Iterative Error Detection and Correction of H.263 Coded Video for Wireless Networks

Ekram Khan, Stefan Lehmann, Hiroshi Gunji, and Mohammed Ghanbari, *Fellow, IEEE*

***Abstract*—Due to the use of variable length code (VLC), a single bit error in the discrete cosine transform (DCT)-based coded bit stream, such as H.263, may propagate up to the end of the group of blocks (GOBs) or slice. In this paper, we propose an iterative error detection and correction algorithm for the slice mode of the H.263 bit stream. The visibly erroneous macroblocks (MBs) in the decoded frames are detected by checking a set of error detection conditions derived from the redundant information (such as neighboring MBs and inner-DCT block similarity measure) inherent within the frame. In each slice, the part of the bit stream before the first erroneous MB is decoded in the conventional manner. The remaining part after this MB is redecoded iteratively, skipping one bit at a time until a decodable subbit stream is found. In case an MB is nondecodable, it is replaced with gray level. Once all the slices are checked, the frame is reconstructed and again checked for erroneous MBs. This process is repeated until the decoder detects no more corrupted MB. The proposed step-by-step decoding technique limits the error into a few MBs only, which can easily be concealed by any error concealment technique. The simulation results demonstrate that our scheme can recover the corrupted frames under the bit error rates up to 1% over binary symmetric channel (BSC), and improve the concealed picture quality by 4–6 dB over the conventional methods.**

***Index Terms*—Error concealment, error detection and correction, error resilience, H263+, noisy channels, video coding, video transmission, wireless network.**

I. Introduction

MOST OF THE standard video codecs like H.263 are based on motion compensation—discrete cosine transform (MC-DCT) coding scheme, which use the variable length codes (VLCs), such as Huffman for further compression. The use of VLC in the erroneous compressed data would not allow even the noncorrupted parts to be correctly decoded until after the synchronization point, i.e., start of the following group of blocks (GOBs) or slice. Moreover, due to loss of synchronization between the encoder and decoder states, the error may also propagate into the temporal domain. Due to these reasons, the emerging video coding techniques include provisions for error resilience particularly in H.263 [1]–[3] and MPEG [4]. To limit the effect of error propagation and hence to improve the video quality against the transmission error, several techniques have been proposed in the literature. These can be grouped into four categories: feedback channel or retransmission approach, forward error correction (FEC) or channel coding approach, error resilience approach and error detection and correction approach.

The use of the feedback channel enables the decoder to inform the encoder about successful or unsuccessful reception by sending positive (ACK) or negative (NACK) acknowledgment [5]. Then to prevent error propagation, the encoder may use multiple frame prediction [6], [7] or utilize INTRA MB [5], [8], [9]. However, feedback channels may introduce additional delay and complexity and they are also affected by channel errors. Moreover, they may not be available in many scenarios such as video broadcasting, multicasting, or multipoint communications. The FEC or channel coding approach is also commonly proposed to limit the effect of error propagation [10]. However, the increase in data rate due to insertion of redundant bits may not be suitable for very low bit rate applications.

To increase resilience of codecs to errors, the new generation video codecs use synchronization markers for spatial error localization [2], [4]. These markers allow bit synchronization of the decoder by resetting spatial location in the decoded frames and prevent the propagation of spatial prediction errors. However if used too frequently, synchronization markers will also increase the bit rate. Use of bit reorganization algorithms such as error resilience entropy coding (EREC) [11], [12], which is an efficient alternative to the insertion of synchronization code, can reduce this overhead. It should be noted that even in EREC technique, the errors in one slot can propagate to the others, though the degradation in quality is somewhat alleviated by packetizing all bits to the fixed length slots with least redundancy. Therefore, it is often preferred to adopt *a posteriori* approach aimed at improving the perceptual image quality of decoded image. These techniques, generally termed as "error concealment," which attempt to restore erroneous blocks in the damaged image by some means of interpolation performed either in spatial, temporal or transform domain by exploiting correctly received information. A general overview of the various error concealment methods can be found in [13]. It should be noted that although the error concealment techniques do not increase the transmission bandwidth, but can yield very poor performance under high channel error rate. Combinations of the above techniques, such as FEC with automatic repeat request (ARQ) [14], FEC with concealment [15], and resynchronization with error concealment [16] have also been tried and reported. Though hybrid schemes work better than individual techniques, they still suffer with the basic problem of individual techniques.

The error detection and correction is another approach, which can be performed at either image level or bit level. The image level approach exploits the resulting disturbances due

Manuscript received December 21, 2001; revised February 25, 2003. This paper was recommended by C. W. Chen.

E. Khan was with the Department of Electronic Systems Engineering, University of Essex, Colchester, U.K. He is now with the Department of Electronics Engineering, Aligarh Muslim University, Aligarh 202002, India (e-mail: ekhan@lycos.com).

S. Lehmann and M. Ghanbari are with the Audio and Video Networking Research Laboratory, Department of Electronic Systems Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: ghan@essex.ac.uk).

S. Gunji is with the Renesas Technology Corporation, Tokyo 187-8588, Japan.

Digital Object Identifier 10.1109/TCSVT.2004.837018

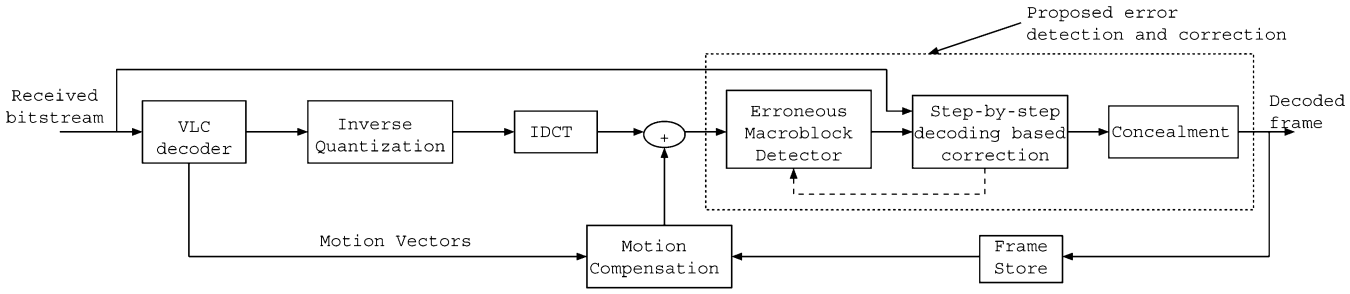

Fig. 1. Modification to the H.263 decoder for proposed error detection and correction.

to the channel errors introduced in the redundant information inherent in the neighboring pixels of the reconstructed frame. In [17], candidate locations for errors in the JPEG-coded VLC bit stream are identified using decoded image characteristics and bits are then flipped in the received VLC bit stream to minimize a cost function. However, this technique employs frequent restart markers in the bit stream to limit error propagation in the JPEG transmission application. In [18] also, the error detection and correction in the image data with frequent use of restart marker has been reported. A neural network based approach for DCT information recovery has been suggested in [19]. A bit level approach is suggested in [20], in which a symmetric VLC codeword is used and the resulting encoded stream can be decoded with both forward and backward (at bit level) from a known location to maximize the amount of decoded data. However, these codes are slightly nonoptimal. Further, the reversible variable length codes (RVLC) [20] are best suited to detect the errors when bits from erroneous bit onward are not decodable or when invalid codes occur. However, in general, all the erroneous bits may not result in invalid codes, and the bit stream may be decoded (although erroneously) even in the presence of errors. Also one cannot find the exact location of errors within the bit stream with RVLC. In addition, some errors in the bit stream have almost no impact or very little impact on the picture quality, and it would be waste of resources to correct them. For these reasons, we have considered error detection at picture level and correction at the bit stream level.

In this paper, we propose an *a posteriori* approach aimed at improving the perceptual video quality after erroneous H.263+ coded bit stream are received. Our error detection algorithm exploits the inherent redundancies within and outside a macroblock (MB) to detect erroneously decoded MBs. The redundancies are measured in terms of a set of parameters (based on MB and inner-DCT block similarities). For each MB, parameters are compared with either absolute or relative thresholds (which are determined experimentally) or both to detect erroneous MBs. In order to detect maximum number of corrupted MBs, the errors in I- and P-frames are detected in two and one passes, respectively. In the error detection phase, more emphasis is given to detect the first erroneous MB of the slice correctly. After the error detection, an iterative redecoding-based correction algorithm is applied to the erroneous slices. A step-by-step decoder initializes a pointer in the bit stream at the location of the first bit of the first erroneous MB in the slice. The part of the slice preceding this point is decoded conventionally, and the pointer is incremented by one bit. The remaining part of the slice from the current pointer location to the end is checked whether decodable or not by checking a set of conditions. If not decodable, the pointer location is again incremented by one bit forward. This process is repeated until decodable bit stream is found. The step-by-step redecoded part is right aligned and any missing MB is replaced with gray values. In principle, when pointer location corresponds to the first bit of any following MB, it is likely that part of the bit stream thereafter is correctly decodable. However, in practice, the step-by-step decoded part of the slice may still contain some erroneous MBs. Hence, the proposed error detection and correction scheme is applied iteratively until no further corrupted MB (except the stuffed gray MB) is detected. Finally, spatial (for I-frame) [21] and temporal [22] (for P-frames) concealment techniques are used to conceal gray-valued MBs.

This paper is organized as follows. A brief overview of H.263 video coder with bit stream syntax modification is described in Section II. The proposed error detection and correction scheme is described in detail in Section III. The simulation results are included in Section IV, followed by concluding remarks in Section V.

II. H.263 Video Codec

In this section, we describe a video-coding framework employed throughout the paper, including the modifications in the bit stream syntax and the decoder.

A. H.263 Video Coder

The standard video coding systems like H.263 [3] and MPEG-4 [4] envisage various methods to improve their resilience toward channel errors. For example the Annex-K of H.263+ [1] supports the slice-structured mode where all MBs of one slice can be decoded independent of the content of other slices by preventing the prediction of motion vectors (MVs) to cross the slice boundaries. There is however, a need for extra information to decode a slice, because information conveyed in the picture header is not repeated in the slice header. In this work, we have used slice-structured mode with slight modification in the bit stream syntax to be discussed in Section II-B. Our proposed error detection and correction technique is incorporated in the modified decoder shown in Fig. 1.

B. Bitstream Syntax Modification

Since in the slice structured mode of H.263 bit stream, the quantization information and MVs are encoded differentially, if one or more MBs in the middle of the slice are omitted, then it is impossible to decode the quantization information and MVs of the following MBs. In order to avoid this problem, the bit stream syntax is slightly modified from the conventional slice structure mode. The modifications are made only in the slice and MB layer, whereas picture and block layers remain unchanged [1]. In the slice layer, SSTUF is modified and the last quantizer information (LQUANT) and the last MV value (LMVV) are added.

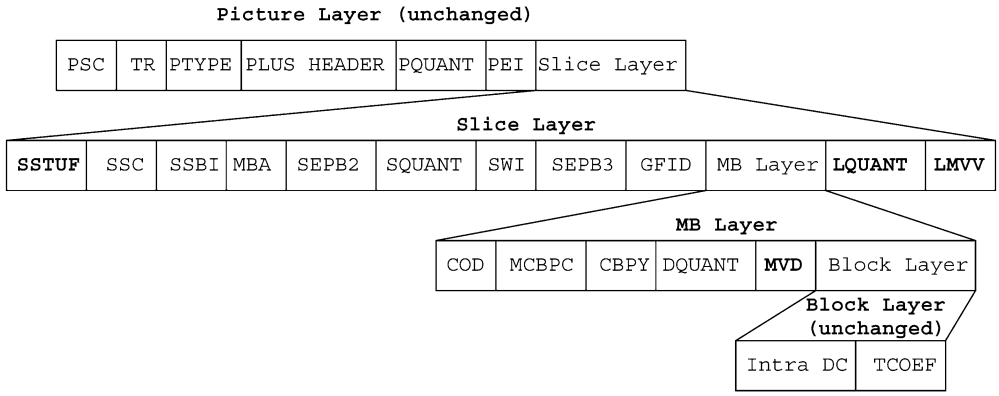

Fig. 2. Modified bit stream syntax.

In the MB layer, motion vector difference (MVD) prediction is modified. Due to these modifications, for QCIF at 64 kbit/s the overall bit rate increases by 1.2%. The modified bit stream syntax is shown in Fig. 2, where added and modified fields are shown in bold faces and are explained below.

SSTUF: The SSTUF field is modified similar to MPEG-4 to incorporate LQUANT and LMVV fields in the slice layer. SSTUF field takes 1–8 bits and is a codeword of variable length, consisting of one zero followed by less than eight 1s such that SSC is byte aligned. This enables the decoder to locate LQUANT and LMVV in the bit stream, which otherwise is not possible with SSTUF specified in the conventional H.263.

LQUANT: LQUANT is a fixed length code word of five bits, which indicates the quantizer information (QUANT) of the last MB in the slice and is added after the MB-layer field. It enables the decoder to check and ensure that QUANT is decodable for the MBs following the erroneous one.

LMVV: The LMVV field is added as the last field of the slice layer and contains last motion vector in the slice. It ensures that motion vectors can be decoded starting from any MB in a slice and allows calculating motion vectors in the backward direction. It is incoded with a prediction value of zero for both horizontal and vertical components and is coded with the same VLC codes of motion vectors, but bits are placed in reverse order. LMVV field is used for intercoded frames only.

MVD: In the MB layer, the MVD prediction is modified according to data partitioned slice mode of H.263 (Annex. V) [1] but with conventional VLC rather than RVLC. The first motion vector in the slice is encoded using a prediction value of zero for both horizontal and vertical components. The motion vectors of the subsequent MBs are encoded using last coded value as the predicted one.

III. Error Detection and Correction

In this section, we describe the details of our error detection and correction scheme which is performed at the MB level in each slice of the frame.

A. Error Detection

In the presence of transmission errors, a decoded frame may contain three types of MBs: 1) nondecodable MB (fatal error); 2) correctly decodable MB; and 3) decodable but erroneous MB. In our scheme, the first type of MB is detected during a slice but the other two types are detected after the decoding.

1) Error Detection While Decoding: When the H.263 decoder is decoding a slice from a corrupted bit stream, the decoder may detect the slice as erroneous if any of the following conditions occurs.

1) Invalid VLC (MCBPC, CBPY, MVD, and TCOEF) code is detected.
2) Quantizer information goes out of range.
3) Invalid INTRA DC code is detected.
4) Escaped TCOEF with level 0 is detected.
5) Coefficient overrun occurred.
6) A motion vector refers out of picture.
7) The number of bits used in decoding of the slice is different from that in the slice.
8) The quantizer information in the last MB is different from LQUANT.
9) The motion vector of the last coded MB is different from LMVV.

If any of the conditions (1)–(6) occur, the decoder will detect a fatal error, and the MBs beyond that point are nondecodable and all pixels of these MBs are replaced with some gray values (say 128). However, if in any slice, conditions (7)–(9) occur, the decoder will decode that slice erroneously, but it cannot determine the location of the error.

2) Error Detection in a Decoded Frame: After a frame is decoded, the visibly erroneous MBs will be detected using redundant information inherent in the neighboring MBs. Since a channel error most often affects either 8×8 pixels DCT blocks or 16×16 pixel MBs (containing four luminance and one each of the chrominance DCT blocks), in the proposed error detection scheme, the emphasis is given on detecting erroneous MBs by exploiting either the characteristics of the neighboring MBs or DCT blocks within an MB. Furthermore, it is very likely that an MB following an erroneous one is itself affected by the error, the main challenge is to detect the first erroneous MB in a slice. In order to detect corrupted MBs, the following similarity measures are exploited:

1) MB characteristics :
   a) MB boundary;
   b) MB mean pixel value.
2) inner DCT block characteristics:
   a) inner boundary;
   b) individual DCT block boundaries;
   c) DCT block's mean pixel value.

For each of these similarity measures, a parameter (defined in the following subsection) is calculated and compared with either an absolute or a relative threshold or both. For an MB under consideration, if the parameter has a value greater than an absolute

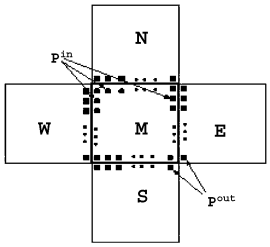

Fig. 3. Boundary pixels of an MB.

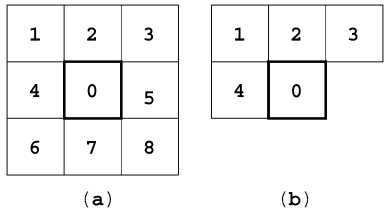

Fig. 4. An MB and its (a) eight neighboring MBs and (b) four preceding MBs.

threshold, the MB is considered as erroneous. However, if the value of the parameter is greater than a relative threshold, it is considered as a candidate of being erroneous (subject to other criteria to be fulfilled).

a) MB characteristics:

1) MB boundary similarity measure: In an image, there exists sufficient similarity among the adjacent pixels, and hence across the MB boundaries, although there may exist clear edges along their boundaries. In order to detect errors, we use a parameter similar to [17], called average intersample difference across boundary (AIDB), which can be defined as the total absolute intersample difference between the pixels across the boundaries of the neighboring MBs divided by the total number of the boundary points. As shown in Fig. 3, let M represent an MB under consideration and N, S, E, and W are its four connected MBs in the north, south, east and west directions respectively. Further, let p^{in} represent boundary pixels inside the MB M, and p^{out} represent boundary pixels (adjacent to p^{in}) in one of the neighboring MBs. Then the AIDB of M with an MB $X \in \{N, S, E, W\}$, $\text{AIDB}(M : X)$ is given as

$$\text{AIDB}(M : X) = \frac{1}{K}\left[\sum_{i=0}^{K-1} \left|p_i^{\text{in}} - p_i^{\text{out}}\right|\right] \tag{1}$$

where an MB is considered as of $K \times K$ dimension ($K = 16$ here). The AIDB can also be calculated for more than one neighboring block boundaries of MB M as

$$\text{AIDB}(M : N, S, E, W) = \frac{1}{4}\left\{\begin{array}{l}\text{AIDB}(M : N) + \text{AIDB}(M : S) \\ +\text{AIDB}(M : E) + \text{AIDB}(M : W)\end{array}\right\}. \tag{2}$$

If AIDB is larger than some threshold, MB M is considered as being corrupted.

2) MB mean pixel value similarity: In this approach of error detection, it is assumed that the mean or DC value of an MB does not change significantly with that of its neighboring MBs. However, if it does, an MB is likely to be corrupted by errors. To measure the mean pixel value similarities, we define a parameter designated as mean pixel difference (MPD), which is defined as the absolute difference of mean pixel value of an MB under consideration to the average of mean pixel values of the neighboring MBs. If this difference is significant, the MB is considered as erroneous. Here we have considered two cases:
   i) Eight surrounding MBs are considered as neighbors as shown on Fig. 4(a).
   ii) Only four preceding MBs are considered as neighbors as shown in Fig. 4(b).

In general, MPD with n neighboring MBs is defined as

$$\text{MPD}_n = \left|\overline{p}_0 - \frac{1}{n}\sum_{m=1}^{n} \overline{p}_m\right| \tag{3}$$

where n is the number of neighboring MBs, and $\overline{p}_m$ is the mean pixel value of the mth neighboring MB (ref. Fig. 4). The MPD_8 [Fig. 4(a)] is used to detect erroneous MBs, whereas MPD_4 with only four preceding MBs [Fig. 4(b)] is useful in detecting the first erroneous MB of the slice.

b) MB characteristics: The MB-characteristic-based measures discussed in the previous sections are good but not always guarantee to detect the erroneous MBs. Since each MB consists of four luminance and one of each chrominance blocks, sometimes it is possible that only one or two inner DCT blocks are corrupted. Therefore, the MB-based parameters may not be suitable for detecting erroneous MBs. In such cases, we propose similar analysis for the inner DCT blocks within an MB. However these can only be applied to the four luminance DCT blocks since there is only one DCT block of each chrominance per MB.

1) DCT block inner boundary similarity: It is based on the principle that in a noncorrupted MB, the pixel transition from one DCT block to another within an MB is generally very smooth, i.e., pixels at the DCT block boundaries are very similar. However, if one of the DCT blocks is erroneous, the boundary pixels of that block are likely to be different from their adjacent pixels in the other blocks. As shown in Fig. 5(a), an MB has two inner boundaries, horizontal and vertical, across DCT blocks totaling 16 pixels (for 16×16 size MB) across each side of the boundary. To measure the similarity among the pixel across boundaries, we define a parameter called inner average inter-sample difference across boundaries (IAIDB), which is similar to AIDB defined earlier, with the difference that AIDB is defined for outer neighboring MBs whereas IAIDB is defined within an MB. It is defined as follows:

$$\text{IAIDB} = \frac{1}{16}\left\{\sum_{i=0}^{15}\left|p_i^{\text{above}} - p_i^{\text{below}}\right| + \sum_{i=0}^{15}\left|p_i^{\text{left}} - p_i^{\text{right}}\right|\right\} \tag{4}$$

where p_i^{above} and p_i^{below} are the ith pixel above and below the horizontal inner boundary and p_i^{left} and p_i^{right} are the ith pixel on the left and right side of the vertical inner boundaries as shown in Fig. 5(a).

2) Individual DCT block inner boundary similarity: It is similar to the DCT blocks inner boundaries, except that only the boundaries of individual DCT blocks are considered. Here again each individual luminance DCT blocks have two boundaries separating them from horizontally and vertically adjacent blocks within an MB as shown in Fig. 5(b). To measure the similarities across the individual DCT blocks, we use the same parameter IAIDB as was

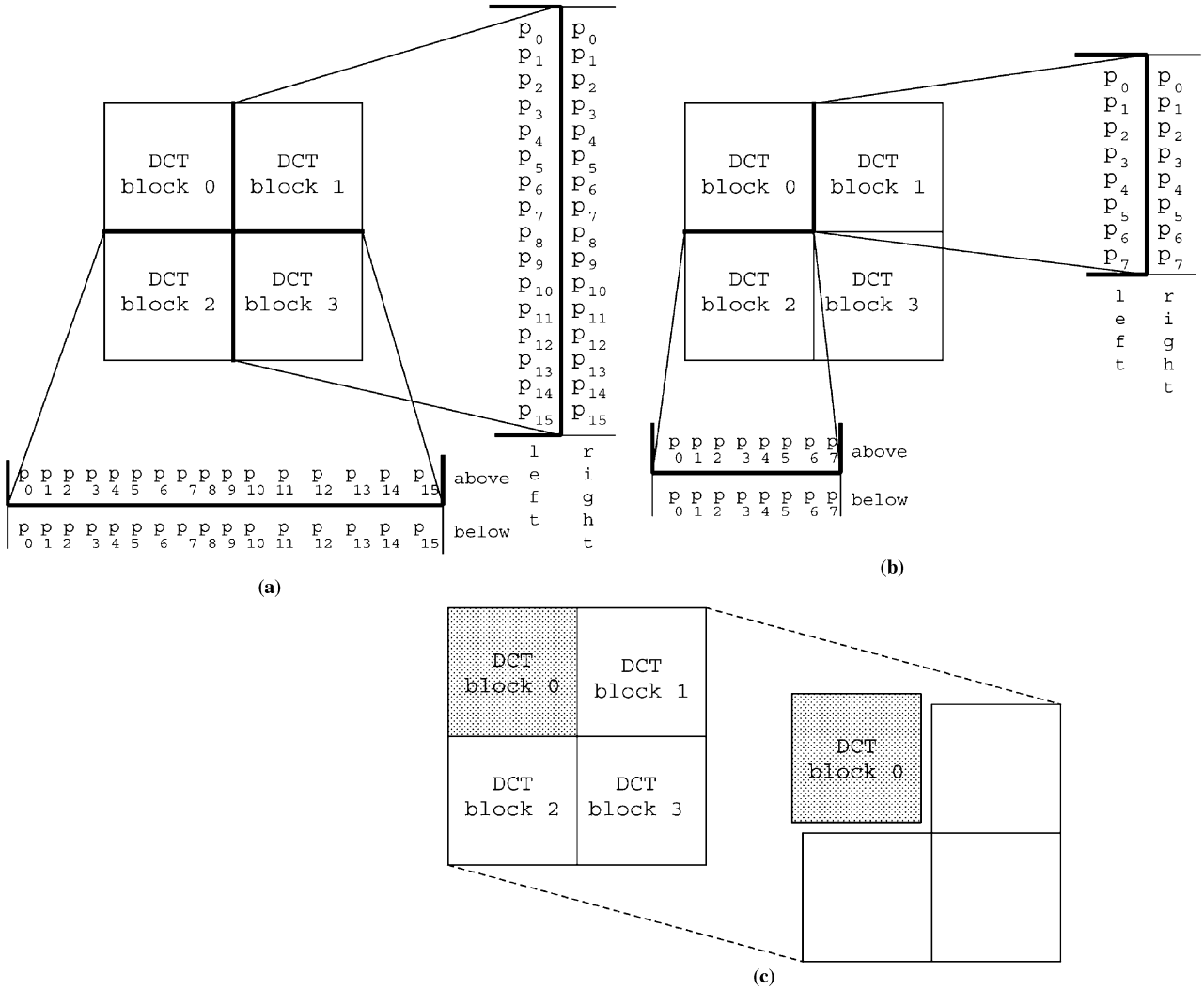

Fig. 5. DCT block analysis for error detection. (a) Inner DCT block boundary. (b) Individual DCT block boundary. (c) DCT block mean pixel value analysis.

defined in the previous subsection with subscript "block" to differentiate that it is the value of the parameter for a DCT block only. It is defined as

$$\mathrm{IAIDB}_{\mathrm{block}} = \frac{1}{8}\left\{\sum_{i=0}^{7}\left|p_i^{\mathrm{above}} - p_i^{\mathrm{below}}\right| + \sum_{i=0}^{7}\left|p_i^{\mathrm{left}} - p_i^{\mathrm{right}}\right|\right\} \quad (5)$$

where symbols have the same meaning as in (4). For this analysis, only a relative threshold is used to decide whether an MB is in error or not.

3) DCT block mean pixel value similarity: It is similar to the MB mean pixel analysis, except that now it is applied to DCT blocks. Again we define a parameter similar to MPD_n defined in (3), for a DCT block within an MB as a measure of the mean pixel value similarity. For any luminance DCT block within an MB, there are only three neighboring blocks in the same MB as shown in Fig. 5(c). We denote this parameter as $\mathrm{MPD}_3^{\mathrm{block}}$, which is the absolute difference between the average of all pixels of one DCT block to the average of all other pixels in the same MB, as given in (6) for the DCT-block0 of Fig. 5(c)

$$\mathrm{MPD}_3^{\mathrm{block}} = \left|\overline{p}_0 - \frac{1}{3}\sum_{i=1}^{3}\overline{p}_i\right| \quad (6)$$

where $\overline{p}_i = (1/64)\sum_{j=0}^{63} p_j^i$, is the mean pixel value of the ith DCT block (8×8) of an MB, and p_j^i represents jth pixel of the ith DCT block. Again because of too many DCT blocks in a frame, it is difficult to determine an absolute threshold for this parameter. Thus, only relative threshold for the luminance components will be used to detect the errors.

3) Error Detection Algorithm: Our simulation shows that if the parameters defined in Section III-A-2 are individually used to detect errors, not more than 20%–40% of erroneous MBs can be detected and also some of the correct MBs may be detected as erroneous. However, various combinations of these parameters can improve the performance by up to 93%. Our algorithm detects the visibly erroneous MBs in one and two passes for P and I-frames respectively. The reason for using two passes for I-frame is as follows. First, an I -frame contains much higher bits than a P-frame, and for a given bit error rate (BER), it is more likely to be corrupted. Second, P-frames can be concealed better than I-frames, since I-frames use intra-frame concealment. Thus in the I-frame, erroneous MBs should be detected more accurately, which is achieved in two passes. In the first pass the parameter AIDB, MPD_8, and IAIDB (in order) for each MB are compared with their corresponding absolute thresholds, and if one of them is larger, then the MB is considered erroneous, otherwise these parameters are compared with a combination of

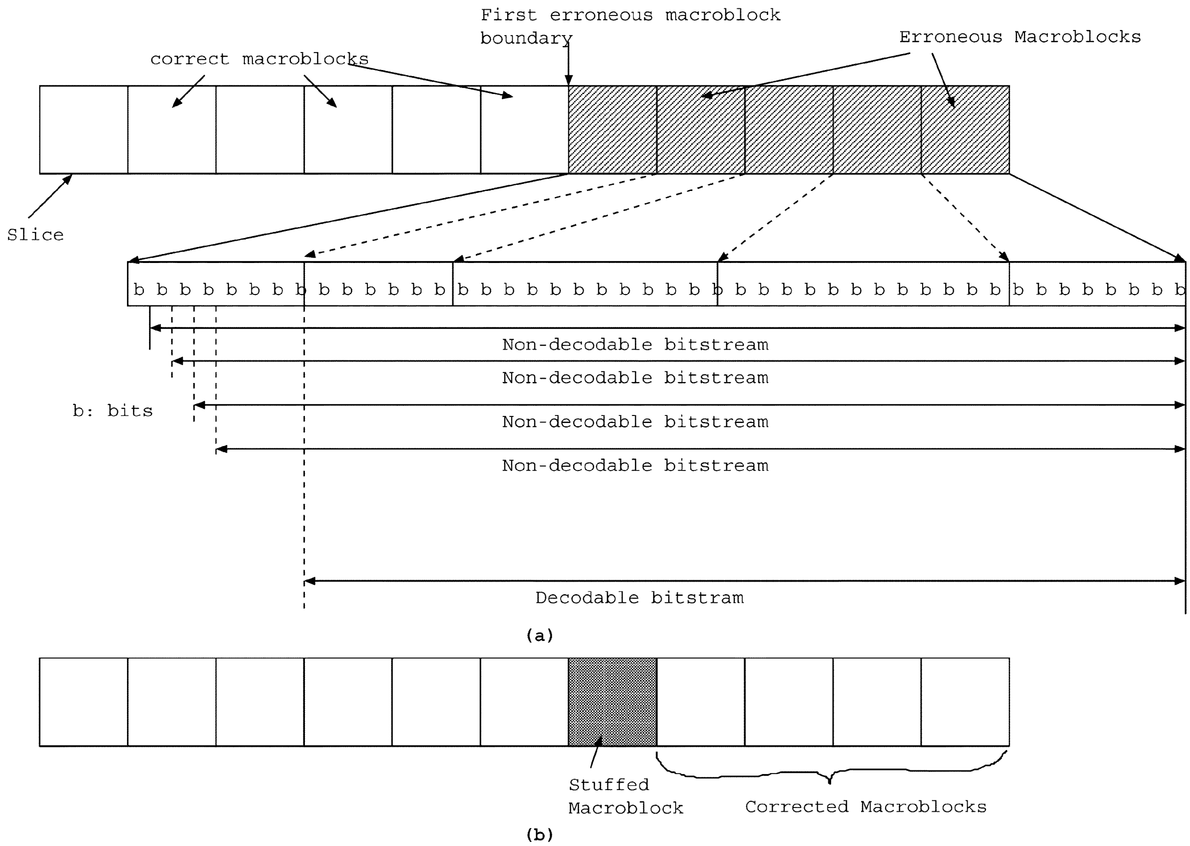

Fig. 6. Principle of step-by-step decoding. (a) Slice before the correction and sequential bit skipping. (b) Slice after correction.

their relative thresholds. By now most of the erroneous MBs are detected. However there may still be some MBs in which only one DCT block is erroneous and are not detected so far. To detect this type of errors, $\mathrm{IAIDB}_{\mathrm{block}}$ and $\mathrm{MPD}_3^{\mathrm{block}}$ are used and compared with the relative thresholds of both. There are still some less visible but erroneous MBs that could not be detected so far. In the second pass, to detect some more erroneous MBs, we detect the errors with a relatively lower threshold (reduced by one standard deviation). In addition, in the second pass further emphasis is given to detect the first erroneous MB of the slice. For that purpose, AIDB with SWN boundaries only (Fig. 3) and MPD_4 with four preceding MBs only [Fig. 4(b)] are used and compared with the relative thresholds of both.

In I-frames, errors are detected by applying first pass to each MB and second pass only to those slices that contain erroneous MB detected in the first pass. However, in P-frames errors are detected by using only absolute thresholds of AIDB, MPD_8, and IAIDB, and all steps of second pass in a single pass for each slice and each MB.

B. Error Correction

The basic idea behind our error correction algorithm is the fact that due to VLCs, the effect of bit errors in one MB may propagate beyond that MB. The objective is to limit the effect of error within the affected MB. This can be achieved if the decoder knows the location of the first erroneous block in the slice, and can decode the rest of the slice without this erroneous one. For this purpose the bit stream syntax is slightly modified so that the decoder can decode the slice from the middle. Since in the modified bit stream, each slice ends with LQUANT and LMVV that corresponds to the QUANT and MV of the last MB of the slice, the quantization and motion vector information of the MBs can be calculated from LQUANT and LMVV in backward direction.

Assuming that the decoder knows the location of the first erroneous MB (if any) in the slice, the step-by-step decoding shown in Fig. 6 works as follows. The part of the slice preceding the first erroneous MB is conventionally decoded and a pointer is initialized at the second bit of the first erroneous MB, i.e., skipping the first bit. The rest of the bit stream from that point is checked to see whether it is decodable or not by checking the condition (1)–(6) mentioned in the Section III-A. If any of these conditions occur, the bit stream is not decodable and the pointer is again incremented by one bit. This process is continued until correctly decodable bit stream is found and decoded. We call this part of the slice as step-by-step decoded part. If the number of decoded MBs in the slice is less than the actual number of MBs (which is known), the step-by-step decoded part is right aligned and any missing MBs are replaced by gray values (say 128). This process is repeated for each erroneous slice (found while decoding the frame) and thus the decoded frame is generated. Although it is expected that the step-by-step decodable part of the slice is obtained when pointer is pointing to the bit of an MB boundary, however, in practice it is possible that rest of the slice is decoded from the point different from MB boundary. It is also likely that decodable bit stream for the rest of the slice also contains some erroneous bits. In such cases the decoded frame may still contain erroneous MBs. For this reason the error detection and step-by-step decoding is repeatedly performed on the decoded frame until the error detector does not detect any more corrupted MBs, except the gray-valued MBs, which are

```
• Decode the frame from received bit stream.
• Check for erroneous macroblocks.
• If (frame contains any erroneous macroblock) do:
  {
     for each slice do:
    {
       if(contains no erroneous MB)
                decode as with conventional decoder
   else
   {
                    • Initialise pointer at first bit location of
                      first erroneous MB;
                    • decoder part of the slice from start to
                      pointer location conventionally;
                    • do
                      {
• increment the pointer by one bit location;
• decode remaining part of the slice starting from current
  pointer address;
• If (part is decodable) break;
                    }
• Decode this part of the slice with right alignment;
• Stuff the gray valued macroblocks between two decoded
  parts, if needed;
          }
    }
• Reconstruct the frame, and check again for error.
• If (no corrupted MB except grayed MBs) break;
  }
```

Fig. 7. Error detection and correction algorithm.

concealed later. A pseudocode for the proposed error detection and correction algorithm is given in Fig. 7.

C. Concealment

After detecting and correcting most of the erroneous MBs, the erroneous slices may still have some erroneous MBs (at least one in which the error actually occurred) which were filled in with gray values during the step-by-step decoding. These MBs can now be efficiently concealed with any standard error concealment technique. There are two basic approaches for error concealment: spatial and temporal. In the spatial interpolation, pixels of missing MBs are reconstructed using the neighboring spatial information whereas in the temporal interpolation, a lost MB is reconstructed from that in the previous frame shifted by an estimated motion vector. In this work, we employ both spatial and temporal interpolation techniques. In spatial interpolation, which is used for intracoded frames, the pixels of missing MBs are reconstructed as the median value of the corresponding pixels in eight surrounding MBs. For inter-coded frames, we use simple motion compensated temporal error concealment, in which the motion vectors of the missing MB is set to the median value of the motion vectors of the eight neighboring MBs. If the surrounding MBs are not available, the corresponding motion vectors are set to zero. Then, the MB from the previous frame at the spatial location specified by this motion vector is copied to the location of the missing MB in the current frame.

IV. Simulation Results

In this section, we experimentally determine the absolute and relative thresholds for each parameter for error detection and compare the performance of the proposed error detection and correction scheme with and without concealment for different noisy channels.

A. Threshold Calculation

As mentioned in Section III, to detect an erroneous MB, we compare the corresponding parameters with two thresholds, absolute and relative. The objective is to minimize the possibility of wrong detection, i.e., maximizing the detection of erroneous blocks and at the same time minimizing the detection of correct blocks as erroneous. An absolute threshold is the safest value of the parameter over which an MB can be considered as erroneous. The absolute thresholds were investigated over many frames of various sequences with different error patterns. For each of the parameters the absolute thresholds for luminance and chrominance (if applicable) components are determined separately and are fixed throughout the experiment. In order to determine an absolute threshold for a parameter x, we calculate that parameter for each MB under consideration, and check the maximum value of parameter x (say $x_{\max}$) that a non-erroneous MB may have. This means that any value of x greater than $x_{\max}$ will definitely correspond to an erroneous MB. As a safety measure the absolute threshold (T^{abs}) is chosen approximately 20% higher than $x_{\max}$.

The relative thresholds depend on the statistics of a frame. In this work, the relative threshold (T^{rel}) is defined as

$$T^{\text{rel}} = \mu + \eta\sigma \tag{7}$$

with

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{8}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{9}$$

where N is the number of MBs in a frame ($N = 99$ for QCIF), x is one of the previously defined parameter calculated for each MB, μ and σ are the mean and standard deviation of parameter x over a frame, respectively, and η is a constant integer which is determined empirically. If for an MB in a frame, the parameter x has a value greater than T^{rel} of that frame, it is considered as a candidate for an erroneous MB (subject to satisfying similar conditions with other parameters). Since μ and σ for a frame are fixed, T^{rel} is directly proportional to η, and the objective is to determine optimum value of η. For efficient detection, we measure the percentage of correctly detected erroneous blocks (blocks which are known to be erroneous and detected as erroneous) and wrongly detected erroneous blocks (blocks which are known to be correct but detected as erroneous) at each value of η. To be more precise, assume E is the number of known erroneous MBs, and D is the total number of detected erroneous MBs. D contains both correctly and wrongly detected erroneous MBs. Further, let C be the number of correctly detected MBs $(C \subseteq E)$, W is the number of wrongly detected MBs $(W \not\subset E)$ such that $D = C + W$. The parameter percentage of correctly detected error (PCD) and the percentage of wrongly detected error (PWD) are used to measure the performance and to determine an optimum value for η and are defined as follows:

$$\text{PCD} = \frac{C}{E} \times 100 \tag{10}$$

$$\text{PWD} = \frac{W}{E} \times 100. \tag{11}$$

It should be noted that PCD would always be less than 100% since $C \leq E$, whereas since W and E are independent from each other, PWD may have any value (sometimes more than

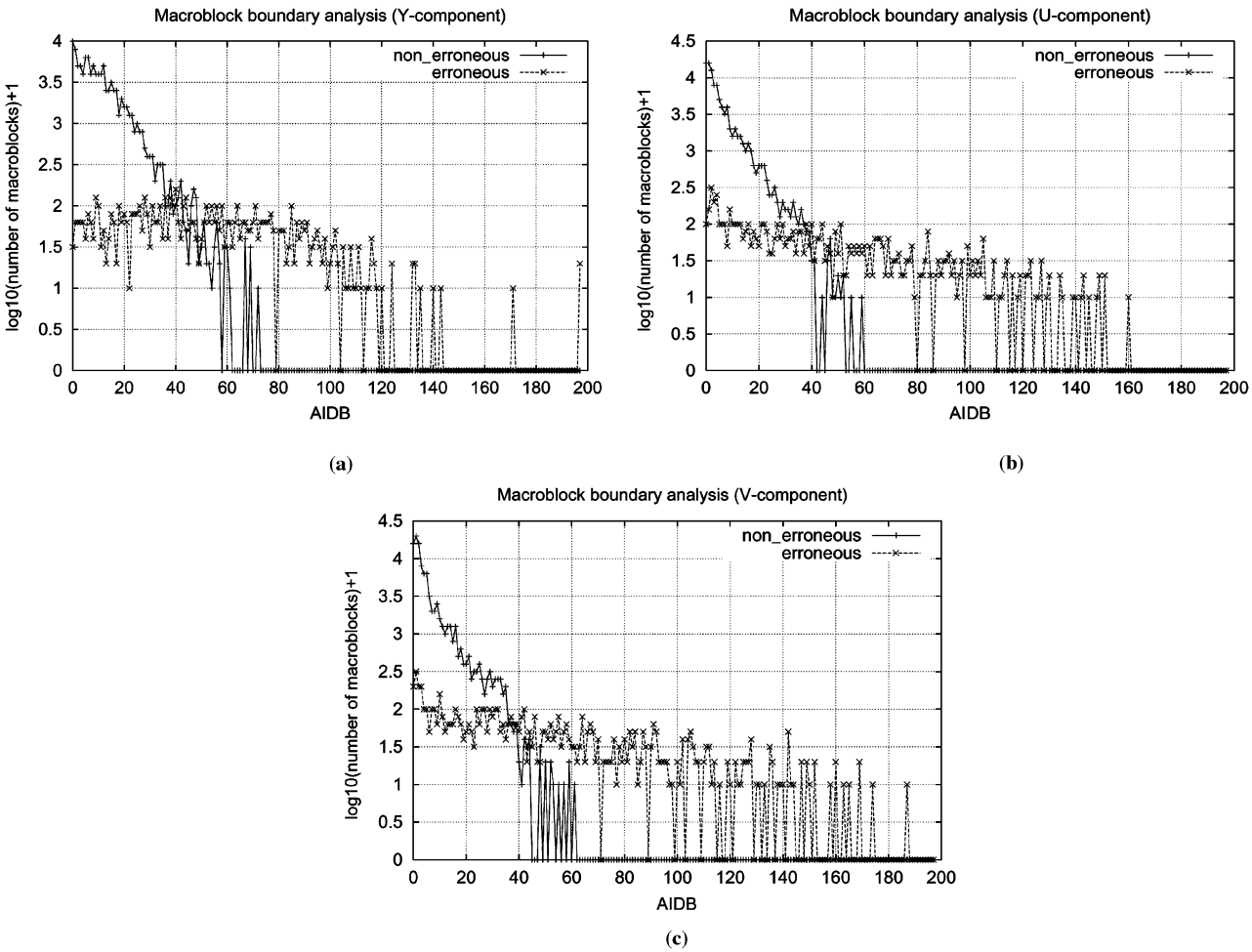

Fig. 8. MB boundary analysis. Distribution of number of MBs (y-axis) for different values of AIDB (x-axis). Nonerroneous MBs (solid lines) and erroneous MBs (dotted lines). (a) Y-components. (b) U-component. (c) V-component.

100%). Wherever possible, these parameters are determined separately for luminance and chrominance components.

In order to determine the absolute and relative thresholds, we have considered a set of 105 frames from 14 different QCIF sequences. Each frame includes between 1 to 38 erroneous MBs. In all the frames under consideration, there are a total of 10 395 MBs, out of which 667 MBs are erroneous. Some of the MBs are only slightly erroneous and have hardly visible artifacts. Since here the emphasis is only to detect visible errors, nonvisible erroneous MBs are considered as not detectable. In our experiment there were a total of 121 nonvisible erroneous MBs, and, hence, only 546 subjectively erroneous MBs remain to detect.

The absolute threshold for a parameter x, $x \in \{\mathrm{AIDB}, \mathrm{MPD}_8, \mathrm{IAIDB}\}$, is determined by plotting a histogram of the numbers of corrupted and noncorrupted MBs for each possible value of x of the luminance and chrominance (where applicable) components. These graphs are shown in Figs. 8–10 for AIDB, MPD_8 and IAIDB, respectively. In each graph the y-axis is the number of MBs (logarithmic representation shifted by 1) for each x value on the x-axis. The solid and dotted lines correspond to correct (nonerroneous) and corrupted MBs, respectively. It can be observed in Fig. 8(a) that all the noncorrupted MBs have values of AIDB less than 75. However, some corrupted MBs also have values less than 75. These are mostly nonvisible erroneous MBs. If we take a threshold less than 75, then many correct MBs will be detected as erroneous. However, most of the visibly erroneous MBs can be detected if we take any threshold above 75. To be on the safer side, the absolute threshold for AIDB is fixed as 100 for luminance 75 for chrominance components. Similarly, observing Fig. 9(a), there is no value of MPD_8 for the luminance component, which can be used as a threshold such that no correct MB is detected as erroneous. However, for chrominance components as shown in Fig. 9(b) and (c), a value of MPD_8 equal to 100 can be chosen safely as a threshold. Since the inner boundary analysis is applicable only for luminance components, Fig. 10 shows the histogram of IAIDB for luminance only. From the graph it can be seen that for inner boundary analysis, the absolute threshold of 100 can be chosen safely.

Next is the calculation of the relative thresholds for AIDB, MPD_n, IAIDB, $\mathrm{IAIDB}_{\mathrm{block}}$, and $\mathrm{MPD}_3^{\mathrm{block}}$. The AIDB, MPD_8 and IAIDB are used to detect the erroneous MBs in both first and second passes. However, $\mathrm{IAIDB}_{\mathrm{block}}$ and $\mathrm{MPD}_3^{\mathrm{block}}$ detect MBs with only one or two erroneous DCT blocks. Also, MPD_4 and $\mathrm{AIDB}(M:N,S,W)$ are used to detect the first erroneous MBs in the slice. In this work, we determine the relative thresholds for AIDB, MPD_8 and IAIDB individually (η for each), but relative thresholds for ($\mathrm{IAIDB}_{\mathrm{block}}$ and $\mathrm{MPD}_3^{\mathrm{block}}$) and [$\mathrm{MPD}_4$ and $\mathrm{AIDB}(M:N,S,W)$] are determined jointly.

In order to determine an optimal value for η for a parameter x, we calculate x for each MB within a frame and compare its

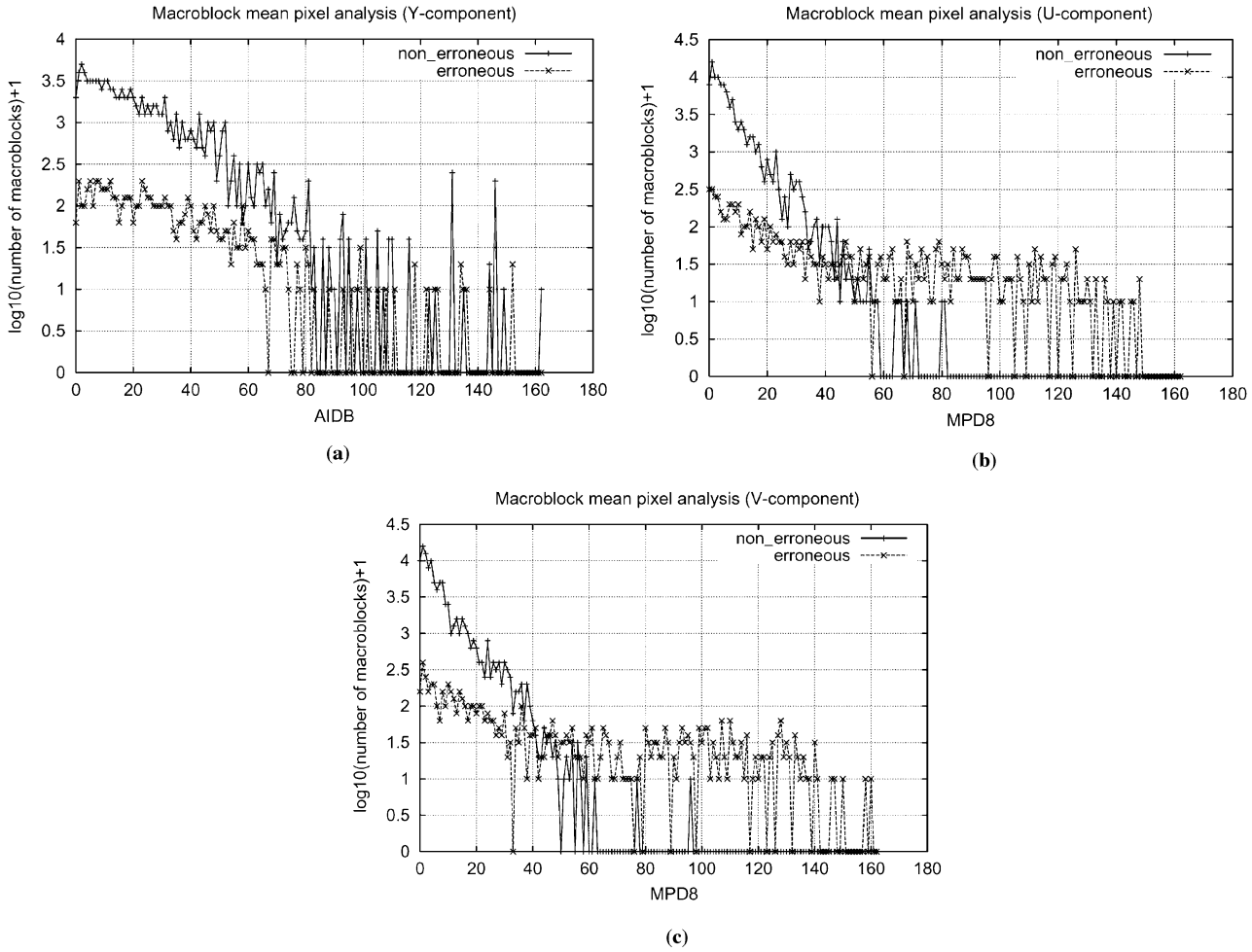

Fig. 9. MB mean pixel value analysis. Distribution of number of MBs (y-axis) for different values of MPD (x-axis). Nonerroneous MBs (solid lines) and erroneous MBs (dotted lines). (a) Y-components. (b) U-component. (c) V-component.

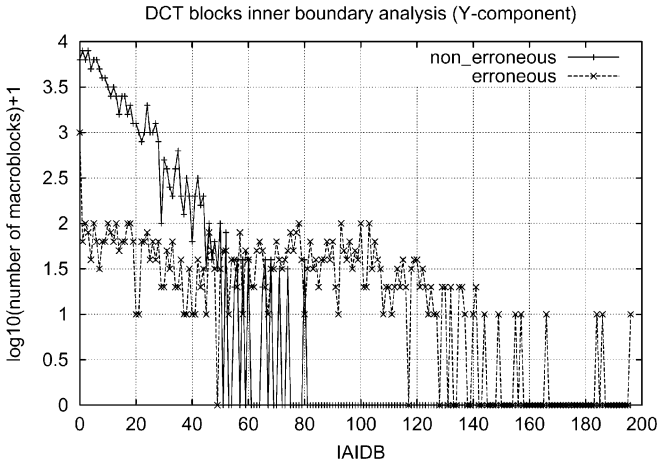

Fig. 10. DCT blocks inner boundary analysis. Distribution of number of MBs (y-axis) for different values of IAIDB (x-axis). Nonerroneous MBs (solid lines) and erroneous MBs (dotted lines).

value with $T^{\mathrm{rel}} = \mu + \eta\sigma$ for that frame at a fixed η. If for an MB, $x > T^{\mathrm{rel}}$, that MB is considered as erroneous, and is then compared with our known database of erroneous MBs to identify that whether this MB is counted as correct (C) or wrong (W). This process is repeated for all the 105 frames under consideration for the same value of η. The PCD and PCW are then calculated from the total counted C and W (for all frames), respectively with a known value of erroneous MBs (E) (in our experiment $E = 546$). The entire process is repeated for various values of $\eta \in \{1, 2, \ldots, 6\}$ and for luminance and chrominance components separately. The PCD and PWD against η for AIDB, MPD_8, and IAIDB are plotted in Fig. 11(a)–(c), respectively. As expected, at lower values of η, there are more wrongly detected erroneous MBs than correctly detected erroneous MBs ($W > C$). As the threshold (or η) increases, the number of correctly detected erroneous MBs decreases, however the number of wrongly detected erroneous MBs decreases much faster than that of correctly detected erroneous MBs. The best value of η for a high PCD and small PWD is between 3 and 5. From the graphs in Fig. 11, it can be observed that $\eta = 4$ (for AIDB), $\eta = 4$ (for MPD_8), and $\eta = 3$ (for IAIDB) are the best choices. These values are used to detect the erroneous blocks in the first pass, whereas in the second pass, each of these is reduced by one. Since at these values of η, although there are very few wrongly detected MBs, but only 20%–30% of erroneous MBs is detected, the relative threshold of these parameters individually is not sufficient for error detection. However, if they are used together, the detection efficiency can be improved drastically.

There are some cases in which the three thresholds determined so far may not be sufficient to detect errors. One of such case is when an MB contains only one or two erroneous DCT blocks. In order to detect such MBs, we calculate the individual

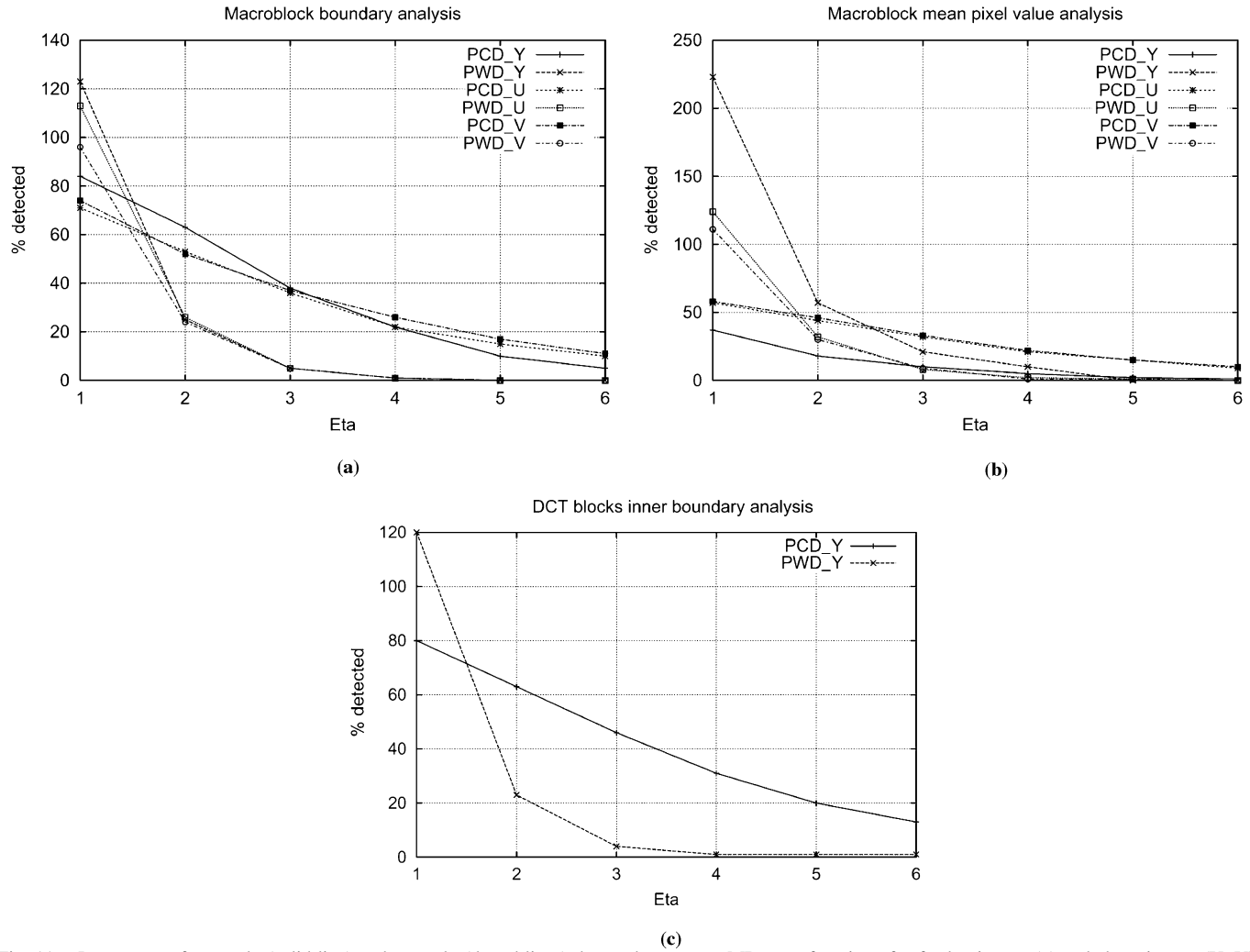

Fig. 11. Percentage of correctly (solid line) and wrongly (dotted lines) detected erroneous MBs as a function of η for luminance (y) and chrominance (U, V) components. (a) MB boundary analysis. (b) MB mean pixel value analysis. (c) DCT blocks inner boundary analysis.

DCT block boundary parameter $(\mathrm{IAIDB_{block}})$ and the DCT block mean pixel parameter $\left(\mathrm{MPD}_3^{\mathrm{block}}\right)$. An MB is considered erroneous if both of these parameters are larger than their respective relative thresholds. Through computer simulations, we obtained that (4, 3) pair of relative thresholds gives minimum number of wrongly and maximum number of correctly detected erroneous MBs. Since our correction technique starts from the first erroneous MB, it is very important that it should be detected correctly. Normally, the first erroneous MB in a slice is only slightly erroneous (they mainly belong to background), it is very likely that it may not be detected by any of the methods discussed above. For this purpose, we analyze the MBs preceding the first detected so far, with the parameters $\mathrm{AIDB}(M:N,S,W)$ and MPD_4. Again if both of these parameters for an MB are greater than their respective relative thresholds, the MB is considered as erroneous. The computer simulations suggest that the pair of $(2, 2)$ for η detect maximum number of first erroneous MBs in a slice correctly. The relative and absolute thresholds of all the parameters are summarized in Table I.

TABLE I
VALUES OF ABSOLUTE AND RELATIVE THRESHOLDS USED FOR ERROR DETECTION

Parameter 'x'	Absolute threshold		Relative threshold (η)	
	luminance	Chrominance	luminance	chrominance
AIDB(M:N, S, W, E)	100	75	4	4
MPD_8	-	100	4	4
IAIDB	100	-	3	-
$\mathrm{IAIDB_{block}}$ & $\mathrm{MPD_3}^{\mathrm{block}}$	-	-	(4, 3)	-
AIDB (M:N, S, W) & MPD_4	-	-	(2, 2)	-

B. Performance of the Algorithm

In order to evaluate the performance of our error detection and correction technique, we have considered one frame of the Salesman sequence decoded from a 64-kb/s H.263-coded bit stream, shown in Fig. 12(a). The errors are artificially introduced in the bit stream, and the erroneously decoded frame is shown in Fig. 12(b). The error detection and correction algorithm is recursively applied to this erroneous frame, and decoded images at each step are given in Fig. 12(c)–(f). By comparing these frames, it can be seen that applying error detection and correction each time improved the quality of the decoded frame. After four iterations, the detector no longer detects any additional erroneous MBs except the inserted gray MBs, and

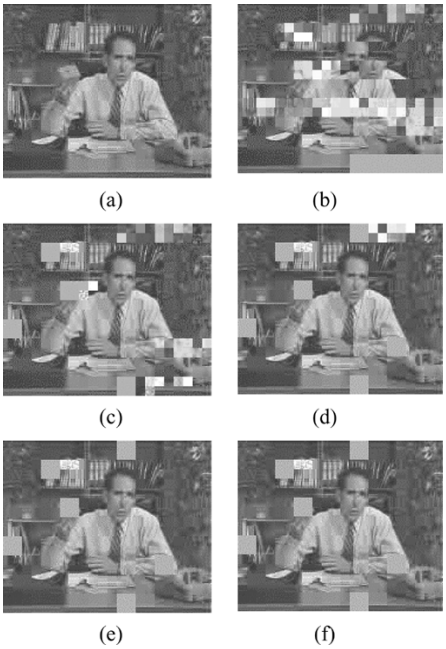

Fig. 12. Results of recursively applying error detection and step-by-step decoding based correction. (a) Nonerroneous frame. (b) erroneous frame. (c)–(f) Reconstructed frame after recusive error detection and correction at each step.

hence the program will terminate. As can be seen from Fig. 12(f) each erroneous slice has now only one erroneous block, which can be concealed better than that of Fig. 12(b). Thus, our proposed technique is highly efficient in restricting the erroneous area.

C. Performance Over Binary Symmetric Channel

To test the performance of our technique over some lossy channels, we have considered simple binary symmetric channel (BSC). This is because for a given BER, the BSC model is actually the worst model yielding the highest probability of error in the slice as compared to channels with correlated errors (burst errors). For our experiment we have considered 2 QCIF (176×144) video sequences in 4:2:0 YUV format, namely Claire and Salesman, at temporal resolution of 10 f /s, coded at 64 kb/s and transmitted over BSC with different BER. It should be noted that Claire is a simple sequence with only lips and head movement, however, Salesman is relatively complex sequence, having complex texture with more motion. It is assumed here that the picture and slice headers including LQUANT and LMVV are not corrupted. To provide statistically significant results, all the experiments are performed for 100 independent channel conditions. The overall PSNR of a frame is measured as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{(\mathrm{MSE}(Y)+\mathrm{MSE}(U)+\mathrm{MSE}(V))}{3}} \qquad (12)$$

where $\mathrm{MSE}(X)$, $X \in \{Y,U,V\}$ is the mean-squared error of component X averaged over all independent channel realizations. The overall video quality is also measured in terms of PSNR, with each MSE is averaged over all frames and all channel conditions. In all these experiments, the first frame is coded as I-frame and the subsequent frames are coded as P-frames. The effects of channel errors are analyzed separately for I- and P-frames.

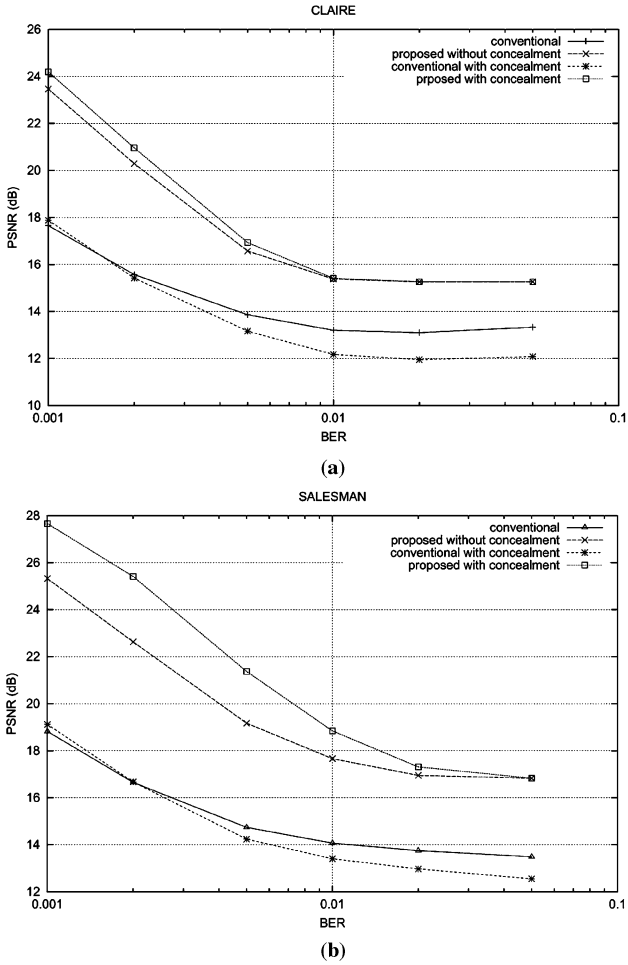

Fig. 13. Performance comparison of proposed and conventional methods with and without spatial concealment for first (intra) frame at different BER for sequence (a) Claire and (b) Salesman.

In the first experiment, we have considered first frame of each sequence, which are coded as I-frame with quantization scale ($q = 13$, default in TMN coder) and are transmitted over BSC at six different BER ($10^{-3}, 2\times10^{-3}, 5\times10^{-3}, 10^{-2}, 2\times10^{-2}$, and 5×10^{-2}). The overall PSNR for the conventional and our proposed decoder, each with and without concealment at different BER are plotted in Fig. 13(a) and (b) for Claire and Salesman sequences, respectively. It should be noted here that since the encoder uses the same quantizer, the number of bits generated for I-frames varies from sequence to sequence (approximately 17 K for Claire and 18 K for Salesman), and, hence, at the given BER, the number of erroneous bits are different. Comparing the results, it can be observed that our proposed error detection and correction without concealment results in 2–5 dB improvement for Claire and 3.5–6 dB for the Salesman sequence over the conventional H.263 decoder. Further, it can be observed that use of spatial concealment with conventional decoder degrades the picture quality, since the use of erroneous MBs for concealment causes errors to spread. Also, some nonerroneous MBs may be concealed, which degrades the performance [23]. Since

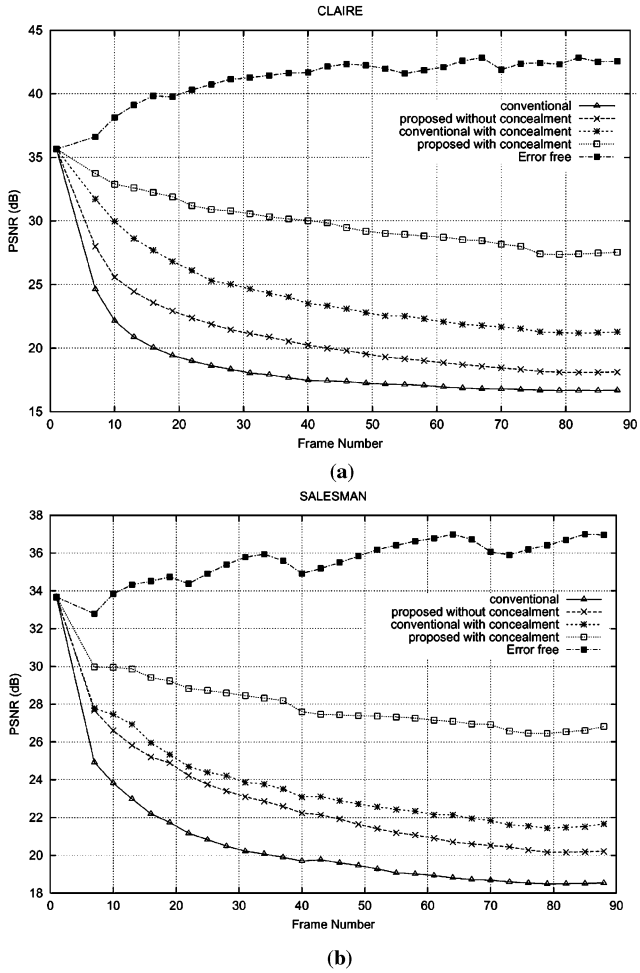

Fig. 14. Frame-by-frame performance comparison of the proposed method with conventional with and without concealment for inter (P) frames transmitted over BSC with $\mathrm{BER} = 10^{-3}$ for sequence (a) Claire and (b) Salesman.

our method limits the error propagation to a few MBs only, concealment further improves the results by up to 0.6 dB for Claire and up to 2 dB for Salesman sequence.

To evaluate the performance for P-frames, we have coded the entire video sequence at 64 kb/s with first frame as intra (I) and the remaining as P-frames. The I-frame is transmitted error free while all the other frames are transmitted over lossy BSC with $\mathrm{BER} = \{2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$. Fig. 14(a) and (b) show the frame-by-frame comparison of the overall PSNR of our method with the conventional one (both with and without temporal concealment) for the two test sequences at the $\mathrm{BER} = 10^{-3}$, for Claire and Salesman, respectively. From these graphs, it is apparent that our proposed method consistently gives improvement over the conventional decoder. However, the relative improvement with concealment is more than that without concealment. This can be attributed to the fact that the isolated grayed (perhaps erroneous) MBs in our method can suitably be concealed. Further, it should be noted that apart from iterative decoding, our method utilizes only up to 1.2% additional bit rate (due to the addition of LMVV and LQUANT in slice header), and thus under no-error conditions, frames can be reconstructed without any significant loss in the quality. However, under the lossy channel conditions, some of the MBs are shaded with gray color or concealed, which results in the loss of subjective and objective quality.

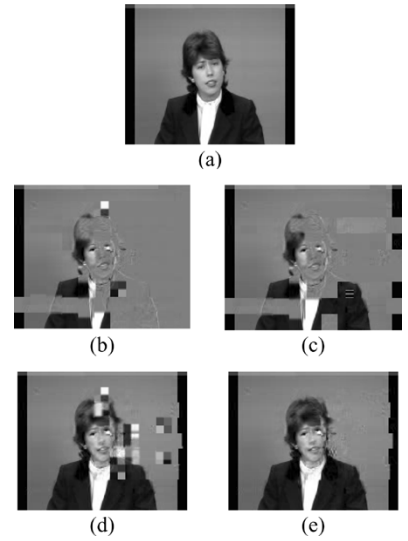

Fig. 15. Claire sequence (frame 88) coded at 64 kb/s with $\mathrm{BER} = 10^{-3}$. (a) without error. (b) Conventional without concealment. (c) Proposed without concealment. (d) Conventional with concealment. (e) Proposed with concealment.

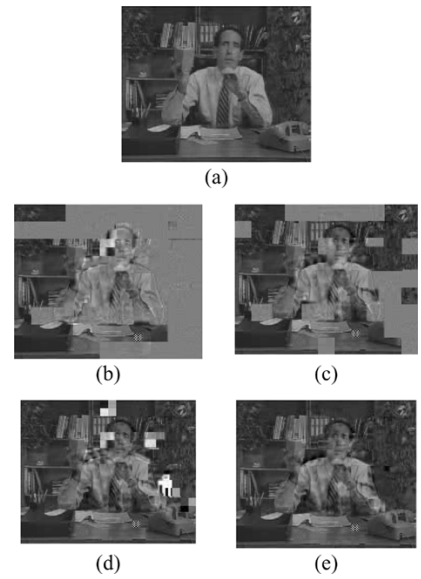

Fig. 16. Salesman sequence (frame 88) coded at 64 kb/s with $\mathrm{BER} = 10^{-3}$. (a) Without error. (b) Conventional without concealment. (c) Proposed without concealment. (d) Conventional with concealment. (e) Proposed with concealment.

Figs. 15 and 16 show the visual comparison of the 88th frame of each sequence (including accumulation of errors). In each of these figures, the top one is the corresponding decoded frame under a no-error condition, and the remaining four are the decoded frames at $\mathrm{BER} = 10^{-3}$. In each case, the left column corresponds to the conventional decoder and the right column to the proposed decoder; the middle row is without concealment and bottom row is with concealment. In Fig. 15, by comparing (b) and (c), it can be seen that the proposed method corrects many of the erroneous MBs, which are visible in (b) but corrected in (c).

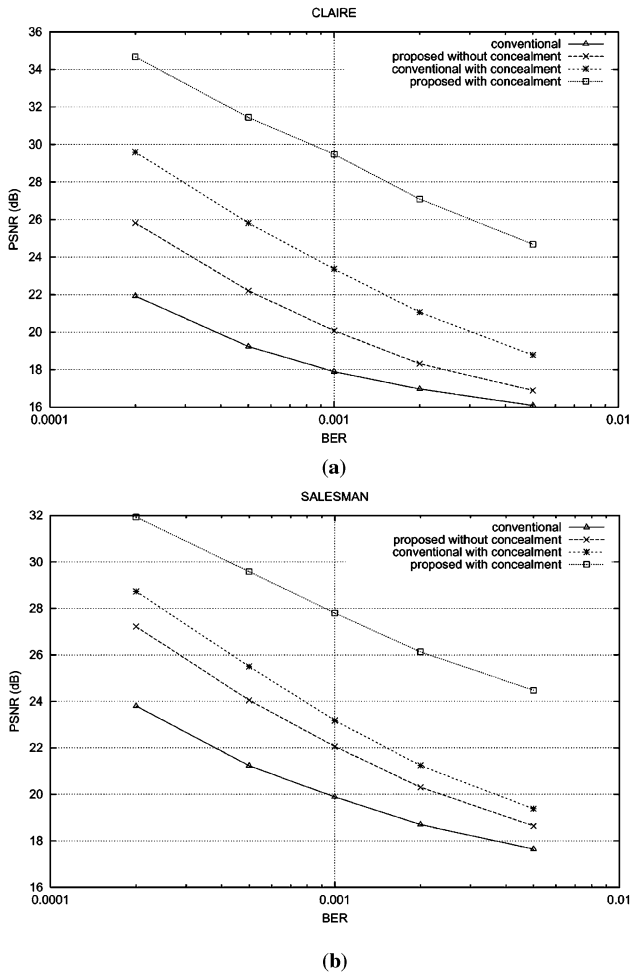

Fig. 17. Performance comparison of proposed and conventional methods with and without spatial concealment for P-frame over BSC at different BER for sequence (a) Claire and (b) Salesman.

Furthermore, comparing (b) with (d), it can be observed that although concealment corrects many of the MBs around the body of Claire, during the concealment some of the corrupted MBs from the previous frame are also added. However, the concealment with our method shown in (e) does not suffer from such problems and can correct most of the MBs. The small distortion around the right side of the face including the right eye is due to the fact that motion vectors estimation in the temporal concealment is never perfect. Similarly, in Fig. 16 for the 88th frame of Salesman, our method corrects many of the erroneous MBs and results in mostly isolated grayed MBs that can better be concealed.

Finally, Fig. 17 illustrates the comparison of resulting overall video quality of the two test sequences for baseline H.263, data-partitioned (DP) mode in H.263 and proposed methods (each with and without concealment) at various BERs. The overall video quality is represented in terms of PSNR defined in (12), where each $\mathrm{MSE}(X)$, $X \in \{Y, U, V\}$ is averaged over all frames and all 100 independent channel conditions. It is worth mentioning here that our method consistently outperforms the conventional one (baseline H.263) at all BERs, but relative improvement is higher at lower BERs than higher BERs. When compared with data-partitioned (DP) mode, which uses RVLC for motion vectors, our method is inferior if no concealment is used, but with concealment our method outperforms by 2.3–4 dB and 0.8–2.5 dB for Claire and Salesman sequences, respectively. The reason for this can be explained as follows. Since in DP mode of H.263, only DCT coefficients are lost and any error in MV is most likely to be corrected with RVLC, the residual error is far better than our method, which relies on error detection at frame level. However, with concealment, in DP mode with RVLC since DCT coefficients are lost after error, although original MVs are used for concealment, the lost DCT coefficients cannot be recovered. But in our case, iterative error correction algorithm recovers the original data for most of the MBs except the one in which error occurs, despite of not using the original MVs, the use of concealment technique in our method gives better results. It is interesting to note that the temporal concealment improves the results in each case, but relative PSNR improvement of our method over conventional and RVLC method is greater with the concealment than without.

V. Conclusion

In this paper, we have proposed a novel error detection and correction technique for H.263 coded video over BSC channels. We have suggested and analyzed the criteria for detecting erroneous MBs in a decoded frame and found that these criteria are sufficient to detect most of the erroneous MBs. The information about each MB, whether erroneous or not, along with the received bit stream are then used in a step-by-step decoding-based correction. The simulation results show that iterative use of our technique can recover frames from the corresponding corrupted frames without using any reverse channel or channel coding for BER up to 0.005–0.01 over BSC. Since proposed method is used at the decoder, no significant loss of quality is observed under no-error condition, which otherwise is very common with channel coding. The spatial and temporal concealment can better be utilized to further improve the quality with our method. Since the error detection is performed at frame level, the quality of decoded image is a constraint on its performance. Thus, the channel BER and transmission rate will affect the error detection. Further, since correction and detection are applied iteratively at slice level, the decoding time is much larger compared to the conventional decoder. The decoding time increases with the increase in channel BER.

References

[1] "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263 version 2, 1998.

[2] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 849–866, Nov. 1998.

[3] S. Wenger, G. Knorr, J. Oti, and F. Kossentini, "Error resilience support in H.263+," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 867–877, Nov. 1998.

[4] R. Talluri, "Error resilient video coding in the MPEG-4 standard," *IEEE Commun. Mag.*, vol. 36, pp. 112–119, June 1998.

[5] E. Steinbach, N. Fraber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 872–881, Dec. 1997.

[6] M. Budagvi and J. D. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Processing*, vol. 10, pp. 252–265, Feb. 2001.

[7] C.-S. Kim, R.-H. Kim, and S.-U. Lee, "Robust transmission of video sequence using double-vector motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1011–1021, Sept. 2001.

[8] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 97, pp. 1707–1723, Oct. 1999.
[9] P.-C. Change and T.-H. Lee, "Precise and fast error tracking for error resilient transmission of H.263 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 600–607, June 2000.
[10] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Area Commun.*, vol. 18, pp. 1012–1032, June 2000.
[11] D. W. Redmill and N. G. Kingsbury, "The EREC: An error resilient technique for coding variable length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 565–574, Apr. 1996.
[12] H. S. Jung, R.-C. Kim, and S.-U. Lee, "An hierarchical synchronization technique based on EREC for robust transmission of H.263 bit stream," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 433–438, Apr. 2000.
[13] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974–996, May 1998.
[14] S. Kallel and D. Haccoun, "Generalized type II hybrid ARQ with punctured convolutional coding," *IEEE Trans. Commun.*, vol. 38, pp. 1938–1946, Nov. 1990.
[15] M. Bystrom, V. Parthasarthy, and J. W. Modestino, "Hybrid error concealment scheme for broadcast video transmission over ATM network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 868–881, Sept. 1999.
[16] S. S. Hemami, "Robust image transmission using resynchronizing variable length codes and error concealment," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 927–939, June 2000.
[17] Y.-H. Han and J.-J. Leou, "Detection and correction of transmission errors in JPEG images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 221–231, Apr. 1998.
[18] M. Abdat *et al.*, "Transmission error detection and concealment in JPEG images," *Signal Process. Image Commun.*, vol. 13, pp. 45–64, July 1998.
[19] F. G. B. De Natale, C. Perra, and G. Vernazza, "DCT information recovery of erroneous image blocks by a neural network predictor," *IEEE J. Select. Areas Commun.*, vol. 18, June 2000.
[20] J. Wen and J. D. Villasenor, "Reversible variable-length codes for robust image and video transmission," in *Proc. 31st Asilomar Conf., Signal, Systems and Computers*, vol. 2, Nov. 1997, pp. 973–979.
[21] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, pp. 1544–1551, Oct. 1993.
[22] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 238–247, Mar. 1993.
[23] C. P. Lim, E. A. W. Tan, S. Ghanbari, and M. Ghanbari, "Cell loss concealment and packetization in packet video," *Int. J. Imaging Syst. Technol.*, vol. 10, no. 1, pp. 54–58, 1999.

Ekram Khan received the B.S. and M.Sc. degrees in electronic engineering from the Z. H. College of Engineering and Technology, Aligarh Muslim University, Aligarh, India, in 1991 and 1994, respectively. He received the Ph.D. degree in video coding and transmission from University of Essex, Colchester, U.K., in 2003.

He joined the Department of Electronics Engineering, Aligarh Muslim University, in 1993 as a Lecturer and is currently working as a Senior Lecturer in the same department. He has a large number of publications to his credit and has authored and coauthored over 35 papers in academic journals and national/international conference proceedings. His areas of research are image/video coding, video transmission over wireless and IP networks, and video segmentation and tracking.

Dr. Khan was a recipient of the prestigious Commonwealth Scholarship to pursue his Ph.D. from University of Essex and is a life member of IETE, India.

Stefan Lehmann received the M.Sc. degree in electrical engineering from the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, in 2002.

During his Ph.D. studies, he spent some time at the Ecole Polytechnique de Montreal, Montreal, Canada, and the University of Essex, Colchester, U.K. He is currently with the Audio and Video Networking Reearch Laboratory, Department of Electronic Systems Engineering, University of Essex.

Hiroshi Gunji received the B.S. and M.S. degrees from Waseda University, Tokyo, Japan, in 1990 and 1992, respectively.

He joined Central Research Laboratory, Hitachi Ltd., Tokyo, in 1992. Since 1995, he was engaged in the research and development for image and audio compression at Semiconductor and Integrated Circuits Division, Hitachi Ltd. From 2000 to 2001, he investigated error resilience on compressed video in the Department of Electronic Systems Engineering, University of Essex, Colchester, U.K. He is currently developing multimedia application software on mobile phones in the Renesas Technology Corporation, Tokyo, Japan.

Mohammed Ghanbari (M'78–SM'97–F01) is a Professor of video networking in the Department of Electronic Systems Engineering, University of Essex, Colchester, U.K.

He is best known for the pioneering work on two-layer video coding for ATM networks, now known as SNR scalability in the standard video codecs. He has registered for 11 international patents and published more than 290 technical papers on various aspects of video networking. He was also a coinvestigator of the European MOSAIC project studying the subjective assessment of picture quality, which resulted to ITU-R Recommendation 500. He is the coauthor of *Principles of Performance Engineering* (Stevenage, U.K.: IEE Press, 1999) and the author of *Video coding: An Introduction To Standard Codecs* (Stevenage, U.K.: IEE Press, 1997) which received the Rayleigh prize as the best book of year 2000 by Institute of Electrical Engineers (IEE), U.K. His recent book is *Standard Codecs: Image Compression to Advanced Video Coding* (Stevenage, U.K.: IEE Press, 2003).

Dr. Ghanbari was the corecipient of the A. H. Reeves Prize for the best paper published in the 1995 *Proceedings of Inst. Elect. Eng.* in the theme of digital coding. He was the General Chair of the 1997 International Workshop on Packet Video, and a Guest Editor of the 1997 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Special Issue on Multimedia Technology and Applications. He has served as Associate Editor to IEEE TRANSACTIONS ON MULTIMEDIA (IEEE-T-MM) and represented the University of Essex as one of the six U.K. academic partners in the Virtual Centre of Excellence in Digital Broadcasting and Multimedia. He is a Fellow of IEE, U.K., and a Chartered Engineer (C.Eng.).

scheduling with temporal distant constraints. NP-completeness results and polynomial time algorithms for a restricted instance of the problem are given in [11]. In spite of the similarities to the current problem, the approximation results for single machine scheduling problems cannot be applied to AV object scheduling because of an entirely different problem domain and additional constraints on AV object scheduling. Approximation algorithms are based on heuristics and domain knowledge is essential to develop good designs. Even though the results of single-machine scheduling are not directly applicable to AV presentations, some of the results can be used in scheduling individual objects on a channel. The results of single-machine scheduling problems as formulated by Lawler in [19], [20] may be used to determine the schedulablity of individual objects.

C. Complexity of Audio Visual Object Scheduling

Theorem 1: Scheduling of an AU in AV presentations (SAV) is NP-complete in the strong sense.

Proof: We prove this by transforming the problem of sequencing within intervals (SWI), proven to be NP-complete in the strong sense [8].

We restate SWI below:

Instance: A finite set T of tasks and, for each $t \in T$, an integer release time $r(t) \geq 0$, a deadline $d(t) \in Z^+$, and a length $l(t) \in Z^+$.

Question: Does there exist a feasible schedule for T, i.e., a function $\sigma : T \rightarrow Z^+$, such that for each $t \in T$, $\sigma(t) \geq r(t)$, $\sigma(t) + l(t) \leq d(t)$, and if $t' \in T - \{t\}$, then either $\sigma(t') + l(t') \leq \sigma(t)$ or $\sigma(t') \geq \sigma(t) + l(t)$?

The basic units of the SWI problem are the tasks $t \in T$. The local replacement for each $t \in T$ is a single AU $A_j(k)$ with $r(A_j(k)) \geq T_j^s(k-1)$, $d(t) = T_j^d(k)$, $l(t) = d_j(k)$. We disregard the buffer and startup delay constraints. It is easy to see that this instance can be created from SWI in polynomial time. Since SWI can be transformed to SAV, SAV is at least as hard as SWI.

Since SAV is NP-complete in the strong sense, it cannot be solved by a pseudopolynomial-time algorithm. We present several polynomial-time algorithms based on heuristics and constraint relaxation and evaluate their performance with respect to speed and efficiency.

III. Bounds on Startup Delay and Terminal Buffer

An MPEG-4 terminal has a finite buffer to store the received data until they are decoded. The amount of buffer capacity required depends on the type and number of elementary streams being buffered. Since there are usually no limits on the number of objects in AV presentations, it is not practical to have sufficient buffer for *all* presentations. A terminal should be designed to support a *class* of presentations. The amount of buffer available also determines the upper bound on the startup delay for a session. The higher the startup delay, the higher the buffer capacity required (with channel capacity remaining the same). When scheduling presentations, a scheduler should assume the minimum allowable buffer for terminals in order to support all terminal types. Even though the knowledge of the buffer occupancy at a terminal may help improve the schedule, it makes the schedule dependent on the buffer model used by the terminals. Since the buffer model and management in a terminal depends on terminal design, we designed the scheduler to be buffer model independent.

Startup delay can be defined as the time a user has to wait from the time a request is made until the time the presentation starts. A startup delay of T_s is *not* equal to buffering T_s seconds of the presentation. The amount of startup delay varies from presentation to presentation and even for the same presentation, it may vary with varying resources (e.g, bandwidth and buffer). Startup delay can be viewed as preloading the beginning of a presentation so that the presentation is played back continuously once the playback starts. The amount of startup delay required for the smooth playback of a presentation depends on the channel capacity. For any channel, the minimum startup delay is the time needed to transmit (buffer) AUs that are presented at time 0 (AUs with timestamp 0).

Consider a presentation composed of several images displayed on the first screen, followed by an audio track. The images to be displayed on the first screen should reach the terminal before the presentation starts, resulting in a startup delay. If the channel bandwidth reserved for the presentation is allocated based on the low bitrate audio stream that follows the images, the startup delay will be higher. On the other hand, if the higher bandwidth is reserved to minimize the startup delay, the capacity may be wasted during the remainder of the presentation when low bitrate audio is delivered. The tradeoff depends on resource availability and startup-delay tolerance of the application.

Given a startup delay T_s, the buffer required is equal to the size of the objects that can be loaded (transmitted to the client) in time T_s. The minimum buffer required for this delay is $T_s^* C$. The minimum startup delay for any presentation is equal to the time required to transmit (load) the objects/instances to be displayed at time 0. We refer to this time as T_s^0. T_s^0 is the optimal startup delay for startup delay-optimal schedules and is the lower bound on startup delay for bandwidth-optimal schedules.

A. Residual Data Volume

We introduce the notion of *data volume* to quickly compute the minimum startup delays needed for a presentation and determine the nonschedulability. Data volume (V_d) is the amount of data (in bits) transferred during a session. The amount of data that can be carried by a channel during a session is the data pipe volume ($V_p = C * D_p$). The amount of data volume exceeding the data pipe volume is the residual data volume ($V_{\text{res}} = V_d - V_p$). A positive V_{res} gives the lower bound on the amount of data to be loaded during startup and hence determines the lower bound on the startup delay for a presentation. A negative value of V_{res} indicates unused channel capacity during the session. We prove the lower bound on channel capacity required in Theorem 2.

Theorem 2: For a presentation of duration D_p, the lower bound on channel capacity required for a startup delay-optimal schedule is

$$C \geq C_{\min}, \quad \text{where } C_{\min} = \frac{V_d}{D_p}$$

and the bound is tight.

Proof:

$$V_d = \sum_{j,k} s_j(k)$$
$$D_p = \max_j \left\{T_j^d(n_j)\right\} - \min_j \left\{T_j^d(n_j)\right\}.$$

For a presentation of length D_p, the data pipe volume at the given pipe capacity is

$$V_p = C * D_p.$$

Assuming that the buffers are filled up at a rate C, the startup delay due to V_{res} is

$$T_s^{\text{res}} = \frac{(V_d - V_p)}{C}.$$

To minimize the startup delay

$$T_s^{\text{res}} = \frac{(V_d - V_p)}{C} = 0 \Rightarrow V_p = V_d.$$

Since $V_p = C * D_p$, substituting V_p we get the lower bound on the channel capacity

$$C_{\min} = \frac{V_d}{D_p}.$$

From constraint (1)

$$T_j^s(1) \le T_j^d(1) - d_j(1) \Rightarrow T_j^s(1) \le T_j^d(1) - \frac{s_j(1)}{C}.$$

From constraint (2)

$$T_j^s(1) \ge T_j^s(0) + \frac{s_j(0)}{C}.$$

Assuming that the presentation starts at time 0

$$T_j^s(0) = 0, \quad \Rightarrow T_j^s(1) \ge \frac{s_j(0)}{C}.$$

From constraints (1) and (2)

$$T_j^s(2) \ge T_j^s(1) + \frac{s_j(1)}{C} \Rightarrow T_j^s(2) \ge \frac{s_j(0)}{C} + \frac{s_j(1)}{C}$$
$$T_j^s(2) \le T_j^d(2) - \frac{s_j(2)}{C} \Rightarrow T_j^d(2) \ge T_j^s(2) + \frac{s_j(2)}{C}$$
$$\Rightarrow T_j^d(2) \ge \frac{s_j(0)}{C} + \frac{s_j(1)}{C} + \frac{s_j(2)}{C}.$$

Similarly

$$T_j^d(n_j) \ge \frac{s_j(0)}{C} + \frac{s_j(1)}{C} + \ldots + \frac{s_j(n_j)}{C}.$$

Since the AUs are transmitted on a single channel

$$D_p = \max\left\{T_j^d(n_j)\right\} \ge \sum_j \frac{s_j(0)}{C} + \frac{s_j(1)}{C} + \ldots + \frac{s_j(n_j)}{C}$$
$$D_p \ge \frac{1}{C}\sum_{j,k} s_j(k) \Rightarrow D_p \ge \frac{1}{C} * C_{\min} * D_p \Rightarrow C \ge C_{\min}.$$

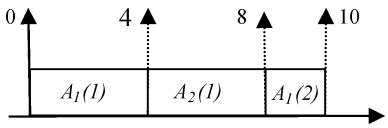

Fig. 3. Example to prove the tightness of the bound.

We can show that the bound is tight by considering the example as shown in Fig. 3.

Object 1 has two AUs and object 2 has one AU. The decoding times and sizes of AUs in bytes are

$$T_1^d(1) = 4, \quad T_1^d(2) = 10, \quad T_2^d(1) = 8,$$
$$s_1(1) = 10, \quad s_1(2) = 5, \quad s_2(1) = 10.$$

With these values, the send times and channel capacity are $T_1^s(1) = 0$, $T_1^s(2) = 8$, $T_2^s(1) = 4$ and $C = C_{\min} = 2.5$ bytes/s.

The actual channel capacity required to minimize startup delay may be higher depending on the timing constraints of AUs. Note that irrespective of the channel capacity, the minimum startup delay remains nonzero and is equal to T_s^0.

Thus, for any given presentation with resource constraints, the minimum startup delay is $T_s^{\min} = \max\{T_s^{\text{res}}, T_s^0\}$, the minimum buffer capacity required is $B_{\min} = T_s^{\min} * C$, and the presentation is schedulable only if the available buffer is at least equal to $B_{\min}$.

IV. Scheduling Algorithms

In this section we describe a family of scheduling algorithms for AV object scheduling. Given an AV presentation, the scheduling algorithms compute a delivery schedule according to the selected criteria. We assume that the terminal buffer is fixed and compute startup delay-optimal or bandwidth-minimizing schedules. The algorithms can also be repurposed to compute the minimum terminal buffer required for a given channel capacity.

A. FullSched Algorithm

This algorithm is based on the last-to-first idea mentioned in [20] for scheduling jobs on a single machine. The main principle behind this algorithm is scheduling an AU with latest deadline first and scheduling it as close to the deadline as possible. The algorithm computes the schedule starting with an AU with the latest decoding time in the presentation. This algorithm computes the schedule, the required startup delay, and any channel idle times. The channel idle times computed are used in the gap-scheduling algorithm described in Section IV-B.

Let S be the set of current AUs to be scheduled. Initialize S to contain the last AU of each of the objects to be scheduled. Let x_j be the index of the next AU of object j to be scheduled

$$x_j = n_j, \quad 1 \le j \le N.$$

Initialize $S = \{A_j(x_j)\}$, $1 \le j \le N$. $S(j)$ is the AU of object j to be scheduled next. S contains at most one AU for every

object j. G is the set of channel idle times. Idle time is given by a tuple $\langle t, d\rangle$, i.e, the channel is idle for duration d starting at time t. Initialize $G = \{\phi\}$. Set current time $i = \infty$. Sort AU of objects in the decreasing order of their decoding times.

```
BEGIN
  while (S ≠ φ) {
    i = min{i, max{T_j^d(k)}},  T_j^d(k) ∋ A_j(k) ∈ S
    T_j^s(x_j) = i − d_j(x_j); //send time for
A_j(x_j)
    i− = d_j(x_j); // Update i
    x_j--;
    // Update S by removing S(j) from S
    S− = S(j);
    // add A_j(x_j) to S
    if (x_j ≠ 0)
      S+ = AU(j, x_j)
    if (i > max{T_j^d(k)}),  T_j^d(k) ∋ A_j(k) ∈ S
      // there is a gap on the channel
      G+ = ({max{T_j^d(k)},  i − {max{T_j^d(k)})
  }
  if T_first^s < 0
  then T_s = |T_first^s|
  T_j^d(k)+ = T_s,        ∀j,  k
END
```

The process begins with S initialized with the last AU of each of the objects in the presentation and G initially empty. In each of the iterations, the AU with the latest decoding time is scheduled as close to the decoding time as possible. Ties are broken arbitrarily. Once the AU of an object is scheduled, the next AU of that object is added to S as long as there are AUs to be scheduled. The current time is given by i. A value of i greater than the largest decoding time of AUs in $S(\max\{T_j^d(k)\})$ indicates idle time on the channel (gaps or slots). The channel is idle because nothing can be scheduled between $\max\{T_j^d(k)\}$ and i. This is illustrated in the example below.

Example: Consider two objects: object O_1 with two AUs and object O_2 with one AU with duration on channel, d, and decoding time stamp, T given as a set of tuples $\langle d, T\rangle$. $O_1 = \{\langle 7,7\rangle, \langle 10,21\rangle\}$ and $O_2 = \{\langle 5,6\rangle\}$. After $A_1(2)$ is scheduled, at time $T_1^s(2) = 11$, nothing can be scheduled between $T_1^d(1) = 7$ and current time $i = 11$ resulting in a gap on the channel. A negative value of the send time indicates that the AU has to be transmitted before the presentation starts giving rise to a startup delay. The FullSched example is depicted in Fig. 4.

When S becomes empty, i.e., all AUs are scheduled, a negative value of i indicates the required startup delay for the presentation and G gives the set of gaps on the channel. Since the decoding times $T_j(k)$ are all nonnegative, once i becomes negative, there are no gaps on the channel indicating that the AUs are tightly packed. A gap is not an indication of the suboptimality of the schedule. However, it may indicate the suboptimality of the bandwidth-optimized schedule, i.e, it may be possible to schedule the presentation at a lower bandwidth. When $N = 1$, this algorithm can be used to determine the schedulability of individual objects and determine the unschedulability

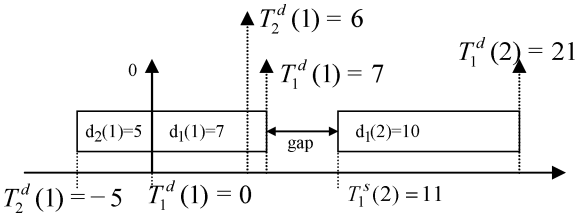

Fig. 4. FullSched example.

of a presentation. This is especially useful during the content creation process where objects are added to create presentations. When an object is added during an editing operation, it is faster to determine the unschedulability of a presentation by computing the independent schedules of objects and adding the startup delays of the independent schedules. However, a full schedule should still be computed after the editing operations to determine the schedulability of the presentation under given resource constraints. This algorithm is not efficient in computing the schedulability during the content creation process, as the full schedule needs to be recomputed every time an object is added. We next present a gap-scheduling algorithm that computes incremental schedules to determine the schedulability of a presentation and is well suited for the content creation process. We also prove that FullSched and gap-scheduling algorithm compute startup delay optimal schedules.

Theorem 3: Algorithm FullSched produces a startup delay-optimal schedule.

Proof: The algorithm selects an AU with the latest decoding time and schedules it as close to the deadline as possible, i.e., the algorithm schedules the AUs in nonincreasing order of their decoding times. On a conceptual timeline, with time increasing from left to right, we are stacking the AUs as much to the right as possible. Gaps occur only when there is nothing to be scheduled in that gap. Any (or part of) AUs that appear to the left of the origin (time = 0) give the startup delay. Since the algorithm always moves the AUs to the right whenever possible, the startup delay is minimized. A smaller startup delay is not possible because, it would mean moving the AUs to the right implying that there is a usable gap on the channel. This cannot be the case because the algorithm would have scheduled an AU in that gap!

B. GapSched Algorithm

The gap-scheduling (GapSched) algorithm schedules AUs in the available gaps on a channel. It starts with available gaps on a channel and tries to fit an AU or a partial AU using the SplitAndSchedule procedure. The initial set of gaps may be obtained by using FullSched to schedule a single object. The algorithm looks for the first available gap starting at a time less than the decoding time of the AU to be scheduled. Since G is already sorted in the decreasing order of gap times, the look up can be done very efficiently. If the gap duration is not long enough to fit an AU, the AU is split, with one part scheduled in the current gap and the other added to S to be scheduled next. The AUs in the presentation are iteratively scheduled until S becomes empty.

S contains all the AU of the object j $S = \{A_k(k)\}$, $1 \le k \le n_j$, and $j \in \{N\}$. Sort AUs in S in the decreasing order

of their decoding times. G = set of available slots $\neq \{\phi\}$. $G(l)$ is the lth tuple in G with start time $G(l).t$ and duration $G(l).d$. $k = n_j$.

```
BEGIN
  while (S ≠ φ) {
    find a slot l, G(l), such that
T_j^d(k) > G(l).t
    if (G(l).d ≥ d_j(k)) {
          T_j^s(k) = G(l).t − d_j(k)  //send
        time for A_j(k)
          k--;
          // update the gap
          if (G(l).d − d_j(k) > 0)
      G(l).d = G(l).d − d_j(k)
     else
      G− = {G(l)};
          // remove AU from the set
        S− = A_j(k);
        }
        else{
          PROCEDURE   SplitAndSchedule
      (A_j(k), G(l));
        }
  }
END
```

Split the AU into two parts, one part that is scheduled in $G(l)$ and the other that is placed back in S.

```
PROCEDURE SplitAndSchedule (A_j(k), G(l))
{
  Create a subAU of length G(l).d with
the last G(l).d * C bytes of the AU.
  t'_j(k) = G(l).t;
  d'_j(k) = d_j(k) − G(l).d;
  G− = {G(l)};
}
```

C. IncSched Algorithm

The incremental scheduling (IncSched) algorithm computes the schedule for a presentation by considering one object at a time. This is a typical content creation scenario where objects are composed to create a presentation. Instead of recomputing the full schedule with FullSched algorithm each time an object is added, this algorithm computes the schedules incrementally by scheduling the AU in the available gaps. Note that not all the gaps are schedulable. A gap is unschedulable if there are no AUs with decoding times greater than the gap time. An unschedulable gap indicates unused bandwidth, which is either due to the structure of the presentation or due to a suboptimal schedule. The IncSched algorithm uses FullSched and GapSched algorithms to schedule a presentation. This algorithm appears to be more efficient than FullSched as it schedules parts of AUs and fills all the gaps. However, this is only as efficient as FullSched as far as startup delay is concerned. Splitting the AUs in order to pack the gaps is not going to decrease the startup delay, as the available channel capacity is the same. The startup delay, like in other cases, is given by the send-time of the first AU transmitted.

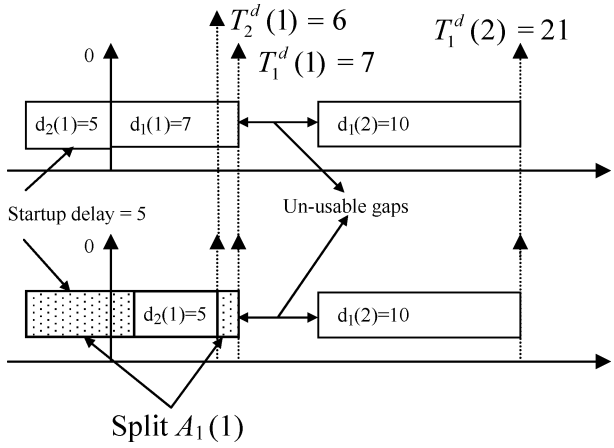

Fig. 5. Schedules computed using FullSched and IncSched.

OBJ is the set of objects in the presentation.

```
BEGIN
   Apply FullSched and compute
schedule for object 1.
   //LRG is a sufficiently large
number to accommodate startup delay.
  G+ = {⟨−LRG, i − LRG⟩}
  for j ∈ OBJ − {1}, apply gap scheduling
GS to j.
  for j   ∈   OBJ, find t_first, the send
time of the first AU to be transmitted
(smallest t_j(k))
  if T_first^s < 0
  then T_s = |T_first^s|
  T_j^d(k)+ = T_s,      ∀j, k
END
```

Theorem 4: The IncSched algorithm is startup delay-optimal.

Proof: The first object is scheduled using FullSched producing a startup delay-optimal schedule for that object. GapSched, when applied iteratively to the remaining objects, packs the AUs tightly, i.e., an AU is scheduled if the gap time is less than the decoding time for that AU. The resulting startup delay is optimal because the algorithm would reduce the startup delay by moving the AU to the right on the timeline if any schedulable gap is available.

Example: Consider two objects: O_1 with two AUs, and O_2 with one AU with duration on channel d and decoding time stamp T given as a set of tuples $\langle d, T\rangle$. $O_1 = \{\langle 7, 7\rangle, \langle 10, 21\rangle\}$ and $O_2 = \{\langle 5, 6\rangle\}$. The top part of Fig. 5 shows the schedule computed using FullSched, and the bottom half shows the schedule computed with IncSched, with object 2 scheduled first using FullSched. The figure also shows unschedulable gaps in both the schedules.

There may be cases where splitting the AUs is necessary, for example, the underlying transport layer may not be able to

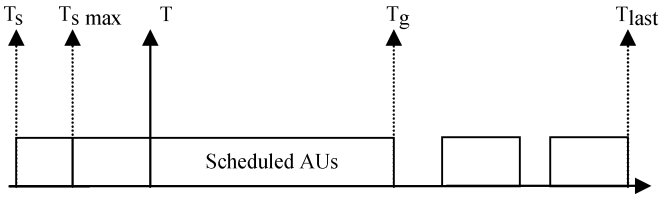

Fig. 6. First gap-time and startup delay of a presentation.

handle large AUs. This result shows that AUs can be split while maintaining the optimality of the schedule. Although the IncSched algorithm produces an optimal schedule and is useful in determining the schedulability of a presentation in applications such as content creation, the schedule generated by FullSched may be more efficient when the overhead due to splitting and packetizing is significant.

D. MinC Algorithm

In scheduling AV objects, we have so far answered two questions: 1) Is the given presentation schedulable under the given resource constraints, and 2) what is the minimum startup delay required for this presentation? If the answer to question 1 is negative (or if bandwidth consumption needs to be minimized), the question we need to address is what are the minimum amounts of resources required to schedule the presentation? Since we cannot make assumptions about decoder buffers in order to keep the schedules player-independent, the only resource that can be acquired is the bandwidth (C). We next present the MinC algorithm that computes the minimum bandwidth (CBR) required to schedule a presentation.

This algorithm is based on the premise that there is a gap on the channel only when everything else *after* the gap-time has been scheduled. Otherwise, an unscheduled AU would have taken up the gap. The presentation is not schedulable because there is not enough channel capacity until the first gap time, T_g (smallest gap time). Consider the case in Fig. 6. T_g is the first gap-time $T_s^{\max}$ is the maximum allowable startup delay with the current channel capacity, and T_s is the current startup delay. The channel capacity should be increased to accommodate $T_s - T_s^{\max}$ in the duration $T_g - T_s^{\max}$. The new value of C then is $C_{\text{new}} = C * ((T_g - T_s)/(T_g - T_s^{\max}))$. The additional bandwidth necessary is therefore equal to $C * ((T_s^{\max} - T_s)/(T_g - T_s^{\max}))$. The algorithm also outputs the BP for the presentation in the form of three tuples ⟨capacity, start, end⟩. Note that the increased channel capacity is not going to affect the schedule from T_g to T_{last}. A finer BP can be obtained by initializing C with $C_{\min}$ and increasing C by a small value in each iteration.

```
BEGIN
  G_c = |G|=gap count gap count, number
of gaps on the channel.
  BP = {φ}
  T_last is the decoding time of the
first AU scheduled (=duration of the
presentation)
  C = C_min, computed using the results
of theorem 2.
    SCHEDULE: compute schedule using
FullSched
    If schedulable goto END
    if (G_c == 0)
      T_g = T_last
    else {
      T_g−old = T_g
      Find the smallest gap time, T_g
      BP+ = {⟨C, T_g, T_g−old⟩}
    }
    C = C + C * ((T_s^max − T_s)/(T_g − T_s^max))
    goto: SCHEDULE
END
```

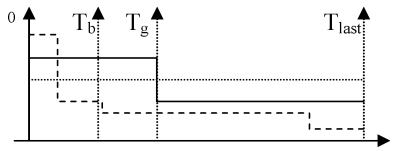

Fig. 7. Typical BP generated by MinC.

The channel capacity output by the algorithm is in the form of a set of three tuples forming a BP. The minimum CBR channel required is given by the maximum value of C in the BP. This profile may also be used to reserve session bandwidth efficiently. Since the schedule is computed from last to first (right-to-left on the timeline), the BP will always be a step function with possible steps (decreasing) from left to right. Fig. 7 shows some sample profiles. This algorithm does not give the best profile to reserve variable session bandwidth since the algorithm does not reduce the bandwidth when it is unused. Consider the example shown in the Fig. 7. At T_g, a capacity increase is necessary. Suppose the increase in C at T_g is sufficient to schedule the presentation. It is possible that the presentation from 0 to T_b could have been scheduled with a much smaller capacity.

E. BestSched Algorithm

When a presentation cannot be scheduled with the given resources, and additional resources cannot be acquired, the only way to schedule the presentation is to drop some AUs. AUs cannot be dropped arbitrarily as they have different effects on the presentation. Content creators should assign priorities to objects and possibly AUs of objects to help a scheduler in determining the AUs to be dropped. The following algorithm schedules a presentation by dropping lower priority objects.

```
BEGIN
SCHEDULE: Compute schedule using
FullSched.
  if (B <= T_s * C) {
    Remove A_j(k) of lower priority ob-
jects such that,
    R = {A_j(k)} ∋ C * Σ d_j(k) ≥ T_s * C − B
    A− = {R}
    goto: SCHEDULE
  }
END
```

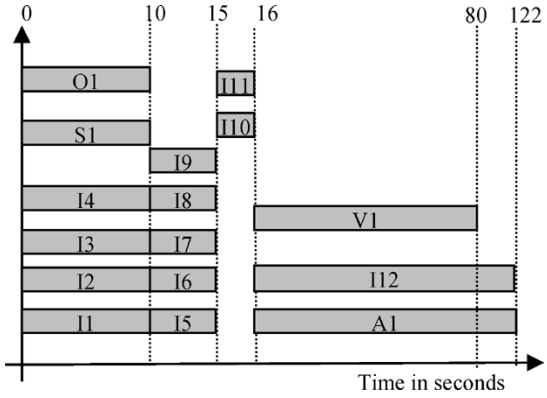

Fig. 8. Structure of the presentation in the example.

V. Results and Discussion

Determining the schedulability of a presentation is of $O(n)$ complexity, where n is the number of AUs in the presentation. Both FullSched and GapSched fall under this category. These algorithms are used to determine the schedulability, compute an optimal startup delay for the given channel capacity, and for computing incremental schedules. The MinC algorithm, used to compute the minimum channel capacity required to schedule the presentation, calls FullSched iteratively with channel capacity incremented in each iteration. The number of iterations depends on the structure of the presentation and the initial value of C. The complexity of this algorithm is $O(Kn) = O(n)$, where K is a constant determined by the structure of the presentation and the initial channel capacity. The proposed algorithms are fast enough to determine the schedulability of the presentations in real-time.

The structure of the presentation has significant impact on the performance of MinC algorithm. To aid the discussion, we consider a relatively complex MPEG-4 presentation with structural overview as shown in Fig. 8. The properties of the objects in the presentation are tabulated in Table I.

In the following discussion we refer to objects by the codes shown in the first column of the table. The presentation is made up of 16 objects including scene description, object description, images, audio, and video. A screenshot of the presentation as seen in an MPEG-4 player is shown in Fig. 9. The presentation is composed of three scenes. Before the scenes are loaded, the scene description and object description streams are received and decoded by the terminal. The first scene consists of four jpeg images (I1–I4) animated to give a breakout effect. The scene is encoded to animate the images for 10 s and then load the second scene. The second scene consists of a background image (I5), four logos with animation effects (I6–I9), and two images (I10 and I11) with descriptive text of the following AV scene. The last scene consists of a background image, an audio stream, and a video stream. The temporal layout of the presentation is shown in Fig. 8. The times indicated are the decoding times of the first AUs of the objects starting at that time. Thus, the first four images (I1–I4), the scene description (S1) and the object descriptor stream (O1) should reach the decoder before anything is displayed on the screen. This amounts to the minimum startup delay for the presentation. The objects I5–I9 should reach the decoder by the time $t = 10$, I10 and I11 by 15, and the first AU of V1 and A1, and the object I12 should reach the terminal by the time $t = 16$. The video ends at $t = 80$ while the audio stream continues until the end of the presentation. The total length of the presentation is 122. This temporal ordering of objects in the presentation results in higher data rates toward the beginning of the presentation (object data to be delivered in the first 16 s: 261 Kb$\sim$ $j = 130$ Kb/s).

TABLE I
Properties of Objects in the Example

Object FileName (ID)	Size (KB)	Start Time	AU Count
Scene.od (O1)	0.5	0	1
Scene.bif (S1)	1 (1025 Bytes)	0	4
Main1.jpg (I1)	25	0	1
Main2.jpg (I2)	22	0	1
Main3.jpg (I3)	19	0	1
Main4.jpg (I4)	20	0	1
main_ui.jpg (I5)	39	10	1
Advent_logo.jpg (I6)	7	10	1
CU_logo.jpg (I7)	9	10	1
Lm_logo.jpg (I8)	6	10	1
Xbind_logo.jpg (I9)	8	10	1
geo_pict.jpg (I10)	23	15	11
dance_pict.jpg (I11)	29	15	1
next_page.jpg (I12)	51	16	1
clip01.h263 (V1)	552	16	974
clip01.g723 (A1)	64	16	3233

Fig. 9. Snapshot of the player with the scheduled presentation.

A. Startup Delay and Capacity Computation

Fig. 10 shows the plot of the minimum channel capacity required for a given startup delay. This is a scenario with variable buffer at the terminal. We assume a work-conserving transport layer that delivers the objects at the minimum required capacity. The amount of buffer available at the terminal should be at least sufficient to store the data during the startup. For a startup delay of T_s, if $C_{\min}$ is the min capacity required, then the buffer at

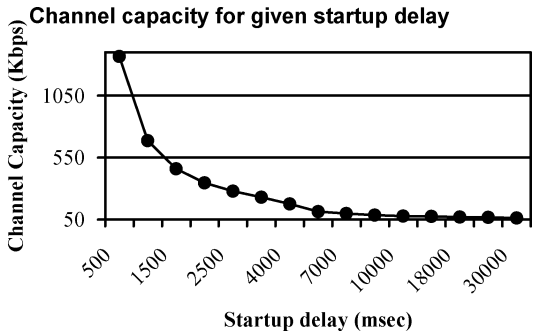

Fig. 10. Computing min capacity using MinC.

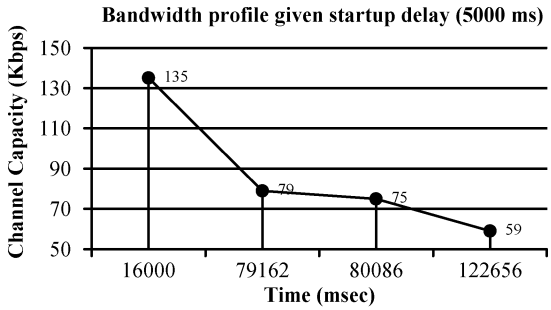

Fig. 11. Computing BP using MinC.

the terminal $B_{\min} > T_s * C_{\min}$. This curve is useful to determine the amount of buffer (delay) required based on the available network capacity, especially with terminals such as PC's with sufficient memory. As mentioned earlier in the discussion of the MinC algorithm, the MinC algorithm also computes the BP for presentations.

Fig. 11 shows the BP computed for a startup delay of 5 s. The minimum capacity in the profile is 59 Kb/s even for the segment (80–120 s) that only has low bit rate audio (6 Kb/s). This is because MinC starts with an initial value of C computed using the residual data volume as described in Section III. This starting point is acceptable for computing a CBR channel required; for a profile to be used in reserving variable network resources, a lower initial value of C should be selected. The final bandwidth jump in the profile gives the minimum channel capacity required for the given delay or buffer.

B. Buffer and Capacity Computation

The available buffer at the terminal determines the amount of startup delay a terminal can support. The available channel capacity imposes a lower limit on the buffer required. Lower channel capacity implies higher startup delays, and hence, larger required buffer. Fig. 12 gives the required buffer at various channel capacities. This can be directly converted to the startup delay at that capacity. Computing the capacity for a given buffer is bit more computationally intensive. Unlike the previous case where we assumed enough capacity to support the required startup delay, the buffer capacity is fixed in this case. This typically the scenario when using dedicated devices such as set-top-boxed with fixed receiving buffers. Since the buffer is fixed, the supported startup delay decreases as channel capacity increases. For MinC algorithm to complete, the reduction in startup delay due to increased channel capacity should

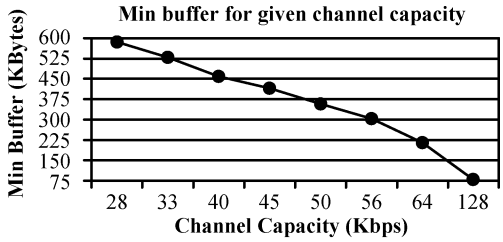

Fig. 12. Minimum required buffer.

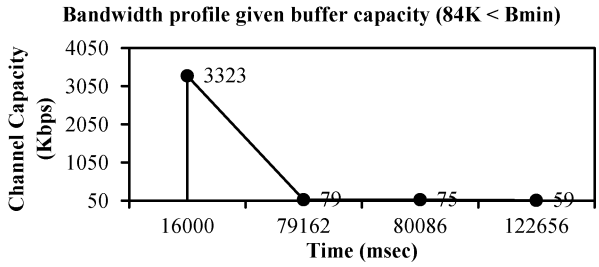

Fig. 13. Partial profile for low terminal buffer.

be greater than the reduction in startup delay supported by the buffer.

Fig. 13 shows a case with terminal buffer $< B_{\min}$. For the given presentation $B_{\min} = 85$ Kb, the size of objects to be decoded at time 0. As discussed in Section III, for a presentation to be schedulable, the available buffer should be greater than $B_{\min}$, the lower bound on the required buffer capacity for the presentation. Since the terminal buffer is less than the required buffer, at any given capacity C, the supported startup delay $(B_{\text{term}}/C) < (B_{\min}/C)$, the required startup-delay. The presentation is hence unschedulable with terminal buffer capacity of 84 Kb. This is depicted in Fig. 13, which shows the presentation is unschedulable even at 3323 Kb/s. The plot shows that no matter how much the channel capacity is increased, the presentation cannot be scheduled because of limited terminal buffer. To avoid infinite loops in MinC, the scheduler should first examine the available and required buffer capacities.

VI. Conclusion

We presented the problem of scheduling object-based AV presentations under resource constraints. The problem is NP-complete in the strong sense. We explored similarities with the problem of sequencing jobs on a single machine and used the idea of last-to-first scheduling to develop heuristic algorithms to determine schedulability and compute startup delay-optimal schedules. The proposed algorithms are applicable in optimizing the delivery of any generic object-based AV presentations and also mixed media presentations described with formats such as SMIL [30]. The algorithms can be applied in scheduling object-based MPEG-4 presentations.

We introduced the notion of residual data volume to compute lower bounds on buffer, channel capacity, and startup delay. Determining the schedulability of presentations online is important for applications like content creation where an additional object may make the presentation unschedulable. We presented an algorithm that computes incremental schedules and produces a startup delay optimal schedule. The incremental scheduling is

very useful in application such as content authoring where authors add and remove objects to presentations during creation. The IncSched algorithm computes the incremental resources required to schedule the additional object. Starting with a lower bound on the channel capacity for scheduling a presentation, the MinC algorithms minimizes the CBR channel capacity required to schedule the presentation. The proposed algorithms are of low complexity and can be implemented efficiently.

REFERENCES

[1] O. Avaro, A. Eleftheriadis, C. Herpel, G. Rajan, and L. Ward, "MPEG-4 systems: Overview," *Signal Process. Image Commun.*, vol. 15, no. 4–5, pp. 281–298, Jan. 2000.

[2] M. C. Buchanan and P. T. Zellweger, "Scheduling multimedia documents using temporal constraints," in *Proc. NOSDAV*, 1992, pp. 223–235.

[3] J. Carlier, "The one machine sequencing problem," *Eur. J. Oper. Res.*, vol. 11, pp. 42–47, 1982.

[4] T. L. Casvant and J. G. Kuhl, "A taxonomy of scheduling in general purpose distributed computing systems," *IEEE Trans. Software Eng.*, vol. 14, no. 2, pp. 141–154, Feb. 1988.

[5] P. Chretienne, E. G. Coffman, Jr., J. K. Lenstra, and Z. Liu, Eds., *Scheduling Theory and Its Applications*. New York: Wiley, 1995.

[6] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*. Reading, MA: Addison-Wesley, 1967.

[7] M. L. Escobar-Molano, "Management of resources to support coordinated display of structured presentations," Ph.D. dissertation, Univ. Southern California, Los Angeles, CA, 1996.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco, CA: Freeman, 1979.

[9] L. Grossglauser and S. Keshav, "On CBR service," in *Proc. INFOCOM*, Mar. 1996, pp. 129–137.

[10] L. A. Hall, "Approximation algorithms for scheduling," in *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, Ed. Boston, MA: PWS-Kent, 1997, pp. 1–45.

[11] C.-C. Han, K.-J. Lin, and J. W.-S. Liu, "Scheduling jobs with temporal distance constraints," *SIAM J. Comput.*, vol. 24, no. 5, pp. 1104–1121, Oct. 1995.

[12] C. Herpel and A. Eleftheriadis, "MPEG-4 systems: elementary stream management," *Signal Process. Image Commun.*, vol. 15, no. 4–5, pp. 299–320, Jan. 2000.

[13] *Generic coding of moving pictures and associated audio (MPEG-4 Systems)—ISO/IEC 14386-1*, ISO/IEC/SC29/WG11, Apr. 1999.

[14] *Delivery multimedia integration framework (DMIF)—ISO/IEC 14 496-6*, ISO/IEC/SC29/WG11, Feb. 1999.

[15] N. S. Jayant, "Signal compression: technology targets and research directions," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 796–818, June 1992.

[16] H. Kalva, L.-T. Cheok, and A. Eleftheriadis, "MPEG-4 systems and applications," in *Demonstration, ACM Multimedia*, Orlando, FL, 1999.

[17] H. Kalva, *Delivering MPEG-4 Based Audio Visual Services*. Norwell, MA: Kluwer, 2000.

[18] L. Kleinrock and A. Nilsson, "On optimal scheduling algorithms for time-shared systems," *J. ACM*, vol. 28, no. 3, pp. 477–486, July 1981.

[19] E. L. Lawler, "A functional equation and its application to resource allocation and sequencing problems," *Manage. Sci.*, vol. 16, no. 1, pp. 77–84, Sept. 1969.

[20] ——, "Optimal sequencing of a single machine subject to precedence constraints," *Manage. Sci.*, vol. 19, no. 5, pp. 544–546, Jan. 1973.

[21] J. K. Lenstra, A. H. G. R. Kan, and P. Brucker, "Complexity of machine scheduling problems," *Annal. Discrete Mathematics 1*, pp. 343–362, 1977.

[22] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 413–427, Dec. 1990.

[23] ——, "Multimedia synchronization protocols for broadband integrated services," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1368–1382, Dec. 1991.

[24] S. Paek and S. F. Chang, "Video server retrieval scheduling and resource reservation for variable bit rate scalable video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 460–474, Apr. 2000, to be published.

[25] A. Puri and A. Eleftheriadis, "MPEG-4: A multimedia coding standard supporting mobile applications," *ACM Mobile Networks Applicat. J.*, vol. 3, no. 1, pp. 5–32, June 1998.

[26] J. Signes, Y. Fisher, and A. Eleftheriadis, "MPEG-4's binary format for scene description," *Signal Process. Image Commun.*, vol. 15, no. 4–5, pp. 321–345, Jan. 2000.

[27] J. Song, A. Dan, and D. Sitaram, "Efficient retrieval of composite multimedia objects in JINSIL distributed system," in *Proc. ACM SIGMETRICS*, June 1997, pp. 260–271.

[28] L. Torres and M. Kunt, Eds., *Video Coding: The Second Generation Approach*. Norwell, MA: Kluwer, 1996.

[29] D. Trietsch, "Scheduling flights at hub airports," *Transpor. Res.*, vol. 27, no. 2, pp. 133–150, 1993.

[30] J. D. Ullman, "NP-complete scheduling problems," *J. Comput. Syst. Sci.*, no. 10, pp. 384–393, 1975.

[31] "Synchronized multimedia integration language (SMIL 1.0)," W3C Recommendation, June 1998.

Hari Kalva (M'00) received the B.Tech. degree in electronics and communications engineering from N.B.K.R. Institute of Science and Technology, S.V. University, Tirupati, India, in 1991, the M.S. degree in computer engineering from Florida Atlantic University, Tallahassee, in 1994, and the M.Phil. and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1999 and 2000, respectively.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Florida Atlantic University (FAU), Boca Raton, FL. Prior to his appointment at FAU, he was a consultant with Mitsubishi Electric Research Labs, Cambridge, MA, where he worked different projects including MPEG-2 to MPEG-4 real-time video transcoding. He was a Cofounder and the Vice President of Engineering of Flavor Software, a New York company founded in 1999, that developed MPEG-4 based solutions for the media and entertainment industry. He has over 24 published works and three patents (eight pending) to his credit. He is the author of one book and coauthor of five book chapters. His research interests include video compression and communications, content representation, content adaptation, and mobile multimedia.

Dr. Kalva is a member of the IEEE Communications Society, the IEEE Signal Processing Society, and the ACM.

Alexandros Eleftheriadis (M'95–SM'02) was born in Athens, Greece, in 1967. He received the Diploma in electrical engineering and computer science from the National Technical University of Athens, Athens, Greece, in 1990, and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1992, 1994, and 1995, respectively.

Since 1995, he has been with the faculty of the Department of Electrical Engineering, Columbia University (currently as an Associate Professor), where he is leading a research team working on multimedia signal processing, with emphasis on multimedia software, video signal processing and compression, and video communication systems, and most recently, music signal processing. In 1999, he cofounded Flavor Software Inc., New York, a company that builds software for creating and distributing rich media content using MPEG-4. In the summers of 1993 and 1994, he was with the Signal Processing Research Department, AT&T Bell Labs, Murray Hill, NJ, where he performed research on very low bit rate coding systems. He has published more than 100 papers in international journals and conferences, and has been awarded 11 patents.

Dr. Eleftheriadis is a recipient of an National Science Foundation CAREER Award and has been elected member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He also served as the Editor of the MPEG-4 Systems (ISO/IEC 14 496-1) specification. He serves in the editorial board of the *Multimedia Tools and Applications Journal*, and has served as a Guest Editor, Committee Member, and Organizer for several international journals and conferences. He is a member of the AES, the ACM, and the Technical Chamber of Greece.