NCS TIB 91-5

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 91-5

INVESTIGATION OF HDTV FOR APPLICATION TO GOVERNMENT TELECONFERENCING

FEBRUARY 1991

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM

WASHINGTON, D.C. 20305

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
February 1991

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Investigation of HDTV for Application to Government Teleconferencing

5. FUNDING NUMBERS
C-DCA100-87-C-0078

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Delta Information Systems, Inc.
300 Welsh Road, Suite 120
Horsham, PA 19044-2273

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Communications System
Office of Technology & Standards
701 S. Court House Road
Arlington, VA 22204-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NCS TIB 91-5

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; distribution unlimited.

12b DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this is to investigate HDTV to determine its potential applicability to teleconferencing within the Government community. Work on this project was divided into three parts; (1) HDTV standards activitiy (2) TV compression technology, and (3) communications considerations.

14 SUBJECT TERMS
High Definition Television (HDTV)
Teleconferencing

15. NUMBER OF PAGES
108

16. PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT
Unclassified

Unclassified

Unclassified

Unlimited

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s) Use the following labels:

C	-	Contract	**PR**	-	Project
G	-	Grant	**TA**	-	Task
PE	-	Program Element	**WU**	-	Work Unit Accession No

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es) Self-explanatory

Block 8. Performing Organization Report Number Enter the unique alphanumeric report number(s) assigned by the organization performing the report

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es) Self-explanatory

Block 10. Sponsoring/Monitoring Agency Report Number *(If known)*

Block 11. Supplementary Notes Enter information not included elsewhere such as Prepared in cooperation with Trans of To be published in When a report is revised include a statement whether the new report supersedes or supplements the older report

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	-	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	-	See authorities.
NASA	-	See Handbook NHB 2200.2.
NTIS	-	Leave blank.

Block 12b. Distribution Code.

DOD	-	Leave blank.
DOE	-	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	-	Leave blank.
NTIS	-	Leave blank.

Block 13. Abstract Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report

Block 14. Subject Terms Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages Enter the total number of pages

Block 16. Price Code Enter appropriate price code *(NTIS only)*

Blocks 17. - 19. Security Classifications Self explanatory Enter U S Security Classification in accordance with U S Security Regulations (i e , UNCLASSIFIED) If form contains classified information, stamp classification on the top and bottom of the page

Block 20 Limitation of Abstract This block must be completed to assign a limitation to the abstract Enter either UL (unlimited) or SAR (same as report) An entry in this block is necessary if [illegible]

NCS TECHNICAL INFORMATION BULLETIN 91-5

INVESTIGATION OF HDTV FOR APPLICATION TO GOVERNMENT TELECONFERENCING

JANUARY 1991

PROJECT OFFICER	APPROVED FOR PUBLICATION:
	Dennis Bodson
GARY M. REKSTAD Electronics Engineer Office of NCS Technology and Standards	DENNIS BODSON Assistant Manager Office of NCS Technology and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identified, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the International Organization for Standardization, and the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union. This Technical Information Bulletin presents and overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of teleconferencing. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS
Washington, DC 20305-2010

INVESTIGATION OF HIGH DEFINITION TELEVISION FOR APPLICATION TO TELECONFERENCING

March, 1991

FINAL REPORT
DCA100-87-C-0078

Submitted to:
NATIONAL COMMUNICATIONS SYSTEM
WASHINGTON, DC

DELTA INFORMATION SYSTEMS, INC.
300 Welsh Road, Ste. 120
Horsham, PA 19044-2273
TEL: (215) 657-5270 FAX: (215) 657-5273

TABLE OF CONTENTS

1.0 INTRODUCTION

This document summarizes work performed by Delta Information Systems, Inc. (Delta) for the National Communications System (NCS), Office of Technology and Standards. The NCS is responsible for the management of the Federal Telecommunications Standards Program, which develops telecommunications standards, whose use is mandatory for all Federal departments and agencies.

This document is a final report for a Task Order on Contract DCA100-87-C-0078. The titles for the contract and Task Order are listed below.

- Contract DCA100-87-C-0078
  Development of Federal Telecommunication Standards Relating to Digital Facsimile and Video Teleconferencing
- Task Order
  Investigation of High Definition Television for Application to Teleconferencing

In recent years, there has been considerable activity in the development of technology and standards related to High Definition Television (HDTV). According to the Advanced Television Systems Committee (ASTC), "the term HDTV refers to television systems with approximately twice the horizontal and vertical emitted resolution of standard NTSC. HDTV systems are wide aspect ratio systems and may include improvements from IDTV (Improved Definition Television) and EDTV (Extended Definition Television)". The purpose of this task is to investigate HDTV to determine its potential applicability to teleconferencing within the Government community. Work on this project was divided into three parts -- (1) HDTV standards activity, (2) TV compression technology, and (3) communications considerations.

- **HDTV Standards Activity**

One of the most fundamental and complex tasks facing the ATSC and the FCC is the selection of the scan format for the HDTV signal (e;g. number of scan lines, interlace vs progressive scan, bandwidth, etc.). This decision could have a significant impact on the ease with which the signal is used for teleconferencing. For this reason, work on this project was directed toward this review of the HDTV standards activity from the perspective of teleconferencing. Work on this subtask is discussed in Section 2.0.

- **Compression of the HDTV Signal**

If teleconferencing is to be practical, the signal must be transmitted over switched communication channels having a bit rate low enough to be economical. The teleconferencing industry has been growing rapidly in recent years because compression technology has successfully reduced the bit rate required for transmission. Since the HDTV signal has such high resolution, the need for compression is even more critical for HDTV then it is for the NTSC signal. The purpose of this task is to examine compression technology for the application of HDTV to teleconferencing. Work on this task was divided into three parts as listed below.

- Compression technology was reviewed in general to provide a broad background for further coding studies. Results of this investigation are summarized in Section 3.0.

- There has been considerable recent interest in the use of sub-band coding to compress the HDTV signal. On this task, Delta analyzed the effectiveness of sub-band coding by means of computer simulation. The results are included in Section 4.0.

- As described in Section 2.0, there is a general trend toward the adoption of a domestic standard for HDTV transmission based upon all digital technology. It is also explained that, at the present time, there are three ATSC proponents of all-digital systems as listed below.

PROPONENT TEAM	SYSTEM	SCAN FORMAT	LUMINANCE PIXELS	CHROMA PIXELS
AT&T, ZENITH	SPECTRUM COMPATIBLE	787.5/1:1	720 X 1280	360 X 640
GENERAL INSTRUMENT, MIT	DIGICIPHER	1050/2:1	960 X 1408	480 X 352
SARNOFF, NBC, PHILIPS, THOMSON	ADVANCED COMPATIBLE TV	1050/2:1	960 X 1440	480 X 720

All three proposed systems employ DCT coding (8 x 8 pixels) and motion compensation which is similar to the coding technique employed in CCITT

Recommendation H.261. An overview of this Recommendation is provided in Section 5.0 for two reasons: (1) it provides information on technology which is similar to the three proposed systems; (2) it may stimulate the adoption of an HDTV standard which is very similar to H.261. This would clearly be advantageous to the video telephony community.

- **Communication Considerations for Teleconferencing**

In general, video teleconferencing requires a high transmission bit rate relative to other services such as voice and data. For that reason, the availability of teleconferencing for the government community is dependent upon the availability of ubiquitous, inexpensive, switched, communication channels operating at high bit rates. The purpose of this section is to, in very general terms, examine communication issues as they relate specifically to video teleconferencing. The discussion is divided into three parts: (1) teleconferencing communication today, (2) narrowband ISDN, and (3) broadband ISDN.

Conclusions drawn from the work performed on this project are summarized in Section 7.0.

2.0 HDTV STANDARDS

After more than four decades of TV broadcasting, first in monochrome and later in color, it has become obvious that the many advances in technology make a much improved picture quality possible and desirable. Unfortunately, the long established picture formats and standards put a straight jacket on the development of new and better systems. Scanning formats (line numbers, interlace and aspect ratio) and RF channel assignments are extremely difficult to change at this time; yet they are due mainly to the historical background and not to technological factors.

A number of proposed Advanced Television (ATV) systems have been proposed which can be categorized as Improved Definition TV (IDTV) and Extended Definition TV (EDTV), and High Definition TV (HDTV). These terms are defined by the Advanced Television Systems Committee (ATSC), the standards group formed by the TV industry, as follows.

IDTV - IMPROVED DEFINITION TELEVISION - The term Improved Definition Television refers to improvements to NTSC television which remain within the general parameters of NTSC emission standards and, as such, would require little or no FCC action. Improvements may be made at the source and/or at the television receiver and may include improvements in encoding, filtering, ghost cancellation, and other parameters that may be transmitted and received as standard NTSC in a 4:3 aspect ratio.

EDTV - EXTENDED DEFINITION TELEVISION - The term Extended Definition Television refers to a number of different improvements that modify NTSC emissions but that are NTSC receiver-compatible (as either standard 4:3 or "letter-box" format). These changes may include one or more of the following:

1. Wide aspect ratio.
2. Extended picture definition at a level less than twice the horizontal and vertical emitted resolution of standard NTSC.
3. Any applicable improvements of IDTV.

For purposes of identification, EDTV transmitted as 4:3 is referred to as EDTV, and when transmitted in a wider aspect ratio, as EDTV-Wide.

HDTV - HIGH DEFINITION TELEVISION - The term High Definition Television refers to television systems with approximately twice the horizontal and vertical emitted resolution of standard NTSC. HDTV systems are wide aspect ratio systems and may include applicable improvements from IDTV and EDTV.

IDTV and EDTV feature compatibility with the existing NTSC system which automatically puts severe limitations on their achievable performance. HDTV eliminates this constraint and thus holds promise for future development.

The focus of the activity of the Federal Communications Commission (FCC), and of the Advanced Television Systems Committee has been to define an HDTV format and transmission standard for terrestrial broadcast in the U.S. The over-the-air channel is the most difficult technical task compared to cable or satellite delivery. These later two can readily be accomplished once the first is a reality. Progress toward this goal has accelerated ever since September 1988 when interested organizations were requested to declare themselves as proponents of record and submit preliminary details of proposed HDTV transmission systems. Fourteen submissions spanning the range of IDTV, EDTV, and HDTV provided a foundation for serious evolution toward a U.S. Standard.

Several significant events have channeled the efforts and narrowed the technical and political range of possibilities thus focusing activity and accelerating progress. The first event was the declaration by the FCC that no additional spectrum would be allocated for HDTV beyond that already allocated for the present NTSC television system.

The second event was the initial call for specific system proposals from proponents by 1 September 1988 which had the effect of changing the atmosphere away from verbal debates between experts at meetings to develop "Advanced Television Systems" concepts to analysis of specific proposed systems, some with actual hardware already to display. This had its intended effect of moving toward definite proposals and also had enormous impact on government and public awareness of HDTV, but did not in itself narrow the very large variety of proposed systems. Virtually all of the proposed systems were "analog" in that they used the transmission channel to send analog image information, albeit highly transformed and with multiple subcarriers and requiring considerable digital processing at both transmitter and receiver.

The third significant event was the proposal by Zenith of a "simulcast" system, inherently different from all others which employed some form of augmentation signal on a second channel or an "NTSC compatible" system wherein additional high definition information was packed into the same 6 MHz. channel as the present NTSC signal. The centerpiece of this proposal was its Spectrum Compatible aspect which permitted usage of the more than half of the presently unused TV channels (the so-called taboo channels) by employing a significantly lower power transmitter carrying a significant amount (but not all) of digital information and coded and synchronized with the neighboring adjacent RF channels such that mutual interference would be virtually eliminated (no longer taboo). This highly welcome aspect of improved channel utilization was carefully scrutinized by the FCC which subsequently determined it to be technically well founded.

The video coding scheme proposed by Zenith is a hybrid analog and digital method which digitized only the low spatial video frequencies which contain a very large percentage of the signal energy, while transmitting the higher frequency components in an amplified analog fashion. This and some other proposals take advantage of sub-band coding work performed at MIT to achieve video compression and was supported by the broadcasters, manufacturers and others. Sub-band coding, facilitated by pyramidal perfect-reconstruction filtering methods discovered and developed in the 1980's, reached a peak in 1989.

The fourth significant event was the declaration by the FCC that it would consider only HDTV proposals for Advanced Television service until a standard had been achieved and only after that possibly consider IDTV and EDTV proposals for interim service. This had the effect of further narrowing and channeling development focus. This has also reduced the number of proponents.[1]

In 1990 the atmosphere among broadcasters and TV industry people changed from skepticism that digital video compression of broadcast quality was still a decade away to a firm and total embrace of it as being just around the corner. The effect on the other proponents has been considerable. There are now six proponents, and between these six, partnerships have been formed to present the best posture toward being a successful proponent. A consortium of Phillips, Thomson, NBS and the Sarnoff Laboratory, actually formed before the GI proposal

[1] One of these proponents is the Faroudja Laboratories which proposed a method called Super-NTSC, which has attracted cable television interest as a means of achieving TV picture quality very noticeably superior to the present NTSC system in the near term. This system enhances the resolution of the present NTSC system while retaining its basic format and dramatically reduces the artifacts normally associated with a color in-band subcarrier modulation technique such as NTSC or PAL.

announced its charge to an all-digital system, has announced an all digital system, presumably using Transform Coding and Motion Compensation. Zenith has announced a broader partnership with AT&T which previously was to supply only the microelectronic implementation of Zenith's hybrid system, such that AT&T would furnish an all-digital realization of video compression and transmission. Again, this would be a Transform Coded Motion Compensated system. Finally, GI, who precipitated the initial switch to all digital transmission, has formed a partnership with MIT to furnish additional technology to their offering. The one remaining serious proposal of an analog system is the Japanese MUSE system which has recently started service over an 8 MHz, (not 6 MHz, as required by the FCC) satellite delivery system in Japan. It can be noted there are still analog systems proposed by other proponents to be tested by the ATSC which occupy testing slots left over by the formation of partnerships. These fill important competitive roles in covering an eventuality that some problem develops in the new, quite untested, all digital transmission schemes. The test schedule for proposed systems is provided in Figure 2.1.

A very healthy competition has developed between the remaining partnerships of proponents to not only furnish a digital high definition broadcast quality video compression system within six MHz, but to make it better than a competitor's offering. Fortunately, these betterments are not just features but direct themselves to higher image quality in terms of resolution and perhaps non-interlaces (flicker-less) imagery. The efforts of the Motion Picture Expert's Group (MPEG) toward a standard has been very helpful in contributing, for example, the motion compensated interpolation process to augment the motion compensated prediction method of the H.261 standard. This facilitates higher compression and less error propagation, a strong consideration with 30 frames per second systems operating with an over-the-air channel.

Current development activities include obtaining higher compression while retaining broadcast quality imagery, methods for obtaining essentially progressive rather than interlaced scan, and or robust transmission of digital signals over the air, especially with low power such as not to interfere with adjacent NTSC channels. This latter aspect is highly important but does not receive the press attention enjoyed by digital video compression. In the end, the winning proponent

ATV TEST SEQUENCE AND CALENDAR

ATV SYSTEM ACCESS PERIOD	INTERFACE CHECK	LAB TEST PERIOD		ATV SYSTEM/PROPONENT	SCANNING FORMAT*
		START	END		
1	4/8/91	4/12/91	6/12/91	ADVANCED COMPATIBLE TELEVISION (ACTV)/DAVID SARNOFF RESEARCH CENTER, ATRC **	525/59.94, 1:1
2	6/13/91	6/19/91	8/12/91	NARROW MUSE/NHK, JAPAN BROADCASTING CORP.	1125/60, 2:1
3	8/27/91	9/3/91	10/24/91	DIGICIPHER/GENERAL INSTRUMENT CORP.	1050/59.94, 2:1
4	10/25/91	10/31/91	12/27/91	SPECTRUM COMPATIBLE HDTV/ ZENITH ELECTRONICS CORP.	787.5/59.94, 2:1
5	12/30/91	1/8/92	3/3/92	DIGITAL SIMULCAST HDTV/ N.A. PHILLIPS CONSUMER ELECTRONICS CO.	1050/59.94, 2:1
6	3/4/92	3/10/92	4/30/92	CHANNEL COMPATIBLE HDTV/ MASSACHUSETTS INSTITUTE OF TECHNOLOGY	787.5/59.94, 1:1

* Number of scanning lines/cycles per second, interlaced (2:1) or progressive (1:1)
** Advanced Television Research Consortium (NBC, Phillips, Sarnoff, Thomson)

FIGURE 2.1

may be the one who demonstrates the most robust transmission system, since all of the proponents are essentially proposing the same basic hybrid motion compensated DCT transform coded compression system.

The ATSC has recently drafted documents for submission to CCIR Task Group 11/1 generally related to HDTV. Some consideration of recent thoughts on future extensions to HDTV standards are included in one document recommended for study titled "Extensibility". Convertibility (between various standards), scalability (capable of being placed in a graduated series, ascending or descending) and extensibility (capable of being extended to higher performance) are all addressed in this document.

The impact on Point-to-Point HDTV systems may be speculated considering what is now likely for Broadcast HDTV and what has already been accomplished for lower resolution point-to-point systems (even though still maturing). The fact that the broadcast HDTV transmission system is likely to be digital places both types in the same digital category since most present point-to-point systems (for non-broadcast use) are compressed digital systems. Therefore, the electronic components - special purpose integrated circuits, primarily - that will be developed to support digital compression and transmission for broadcast use can also likely be employed for point-to-point use.

As already indicated, the architectural organization of the hybrid motion compensated transform coded video compression system has served several levels of image quality and is now known to be adaptable to transmission rates at the low end of 64 Kbps and broadcast quality rates of 5 Mbps, a range of about 100. More recently excellent quality HDTV motion imagery at 20 to 30 Mbps has been demonstrated. Also, the integrated circuits which have been designed to support this method, the DCT and Motion Estimation tasks as well as others, have the capability of operating to rather high frequencies - 30 MHz. to 40 MHz. - to accommodate a broad range of NTSC and PAL based applications. These same IC's can be used in parallel with each other to accommodate the rates required for HDTV. In a sense hardware is already here and the standard is still awaited. It can be expected that IC's will later be combined by their manufacturers to provide both more integrated functionality as well as even higher speed capability. For example, LSI Logic presently builds a chip set, the L647XX which provides all of the elements of a video compression encoder except the Frame Store and its controller, input buffering and color space conversion (if required), a Motion Compensator, a Rate Buffer and a few other things. The chip set consists of 7 IC's at an encoder and 4 at a decoder. There are seven different types. LSI Logic is now planning the same functionality in a smaller set of chips. Other manufacturers supply some of the same functions and at least one manufacturer already supplies an integrated group of functions in a single package, although the particular combination may be too constraining for some uses. One could certainly build a HDTV compression system today for point-to-point applications using electronic components already available for already available HDTV camera and display equipment in the 1125/60/2:1 format. However, at this time, it would have to be without benefit of any common standard.

The issue of common world standards for HDTV unfortunately does not appear to be making much progress. Countries with the PAL standard prefer a HDTV technical standard with line and frame rates directly related by integer factors to present PAL rates, and countries with NTSC standards prefer similar relationships with present standards. Thus ATSC is favoring a system with 1050 lines, aspect ratio 16:9, a 59.94 Hz field rate and 2:1 interlace. A non-interlaced format is being considered. The European EUREKA standard calls for 1250 lines with a 50 Hz field rate and 2:1 interlace. The MUSE system which has been originated and implemented in Japan for direct broadcast satellite analog transmission employs 1125 lines with 60 fields and 2:1 interlace.

A point of hope for future HDTV international teleconferencing standardization can be found int he recently adopted CCITT Recommendation H.261 which is applicable to bit rates in the P x 64 Kbps hierarchy with values of p from 1 to 30. It establishes a Common Interface Format (CIF) for teleconferencing transmission which can be adapted to any local standaid and uses a motion compensated compression algorithm with adaptive frame rate. It stands to reason that a somewhat similar format and digital algorithm can be developed to satisfy the requirements of HDTV. It would also use P x 64 Kbps bit rates for transmission with the value of p going up to 60 and possibly higher, depending on the definition and motion rendition requirements imposed on the picture.

To summarize, it is useful to list the key organizations contributing to the standardization of HDTV along with the status of their standardization effort.

- **Federal Communication Commission (FCC)**
  - Purpose: Develop federal policy regarding communication; e.g. frequency spectrum allocation.
  - Status: Has declared that (1) no additional spectrum will be allocated to HDTV, (2) HDTV signals shall be "simulcast", and (3) it will consider only HDTV proposals for ATV service until an HDTV standard is achieved.

- **Advanced TV Systems Committee (ATSC)**
  - Purpose: Develop voluntary standards for HDTV.
  - Members: Electronic Industries Association; IEEE; National Association of Broadcasters; National Cable TV Association; Society of Motion Picture and TV Engineers; etc.
  - Status: Coordinating the evaluation of the six proposals shown in Figure 2.1.

- **Advanced TV Test Center (ATTC)**
  - Purpose: Testing of ATV systems
  - Members: National Association of Broadcasters; Association of Maximum Service Telecasters; Association of Independent TV Stations; Capital

Cities/ABC; CBS; NBC; PBS; etc.

- Status: Will perform tests on proposed HDTV systems according to the schedule in Figure 2.1.

- **Society of Motion Picture and TV Engineers (SMPTE)**
  - Purpose: Technical society developing voluntary recommedned standards and practices.
  - Members: Various end users, manufacturers, and individual engineers from the TV and film industry.
  - Status: Developed the standard 240M (1125 lines, 60 fields/sec) for an analog HDTV signal which is used extensively in TV production studios in the U.S. and elsewhere. Has developed a draft standard for a digital version of 240M.

- **International Radio Consultative Committee (CCIR)**
  - Purpose: Develop Recommendations in the technical area of radio communications for use on a worldwide basis.
  - Members: Tha CCIR is part of the United Nations. Each nation which is a member of the CCIR has one vote.
  - Status: Attempting to harmonize the conflicting technical/political objectives of the European community with the objectives of other nations of the world. It appears that a single worldwide HDTV standard will not be developed in the immediate future.

- **Consultative Committee for International Telegraph and Telephony (CCITT)**
  - Purpose: Develop international recommendations for telecommunications.
  - Members: Same as CCIR.
  - Status: The CCITT has issued Recommendation H.261 (see Section 5.0) which defines a video coding algorithm which shows promise for application to HDTV coding. The CCITT has recently created a new experts group to develop a video coding algorithm for TV transmission over the B-ISDN. One objective of this compression technique is the coding of HDTV signals.

- **International Standards Organization (ISO)**
  - Purpose: Develop international standards in the areas of computers and

communication.

- Members: Industrial organizations
- Status: ISO has an organization called the Motion Picture Experts Group (MPEG) which recently finalized a standard, known as MPEG 1, for coding TV with VCR quality at 1.5 mbps. This coding algorithm used in this standard is similar to the coding technique used in H.261 and proposed for HDTV.

3.0 INVESTIGATION OF IMAGE COMPRESSION TECHNIQUES

3.1 Overview

Figure 3.1.1 is a functional block diagram of a generic system which digitally transmits images over a communication channel. At the transmitter, the input analog signal is first filtered such that the upper cut off frequency of the signal is N cycles/sec. The filtered signal is next sampled at a rate of at least 2N samples per second (the Nyquist rate) to avoid aliasing distortion. Each sample is defined as a pixel (picture element) which is commonly encoded with 8-bit accuracy because this precision is required to avoid any visible distortion in the output image. At this point the bit rate is typically 16N bits/sec. which may exceed the bit rate of the transmission channel (C bits/sec). The purpose of the compressor is to reduce the 16N bit rate by reducing the pixel-to-pixel redundancy inherent in the image. The channel coder (e.g. modem) processes the binary compressed signal for efficient transmission over the communication channel. The compressor is commonly referred to as a source coder (signal source) as contrasted with the channel coding process. As shown in Figure 3.1.1, the functions at the receiver are the inverse of those at the transmitter.

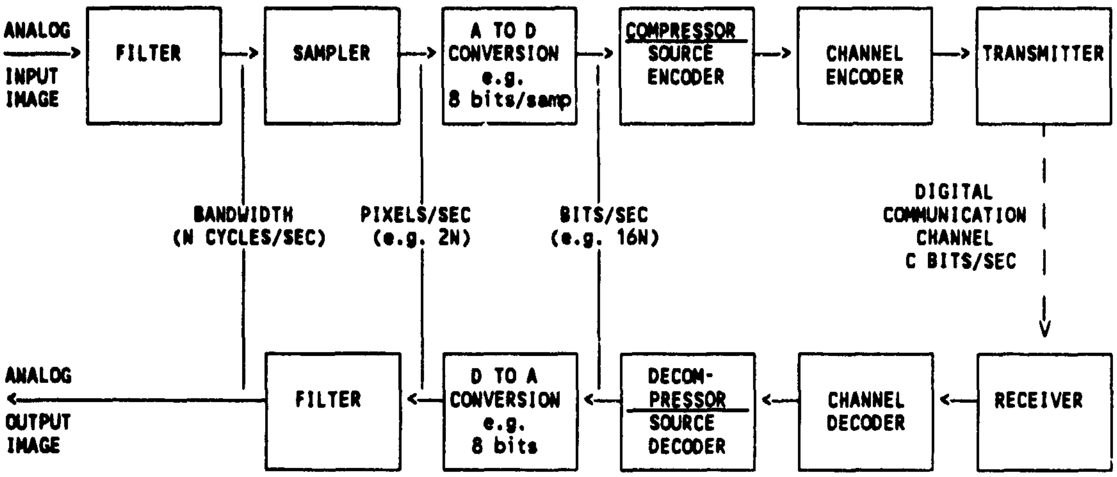


FIGURE 3.1.1
A GENERIC SYSTEM FOR THE DIGITAL TRANSMISSION OF IMAGES

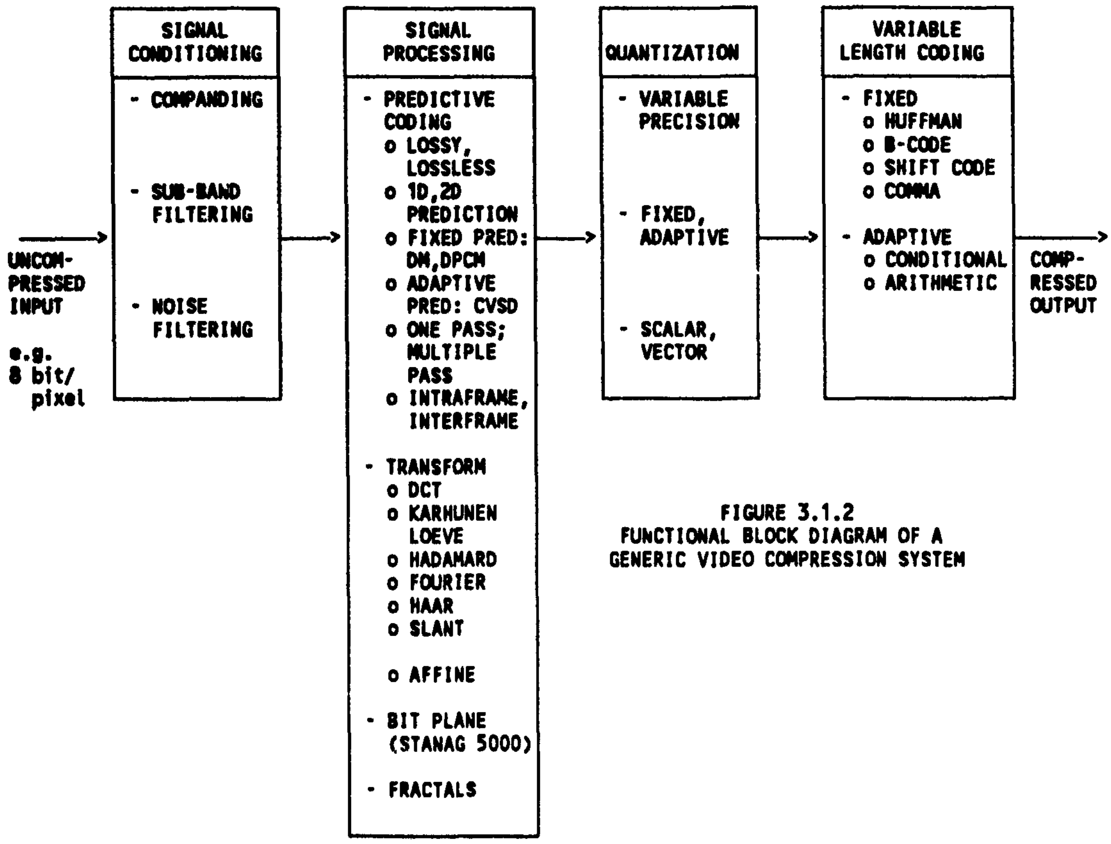


FIGURE 3.1.2
FUNCTIONAL BLOCK DIAGRAM OF A
GENERIC VIDEO COMPRESSION SYSTEM

Figure 3.1.2 is a functional block diagram of a generic compression system illustrating the various compression techniques which could be applied to video signals. The diagram shows that any image compressor can be viewed as having four sequential functions: signal conditioner, signal processor, quanitzer, and variable length coder. The purpose of the signal conditioner is to prepare the input uncompressed signal for the subsequent coding process. The Signal Processing (SP) function is probably the heart of the overall compression subsystem. In the case of Predictive Coding, the SP performs the prediction function. In the case of Transform Coding, the SP performs the transform function. In these two particular cases the output of the SP is a prediction error signal and transform coefficients respectively. In all cases the SP output signal is quantized for transmission. The output of the quantization process is a series of binary codes or words each defining a single pixel or block of pixels. These codes are not equally probable, i.e. redundancy exists. At this point variable length coding (VLC) is employed to reduce this redundancy. Short codes are assigned to likely events, and longer codes are assigned to unlikely events. VLC is a lossless, transparent process which does not degrade the coding accuracy.

The Signal Conditioning and the Variable Length Coding functions are universally used in all compression systems. Therefore, they are discussed first in Sections 3.2 and 3.3 respectively. Five particular coding algorithms which are potentially applicable to the EOVS system are then presented in the next five sections. The first four techniques are intraframe coding algorithms; i.e. they reduce redundancy of adjacent pixels within a TV picture. The fifth technique (interframe) reduces redundancy of pixels in adjacent frames as well. Conclusions are presented in the last section.

Section	Title
3.4	Differential Pulse Code Modulation (DPCM)
3.5	Transform Coding
3.6	Vector Quantization
3.7	Bit Plane Coding
3.8	Interframe DCT Coding
3.9	Conclusions

3.2 Signal Conditioning

Signal conditioning techniques are able to be cascaded with each other and with the subsequent coding techniques. Sub-band filtering could be advantageous because the signal may have different properties in the various frequency bands which could be most efficiently encoded by different compression algorithms. Companding is the name for a general process wherein the transfer function of the input signal is adjusted for optimum compression -- not too small, not too large to cause limiting. This gain adjustment may appear trivial, but it is difficult to do well. If the input transfer function is linear it is frequently desirable to modify (compand) the signal so that low level signals are encoded more precisely than high level signals. This is commonly done to match the logarithmic characteristic of the eye.

3.3 Variable Length Coding

Variable Length Coding (VLC), also called Entropy coding, is a technique whereby each event is assigned a code that may have a different number of bits. In order to obtain compression, short codes are assigned to frequently occurring events, and long codes are assigned to infrequent events. The expectation is that the average code length will be less than the fixed code length that would otherwise be required. If all events are equally likely, or nearly so, then VLC will not provide compression.

All codes considered must be uniquely decodable; that is, there must be only one way that a concatenation of VLC's can be decoded. In addition, it is highly desirable that the code be instantaneous; that is, each code word can be decoded without reference to subsequent code words. Taken together, these requirements mean that no code word can be the beginning of another code word. For example, we may not have 01 and 0110 as code words, since the second code word starts with the first code word. In decoding, it is not known whether 01 is the first code word, or just the start of the second code word.

A major advantage of VLC is that it does not degrade the signal quality in any way. That is, the reconstituted signal will exactly match the input signal so that if the signal is adequately described by a series of events, using VLC's to communicate them to the decoder will not change the events. Therefore the system is transparent to the VLC used.

The disadvantage of VLC's is that they only provide compression in an average sense. Therefore, sometimes the code could be longer for a specific section of signal. This characteristic gives rise to the need for a buffer to match the variable rate of bit generation with the fixed bit rate of the communication channel, and a control strategy to prevent long-term overflows or underflows of the buffer. Also the establishment of frames or packets of data becomes more difficult with VLC's.

Seven VLC's will be discussed here. They are Comma codes, Shift codes, B codes, Huffman codes, Conditional, Coded Arithmetic codes, and Two dimensional codes.

3.3.1 Comma Code

The Comma Code is the simplest of the VLC's. It assigns to each event a different length of code, starting at 1. A particular bit polarity marks the end of the code word. For example:

Code
0

10
110
1110
11110
111110
.
.
.

The advantage of the Comma Code is that it is simple to generate and decode, requiring only counters to count the number of ones. However, it is rare that this code accurately matches the statistics of the events, so it is used primarily where simplicity of implementation is important.

3.3.2 Shift Code

In the case where the probabilities of the events decrease monotonically as the magnitudes increase, a great simplification can be obtained by the use of a systematic VLC, such as a Shift Code. In this code, each code word consists of a series of sub-words, each of length L bits. The first sub-word is capable of conveying 2^L values, one of which is a shift that indicates that the value of the code word is contained in the following sub-word. In this way, any length of code word can be obtained by concatenating a number of sub-words together.

The following are examples of the beginnings of Shift Code tables for L = 1, 2, and 3, where a sub-word of all ones indicates a shift.

L = 1	L = 2	L = 3
0	00	000
10	01	001
110	10	010
1110	1100	011
11110	1101	100
111110	1110	101
1111110	111100	110
11111110	111101	111000
111111110	111110	111001

1111111110	11111100	111010
11111111110	11111101	111011
.	.	.
.	.	.
.	.	.

Note that for L=1, the Shift Code reduces to the Comma Code. The Shift Code is best suited to cases where the probabilities drop off rapidly, since the number of codes available only increases linearly with the length of the code. For example, for L=3, there are 7 codes with length 3 ($2^3 - 1 = 7$). Increasing the length to 6 only adds 7 more codes.

3.3.3 B Code

Another variable length code that is systematic is the B Code. Again the code consists of a sequence of sub-words, each of length L. But in this case, one bit of the sub-word is used to designate whether another sub-word is to be added to the code word. Therefore the remaining L-1 bits in the sub-word can be used as part of the code. For L=1, the B Code also reduces to the Comma Code.

The following are examples of the beginning of B Code tables for L = 1, 2, and 3:

L = 1	L = 2	L = 3
***********	* * * *	* * *
0	00	000
10	01	001
110	1000	010
1110	1001	011
11110	1100	100000
111110	1101	100001
1111110	101000	100010
11111110	101001	100011
111111110	101100	101000
1111111110	101101	101001
11111111110	111000	101010

.	111001	101011
.	111100	110000
.	111101	110001
	10101000	110010
	10101001	110011
	10101100	111000
	10101101	111001
	10111000	111010
	10111001	111011
	10111100	100100000
	.	.
	.	.
	.	.

In this table, the * marks the columns containing the continuation bits, where '1' indicates continue and '0' marks the last sub-word of the code word. This version of the B Code is instantaneous. In another version, the continuation bit is the same value for all sub-words in the code word, but alternates with each succeeding code word. That version is not instantaneous.

The B Code is best suited to cases where the probabilities drop off slowly, since the number of codes available increases geometrically with increasing code length. For example, for $L=4$, there are 8 codes with length 4 (2^{L-1}). Increasing the length to 8 increases the number of codes by 64 ($2^{2(L-1)}$).

3.3.4 Huffman Code

The Huffman code is a VLC that provides the shortest average code length for a given distribution of input probabilities. The method for generating the code is well known, but a distribution of input probabilities, either theoretical or measured, is required before the code words can be calculated. If the actual distribution differs from the one used to calculate the code, then the average code length may not be less than other codes. If a large enough sample can be obtained to measure the distribution accurately, the Huffman Code may be an attractive choice. In any event, it provides a reference against which other codes can be compared, if the distribution is measured on the image being coded.

3.3.5 Conditional Variable-Length Codes

In general the most likely sample values to be encoded are near zero, and therefore the small values are given the shortest codes. However, zero is the most likely sample value only in the absence of information about other samples. If the values of neighboring samples are known, then the distribution of the current sample value can be markedly changed. In the simplest case, only the previous sample is used. Although in principal other samples can be used for each value of the previous sample, the frequency of occurrence of each of the current samples can be obtained, and a set of VLC's devised for each. Since both the encoder and decoder know the value of previous sample, decoding can take place without significant delay.

3.3.6 Arithmetic Coding

In arithmetic coding, the frequency of occurrence of the symbols to be coded is continuously measured by both the encoder and decoder. In the resulting code, there is not a one-to-one correspondence between the events and specific bits. It is possible to generate arithmetic codes at a rate of less than one bit per event, whereas a Huffman code requires at least one bit per event. Since arithmetic coders adapt dynamically to the statistics of the image being transmitted, the compression is generally superior to that for conventional non-adaptive VLC's.

3.3.7 Two-Dimensional VLC for Coding Transform Coefficients

A VLC has recently been developed which is particularly designed to code transform coefficients. It is a two-dimensional code where the two dimensions are: (1) the number of zero-value coefficients in a row (usually from a zig-zag scan) and, (2) the value of the next non-zero coefficient. This VLC has proven to be particularly efficient and would probably be used in the NSCS system if a DCT approach is employed.

3.4 Predictive Coding

PCM transmits each pixel as an independent sample without taking advantage of the high degree of pixel-to-pixel correlation existing in most pictures. Predictive coding is a basic bit-rate reduction technique that reduces this pixel-to-pixel redundancy. Figure 3.4.1 is a functional block diagram illustrating the basic predictive coding process. A predictor predicts the brightness value of each new pixel based solely on the pixels previously quantized and transmitted. The predicted brightness is subtracted from the actual value of the new pixel, resulting in a bipolar prediction error signal. This error signal is quantized and transmitted. This quantization process can vary over a wide range of complexities. The quantization may be fixed, or it can adapt to the data. The quantizer can also vary over a wide range of accuracies. If one-bit quantization is employed, the system becomes the well-known Delta Modulation technique. If the predictive quantizer employs multiple bits per pel, the technique is commonly defined as Differential PCM (DPCM). At the receiver, the inverse of the quantization process is performed, and the decoded error signal is added to the predicted value to form the output signal for viewing. The output signal is fed to the predictor to be used for prediction of the next pixel. Referring back to the predictive encoder, the reader will note that the transmitted signal is decoded at the transmitter using exactly the same decoding process which is used at the receiver. The predictive encoder can be viewed as a servo loop which continually forces the decoded output signal to be as close as possible to the input signal.

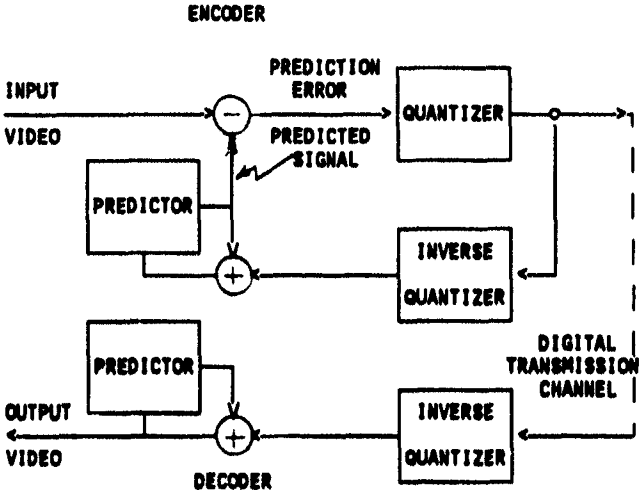


FIGURE 3.4.1
FUNCTIONAL BLOCK DIAGRAM OF A GENERIC PREDICTIVE CODING SYSTEM

Figure 3.4.2 illustrates the transfer function of a typical three-bit DPCM predictive coder. The quanitzer is usually nonlinear because the eye is very sensitive to small changes in low detail portions of a picture (small prediction error), but the eye is insensitive to coarse quantization of high-contrast edges (large predictive error). The design of this quantizer is a compromise between conflicting objectives. It is desirable that the quantizer precision be fine, particularly for small error signals, to keep the background granular noise in the output picture at an acceptably low level. On the other hand, the quantizer steps

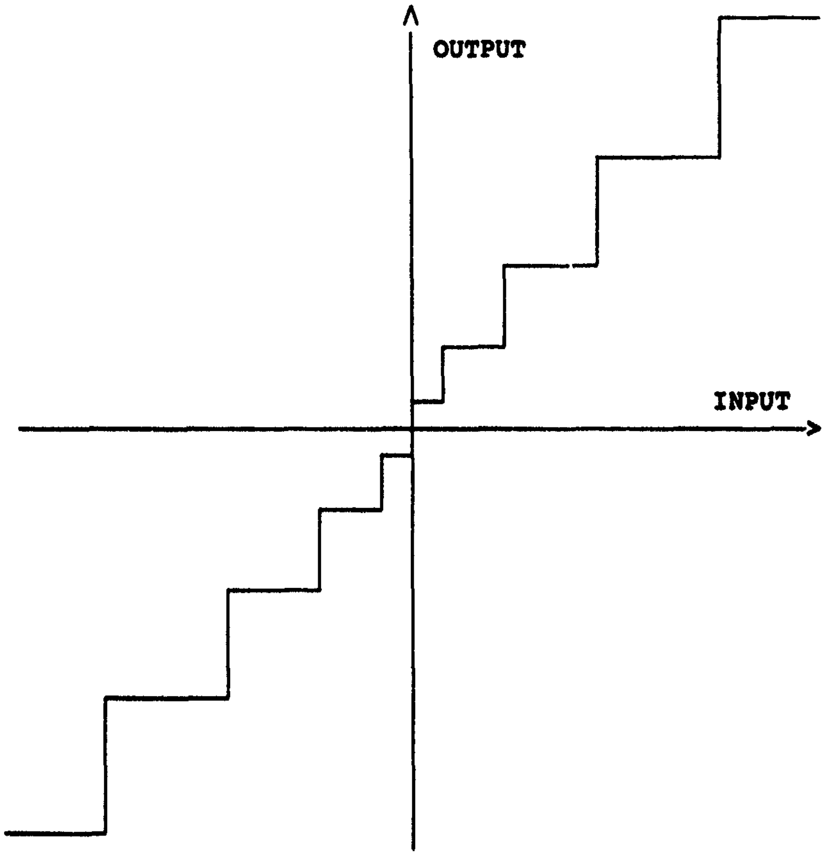


FIGURE 3.4.2
QUANTIZER FOR PREDICTIVE ENCODER

must be large enough, particularly the largest increment, so the output can respond reasonably well to high-contrast changes in the input picture. if the largest increment is too small, slope overluad occurs resulting in picture blurring.

3.5 Transform Coding

Transform coding algorithms, generally speaking, operate as two step processes. In the first step a linear transformation of the original signal (separated into sub-blocks of N x N pels each) is performed, in which signal space is mapped into transform space. In the second step, the transformed signal is compressed by encoding each sub-block through quantization. The reconstruction operation involves performing an inverse transformation of each decoded transformed sub-block. The function of the transformation operation is to make the transformed samples more independent than the original samples, so that the subsequent operation of quantization may be done more efficiently.

The transformation operation itself does not provide compression; rather, it is a re-mapping of the signal into another domain in which compression can be achieved more effectively. Compression can be achieved for two reasons. First, not all of the transform domain coefficients need to be transmitted in order to achieve acceptable picture quantity. Second, the coefficients that are transmitted can be encoded with reduced precision without seriously affecting image quality.

3.5.1 Transformation Techniques

Transforms that have proven useful include the Karhunen-Loeve, Discrete Fourier, Discrete Cosine, and Walsh-Hadamard transforms. The Karhunen-Loeve transform (KLT) is considered to be an optimum transformation, and for this reason many other transformations have been compared to it in terms of performance. However, the KLT has certain characteristics that make it less than ideal for image processing. These include the necessity to estimate the covariance matrix before processing in both row and column operations. Also, the actual eigenvector determination must be carried out to generate the basis matrix. These drawbacks would not be significant if the efficiency of the KLT was much greater those that of other transforms. However, for data having high inter-element correlation, the performance of other transforms (such as the Discrete Cosine transform) is virtually indistinguishable from that of the KLT, and thus usually does not warrant its added complexity.

The Discrete Fourier Transform (DFT) is one of the few complex transforms used in data coding schemes. There are disadvantages in using a complex transform for data coding, the most obvious of which is the storage and manipulation of complex numbers. Again, as in the case of the KLT, this complexity issue would not be a factor if the performance of the DFT was significantly greater than that of other transforms. However, other transforms which are less complex perform better than the DFT.

The Discrete Cosine Transform (DCT) is one of an extensive family of sinusoidal transforms. In their discrete form, the basis vectors consist of sampled values of sinusoidal or cosinusoidal functions that, unlike those of the DFT, are real number quantities. The DCT has been singled out for special attention by workers in the image processing field, principally because, for conventional image data having reasonably high inter-element correlation, the DCT's performance is virtually indistinguishable from that of other transforms which are much more complex to implement.

The three transforms mentioned previously have basis functions which are either cosinusoidal, i.e. the fourier and Discreτe Cosine, or are a good approximation of a sinusoidal function, such as the Karhunen-Loeve Transform. The Walsh-Hadamard Transform is an approximation of a rectangular orthonomal function. The actual transform consists of a matrix of +1 and -1 values, which eliminates multiplications from the transform process. The elimination of multiplications is a significant property, since the aforementioned transforms require real or complex multiplications. However, the Walsh-Hadamard transform does not provide the excellent performance that the Discrete Cosine Transform provides.

Since the Discrete Cosine Transform is universally accepted as the preferred transform for image coding it is useful to provide more detail on its implementation. The execution of the Discrete Cosine Transform algorithm requires the division of an image into a series of (NxN) sub-blocks of pixels. Each sub-block is transformed by a two-dimensional (NxN) Discrete Cosine Transform process as follows:

$$[T] = [C] \cdot [D] \cdot [C]^T$$

where [T] is the transformed sub-block, [C] is the DCT basis matrix, and [D] is the input data sub-block ($[C]^T$ is the transpose of the DCT basis matrix). The DCT basis matrix coefficients were determined from the following relation:

$$C_{ij} = C_0 \cdot \sqrt{(2/N)} \cdot \{\cos[i \cdot (j + 0.5) \cdot (\pi/N)]\}$$

where $C_0 = \sqrt{1/2}$ for i = 0, $C_0 = 1$ otherwise, and i,j=0 to N-1. Figure 3.5.1 illustrates the basis functions for an 8 x 8 DCT. This transformation converts each (NxN) sub-block of pixels into an (NxN) matrix of transform coefficients, which consists of one DC coefficient and (NxN - 1) AC coefficients. The sum of the squares of all of the AC coefficients in a given transform matrix is known as the AC energy of that transform matrix.

3.5.2 Coding of Transform Coefficients

As explained in the previous section the Discrete Cosine Transform is usually used when pictures are transmitted using transform techniques. This transformation merely creates a set of coefficients equal in number to the original set of pels. At this point no compression has been accomplished except that the original set of pels with uniformly high redundancy have been decorrelated, and the

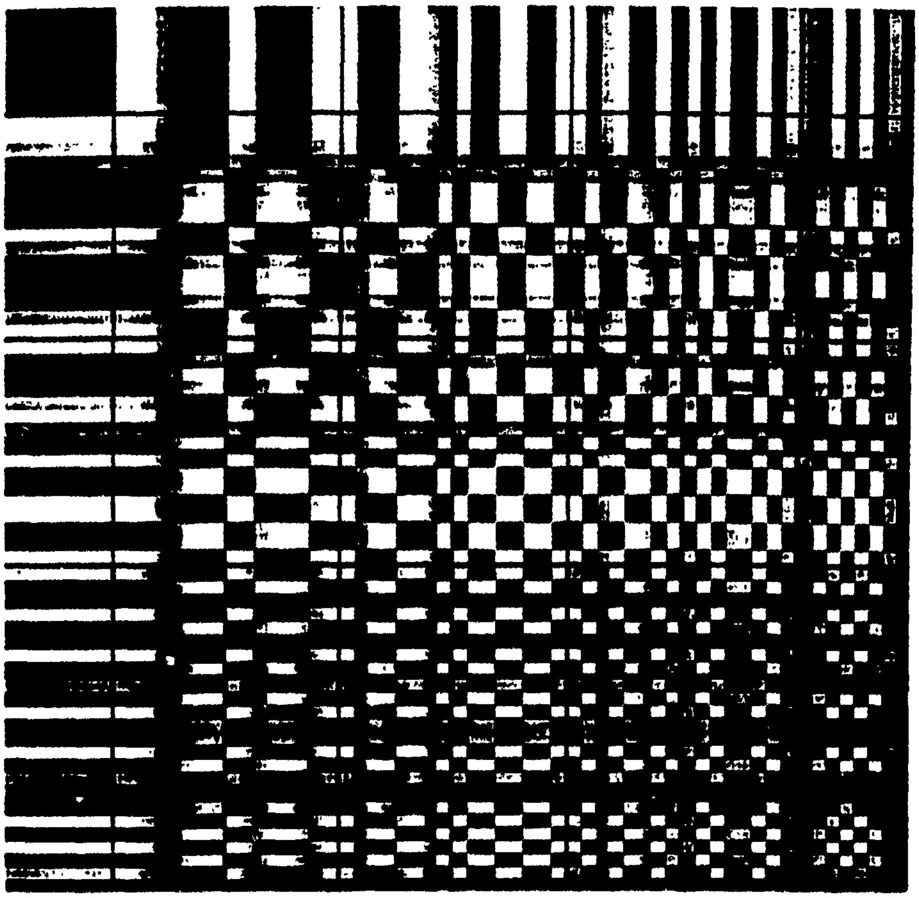


FIGURE 3.5.1
DCT BASIS FUNCTIONS

information has been compacted in the lower spatial frequency coefficients. The purpose of this section is to address the second part of the two step process; how to encode the transform coefficients for transmission.

The first step in the coding process is to determine which coefficients are to be transmitted and which are to be deleted. Figure 3.5.1 illustrates the set of 64 transform coefficients corresponding to a 8 x 8 block of pels to be coded. Coefficient number one is the DC coefficient which is a measure of the average brightness of the block. Coefficients in the top row measure of spatial frequency content in the horizontal direction. Coefficients in the left column measure frequencies in the vertical direction, and all others measure various combinations thereof. In general, most of the energy is contained in the low frequency coefficients with relatively little signal strength in the high frequency coefficients.

75	76	77	78	79	80	81	82
77	78	79	80	81	82	83	84
79	80	81	82	83	84	85	86
81	82	83	84	85	86	87	88
83	84	85	86	87	88	89	90
85	86	87	88	89	90	91	92
87	88	89	90	91	92	93	94
89	90	91	92	93	94	95	96

a) ORIGINAL BLOCK (8x8x8 = 512 BITS)

76	76	77	79	80	81	82	83
77	77	78	80	81	82	83	84
79	79	80	81	83	84	85	86
81	82	83	84	85	87	88	88
84	84	85	87	88	89	90	91
86	87	88	89	91	92	93	93
88	89	90	91	92	94	95	95
89	90	91	92	93	95	96	96

f) RECONSTITUTED BLOCK

684	-19	-1	-2	0	-1	0	-1
-37	0	-1	0	0	0	0	-1
0	0	0	0	0	0	0	0
-4	-1	-1	-1	-1	0	-1	-1
0	0	0	0	0	0	0	0
-2	0	0	-1	0	-1	0	-1
0	0	0	0	-1	-1	-1	-1
-1	-1	-1	0	-1	0	-1	0

b) TRANSFORMED BLOCK COEFFICIENTS

688	-21	0	0	0	0	0	0
-39	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

e) INVERSE QUANTIZED COEFFICIENTS

86	-3	0	0	0	0	0	0
-6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

c) QUANTIZED COEFFICIENT LEVELS

RUN	LEVEL	CODE
0	86	01010110
0	-3	001011
0	-6	001000011
	EOB	10

TOTAL CODE LENGTH = 25

d) COEFFICIENTS IN ZIG-ZAG ORDER AND VARIABLE LENGTH CODED

FIGURE 3.5.2
SAMPLE INTRA BLOCK CODING

Figure 3.5.2 shows a simple example of how each 8 x 8 block is coded. Figure 3.5.2a shows the original block to be coded. The block has a constant slope or shading from the upper left corner to the lower right. Without compression, this would take 8 bits to code each of the 64 pixels, or a total of 512 bits. First, the block is transformed, using the two-dimensional Discrete Cosine Transform (DCT), giving the coefficients of Figure 3.5.2b. Note that most of the energy is concentrated into the upper left-hand corner of the coefficient matrix.

Essentially, the DCT is performed by multiplying the input block by each of the 64 basis functions shown graphically in Figure 3.5.1. The results of each of these multiplications, also 8 x 8 arrays, are summed to give the 64 transform coefficients. In the upper left-hand corner of Figure 3.5.1, the first basis function is constant over the block, and therefore gives rise to the DC value of the input block. At the opposite corner, the basis function is a checkerboard, and will give significant coefficient values only if there are elements of this pattern in the input block. Of course, the coefficients are in practice calculated by a chip in a more efficient manner than described here.

Next, the coefficients of Figure 3.5.2b are quantized with a step size of 6. (The first term {DC} always uses a step size of 8.) This produces the values of Figure 3.5.2c, which are much smaller in magnitude than the original coefficients and most of the coefficients become zero. The larger the step size, the smaller the values produced, resulting in more compression.

The coefficients are then reordered, using the Zig-Zag scanning order of Figure 3.5.3. All zero coefficients are replaced with a count of the number of zero's before each non-zero coefficient (RUN). Each combination of RUN and VALUE produces a Variable Length Code (VLC) that is sent to the decoder. The last non-zero VALUE is followed by an End of Block (EOB) code. The total number of bits used to describe the block is 25, a compression of 20:1.

At the decoder, the step size and VALUE's are used to reconstruct the inverse quantized coefficients, which, as shown in Figure 3.5.2e are similar to, but not exactly equal to, the original coefficients. When these coefficients are inverse transformed, the result of Figure 3.5.2f is obtained. Note that the differences between this block and the original block are quite small.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

FIGURE 3.5.3
SCANNING ORDER IN A BLOCK

Figure 3.5.4 shows a slightly different example that shows more clearly some of the features of DCT coding. In this example, in addition to shading, the block contains a checkerboard pattern that matches the highest order basis function. This causes the last coefficient to be transmitted. There are 60 zero-value coefficients between the previous non-zero coefficient and the last one, so the run length is 60. The last coefficient is coded as a 6-bit escape code (000001), a 6-bit run code (111110), and an 8-bit level code (00000010).

64	59	62	57	60	55	58	53
59	62	57	60	55	58	53	56
62	57	60	55	58	53	56	51
57	60	55	58	53	56	51	54
60	55	58	53	56	51	54	49
55	58	53	56	51	54	49	52
58	53	56	51	54	49	52	47
53	56	51	54	49	52	47	50

a) ORIGINAL BLOCK

440	19	-1	1	0	0	-1	0
18	0	-1	0	-1	0	0	2
0	-1	-1	0	0	0	0	-1
1	0	-1	0	-1	1	0	3
0	0	0	0	0	0	0	0
0	0	0	1	-1	1	0	4
0	0	0	0	0	0	0	0
0	2	-1	3	0	4	0	13

b) TRANSFORMED BLOCK COEFFICIENTS

55	3	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2

c) QUANTIZED COEFFICIENT LEVELS

RUN	LEVEL	CODE
0	55	00110111
0	3	001010
0	3	001010
60	2	00000111110000000010
	EOB	10

TOTAL CODE LENGTH = 42

d) COEFFICIENTS IN ZIG-ZAG ORDER AND VARIABLE LENGTH

440	21	0	0	0	0	0	0
21	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	15

e) INVERSE QUANTIZED COEFFICIENTS

62	61	61	59	59	56	56	55
61	62	58	61	55	58	54	55
61	58	62	55	59	52	56	53
59	61	55	60	51	57	51	53
59	55	59	51	57	49	53	50
56	58	52	57	49	53	48	50
56	54	56	51	53	48	50	48
55	55	53	53	50	50	48	48

f) RECONSTITUTED BLOCK

FIGURE 3.5.4
AN EXAMPLE OF DCT CODING

Two types of distortion appear in transform coded pictures: truncation error and quantizations errors. Quantization errors are noiselike whereas truncation errors cause a loss of resolution. In practice the truncation threshold and quantization precision must be adjusted experimentally to achieve the maximum compression and acceptable picture quality. In general transform coding is preferable to predictive coding for compression to bit rates below 1 or 2 bits per pel for single pictures. However in those applications where cost and complexity are important issues the choice between these two algorithms may be less clear.

3.6 Vector Quantization

Vector Quantization begins by dividing an image to be transmitted into rectangular blocks of pixels, all blocks having the same dimensions. The transmitter compares each block with a large library of typical blocks, called a "codebook," and selects the library block that best approximates the block to be transmitted. The transmitter then encodes and transmits the index to the selected library block. The receiver, equipped with a copy of the codebook, decodes the index, retrieves the selected library block and inserts it into the output image.

This process is called Vector Quantization because, both theoretically and computationally, each block is treated as a vector. The vector representation of a block can be thought of as laying out all the gray-scale values of the block pixels in a single string, that of the upper left pixel first, and of the lower right last. Such a string of numbers comprises a vector in k-dimensional space, where k is the number of pixels in the block. When the block is treated in this manner, the entire body of mathematical knowledge of vector analysis and multi-dimensional analytical geometry can be brought to bear on the Vector Quantization problem. In the balance of this discussion, the terms "block" and "vector" will be used interchangeably, with "block" referring to a rectangular array of pixels in an image, and "vector" referring to the representation of these pixels as a string of numbers.

In all the variations of Vector Quantization there is a trade-off between image quality and data compression. In the theoretical limit of zero distortion, the codebook would contain vectors representing all possible blocks. An exact match would always be found. Distortionless transmission would, however, entail an enormous codebook and little data compression, even with optimal coding. At the other extreme, a codebook containing few vectors (representative blocks) would yield large compression ratios, but poor image quality. The objective of any Vector Quantization system design is, therefore, to achieve the best compromise among codebook size, data compression and received image quality.

A review of published papers reveals many variations on the Vector Quantization theme. Gersho [1] presents a mathematical treatment of the problem. The codebook is, in effect, the vector quantizer in that it quantizes" the multi-dimensional vector "space" into a finite set of representative vectors. Gersho goes on to explore the partitioning problem, and concludes that the only practical way to design the quantizer (select the vectors to be included in the codebook) is to take advantage of vector clustering.

The basic vector clustering algorithm was thoroughly developed by Linde, Buzo and Gray [2]. This algorithm, known as the LBG algorithm, takes advantage of the fact that the vector representations of image blocks tend to cluster in the vector space. A codebook containing vectors representing the cluster centroids offers the best compromise between codebook size and received image quality. The method consists of using a long sequence of "training" vectors to design the codebook. Gray and Linde [3] explain and compare several variations on the LBG codebook generation method and compare the resulting performances of Gauss-Markov sources. In particular, the authors show that tree-assisted codebook searches allow the use of codebooks much larger than those practical with exhaustive searches at the expense of a suboptimal codebook. The performance is only slightly degraded with respect to the exhaustive-search approach. Hang and Woods [4] discuss predictive vector quantization, which consists of a combination of predictive filtering and vector quantization. The purpose of the predictive filtering is to remove redundancy before vector quantizing the residue. A vector quantization method that offers great promise of good compression and low distortion is described in Japan Annex 4 [5]. This method combines DPCM (Differential Pulse Code Modulation) and vector quantization. Other references include Helden and Boekee, [6] and Gersho and Ramamarthi [7].

Vector Quantization, in all its forms, requires a large codebook of vectors from which one is selected for each block to be transmitted. Two very important issues are therefore: (1) codebook search and (2) codebook generation. There are two basic search methods: exhaustive and tree-assisted. The exhaustive method is guaranteed to select the codebook vector that best matches the input vector. This method is practical, however, only for very small block sizes, because the search is much faster. A binary tree search begins with a choice between two codebook vectors that act as "keys" to the next search level. The selection of one of these keys leads to another two-way choice, which leads to a better approximation of the input vector, which leads to yet another two-way choice, etc.. This method, though much faster than the exhaustive search, may fail to find the best match, because once a two-way choice has been made in a given tree level, the search may be directed to a subtree that does not contain the best match. The general m-way tree search, in which an m-way choice is made at each decision level, gives better performance as the value of m increases, at the expense of longer search time. The exhaustive search is the limiting case of one M-way decision, where M is the total number of codebook vectors.

The codebook generation objective is a codebook that gives low image distortion while minimizing the codebook size. Minimizing the codebook size is important, not only to minimize memory and search time, but also to achieve high compression ratios.

All codebook generation methods reported in the literature are variations on the LBG (Linde, Buzo and Gray) method. In principle, if one knew the statistics of all images to be transmitted, one could generate a codebook analytically. The most commonly used method consists, however, of using a large number of training vectors, each training vector representing a "typical" image block.

The following is a summary of the LBG codebook generation method. Assume, for the moment, a partially optimized codebook. Each training vector "belongs" to a codebook vector in that the training vector matches the codebook vector at least as well as it matches any other. (Ties are broken in various ways depending on the specific method used.)

The codebook is updated to make each codebook vector the centroid of the set of training vectors that belong to it, thus minimizing the average distortion with respect to that set of training vectors. The update may, however, cause some of the training vectors that belonged to a given codebook vector before the update to belong to a different codebook vector afterward. Another iteration is therefore performed to compute new centroids, and the codebook is updated again. This process is repeated until there is no further improvement, or the improvement is less than some specified value.

This iterative method of codebook improvement leads to a local minimum of average distortion. A slight perturbation of the codebook vectors gives greater distortion. This method leaves the possibility that some large change to the codebook might give even less distortion; hence the local minimum is not necessarily the global minimum (best possible codebook for the training vector set).

Codebook generation begins with one codebook vector which is the centroid of all the training vectors. This vector is then split into two vectors very close to each other. The splitting objective is to make the numbers of training vectors belonging to the two codebook vectors approximately equal. The codebook is then optimized, as described above. The two (now optimized) vectors are then split into four, and optimization is repeated. The process is continued until a codebook of the required size is achieved.

The typical block size employed in Vector Quantization systems is 4 x 4 pels, and the bit-rate reductions which can be achieved is comparable to Transform coding. However the VQ technology, particularly in the area of codebook generation and search, is not fully mature and consequently few operational systems have been implemented. Nevertheless it is anticipated that VQ will play a significant role in gray scale coding in the future.

References for Vector Quantization

1 Gersho, A., "On the Structure of Vector Quantizers," IEEE Transactions on Information Theory, V. IT-28, No. 2, March 1982, pp 157-166.

2 Linde, Y., Buzo, A., and Gray, R. M., "An Algorithm for Vector Quantizer Design, "IEEE Transactions on Communications, V. COM-28, No. 1, Jan. 1980, pp. 84-95.

3 Gray. R. M. and Linde, Y., "Vector Quantizers and Predictive Quantizers for Gauss-Markov Sources," IEEE Transactions on Communication, V. COM-30, No. 2, February, 1982, pp. 381-389.

4 Huang, H-M and Woods, J. W., "Predictive Vector Quantization of Images," IEEE Transactions on Communication, V. COM-33, No. 11, November 1985, pp. 1208-1219.

5 "Component Vector Quantization," Annex 4 of CCITT Study Group VIII, Geneva, 1-12 December 1986.

6 Helden, J. and Boekee, D. E., "Vector Quantization Using a Generalized Tree Search Algorithm," Proc. 5th Symposium on Information Theory in the Benelux, Aalten, May 1984, pp. 21-27.

7 Gersho, A. and Ramamurthi, B., "Image Coding Using Vector Quantization," Proc. ICASSP, 1982, Paris.

3.7 Bit Plane Coding

Most of the picture coding techniques described in previous sections are inexact in that they do not usually transmit an exact replica of the original PCM picture. Bit Plane Coding (BPC) is usually a lossless coding technique which does exactly reproduce the input image. BPC requires the storage of at least one complete scan line at the transmitter prior to encoding and at the receiver after decoding. Consider the case where a 4 bit PCM image is to be transmitted. In BPC the 4 bits for all the pels in a scan line are not transmitted pixel-by-pixel, but sequentially in accordance with the coding precision. First all the most significant bits of all the pels in the line are transmitted. This is defined to be the most significant bit "plane". Then the second most significant bit plane is transmitted. And so on until all four bit planes are transmitted. Bit-rate reduction is achieved because each plane is encoded for transmission using a binary image compression technique. At the receiver, all planes are reassembled in the normal multi-bit-per-pel word structure such that the image can be displayed.

The NATO countries have adopted a standard for coding gray scale images which is known as Stanag 5000. Pixels are transmitted using the two-line wobble pattern illustrated in Figure 3.7.1. The coding technique is the 4-bit BPC concept described above. Each pixel is defined using the Gray code in Figure 3.7.2 rather

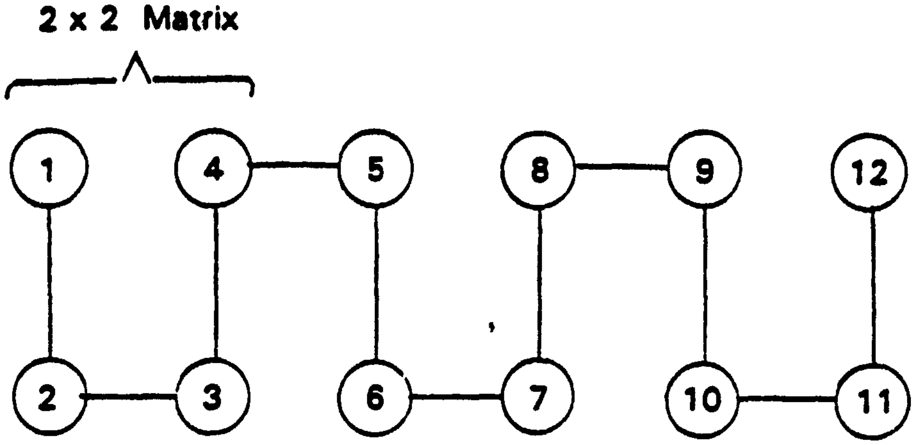


FIGURE 3.7.1
PELS USED FOR AUTORESOLUTION

than the conventional Binary code. Compression is increased further by adaptively reducing the resolution in particular portions of a bit plane. Each 2 x 2 matrix of pels within the wobble scan (see Figure 3.7.1) is examined to determine whether there is high detail or low detail present. If there is little detail the block of four pels in transmitted as a single pel. The criteria for use of the low resolution mode is provided below.

- Bit plane 1 (most significant) -- Low resolution never used;
- Bit plane 2 -- Low resolution used if transition threshold is not exceeded;
- Bit plane 3 -- Low resolution used if transition threshold not exceeded or if bit plane 2 uses low resolution;
- Bit plane 4 -- Low resolution always used.

Intensity Level	Normal Binary				4-bit Gray Code			
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0	1
2	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	0	1	0
5	0	1	0	1	0	0	0	0
6	0	1	1	0	1	0	0	0
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	0	0	1	1
10	1	0	1	0	0	1	1	1
11	1	0	1	1	1	1	1	1
12	1	1	0	0	1	1	1	0
13	1	1	0	1	1	1	0	0
14	1	1	1	0	1	1	0	1
15	1	1	1	1	1	0	0	1
	1	3	7	15	3	4	4	4

FIGURE 3.7.2
FOUR-BIT GRAY CODES

3.8 Interframe DCT Coding

The four coding techniques described above reduce only intraframe redundancy; i.e. reduce correlation of a pixel relative to its neighbors within the frame. For the HDTV application, it is important to achieve a very high level of compression. Therefore, it is desirable to consider techniques which reduce frame-to-frame redundancy as well as intraframe.

One promising interframe coding system combines the features of predictive coding (Figure 3.4.1) and the DCT. A functional block diagram for such a system is shown in Figure 3.8.1. Basically the system subtracts a predicted block of 8 x 8 pixels from the corresponding block of incoming video. A block of error pixels is generated and fed to the DCT encoder, quantizer, and VLC for transmission. At the receiver the error block is decoded and added to the predicted block for viewing. Since the predicted block and incoming video block are highly correlated the error block will tend toward zero and be encoded with few bits.

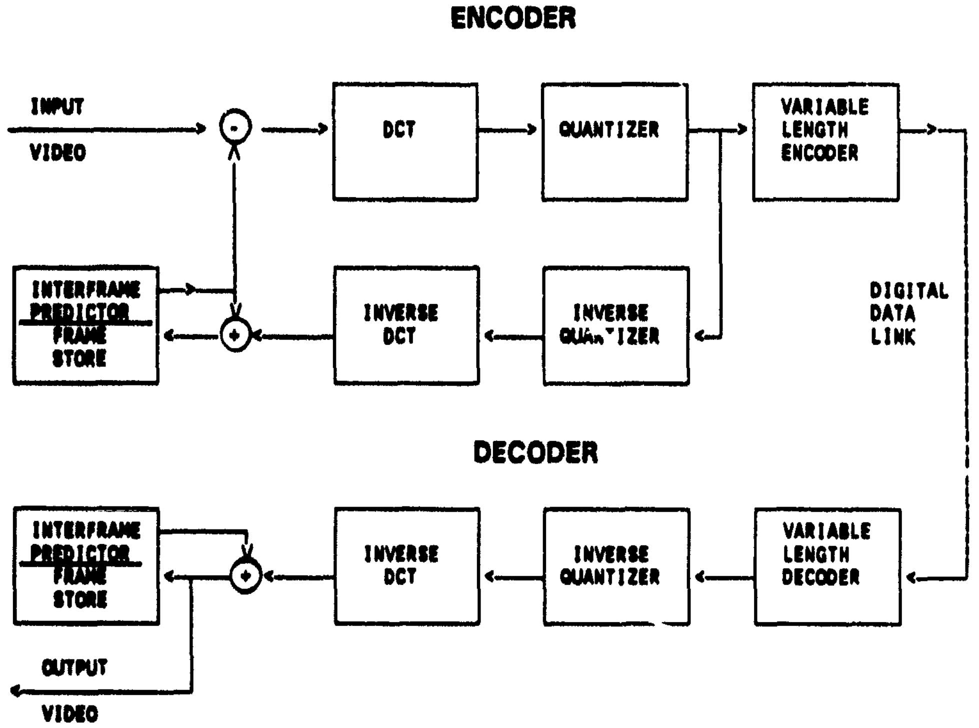


FIGURE 3.8.1
FUNCTIONAL BLOCK DIAGRAM OF AN
INTERFRAME CODEC USING PREDICTIVE AND DCT CODING

The system measures the magnitude of the error signal on a block basis. If the error signal is below a threshold no information would be transmitted about that block with a resultant very high compression ratio. Table 3.8.1 illustrates the budget for the bits which may be allocated to encode a typical TV frame consisting of 256 pixels/line and 240 lines.

The reader will note that approximately 14% of the bits are used for intraframe coding. This is necessary because if a block encoded in the interframe mode is contaminated by a transmission error the distortion from that error could be retained indefinitely. To correct this defect each block is transmitted by intraframe coding every two seconds.

It is concluded in Table 3.8.1 that 3,584 bits of information are needed to define each typical TV frame. This bit count must be increased by approximately 3% to account for the overhead structure. It is also desirable to include forward error control (FEC) to make the transmitted signal more robust. The typical FEC overhead is 4%. These two sources of overhead would increase the total bit count per frame to 3,839 which yields a net coding rate of .06 bits/pixel.

TABLE 3.8.1 BIT ALLOCATION FOR CODING A TYPICAL TV FRAME

CODING MODE	BLOCKS	BITS/PIXEL	BITS/FRAME
BLOCKS NOT TRANSMITTED	752 (78%)	-----	-----
BLOCKS INTERFRAME CODED	192 (20%)	.25	3,072
BLOCKS INTRAFRAME CODED	16 (2%)	.5	512
TOTAL	960 (100%)		3,584

3.9 SUMMARY

TV compression technology has been reviewed in general and the five coding techniques listed below have been discussed in some detail.

CODING TECHNIQUE	CODING RATE TO PROVIDE ESSENTIALLY EQUIVALENT PICTURE QUALITY (BITS/PIXEL)
DPCM	1.5
INTRAFRAME DCT	0.5
VECTOR QUANTIZATION	0.5
BIT PLANE	1.5
INTERFRAME DCT	0 06

Since interframe DCT provides a higher level of compression than the intraframe technique, it is most promising for the HDTV application.

4.0 COMPUTER SIMULATION OF SUB-BAND CODING

Sub-band coding is a relatively old concept for compressing pictures when considering earlier systems such as split-band coding. There has been a renewed interest in this class of coding for potential applications to HDTV. This heightened interest is based on improved filtering technology (e.g. Quadrature Mirror Filters) and the filtering of the input signal into many more than two bands.

In March 1990, Bellcore made a presentation of a comprehensive concept for the digital coding of HDTV signals, for the purpose of transmitting them over B-ISDN networks. The details of their concept are presented in Reference 1. In order to evaluate this proposal, the sub-band algorithm was simulated using the Aerial image.

4.1 Description of Sub-band Coding Algorithm

Sub-band coding belongs to the class of transform coding. As such, it bears some similarity to DCT coding. This algorithm can be divided into the following parts:

a) Pre-filtering
b) Sub-band filtering into six bands by means of Quadrature Mirror Filters (QMF)
c) Decimation to provide the proper number of coefficients per block
d) Differentially coding the Band 1 coefficients
e) Non-linear quantization of each coefficient
f) Run-length coding of zeros
g) Variable Length Coding (VLC) of runs and quantized coefficients

Pre-filtering is used to reduce those high frequency components that are so high that they are beyond the ability of the eye to percieve them, and results in more efficient coding.

The image is divided into contiguous blocks that are 8 pixels wide by 2 pixels high, as shown in Figure 4.1. By successive QMF, in horizontal and vertical directions, the block in pixel format is transformed into six sub-bands in the two-dimensional frequency domain, as shown in Figure 4.1. Note that there are still 16 coefficients, the same number as the number of pixels in the block.

```
           SPATIAL DOMAIN                      SPECTRAL DOMAIN

          1 2 3 4 5 6 7 8
         |---------------|                  |---------------|
    1    | | | | | | | | |                  |1|2| 3 |   5   |
         |---------------|                  |---------------|
    2    | | | | | | | | |                  |   4   |   6   |
         |---------------|                  |---------------|
```

FIGURE 4.1 HDTV SUB-BAND CODING

The numbers in the spectral domain represent the six bands. As in DCT, the coefficient in the upper left hand corner (Band 1) represents the DC value of the block. Bands 2, 3, and 5 represent increased definition in the horizontal direction, while Band 4 represents increased definition in the vertical direction, and Band 6 represents a diagonal term. These concepts are made more clear in Figure 4.2, in which a simple block is transformed into coefficients, and then inverse transformed using only a limited number of bands. The inverse transform of the DC coefficient (Band 1) gives a block with uniform pixel values. As more and more bands are added, the detail in the spatial domain increases. Note that before Band 6 is added, the sum of the diagonal pixels in a 2 x 2 block are equal: that is, 9 + 19 = 10 + 18. When Band 6 is added, this is no longer true. Thus, it can be seen that the inclusion of Band 6 results in only subtle changes in the reconstituted image.

Decimation is used to arrive at the proper number of coefficients for each band: that is, Bands 1 and 2 have only one coefficient per block, Band 3 has two, and Bands 4, 5, and 6 have four coeffients each.

For the Band 1 coefficients only, a different process is used. The Band 1 coefficients are differentially coded by transmitting the difference of the coefficient of the current block from the coefficient of the previous block. This insures that the Band 1 coefficients are relatively small values centered about zero, just as in the other bands.

Next the coefficients are quantized to reduce the magnitudes and increase the number of zero values. Different quantizers are used for the different bands, since the essence of sub-band coding is that the higher frequency bands can be quantized more coarsely than the lower bands with less discernable effects. This is equivalent to the visibility matrix in DCT coding, as in JPEG. Three quantizers are used: one for Band 1, one for Bands 2 and 3, and one for Bands 4, 5, and 6. In addition, the quantizers are non-linear, with larger step sizes for larger

```
          PIXELS                                  COEFFICIENTS

|-------------------------|       |-----------------------------------|
|10|20|30|40|50|60|70|80|        |670|-294|-72|-78|-18|-18|-18|-20|
|-------------------------|  ->   |-----------------------------------|
|10|18|26|34|42|50|60|70|        | 2 | 10 | 18| 20| -2| -2| -2| 0 |
|-------------------------|       |-----------------------------------|

                                   BAND 1 (DC)
|-------------------------|       |-----------------------------------|
|                         |       |670|                               |
|           41            |  <-   |---|                               |
|                         |       |                                   |
|-------------------------|       |-----------------------------------|

                          BANDS 1 & 2
|-------------------------|       |-----------------------------------|
|            |            |       |670|-294|                          |
|     23     |     60     |  <-   |--------|                          |
|            |            |       |                                   |
|-------------------------|       |-----------------------------------|

                       BANDS 1, 2, & 3
|-------------------------|       |-----------------------------------|
|     |      |      |      |       |670|-294|-72|-78|                 |
| 14  |  32  |  50  |  70  |  <-   |----------------|                 |
|     |      |      |      |       |                                   |
|-------------------------|       |-----------------------------------|

                     BANDS 1, 2, 3, & 4
|-------------------------|       |-----------------------------------|
| 15  | 35   | 55   | 75   |       |670|-294|-72|-78|                 |
|-------------------------|  <-   |----------------|                 |
| 14  | 30   | 46   | 65   |       | 2 | 10 | 18| 20|                 |
|-------------------------|       |-----------------------------------|

                   BANDS 1, 2, 3, 4, & 5
|-------------------------|       |-----------------------------------|
|10|19|30|39|50|59|70|80|        |670|-294|-72|-78|-18|-18|-18|-20|
|-------------------------|  <-   |-----------------------------------|
| 9|18|25|34|41|50|60|70|        | 2 | 10 | 18| 20|                 |
|-------------------------|       |-----------------------------------|

                          ALL BANDS
|-------------------------|       |-----------------------------------|
|10|20|30|40|50|60|70|80|        |670|-294|-72|-78|-18|-18|-18|-20|
|-------------------------|  <-   |-----------------------------------|
|10|18|26|34|42|50|60|70|        | 2 | 10 | 18| 20| -2| -2| -2| 0 |
|-------------------------|       |-----------------------------------|
```

FIGURE 4.2 EXAMPLE OF RECONSTRUCTION USING LIMITED SUB-BANDS

coefficients, similar to the practice for Max quantizers. In other words, when the coefficient is large, it is not as important to know it as accurately as when the coefficient is small. Finally, the quantization step sizes can be adjusted to effect a trade-off between compression and the quality of the reconstructed image. This is important in controlling the output buffer of the encoder.

The quantized coefficients are shuffled, so that all the Band 1 coefficients appear together (for a pair of horizontal lines), then all the Band 2 quantized coefficients, etc. In this way, long runs of zero value quantized coefficients can be obtained, especially for the higher bands. The output of the run length coder is a series of events, which are either runs or quantized coefficients.

The events are then variable length coded, using a Huffman code table. There is a separate table for each group of bands: that is, for Band 1, for Bands 2 and 3, and for Bands 4, 5, and 6. A common Huffman code is used for both types of events (runs and quantized coefficients) so that the decoder can distinguish between them. For the lower bands the quantized coefficients tend to get the shorter codes, while for the high bands the run lengths tend to get the shortest codes, since there will be many more zeros. This is a one-dimensional code, as opposed to H.261 and MPEG which use two-dimensional codes.

The variable length codes are concatinated together, a band at a time for a pair of scan lines, to form the output signal. Decoding follows the reverse process.

4.2 Simulation Results

The algorithm described above was simulated to compare its performance with other algorithms. Tables from the Bellcore document were used, except that Huffman codes for events (runs and coefficients) were derived from a preliminary pass through the image to be compressed, since only one code was supplied in the referenced document. The image "Aerial" was used for the simulation. The quantization step size was varied to obtain several levels of compression and image quality. The reconstructed image is compared to the original to obtain the RMS error. The results are as follows:

BITS PER PIXEL	RMS ERROR
2.61	2.02
2.04	3.14
1.25	4.82
1.12	5.60
0.88	7.80

The computer code used in this simulation is included as Appendix A. These results should be compared with the best results obtained on the same image using a competitive algorithm. This is Discrete Cosine Transform with Q coder, described in Reference 2 which provides an RMS error of 2.07% for 1.12 bits per pixel. This compares with an error of 2.02% for 2.61 bpp for sub-band coding. In general, sub-band coding takes 2.5 times as many bits as DCT for the same picture quality.

REFERENCES

1. Research on Advanced Television for Broadband ISDN, Bellcore, Special Report SR-ARH-001637, March 19, 1990.

2. Investigation of the Adaptive Discrete Cosine Transform Technique for Group 4 Facsimile, NCS Contract DCA100-87-C-0078, Delta Information Systems, August, 1989.

5.0 CCITT RECOMMENDATION H.261

As described in Section 2.0, there is a general trend toward the adoption of a domestic standard for HDTV transmission based upon all digital technology. It was also explained that, at the present time, there are three proponents of all digital systems as listed below.

PROPONENT TEAM	SYSTEM	SCAN FORMAT	LUMINANCE PIXELS	CHROMA PIXELS
AT&T, ZENITH	SPECTRUM COMPATIBLE	787.5/1:1	720 X 1280	360 X 640
GENERAL INSTRUMENT, MIT	DIGICIPHER	1050/2:1	960 X 1408	480 X 352
SARNOFF, NBC, PHILIPS, THOMSON	ADVANCED COMPATIBLE TV	1050/2:1	960 X 1440	480 X 720

All three proposed systems employ DCT coding (8 x 8 pixels) and motion compensation which is similar to the coding technique employed in CCITT Recommendation H.261. An overview of this Recommendation is provided in this section for the two reasons listed below.

- It provides information on technology which is similar to the three proposed systems.

- It may stimulate the adoption of an HDTV standard which is very similar to H.261. This would clearly be advantageous to the video telephony community. A copy of the H.261 Recommendation is included in Appendix B for reference purposes.

Figure 5.1 is a functional block diagram of the video codec as defined in Recommendation H.261. The heart of the system is the source coder which compresses the incoming video signal by reducing redundancy inherent in the TV signal. The multiplexer combines the compressed data with various side information which indicates alternative modes of operation. A transmission buffer is employed to smooth the varying bit rate from the source encoder to adapt it for the fixed bit rate communication channel. A transmission coder includes functions such as forward error control to prepare the signal for the data link.

One of the most challenging problems to be solved by the codec was the

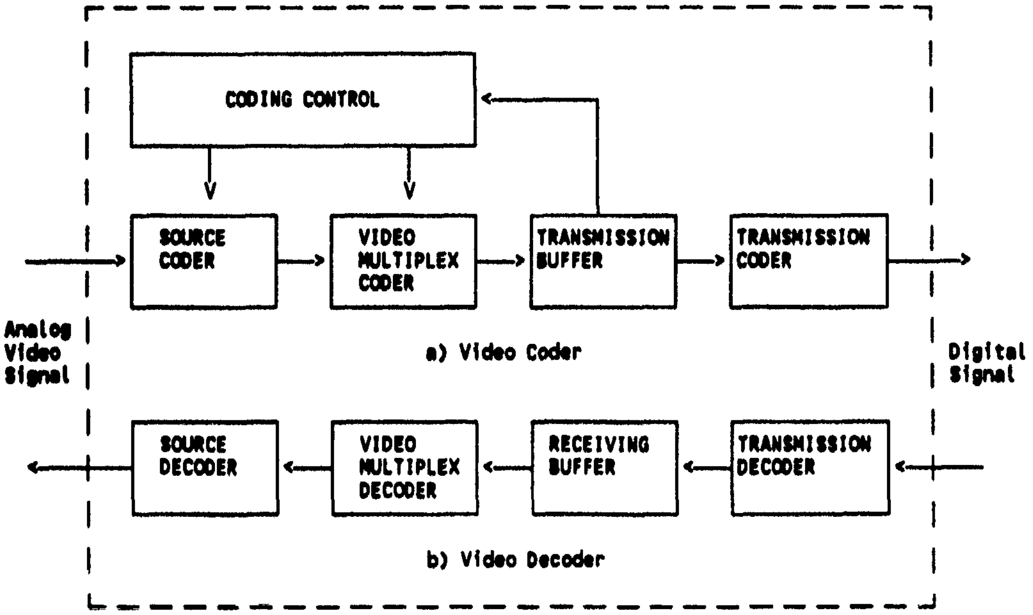


FIGURE 5.1 BLOCK DIAGRAM OF THE VIDEO CODEC

	CIF	QCIF
Coded Pictures per Second	29.97	(or integral submultiples)
Coded Luminance pixels per line	352	176
Coded Luminance lines per picture	288	144
Coded Color pixels per line	176	88
Coded Color lines per picture	144	72

TABLE 5.1 CIF AND QCIF PARAMETERS

reconciliation of the incompatibility between European TV standards (PAL, SECAM) and those in most other areas of the world (NTSC). PAL and SECAM employ 625 lines and a 50 Hz field rate while NTSC has 525 lines and a 60 Hz field rate. This conflict was resolved by adopting a Common Intermediate Format (CIF) and QCIF (Quarter CIF) as the picture structure which must be employed for any transmission adhering to H.261. The CIF and QCIF parameters are defined in Table 5.1.

The QCIF format, which employs half the CIF spatial resolution in both horizontal and vertical directions, is the mandatory H.261 format: full CIF is optional. It is anticipated that QCIF will be used for videophone applications where head-and-shoulders pictures are sent from desk to desk. Conversely, it is assumed that the full CIF format will be used for teleconferencing where several people must be viewed in a conference room.

Figure 5.2 is a functional block diagram outlining the H.261 source coder. Interframe prediction is first carried out in the pixel domain. The prediction errors are encoded by the Discrete Cosine Transform using blocks of 8 pels x 8 pels. The Transform coefficients are next quantized and fed to the multiplexer. Motion compensation is included in the prediction on an optional basis.

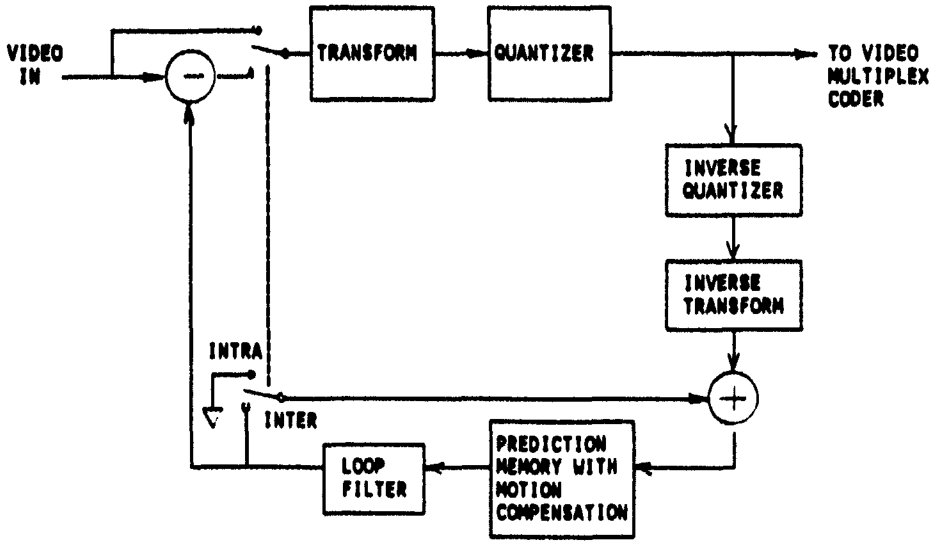


FIGURE 5.2 SOURCE CODER

PICTURE STRUCTURE

In the encoding process, each picture is subdivided into Groups of Blocks (GOB). As shown in Figure 5.3, the CIF picture is divided into 12 GOB's while QCIF has only three GOB's. From the GOB level down, the structure of CIF and QCIF is identical. A header at the beginning of the GOB permits resynchronization and changing the coding accuracy.

Each GOB is further divided into 33 macroblocks, as shown in Figure 5.4. The macroblock header defines the location of the macroblock within the GOB, the type of coding to be performed, possible motion vectors, and which blocks within the macroblock will actually be coded. There are two basic types of coding. In

1	2
3	4
5	6
7	8
9	10
11	12

CIF

1
3
5

QCIF

FIGURE 5.3 ARRANGEMENT OF GOBs IN A PICTURE

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

FIGURE 5.4 ARRANGEMENT OF MACROBLOCKS IN A GOB

Intra coding, coding is performed without reference to previous pictures. This mode is relatively rare, but is required for forced updating, and every macroblock must occasionally be Intra coded to control the accumulation of inverse transform is match error. The more common coding type is Inter, in which only the difference between the previous picture and the current one is coded. Of course, for picture areas without motion, the macroblock does not have to be coded at all.

Each macroblock is further divided into six blocks, as shown in Figure 5.5.

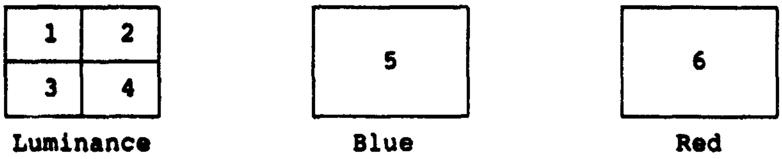


FIGURE 5.5 ARRANGEMENT OF BLOCKS IN A MACROBLOCK

Four of the blocks represent the luminance, or brightness, while the other two represent the red and blue color differences. Each block is 8 x 8 pixels, so it can be seen that the color resolution is half of the luminance resolution in both dimensions.

EXAMPLE OF BLOCK CODING

Figure 5.6 shows a simple example of how each 8 x 8 block is coded. In this case, Intra coding is used, but the principle is the same for Inter coding. Figure 5.6a shows the original block to be coded. Without compression, this would take 8 bits to code each of the 64 pixels, or a total of 512 bits. First, the block is transformed, using the two-dimensional Discrete Cosine Transform (DCT),

75	76	77	78	79	80	81	82
77	78	79	80	81	82	83	84
79	80	81	82	83	84	85	86
81	82	83	84	85	86	87	88
83	84	85	86	87	88	89	90
85	86	87	88	89	90	91	92
87	88	89	90	91	92	93	94
89	90	91	92	93	94	95	96

a) ORIGINAL BLOCK (8x8x8 = 512 BITS)

76	76	77	79	80	81	82	83
77	77	78	80	81	82	83	84
79	79	80	81	83	84	85	86
81	82	83	84	85	87	88	88
84	84	85	87	88	89	90	91
86	87	88	89	91	92	93	93
88	89	90	91	92	94	95	95
89	90	91	92	93	95	96	96

f) RECONSTITUTED BLOCK

684	-19	-1	-2	0	-1	0	-1
-37	0	-1	0	0	0	0	-1
0	0	0	0	0	0	0	0
-4	-1	-1	-1	-1	0	-1	-1
0	0	0	0	0	0	0	0
-2	0	0	-1	0	-1	0	-1
0	0	0	0	-1	-1	-1	-1
-1	-1	-1	0	-1	0	-1	0

b) TRANSFORMED BLOCK COEFFICIENTS

688	-21	0	0	0	0	0	0
-39	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

e) INVERSE QUANTIZED COEFFICIENTS

86	-3	0	0	0	0	0	0
-6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

c) QUANTIZED COEFFICIENT LEVELS

RUN	LEVEL	CODE
0	86	01010110
0	-3	001011
0	-6	001000011
	EOB	10

TOTAL CODE LENGTH = 25

d) COEFFICIENTS IN ZIG-ZAG ORDER AND VARIABLE LENGTH CODED

FIGURE 5.6 SAMPLE INTRA BLOCK CODING

giving the coefficients of Figure 5.6b. Note that most of the energy is concentrated into the upper left-hand corner of the coefficient matrix. Next, the coefficients of Figure 5.6b are quantized with a step size of 6. (The first term {DC} always uses a step size of 8.) This produces the values of Figure 5.6c, which are much smaller in magnitude than the original coefficients and most of the coefficients become zero. The larger the step size, the smaller the values produced, resulting in more compression.

The coefficients are then reordered, using the Zig-Zag scanning order of Figure 5.7. All zero coefficients are replaced with a count of the number of zero's before each non-zero coefficient (RUN). Each combination of RUN and VALUE produces a Variable Length Code (VLC) that is sent to the decoder. The last non-zero VALUE is followed by an End of Block (EOB) code. The total number of bits used to describe the block is 25, a compression of 20:1.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

FIGURE 5.7 SCANNING ORDER IN A BLOCK

At the decoder (and at the coder to produce the prediction picture), the step size and VALUE's are used to reconstruct the inverse quantized coefficients, which, as shown in Figure 5.6e are similar to, but not exactly equal to, the original coefficients. When these coefficients are inverse transformed, the result

of Figure 5.6f is obtained. Note that the differences between this block and the original block are quite small.

MOTION COMPENSATION

The operation of motion compensation is shown in Figure 5.8. Block "A" is a block in the current picture that is to be coded. Block "B" is the block at the same position as "A" but in the picture that was previously stored in both coder and decoder. Because of image motion, block "A" more closely resembles the pixel data from block "C" than that from block "B". The displacement of block "C" from block "B", measured in pixels in x and y directions, is the motion vector. The pixel-by-pixel difference between blocks "A" and "C" is transformed and coded. The motion vector and code data are transmitted to the decoder, where the inverse

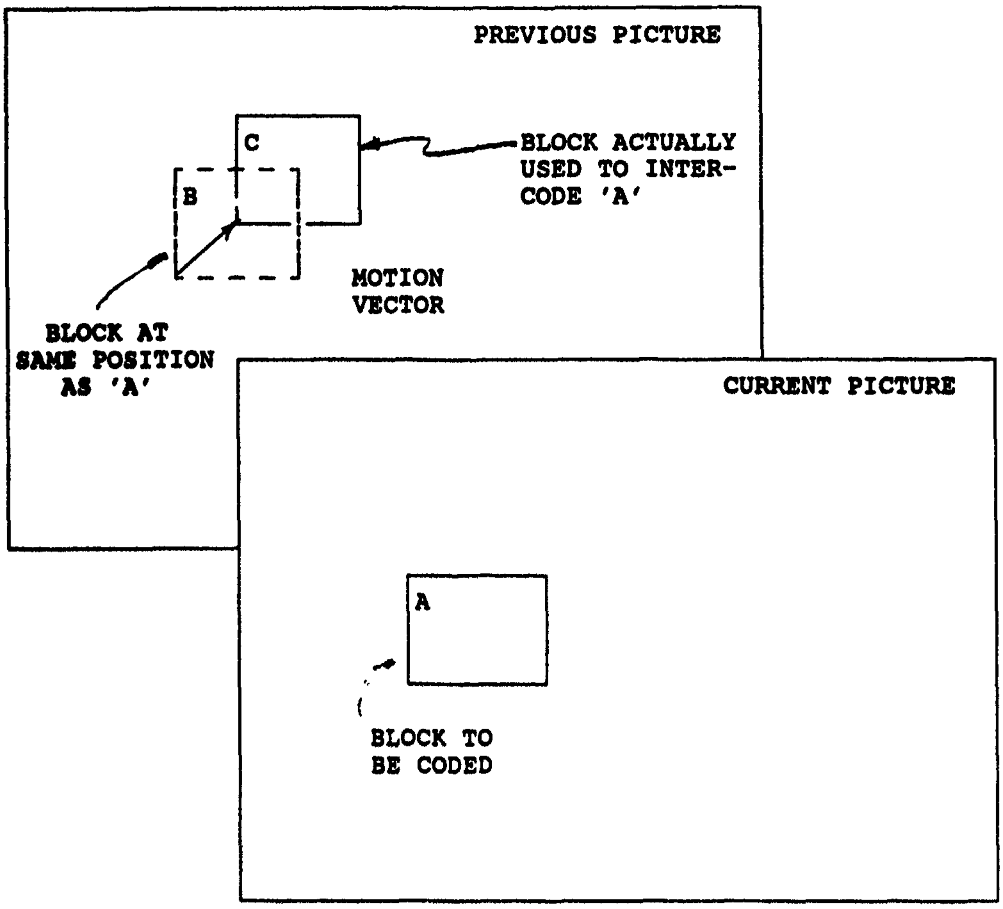

PREVIOUS PICTURE
C
B
BLOCK ACTUALLY USED TO INTER-CODE 'A'
MOTION VECTOR
BLOCK AT SAME POSITION AS 'A'
CURRENT PICTURE
A
BLOCK TO BE CODED

FIGURE 5.8 INTER-FRAME CODING WITH MOTION VECTORS

transformed block data is added to the data in block "C" pointed to by the motion vector, and placed in the block "A" position.

The use of motion vectors is optional in the coder, where the calculation of the optimum motion vectors is complex, but required in the decoder, where the reconstruction of the motion is relatively simple.

H.261 EXTENSION FOR HDTV

It is recognized that HDTV could be very valuable to the government community for both command/control applications and desk-to-desk communications of high resolution color graphics/imagery. The challenge is to provide for the efficient digital communications of this imagery over switched communication networks available today and in the near term. We suggest that it would be very practical to transmit HDTV signals over existing digital networks using the H.261 draft Recommendation as the coding algorithm. With virtually no changes, the H.261 Recommendation could handle HDTV signals. The SMPTE 240M is a representative HDTV signal and has a resolution of 1920 x 1035 pixels. This corresponds to 11 x 22 Groups of Blocks as viewed from an H.261 perspective (vs. 2 x 6 for CIF and 1 x 3 for QCIF). Such an HDTV picture could be completely updated in one second over a T-1 circuit using H.261.

The only limitation in the H.261 format as far as image size is concerned, is the GOB number. H.261 allows for GOB numbers from 1 to 15, although only 1 to 12 are used for CIF. For the 240M format, 11 x 22 = 242 GOBs are needed. However, according to H.261, every GOB header must appear. Therefore, there would be no problem if GOBs were numbered module 16 (excluding zero, which is used for the Picture Start Code). Only a massive error burst, involving thousands of bits, could cause any confusion about which GOB is being coded.

Using the full H.261 algorithm (interframe coding plus motion compensation) should give 0.1 bits per pixel for most images. For the 240M format, this should permit updating at a 7.5 frames/sec. rate, using the full T.1 data rate. The 0.1 bits/pixel coding rate is justified because of the high degree of correlation of adjacent pixels at such a high resolution.

For low bit rates, most of the capacity is used up by overhead, mainly in the form of GOB headers. For a switched 56 Kbps channel carrying only video, there is a net video bit rate of 52,275 bits per second, after taking out FAS, BAS, and FEC bits. Each frame requires 6,324 bits of header data, so the maximum frame rate is $\frac{52,275}{6,324} = 8.27\ \mathit{frames/sec.}$ even if there are no changes. However, if the image is a highly detailed status chart, changes involving only a small fraction of the image would be updated virtually instantaneously. Initializing the status chart, or changing from one to another, would take about 20 seconds at 0.5 bits per pixel for intraframe coding.

6.0 COMMUNICATION CONSIDERATIONS FOR TELECONFERENCING

In general, video teleconferencing requires a high transmission bit rate relative to other services such as voice and data. For that reason the availability of teleconferencing for the government community is dependent upon the availability of ubiquitous, inexpensive, switched, communication channels operating at high bit rates. The purpose of this section is to, in very general terms, examine communication issues as they relate specifically to video teleconferencing. The discussion will be divided into three parts: (1) teleconferencing communications today, (2) narrowband ISDN, and (3) broadband ISDN.

6.1 Teleconferencing Communications Today

Teleconferencing systems which are being installed today fall into two general categories -- narrowband (switched 56 Kbps), and wideband (384 Kbps, 768 Kbps, 1.544 mbps). Typical wideband services are implemented using either dedicated private T1 circuits or a switched service from AT&T (Accunet Reserved) or Sprint (Meeting Channel). In either case, a dedicated T1 type trunk circuit must be brought to the user's premises. In the case of a switched service, the user's access line is connected to the existing network implemented by AT&T (Accunet Reserved) or Sprint (Meeting Channel). As indicated above, the typical transmission bit rates employed over these wideband networks are 384 Kbps, 768 Kbps, or 1.544 mbps depending upon the quality of service required.

In the case of narrowband T/C systems, a typical T/C terminal would require two parallel switched 56 Kbps circuits be brought to the users premises. The terminal typically reallocates the total 112 Kbps capacity by assigning 32 Kbps to audio and 80 Kbps to video for example. The video quality at 80 Kbps is obviously reduced relative to that provided for wideband teleconferencing. Nevertheless, it has been found to be very effective for problem solving sessions and a wide range of teleconference applications.

Both the narrowband and wideband teleconference network approaches described above are directly applicable to the transmission of HDTV signals for teleconferencing applications. For example, HDTV signals can be compressed by the CCITT standard algorithm for transmission at 1.544 mbps and provide very reasonable quality. In addition, it is possible to transmit HDTV signals over existing switched 56 Kbps networks for command and control applications (status board, computer graphics, etc.) where the data does not change rapidly.

The Department of Defense has established the Defense Communications Teleconference Network (DCTN) to provide teleconference services within that agency. This network provides switched service at a transmission bit rate of 1.544 mbps. Considerations are presently underway for reducing this bit rate to 768 or 384 Kbps.

The FTS 2000 provides telecommunication "services" to the U.S. Government. These services, including teleconference services, are now provided by two contractors, AT&T and Sprint. The Government does not specify the network configurations or the hardware of the network, merely the delivered service. At the present time, the FTS 2000 does not specify a teleconference service having HDTV resolution. Nevertheless, this could be done in the future.

6.2 Narrowband ISDN (N-ISDN)

The main feature of the ISDN concept is the support of a wide range of voice and non-voice applications in the same network. A key element of service integration for an ISDN is the provision of a range of services (part II of the I-series of Recommendations) using a limited set of connection types and multipurpose user-network interface arrangements (parts III and IV of the I-series of Recommendations). ISDNs support a variety of applications including both switched and non-switched connections. Switched connections in an ISDN include both circuit-switched and packet-switched connections and their concatenations. A layered protocol structure is used for the specification of the access to an ISDN.

A digital pipe between the central office and the ISDN subscriber is used to carry a number of communication channels. The capacity of the pipe, and therefore the number of channels carried will vary from user to user. The transmission structure of any access link will be constructed from the following types of channels:

- B channel: 64 Kbps
- D channel: 16 or 64 Kbps
- H_0 channel: 384 Kbps
- H_{11} channel: 1.536 Mbps
- H_{12} channel: 1.92 Mbps

The B channel is the basic user channel to carry digital data. The D channel serves two main purposes. First, it carries common-channel signaling information to control circuit-switched calls on associated B channels at the user interface. In addition, the D channel may be used for packet-switching or low-speed telemetry at times when no signaling information is waiting. H channels are provided for user information at higher bit rates. The user may use such a channel as a high-speed trunk or subdivide the channel according to the user's own TDM scheme.

Basic access consists of two full-duplex 64-Kbps B channels and a full-duplex 16-Kbps D channel. The total bit rate, by simple arithmetic, is 144 Kbps. However, framing, synchronization, and other overhead bits bring the total bit rate on a basic access link to 192 Kbps.

Primary access is intended for users with greater capacity requirements, such as offices with a digital PBX or a LAN. Because of differences in the digital transmission hierarchies used in different countries, it was not possible to get agreement on a single data rate. The United States, Canada, and Japan make use of a transmission structure based on 1.544 Mbps; this corresponds to the T1 transmission facility of AT&T. In Europe, 2.048 Mbps is the standard rate. Both of these data rates are provided as a primary interface service. Typically, the channel structure for the 1.544 Mbps rate will be 23 B channels plus one 64 Kbps D channel and, for the 2.048 Mbps rate, 30 B channels plus one 64 Kbps D channel.

As explained earlier, the H.261 Recommendation was established specifically for the N-ISDN. In fact, the term P x 64 is frequently used synonymously with H.261 to represent the transmission bit rates where P is any integer from 1 through 30. Unfortunately, the N-ISDN is not universally available today even in the metropolitan areas. However, by 1992 and 1993 this service should be generally available and fully tariffed, and this time schedule is reasonably consistent with a potential introduction of any HDTV service.

Although the H.261 Recommendation stresses the ISDN, it should be clearly stated that the standard is fundamentally capable of operating at non-ISDN rates and in non-ISDN modes. As explained in Section 6.1, the H.320 audio visual terminal will be operating at the North American rates (56 Kbps, 768 Kbps, 1.544 mbps) for many years before ISDN is fully deployed.

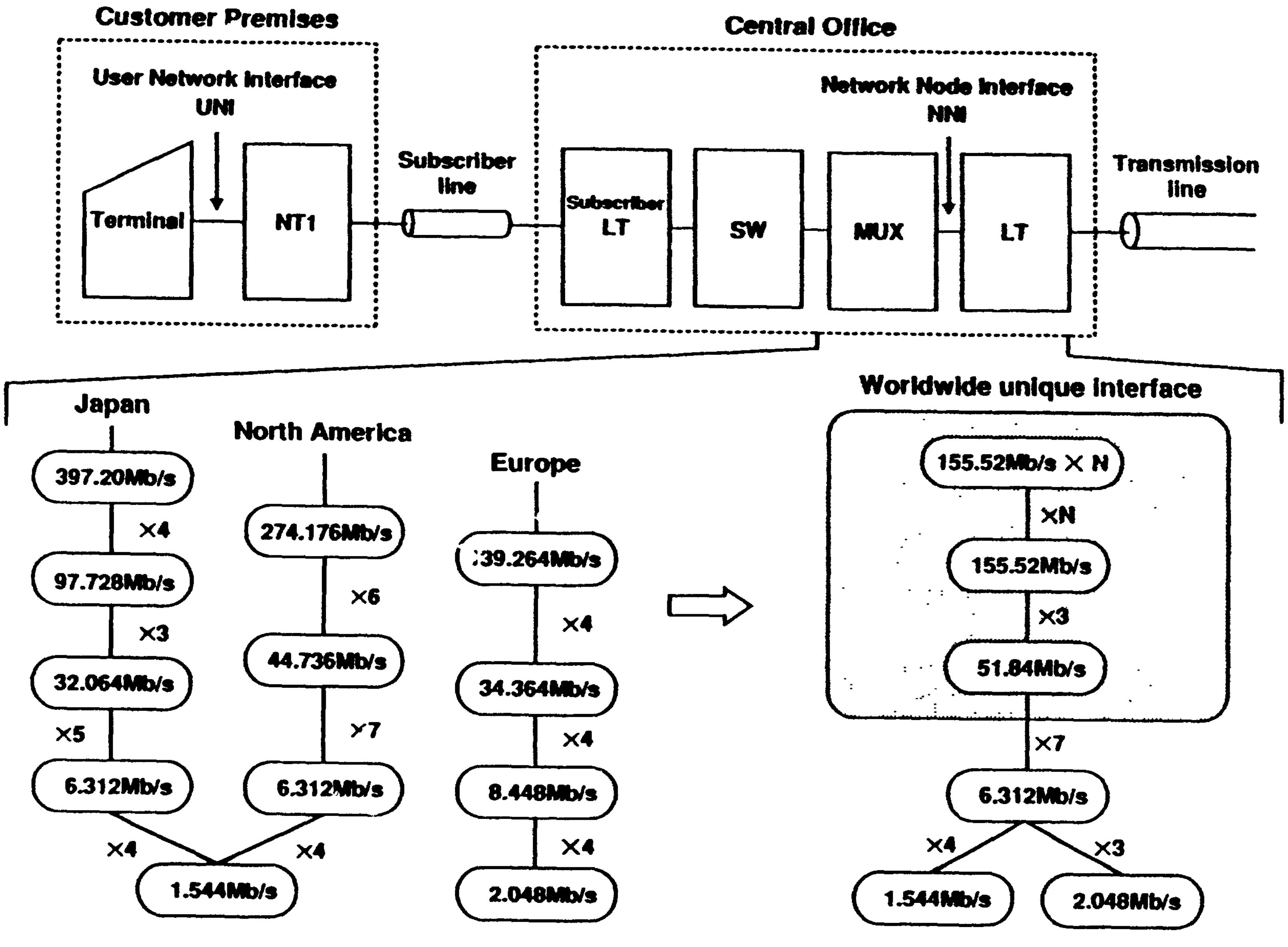
6.3 Broadband ISDN (B-ISDN)

The Broadband ISDN refers to that segment of the communication hierarchy where the transmitted bit rate exceed the primary rate which in the U.S. is 1.544

Mbps. Broadband aspects of the ISDN (B-ISDN) are being studied by CCITT Study Group XVIII for a future customer-switched digital network. SGXVIII decided to standardize the Network Node Interface (NNI) by a worldwide unique Synchronous Digital Hierarchy (SDH). This was achieved by Working Party 7 which is responsible for transmission aspects of digital networks. Figure 6.1 illustrates the world-wide unique NNI. The SDH specifies 155.52 Mb/s as the world-wide unique interface bit rate. The proposal of Study Group XVIII for B-ISDN as described in Recommendation I.121 is that the target transfer mode is the Asyncrhonous Transfer Mode (ATM), in which the data is transmitted in a series of fixed size blocks called cells. Packet-switched networks already exist for the transmission of digital data for non-real time services (for example, the exchange of information between computer databases). In this instance, if a packet is corrupted or lost, the receiving terminal can request that the particular packet be retransmitted. Recommendation I.121, however, envisages that the B-ISDN will carry all the telecommunications services provided in the future including real-time services such as telephony, videoconferencing, and videophony, as well as television and sound contribution and distribution services. For these real-time services, if a cell is corrupted or lost, retransmission of cells is not possible and so degradation of the signal may occur.

The main advantages claimed for ATM is that the network switches are no longer bit-rate and service specific; in the B-ISDN all services (including future new, and as yet unspecified services) are expected to be carried, and a common user-network interface will exist for all services. Many of the important parameters of B-ISDN have still to be specified. However, an ATM-based network will introduce some effects not experienced in synchronous networks, such as cell delay jitter and occasional cell loss.

An ATM-based network will, in principle, provide the user with whatever bit rate is required (within the constraints of the interface and the network), so that teleconference users, for example, could decide on the optimum picture quality required by sessions. Additionally, new television services at different bit rates could be transmitted over the network through the same user-network interface. With continuing improvements in picture coding algorithms, and with advances in technology allowing more complex algorithms to be implemented, service providers could, in the future, offer either an improved quality of service at the same average bit rate, or the same quality of service at a lower average bit rate. An ATM-based



Existing Digital Hierarchy

New Synchronous Digital Hierarchy

FIGURE 6-1

network will in principle have the flexibility to provide additional transmission capacity when required, and could allow the development of a variable bit rate/constant quality coding scheme.

7.0 CONCLUSIONS

The following conclusions are drawn from the work performed on this project.

- There is a general trend by the ATSC toward the adoption of a domestic standard for HDTV transmission based upon all digital technology. At the present time, there are three proponents of all digital systems as listed below.

PROPONENT TEAM	SYSTEM	SCAN FORMAT	LUMINANCE PIXELS	CHROMA PIXELS
AT&T, ZENITH	SPECTRUM COMPATIBLE	787.5/1:1	720 X 1280	360 X 640
GENERAL INSTRUMENT, MIT	DIGICIPHER	1050/2:1	960 X 1408	480 X 352
SARNOFF, NBC, PHILIPS, THOMSON	ADVANCED COMPATIBLE TV	1050/2:1	960 X 1440	480 X 720

All three proposed systems employ DCT coding (8 x 8 pixels) and motion compensation which is similar to the coding technique employed in CCITT Recommendation H.261. This trend toward all-digital transmission is obviously a very favorable development for the community interested in teleconferencing.

A typical terrestrial broadcast system proposed to the ATSC will transmit an HDTV signal at 20 mbps at a low power level, in the taboo channels, simultaneously with the transmission of the NTSC signal over the present conventional channels.

- There is little progress towards an agreement on international standards for HDTV. The Japanese are moving forward with the MUSE system (1125 lines; 60 fields/sec.) while the Europeans are proceeding vigorously with the EUREKA system (1250 lines; 50 fields/sec.). Neither of these are related to any of the proposals before the ATSC.

- Compression techniques were reviewed as they might apply to the coding of the HDTV signal. It is concluded that Sub-band coding and transform coding, which provides the basis for Recommendation H.261, is well suited for coding the HDTV signal. Since the three major ATSC proponents are focusing on interframe DCT with motion compensation, the compression algorithm issue is apparently resolved.

- The communications infrastructure is well positioned to exploit the HDTV technology for teleconferencing and videophone applications. At the present time switched 56 Kbps channels and switched 1.544 mbps channels are available. As the narrowband ISDN becomes more available, the 56 Kbps channels will be replaced with 64 Kbps and a switched 384 Kbps service will become available. In the longer term, the Broadband ISDN will become available having an interface at 152 mbps.

- It is likely that, in the immediate future, the primary channel rate (1.544 mbps) will not be adequate to provide acceptable picture quality for moving teleconference scenes, with HDTV resolution, using the H.261-like algorithm. It is estimated that adequate quality could be provided in the 6-8 mbps region which can easily be provided by satellite today.

- It is concluded that HDTV could play an immediate significant role within the U.S. Government for teleconferencing. Existing HDTV technology (e.g. SMPTE 240M cameras and monitors) can be combined with existing switched data channels (56 Kbps, 1.544 mbps) and the existing coding algorithm (H.261) to provide a valuable service within the U.S. Government today. High resolution status displays could be updated, on a virtual instantaneous basis, for viewing on an analyst's desk or a large screen display for group viewing.

APPENDIX A

SOFTWARE FOR SIMULATION OF SUB-BAND CODING

```
$files 3,3,8
***************************************************************************
**
*
 *
*      program name: sub_band
 *
*
 *
*      description: This program encodes, tabulates statistics, and
 *
*                   decodes an image using a sub-band encoding algorithm.
 *
*                   A histogram of the input to the variable length coder
 *
*                   is also produced for subsequent use in the VLC.
 *
*
 *
***************************************************************************
**

      program sub_band
      implicit none

      integer i,j,low,high

      integer*4 tot1,tot23,tot456

      include filter.inc
      include filterda.inc

*** runstring: ru,sub_band.run,<format file>

*** get parameters from format file

      call fparm(formatfile)

      open (unit=imglu,file=formatfile,iostat=istat)
      if (istat.ne.0) stop 'open format file error'

*** get size of image
      read (unit=imglu,fmt=*,iostat=istat) length,recrds
*** get image source and destination
      read (unit=imglu,fmt=*,iostat=istat) imgfile
      read (unit=imglu,fmt=*,iostat=istat) imgout
*** get histogram destination
      read (unit=imglu,fmt=*,iostat=istat) hist1
      read (unit=imglu,fmt=*,iostat=istat) hist23
      read (unit=imglu,fmt=*,iostat=istat) hist456
*** get coding parameters
      read (unit=imglu,fmt=*,iostat=istat) ab
      read (unit=imglu,fmt=*,iostat=istat) ae
      read (unit=imglu,fmt=*,iostat=istat) scaler1
      read (unit=imglu,fmt=*,iostat=istat) scaler23
      read (unit=imglu,fmt=*,iostat=istat) scaler456
*** get VLC tables
      read (unit=imglu,fmt=*,iostat=istat) vlc1file
      read (unit=imglu,fmt=*,iostat=istat) vlc23file
      read (unit=imglu,fmt=*,iostat=istat) vlc456file
      close(unit=imglu)

      data total/0/                           ! total # of bits encoded
      data tot1,tot23,tot456/0,0,0/
      data mse,sume/0,0/                      ! mean squared error, sum error

*** quantization tables from Bellcore documents (appendix VIc)
      data dpcmtable/0,1,2,3,4,5,6,7,8,9,2*10,2*11,2*12,2*13,
     &               2*14,2*15,2*16,2*17,2*18,2*19,2*20,4*21,
     &               6*22,6*23,6*24,6*25,6*26,6*27,6*28,6*29,6*30,
     &               6*31,6*32,8*33,10*34,10*35,10*36,10*37,10*38,
     &               10*39,10*40,10*41,10*42,10*43,10*44,10*45,
     &               10*46,10*47,10*48,10*49,10*50,10*51,10*52,
     &               10*53,10*54,10*55,10*56,10*57,10*58,10*59,
     &               10*60,10*61,10*62,112*63/

      data lowtable/0,1,2*2,3*3,3*4,3*5,3*6,3*7,3*8,3*9,3*10,3*11,
     &              3*12,3*13,3*14,3*15,3*16,3*17,3*18,3*19,3*20,
     &              3*21,3*22,3*23,3*24,3*25,3*26,3*27,3*28,3*29,
     &              3*30,3*31,3*32,3*33,3*34,3*35,3*36,3*37,3*38,
     &              3*39,3*40,3*41,3*42,3*43,3*44,3*45,3*46,3*47,
     &              3*48,3*49,3*50,3*51,3*52,3*53,3*54,3*55,3*56,
     &              3*57,3*58,3*59,3*60,3*61,3*62,72*63/

      data high3table/3*0,3*1,5*2,6*3,7*4,7*5,7*6,7*7,8*8,9*9,66*10/
***   correct table?

*** calculate inverse DPCM quantization table for DC band (band 1) ***
      j=0
      do i=0,63
        low=j
        do while (dpcmtable(j).eq.i.and.j.lt.512)
          j=j+1
        end do
        high=j-1
        invdpcmtable(i)=(low+high)/2  !reconstruction level in center of ra
nge
      end do

*** calculate inverse PCM quantization table for low-low bands (bands 2&3)
***
      j=0
      do i=0,63
        low=j
        do while (lowtable(j).eq.i.and.j.lt.256)
          j=j+1
        end do
        high=j-1
        invlowtable(i)=(low+high)/2
      end do
```

```
*** calculate inverse PCM quantization table for high bands (bands 4,5,6) *
**
      j=0
      do i=0,31
        low=j
        do while (high3table(j).eq.i.and.j.lt.128)
          j=j+1
        end do
        high=j-1
        invhigh3table(i)=(low+high)/2
      end do
      invhigh3table(0)=0

*** read vlc tables

      open(unit=imglu,file=vlc1file,iostat=istat)
      if (istat.ne.0) stop 'open vlc1 error'
      do i=0,255
        read (unit=imglu,fmt=*,iostat=istat) n1,n2
        if (istat.ne.0) stop 'read vlc1 error'
        vlctable1(n1)=n2
      end do
      close (unit=imglu)

      open(unit=imglu,file=vlc23file,iostat=istat)
      if (istat.ne.0) stop 'open vlc23 error'
      do i=0,255
        read (unit=imglu,fmt=*,iostat=istat) n1,n2
        if (istat.ne.0) stop 'read vlc23 error'
        vlctable23(n1)=n2
      end do
      close (unit=imglu)

      open(unit=imglu,file=vlc456file,iostat=istat)
      if (istat.ne.0) stop 'open vlc456 error'
      do i=0,255
        read (unit=imglu,fmt=*,iostat=istat) n1,n2
        if (istat.ne.0) stop 'read vlc456 error'
        vlctable456(n1)=n2
      end do
      close(unit=imglu)

*** open input file

      inquire (file=imgfile,recl=reclen,maxrec=numrec,
     &         exist=exists,access=acctyp,iostat=istat)
      if (istat.ne.0) stop 'inquire error'
      if (.not.exists) stop 'file does not exist'
      if (acctyp.eq.'SEQUENTIAL') then
        if (reclen.ne.0.or.numrec.ne.0) stop
     &             'neither direct access or mag tape file'
        open(unit=imglu,file=imgfile,access=acctyp)
        call lgbuf(ftn77,bufmax)
        read (unit=imglu,iostat=istat)
     &       (recbuf(n1),n1=1,bufmax)
        if (istat.eq.0) stop 'no error is an error'
        reclen=itlog()
        numrec=recrds
        backspace(unit=imglu,iostat=istat)
        if (istat.ne.0) stop 'backspace error'
      else if (acctyp.eq.'DIRECT') then
        open(unit=imglu,file=imgfile,status='old',access=acctyp,
     &       recl=reclen,maxrec=numrec,use='nonexclusive')
      else
        stop 'access type undefined'
      end if

*** open output file

      open(unit=outlu,file=imgout,access='DIRECT',recl=reclen,
     &     maxrec=numrec+1,iostat=istat,status='UNKNOWN')
      if (istat.ne.0) stop 'image output file open error'

*** set up parameters

      numbpp=8
      numppl=min(length,reclen/numbpp*nbitpb)
      numrec=min(numrec,recrds)
      numwpr=reclen/nbytpw

      do n1=1,numrec/2

*** read in 2 lines of video data

        call lgbuf(ftn77,bufmax)
        read(unit=imglu,iostat=istat) (recbuf(n2),n2=1,numwpr)
        if (istat.ne.0) then
          write (term,*) 'i/o error ',istat,' at read1'
          stop
        end if
        call transfer(recbuf,start1,numppl,numbpp)

        call lgbuf(ftn77,bufmax)
        read(unit=imglu,iostat=istat) (recbuf(n2),n2=1,numwpr)
        if (istat.ne.0) then
          write (term,*) 'i/o error ',istat,' at read2'
          stop
        end if
        call transfer(recbuf,start2,numppl,numbpp)

*** pre filter lines

        call prefilter(start1,pref1,numppl,ab)
        call prefilter(start2,pref2,numppl,ab)

*** filter data into 6 channels

        call filterhor(pref1,tempb1,tempa1,numppl)
        call filterhor(pref2,tempb2,tempa2,numppl)
        call filtervert(tempa1,tempa2,filt5,filt6,numppl/2)
        call filtervert(tempb1,tempb2,tempc,filt4,numppl/2)
        call filterhor(tempc,tempd,filt3,numppl/2)
        call filterhor(tempd,filt1,filt2,numppl/4)

*** round down channels

        call round(filt1,round1,numppl/8,.125/scaler1)
        call round(filt2,round2,numppl/8,.125/scaler23)
        call round(filt3,round3,numppl/4,.25/scaler23)
        call round(filt4,round4,numppl/2,.25/scaler456)  ! round more if
        call round(filt5,round5,numppl/2,.25/scaler456)  ! scaler>1
        call round(filt6,round6,numppl/2,.25/scaler456)  ! lowers bit rate

*** dpcm channel 1 and pcm channels 2-6 using tables

        call dpcm(round1,dpcm1,numppl/8,dpcmtable,invdpcmtable)
        call pcm(round2,pcm2,numppl/8,lowtable)
        call pcm(round3,pcm3,numppl/4,lowtable)
        call pcm(round4,pcm4,numppl/2,high3table)
        call pcm(round5,pcm5,numppl/2,high3table)
        call pcm(round6,pcm6,numppl/2,high3table)

*** count number of bits used with run length and variable length coding
*** add result to total
*** keep histogram of output bytes in count arrays

        call rlc_vlc(dpcm1,numppl/8,tot1,vlctable1,count1,8)
        call rlc_vlc(pcm2,numppl/8,tot23,vlctable23,count23,32)
        call rlc_vlc(pcm3,numppl/4,tot23,vlctable23,count23,32)
        call rlc_vlc(pcm4,numppl/2,tot456,vlctable456,count456,128)
        call rlc_vlc(pcm5,numppl/2,tot456,vlctable456,count456,128)
        call rlc_vlc(pcm6,numppl/2,tot456,vlctable456,count456,128)

*** inverse dpcm and inverse pcm

        call invdpcm(dpcm1,round1,numppl/8,invdpcmtable)
        call pcm(pcm2,round2,numppl/8,invlowtable)
        call pcm(pcm3,round3,numppl/4,invlowtable)
        call pcm(pcm4,round4,numppl/2,invhigh3table)
        call pcm(pcm5,round5,numppl/2,invhigh3table)
        call pcm(pcm6,round6,numppl/2,invhigh3table)

*** shift bits left (anti-rounding)

        call round(round1,filt1,numppl/8,8.0*scaler1)
        call round(round2,filt2,numppl/8,8.0*scaler23)
        call round(round3,filt3,numppl/4,4.0*scaler23)
        call round(round4,filt4,numppl/2,4.0*scaler456)
        call round(round5,filt5,numppl/2,4.0*scaler456)
        call round(round6,filt6,numppl/2,4.0*scaler456)

*** inverse filter 6 channels into output

        call invfilthor(filt1,filt2,tempd,numppl/8)
        call invfilthor(tempd,filt3,tempc,numppl/4)
        call invfiltvert(tempc,filt4,tempb1,tempb2,numppl/2)
        call invfiltvert(filt5,filt6,tempa1,tempa2,numppl/2)
        call invfilthor(tempb1,tempa1,pref1,numppl/2)
        call invfilthor(tempb2,tempa2,pref2,numppl/2)

        call prefilter(pref1,post1,numppl,ae)
        call prefilter(pref2,post2,numppl,ae)

*** add up mean squared error

        do i=1,numppl
          j=post1(i)-start1(i)
          mse=mse+j*j
          sume=sume+j
          j=post2(i)-start2(i)
          mse=mse+j*j
          sume=sume+j
        end do

*** write output image (2 lines at a time)

        call invtransfer(post1,outbuf,numppl,numbpp)
        call lgbuf(ftn77,bufmax)
        write (unit=outlu,iostat=istat) (outbuf(n2),n2=1,numwpr)

        call invtransfer(post2,outbuf,numppl,numbpp)
        call lgbuf(ftn77,bufmax)
        write (unit=outlu,iostat=istat) (outbuf(n2),n2=1,numwpr)

        write (term,*) 'lines done:',n1*2,'  total bits:',tot1,tot23,
     &                 tot456,'  mse:',mse
      end do

      close (unit=imglu,iostat=istat)
      if (istat.ne.0) stop 'closing input file error'
      close (unit=outlu,iostat=istat)
      if (istat.ne.0) stop 'closing output file error'

      total=tot1+tot23+tot456
      write (term,*) 'picture used ',total,' bits to transmit'

*** open, write, and close histogram file

      open (unit=outlu,file=hist1,iostat=istat)
      if (istat.ne.0) stop 'could not open histogram1 file'
      write (outlu,*) 'band 1 histogram for ',imgfile
      write (outlu,*) '8 bit entries'
      write (outlu,*) '  '
      write (outlu,*) 'num    count'
      do n1=0,255
  100   format(i7,i9,i9,i9)
        write (unit=outlu,fmt=100,iostat=istat)
     &        n1,count1(n1)
        if (istat.ne.0) stop 'writing histogram error'
      end do
      close (unit=outlu,iostat=istat)
      if (istat.ne.0) stop 'closing histogram error'

      open (unit=outlu,file=hist23,iostat=istat)
      if (istat.ne.0) stop 'could not open histogram1 file'
      write (outlu,*) 'band 2&3 histogram for ',imgfile
      write (outlu,*) '8 bit entries'
      write (outlu,*) '  '
      write (outlu,*) 'num   count'
      do n1=0,255
        write (unit=outlu,fmt=100,iostat=istat)
     &        n1,count23(n1)
        if (istat.ne.0) stop 'writing histogram error'
      end do
      close (unit=outlu,iostat=istat)
      if (istat.ne.0) stop 'closing histogram error'

      open (unit=outlu,file=hist456,iostat=istat)
      if (istat.ne.0) stop 'could not open histogram1 file'
      write (outlu,*) 'band 4,5,&6 histogram for ',imgfile
      write (outlu,*) '8 bit entries'
      write (outlu,*) '  '
      write (outlu,*) 'num   count'
      do n1=0,255
        write (unit=outlu,fmt=100,iostat=istat)
     &        n1,count456(n1)
        if (istat.ne.0) stop 'writing histogram error'
      end do
      close (unit=outlu,iostat=istat)
      if (istat.ne.0) stop 'closing histogram error'
      stop 'done.'
      end
```

```
*****************************************************************
* transfer breaks line into one-word (16 bit) separated pixels *
*****************************************************************
      subroutine transfer(source,dest,n,size)
        ema dest
        integer source,dest,n,size
        dimension dest(n)
        integer i

        do i=1,n
          dest(i)=i4b(source,i*size-size+1,size)
        end do
      end

      subroutine invtransfer(source,dest,n,size)
        ema source
        integer source,dest,n,size
        dimension source(n)
        integer i,j

        do i=1,n
          j=source(i)
          if (j.gt.255) j=255
          if (j.lt.0) j=0
          call mi2b(j,dest,i*size-size+1,size)
        end do
      end

****************************************************
*  prefilter implements an a,32-2a,a (/32) filter *
*  on each horizontal line.  The paramater a       *
*  controlls the compression rate by filtering     *
*  out the high frequency components               *
****************************************************
      subroutine prefilter(source,dest,n,a)
        ema source,dest
        integer source,dest,n,a
        dimension source(n),dest(n)
        integer curr,prev,next,sum

        do curr=1,n
          prev=curr-1
          if (prev.lt.1) prev=1
          next=curr+1
          if (next.gt.n) next=n
          sum=a*source(prev)+a*source(next)+(32-2*a)*source(curr)
          sum=sum/16
          if (sum.ne.(sum/2)*2) sum=sum+1          ! round up
          dest(curr)=sum/2
        end do
      end

************************************************************
*    filterhor filters and 2:1 decimates a row of         *
*    pixels horizontally.                                 *
*   source: the location of row of pixels                      *
*   dest:   place to put new row (1/2 length of source)        *
*   n:      length in pixels of source                         *
****************************************************************
      subroutine filterhor(source,destlow,desthi,n)
        ema source,destlow,desthi
        integer source,destlow,desthi,n
        dimension source(n),destlow(n/2),desthi(n/2)
        integer i

        do i=1,n/2
          destlow(i)=source(2*i-1)+source(2*i)
          desthi(i)=source(2*i-1)-source(2*i)
        end do
      end

*****************************************************************
*  same as above, but filters 2 lines of data vertically       *
*****************************************************************
      subroutine filtervert(source1,source2,destlow,desthi,n)
        ema source1,source2,destlow,desthi
        integer source1,source2,destlow,desthi,n
        dimension source1(n),source2(n),destlow(n),desthi(n)
        integer i

        do i=1,n
          destlow(i)=source1(i)+source2(i)
          desthi(i)=source1(i)-source2(i)
        end do
      end

******************************************************************
*  applies a dpcm to source and stores it in dest.               *
*  quantization levels are stored in table and invtable          *
******************* ************************************
      subroutine dpcm(source,dest,n,table,invtable)
        ema source,dest
        integer source,dest,n,table,invtable
        dimension source(n),dest(n),table(0:*),invtable(0:*)
        integer i,j,diff,quantdiff,approx

        approx=255                                      ! good number?
        do i=1,n
          diff=source(i)-approx
          if (diff.ge.0) then
            quantdiff=table(diff)
          else
            quantdiff=-table(-diff)
          end if
          dest(i)=quantdiff
          if (quantdiff.ge.0) then
            approx=approx+invtable(quantdiff)
          else
            approx=approx-invtable(-quantdiff)
          end if
        end do
      end

      subroutine invdpcm(source,dest,n,invtable)
        ema source,dest
        integer source,dest,n,invtable
        dimension source(n),dest(n),invtable(0:*)
        integer i,last

        last=255
        do i=1,n
          if (source(i).ge.0) then
            last=last+invtable(source(i))
          else
            last=last-invtable(-source(i))
          end if
          dest(i)=last
        end do
      end

*****************************************
*  rounds data by factor of shift       *
*****************************************
      subroutine round(source,dest,n,shift)
        ema source,dest
        integer source,dest,n
        real shift
        dimension source(n),dest(n)
        integer i
        real val

        do i=1,n
          val=abs(source(i)*shift)
          if (source(i).ge.0) then
            dest(i)=nint(val)
          else
            dest(i)=-nint(val)
          end if
        end do
      end

*****************************************************
*  performs pcm on source using data in the table  *
*  note that this subroutine also performs          *
*  inverse pcm given an inverse table               *
*****************************************************
      subroutine pcm(source,dest,n,table)
        ema source,dest
        integer source,dest,n,table
        dimension source(n),dest(n),table(0:*)
        integer i,j

        do i=1,n
          if (source(i).ge.0) then
            dest(i)=table(source(i))
          else
            dest(i)=-table(-source(i))
          end if
        end do
      end

************************************************************
*  counts the number of bits used to code the source.      *
*  uses both run length coding and variable length coding *
************************************************************
      subroutine rlc_vlc(source,n,total,table,countarray,maxrunlen)
        ema source
        integer source,n,table,maxrunlen
        integer*4 total,countarray
        dimension source(n),table(0:255),countarray(0:255)
        integer i,j,zerocount

        zerocount=0
        do i=1,n
          j=source(i)
          j=ibits(j,0,7)

          if (j.eq.0) then
            zerocount=zerocount+1
            if (i.eq.n.or.source(i+1).ne.0.or.
     &          zerocount.eq.maxrunlen) then
              j=zerocount+127           ! run length code=127+# of zeros
              zerocount=0
            else
              goto 100                  ! do not output anything- still in run
            end if
          end if

          total=total+table(j)          ! VLC coding length

C         make a histogram of j
          countarray(j)=countarray(j)+1

  100   end do
      end

********************************************
*  inverse filters two lines horizontally  *
*  and puts result in dest                 *
********************************************
      subroutine invfilthor(sourcelow,sourcehi,dest,n)
        ema sourcelow,sourcehi,dest
        integer sourcelow,sourcehi,dest,n
        dimension sourcelow(n),sourcehi(n),dest(2*n)
        integer i

        do i=1,n
          dest(2*i-1)=(sourcelow(i)+sourcehi(i))/2
          dest(2*i)=(sourcelow(i)-sourcehi(i))/2
        end do
      end

************************************
*  same as above but vertically    *
************************************
      subroutine invfiltvert(sourcelow,sourcehi,dest1,dest2,n)
        ema sourcelow,sourcehi,dest1,dest2
        integer sourcelow,sourcehi,dest1,dest2,n
        dimension sourcelow(n),sourcehi(n),dest1(n),dest2(n)
        integer i

        do i=1,n
          dest1(i)=(sourcelow(i)+sourcehi(i))/2
          dest2(i)=(sourcelow(i)-sourcehi(i))/2
        end do
      end
      integer n1,n2                     ! counters
      integer nbitpw,nbytpw,nbitpb      ! constants
      integer term,imglu,outlu          ! logical units
      integer bufmax                    ! maximum buffer capacity
      integer bufsiz                    ! buffer size

      parameter (bufmax=530,nbitpw=16,nbytpw=2,nbitpb=8,bufsiz=bufmax)

      integer numbpp                    ! bits per pixel
      integer numwpr                    ! words per record
      integer numppl                    ! pixels per line

      integer i4b,mi2b

      character*30 formatfile           ! file names

      character*30 imgfile
      character*30 imgout
      character*30 hist1,hist23,hist456
      character*30 vlc1file,vlc23file,vlc456file
      character*12 acctyp
      logical      exists
      integer      istat
      integer*4    reclen
      integer*4    numrec
      integer      length
      integer      recrds
      integer      itlog
      integer      ftn77(bufmax)
      integer      recbuf(bufsiz)         ! input buffer
      integer      outbuf(bufsiz)         ! output buffer
      ema start1,start2                 ! in extended memory area: tight on
mem
      integer start1(bufmax)            ! two-line buffer
      integer start2(bufmax)
      ema pref1,pref2
      integer pref1(bufmax)             ! buffers for prefiltered data
      integer pref2(bufmax)
      ema post1,post2
      integer post1(bufmax)
      integer post2(bufmax)
      ema tempa1,tempa2,tempb1,tempb2,tempc,tempd
      integer tempa1(bufmax/2)
      integer tempa2(bufmax/2)
      integer tempb1(bufmax/2)          ! buffers used during filtering
      integer tempb2(bufmax/2)
      integer tempc(bufmax/2)
      integer tempd(bufmax/4)
      ema filt1,filt2,filt3,filt4,filt5,filt6
      integer filt1(bufmax/8)
      integer filt2(bufmax/8)
      integer filt3(bufmax/4)           ! buffers for filtered data
      integer filt4(bufmax/2)
      integer filt5(bufmax/2)
      integer filt6(bufmax/2)
      ema round1,round2,round3,round4,round5,round6
      integer round1(bufmax/8)
      integer round2(bufmax/8)
      integer round3(bufmax/4)          ! buffers for rounded data
      integer round4(bufmax/2)
      integer round5(bufmax/2)
      integer round6(bufmax/2)
      ema dpcm1,pcm2,pcm3,pcm4,pcm5,pcm6
      integer dpcm1(bufmax/8)
      integer pcm2(bufmax/8)
      integer pcm3(bufmax/4)            ! buffers for dpcm/pcm'd data
      integer pcm4(bufmax/2)
      integer pcm5(bufmax/2)
      integer pcm6(bufmax/2)

      integer dpcmtable(0:511)          ! look up tables for dpcm channel 1
      integer invdpcmtable(0:63)
      integer lowtable(0:255)           ! look up tables for pcm channels 2,
3
      integer invlowtable(0:63)
      integer high3table(0:127)         ! look up tables for pcm channels 45
6
      integer invhigh3table(0:31)

      integer vlctable1(0:255)          ! length of vlc codes for channel 1
      integer vlctable23(0:255)         ! length of vlc codes for channels 2
,3
      integer vlctable456(0:255)        ! length of vlc codes for channels 4
56

      integer*4 count1(0:255)           ! histogram for channel 1
```

```
        integer*4 count23(0:255)          ! histogram for channels 2,3
        integer*4 count456(0:255)         ! histogram for channels 4,5,6

        integer*4 total                   ! total number of bits to send image

        integer ab,ae                     ! beginning and end filter parameter
s
        real scaler1,scaler23,scaler456   ! quantizer scale factor

        integer*4 mse,sume                ! mean squared error, sum of error

        data    term,imglu,outlu/1,3,5/
```

APPENDIX B

RECOMMENDATION H.261

4. Recommendation H.261

VIDEO CODEC FOR AUDIOVISUAL SERVICES AT p x 64 kbit/s

CONTENTS

The CCITT,

<u>considering</u>

(a) that there is significant customer demand for videophone, videoconference and other audiovisual services;

(b) that circuits to meet this demand can be provided by digital transmission using the B, HO rates or their multiples up to the primary rate or H11/H12 rates;

(c) that ISDNs are likely to be available in some countries that provide a switched transmission service at the B, HO or H11/H12 rate;

(d) that the existence of different digital hierarchies and different television standards in different parts of the world complicates the problems of specifying coding and transmission standards for international connections;

(e) that a number of audiovisual services are likely to appear using basic and primary rate ISDN accesses and that some means of intercommunication among these terminals should be possible;

(f) that the video codec provides an essential element of the infrastructure for audiovisual services which allows such intercommunication in the framework of Recommendation H.200;

(g) that Recommendation H.120 for videoconferencing using primary digital group transmission was the first in an evolving series of Recommendations.

<u>appreciating</u>

that advances have been made in research and development of video coding and bit rate reduction techniques which lead to the use of lower bit rates down to 64 kbit/s so that this may be considered as the second in the evolving series of Recommendations,

<u>and noting</u>

that it is the basic objective of the CCITT to recommend unique solutions for international connections,

<u>recommends</u>

that in addition to those codecs complying to Recommendation H.120, codecs having signal processing and transmission coding characteristics described below should be used for international audiovisual services.

<u>Note 1</u> - Codecs of this type are also suitable for some television services where full broadcast quality is not required.

<u>Note 2</u> - Equipment for transcoding from and to codecs according to Recommendation H.120 is under study.

1. Scope

This Recommendation describes the video coding and decoding methods for the moving picture component of audiovisual services at the rates of p x 64 kbit/s, where p is in the range 1 to 30.

2. Brief specification

An outline block diagram of the codec is given in Figure 1/H.261.

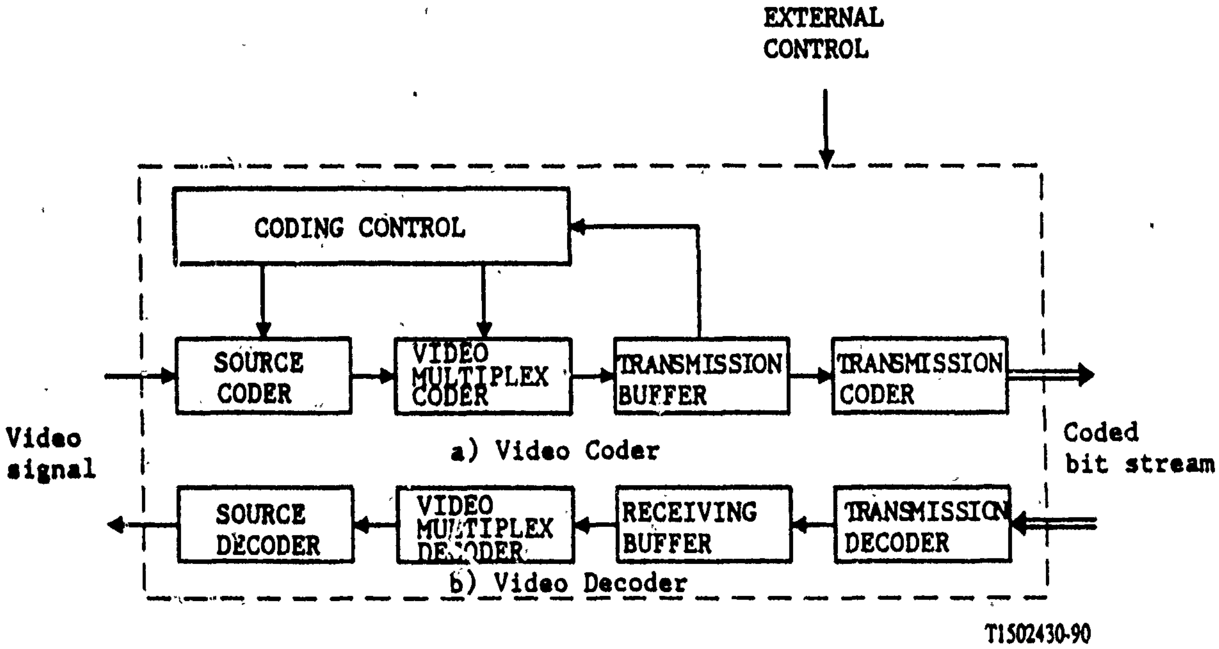


FIGURE 1/H.261

Outline block diagram of the video codec

2.1 Video input and output

To permit a single Recommendation to cover use in and between regions using 625- and 525-line television standards, the source coder operates on pictures based on a common intermediate format (CIF). The standards of the input and output television signals, which may, for example, be composite or component, analogue or digital and the methods of performing any necessary conversion to and from the source coding format are not subject to recommendation..

2.2 Digital output and input

The video coder provides a self-contained digital bit stream which may be combined with other multi-facility signals (for example as defined in Recommendation H.221). The video decoder performs the reverse process.

2.3 Sampling frequency

Pictures are sampled at an integer multiple of the video line rate. This sampling clock and the digital network clock are asynchronous.

2.4 Source coding algorithm

A hybrid of inter-picture prediction to utilize temporal redundancy and transform coding of the remaining signal to reduce spatial redundancy is adopted. The decoder has motion compensation capability, allowing optional incorporation of this technique in the coder.

2.5 Bit rate

This Recommendation is primarily intended for use at video bit rates between approximately 40 kbit/s and 2 Mbit/s.

2.6 Symmetry of transmission

The codec may be used for bidirectional or unidirectional visual communication.

2.7 Error handling

The transmitted bit-stream contains a BCH (511,493) Forward Error Correction Code. Use of this by the decoder is optional.

2.8 Multipoint operation

Features necessary to support switched multipoint operation are included.

3. Source coder

3.1 Source format

The source coder operates on non-interlaced pictures occurring 30000/1001 (approximately 29.97) times per second. The tolerance on picture frequency is ±50 ppm.

Pictures are coded as luminance and two colour difference components (Y, C_B and C_R). These components and the codes representing their sampled values are as defined in CCIR Recommendation 601.

Black - 16
White - 235
Zero colour difference - 128
Peak colour difference - 16 and 240

These values are nominal ones and the coding algorithm functions with input values of 1 through to 254.

Two picture scanning formats are specified.

In the first format (CIF), the luminance sampling structure is 352 pels per line, 288 lines per picture in an orthogonal arrangement. Sampling of each of the two colour difference components is at 144 lines, 176 pels per line, orthogonal. Colour difference samples are sited such that their clock boundaries coincide with luminance block boundaries as shown in Figure 2/H.261. The picture area covered by these numbers of pels and lines has an aspect ratio of 4:3 and corresponds to the active portion of the local standard video input.

Note - The number of pels per line is compatible with sampling the active portions of the luminance and colour difference signals from 525- or 625-line sources at 6.75 and 3.375 MHz respectively. These frequencies have a simple relationship to those in CCIR Recommendation 601.

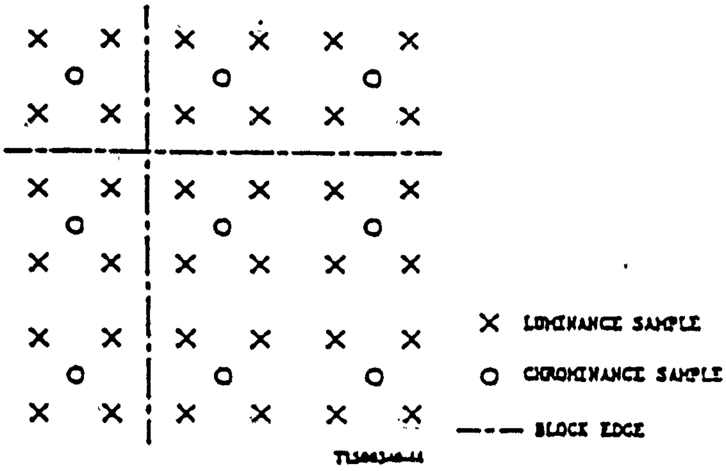


FIGURE 2/H.261

Positioning of luminance and chrominance samples

The second format, Quarter-CIF (QCIF), has half the number of pels and half the number of lines stated above. All codecs must be able to operate using QCIF. Some codecs can also operate with CIF.

Means shall be provided to restrict the maximum picture rate of encoders by having at least 0, 1, 2 or 3 non-transmitted pictures between transmitted ones. Selection of this minimum number and CIF or QCIF shall be by external means (for example via Recommendation H.221).

3.2 Video source coding algorithm

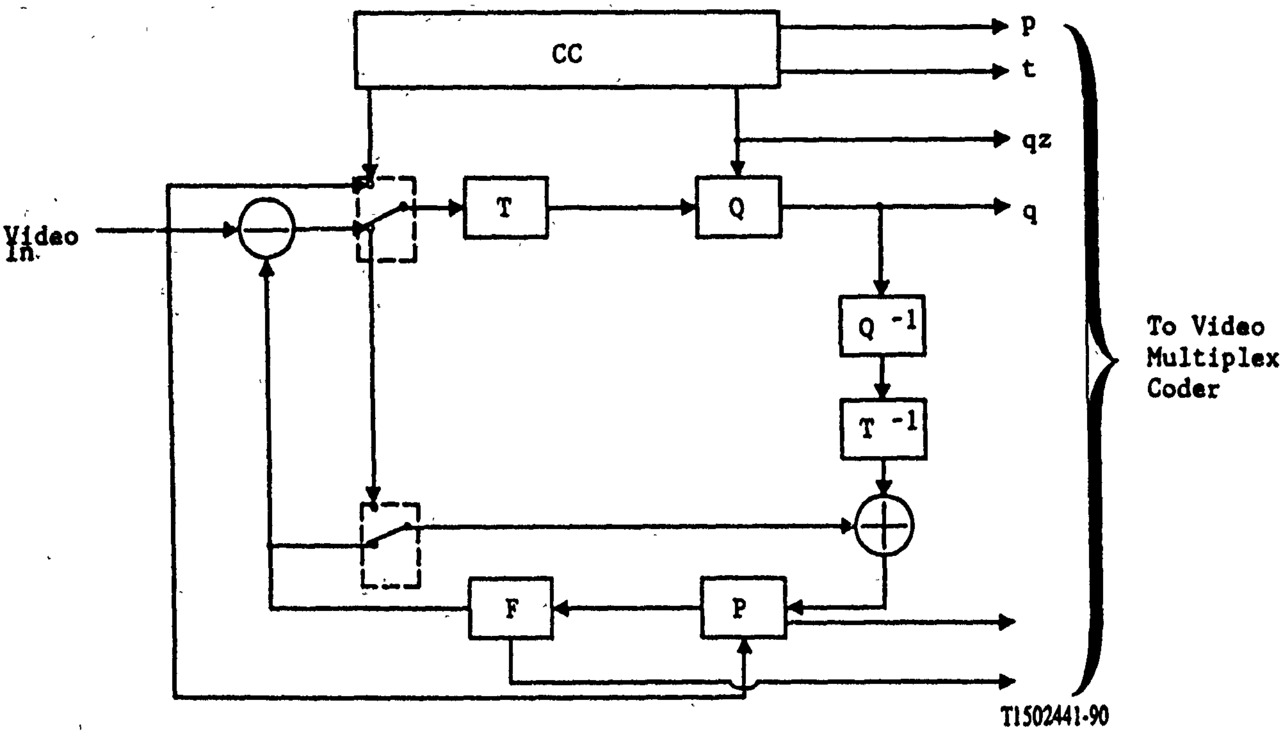
The source coder is shown in generalized form in Figure 3/H.261. The main elements are prediction, block transformation and quantization.

The prediction error (INTER mode) or the input picture (INTRA mode) is subdivided into 8 pel by 8 line blocks which are segmented as transmitted or non-transmitted. Further, four luminance blocks and the two spatially corresponding colour difference blocks are combined to form a macroblock as shown in Figure 10/H.261 of § 4.2.4.

The criteria for choice of mode and transmitting a block are not subject to recommendation and may be varied dynamically as part of the coding control strategy. Transmitted blocks are transformed and resulting coefficients are quantized and variable length coded.

3.2.1 Prediction

The prediction is inter-picture and may be augmented by motion compensation (§ 3.2.2) and a spatial filter (§ 3.2.3)



T: Transform
Q: Quantizer
P: Picture Memory with motion compensated variable delay
F: Loop filter
CC: Coding control

p: Flag for INTRA/INTER
t: Flag for transmitted or not
qz: Quantizer indication
q: Quantizing index for transform coefficients
v: Motion vector
f: Switching on/off of the loop filter

FIGURE 3/H.261

Source coder

3.2.2 Motion compensation

Motion compensation (MC) is optional in the encoder. The decoder will accept one vector per macroblock. Both horizontal and vertical components of these motion vectors have integer values not exceeding ±15. The vector is used for all four luminance blocks in the macroblock. The motion vector for both colour difference blocks is derived by halving the component values of the macroblock vector and truncating the magnitude parts towards zero to yield integer components.

A positive value of the horizontal or vertical component of the motion vector signifies that the prediction is formed from pels in the previous picture which are spatially to the right or below the pels being predicted.

Motion vectors are restricted such that all pels referenced by them are within the coded picture area.

3.2.3 Loop filter

The prediction process may be modified by a two-dimensional spatial filter (FIL) which operates on pels within a predicted 8 by 8 block.

The filter is separable into one-dimensional horizontal and vertical functions. Both are non-recursive with coefficients of 1/4, 1/2, 1/4 except at block edges where one of the taps would fall outside the block. In such cases the 1-D filter is changed to have coefficients of 0, 1, 0. Full arithmetic precision is retained with rounding to 8 bit integer values at the 2-D filter output. Values whose fractional part is one half are rounded up.

The filter is switched on/off for all six blocks in a macroblock according to the macroblock type (see § 4.2.3 MTYPE).

3.2.4 Transformer

Transmitted blocks are first processed by a separable two-dimensional Discrete Cosine Transform of size 8 by 8. The output from the inverse transform ranges from -256 to +255 after clipping to be represented with 9 bits. The transfer function of the inverse transform is given by:

$$f(x,y) = 1/4 \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)\ C(v)\ F(u,v)\ \cos[P(2x + 1)u/16]\ \cos[P(2y + 1)v/16]$$

with u, v, x, y = 0, 1, 2, ...7

where x,y = spatial coordinates in the pel domain

u,v = coordinates in the transform domain

$C(u) = 1/\sqrt{2}$ for u = 0, otherwise 1
$C(v) = 1/\sqrt{2}$ for v = 0, otherwise 1

Note - Within the block being transformed, x = 0 and y = 0 refer to the pel nearest the left and top edges of the picture respectively.

The arithmetic procedures for computing the transforms are not defined, but the inverse one should meet the error tolerance specified in Annex 1.

3.2.5 Quantization

The number of quantizers is 1 for the INTRA dc coefficient and 31 for all other coefficients. Within a macroblock the same quantizer is used for all coefficients except the INTRA dc one. The decision levels are not defined. The INTRA dc coefficient is nominally the transform value linearly quantized with a stepsize of 8 and no dead-zone. Each of the other 31 quantizers is also nominally linear but with a central dead-zone around zero and with a step size of an even value in the range 2 to 62.

The reconstruction levels are as defined in § 4.2.4.

Note - For the smaller quantization step sizes, the full dynamic range of the transform coefficients cannot be represented.

3.2.6 Clipping of reconstructed picture

To prevent quantization distortion of transform coefficient amplitudes causing arithmetic overflow in the encoder and decoder loops, clipping functions are inserted. The clipping function is applied to the reconstructed picture which is formed by summing the prediction and the prediction error as modified by the coding process. This clipper operates on resulting pel values less than 0 or greater than 255, changing them to 0 and 255 respectively.

3.3 Coding control

Several parameters may be varied to control the rate of generation of coded video data. These include processing prior to the source coder, the quantizer, block significance criterion and temporal subsampling. The proportions of such measures in the overall control strategy are not subject to recommendation.

When invoked, temporal subsampling is performed by discarding complete pictures.

3.4 Forced updating

This function is achieved by forcing the use of the INTRA mode of the coding algorithm. The update pattern is not defined. For control of accumulation of inverse transform mismatch error a macroblock should be forcibly updated at least once per every 132 times it is transmitted.

4. Video multiplex coder

4.1 Data structure

Unless specified otherwise the most significant bit is transmitted first. This is bit 1 and is the leftmost bit in the code tables in this document. Unless specified otherwise all unused or spare bits are set to "1". Spare bits must not be used until their functions are specified by the CCITT.

4.2 Video multiplex arrangement

The video multiplex is arranged in a hierarchical structure with four layers. From top to bottom the layers are:

Picture
Group of blocks (GOB)
Macroblock (MB)
Block

A syntax diagram of the video multiplex coder is shown in Figure 4/H.261. Abbreviations are defined in later sections.

4.2.1 Picture layer

Data for each picture consists of a picture header followed by data for GOBs. The structure is shown in Figure 5/H.261. Picture headers for dropped pictures are not transmitted.

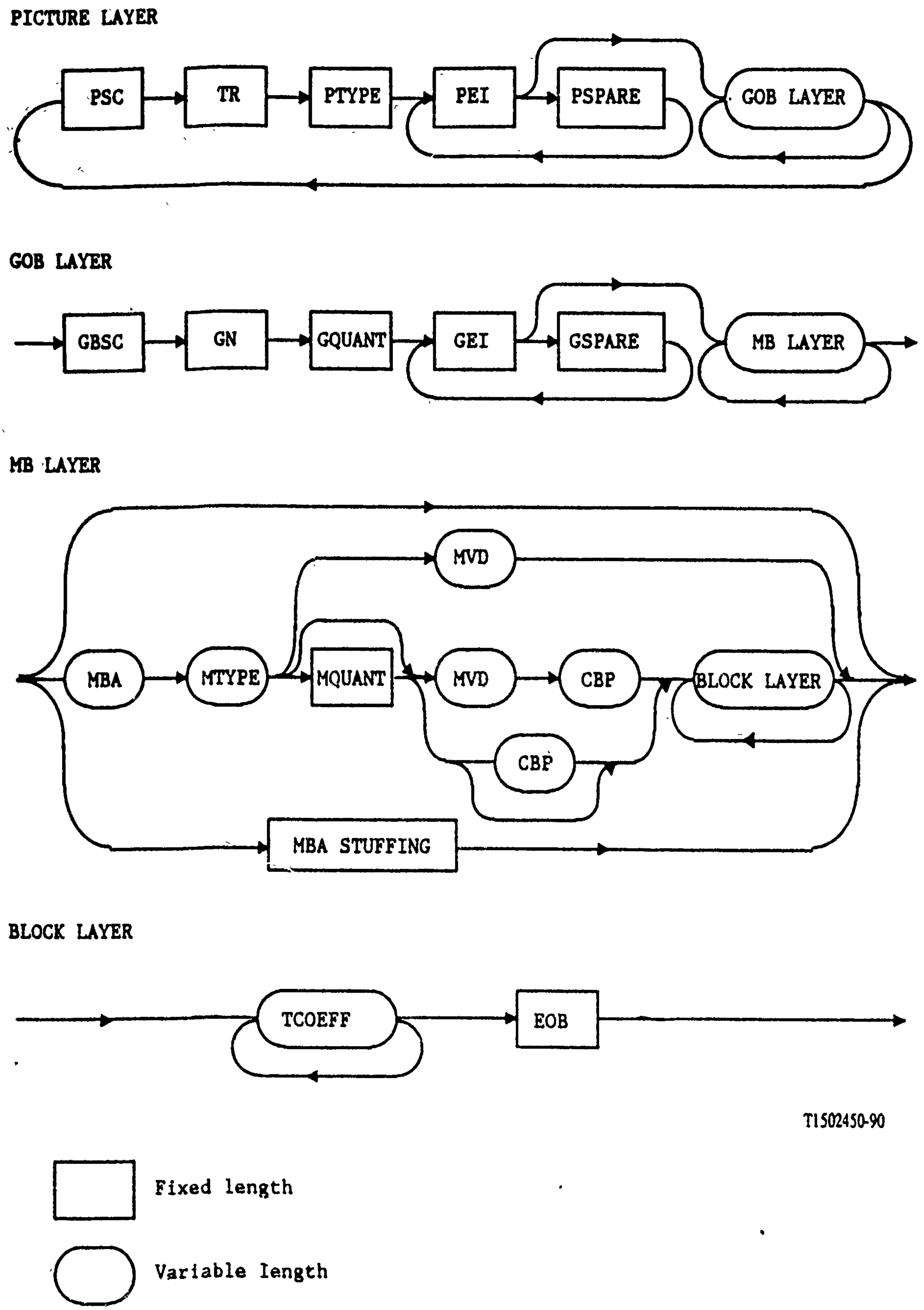


FIGURE 4/H.261

Syntax diagram for the video multiplex coder

```
 ------------------------   ----------------   ------------
| PSC : TR : PTYPE : PEI :   PSPASRE : PEI :      GOB Data |
 ------------------------   ----------------   ------------
```

FIGURE 5/H.261

Structure of picture layer

Picture Start Code (PSC) 20 bits

A word of 20 bits. Its value is 0000 0000 0000 0001 0000.

Temporal Reference (TR) 5 bits

A 5-bit number which can have 32 possible values. It is formed by incrementing its value in the previously transmitted picture header by one plus the number of non-transmitted pictures (at 29.97 Hz) since that last transmitted one. The arithmetic is performed with only the five LSBs.

Type Information (PTYPE) 6 bits

Information about the complete picture:

- Bit 1: Split screen indicator. "0" off, "1" on.
- Bit 2: Document camera indicator. "0" off, "1" on.
- Bit 3: Freeze Picture Release. "0" off, "1" on.
- Bit 4: Source Format. "0" QCIF, "1" CIF.
- Bits 5 to 6: Spare.

Extra Insertion Information (PEI) 1 bit

A bit which when set to "1" signals the presence of the following optional data field.

Spare Information (PSPARE) 0/8/16 ... bits

If PEI is set to "1", then 9 bits follow consisting of 8 bits of data (PSPARE) and then another PEI bit to indicate if a further 9 bits follow and so on. Encoders must not insert PSPARE until specified by the CCITT. Decoders must be designed to discard PSPARE if PEI is set to 1. This will allow the CCITT to specify future "backward" compatible additions in PSPARE.

4.2.2 Group of blocks layer

Each picture is divided into groups of blocks (GOBs). A group of blocks (GOB) comprises one twelfth of the CIF or one third of the QCIF picture areas (see Figure 6/H.261). A GOB relates to 176 pels by 48 lines of Y and the spatially corresponding 88 pels by 24 lines of each of C_B and C_R.

Data for each group of blocks consists of a GOB header followed by data for macroblocks. The structure is shown in Figure 7/H.261. Each GOB header is transmitted once between Picture Start Codes in the CIF or QCIF sequence numbered in Figure 6/H.261, even if no macroblock data is present in that GOB.

Group of blocks Start Code (GBSC) 16 bits

A word of 16 bits, 0000 0000 0000 0001.

```
 ----------------------          ---------------------
|    1     |     2     |        |                     |
|----------------------|        |          1          |
|    3     |     4     |        |                     |
|----------------------|        |---------------------|
|    5     |     6     |        |                     |
|----------------------|        |          3          |
|    7     |     8     |        |                     |
|----------------------|        |---------------------|
|    9     |    10     |        |                     |
|----------------------|        |          5          |
|   11     |    12     |        |                     |
 ----------------------          ---------------------
          CIF                            QCIF
```

FIGURE 6/H.261

Arrangement of GOBs in a picture

```
 ----------------------------    --------------    -----------
| GBSC  : GN : GQUANT : GEI :    GSPARE : GEI :     MB Data  |
 ----------------------------    --------------    -----------
```

FIGURE 7/H.261

Structure of group of blocks layer

Group Number (GN)   4 bits

Four bits indicating the position of the group of blocks. The bits are the binary representation of the number in Figure 6/H.261. Group numbers 13, 14 and 15 are reserved for future use. Group number 0 is used in the PSC.

Quantizer Information (GQUANT)   5 bits

A fixed length codeword of 5 bits which indicates the quantizer to be used in the group of blocks until overridden by any subsequent MQUANT. The codewords are the natural binary representations of the values of QUANT (§ 4.2.4) which, being half the step sizes, range from 1 to 31.

Extra Insertion Information (GEI) 1 bit

A bit which when set to "1" signals the presence of the following optional data field.

Spare Information (GSPARE)   0/8/16 ... bits

If GEI is set to "1", then 9 bits follow consisting of 8 bits of data (GSPARE) and then another GEI bit to indicate if a further 9 bits follow and so on. Encoders must not insert GSPARE until specified by the CCITT. Decoders must be designed to discard GSPARE if GEI is set to 1. This will allow the CCITT to specify future "backward" compatible additions in GSPARE.

Note - Emulation of start codes may occur if the future specification of GSPARE has no restrictions on the final GSPARE data bits.

4.2.3 Macroblock layer

Each GOB is divided into 33 macroblocks as shown in Figure 8/H.261. A macroblock relates to 16 pels by 16 lines of Y and the spatially corresponding 8 pels by 8 lines of each of C_B and C_R.

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

FIGURE 8/H.261

Arrangement of macroblocks in a GOB

Data for a macroblock consists of a MB Header followed by data for blocks (Figure 9/H.261). MQUANT, MVD and CBP are present when indicated by MTYPE.

MBA	MTYPE	MQUANT	MVD	CBP	Block Data

FIGURE 9/H.261

Structure of macroblock layer

Macroblock Address (MBA)   Variable Length

A variable length codeword indicating the position of a macroblock within a group of blocks. The transmission order is as shown in Figure 8/H.261. For the first transmitted macroblock in a GOB, MBA is the absolute address in Figure 8/H.261. For subsequent macroblocks, MBA is the difference between the absolute addresses of the macroblock and the last transmitted macroblock. The code table for MBA is given in Table 1/H.261.

An extra codeword is available in the table for bit stuffing immediately after a GOB header or a coded macroblock (MBA Stuffing). This codeword should be discarded by decoders.

The VLC for start code is also shown in Table 1/H.261.

TABLE 1/H.261

VLC table for macroblock addressing

MBA	CODE	MBA	CODE
1	1	17	0000 0101 10
2	011	18	0000 0101 01
3	010	19	0000 0101 00
4	0011	20	0000 0100 11
5	0010	21	0000 0100 10
6	0001 1	22	0000 0100 011
7	0001 0	23	0000 0100 010
8	0000 111	24	0000 0100 001
9	0000 110	25	0000 0100 000
10	0000 1011	26	0000 0011 111
11	0000 1010	27	0000 0011 110
12	0000 1001	28	0000 0011 101
13	0000 1000	29	0000 0011 100
14	0000 0111	30	0000 0011 011
15	0000 0110	31	0000 0011 010
16	0000 0101 11	32	0000 0011 001
		33	0000 0011 000
		MBA Stuffing	0000 0001 111
		Start code	0000 0000 0000 0001

MBA is always included in transmitted macroblocks.

Macroblocks are not transmitted when the contain no information for that part of the picture.

Type Information (MTYPE) Variable Length

Variable length codewords giving information about the macroblock and which data elements are present. Macroblock types, included elements and VLC words are listed in Table 2/H.261.

MTYPE is always included in transmitted macroblocks.

TABLE 2/H.261

VLC table for MTYPE

Prediction	MQUANT	MVD	CBP	TCOEFF	VLC
Intra				x	0001
Intra	x			x	0000 001
Inter			x	x	1
Inter	x		x	x	0000 1
Inter + MC		x			0000 0000 1
Inter + MC		x	x	x	0000 0001
Inter + MC	x	x	x	x	0000 0000 01
Inter + MC + FIL		x			001
Inter + MC + FIL		x	x	x	01
Inter + MC + FIL	x	x	x	x	0000 01

Note 1 - "x" means that the item is present in the macroblock.

Note 2 - It is possible to apply the filter in a non-motion compensated macroblock by declaring it as MC + FIL but with a zero vector.

Quantizer (MQUANT) 5 bits

MQUANT is present only if so indicated by MTYPE.

A codeword of 5 bits signifying the quantizer to be used for this and any following blocks in the group of blocks until overridden by any subsequent MQUANT.

Codewords for MQUANT are the same as for GQUANT.

Motion Vector Data (MVD) Variable length

Motion Vector Data is included for all MC macroblocks. MVD is obtained from the macroblock vector by subtracting the vector of the preceding macroblock. For this calculation the vector of the preceding macroblock is regarded as zero in the following three situations:

1) Evaluating MVD for macroblocks 1, 12 and 23.

2) Evaluating MVD for macroblocks in which MBA does not represent a difference of 1.

3) MTYPE of the previous macroblock was not MC.

MVD consists of a variable length codeword for the horizontal component followed by a variable length codeword for the vertical component. Variable length codes are given in Table 3/H.261.

Advantage is taken of the fact that the range of motion vector values is constrained. Each VLC word represents a pair of difference values. Only one of the pair will yield a macroblock vector falling within the permitted range.

Coded Block Pattern (CBP) Variable length

CBP is present if indicated by MTYPE. The codeword gives a pattern number signifying those blocks in the macroblock for which at least one transform coefficient is transmitted. The pattern number is given by:

$$32*P_1 + 16*P_2 + 8*P_3 + 4*P_4 + 2*P_5 + P_6$$

where P_n is 1 if any coefficient is present for block n, else 0. Block numbering is given in Figure 10/H.261.

The codewords for CBP are given in Table 4/H.261.

TABLE 3/H.261

VLC table for MVD

MVD			CODE
-16	&	16	0000 0011 001
-15	&	17	0000 0011 011
-14	&	18	0000 0011 101
-13	&	19	0000 0011 111
-12	&	20	0000 0100 001
-11	&	21	0000 0100 011
-10	&	22	0000 0100 11
-9	&	23	0000 0101 01
-8	&	24	0000 0101 11
-7	&	25	0000 0111
-6	&	26	0000 1001
-5	&	27	0000 1011
-4	&	28	0000 111
-3	&	29	0001 1
-2	&	30	0011
-1			011
0			1
1			010
2	&	-30	0010
3	&	-29	0001 0
4	&	-28	0000 110
5	&	-27	0000 1010
6	&	-26	0000 1000
7	&	-25	0000 0110
8	&	-24	0000 0101 10
9	&	-23	0000 0101 00
10	&	-22	0000 0100 10
11	&	-21	0000 0100 010
12	&	-20	0000 0100 000
13	&	-19	0000 0011 110
14	&	-18	0000 0011 100
15	&	-17	0000 0011 010

TABLE 4/H.261

VLC table for CBP

CBP	CODE	CBP	CODE
60	111	35	0001 1100
4	1101	13	0001 1011
8	1100	49	0001 1010
16	1011	21	0001 1001
32	1010	41	0001 1000
12	1001 1	14	0001 0111
48	1001 0	50	0001 0110
20	1000 1	22	0001 0101
40	1000 0	42	0001 0100
28	0111 1	15	0001 0011
44	0111 0	51	0001 0010
52	0110 1	23	0001 0001
56	0110 0	43	0001 0000
1	0101 1	25	0000 1111
61	0101 0	37	0000 1110
2	0100 1	26	0000 1101
62	0100 0	38	0000 1100
24	0011 11	29	0000 1011
36	0011 10	45	0000 1010
3	0011 01	53	0000 1001
63	0011 00	57	0000 1000
5	0010 111	30	0000 0111
9	0010 110	46	0000 0110
17	0010 101	54	0000 0101
33	0010 100	58	0000 0100
6	0010 011	31	0000 0011 1
10	0010 010	47	0000 0011 0
18	0010 001	55	0000 0010 1
34	0010 000	59	0000 0010 0
7	0001 1111	27	0000 0001 1
11	0001 1110	39	0000 0001 0
19	0001 1101		

4.2.4 Block layer

A macroblock comprises four luminance blocks and one of each of the two colour difference blocks (Figure 10/H.261).

```
 -------      -------      -------
| 1 | 2 |    |       |    |       |
|-------|    |   5   |    |   6   |
| 3 | 4 |    |       |    |       |
 -------      -------      -------
    Y            CB           CR
```

FIGURE 10/H.261

Arrangement of blocks in a macroblock

Data for a block consists of codewords for transform coefficients followed by an end of block marker (Figure 11/H.261). The order of block transmission is as in Figure 10/H.261.

```
 --------------
| TCOEFF | EOB |
 --------------
```

FIGURE 11/H.261

Structure of block layer

Transform Coefficients (TCOEFF)

Transform coefficient data is always present for all six blocks in a macroblock when MTYPE indicates INTRA. In other cases MTYPE and CBP signal which blocks have coefficient data transmitted for them. The quantized transform coefficients are sequentially transmitted according to the sequence given in Figure 12/H.261.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

```
----------> increasing cycles per
|           picture width
|
|
|
V

increasing cycles per
picture height
```

FIGURE 12/H.261

Transmission order for transform coefficients

The most commonly occurring combinations of successive zeros (RUN) ard the following value (LEVEL) are encoded with variable length codes. Other combinations of (RUN, LEVEL) are encoded with a 20-bit word consisting of 6 bits ESCAPE, 6 bits RUN and 8 bits LEVEL. For the variable length encoding there are two code tables, one being used for the first transmitted LEVEL in INTER, INTER+MC and INTER+MC+FIL blocks, the second for all other LEVELs except the first one in INTRA blocks which is fixed length coded with 8 bits.

Codes are given in Table 5/H.261.

TABLE 5/H.261

VLC table for TCOEFF

The most commonly occurring combinations of zero-run and the following value are encoded with variable length codes as listed in the table below. End of block (EOB) is in this set. Because CBP indicates those blocks with no coefficient data, EOB cannot occur as the first coefficient. Hence EOB can be removed from the VLC table for the first coefficient.

The last bit "s" denotes the sign of the level, "0" for positive
"1" for negative.

RUN	| LEVEL |	CODE
EOB		10
0	1	1s IF FIRST COEFFICIENT IN BLOCK (<u>Note</u> - Never used in INTRA macroblocks)
0	1	11s NOT FIRST COEFFICIENT IN BLOCK
0	2	0100 s
0	3	0010 1s
0	4	0000 110s
0	5	0010 0110 s
0	6	0010 0001 s
0	7	0000 0010 10s
0	8	0000 0001 1101 s
0	9	0000 0001 1000 s
0	10	0000 0001 0011 s
0	11	0000 0001 0000 s
0	12	0000 0000 1101 0s
0	13	0000 0000 1100 1s
0	14	0000 0000 1100 0s
0	15	0000 0000 1011 1s
1	1	011s
1	2	0001 10s
1	3	0010 0101 s
1	4	0000 0011 00s
1	5	0000 0001 1011 s
1	6	0000 0000 1011 0s
1	7	0000 0000 1010 1s
2	1	0101 s
2	2	0000 100s
2	3	0000 0010 11s
2	4	0000 0001 0100 s
2	5	0000 0000 1010 0s

3	1	0011 1s
3	2	0010 0100 s
3	3	0000 0001 1100 s
3	4	0000 0000 1001 1s
4	1	0011 0s
4	2	0000 0011 11s
4	3	0000 0001 0010 s
5	1	0001 11s
5	2	0000 0010 01s
5	3	0000 0000 1001 0s
6	1	0001 01s
6	2	0000 0001 1110 s
7	1	0001 00s
7	2	0000 0001 0101 s
8	1	0000 111s
8	2	0000 0001 0001 s
9	1	0000 101s
9	2	0000 0000 1000 1s
10	1	0010 0111 s
10	2	0000 0000 1000 0s
11	1	0010 0011 s
12	1	0010 0010 s
13	1	0010 0000 s
14	1	0000 0011 10s
15	1	0000 0011 01s
16	1	0000 0010 00s
17	1	0000 0001 1111 s
18	1	0000 0001 1010 s
19	1	0000 0001 1001 s
20	1	0000 0001 0111 s
21	1	0000 0001 0110 s
22	1	0000 0000 1111 1s
23	1	0000 0000 1111 0s
24	1	0000 0000 1110 1s
25	1	0000 0000 1110 0s
26	1	0000 0000 1101 1s
ESCAPE		0000 01

The remaining :ombinations of (RUN, LEVEL) are encoded with a 20-bit word[1] consisting of 6 bits ESCAPE, 6 bits RUN and 8 bits LEVEL.

[1] Use of this 20-bit word form for encoding the combinations listed in the VLC table is not prohibited.

RUN is a 6 bit fixed length code.

RUN	CODE
0	0000 00
1	0000 01
2	0000 10
.	.
.	.
63	1111 11

LEVEL is an 8 bit fixed length code.

LEVEL	CODE
-128	FORBIDDEN
-127	1000 0001
.	.
-2	1111 1110
-1	1111 1111
0	FORBIDDEN
1	0000 0001
2	0000 0010
.	.
127	0111 1111

For all coefficients other than the INTRA dc one the reconstruction levels (REC) are in the range -2048 to 2047 and are given by clipping the results of the following formulae:

$$\left.\begin{array}{l} \text{REC} = \text{QUANT}*(2*\text{LEVEL}+1) \quad ;\ \text{LEVEL}>0 \\ \text{REC} = \text{QUANT}*(2*\text{LEVEL}-1) \quad ;\ \text{LEVEL}<0 \end{array}\right\} \text{QUANT} = \text{"odd"}$$

$$\left.\begin{array}{l} \text{REC} = \text{QUANT}*(2*\text{LEVEL}+1)-1;\ \text{LEVEL}>0 \\ \text{REC} = \text{QUANT}*(2*\text{LEVEL}-1)+1;\ \text{LEVEL}<0 \end{array}\right\} \text{QUANT} = \text{"even"}$$

$$\text{REC} = 0;\ \text{LEVEL}=0$$

<u>Note</u> - QUANT ranges from 1 to 31 and is transmitted by either GQUANT OR MQUANT.

TABLE 6/H.261

<u>Reconstruction levels (REC)</u>

	QUANT												
LEVEL	1	2	3	4	.	8	9	.	17	18	.	30	31
-127	-255	-509	-765	-1019	.	-2039	-2048	.	-2048	-2048	.	-2048	-2048
-126	-253	-505	-759	-1011	.	-2023	-2048	.	-2048	-2048	.	-2048	-2048
.	.	.	.	.	.	.	.	.	.	.	.	.	.
-2	-5	-9	-15	-19	.	-39	-45	.	-85	-89	.	-149	-155
-1	-3	-5	-9	-11	.	-23	-27	.	-51	-53	.	-89	-93
0	0	0	0	0	.	0	0	.	0	0	.	0	0
1	3	5	9	11	.	23	27	.	51	53	.	89	93
2	5	9	15	19	.	39	45	.	85	89	.	149	155
3	7	13	21	27	.	55	63	.	119	125	.	209	217
4	9	17	27	35	.	71	81	.	153	161	.	269	279
5	11	21	33	43	.	87	99	.	187	197	.	329	341
.	.	.	.	.	.	.	.	.	.	.	.	.	.
56	113	225	339	451	.	903	1017	.	1921	2033	.	2047	2047
57	115	229	345	459	.	919	1035	.	1955	2047	.	2047	2047
58	117	233	351	467	.	935	1053	.	1989	2047	.	2047	2047
59	119	237	357	475	.	951	1071	.	2023	2047	.	2047	2047
60	121	241	363	483	.	967	1089	.	2047	2047	.	2047	2047
.	.	.	.	.	.	.	.	.	.	.	.	.	.
125	251	501	753	1003	.	2007	2047	.	2047	2047	.	2047	2047
126	253	505	759	1011	.	2023	2047	.	2047	2047	.	2047	2047
127	255	509	765	1019	.	2039	2047	.	2047	2047	.	2047	2047

Note - Reconstruction levels are symmetrical with respect to the sign of LEVEL except for 2047/-2048.

For INTRA blocks the first coefficient is nominally the transform dc value linearly quantized with a step size of 8 and no dead-zone. The resulting values are represented with 8 bits. A nominally black block will give 0001 0000 and a nominally white one 1110 1011. The code 0000 0000 is not used. The code 1000 0000 is not used, the reconstruction level of 1024 being coded as 1111 1111 (see Table 7/H.261).

Coefficients after the last non-zero one are not transmitted. EOB (end of block code) is always the last item in blocks for which coefficients are transmitted.

TABLE 7/H.261

Reconstruction levels for INTRA-mode dc coefficient

FLC	Reconstruction level into inverse transform
0000 0001 (1)	8
0000 0010 (2)	16
0000 0011 (3)	24
.	.
.	.
0111 1111 (127)	1016
1111 1111 (255)	1024
1000 0001 (129)	1032
.	.
.	.
1111 1101 (253)	2024
1111 1110 (254)	2032

Note - The decoded value corresponding to FLC "n" is 8n except FLC 255 gives 1024.

4.3 Multipoint considerations

The following facilities are provided to support switched multipoint operation.

4.3.1 Freeze picture request

Causes the decoded to freeze its displayed picture until a freeze picture release signal is received or a timeout period of at least six seconds has expired. The transmission of this signal is via external means (for example by H.221).

4.3.2 Fast update request

Causes the encoder to encode its next picture in INTRA mode with coding parameters such as to avoid buffer overflow. The transmission method for this signal is via external means (for example by H.221).

4.3.3 Freeze picture release

A signal from an encoder which has responded to a Fast Update Request and allows a decoder to exit from its freeze picture mode and display decoded pictures in the normal manner. This signal is transmitted by bit 3 of PTYPE (see § 4.2.1) in the picture header of the first picture coded in response to the Fast Update Request.

5. Transmission coder

5.1 Bit rate

The transmission clock is provided externally (for example from an I.420 interface).

5.2 Video data buffering

The encoder must control its output bitstream to comply with the requirements of the Hypothetical Reference Decoder defined in Annex 2.

When operating with CIF the number of bits created by coding any single picture must not exceed 256 kbit/s. K = 1024.

When operating with QCIF the number of bits created by coding any single picture must not exceed 64 kbit/s.

In both the above cases the bit count includes the Picture Start Code and all other data related to that picture including PSPARE, GSPARE and MBA Stuffing. The bit count does not include error correction framing bits, fill indicator (Fi), fill bits or error correction parity information described in § 5.4 below.

Video data must be provided on every valid clock cycle. This can be ensured by the use of either the fill bit indicator (Fi) and subsequent fill all 1's bits in the error corrector block framing (see Figure 13/H.261) or MBA Stuffing (§ 4.2.3) or both.

5.3 Video coding delay

This item is included in this Recommendation because the video encoder and video decoder delays need to be known to allow audio compensation delays to be fixed when H.261 is used to form part of a conversational service. This will allow lip synchronization to be maintained. Annex 3 recommends a method by which the delay figures are established. Other delay measurement methods may be used but they must be designed in a way to produce similar results to the method given in Annex 3.

5.4 Forward Error Correction for coded video signal

5.4.1 Error correcting code

The transmitted bitstream contains a BCH (511,493) Forward Error Correction Code. Use of this by the decoder is optional.

5.4.2 Generator polynomial

$$g(x) = (x^9 + x^4 + 1)(x^9 + x^6 + x^4 + x^3 + 1)$$

Example: for the input data of "01111 ... 11" (493 bits) the resulting correction parity bits are "011011010100011011" (18 bits).

5.4.3 Error correction framing

To allow the video data and error correction parity information to be identified by a decoder an error correction framing pattern is included. This consists of a multiframe of eight frames, each frame comprising 1 bit framing, 1 bit fill indicator (Fi), 492 bits of coded data (or fill all 1s) and 18 bits parity. The frame alignment pattern is:

$$(S_1S_2S_3S_4S_5S_6S_7S_8) = (00011011).$$

See Figure 13/H.261 for the frame arrangement. The parity is calculated against the 493-bits including fill indicator (Fi).

The fill indicator (Fi) can be set to zero by an encoder. In this case only 492 consecutive fill bits (fill all 1s) plus parity are sent and no coded data is transmitted. This may be used to meet the requirement in § 5.2 to provide video data on every valid clock cycle.

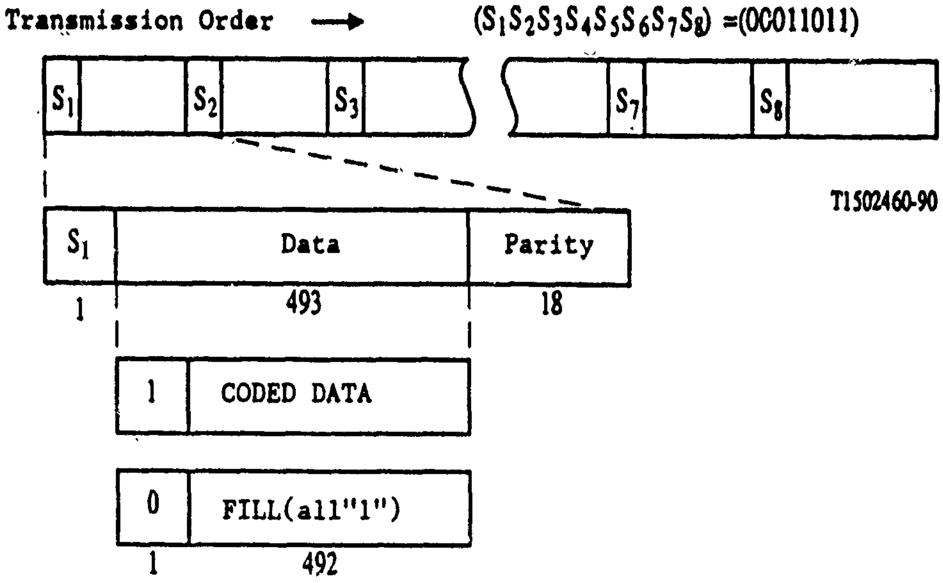


FIGURE 13/H.261

Error correcting frame

5.4.4 Relock time for error corrector framing

Three consecutive error correction framing sequences (24 bits) should be received before frame lock is deemed to have been achieved. The decoder should be designed such that frame lock will be re-established within 34000 bits after an error corrector framing phase change.

Note - This assumes that the video data does not contain three correctly phased emulations of the error correction framing sequence during the relocking period.

ANNEX 1

(to Recommendation H.261)

Inverse transform accuracy specification

1. Generate random integer pel data values in the range -L to +H according to the random number generator given below ("C" version). Arrange into 8 by 8 blocks. Data set of 10,000 blocks should each be generated for (L = 256, H = 255), (L = H = 5) and (L = H = 300).

2. For each 8 by 8 block, perform a separable, orthonormal, matrix multiply, Forward Discrete Cosine Transform using at least 64-bit floating point accuracy.

$$F(u,v) = 1/4\ C(u)\ C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos[Pi(2x + 1)u/16] \cos[Pi(2y + 1)v/16]$$

with u, v, x, y = 0, 1, 2, ...7

where x,y = spatial coordinates in the pel domain

u,v = coordinates in the transform domain

$C(u) = 1/\sqrt{2}$ for u = 0, otherwise 1
$C(v) = 1/\sqrt{2}$ for v = 0, otherwise 1

3. For each block, round the 64 resulting transformed coefficients to the nearest integer values. Then clip them to the range -2048 to +2047. This is the 12-bit input data to the inverse transform.

4. For each 8 by 8 block of 12-bit data produced by step 3, perform a separable, orthonormal, matrix multiply, Inverse Discrete Transform (IDCT) using at least 64-bit floating point accuracy. Round the resulting pels to the nearest integer and clip to the range -256 to +255. These blocks of 8 by 8 pels are the "reference" IDCT input data.

5. For each 8 by 8 block produced by step 3, apply the IDCT under test and clip the output to the range -256 to +255. These blocks of 8 by 8 pels are the "test" IDCT output data.

6. For each of the 64 IDCT output pels, and for each of the 10,000 block data sets generated above, measure the peak, mean and mean square error between the "reference" and the "test" data.

7. For any pel, the peak error should not exceed 1 in magnitude.

For any pel, the mean square error should not exceed 0.06.

Overall, the mean square error should not exceed 0.02.

For any pel, the mean error should not exceed 0.015 in magnitude.

Overall, the mean error should not exceed 0.0015 in magnitude.

8. All zeros in must produce all zeros out.

9. Re-run the measurements using exactly the same data values of step 1, but change the sign on each pel.

"C" Program for random number generation

```
/* L and H must be long, that is 32 bits */
long rand(L,H)
long      L,H;
{
  static long  randx = 1;  /* long is 32 bits */
  static double  z = (double)0x7fffffff;

  long i,j;
  double x;                      /* double is 64 bits */

  randx = (randx * 1103515245) + 12345;
  i = randx & 0x7ffffffe;       /* keep 30 bits */
  x = ( (double)i ) / z;        /* range 0 to 0.99999... */
  x * = (L+H+1):                /* range 0 to < L+H+1 */
  j = x;                        /* truncate to integer */
  return( j - L );              /* range -L to H */
}
```

ANNEX 2

(to Recommendation H.261)

Hypothetical Reference Decoder

The Hypothetical Reference Decoder (HRD) is defined as follows:

1. The HRD and the encoder have the same clock frequency as well as the same CIF rate, and are operated synchronously.

2. The HRD receiving buffer size is (B + 256 kbit/s). The value of B is defined as follows:

$B = 4R_{max}/29.97$ where R_{max} is the maximum video bit rate to be used in the connection.

3. The HRD buffer is initially empty.

4. The HRD buffer is examined at CIF intervals (≃33 ms). If at least one complete coded picture is in the buffer then all the data for the earliest picture is instantaneously removed (e.g. at t_{n+1} in Figure A.1/H.261). Immediately after removing the above data the buffer occupancy must be less than B. This is a requirement on the coder output bitstream including coded picture data and MBA stuffing but not error correction framing bits, fill indicator (Fi), fill bits or error correction parity information described in § 5.4.

To meet this requirement the number of bits for the (N+1)th coded picture d_{N+1} must satisfy:

$$d_{N+1} > B_N + \int_{t_N}^{t_{N+1}} R(t)dt - B$$

where B_N is buffer occupancy just after the time t_N,

t_N is the time the Nth coded picture is removed from the HRD buffer,

R(t) is the video bit rate at the time t.

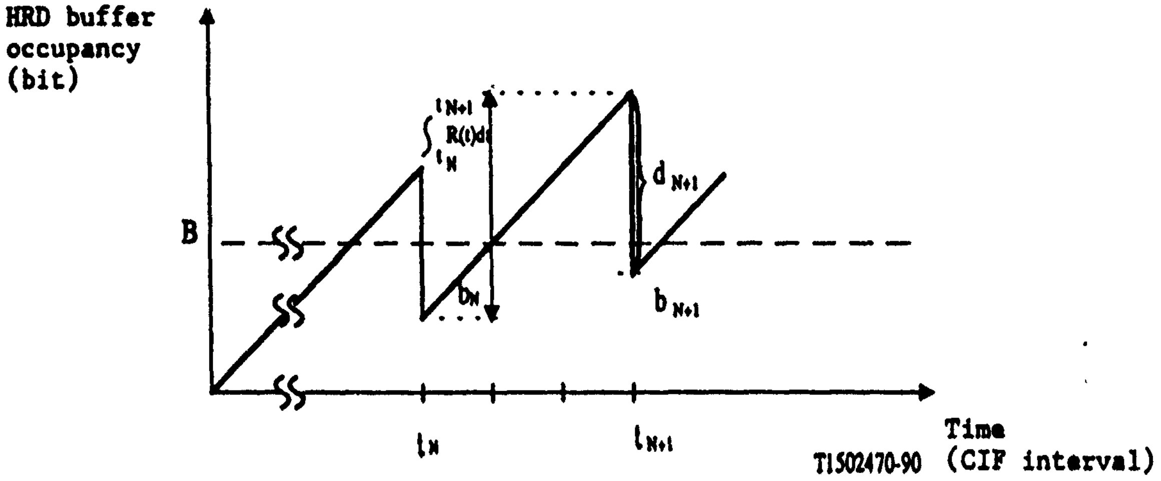


FIGURE A.1/H.261

HRD buffer occupancy

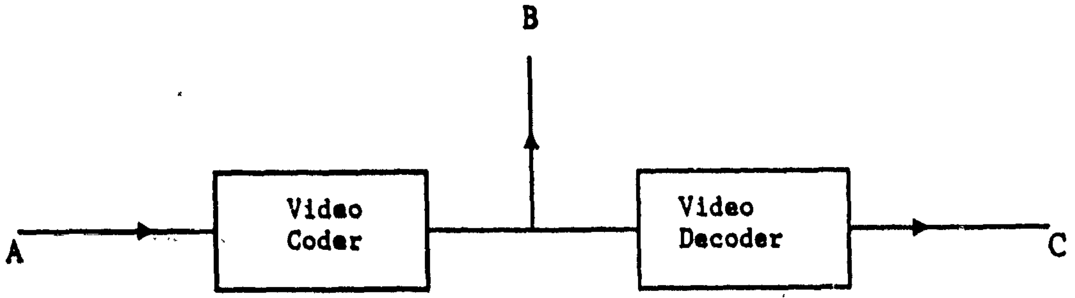
Note - Time (t_{N+1} - t_N) is an integer number of CIF picture periods (1/29.97, 2/29.97, 3/29.97, ...).

ANNEX 3

(to Recommendation H.261)

Codec delay measurement method

The video encoder and video decoder delays will vary depending on implementation. The delay will also depend on the picture format (QCIF, CIF) and data rate in use. This section specifies the method by which the delay figures are established for a particular design. To allow correct audio delay compensation the overall video delay needs to be established from a user perception point of view under typical viewing conditions.



T1502480-90

FIGURE A.2/H.261

Measuring points

Point A is the video input to the video coder. Point B is thu channel output from the video terminal (i.e. including any FEC, channel framing etc.). Point C is the video output from the decoder.

A video sequence lasting more than 100 seconds is connected to the video coder input (point A) in Figure A.2/H.261 above. The video sequence should have the following characteristics:

- it should contain a typical moving scene consistent with the intended purpose of the video codec;
- it should produce a minimum coded picture rate of 7.5 Hz at the bit rate in use;
- it should contain a visible identification mark at intervals throughout the length of the sequence. The visible identification should change every 97 video input frames and be located within the picture area represented by the first GOB in the picture. For example, the first block in the picture could change from black to white at intervals of 97 video frame periods. The identification mark should be chosen so that it can be detected at point B and does not significantly contribute to the overall coding performance.

The codec and video sequence should be arranged so that the bitstream contains less than 10% stuffing (MBA stuffing + error correction fill bits).

The encoder delay is obtained by measuring the time from when the visible identification changes at point A to the time that the change is detected at point B. Similarly, the decoder delay is obtained by taking measurements at points B and C.

Several measurements should be made during the sequence length and the average period obtained. Several tests should be made to ensure that a consistent average figure can be obtained for both encoder and decoder delay times.

Average results should be obtained for each combination of picture format and bit rate within the capability of the particular codec design.

Note - Due to pre- and post-temporal processing it may be necessary to take a mid-level for establishing the transition of the identification mark at points B and C.