IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

U.S. PHILIPS CORPORATION, Plaintiff, v. EASTMAN KODAK COMPANY, Defendant.	)	Civil Action No. 06-251-GMS

NOTICE OF SERVICE OF SUBPOENA

TO: COUNSEL OF RECORD ON ATTACHED SERVICE LIST

PLEASE TAKE NOTICE that pursuant to Rule 45 of the Federal Rules of Civil Procedure, defendant Eastman Kodak Company provides notice of service of subpoena on the following persons or entities as indicated:

1. Forgent Networks, Inc. A copy of the subpoena is attached hereto as Exhibit A.

2. AT&T Corp. A copy of the subpoena is attached hereto as Exhibit B.

Dated: June 29, 2007

/s/ Kristen H. Cramer

Francis DiGiovanni (#3189)
Kristen H. Cramer (#4512)
CONNOLLY BOVE LODGE & HUTZ LLP
The Nemours Building
1007 N. Orange Street
Wilmington, DE 19899
Phone (302) 658-9141
e-mail: kcramer@cblh.com

John Allcock
Sean C. Cunningham
Tim Lohse
John D. Kinton
Jesse Hindman
DLA Piper US LLP
401 B Street, Suite 1700
San Diego, CA 92101-4297
Phone (619) 699-2700

Attorneys for Defendant/Counterclaimant
Eastman Kodak Company

CERTIFICATE OF SERVICE

I, Kristen H. Cramer, hereby certify that on June 29, 2007, I caused to be electronically filed a true and correct copy of the foregoing document with the Clerk of the Court using CM/ECF, which will send notification that such filing is available for viewing and downloading to registered counsel of record via e-mail.

I hereby further certify that on June 29, 2007, I caused a copy of the foregoing document to be served on the following counsel of record by the manner so indicated:

BY E-MAIL AND HAND DELIVERY:
Steven J. Balick
John G. Day
Tiffany Geyer Lydon
222 Delaware Avenue, 17th Floor
P.O. Box 1150
Wilmington, DE 19899

BY E-MAIL AND U.S. MAIL:
Thomas W. Winland
Steven M. Anzalone
Frank A. DeCosta, III
Joyce Craig-Rient
Finnegan Henderson Farabow Garrett and Dunner LLP
901 New York Avenue, NW
Washington, DC 20001

Kristen H. Cramer
Kristen H. Cramer (#4512)

EXHIBIT
A

Issued by the

UNITED STATES DISTRICT COURT

WESTERN DISTRICT OF TEXAS

U.S. PHILIPS CORPORATION

Plaintiff

V.

EASTMAN KODAK COMPANY

Defendant.

SUBPOENA IN A CIVIL CASE

Case Number:[1] Civil Action No. 06-00251
USDC, District of Delaware

TO: Forgent Networks, Inc.
108 Wild Basin Drive
Austin, TX 78746
888.323.8835
512.437.2700

Note: Mr. Jay Peterson, Forgent's Chief Financial Officer, is authorized to accept service of process.

☐ YOU ARE COMMANDED to appear in the United States District court at the place, date, and time specified below to testify in the above case.

PLACE OF TESTIMONY	COURTROOM
	DATE AND TIME

☒ YOU ARE COMMANDED to appear at the place, date, and time specified below to testify at the taking of a deposition in the above case. See Attachment A for deposition topics.

PLACE OF DEPOSITION	DATE AND TIME
DLA Piper US LLP 1221 S. MoPac Expressway, Suite 400 Austin, TX 78746 512.457.7000	July 27, 2007 9:00 a.m.

☒ YOU ARE COMMANDED to produce and permit inspection and copying of the following documents or objects at the place, date, and time specified below (list documents or objects):

See Attachment A

PLACE	DATE AND TIME
DLA Piper US LLP 401 B Street, Suite 1700 San Diego, CA 92101	July 13, 2007

☐ YOU ARE COMMANDED to permit inspection of the following premises at the date and time specified below.

PREMISES	DATE AND TIME

Any organization not a party to this suit that is subpoenaed for the taking of a deposition shall designate one or more officers, directors, or managing agents, or other persons who consent to testify on its behalf, and may set forth, for each person designated, the matters on which the person will testify. Federal Rules of Civil Procedure, 30(b)(6).

ISSUING OFFICER'S SIGNATURE AND TITLE (INDICATE IF ATTORNEY FOR PLAINTIFF OR DEFENDANT)	DATE
Attorney for Defendant Eastman Kodak Company Kristen H. Cramer (#4512)	June 29, 2007

ISSUING OFFICER'S NAME, ADDRESS AND PHONE NUMBER

Kristen H. Cramer
Connolly Bove Lodge and Hutz
1007 N. Orange St.
P.O. Box 2207
Wilmington, DE 19899
302.658.9141

(See Rule 45, Federal Rules of Civil Procedure, Subdivisions (c), (d), and (e), on next page)

[1] If action is pending in district other than district of issuance, state district under case number.

PROOF OF SERVICE

DATE	PLACE
SERVED	
SERVED ON (PRINT NAME)	MANNER OF SERVICE
SERVED BY (PRINT NAME)	TITLE

DECLARATION OF SERVER

I declare under penalty of perjury under the laws of the United States of America that the foregoing information contained in the Proof of Service is true and correct.

Executed on _____
DATE

_____
SIGNATURE OF SERVER

_____
ADDRESS OF SERVER

Rule 45, Federal Rules of Civil Procedure, Subdivisions (c), (d), and (e), as amended on December 1, 2006:

(C) PROTECTION OF PERSONS SUBJECT TO SUBPOENAS.

(1) A party or an attorney responsible for the issuance and service of a subpoena shall take reasonable steps to avoid imposing undue burden or expense on a person subject to that subpoena. The court on behalf of which the subpoena was issued shall enforce this duty and impose upon the party or attorney in breach of this duty an appropriate sanction, which may include, but is not limited to, lost earnings and a reasonable attorney's fee.

(2) (A) A person commanded to produce and permit inspection, copying, testing, or sampling of designated electronically stored information, books, papers, documents or tangible things, or inspection of premises need not appear in person at the place of production or inspection unless commanded to appear for deposition, hearing or trial.

(B) Subject to paragraph (d)(2) of this rule, a person commanded to produce and permit inspection, copying, testing, or sampling may, within 14 days after service of the subpoena or before the time specified for compliance if such time is less than 14 days after service, serve upon the party or attorney designated in the subpoena written objection to producing any or all of the designated materials or inspection of the premises — or to producing electronically stored information in the form or forms requested. If objection is made, the party serving the subpoena shall not be entitled to inspect, copy, test, or sample the materials or inspect the premises except pursuant to an order of the court by which the subpoena was issued. If objection has been made, the party serving the subpoena may, upon notice to the person commanded to produce, move at any time for an order to compel the production, inspection, copying, testing, or sampling. Such an order to compel shall protect any person who is not a party or an officer of a party from significant expense resulting from the inspection, copying, testing, or sampling commanded.

(3) (A) On timely motion, the court by which a subpoena was issued shall quash or modify the subpoena if it

(i) fails to allow reasonable time for compliance;

(ii) requires a person who is not a party or an officer of a party to travel to a place more than 100 miles from the place where that person resides, is employed or regularly transacts business in person, except that, subject to the provisions of clause (c)(3)(B)(iii) of this rule, such a person may in order to attend trial be commanded to travel from any such place within the state in which the trial is held;

(iii) requires disclosure of privileged or other protected matter and no exception or waiver applies; or

(iv) subjects a person to undue burden.

(B) If a subpoena

(i) requires disclosure of a trade secret or other confidential research, development, or commercial information, or

(ii) requires disclosure of an unretained expert's opinion or information not describing specific events or occurrences in dispute and resulting from the expert's study made not at the request of any party, or

(iii) requires a person who is not a party or an officer of a party to incur substantial expense to travel more than 100 miles to attend trial, the court may, to protect a person subject to or affected by the subpoena, quash or modify the subpoena or, if the party in whose behalf the subpoena is issued shows a substantial need for the testimony or material that cannot be otherwise met without undue hardship and assures that the person to whom the subpoena is addressed will be reasonably compensated, the court may order appearance or production only upon specified conditions.

(D) DUTIES IN RESPONDING TO SUBPOENA.

(1) (A) A person responding to a subpoena to produce documents shall produce them as they are kept in the usual course of business or shall organize and label them to correspond with the categories in the demand.

(B) If a subpoena does not specify the form or forms for producing electronically stored information, a person responding to a subpoena must produce the information in a form or forms in which the person ordinarily maintains it or in a form or forms that are reasonably usable.

(C) A person responding to a subpoena need not produce the same electronically stored information in more than one form.

(D) A person responding to a subpoena need not provide discovery of electronically stored information from sources that the person identifies as not reasonably accessible because of undue burden or cost. On motion to compel discovery or to quash, the person from whom discovery is sought must show that the information sought is not reasonably accessible because of undue burden or cost. If that showing is made, the court may nonetheless order discovery from such sources if the requesting party shows good cause, considering the limitations of Rule 26(b)(2)(C). The court may specify conditions for the discovery.

(2) (A) When information subject to a subpoena is withheld on a claim that it is privileged or subject to protection as trial-preparation materials, the claim shall be made expressly and shall be supported by a description of the nature of the documents, communications, or things not produced that is sufficient to enable the demanding party to contest the claim.

(B) If information is produced in response to a subpoena that is subject to a claim of privilege or of protection as trial-preparation material, the person making the claim may notify any party that received the information of the claim and the basis for it. After being notified, a party must promptly return, sequester, or destroy the specified information and any copies it has and may not use or disclose the information until the claim is resolved. A receiving party may promptly present the information to the court under seal for a determination of the claim. If the receiving party disclosed the information before being notified, it must take reasonable steps to retrieve it. The person who produced the information must preserve the information until the claim is resolved.

(e) CONTEMPT. Failure of any person without adequate excuse to obey a subpoena served upon that person may be deemed a contempt of the court from which the subpoena issued. An adequate cause for failure to obey exists when a subpoena purports to require a nonparty to attend or produce at a place not within the limits provided by clause (ii) of subparagraph (c)(3)(A).

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

U.S. PHILIPS CORPORATION, Plaintiff, v. EASTMAN KODAK COMPANY, Defendant.	)	Civil Action No. 06-00251-GMS

EASTMAN KODAK COMPANY'S SUBPOENA *AD TESTIFICANDUM* AND *DUCES TECUM* OF FORGENT NETWORKS, INC.

PLEASE TAKE NOTICE THAT, pursuant to Rules 30 and 45 of the Federal Rules of Civil Procedure, Eastman Kodak Company ("Kodak"), by and through its counsel, will take the deposition of Forgent Networks, Inc. ("Forgent"), on July 27, 2007 at 9:00 a.m. at the offices of DLA Piper US LLP, 1221 S. MoPac Expressway, Suite 400, Austin, Texas 78746, (512) 457-7000 (or at a time and place as otherwise agreed by the parties).

The deposition, on the deposition topics listed in Attachment A, will take place upon oral examination pursuant to the Federal Rules of Civil Procedure before an officer duly authorized by law to administer oaths, and will continue from day to day until completed. Some or all of the deposition may be recorded stenographically and may be recorded by videotape. Some or all of the deposition testimony may involve real-time computer connection between the deposition-taker and stenographer using software such as "LiveNote."

Before the deposition, on July 13, 2007 at the offices of DLA Piper US LLP, 401 B Street, Suite 1700 San Diego, CA 92101 (or at the time and place as otherwise agreed by the parties), Forgent must produce documents in its possession, custody and control in response to the document requests listed in Attachment A. Fed.R.Civ.P. 45(a)(1)(C).

Dated: June 29, 2007

John Allcock
Sean C. Cunningham
Tim Lohse
John D. Kinton
Jesse Hindman
Nicole Wyll
DLA PIPER US LLP
401 B Street, Suite 1700
San Diego, CA 92101-4297
Tel: 619.699.2700
Fax: 619.699.2701
Attorneys for Defendant
Eastman Kodak Company

ATTACHMENT A

TO THE SUBPOENA FOR FORGENT NETWORKS, INC.

DEFINITIONS

1. "Forgent," "you" or "your" means Forgent Networks, Inc., including without limitation all of its corporate locations, predecessors, affiliates, successors, subsidiaries, divisions or parts thereof, including Compression Labs, Inc. and Widcom, Inc.; all past or present directors, officers, agents, representatives, employees, consultants, attorneys, and entities acting in joint-venture or partnership relationships with Forgent; and others acting on behalf of Forgent.

2. "CLI" means Compression Labs, Inc., and, where applicable, its subsidiaries, divisions, affiliates, partners, corporate parents (including Forgent Networks, Inc.), predecessors, successors, officers, agents, assigns, employees, attorneys, accountants, consultants, and/or all other persons acting, or purporting to act, on their behalf.

3. The term "Widcom" means Widcom, Inc., including its predecessors, successors, assigns, past and present subsidiaries, past and present corporate parents, divisions, affiliates, officers, directors, agents and employees, attorneys and other persons acting on its behalf, including Widergren Communications, Inc., and Robert Widergren.

4. The term "Philips" means U.S. Philips Corporation including its predecessors, successors, assigns, past and present subsidiaries, past and present corporate parents, past or present subsidiaries of its corporate parents, divisions, affiliates, officers, directors, agents and employees, attorneys and other persons acting on its behalf, including Royal Philips Electronics, Philips Kommunikations Industrie AG (Germany), Philips Research Laboratories (United Kingdom), Philips Data Systems, Philips International B.V. (Netherlands), Philips

Telecommuicatie Industrie (Netherlands), Philips Consumer Electronics B.V. (Netherlands), Philips Patentverwaltung GmbH, N. V. Philips Gloeilampenfabrieken and Peter Vogel.

5. "Conception" means the formation in the mind of the inventor of a definite and permanent idea of the complete and operative invention, as the invention is to be applied in practice. *See, e.g., Sewall v. Walters*, 21 F.3d 411, 415 (Fed. Cir. 1994).

6. As used herein, "product" refers to any system, apparatus, method, device, software, encoder, decoder, codec, and/or process using Image Coding Technology.

7. "Image Coding Technology" means any technology for compressing and/or decompressing data representing moving images, facsimile images and/or still images, including but not limited to the coding methods described in the JPEG Standard and MPEG Standards.

8. "Standard" or "Standards" or "Recommendation" refers to any documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and/or services are fit for their purpose.

9. "ITU-T" means the International Telecommunication Union Telecommunication Standardization Sector and all predecessors, including the International Telephone and Telegraph Consultative Committee ("CCITT").

10. "ISO" means the International Organization for Standardization.

11. "IEC" means the International Electrotechnical Commission.

12. The phrase "JPEG Standard" refers to the Standard Recommendation ISO/IEC 10918-1|ITU-T Rec. T.81, entitled "Information Technology — Digital Compression and Coding of Continuous-Tone Still Images — Requirements and Guidelines."

13. The phrase "MPEG Standards" refer collectively to one or more of the ISO/IEC 11172-2 Standard for "MPEG-1," the ISO/IEC 13818-2 Standard for "MPEG-2," and the ISO/IEC 14496-2 Standard for "MPEG-4."

14. "Chen patent" means United States Patent No. 4,698,672 to Wen-hsiung Chen and Daniel J. Klenke, entitled "Coding System for Reducing Redundancy," filed October 27, 1986 and issued October 6, 1987, attached hereto as Exhibit A.

15. "Chen article" means "Scene Adaptive Coder" by Wen-hsiung Chen, published in the IEEE Transactions on Communications, Vol. Com-32, No. 3 (March 1984), attached hereto as Exhibit B.

16. "Widcom VTC-56" and "VTC-56" mean the coder-decoder, or "codec," that was developed by Widcom and was claimed to be capable of compressing a video image and transmitting it over 56 kilobits-per-second lines or links, and includes all of its prototypes as well as its previous, predecessor, and subsequent models.

17. "Widcom's Personal Videoconferencing Station" and "PVS" mean the videoconferencing system developed by Widcom and referred to by those names, and includes all of its prototypes as well as its previous, predecessor, and subsequent models.

18. "Forgent Litigations" includes but is not limited to, Civil Action Case Numbers M: C 05-01654 JF (RS) and C 05-01567 JF (RS) for litigation that occurred in the United States District Court for the Northern District of California, San Jose Division.

INSTRUCTIONS

19. This subpoena calls for you to produce all documents identified below that are within your possession, custody, or control, or are otherwise available to you.

20. Electronics records and computerized information that are produced must be readable with standard commercial software or must be accompanied by a description of the system from which it was derived sufficient to render such materials readable.

21. If, after exercising due diligence to secure the information requested, you cannot fully comply with a specific Request, or any part thereof, please state the reason(s) for your inability to reply and respond to the fullest extent possible.

DOCUMENTS REQUESTED- GENERAL

1. All documents authored by Forgent, including but not limited to Compression Labs, Inc. and/or Widcom, Inc., or by any of its employees, including but not limited to Wen-hsiung Chen, Daniel J. Klenke and/or William K. Pratt, concerning Image Coding Technology prior to September 11, 1987.

2. Documents sufficient to demonstrate the conception and/or invention dates for the Chen Patent.

3. Documents sufficient to demonstrate the actual and constructive dates of reduction to practice for the Chen Patent, including inventor's notebooks and prosecution histories, further including documents sufficient to demonstrate diligence in reduction to practice from the dates of conception and/or invention to the dates of actual and/or constructive reduction to practice.

4. Documents sufficient to fully describe the methods for image encoding and decoding used by any Image Coding Technology for: (1) any projects or products that lead to the inventions of the Chen article and/or Chen patent; (2) any projects or products that were based on the inventions of the Chen article and/or Chen patent prior to September 11, 1987; (3) any other products made by or at the direction of Forgent that used Image Coding Technology prior to

September 11, 1987; and (4) any other projects of Forgent that used Image Coding Technology prior to September 11, 1987.

5. All documents alleged to be prior art to the Chen patent and produced during the Forgent Litigation or any other litigation involving the Chen patent, including software, systems, and documents, further including but not limited to all versions of Compression Labs, Inc.'s Rembrandt video conferencing system and Widcom Inc.'s Widcom VTC-56 video conferencing system.

6. All documents concerning the prior art described in Document Request Number 5, including analyses, expert reports, and claim charts.

7. Documents sufficient to describe any Forgent projects or products disclosed in documents produced in response to Document Request Numbers 4-6, including documents sufficient to describe the project and/or product names and model numbers.

8. Documents sufficient to demonstrate the public use, knowledge, availability, offer for sale and/or sale of any project or product disclosed in documents produced in response to Document Request Numbers 4-6 and which occurred prior to September 11, 1987, including documents sufficient to demonstrate such use, knowledge, availability, offer for sale and/or sale that occurred prior to September 11, 1987.

9. All documents concerning patent licenses from Forgent relating to the Chen patent, regardless of whether a license was actually executed, including patent licenses for the Chen Patent and the settlement agreement for the Forgent Litigation.

10. All documents concerning, referring or relating to Forgent's litigation or pre-litigation investigations involving CLI or Widcom, and those company's respective patents claimed to be related to the JPEG Standard.

11. All documents concerning any device, apparatus or product made by or otherwise associated with Widcom, including but not limited to the Widcom VTC-56 and/or the Widcom Personal Videoconferencing Station, and including but not limited to documents concerning the research, development, conception, reduction to practice, use, sale, offer for sale, distribution, display, and marketing thereof by any person or entity.

12. All documents concerning any video conferencing codec, video compressor or video compression system or technology developed by Widcom for other companies, including without limitation Bell & Howell and Defense Advanced Research Projects Agency ("DARPA"), including but not limited to documents concerning the research, development, conception, reduction to practice, use, sale, offer for sale, distribution, display, and marketing thereof by any person or entity.

13. All documents concerning papers, articles, publications, public statements, reviews, announcements, and news reports by anyone regarding any product associated with Widcom, including but not limited to the Widcom VTC-56, the Widcom Personal Videoconferencing Station, any video compressor developed by Widcom for Bell & Howell, or any video conferencing codec developed by Widcom for DARPA, including but not limited to all drafts and final versions of any such materials and any memoranda or correspondence between co-authors or others concerning either the work reflected in the materials or the publications themselves.

DOCUMENTS REQUESTED- STANDARDS SETTING

14. All documents relating to any disclosure by Forgent of Image Coding Technology to the ISO and/or CCITT Standards Bodies concerning the JPEG Standard prior to February 15,

1994, including but not limited to any proposal, submission, description, contribution and/or technical specification.

15. All documents relating to any participation by Forgent, or any employee of Forgent, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994, with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification.

16. All Communications between Forgent and any of its employees and the ISO and/or CCITT Standards Bodies and any sub-group of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee relating to the JPEG Standard and/or MPEG Standards.

17. All documents relating to any participation by Philips, or any employee of Philips, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994, with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including (but not limited to) any communication, proposal, submission, contribution, and/or technical specification, and including participation by current or former Philips employees, including, but not limited to, Loeck Zeckendorf, Gerald Weth, Ulf Rothgordt, Peter Vogel, Karl Heinz Wenzel, Dr. Cord Brandis, Jan van der Meer, Norman

Richards, John Morris, Simon Turner, Hendrick Eisema, Louis Van Loon, Erik Schylander, P. De Wit, Burkhard Krupa and Joachim Spiedel.

DEPOSITION TOPICS

1. The authenticity of any documents produced by Forgent in response to this subpoena.

2. The search for and collection of any documents relevant or responsive to this subpoena.

3. The conception and invention date for the Chen patent.

4. The actual and constructive dates of reduction to practice for the Chen patent, including demonstrating diligence in reduction to practice from the dates of conception and/or invention to the dates of actual and/or constructive reduction to practice.

5. The methods for image encoding and decoding used by any Image Coding Technology for: (1) any projects or products that led to the inventions of the Chen patent and/or Chen article; (2) any projects or products that were based on the inventions of the Chen patent and/or the Chen article prior to September 11, 1987; (3) any other products made by or at the direction of Forgent that used Image Coding Technology prior to September 11, 1987; and (4) any other projects of Forgent that used Image Coding Technology prior to September 11, 1987.

6. Any Forgent projects or products disclosed in documents produced in response to Document Request Number 4, including project and/or product names and model numbers.

7. The public use, knowledge, availability, offer for sale and/or sale of any project or product disclosed in documents produced in response to Document Request Numbers 4 and 8 and which occurred prior to September 11, 1987, including such use, knowledge, availability, offer for sale and/or sale that occurred prior to September 11, 1987.

8. Patent licenses or attempts by Forgent to license the Chen patent, including discussions concerning licensing of the Chen patent and the settlement agreement for the Forgent Litigation.

9. Documents authored by Forgent or by any of its employees concerning Image Coding Technology prior to September 11, 1987.

10. Forgent's purchase or knowledge of any Widcom product, including without limitation the Widcom VTC-56 and the Widcom Personal Videoconferencing Station.

11. Litigation involving CLI and Widcom.

DEPOSITION TOPICS- STANDARDS RELATED

12. Participation by Forgent, or any employee of Forgent, in any ISO and/or CCITT Standards Bodies prior to February 15, 1994 with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification, including those documents currently or formerly in the possession of the employees referenced in Document Request Number 15 above.

13. Communications between Forgent and any of its employees and the ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies relating to the JPEG Standard and/or MPEG Standards.

14. Participation by Philips, or any employee of Philips, including those employees listed in Document Request 17 above, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994 with respect to Image Coding

Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification.

15. Communications between Philips and any of its employees and the ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies relating to the JPEG Standard and/or MPEG Standards, including those employees listed in Document Request Number 17 above.

EXHIBIT A

United States Patent [19]

Chen et al.

[11] **Patent Number: 4,698,672**

[45] **Date of Patent: Oct. 6, 1987**

[54] **CODING SYSTEM FOR REDUCING REDUNDANCY**

[75] Inventors: **Wen-hsiung Chen**, Sunnyvale; **Daniel J. Klenke**, Milpitas, both of Calif.

[73] Assignee: **Compression Labs, Inc.**, San Jose, Calif.

[21] Appl. No.: **923,630**

[22] Filed: **Oct. 27, 1986**

[51] Int. Cl.[4] H04N 7/133; H04N 7/137

[52] U.S. Cl. 358/136; 358/261; 358/262; 375/27

[58] Field of Search 358/136, 135, 133, 261, 358/262; 375/27, 31, 33

[56] **References Cited**

U.S. PATENT DOCUMENTS

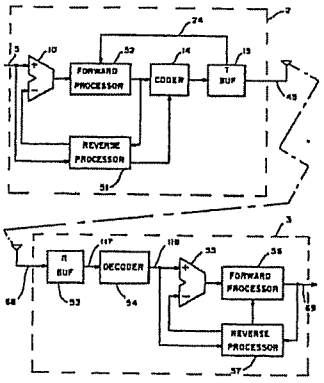
4,302,775	11/1981	Widergren	358/136
4,476,495	10/1984	Fujisawa	358/262
4,520,490	5/1985	Wei	375/27
4,558,370	12/1985	Mitchell	358/262
4,633,325	12/1986	Usubuchi	358/133

Primary Examiner—Howard W. Britton
Attorney, Agent, or Firm—Fliesler, Dubb, Meyer & Lovejoy

[57] **ABSTRACT**

The present invention relates to methods and apparatus for processing signals to remove redundant information thereby making the signals more suitable for transfer through a limited-bandwidth medium. The present invention specifically relates to methods and apparatus useful in video compression systems. Typically, the system determines differences between the current input signals and the previous input signals using mean-square difference signals. These mean-square signals are processed and compared with one or more thresholds for determining one of several modes of operation. After processing in some mode, the processed signals are in the form of digital numbers and these digital numbers are coded, using ordered redundancy coding, and transmitted to a receiver.

46 Claims, 4 Drawing Figures



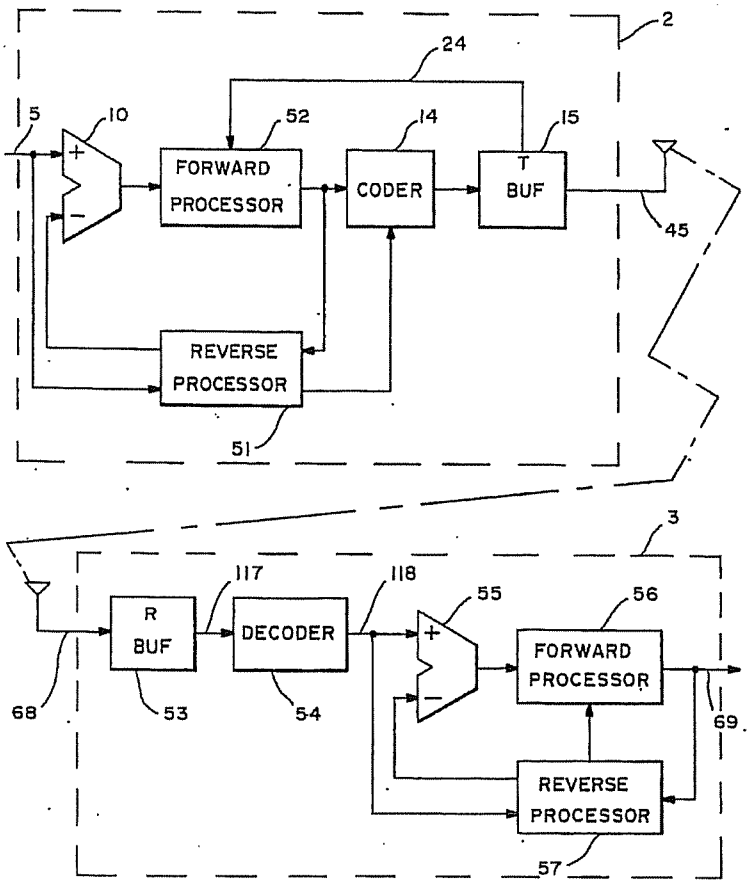

2
24
5
10
52
14
15
FORWARD PROCESSOR
CODER
T BUF
45
REVERSE PROCESSOR
51
3
117
118
55
56
R BUF
DECODER
FORWARD PROCESSOR
68
53
54
69
REVERSE PROCESSOR
57

FIG.—1

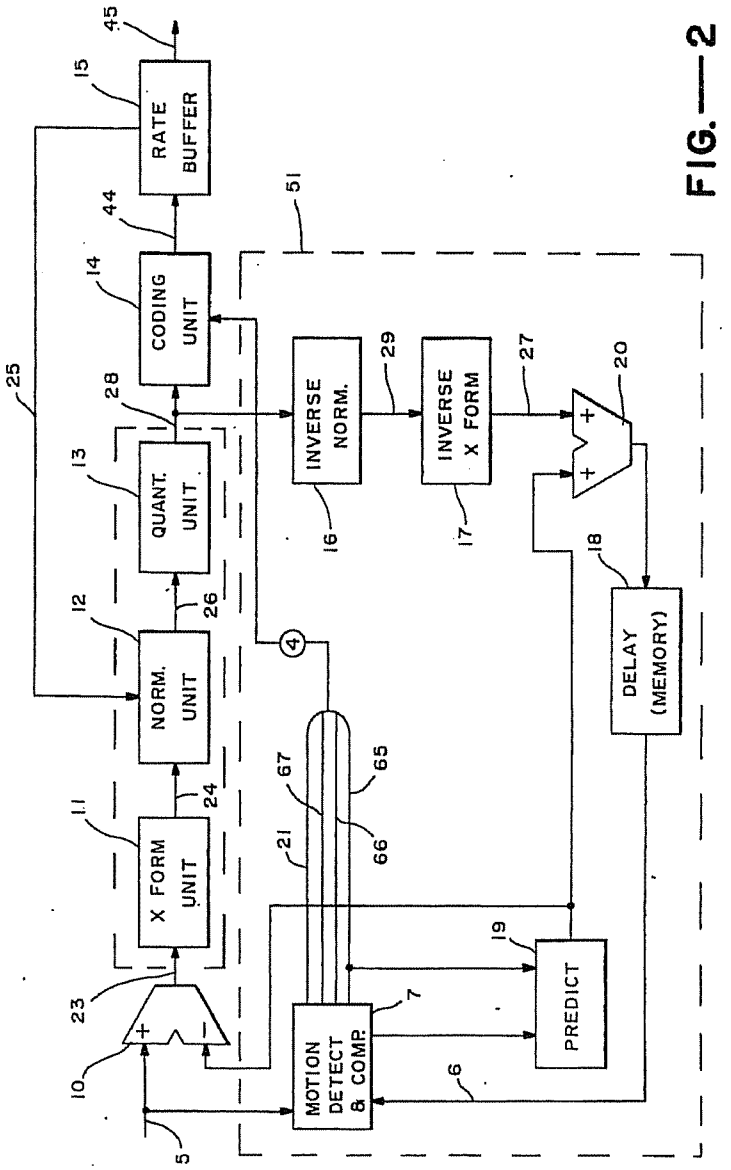

RATE BUFFER
CODING UNIT
QUANT. UNIT
NORM. UNIT
X FORM UNIT
INVERSE NORM.
INVERSE X FORM
DELAY (MEMORY)
MOTION DETECT & COMP.
PREDICT
45
15
44
14
13
12
11
51
25
28
29
27
20
16
17
18
26
24
67
65
66
21
19
7
23
10
6
5
4

FIG.—2

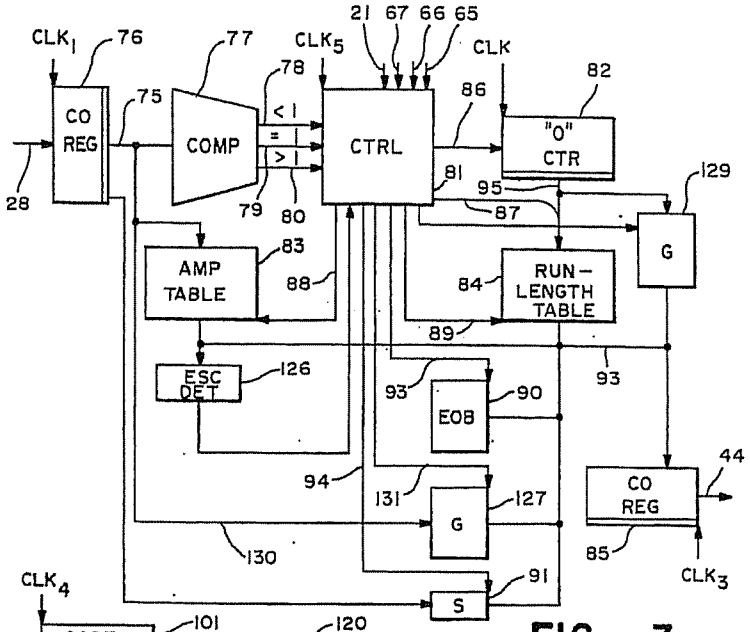

CLK1
76
77
CLK5
21 67 66 65
CLK
82
78
86
CO REG
75
COMP
< I
= I
> I
CTRL
"0" CTR
28
79
80
81
95
87
129
G
AMP TABLE
83
88
84
RUN-LENGTH TABLE
89
93
ESC DET
126
93
EOB
90
94
131
G
127
CO REG
44
130
85
CLK3
91
S
CLK4
CODE REG
101
120
111
117
DET
102
I RUNLENGTH TABLE
103
117
I AMP TABLE
104
G
116
105
119
115
CLK6
118
125
RL CTR
CO REG
CLK5
118
114
RIR'
112
CTRL
119
G
108
109
110
107

FIG.—3

FIG.—4

CODING SYSTEM FOR REDUCING REDUNDANCY

CROSS-REFERENCE TO RELATED APPLICATION

Title: A COMBINED INTRAFRAME AND INTERFRAME TRANSFORM CODING SYSTEM
Ser. No.: 479,766 Filed: 83/03/28 (now abandoned)
Inventors: Wen-hsiung Chen, James Parker Elliott, Robert Edwin George Newell, Ralph Emerson Nichols, Albert Edwards Rackett

BACKGROUND OF THE INVENTION

The present invention relates to methods and apparatus for processing signals to remove redundant information thereby making the signals more suitable for transfer through a limited-bandwidth medium. The present invention specifically relates to methods and apparatus useful in video compression systems.

Many signal processing techniques useful in video compression systems are known. For example, digital encoding is often employed in processing television signals which are to be transferred over transmission channels since digital data streams are more immune to noise degradation.

In order to digitally encode a television signal, a signficant number of bits, 4 or more, may be required to provide for an acceptable range of gray scale for each of the hundreds of thousands of separate picture elements (pixels) which form an image. Consequently, data rates for unprocessed digitalized television signals typically require a bandwidth greater than 40 megabits per second. If the communications link is an earth satellite, an unprocessed video signal typically occupies nearly the entire bandwidth of the satellite, with very few channels, if any, left over for other uses. A T1 communication channel is typical and has only a 1.5 megabit per second bandwidth. A practical yet effective way to reduce the bandwidth of digitalized television signals is needed so that fewer channels are required for transmission over a communications path and so that the quality of transmitted signals is maintained even when reduced bandwidth transmission is employed.

U.S. Pat. No. 4,302,775, assigned to the same assignee as the present invention, describes a scene adaptive coding technique which eliminates redundant information and thereby reduces the bandwidth.

The patent describes a single-pass digital video compression system which implements a two-dimensional cosine transform with intraframe block-to-block comparisons of transform coefficients without need for preliminary statistical matching or preprocessing.

Each frame of the video image is divided into a predetermined matrix of spatial subframes or blocks. The system performs a spatial domain to transform domain transformation of the picture elements of each block to provide transform coefficients for each block. The system adaptively normalizes the transform coefficients so that the system generates data at a rate determined adaptively as a function of the fullness of a transmitter buffer. The transform coefficient data thus produced is encoded in accordance with amplitude Huffman codes and zero-coefficient runlength Huffman codes which are stored asynchronously in the transmitter buffer. The encoded data is output from the buffer at a synchronous rate for transmission through a limited-bandwidth medium. The system determines the buffer fullness and adaptively controls the rate at which data is generated so that the buffer is never completely emptied and never completely filled.

In the system receiver, the transmitted data is stored in a receiver buffer at the synchronous data rate of the limited-bandwidth medium. The data is then output from the receiver buffer asynchronously and is decoded in accordance with an inverse of the encoding in the transmitter. The decoded data is inversely normalized and inversely transformed to provide a representation of the original video image.

The U.S. Pat. No. 4,302,775 patents reduces redundancy by employing intraframe coding techniques utilizing intraframe comparisons of cosine transform coefficients. While the patent provides significant improvement over other techniques, there is a need for even greater compression.

In addition to intraframe coding techniques, interframe coding techniques have been used to reduce the rate required for video transmission as described, for example, in the above-identified application. Typically, each video frame is held in memory at both the transmitter and the receiver and only frame-to-frame changes are transmitted over the communication link. In contrast to intraframe coding schemes in which the quality of coded images is dependent upon the amount of detail in each single image frame, the quality of the coded image in interframe coding is dependent upon the differences from frame to frame. Frame-to-frame differences are often referred to as "motion".

Interframe coding techniques are broadly classified into two categories, namely, spatial domain coding and transform domain coding. In real-time interframe spatial-domain coding systems, spatial domain data can be threshold processed to obtain and store frame difference signals in a transmitter buffer. The threshold value can be adaptively determined as a function of the transmitter buffer fullness. In order to eliminate the image breakdown, both spatial and temporal subsampling has been proposed.

The above-identified U.S. patent application entitled "A Combined Intraframe and Interframe Transform Coding System" employs intraframe and interframe variable prediction transform coding. Images are represented by sequential frames of two-dimensional arrays of digital signals. The digital signals are transformed to form transform coefficients for each frame. Predicted transform coefficients are formed using sets of variable prediction factors. The predicted transform coefficients for each frame are compared with corresponding actual transform coefficients for the frame to form transform coefficient difference signals. The difference signals are processed to control their range of values. The processed difference signals are statistically coded such that the more frequently occurring values are represented by shorter code lengths and the less frequently occurring values are represented by longer code lengths. The coded signals are stored in a buffer memory for transmission. The coded signals in the buffer memory are transmitted, over a limited-bandwidth medium, to the receiver along with processing information. The processing information includes codes identifying the set of variable prediction factors utilized in the transmitter. The same set of variable prediction factors is utilized in the receiver to reconstruct predicted transform coefficients which in turn are used to reconstruct representations of the original images in the transmitter.

The extension of the Scene Adaptive Coding of U.S. Pat. No. 4,302,775 from intraframe coding to interframe coding has proven very significant in terms of improving image quality and reducing bandwidth. These improvements, however, have created a need for improved coding systems for reducing redundancy and there continues to be a need for improved signal processing methods and apparatus for data compression systems.

SUMMARY OF THE INVENTION

The present invention is a signal processor and method for efficiently processing signals using ordered redundancy (OR) coding and any one of a number of different modes.

The signals to be coded are typically multiple values where the multivalued digital numbers, X(k) are typically the integers 0, 1, 2, 3, 4, . . . , and so on arranged in any order. Frequently, some values are repeated in forming digital numbers and hence the probable frequency of occurrence of some values is different than for other values. In one example of digital numbers, the highest frequency of occurrence is the value 0, the next highest frequency of occurrence is the value 1 and the other values greater than 1 (namely 2, 3, 4, 5, and so on) occur least frequently. With such order to the frequency of occurrence of values to be coded, the ordered redundancy coding of the present invention is most efficient.

Using ordered redundancy coding, the system codes the highest most frequently occurring values (0's in the usual example) using runlength coding. In the most preferable example, the runlength encoding is of two types, R and R'. The first type, R, is utilized when a runlength of consecutive zeros (0's) is followed by the next most frequently occurring value (1 in the usual case) and the other type, R', is utilized when the runlength of consecutive zeros (0's) is followed by some other value, one of the least frequently occurring values (usually greater than 1 such as 2, 3, and so on). Whenever the second type, R', of runlength coding is employed, the runlength code is typically followed by an amplitude code which explicitly encodes the actual amplitude (2, 3, . . .) of the following other value. Whenever the first type, R, of runlength coding is employed, no coding of the second value (usually 1) is required because an amplitude of 1 is implied simply by the use of the first type, R, of runlength coding.

The ordered redundancy coding of the present invention is typically utilized in a system that processes input signals, such as spatial domain image signals occurring in successive frames, to form processed signals for each frame. Any number of different processing modes are possible. The processed signals are in the form of a plurality of multivalued digital numbers, X(k), typically one number, X(k), for each frame.

In one particular embodiment, the processing modes include two replenishment modes (one with motion compensation and one without), two DPCM modes (one with motion compensation and one without) and one intraframe mode. The decision as to which mode to select is made based upon an analysis of the frame-to-frame differences (motion) between the current input signals and the previous input signals.

Typically, the system determines differences between the current input signals and the previous input signals using mean-square difference signals. These mean-square signals are processed and compared with one or more thresholds for determining one of several modes of operation. After processing in some mode, the processed signals are in the form of digital numbers and these digital numbers are coded, using ordered redundancy coding, and transmitted to a receiver.

After transmission of the coded signals, the received signals are decoded and processed in reverse of the particular one of the modes by which the signals were processed in the transmitter.

In accordance with the above summary, the present invention achieves the objective of providing an improved signal processor for reducing redundancy using ordered redundancy coding.

The foregoing and other objects, features and advantages of the invention will be apparent from the following detailed description in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 depicts a block diagram of a transmitter and receiver system.

FIG. 2 depicts further details of the transmitter of the FIG. 1 system.

FIG. 3 depicts further details of the coder used in the FIG. 2 transmitter.

FIG. 4 depicts further details of the decoder used in the receiver.

DETAILED DESCRIPTION

Overall System—FIG. 1

In FIG. 1, a block diagram of a transmitter and a receiver in accordance with the present invention is shown. Digital signals to be processed are input on lines 5 to the transmitter 2. The input signals on lines 5 are processed in one of a number of different modes so as to efficiently compress the data input signals to form processed signals for transmission to a receiver. The processed signals are coded and output on lines 45 from the transmitter 2 and are transmitted to the receiver 3.

The transmitter 2 includes a forward processor 52 and a feedback (reverse) processor 51. Typically, the input signals on lines 5 represent images and are presented in the space domain as frames in accordance with well known techniques. The forward processor 52 typically processes the spatial domain input signals to form processed signals which typically are transform domain signals arranged in blocks of transform domain coefficients. The forward processor 52 processes the current input signals from the most current frame.

The reverse processor 51 typically inverse processes signals from transform domain to spatial domain. Processor 51 stores signals representing the previous frame of data and also receives the current input signals so as to enable a comparison to be made between the previous inverse processed input signals and the current input signals. When the current input signals have been transformed from the spatial domain to the transform domain, the reverse processor performs an inverse transform to convert the transform domain signals back to spatial domain signals and stores those spatial domain signals for comparison with the current input spatial domain signals.

The reverse processor 51 determines changes between the current signals and the previous signals. Typically, these differences are determined using mean-square signals, d_0 and d_b, hereinafter defined. These mean-square signals are processed and compared with

one or more thresholds for determining one of several modes of operation for the system of FIG. 1.

Any number of different modes are possible. In one particular embodiment, two replenishment modes (one with motion compensation and one without), two DPCM modes (one with motion compensation and one without) and one intraframe mode are employed. The decision as to which mode to select is made based upon an analysis of the frame-to-frame differences (motion) of the input data. After the processing by the processor 52 and the processor 51, the processed signals are input to the coder 14.

The coder 14 encodes the processed signals using statistical frequency coding. With satistical frequency coding, signals with a statistically higher frequency of occurrence are encoded with a shorter code length than signals with a statistically lower frequency of occurrence. Additionally, the coder 14 utilizes a novel ordered redundancy (OR) coding technique. In the ordered redundancy coding, the processed signals to be coded have multiple values. For example, values are typically 0, 1, 2, 3, 4, . . . , and so on. Typically, the statistical frequencies of the values to be coded have an order. Particularly, that order is based upon the probable frequency of occurrence of the different values. The highest frequency of occurrence is typically the value 0, the next most frequently occurring value is 1 and the other values greater than 1 (2, 3, 4, 5, and so on) occur least frequently. With such order to the signals to be coded, the ordered redundancy coding of the present invention is most efficient.

Using OR coding, the coder 14 of FIG. 1 codes the highest most frequently occurring values (0's in the usual example) using runlength coding. In the most preferable example, the runlength encoding is of two types, R and R'. The first type, R, is utilized when the runlength of 0's is followed by the next most frequently occurring value (1 in the usual case) and the other type, R', is employed when the runlength of 0's is followed by some other value of the least frequently occurring type (usually greater than 1 such as 2, 3, and so on). Whenever the R' type of runlength coding is employed, the runlength code is typically followed by an amplitude code which explicitly encodes the actual amplitude of the other value. Whenever the first type, R, of runlength coding is employed, no coding of the second value (usually 1) is required because an amplitude of 1 is implied simply by the use of the first type, R, of runlength coding.

After the ordered redundancy coding in coder 14, data is transferred to the transmitter buffer 15. The buffer 15 provides a feedback signal on line 25 to control the forward processor 52 data rate.

In FIG. 1, the data from line 45 is input on line 68 after transmission over some conventional transmission medium to the receiver 3. In the receiver 3, a receiver buffer 53 stores the received data. A decoder 54 decodes the received data. Thereafter, the decoded data is processed in reverse of the particular one of the modes by which the data was processed in the transmitter 2. The reconstituted data appears on output line 69.

Transmitter—FIG. 2

FIG. 2 is a block diagram of a transmitter for motion compensated combined interframe and intraframe coding system of FIG. 1. Motion compensation is incorporated into a combined interframe and intraframe coding system using the spatial pixels in the inverse loop 9. In operation, the original spatial image on input lines 5 is compared to the reconstructed spatial image on lines 6 of the previous frame on a block-by-block basis through a motion detector 7. The reconstructed spatial image is obtained from the memory 18 of the feedback DPCM loop 9.

The feedback loop 9 includes the inverse normalizer 16, inverse transformer 17, the sum unit 20, the delay (memory) 18, the prediction unit 19, and the motion detector and compensator 7. If the motion detector 7 determines that there is little difference between the blocks, a "replenishment mode" is selected. On the other hand, if enough difference is detected, the block in the current frame is compared to the neighborhood of the corresponding reconstructed block in the previous frame to find the best match of the block. For the purpose of increasing the system performance a sub-pixel match is employed. If the difference between the current block and its best matched block does not result in a reasonable improvement over the difference between the current block and its original counterpart, a motion compensation is not justified. In this case, a "DPCM mode" with variable predictions is selected to handle the block difference. On the other hand, if the difference between the current block and its best matched block is reasonably smaller than the difference between the current block and its original counterpart, a motion compensation is initiated. In this case, the difference between the current block and its best matched block is screened to determine if the block belongs to a "motion compensated replenishment" block or a "motion compensated DPCM" block. The forward loop of the DPCM system encodes the "DPCM" or "motion compensated DPCM" data in the transform domain. Statistical frequency coding is employed to improve the efficiency. The feedback loop of the DPCM system is operated in the spatial domain with variable predictions.

Motion Detection and Compensation

The motion detection serves two purposes. It compares the block pixels in the present frame to the neighborhood pixels of the corresponding block in the previous frame to find the sub-pixel displacement of the block that gives the best match. It also tracks the displacement vectors and the degree of differences during the matching process for a subsequent modification of the DPCM frame memory and controlling of the predictor parameters in the feedback DPCM loop. Three basic types of modes (replenishment modes, intraframe mode, and DPCM modes) are determined from the motion detection. A decision process among the modes is employed. The decision process relies in part on a determination as to whether motion-compensation or non-motion-compensation is to be employed. Motion compensation is determined using the mean-square difference, d_0, and the mean square error, d_b.

The mean-square difference, d_0, is formed as follows:

$$d_0 = (1/N^2) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} [f(j,k) - \bar{f}(j,k)]^2 \qquad \text{Eq. (1)}$$

where $f(j,k)$ are spatial pixels (on lines 5 of FIG. 2) of the current frame and $\bar{f}(j,k)$ are the corresponding pixels (on line 6 of FIG. 2) of the reconstructed previous frame. N is the transform block size.

The mean-square error, d_b, is formed as follows:

$$d_b = (1/N^2)\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}[f(j,k) - \tilde{f}(j+\Delta j, k+\Delta k)]^2 \quad \text{Eq. (2)}$$

where f(j,k) are the block pixels in the present frame and f(j+Δj,k+Δk) are the best matched pixels in the previously reconstructed frame where Δj,Δk are the displacement (vector) for the best match.

Replenishment Modes

The replenishment modes are either motion-compensated or non-motion-compensated. The decision process selects compensation or non-compensation based upon motion detection. The motion detection unit 7 of FIG. 2 determines the difference between the incoming spatial pixels of a block and the reconstructed spatial pixels of the corresponding block in the previous frame. If the motion detection process determines that there is little frame-to-frame difference between corresponding blocks, a non-motion-compensated replenishment mode is selected and a code word is sent on line 21 from unit 7 of FIG. 2 to the encoder 14 to identify the mode.

If the motion detection process determines that the frame-to-frame block difference is great enough then, under some circumstances, a motion-compensated replenishment mode is selected. The detection process typically uses the mean-square difference, d_0, and compares it to a predetermined non-motion-compensated replenishment threshold, T_R. This process is written as follows:

if $(d_0-d_b)<T_M$ and $d_0<T_R$, select non-motion-compensated replenishment mode.

The detection process compares the mean square error, d_b, with a predetermined motion-compensated replenishment threshold, $T_{D/R}$, as follows:

if $(d_0-d_b)>T_M$ and $d_b<T_{D/R}$, select motion-compensated replenishment mode.

The identification code words for the replenishment modes are typically Huffman coded. Typically, a one-bit code (0), on line 21 of FIG. 2 is used if the non-motion-compensated replenishment mode appears most frequently statistically. Once this code word is identified at the receiver, the reconstructed block pixels in the previous frame are repeated to form the present block in the receiver.

For the motion compensated replenishment block, typically a four-bit code (1111) is used, along with the displacement vector representing the best match, and appears on line 67 in FIG. 2. At the receiver, the vector uses the compensated block pixels from the reconstructed previous frame to form the presently reconstructed block.

DPCM Modes

The DPCM modes are either non-motion-compensated or motion-compensated. Selection of the compensation or non-compensation DPCM modes is dependant in part on motion detection. The motion detection searches for the best matched block pixels from the reconstructed previous frame. The difference, d_b, between the present block pixels and the best matched block pixels is then computed. If this difference is smaller than the motion threshold, T_M, no motion compensation is justified due to the necessity of sending the displacement vector as coding overhead. In this case, the difference, d_0, is compared to a DPCM threshold, $T_{D/I}$, to determine if the block belongs to a DPCM mode. The decision process is given as follows:

if $(d_0-d_b)<T_M$ and $d_0<T_{D/I}$, select non-motion-compensated DPCM mode.

If a non-motion-compensated DPCM mode is selected, the predictor in the feedback loop is enabled and the difference is sent to the discrete cosine transformer for subsequent encoding. Again, the mode identification is Huffman coded. Typically, a two-bit code (10) used for the non-motion-compensated DPCM mode and appears on line 66 in FIG. 2.

At the receiver, the DPCM data are inversely transformed and added onto the block pixels from the reconstructed previous frame to form the present block pixels.

For the motion-compensated DPCM mode, the difference, d_b, between the current block pixels and the best matched block pixels is compared to a predetermined motion-compensated replenishment threshold, $T_{D/R}$. If d_b is larger than the threshold, a motion-compensated DPCM mode is selected to handle the pixel differences.

The decision processs is given as follows:

If $(d_0-d_b)>T_M$ and $d_b>T_{D/R}$, select motion-compensated DPCM mode.

For the motion compensated DPCM blocks, typically a three-bit code (110) is used together with the displacement vector representing the best match of the block along with the motion compensated DPCM data (transform coefficient differences between the present block and the best matched block from the reconstructed previous frame). The mode ID and vector appear on line 65 in FIG. 2. At the receiver, these DPCM data are inverse transformed and added onto the compensated block pixels from the reconstructed previous frame to form the present block pixels.

Intraframe Mode

The intraframe mode is selected when neither the motion-compensated mode nor the DPCM mode is justified. The difference, d_0, between the current block pixels and the reconstructed previous block pixels is compared with the predetermined DPCM threshold, $T_{D/I}$. The decision process is as follows:

If $(d_0-d_b)<T_M$ and $d_0>T_{D/I}$, select intraframe mode.

If the intraframe mode is selected, the predictor is disabled and the current block pixels are sent to the transformer with unit 11 of FIG. 2. Typically, a four-bit code (1110) appearing on line 66 in FIG. 2 is used to identify the "intraframe mode". The intraframe data in the receiver are inversely transformed to form the present block pixels.

Compensation Range and Resolution

The performance of the motion compensated system is dependent upon the range and resolution of the matching process. The larger the range and the finer the resolution, the better the system performs. However, due to the necessity of encoding the vector information as system overhead, the range and resolution of the searching process is somewhat limited.

Searching Algorithm

The search for the best matched position is a very time consuming process. As one example, a simple binary search algorithm for a maximum range of 1.75 can be employed. Using such an algorithm, the nine whole-pixel positions centered around the position of the present block are first examined to find the best match. Next, the eight half-pixel neighborhood positions centered around the best matched whole-pixel position are examined. The process continues until the best matched quarter-pixel position is located. The horizontal and

vertical addresses of this location are then recorded as a vector and encoded accordingly. The number of steps required for a binary search is many times lower than that of a brute force search.

Subpixel translation is done by performing bilinear interpolation taking weighted averages of the four nearest values at integral pixel positions surrounding the subpixel location. The weighting factors that are used are linear functions of the horizontal and vertical distance of the fractional displacement from the integral pixel positions. As an example, a displacement of 1.25 horizontally, and 0.75 vertically is performed as follows:

$$f(j+1.25,k-0.75)=w_1f(j+1,k)+w_2f(j+1,k-1)+w_3 f(j+2,k)+w_4f(j+2,k-1) \quad \text{Eq. (3)}$$

where $w_1=(0.75)\ (0.25)$, $w_2=(0.75)\ (0.75)$, $w_3=(0.25)\ (0.25)$, and $w_4=(0.25)\ (0.75)$

DPCM Loop

Referring to FIG. 2, the Differential Pulse Code Modulated (DPCM) loop consists of a cosine transform unit **11**, a normalization unit **12**, a quantization unit **13**, an inverse normalization unit **16**, an inverse transform unit **17**, a delay memory **18**, and a prediction unit **19**. In operation, an input pixel block on lines **5** from the present frame is first subtracted in subtractor **10** by its estimation from the previous frame on line **23** on a pixel-by-pixel basis to generate block differences. These differences are then cosine transformed in transform unit **11** to form the coefficient differences on lines **24**. The coefficient differences are next scaled in normalizer unit **12** according to a feedback parameter on lines **25** from the output rate buffer **15**. The scaled coefficient difference on lines **26** are then quantized in unit **13** and fed into both the coder unit **14** and the inverse DPCM loop **9**. In the inverse DPCM loop **9**, the quantized and scaled data are inversely normalized in unit **16** and inversely transformed in unit **17**, to form the quantized coefficient differences on lines **27**. These differences are then added in adder **20** to the motion compensated estimation on lines **3** to form the reconstructed pixel block in the frame memory **18**. After a single-frame delay, in memory **18**, the motion detector **7** uses the motion compensated block from the memory **18**, multiplies it by a prediction weighting factor, and is ready for the next frame of operation. At the receiver, the received data follows the inverse DPCM loop to reconstruct the spatial pixels in the output block.

Cosine Transform

The coefficient differences between the input pixels from the present frame on lines **5** and the estimations from the previously reconstructed frame on lines **3** are formed by the difference circuit **10** on lines **23** and are expressed as follows:

$$e_n(j,k)=f_n(j,k)-p(j,k)\bar{f}_{N-1}(j+\Delta j,k+\Delta k) \quad \text{Eq. (4)}$$

where Δj and Δk represent the vector values for the best match determined by the motion detector and where $p(j,k)$ represents the estimation. These differences within a N×N block are cosine transformed in transformer **11** to form the coefficient differences on lines **24**. The cosine transform is defined as follows:

$$E_n(u,v)=4[C(u)C(v)]/N^2 \sum_{j=0}^{N-1}\sum_{k=0}^{N-1} e_n(j,k) \quad \text{Eq. (5)}$$

-continued

$\cos[(2j+1)u\pi/2N]\cos[(2k+1)v\pi/2N]$

for $u,v=0,1,\ldots,N-1$

$C(w)=1/(2^{\frac{1}{2}})$ for $w=0$

$=1$ for $w=1,2,\ldots,N-1$

where w=u or v

where (j,k) and (u,v) represent indices in the horizontal and vertical directions for the pixel difference and coefficient difference blocks, respectively, and where C(w) represents C(u) or C(v). The cosine transform restructures the spatial domain data into the coefficient domain such that it will be beneficial to the subsequent coding and redundancy removal processes.

Normalization

The coefficient differences, $E_n(u,v)$, are scaled according to a feedback normalization factor, D, on lines **25**, from the output rate buffer **15** according to the relation,

$$I_n(u,v)=E_n(u,v)/D \quad \text{Eq. (6)}$$

The scaling process adjusts the range of the coefficient differences such that a desired number of code bits can be used during the coding process.

Quantization

The quantization process in unit **13** is any conventional linear or non-linear quantization. The quantization process will set some of the differences to zeros and leave a limited number of significant other differences to be coded. The quantized coefficient differences on lines **28** are represented as follows:

$$\bar{I}_n(u,v)=Q[I_n(u,v)] \quad \text{Eq. (7)}$$

where Q[ ] is a quantization function.

It should be noted that a lower bound is determined for the normalization factor in order to introduce meaningful coefficient differences to the coder. Generally speaking, setting the minimum value of D to one is sufficient for a low rate compression applications involving transform blocks of 16 by 16 pixels. In this case the worst mean square quantization error is less than 0.083. This mean square error corresponds to a peak signal-to-quantization-noise ratio of 40.86 db which is relatively insignificant for low rate applications.

Inverse Normalization

The process of inverse normalization in unit **16** produces the quantized coefficient differences on lines **29** in the inverse DPCM loop **9**. This process is represented as follows:

$$\bar{E}_n(u,v)=\bar{I}_n(u,v)D \quad \text{Eq. (8)}$$

Inverse Cosine Transform

The inverse cosine transform process in unit **17** in the inverse DPCM loop **9** converts the quantized coefficient differences on lines **29** back to the spatial domain pixel differences on lines **27**. This process is defined as follows:

$$\bar{e}_n(j,k)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)\bar{E}_n(u,v) \quad \text{Eq. (9)}$$

$\cos[(2j+1)u\pi]/2N\ \ \cos[(2k+1)v\pi]/2N$

for $j,k,=0,1,\ldots,N-1$.

Frame Memory

The frame memory 18 contains the reconstructed input pixels in the inverse DPCM loop. The quantized pixel differences from the inverse cosine transformer on lines 27 and the motion compensated estimations from the previously reconstructed frame on lines 3 are added together in adder 20 to form the reconstructed pixels, $\hat{f}_n(j,k)$, which replace the block pixels in the memory 18. This process is represented as follows:

$$\hat{f}_n(j,k)=\bar{e}_n(j,k)+\rho(j,k)\hat{f}_{n-1}(j+\Delta j,k+\Delta k) \quad \text{Eq. (10)}$$

Prediction

The prediction process in unit 19 finds an estimation of a datum from its surrounding data. By way of example for a simple predictor that uses the previous frame as a base for the estimation, the estimated value is termed as the correlation coefficient, $\rho(j,k)$, given as follows:

$$\rho(j,k)=E[e_n(j,k)e_{n-1}(j+\Delta j,k+\Delta k)]/\sigma^2(j,k) \quad \text{Eq. (11)}$$

where E represents expected value and $\sigma^2(j,k)$ represents the variance of $e_n(j,k)$. The correlation coefficient, termed as leak factor, ranges from 0 to 1 depending on the frame-to-frame pixel differences. The value is very close to 1 for a limited motion sequence. However, during a scene cut or a rapid zooming sequence, the value is way below the value of 1. Because different leak factors have to be identified in the encoding of the DPCM process, it represents a significant overhead for the low rate system if too many values are to be identified. In one embodiment, only two leak factor values are used for the five-mode motion detection system: 1 for the non-motion-compensated DPCM and motion compensated DPCM modes and 0 for the intraframe mode.

Coding

In order to minimize overhead code bits, in one typical example the encoding process in unit 14 for the FIG. 2 system is performed on a frame by frame bases. The coded bit stream includes sync, header, scaling factor (NF), and variable-length data as follows:

TABLE 1

			variable
SYNC	HEADER	NF	DATA

In the header, at least one bit is reserved for the identification of full motion and graphic operations. The data portion includes the block-to-block mode identifiers, the vector values, DPCM and intraframe data. The bit allocations are dependent upon each individual block which is illustrated in TABLE 2.

TABLE 2

1			
MODE			
Replenishment Block			

4	8		
MODE	VECTOR		
Replenishment of Motion Compensated Block			

2	variable	
MODE	DPCM	EOB
DPCM of Non-motion Compensated Block		

4	variable	
MODE	INTRAFRAME	EOB
Intraframe of Non-motion Compensated Block		

TABLE 2-continued

3	8	variable	
MODE	VECTOR	DPCM	EOB
DPCM of Motion Compensated Block			

DPCM Encoding

The Scene Adaptive Coding (SAC) is very efficient in terms of coding the intraframe transform coefficients. When this scheme is applied to a coding system involving intraframe, interframe and motion compensation, the coding efficiency is somewhat reduced due to the structure of coefficient differences or motion compensated coefficient differences caused by the additional removal of redundancies. One observation that can be made in the motion compensated coefficient differences (non-zero after normalization and quantization) and, to a certain degree, the interframe coefficient differences (non-zero differences) is that most of these differences are sparsely distributed with an overwhelming majority of them having an absolute value of one. Also, within these differences of ones, a significant portion of them are isolated (surrounded by zero-valued coefficients) along the path of a scanning. It is wasteful to use one amplitude code word to code each of these isolated ones in addition to using one runlength code word to identify their address (Runlength alone should be enough).

Ordered Redundancy Coding

A new Ordered Redundancy (OR) coding algorithm is specifically designed to code multi-valued digital numbers where the statistical frequency of occurrence of some values in the series of values forming the digital number is greater than the statistical frequency of occurrence for other values in the series of values forming the digital number. The values forming the digital numbers are generally the integers 0, 1, 2, 3, . . . and so on.

In general, a K-valued digital number, X(k), is formed by a series of K values, x(k), as follows:

$$X(k)=x(1), x(2), x(3), \ldots x(k), \ldots, x(K)$$

where $1 \leqq k \leqq K$. Each value, x(k), has some value, V_j, from the set of J values,

$$V_1, V_2, V_3, \ldots, V_j, \ldots, V_J$$

where $1 \leqq j \leqq J$.

The occurrence of i consecutive values, V_j, within the series X(k) is the runlength of such values denoted by V_j^i.

In a first example with k=1, . . . , 14, if the digital number $X_1(k)=01000000100021$, $V_0=0$, $V_1=1$ and $V_2=2$ then $X_1(k)=V_0^1, V_1^1, V_0^6, V_1^1, V_0^3, V_2^1, V_1^1$. In the series values forming $X_1(k)$, the first value $V_0=0$ occurs most frequently, the second value $V_1=1$ occurs next most frequently, and the other value $V_2=2$ occurs least frequently.

In a second example with k=1, . . . , 14, if the digital number $X_2(k)=02111110001130$, and $V_0=1$, $V_1=0$, $V_2=2$, and $V_3=3$; then $X_2(k)=V_1^1, V_2^1, V_0^5, V_1^3, V_0^2, V_3^1, V_1^1$. In the series of values forming $X_2(k)$, the first value, $V_0=1$, occurs most frequently, the second value $V_1=0$ occurs next most frequently, and the other values, $V_2=2$ and $V_3=3$, occur next most frequently.

Digital numbers formed with such frequencies of occurrence of values such as for $X_1(k)$ and $X_2(k)$ above, are defined as having ordered redundancy. In the typi-

cal example described for $X_1(k)$, 0's are most redundant, 1's are next most redundant, and so on. The frequency of occurrence order of values 0, 1, 2, . . . and so on described is merely one typical example. Any frequency of occurrence order is possible, for example, the 2's may occur more frequently than 1's and 0's may occur more frequently than 2's.

Digital numbers, X(k), will often have ordered redundancy of the values, V_j, forming the number. Ordered redundancy means that the frequency of occurrence of some of the values, V_j, formng the number (or groups of such values) is greater than that for other values (or other groups of such values) forming the number and that such frequencies of occurrence are predictable for a number of digital numbers, X(k).

When such ordered redundancy occurs, the ordered redundancy coding of the present invention is useful in making the coding more efficient. In the present invention, the presence of a first value (or a first set of values) is used to imply the existence of a second value (or a second set of values) thereby eliminating the need to code the second value (or second set of values).

By way of example, the coding of the digital number $X_1(k)$ above is achieved as follows. Assume that when the first value, V_0, is followed by the second value, V_1, that the second value is implied and such code is denoted $C_{01}{}^i$ where i represents the number of consecutive first values V_0 preceding the implied second value, V_1. Assume that when the first value V_0, is not followed by the second value, V_1, such code is denoted $C_{0\bar{1}}{}^i$. Assume that any other value is amplitude coded with $A_2=2$ and $A_3=3$. With such a notation, $X_1(k)=C_{01}{}^1$, $C_{01}{}^6$, $C_{0\bar{1}}{}^3$, A_3, $C_{01}{}^0$.

By way of the second example, $X_2(k)$ above, the first value, V_0-1 implies the second value, $V_1=0$ such that $X_2(k)=C_{01}{}^0$, $C_{0\bar{1}}{}^3$, A_2, $C_{01}{}^5$, $C_{01}{}^0$, $C_{01}{}^0$, $C_{01}{}^0$, $C_{0\bar{1}}{}^2$, A_3, $C_{10}{}^0$.

In order to code $X_1(k)=C_{01}{}^1$, $C_{01}{}^6$, $C_{0\bar{1}}{}^3$, A_3, $C_{01}{}^0$, each of the values $C_{01}{}^1$, $C_{01}{}^6$ and so forth are represented by a unique statistical code (typically a binary code) from a runlength table such that the statistically more frequently occurring values have shorter code lengths and the statistically less frequently occurring values have longer code lengths.

A series of values in digital numbers having a large percentage of zeros (0's) followed by ones (1's) is termed "One's Redundancy". One's Redundancy Coding is one example of Ordered Redundancy (OR) coding. The OR coding procedures for One's Redundancy appear in TABLE 3 and are based upon 16×16 transform blocks of values where each such block gives rise to a digital number, X(k), having 256 values. Of course, any size blocks (N×M) of digital values can be selected. Also, the digital values can be in block form representing transform coefficients or can be multi-valued digital signals, X(k), of any form.

In order to identify the beginning or end of the values forming a number, X(k), a special "End of Block" signal, EOB, is utilized. When a plurality of numbers $X_1(k)$, $X_2(k)$, $X_3(k)$, . . . and so on are to be coded and transmitted, the EOB code is inserted between the numbers, usually once after each number.

The TABLE 3 example is premised upon digital signals having first values $V_1=0$, second values $V_2=1$, and a set of other values, $V_{\bar{2}}$, greater than 1 (2, 3, 4, . . .). Also, TABLE 3 has a runlength table partitioned into first and second parts, a first part, R (or C_{01}), and a second part, R' (or $C_{0\bar{1}}$). The first part, R, implies that a runlength of 0's is followed by a 1. The second part, R', implies that a runlength of 0's is followed by another value greater than 1 (2, 3, 4, . . .). The TABLE 3 formulation is for one preferred embodiment of the ordered redundancy coding. Many variations, some hereinafter described, are possible.

TABLE 3

1. From the magnitude (without sign) of quantized coefficient difference, form the following sets of histograms
   a. Runlength of consecutive zero-value coefficient differences (including runlength of zero length) with absolute amplitude value of one at the end of the runlength.
   b. Runlength of consecutive zero-valued coefficient differences (including runlength of zero length) with absolute amplitude value of greater than one at the end of the runlength.
   c. Occurrence of end of blocks (EOB, all 0's)
2. Get runlength Huffman code table from the histogram of 1 above. The entries of this table can be represented as R_0, R_1, R_2, . . . , R_{255}, R'_0, R'_1, R'_2, . . . , R'_{255}, EOB.
3. From case b of 1, get the histogram of the amplitudes (with values greater than one) at the end of the runlength.
4. Get amplitude Huffman code table from the histogram of 3 above. The entries of this table can be represented as A_2, A_3, A_4, . . . , A_{510}.
5. Encode the coefficient differences along the zig-zag path from the Huffman tables generated from 2 and 4 in the following fashion.
   a. Coefficient differences of one at the end of the consecutive zeros—encode with R+SIGN, n=1, 2, 3, . . . , 255.
   b. Coefficient differences of greater than one at the end of consecutive zeros—encode with $R'+A_m+$-SIGN, n=1, 2, 3, . . . , 255 and m=2, 3, 4, . . . , 510.
6. Encode with EOB at the end of each block.

As can be seen from TABLE 3, two Huffman tables or equivalent statistical coding tables are specified in the "One's Redundancy" (OR) coding. The runlength table (including EOB) consists of two parts, R and R', with a total of 513 entries (256 each for the first part R and the second part R' and 1 for EOB). The amplitude table consists of 509 entries (amplitude values of 2 to 510). In a practical implementation, these two tables can be shortened with little performance degradation.

Specific examples of the two tables specified in accordance with TABLE 3 appear as the following TABLES 6 and 7. TABLE 6 is a runlength table of the two part example (R and R' or R_1 and R_2) where R implies a runlength of 0's followed by a 1. TABLES 6 and 7 are derived based upon the hardware constraints (which are intended to be representative of a practical system, but are not intended to be limiting) of the following TABLE 4:

TABLE 4

1. Every code word must belong to part of a complete "tree".
2. The longest code word (including runlength escape, runlength code and sign, or amplitude escape and amplitude code) must not exceed 16 bits in length.
3. The maximum number of entries for each runlength or amplitude table must not exceed 32.

TABLE 5 gives four comparative examples for coding digital numbers using Scene Adaptive Coding (SAC) and One's Redundancy (OR) coding. The One's Rendundancy coding examples utilize TABLES 6 and 7 and the Scene Adaptive Coding examples utilize TABLES 8 and 9. As can be seen from TABLE 5, the OR coding is considerably shorter than the SAC coding and hence OR coding is more efficient.

TABLE 5

COMPARISON OF "OR" AND "SAC" CODING

1.	CO	0000000000000000001 EOB
	SAC	$RLP+R_{19}+A_1+S+EOB$ 01/1110111/11/0/100001
	OR	$R_{19}+S+EOB$ 0001000/0/0010
2.	CO	001-1000001000-1 EOB
	SAC	$RLP+R_2+A_1+S+A_1+\bar{S}+RLP+R_5+A_1+S+RLP+R_3+A_1+\bar{S}+EOB$ 01/1111/11/0/11/1/01/11010/11/0/01/1011/11/1/100001
	OR	$R_2+S+R_0+\bar{S}+R_5+S+R_3+\bar{S}+EOB$ 1110/0/10/1/00011/0/0000/1/0010
3.	CO	20000000-1 EOB
	SAC	$A_2+S+RLP+R_7+A_1+\bar{S}+EOB$ 101/0/01/110011/11/1/100001
	OR	$R_0'+A_2+S+R_7+\bar{S}+EOB$ 110/1/0/011110/1/0010
4.	CO	1001-200001 EOB
	SAC	$A_1+S+R_2+A_1+S+A_2+\bar{S}+R_4+A_1+S+EOB$ 11/0/1111/11/0/101/1/11100/11/0/10001
	OR	$R_0+S+R_2+S+R_0'+A_2+\bar{S}+R_4+S+EOB$ 10/0/1110/0/11011/01101/0/0010

where,

R=runlength, A=amplitude, S=positive sign, $\bar{S}$=negative sign, RLP=Run Length Prefix (01), EOP=End Of Block, CO=digital number to be coded

TABLE 6

RUN LENGTH CODE TABLE FOR THE "ONE'S REDUNDANCY" CODING

RUN LENGTH CODES FOR DPCM MODE

T	L	FREQ	# of BITS	CODE	OCTAL EQUIV
R	0	26644.	2	10	2
R'	0	15621.	3	110	6
R	1	12324.	3	010	2
R	2	7148.	4	1110	16
R	3	4610.	4	0000	0
R	4	3384.	5	01101	15
R'	1	3143.	5	01100	14
R	5	2577.	5	00011	3
R	6	1967.	6	111100	74
R	7	1764.	6	011110	36
R	8	1452.	6	001111	17
R	9	1327.	6	001101	15
R	10	1089.	6	000101	5
R'	2	1013.	7	1111011	173
R	11	994.	7	1111010	172
R	12	884.	7	0111011	73
R	13	876.	7	0111010	72
R	14	861.	7	0011100	71
R	15	687.	7	0011100	34
R	16	673.	7	0011001	31
R	17	602.	7	0011000	30
R	18	550.	7	0001001	11
R	19	496.	7	0001000	10
R	20	485.	8	01111101	175
R	21	455.	8	01111100	174
R	22	413.	8	01110001	161
R'	3	402.	8	01110000	160
R	23	370.	8	00111011	73
R	24	345.	8	00111010	72
R	ESC	4599.	5	11111	37
R'	ESC	982.	7	0111111	77

TABLE 6-continued

RUN LENGTH CODE TABLE FOR THE "ONE'S REDUNDANCY" CODING

RUN LENGTH CODES FOR DPCM MODE

T	L	FREQ	# of BITS	CODE	OCTAL EQUIV
EOB		5047.	4	0010	2

where,

R ESC=code used whenever R-type value not in table

R' ESC=code used when R'-type value not in table.

TABLE 7

AMPLITUDE CODE TABLE FOR THE "ONE'S REDUNDANCY" CODING

AMPLITUDE CODES FOR DPCM MODE

	A	FREQ	# of BITS	CODE	OCTAL EQUIV
A	2	11076.	1	1	1
A	3	3846.	2	00	0
A	4	1751.	4	0110	6
A	5	982.	5	01111	17
A	6	663.	5	01010	12
A	7	435.	6	01100	34
A	8	347.	6	010011	23
A	9	277.	6	010001	21
A	10	173.	7	0101100	54
A	11	178.	7	0101101	55
A	12	137.	7	0100100	44
A	13	113.	8	01110101	165
A	14	116.	8	01110110	166
A	15	79.	8	01001010	112
A	16	68.	8	01000011	103
A	17	67.	8	01000010	102
A	18	58.	9	011101110	356
A	19	49.	9	011101000	350
A	20	50.	9	011101001	351
A	21	30.	10	0111011111	737
A	22	32.	9	010000010	202
A	23	33.	9	010000011	203
A	24	20.	10	0100101100	454
A	25	31.	9	010000001	201
A	26	22.	10	0100101101	455
A	27	30.	9	010000000	200
A	28	23.	10	0100101110	456
A	29	14.	11	01001011111	1137
A	30	14.	11	01110111100	1674
A	31	10.	11	01001011110	1136
A	32	14.	11	01110111101	1675
A	ESC	423.	6	010111	27

where,

ESC=code used when amplitude value not in table.

TABLE 8

RUN-LENGTH CODES FOR "SCENE ADAPTIVE CODING"

VALUE	LENGTH	HUFFMAN CODE
1	1	0
2	4	1111
3	4	1011
4	5	11100
5	5	11010
6	5	10000
7	6	110011
8	6	110010
9	6	110001
10	6	110000
11	6	101011
12	6	101001
13	6	101000
14	6	100111
15	6	100110
16	6	100101
17	6	100100
18	6	100010

TABLE 8-continued

RUN-LENGTH CODES FOR "SCENE ADAPTIVE CODING"

VALUE	LENGTH	HUFFMAN CODE
19	7	1110111
20	7	1110110
21	7	1101111
22	7	1101110
23	7	1101101
24	7	1101100
25	7	1010101
26	7	1000111
27	7	1000110
28	8	10101000
29	9	101010011
30	9	101010010
RL-ESC	6	111010

TABLE 9

AMPLITUDE CODES FOR SCENE ADAPTIVE CODING

VALUE	LENGTH	HUFFMAN CODE
1	2	11
2	3	101
3	3	000
4	4	0011
5	5	10001
6	5	00100
7	6	100101
8	6	100000
9	7	1001110
10	7	1001100
11	7	0010111
12	8	10011111
13	8	10011011
14	8	10010011
15	8	10010001
16	8	00101101
17	9	100111101
18	9	100110101
19	9	100110100
20	9	100100100
21	9	100100000
22	9	001011001
23	9	001011000
24	10	1001111001
25	10	1001111000
26	10	1001001011
27	10	1001001010
28	10	1001000011
29	10	1001000010
AMP-ESC	6	001010
EOB	6	100001
RL-PREFIX	2	01

Ordered Redundancy Variations

Additional variations are possible, for example, three or more parts or their equivalent may be used in the runlength table. A typical example with three parts (R, R' and R") is as follows. Runlengths of consecutive first values (V_1=0) are runlength encoded with three different parts (R_1, R_2, or R_3) depending upon the value following the runlength of 0's. If the following value is a second value (such as V_2=1), then R_1 is selected for encoding the runlength of the first value (0's in this case). If the following value is a third value (such as V_3=2), then R_2 is selected for encoding the runlength of the first value (0's in this case). If the following value is another value (greater than 2 such as 3, 4, 5, . . .), then R_3 is selected for encoding the runlength of the first value (0's in this case). If R_3 is selected, then R_3 is followed by an amplitude code to specify the exact value (3, 4, 5, . . .) following the runlength of first values (0's).

The runlength table utilized with ordered redundancy coding can be of two parts (R and R'), three parts (R_1, R_2, and R_3), or more generally of "n" parts (R_1, R_2, . . . , R_n), where n is equal to or greater than 2.

The TABLES 6 and 7 were formed based upon the assumption that a separate sign bit, S or $\bar{S}$, not in the tables is to be used to indicate the sign of each value coded in the manner indicated in TABLE 5. Alternatively, the sign information can be encoded into TABLE 6 or TABLE 7. For example, a table like TABLE 6 can be used to represent runlengths of 0's that are followed both by positive and by negative non-zero numbers. Such a table would be greater in length than TABLE 6 (expanded essentially to double the length) to provide entries for runlengths of 0's followed by both negative and positive non-zero numbers. Of course, such a table would be ordered in accordance with the statistical frequency of both positive and negative numbers.

The two tables, TABLES 6 and 7, were formed based upon the assumption that the values to be coded were categorized into three basic groups or values, namely a first value, V_1, a second value, V_2, and all other values. In the particular example of coding, the first value V_1 is 0, the second value V_2 is 1, and the third value is one within the set of all values greater than 1. It often occurs that in a block of values to be coded, the value 0 (the first value) occurs statistically most frequently, the value 1 (the second value) occurs statistically second most frequently, and the other values (the third values) the least frequently.

With such a distribution having ordered redundancy, the coding of the second value (1's in this case) is avoided because the first value (0's in this case) is runlength coded in two parts, one part that implies that the number following the runlength of 0's is the second value (1 in this case) and the other part that indicates that the number following the runlength of 0's is within the set of third values (values greater than 1 in this case).

Alternative formulations are possible. For example, rather than categorizing the values to be coded into three groups as done in connection with TABLE 6, four or more groups are possible. For four groups, the first value (for example V_1=0) is coded in three parts, namely, a first part for implying a second value (for example V_2=1), a second part for implying a third value (for example V_3=2) and a third part for indicating a set of fourth values (values greater than 2).

In general, a multivalued digital number, X(k), to be coded with n-1 implied values has a first value, V_1, a second value, V_2, . . . , a j-value, V_j, a (J+1)-value, V_{j+1}, . . . , a n-value, V_n, for j ranging from 1 to n, and has other values. The digital signals are coded with n-1 implied values to form statistically coded signals such that the more frequently occurring values of the digital signals are represented by shorter code lengths and the less frequently occurring values of coded signals are represented by longer code lengths. The coding includes, for each value, V_j, for j from 1 to n, forming j^{th} runlength code values representing the number of consecutive first values followed by the j+1 value, forming additional runlength code values representing the number of consecutive first values followed by any of said other values.

While the embodiments described have used one code (such as R) based upon the existence of a runlength of a first value to imply a second value, the implied code is not limited to a single value but can be itself multivalued. For example, a runlength of 0's followed by two 1's can be implied by a code R".

While the implied coding of the second value was typically as a result of runlength coding the first value, other types of coding of the first value are included within the present invention.

As another alternative, the statistically most frequent value is not necessarily the value that is runlength encoded. Where three groups of values are employed (such as 0's, 1's and greater than 1's), the second value (1's in this case) can be runlength encoded to imply the first value (0's in this case) or to specify the third values (numbers greater than 1 in this case).

In an example where the number of values V_j, are limited, the need for amplitude coding can be eliminated. For example, if only the values $V_1=0$ and $V_2=1$ are present in the number X(k), then no amplitude coding is required since the $V_1=0$ values can be runlength coded and the values of $V_1=1$ can be implied. Similarly, for an example with only the values $V_1=0$, $V_2=1$, and $V_3=2$, the values of $V_1=0$ can be runlength coded while both $V_2=1$, and $V_3=2$, are implied using a two-part runlength table as previously described.

In an example where all of the values have the same sign, the sign coding can be eliminated.

Coder Details—FIG. 3

In FIG. 3, further details of the coder 14 of FIG. 2 are shown. In FIG. 3, each digital value, V_j, of a digital number, X(k), to be coded is input to the CO register 76. Typically, the register 76 is a 16-bit register for storing 16-bit values where the digital number, X(k), is formed of K 16-bit values, each value clocked into register 76 in sequence and one at a time. The comparator 77 compares the absolute value of each value in register 76 to determine if that absolute value is less than 1, equal to 1, or greater than 1. Comparator 78 provides a less-than-1 output signal on line 78, an equal-to-1 signal on line 79, and a greater-than-1 signal on line 80 as a function of the value in register 76. The less-than-1 signal on line 78 indicates an equal-to-0 condition. The control 81 receives the three control values on line 78, 79 and 80 from comparator 77 and controls, in a conventional manner, the coder operations.

The "zero" counter 82 counts the runlength of consecutive zeros detected by the comparator 77. Line 86 from control 81 causes counter 82 to be set to a counting mode for counting consecutive 0 values in register 76. Line 86 causes counter 82 to be reset after each runlength of zeros is counted. After being reset and with line 86 setting counter 82 to the counting mode, counter 82 will count zeros until a non-zero value is detected in register 76. If a non-zero value is detected, either a equal-to-1 signal on line 79 or a greater-than-1 signal on line 80 is enabled and detected by control 81. If an equal-to-1 signal is detected, control 81 asserts the line 87 to specify the R type of operation. The enable line 87 together with the runlength count from counter 82 addresses the runlength table 84. Runlength table 84 is typically a random access memory or a read only memory storing coded runlength values like those of TABLE 6. The 0 runlength output on line 95 from counter 82 together with the 1-bit on line 87 address the table 84 to provide a runlength coded value output on lines 93. The output from table 84 is under control of the signal on line 89 from control 81 and loads the code register 85 with the runlength coded value from the CODE column of TABLE 6. The runlength coded value implies that a runlength of zeros is followed by a 1 in the manner previously described.

After a coded value is loaded into register 85, the sign bit from register 76 is enabled to be stored in register 85 by the enable gate 91 under control of the signal 94 from the control 81.

Thereafter, the next value, V_j, of the number, X(k), is loaded into register 76. Counter 82 is cleared and a new runlength of zeros is counted until comparator 77 detects a non-zero value by asserting either an equal-to-1 signal on line 79 or a signal on line 80 signifying a greater-than-1 value in register 76. If the runlength of zeros is followed by a value greater than 1, then line 80 is asserted and control 81 causes line 87 to be not asserted, thereby signifying an R'-type of operation. The runlength value from counter 82 on line 95 together with the non-asserted signal on line 87 causes the runlength table 84 to be addressed to obtain a R' value from table 84. Line 89 causes the output from table 84 to be gated to the code register 85.

Because of a greater than 1 value in register 76, control 81 causes the line 88 to be next enabled to provide an output from the amplitude table 83. The amplitude table 83 is a random access memory or read only memroy loaded with amplitude values like those of TABLE 7. The value in register 76 addresses the amplitude table 83 to provide the appropriate amplitude value output on line 93 for storage in the code register 85. Thereafter, the control 81 causes line 94 to be enabled to cause the sign value from register 76 to be stored in the code register 85.

The FIG. 3 coder continues to process code values in register 76 until an entire block of code values (all values for a digital number, X(k)) has been processed. Control 81 includes counters and other appropriate means for counting or otherwise determining all values comprising a digital number. When a full series of values for a digital number X(k) has been processed, control 81 enables the output line 93 to provide an end of block, EOB, signal on line 93 for storage in the control register 85. Control 81 provides the CLK_1 signal for clocking each new value into register 76, provides the CLK_2 signal for incrementing the zero counter 83 and CLK_3 signal for clocking values into register 44. In a conventional manner, control 81 is controlled by a master clock signal CLK, from the transmitter of FIG. 2.

In FIG. 3, when the amplitude table 83 is addressed and produces the ESC code, the ESC detector 126 senses that no amplitude value is available in the table and signals control 81. The ESC value from table 83 is gated into the code register 85. Thereafter, control 81 enables gate 127 via line 181 to gate the value from register 76 into the code register 85. Alternatively, an additional table (not shown) can be provided for storing Huffman coded values of amplitudes not in the table 83. Such an additional Huffman table would provide compression of additional amplitude values.

In FIG. 3, when the runlength table 84 provides the R ESC or the R' ESC code value, the ESC detector 126 senses the ESC value and signals the control 81 on line 130. The ESC code value is clocked into register 85, and on the next cycle, control 81 causes alternate processing to occur. In the example described, gage 129 is enabled to enter directly the value from counter 82 into the code register 85 so that runlengths not in the runlength table 84 are directly entered after the ESC code. Alternatively, an additional runlength table with Huffman coded runlength values can be employed to provide additional compressed runlengths not in the table 84.

While FIG. 3 depicts one embodiment for implementing the coder 14 of FIG. 2, many other software and hardware implementations of the coder are, of course, possible.

Decoder Detail—FIG. 4

In FIG. 4, further details of the decoder 54 of FIG. 1 are shown. The serial-by-bit data is input on line 117 to the code register 101. The input data, as it is clocked into the register 101 by the CLK_4 signal, is continuously detected by the detector 102. Detector 102 senses the synchronization, header and other control information and signals the control 107 when coded data is to follow. The coded data is clocked into register 101 one bit at a time. A code value clocked into register 101 is presented in left-to-right order when viewing the CODE column of TABLE 6. With each new code value bit, the coded data from register 101 is input to the inverse runlength table 103 and to the inverse amplitude table 104. The runlength table 103 includes the data of TABLE 6 organized in an inverse order. The inverse order means that table 103 of FIG. 4 is addressed by the CODE column code values and provides as an output the type (R or R') from column T and the length from column L. The type information appears on output line 113. Line 113 is one binary value (for example 1) when the addressed value is of type R and is another binary value (for example 0) when the type is R'.

The R/R' information on line 113 is connected to the control 107. The L information from table 103 is input on line 119 to the runlength counter 105. Typically, the L information is a binary count and runlength counter 105 is parallel loaded with the binary count under control of line 114 from control 107.

If an R ESC or an R' ESC value is detected by detector 102, control 107 is signaled that no valid runlength will be derived from table 103. When control 107 senses that the ESC code has appeared in register 101, control 107 causes the content of the register 101 through gate 125 to be gated into the runlength counter 105. Thereafter, runlength counter 105 is decremented in the manner previously described.

Line 116 output from the table 103 is a validity bit indicating that a valid entry has been found in table 103. As each new code value bit is clocked into register 101, table 103 is addressed to determine if a valid entry is found. Not all input codes from register 101 will find a valid entry in table 103. All valid entries in table 103 provide a validity bit output on line 116 for signalling the control 107. When control 107 receives a valid bit from line 116, the length value for the addressed entry is stored into the runlength counter 105. Thereafter, the runlength counter 105 is decremented by the CLK_5 signal thereby counting out the runlength of zeros. Control line 118 inhibits any output from the amplitude table 104 whenever counter 105 is being decremented thereby loading zeros into the CO register 109. When the counter 105 has been counted down and the entire runlength of zeros has been loaded into register 109, control 107 has sensed the R or R' signal from line 113 and thereafter provides the following sequencing.

If line 113 indicated an R-type operation, then line 121 loads a 1 into the register 109 since R-type operations imply a 1 after a runlength of zeros. When line 121 writes a 1 into register 109, gate 108 is enabled by line 119 to load the sign bit, which will be the next bit in order clocked into register 101 into the register 109. Thereafter regiser 101 will be cleared and clocked to receive the next code bits.

If line 113 indicates an R'-type operation, then line 121 is not enabled and line 118 is enabled to read out an amplitude from amplitude table 104. Amplitude table 104 contains the information of TABLE 7 in inverse order. The inverse order indicates that table 104 is addressed by the information in the CODE column and provides an output on line 120 from the A column. Typically, the output value from the A column is a binary number representing the amplitude.

If an ESC value is called for, detector 102 signals control 107 to indicate that no valid amplitude will be obtained from table 104. When the A ESC code appears in the code register 101, the control 107 causes the next amplitude value in code register 101 to be gated directly via gate 108 to the CO register 109.

After an amplitude value is loaded into register 109 from table 104 or register 101, control 107 then signals via line 119 the loading of the sign bit from register 101 into register 109. Register 101 is then cleared to receive the next code value on line 117 from the buffer 53 of FIG. 1.

While FIG. 4 depicts one embodiment of a decoder in accordance with the present invention, many other software and hardware embodiments of the FIG. 5 decoder are possible.

Rate Buffer

The rate buffer 15 in FIG. 2 performs channel rate equalization. The buffer has a variable rate data input on lines 44 and a constant rate data output on lines 44. The differentials are monitored from frame to frame, and the status is converted into a scaling factor that is fed to the normalizer on lines 25. The buffer always forces the coder to adjust to the local coding variations, while ensuring global performance at a desired level.

Let B(n) represent the number of bits into the rate buffer for the nth frame and let S(n) represent the buffer status (difference between the read and write pointers of the FIFO) at the end of the nth frame. Then, B(n) and S(n) can be written as follows:

$$B(n) = K + N_1 + 12N_2 + 2N_3 + \sum_{i \in N_3} \sum H([\bar{I}(u,v)]_i) + 11N_4 + \sum_{i \in N_4} \sum H([\bar{I}(u,v)]_i) + 4N_5 + \sum_{i \in N_5} \sum H([\bar{I}(u,v)]_i) \quad \text{Eq. (12)}$$

$$S(n) = S(n-1) + [B(n) - N^2R] \quad \text{Eq. (13)}$$

where

N_1=number of blocks in replenishment mode

N_2=number of blocks in motion compensated replenishment mode

N_3=number of blocks in DPCM mode

N_4=number of blocks in motion compensated DPCM mode

N_5=number of blocks in intraframe mode

$[\bar{I}(u,v)]_i$=normalized and quantized coefficient differences in ith block

H(.)=“One's Redundancy” coding function R=average coding rate

N=transform block size

K=sync, header, and NF

$i \in N_3$=i belongs to N_3 DPCM block

$i_\epsilon N_4 = i$ belongs to N_4 DPCM block
$i_\epsilon N_5 = i$ belongs to N_5 DPCM block

The buffer status S(n) is used to select an instantaneous scaling factor $D^*(n)$ according to an empirically determined "scaling factor versus status" curve. This relationship is described by

$$D^*(n) = \Phi\{S(n)\} \quad \text{Eq. (14)}$$

In order to smooth out this instantaneous scaling factor such that the desired scaling factor does not fluctuate too much, a recursive filtering process is applied as follows:

$$D(n) = (1-c)D(n-1) + cD^*(n) \quad \text{Eq. (15)}$$

where c is a constant with value less than unity. The rate buffer can be guaranteed not to overflow by introducing a frame repetition mechanism. It can also be prevented from underflow by introducing fill bits.

Frame Repetition

The requirement of a frame repetition in the Motion Compensated Combined Interframe and Intraframe Coding System of FIG. 2 is well justified. Due to the usage of only one normalization factor per frame, an excessive amount of data can flow into the buffer during a scene cut or fast zooming operations. Only instantaneous shutting off of the input data like the frame repetition will prevent the rate buffer from overflowing. Also, in order to prevent the scaling factor from getting too large to introduce blocking artifacts, a frame repetition is desired.

To establish frame repetition in the rate buffer, a threshold in the rate buffer is first established. During the encoding process, if the data within the buffer exceeds this threshold at the end of the frame, frame repetition is initiated to stop the input data. The repetition process is stopped when the data within the buffer is reduced to a level lower than the threshold.

While the invention has been particularly shown and described with, reference to preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A method for processing digital signals, where the digital signals have first values, second values and other values, to reduce the amount of data utilized to represent the digital signals and to form statistically coded signals such that the more frequently occurring values of digital signals are represented by shorter code lengths and the less frequently occurring values of digital signals are represented by longer code lengths, comprising,

forming first runlength code values representing the number of consecutive first values of said digital signals followed by said second value,

forming second runlength code values representing the number of consecutive first values of said digital signals followed by one of said other values.

2. The method of claim 1 further including the step of amplitude encoding said other values.

3. The method of claim 1 further including the step of encoding said first and second runlength code values with a sign value.

4. The method of claim 1 wherein said first values have amplitude zero, said second values have absolute amplitude one, and said other values have absolute amplitudes greater than one whereby said first and second runlength codes values are formed representing the number of consecutive zeros.

5. The method of claim 1 wherein said first values have the highest frequency of occurrence in said digital signals, wherein said second values have the next highest frequency of occurrence in said digital signals, and wherein said other values have the lowest frequency of occurrence in said digital signals.

6. A method for processing input signals to reduce the amount of data utilized to represent the input signals, the steps comprising,

processing the input signals to form processed signals where the processed signals are digital numbers having first values, second values, and other values,

coding each digital number to form statistically coded signals such that the more frequently occurring values in the digital numbers are represented by shorter code lengths and the less frequently occurring values of coded signals are represented by longer code lengths, said coding including,

forming first runlength code values representing the number of consecutive first values followed by said second value in a digital number,

forming second runlength code values representing the number of consecutive first values followed by one of said other values in the digital number.

7. The method of claim 6 wherein said coding step includes the step of amplitude encoding said other values.

8. The method of claim 6 wherein said coding step includes the step of encoding said first and second runlength code values with a sign value.

9. The method of claim 6 wherein said processing step forms said first values with amplitude zero, forms said second values with absolute amplitude one, and forms said other values with absolute amplitudes greater than one.

10. The method of claim 6 wherein a table is provided storing a plurality of runlength code values representing a plurality of different numbers of consecutive first values followed by said second value, and storing a plurality of second runlength code values representing a plurality of different numbers of consecutive first values followed by one of said other values, said first runlength code values and said second runlength code values statistically organized in said table such that the statistically more frequently occurring runlength code values are represented by shorter code lengths and the less frequently occurring values are represented by longer code lengths, and wherein

said step of forming first runlength code values is performed by table lookup from said table,

said step of forming second runlength code values is performed by table lookup from said table.

11. The method of claim 6 wherein said coding step further includes the step of providing an end code to designate the end of a digital number.

12. A method for processing digital signals, where the digital signals have first values, second values and other values, where the processing reduces the amount of data utilized to represent the digital signals and where the processing forms statistically coded signals such that the more frequently occurring values of digital signals are represented by shorter code lengths and the

less frequently occurring values of digital signals are represented by longer code lengths, comprising,

forming a first code value representing a set of said first values followed by said second value,

forming a second code value representing a set of said first values followed by one or more of said other values.

13. A method for processing digital signals to reduce the amount of data utilized to represent the digital signals, the steps comprising,

processing the digital signals to form processed signals where the processed signals are multivalued digital numbers and have first values, second values, . . . , j-values, (j+1)-values, . . . , n-values for j ranging from 1 to n, and have other values,

coding said processed signals to form statistically coded signals such that the more frequently occurring values of the processed signals are represented by shorter code lengths and the less frequently occurring values of coded signals are represented by longer code lengths, said coding including,

forming j^{th} runlength code values representing the number of consecutive processed signals of said first value followed by said j+1 value, for each value of j from 1 to n,

forming additional runlength code values representing the number of consecutive processed signals of said first value followed by any of said other values.

14. The method of claim 13 wherein said coding step includes the step of amplitude encoding said other values.

15. The method of claim 13 wherein said coding step includes the step of encoding said j runlength code values with a sign value.

16. The method of claim 13 wherein said processing step with n=2 forms said first values with j=1 equal to amplitude zero, forms said second values with j=2 equal to absolute amplitude one, and forms said other values with absolute amplitudes greater than one.

17. The method of claim 13 wherein said processing step forms said first values with j=1 equal to amplitude zero, forms said second values with j=2 equal to absolute amplitude one, and forms third values with j=3 equal to absolute amplitude two, and forms other values for n=3 with absolute amplitude greater than 2.

18. The method of claim 6 wherein said processing step includes multiple modes of processing said digital signals to form said processed signals, and includes the step of selecting one of said modes based upon differences in said input signals.

19. The method of claim 6 wherein said input signals represent images and are presented in sequential frames, said processing step including multiple processing modes for processing said input signals to form said processed signals, and including the step of forming the mean-square difference, d_0, between input signals from the current frame and representations of input signals from the previous frame and includes the step of forming the mean-square error, d_b, between input signals from the present frame and the best matched representation of input signals from the previous frame, said processing step including the step of comparing the difference, d_0-d_b, with a motion threshold T_M, and selecting one of said modes based on said comparison.

20. The method of claim 19 including the step of determining when d_0-d_b is less than T_M and further including the step of selecting a non-motion-compensated replenishment mode when d_0 is less than a predetermined threshold T_R and d_0-d_b is less than T_M.

21. The method of claim 19 including the step of determining when d_0-d_b is less than T_M and further including the step of selecting a non-motion-compensated DPCM mode when d_b is less than a predetermined threshold $T_{D/I}$ and d_0-d_b is less than T_M.

22. The method of claim 19 including the step of selecting an intraframe mode when d_0 is greater than a predetermined threshold $T_{D/I}$.

23. The method of claim 19 including the step of determining when d_0-d_b is greater than T_M and further including the step of selecting a motion compensated replenishment mode when d_b is less than predetermined threshold $T_{D/R}$ and d_0-d_b is greater than T_M.

24. The method of claim 19 including the step of determining when d_0-d_b is greater than T_M and further including the step of selecting a motion compensated DPCM mode whenever d_b is greater than a predetermined threshold $T_{D/R}$ and d_0-d_b is greater than T_M.

25. An apparatus for processing digital signals, where the digital signals have first values, second values and other values, to reduce the amount of data utilized to represent the digital signals and to form statistically coded signals such that the more frequently occurring values of digital signals are represented by shorter code lengths and the less frequently occurring values of digital signals are represented by longer code lengths, comprising,

means for forming first runlength code values representing the number of consecutive first values of said digital signals followed by said second value,

means for forming second runlength code values representing the number of consecutive first values of said digital signals followed by one of said other values.

26. The apparatus of claim 25 further including means for amplitude encoding said other values.

27. The apparatus of claim 25 further including means for encoding said first and second runlength code values with a sign value.

28. The apparatus of claim 25 wherein said first values have amplitude zero, said second values have absolute amplitude one, and said other values have absolute amplitudes greater than one whereby said first and second runlength codes values are formed representing the number of consecutive zeros.

29. The apparatus of claim 25 wherein said first values have the highest frequency of occurrence in said digital signals, wherein said second values have the next highest frequency of occurrence in said digital signals, and wherein said other values have the lowest frequency of occurrence in said digital signals.

30. An apparatus for processing input signals to reduce the amount of data utilized to represent the input signals, the apparatus comprising,

means for processing the input signals to form processed signals where the processed signals are digital numbers having first values, second values, and other values,

means for coding each digital number to form statistically coded signals such that the more frequently occurring values in the digital numbers are represented by shorter code lengths and the less frequently occurring values in the digital numbers are represented by longer code lengths, said means for coding including,

means for forming first runlength code values representing the number of consecutive first values followed by said second value in a digital number,

means for forming second runlength code values representing the number of consecutive first values followed by one of said other values in the digital number.

31. The apparatus of claim 30 wherein said means for coding includes means for amplitude encoding said other values.

32. The apparatus of claim 30 wherein said means for coding includes means for encoding said first and second runlength code values with a sign value.

33. The apparatus of claim 30 wherein said means for processing forms said first values with amplitude zero, forms said second values with absolute amplitude one, and forms said other values with absolute amplitudes greater than one.

34. The apparatus of claim 30 including an addressable table storing runlength code values representing different numbers of consecutive first values followed by said second value, and storing a plurality of second runlength code values representing different numbers of said first values followed by one of said other values, said first runlength code values and said second runlength code values organized in said table such that the statistically more frequently occurring runlength code values in digital numbers are represented by shorter code lengths and the less frequently occurring values in digital numbers are represented by longer code lengths, and wherein

said means for forming first runlength code values includes means for addressing said addressable table with a runlength number representing the runlength of said first value followed by said second value in order to obtain said first runlength code value from said table, and

said means for forming second runlength code values includes means for addressing said addressable table with a runlength number representing the runlength of said first value followed by one of said other values in order to obtain said second runlength code value.

35. The apparatus of claim 30 wherein said means for coding further includes means for providing an end code to designate an end of a digital number.

36. An apparatus for processing digital signals to reduce the amount of data utilized to represent the digital signals, comprising,

means for processing the digital signals to form processed signals where the processed signals are multivalued digital numbers and have first values, second values, . . . , j-values, (j+1)-values, . . . , n-values for j ranging from 1 to n, and have other values,

means for coding said processed signals to form statistically coded signals such that the more frequently occurring values in the digital numbers are represented by shorter code lengths and the less frequently occurring values in the digital numbers are represented by longer code lengths, said means for coding including,

means for forming j^{th} runlength code values representing the number of consecutive processed signals of said first value followed by said j+1 value, for each value of j from 1 to n,

means for forming additional runlength code values representing the number of consecutive processed signals of said first value followed by any of said other values.

37. The apparatus of claim 36 wherein said digital signals represent pixels forming images in sequential frames, said means for processing includes multiple mode processing means for processing said digital signals to form said processed signals, and includes means for forming the mean-square difference, d_0, between digital signals representing pixels of the current frame and digital signals representing pixels of the previous frame and includes means for forming the mean-square error, d_b, between the digital signals representing pixels in the present frame and digital signals representing the best matched pixels of the previous frame, said means for processing further including means for comparing the difference, d_0-d_b, with a motion threshold T_M, and means for selecting one of said modes based on said comparison.

38. A method for processing digital signals, where the digital signals have first values, second values and other values, where the processing reduces the amount of data utilized to represent the digital signals and where the processing forms statistically coded signals such that the more frequently occurring values of digital signals are represented by shorter code lengths and the less frequently occurring values of digital signals are represented by longer code lengths, where

a first code value is formed representing a set of said first values followed by said second value,

a second code value is formed representing a set of said first values followed by one or more of said other values

comprising,

decoding said first code value to form a set of said first values followed by said second value,

decoding said second code value to form a set of said first values followed by one or more of said other values.

39. A method for processing digital signals to reduce the amount of data utilized to represent the digital signals, the steps comprising,

processing the digital signals to form processed signals where the processed signals are multivalued digital numbers and have first values, second values, . . . , j-values, (j+1)-values, . . . , n-values for j ranging from 1 to n, and have other values,

coding said processed signals to form statistically coded signals such that the more frequently occurring values of the processed signals are represented by shorter code lengths and the less frequently occurring values of coded signals are represented by longer code lengths, said coding including,

forming j^{th} runlength code values representing the number of consecutive processed signals of said first value followed by said j+1 value, for each value of j from 1 to n,

forming additional runlength code values representing the number of consecutive processed signals of said first value followed by any of said other values

transmitting said j^{th} runlength code values and said additional runlength code values to a receiver to form received signal including received j^{th} runlength code values and received additional runlength code values,

decoding said received signals to form decoded signals, said decoding including,

decoding said received j^{th} runlength code values to form a number of consecutive decoded signals of said first value followed by said j+1 value, for each value of j from 1 to n,

decoding said received additional runlength code values to form a number of consecutive decoded signals of said first value followed by any of said other values.

40. The method of claim 39 wherein said coding step includes the step of amplitude encoding said other values.

41. The method of claim 39 wherein said coding step includes the step of encoding said j runlength code values with a sign value.

42. An apparatus for processing input signals to reduce the amount of data utilized to represent the input signals, the apparatus comprising,

means for processing the input signals to form processed signals where the processed signals are digital numbers having first values, second values, and other values,

means for coding each digital number to form statistically coded signals such that the more frequently occurring values in the digital numbers are represented by shorter code lengths and the less frequently occurring values in the digital numbers are represented by longer code lengths, said means for coding including,

means for forming first runlength code values representing the number of consecutive first values followed by said second value in a digital number,

means for forming second runlength code values representing the number of consecutive first values followed by one of said other values in the digital number,

means for transmitting said j^{th} runlength code values and said additional runlength code values to a receiver to form received signal including received j^{th} runlength code values and received additional runlength code values,

means for decoding said received signals to form decoded signals, said means for decoding including,

means for decoding said received j^{th} runlength code values to form a number of consecutive decoded signals of said first value followed by said j+1 value, for each value of j from 1 to n,

means for decoding said received additional runlength code values to form a number of consecutive decoded signals of said first value followed by any of said other values.

43. The apparatus of claim 42 wherein said means for coding includes means for amplitude encoding said other values.

44. The apparatus of claim 42 wherein said means for coding includes means for encoding said first and second runlength code values with a sign value.

45. The apparatus of claim 42 wherein said means for processing forms said first values with amplitude zero, forms said second values with absolute amplitude one, and forms said other values with absolute amplitudes greater than one.

46. The apparatus of claim 42 including an addressable table storing runlength code values representing different numbers of consecutive first values followed by said second value, and storing a plurality of second runlength code values representing different numbers of said first values followed by one of said other values, said first runlength code values and said second runlength code values organized in said table such that the statistically more frequently occurring runlength code values in digital numbers are represented by shorter code lengths and the less frequently occurring values in digital numbers are represented by longer code lengths, and wherein

said means for forming first runlength code values includes means for addressing said addressable table with a runlength number representing the runlength of said first value followed by said second value in order to obtain said first runlength code value from said table, and

said means for forming second runlength code values includes means for addressing said addressable table with a runlength number representing the runlength of said first value followed by one of said other values in order to obtain said second runlength code value.

* * * * *

EXHIBIT B

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-32, NO. 3, MARCH 1984

Scene Adaptive Coder

WEN-HSIUNG CHEN, MEMBER, IEEE, AND WILLIAM K. PRATT, SENIOR MEMBER, IEEE

Abstract—**An efficient single-pass adaptive bandwidth compression technique using the discrete cosine transform is described. The coding process involves a simple thresholding and normalization operation on the transform coefficients. Adaptivity is achieved by using a rate buffer for channel rate equalization. The buffer status and input rate are monitored to generate a feedback normalization factor. Excellent results are demonstrated for coding of color images at 0.4 bits/pixel corresponding to real-time color television transmission over a 1.5 Mbit/s channel.**

I. Introduction

TRANSFORM image coding, developed about 15 years ago, has been proven to be an efficient means of image coding [1]–[6]. In the basic transform image coding concept, an image is divided into small blocks of pixels, and each block undergoes a two-dimensional transformation to produce an equal-sized array of transform coefficients. Among various transforms investigated for image coding applications, the cosine transform has emerged as the best candidate from the standpoint of compression factor and ease of implementation [7]–[11]. With the basic system, the array of transform coefficients is quantized and coded using a zonal coding strategy [3]; the lowest spatial frequency coefficients, which generally possess the greatest energy, are quantized most finely, and the highest spatial frequency coefficients are quantized coarsely. Binary codes are assigned to the quantization levels, and the code words are assembled in a buffer for transmission. At the receiver, inverse processes occur to decode the received bit stream, and to inverse transform the quantized transform coefficients to reconstruct a block of pixels.

The basic transform image coding concept, previously described, performs well on most natural scenes. A pixel coding rate of about 1.5 bits/pixel is achievable, with no apparent visual degradation. To achieve lower coding rates, without increasing coding error, it is necessary to adaptively quantize transform coefficients so that those blocks of coefficients containing large amounts of energy are allocated more quantization levels and code bits than low energy blocks. In almost all adaptive transform coding designs to date, transforms are computed, and transform energy is measured or estimated on a first pass through the image. This information is then utilized to determine the quantization levels and code words for a second pass [9]. With this scheme, compression factors can be reduced by a factor of two or more as compared to nonadaptive coding. The practical difficulties are the memory required for the second pass and the complexity of the quantization algorithm. Both these problems are eliminated in the scene adaptive coder described in this paper.

The scene adaptive coder is a single-pass adaptive coder of relative simplicity. The following sections describe the coding scheme and present subjective and quantitative performance evaluations.

Paper approved by the Editor for Communication Theory of the IEEE Communications Society for publication after presentation at the International Conference on Communications, Philadelphia, PA, June 1981. Manuscript received September 2, 1982; revised July 8, 1983.

W. Chen is with Compression Labs, Inc., San Jose, CA 95131.
W. K. Pratt is with VICOM Systems, Inc., San Jose, CA 95131.

II. Cosine Transform Representation

The two-dimensional discrete cosine transform of a sequence $f(j, k)$ for $j, k = 0, 1, \cdots, N-1$, can be defined as [6]

$$F(u,v) = \frac{4C(u)C(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \cdot \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (1)$$

for $u, v = 0, 1, \cdots, N-1$, where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } w = \phi \\ 1 & \text{for } w = 1, 2, \cdots, N-1. \end{cases}$$

The inverse transform is given by

$$f(j,k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cdot \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (2)$$

for $j, k = 0, 1, \cdots, N-1$. Among the class of transform possessing fast computational algorithms, the cosine transform has a superior energy compaction property [6]–[9]. The following sections present some other properties of the cosine transform, which are useful to the subsequent discussion.

A. Statistical Description of DCT Coefficients

Let the pixel array $f(j, k)$ represent a sample of a random process with zero mean represented in two's complement format over an integer range $-M \leq f(j, k) \leq (M-1)$. The probability density of the cosine transform coefficients $F(u, v)$ has been modeled by a number of functions [3], [12]. Among them, the Laplacian density has been shown to provide the best fit [12]. This function can be written as

$$p(x; u, v) = \frac{1}{\sqrt{2}\,\sigma(u,v)} \exp\left\{\frac{\sqrt{2}|x|}{\sigma(u,v)}\right\} \quad (3)$$

where $\sigma(u, v)$ denotes the standard deviation of a coefficient.

B. Coefficient Bound

The maximum coefficient value for the cosine transform can be derived from (1) as

$$F_{max}(0,0) = 2f_{max} \quad (4a)$$

and

$$\frac{16}{\pi^2} f_{max} \leq F_{max}(u,v) \leq 2f_{max} \quad (4b)$$

0090-6778/84/0300-0225$01.00 © 1984 IEEE

where f_{max} is the maximum value of the discrete array $f(j, k)$ and F_{max} is the maximum value for the coefficient $F(u, v)$. If the transform is performed in a vector length of $N = 16$, then

$$F_{max}(0, 0) = 2f_{max} \tag{5a}$$

$$F_{max}(u, v) = 1.628 f_{max}. \tag{5b}$$

C. Mean Square Error Representation

The mean square quantization error between an original image $f(j, k)$ and its reconstructed image $\hat{f}(j, k)$ can be written as

$$\text{MSE} = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} E\{[f(j,k) - \hat{f}(j,k)]^2\}. \tag{6}$$

The unitary property of the cosine transform allows one to express the MSE in the transform domain as

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} E\{[F(u,v) - \hat{F}(u,v)]^2\} \tag{7}$$

which reduces to

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{n=-\infty}^{\infty} \int_{D_{n-1}}^{D_n} (x - x_n)^2 \cdot p(x; u, v)\, dx \tag{8}$$

where $p(x; u, v)$ is the probability density function, D_n is a set of decision levels, and x_n is a set of reconstruction levels. With Laplacian modeling of the probability density function, as represented in (3), the MSE becomes

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \left\{ \sigma^2(u,v) - \sum_{k=1}^{\infty} \left[(2x_n D_{n-1} + \sqrt{2}\sigma(u,v)x_n - x_n^2) \exp\left(\frac{-\sqrt{2}D_{n-1}}{\sigma(u,v)}\right) - (2x_n D_n + \sqrt{2}\sigma(u,v)x_n - x_n^2) \cdot \exp\left(\frac{-\sqrt{2}D_n}{\sigma(u,v)}\right) \right] \right\}. \tag{9}$$

This result has been verified by computer simulation.

III. Scene Adaptive Coder

Fig. 1 contains a block diagram of the scene adaptive coder. In operation, the input image undergoes a cosine transform in 16 X 16 pixel blocks. An initial threshold is established, and those transform coefficients whose magnitudes are greater than the threshold are scaled according to a feedback parameter from the output rate buffer. The scaled coefficients are quantized, Huffman coded, and fed into the rate buffer. The rate buffer operates with a variable rate input, dependent upon the instantaneous image energy, and a constant channel output rate. The buffer status (fullness) and input rate are monitored to generate the coefficient scaling factor. At the receiver, the received fixed rate data are fed to a rate buffer that generates Huffman code words at a variable rate to the decoder. The decoded transform coefficients are then inverse normalized by the feedback parameter, added to the threshold, and inverse transformed to reconstruct the output pixel block. The following sections describe the coding algorithm in greater detail.

A. Cosine Transform

Referring to the block diagram in Fig. 1, the original image is first partitioned into 16 X 16 pixel blocks. Each block of data is then cosine transformed as defined by (1). The resultant transform coefficients $F(u, v)$ are stored in a register according to the zigzag scan of Fig. 2. Scanning the data in this fashion minimizes the usage of runlength codes during the subsequent coding process.

B. Thresholding

The transform coefficients in the register undergo a threshold process in which all the coefficients, except $F(0, 0)$, that are below the threshold are set to zero, and those coefficients above the threshold are subtracted by the threshold. This results in

$$F_T(u, v) = \begin{cases} F(u,v) - T & \text{if } F(u,v) > T \\ 0 & \text{if } F(u,v) \leq T \end{cases} \tag{10}$$

where T is the threshold. Fig. 3 shows a plot of the percentage of coefficients below a threshold versus the threshold for the original images exhibited in Figs. 8(a) and 9(a). As demonstrated, more than 90 percent of the coefficients have absolute magnitude of less than a value of 3, even though the maximum coefficient value could be as large as $1.628 f_{max}$ [see 5(b)]. Therefore, the thresholding process indicated in (10) will set a major portion of coefficients to zero, and thereby limit the number of coefficients to be quantized. The value of the threshold varies with respect to the globally desired bit rate. However, it can be adjusted locally on a block-to-block basis if desired.

C. Normalization and Quantization

The threshold subtracted transform coefficients $F_T(u, v)$ are scaled by a feedback normalization factor D from the output rate buffer according to the relation

$$F_{TN}(u, v) = \frac{F_T(u, v)}{D}. \tag{11}$$

The scaling process adjusts the range of the coefficients such that a desired number of code bits can be used during the coding process.

The quantization process is simply a floating point to integer roundoff conversion. No decision and reconstruction tables are required. Therefore, there is a significant simplification and saving for a hardware implementation. Because many of the threshold subtracted coefficients are of fractional value, the roundoff process will set some of the coefficients to zero and leave only a limited number of significant coefficients to be amplitude coded. The quantized coefficients can now be represented as

$$\hat{F}_{TN}(u, v) = \text{integer part of } [F_{TN}(u, v) + 0.5]. \tag{12}$$

It should be noted that a lower bound has to be set for the normalization factor in order to introduce meaningful transform coefficients to the coder. This lower bound is dependent upon how accurately the cosine transform is computed. Generally speaking, setting the minimum value of D to unity is sufficient for most of the compression applications. In this case, the worst-case quantization error can be obtained from (9) by letting $D_{n-1} = n - 0.5$, $D_n = n + 0.5$, and $x_n = n$.

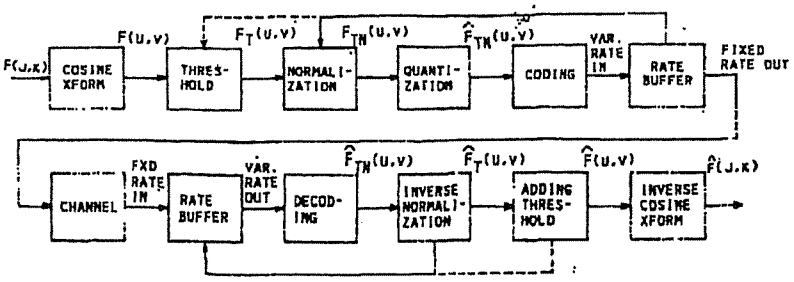


Fig. 1. Block diagram of scene adaptive coding/decoding system.

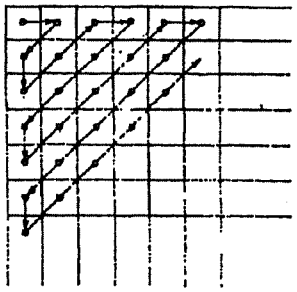


Fig. 2. Zigzag scan of cosine transform coefficients.

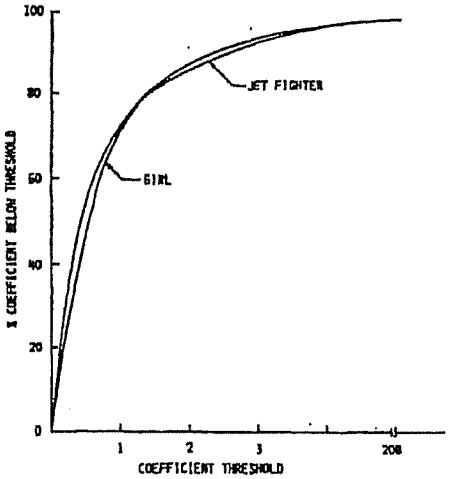


Fig. 3. Distribution of cosine transform coefficients.

Thus

$$\mathrm{MSE} = \frac{1}{4}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} e(u,v) \tag{13}$$

where

$$e(u,v) = \sigma^2(u,v) - (1/2)\sum_{n=1}^{\infty}\left\{[n^2 + (\sqrt{2}\sigma(u,v) - 1)n] \cdot \exp\left[\frac{-\sqrt{2}(n-0.5)}{\sigma(u,v)}\right] - [n^2 + (\sqrt{2}\sigma(u,v) + 1)n]\exp\left[\frac{-\sqrt{2}(n+0.5)}{\sigma(u,v)}\right]\right\}.$$

Fig. 4 shows the functional relationship between $e(u, v)$ and $\sigma(u, v)$. As can be seen, $e(u, v)$ is always less than 1/12. (Note: 1/12 is the quantization error for a uniform density.) Therefore, the MSE represented by (13) is always less than $N^2/48$. For $N = 16$, this MSE is less than 16/3, which corresponds to a normalized error of 0.0082 percent. This MSE also corresponds to a peak-to-peak signal-to-quantization-noise ratio of more than 40.9 dB, which is relatively insignificant.

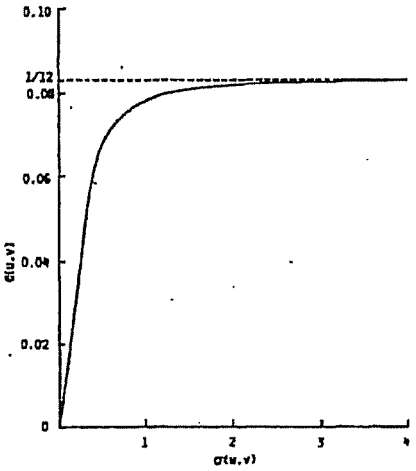


Fig. 4. Quantization error $e(u, v)$ versus coefficient standard deviation $\sigma(u, v)$ at normalization factor of unity and coding threshold of zero. (Probability density function of the coefficient is assumed to be Laplacian.)

D. Coding

The coefficient $F(0, 0)$ in the upper left-hand corner of each luminance transform block is proportional to the average luminance of that block. Because block-to-block luminance variations resulting from quantization of $F(0, 0)$ are easily discerned visually, $F(0, 0)$ is linearly quantized and coded with a 9 bit code. As for the other nonzero coefficients, their magnitudes are coded by an amplitude lookup table, and the addresses of the coefficients are coded using a runlength lookup table. The amplitude and runlength lookup tables are simply Huffman codes derived from the histograms of typical transform coefficients. As demonstrated by the histograms of Fig. 5, the domination of low amplitudes and short runs of zero-valued coefficients indicates that both Huffman tables are relatively insensitive to the type of input images and the de-

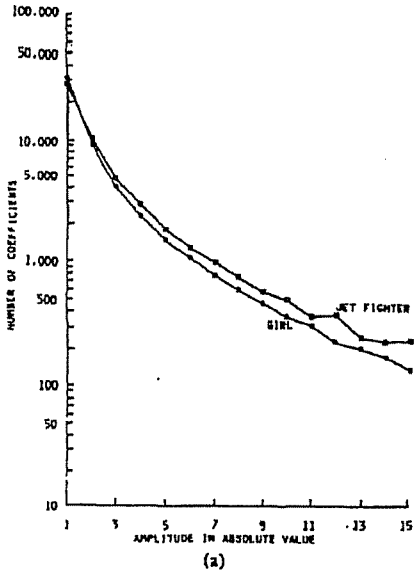


Fig. 5. (a) Histogram of cosine transform coefficients obtained with threshold of zero and normalization factor of one. (b) Histogram of runs of consecutive zero counts obtained with threshold of zero and normalization factor of one.

sired bit rate. This suggests that only two predetermined tables are needed for the coding process.

The length of the Huffman table for the amplitude codes is a function of the normalization factor and the transform bound represented by (5b). For a two-dimensional cosine transform of 16 X 16 pixels, the maximum length is

$$L_A = \frac{1.628\,|f_{max}|}{D_{min}} \tag{14}$$

where D_{min} is the minimum allowable normalization factor. For $|f_{max}|$ and D_{min} equal to 128 and 1, respectively, the length will be 208. As for the runlength of zero counts, the

TABLE I
HUFFMAN CODE TABLE FOR COEFFICIENT AMPLITUDE IN ABSOLUTE VALUE

AMPLITUDE	NUMBER OF CODE BITS	HUFFMAN CODES
1	1	1
2	3	001
3	4	0[illegible]1
4	5	00001
5	5	01101
6	6	011001
7	7	0000001
8	7	0110001
9	8	00000000
10	8	01100000
11	8	00000001
12	8	01100001
13	6+8	000001+8 BITS
EOB	4	0001
RL PREFIX	3	010

length of the table can be represented by

$$L_R = (N^2 - 1) \tag{15}$$

where N is the transform block size. The subtraction of unity is because the dc coefficient in individual blocks is separately encoded. For a transform block size of 16 X 16 pixels, the length is 255. In practice, the length of both tables can be shortened to less than 30 entries by assigning Huffman codes to only the low amplitude coefficients and short runlengths (using a fixed length code elsewhere). Again, due to the domination of the low amplitude coefficients and short runlengths, the loss of coding efficiency is insignificant.

Tables I and II show typical truncated Huffman code tables for the amplitude and runlength, respectively. It should be noted that the amplitude codes in Table I include an "end of block (EOB)" code and a "runlength prefix" code. The EOB code is used to terminate coding of the block as soon as the last significant coefficient of the block is coded. The runlength prefix code is required in order to distinguish the runlength code from the amplitude code.

It should be noted that there are many ways to improve the coding efficiency of the coder. One way is to cut down the number of runs in the runlength coder. This can be accomplished by skipping single isolated coefficients with absolute magnitude of one, or by introducing an amplitude code for the isolated coefficients with zero amplitude. The coding improvement is generally quite significant; this is especially true if the average coding rate is low.

E. Rate Buffer

The rate buffer in the SAC performs channel rate equalization. The buffer has a variable rate data input and a constant data output. The differentials are monitored from block to block, and the status is converted into a scaling factor that is fed back to the normalizer. The buffer always forces the coder to adjust to the local coding variations, while ensuring global performance at a desired level. The general operation of a rate buffer is well documented [13], [14]. The specific method used in this paper is described as follows.

Let $B(m)$ represent the number of bits into the rate buffer for the mth block and let $S(m)$ represent the normalized buffer status at the end of the mth block ($-0.5 < S(m) < 0.5$). Then, $B(m)$ and $S(m)$ can be written as

$$B(m) = \sum_{\substack{u=0 \\ (u,v)\neq(0,0)}}^{15} \sum_{v=0}^{15} H\{[\hat{F}_{TN}(u,v)]_m\} + 9 \tag{16}$$

TABLE II
HUFFMAN CODE TABLE FOR THE NUMBER OF CONSECUTIVE ZERO-VALUED COEFFICIENTS

RUN-LENGTH	NUMBER OF CODE BITS	HUFFMAN CODE
1	2	:1
2	3	:01
3	3	011
4	4	0101
5	4	0011
6	5	01000
7	5	10010
8	5	01001
9	5	10001
10	5	10011
11	5	001000
12	6	100000
13	6	001010
14	6	001001
15	6	100001
16	6	000011
17	6	001011
18	7	0000000
19	7	0000100
20	7	0000010
21	7	0001110
22	7	0000001
23	7	0000101
24	7	0000011
25	7	0001111
26	8	00011000
27	8	00011010
28	8	00011001
29	8	00011011
30	5+8	00010+8 BITS

$$S(m) = S(m-1) + \frac{[B(m) - 256R]}{L} \tag{17}$$

where

$[\hat{F}_{TN}(u, v)]_m$	quantized coefficients of the mth block, as defined in (12)
$H\{\cdot\}$	Huffman coding function
R	average coding rate
L	rate buffer size.

The buffer status $S(m)$ is used to select an instantaneous normalization factor $\hat{D}(m)$ according to an empirically determined "normalization factor versus status" curve. This relationship is described by

$$\hat{D}(m) = \Phi\{S(m)\}. \tag{18}$$

In order to smooth out this instantaneous normalization factor such that the desired normalization factor does not fluctuate too much, a recursive filtering process is applied to generate

$$D(m) = cD(m-1) + (1-c)\hat{D}(m) \tag{19}$$

where c is a constant with value less than unity.

The desired operating conditions for the rate buffer algorithm are: a) the feedback normalization factor is as stable as possible; b) the buffer status is able to converge rapidly and stay as close to the half full position ($S(m) = 0$) as possible. Both these conditions may be satisfied using the above set of equations. Fig. 6 shows typical values of the normalization factor and buffer status as a function of block indexes for the images shown in Figs. 8(a) and 9(a).

The rate buffer can be guaranteed not to overflow. This is because the normalization factor can get very large within a few blocks of operation, and effectively limit the data going into the buffer. However, there is no guarantee that the buffer will not underflow if a minimum allowable normalization factor is set to a fixed value. Therefore, the buffer status has to be constantly monitored and, if the status is closer to -0.5, fill bits must be introduced into the channel.

IV. Scene Adaptive Coding of Color Images

Fig. 7 contains a block diagram of a color image coding system based on the scene adaptive coder. In this system, a color image, represented by tristimulus signals $R(j, k)$, $G(j, k)$, $B(j, k)$, is first converted to a new three-dimensional space defined by

$$\begin{bmatrix} Y(j,k) \\ I(j,k) \\ Q(j,k) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.589 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.253 & 0.312 \end{bmatrix} \begin{bmatrix} R(j,k) \\ G(j,k) \\ B(j,k) \end{bmatrix} \tag{20}$$

where $Y(j, k)$ is the luminance signal and $I(j, k)$ and $Q(j, k)$ are chrominance signals. This conversion compacts most of the signal energy into the Y plane such that more efficient coding can be accomplished [15]. The I and Q chrominance planes are spatially averaged and subsampled by a factor of 4 to 1 in both the horizontal and vertical directions. The luminance and subsampled chrominance images are then partitioned into 16 X 16 pixel blocks and coded by the SAC in the order of 32 Y, two I, and two Q sequences. At the receiver, the received code bits are decoded. Inverse cosine transform and inverse coordinate conversions are then performed to reconstruct the source tristimulus signals. The inverse coordinate conversion is described by

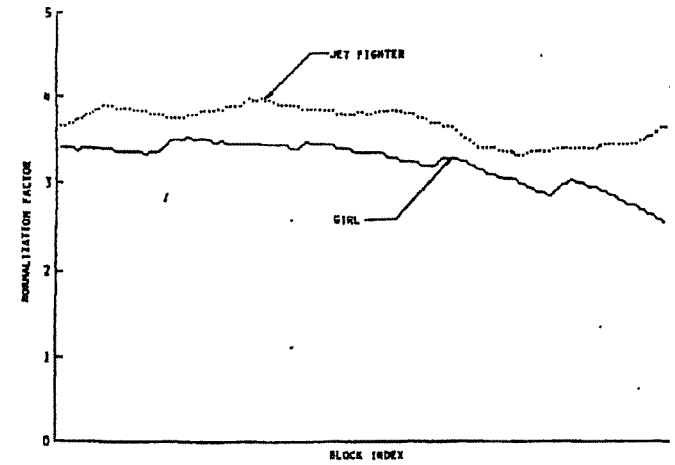
$$\begin{bmatrix} R(j,k) \\ G(j,k) \\ B(j,k) \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y(j,k) \\ I(j,k) \\ Q(j,k) \end{bmatrix}. \tag{21}$$

V. Simulation Results

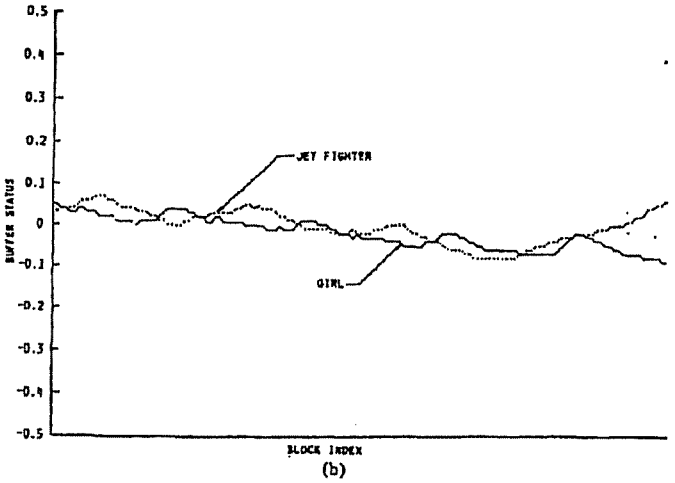
Computer simulations have been conducted to evaluate the performance of the scene adaptive coder. The original test images shown in Figs. 8(a) and 9(a) are of size 512 X 512 pixels with each red, green, and blue tristimulus value uniformly quantized to 8 bits/pixel. Figs. 8(b) and 9(b) show the reconstructed images at a combined average bit rate of 0.4 bits/pixel. This rate corresponds to a channel bandwidth of 1.5 Mbits for a 15 frame/s intraframe coding system. The excellent reconstruction of the images is clearly demonstrated. Table III tabulates the average mean square error between the original and the reconstructed images. Also included in the table is the peak signal-to-noise ratio for the reconstructed image.

VI. Summary

The scene adaptive coder described herein encodes cosine transform coefficients in a simple manner. The coding process involves only thresholding, normalization, roundoff, and rate buffer equalization. The performance of the coder is quite good in terms of mean square error and subjective evaluation. Because the coding process is dependent upon the instantaneous coefficient content inside the block and the accumulated rate buffer content, it is well suited for intraframe coding of moving images. At Compression Labs, Inc., the coder has been implemented with real-time hardware to code NTSC color video at a channel rate of 1.5 Mbits/s.



(a)



(b)

Fig. 6. Buffer status for the last 114 blocks of images shown in Figs. 8(a) and 9(a).

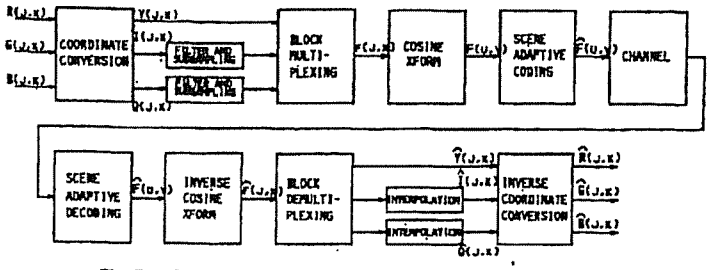


Fig. 7. Cosine transform color image coding/decoding system.



(a)

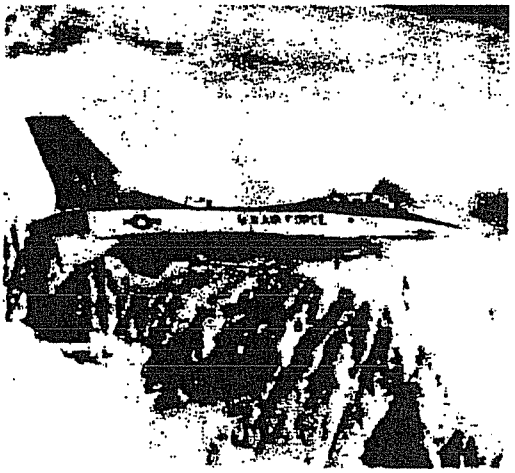


(b)

Fig. 8 Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.

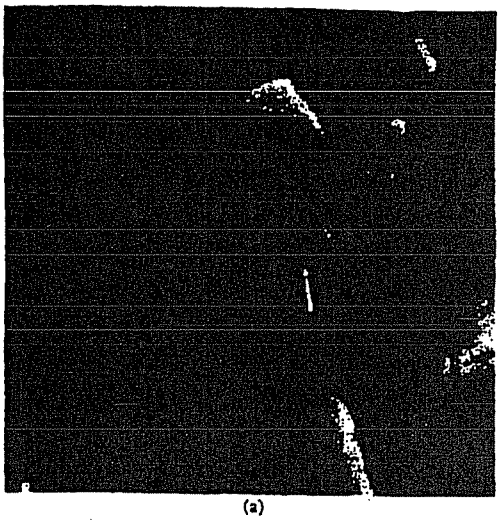


(a)

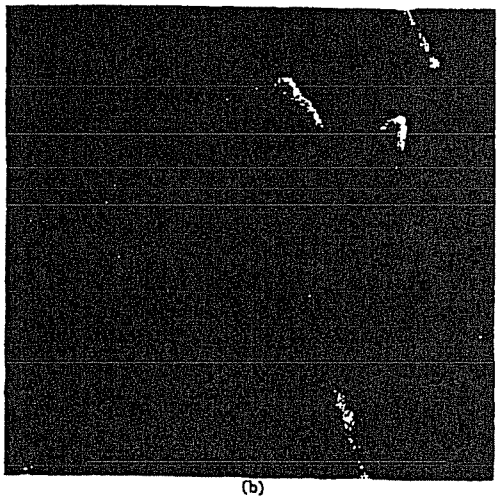


(b)

Fig. 9. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.



(a)

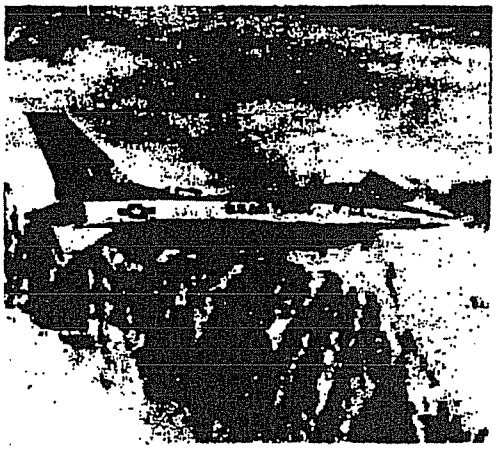


(b)

Fig. 8. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.



(a)



(b)

Fig. 9. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.

TABLE III
MEAN SQUARE ERROR BETWEEN ORIGINAL AND RECONSTRUCTED IMAGES AT 0.4 BITS/PIXEL; MEAN SQUARE ERROR IS COMPUTED AT THE INPUT OF CODER AND THE OUTPUT OF DECODER WITH $f_{max}(j, k)$ NORMALIZED TO 1

IMAGES	MEAN SQUARE ERROR	SNR
GIRL	0.0359%	34.45 DB
JET	0.0521%	32.84 DB

ACKNOWLEDGMENT

The authors wish to express their sincere gratitude to Dr. A. Tescher for his valuable discussions and suggestions in the area of rate buffer stabilizations.

REFERENCES

[1] H. C. Andrews and W. K. Pratt, "Fourier transform coding of images," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan. 1968, pp. 677–678.
[2] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, pp. 58–68, Jan. 1969.
[3] W. K. Pratt, W. H. Chen, and R. Welch, "Slant transform image coding," *IEEE Trans. Commun.*, vol. COM-22, pp. 1075–1093, Aug. 1974.
[4] A. Habibi and P. A. Wintz, "Image coding by linear transformation and block quantization," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 50–62, Feb. 1971.
[5] W. K. Pratt, *Digital Image Processing*. New York: Wiley-Interscience, 1978.
[6] N. Ahmed, T. Natarjan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.
[7] M. Hamidi and J. Pearl, "Comparison of the cosine and Fourier transforms of Markov-1 signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 428–429, Oct. 1976.
[8] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, Oct. 1979.
[9] W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1285–1292, Nov. 1977.
[10] W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004–1009, Sept. 1977.
[11] R. M. Haralick, "A storage efficient way to implement the discrete cosine transform," *IEEE Trans. Comput.*, vol. C-25, pp. 764–765, July 1976.
[12] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, pp. 835–839, June 1983.
[13] A. Tescher, "Rate adaptive communication," in *Proc. Nat. Telecommun. Conf.*, 1978.
[14] ——, "A dual transform coding algorithm," in *Proc. Nat. Telecommun. Conf.*, 1980.
[15] W. K. Pratt, "Spatial transform coding of color images," *IEEE Trans Commun. Technol.*, vol. COM-19, pp 980–992, Dec. 1971.



Wen-hsiung Chen (M'82) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1962, the M.S. degree from Kansas State University, Manhattan, in 1966, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1973, all in electrical engineering.

From 1966 to 1968 he was employed with the Allis-Chalmers Company, Harvey, IL, responsible for solid-state circuitry development. He was a Teaching and Research Assistant and Research Associate at the Department of Electrical Engineering, University of Southern California, from 1968 to 1973. In 1973 he joined Ford Aerospace and Communications Corporation, Palo Alto, CA, to work on communication systems analysis and digital image processing. In 1977 he left to help form Compression Labs, Inc., San Jose, CA, where he developed the combined symbol matching coder for coding facsimile images, the intraframe coder for coding NTSC color video at 1.5 Mbits, and the combined intraframe and interframe coder for coding color video at less than 750 kbits. As Chief Scientist of the company, he is responsible for research and development work in digital image processing.

★



William K. Pratt (S'57–M'61–SM'75) received the B.S. degree in electrical engineering from Bradley University, Peoria, IL, in 1959 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1961 and 1965, respectively.

He received Master's and Doctoral Fellowships from Hughes Aircraft Company and was employed there from 1959 to 1965. He became an Assistant Professor of Electrical Engineering at the University of Southern California in 1965, an Associate Professor in 1969, and a Full Professor in 1975. At U.S.C. he was the Director of the Image Processing Institute and of the Engineering Computer Laboratory. He is now the President and Chairman of the Board of Vicom Systems, Inc., San Jose, CA.

In 1976 Dr. Pratt was awarded a Gugenheim Fellowship for research and image analysis techniques. He is a member of Sigma Tau, Omicron Delta Kappa, and Sigma Xi.

EXHIBIT B

Issued by the

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF TEXAS

U.S. PHILIPS CORPORATION

Plaintiff

V.

EASTMAN KODAK COMPANY

Defendant.

SUBPOENA IN A CIVIL CASE

Case Number:[1] Civil Action No. 06-00251
USDC, District of Delaware

TO: AT&T Corp.
208 South Akard
Dallas, TX 75202
214.464.8331

Service of Process on:
CT Corporation System
Attn: Service of Process
111 8th Avenue, 13th Floor
New York, NY 10011
212.894.8940

☐ YOU ARE COMMANDED to appear in the United States District court at the place, date, and time specified below to testify in the above case.

PLACE OF TESTIMONY	COURTROOM
	DATE AND TIME

☒ YOU ARE COMMANDED to appear at the place, date, and time specified below to testify at the taking of a deposition in the above case. See Attachment A for deposition topics.

PLACE OF DEPOSITION	DATE AND TIME
DLA Piper US LLP 1717 Main Street, Suite 4600 Dallas, TX 75201-4629 214.743.4500	August 3, 2007 at 9:00 a.m.

☒ YOU ARE COMMANDED to produce and permit inspection and copying of the following documents or objects at the place, date, and time specified below (list documents or objects):

See Attachment A

PLACE	DATE AND TIME
DLA Piper US LLP 401 B Street, Suite 1700 San Diego, CA 92101 619.699.2700	July 20, 2007

☐ YOU ARE COMMANDED to permit inspection of the following premises at the date and time specified below.

PREMISES	DATE AND TIME

Any organization not a party to this suit that is subpoenaed for the taking of a deposition shall designate one or more officers, directors, or managing agents, or other persons who consent to testify on its behalf, and may set forth, for each person designated, the matters on which the person will testify. Federal Rules of Civil Procedure, 30(b)(6).

ISSUING OFFICER'S SIGNATURE AND TITLE (INDICATE IF ATTORNEY FOR PLAINTIFF OR DEFENDANT)	DATE
Attorney for Defendant Eastman Kodak Company Kristen H. Cramer	June 29, 2007

ISSUING OFFICER'S NAME, ADDRESS AND PHONE NUMBER

Kristen H. Cramer
Connolly Bove Lodge and Hutz
1007 N. Orange St.
P.O. Box 2207
Wilmington, DE 19899
302.658.9141

(See Rule 45, Federal Rules of Civil Procedure, Subdivisions (c), (d), and (e), on next page)

[1] If action is pending in district other than district of issuance, state district under case number.

PROOF OF SERVICE

DATE	PLACE
SERVED	
SERVED ON (PRINT NAME)	MANNER OF SERVICE
SERVED BY (PRINT NAME)	TITLE

DECLARATION OF SERVER

I declare under penalty of perjury under the laws of the United States of America that the foregoing information contained in the Proof of Service is true and correct.

Executed on _____ (DATE) _____ (SIGNATURE OF SERVER)

_____ (ADDRESS OF SERVER)

Rule 45, Federal Rules of Civil Procedure, Subdivisions (c), (d), and (e), as amended on December 1, 2006:

(C) PROTECTION OF PERSONS SUBJECT TO SUBPOENAS.

(1) A party or an attorney responsible for the issuance and service of a subpoena shall take reasonable steps to avoid imposing undue burden or expense on a person subject to that subpoena. The court on behalf of which the subpoena was issued shall enforce this duty and impose upon the party or attorney in breach of this duty an appropriate sanction, which may include, but is not limited to, lost earnings and a reasonable attorney's fee.

(2) (A) A person commanded to produce and permit inspection, copying, testing, or sampling of designated electronically stored information, books, papers, documents or tangible things, or inspection of premises need not appear in person at the place of production or inspection unless commanded to appear for deposition, hearing or trial.

(B) Subject to paragraph (d)(2) of this rule, a person commanded to produce and permit inspection, copying, testing, or sampling may, within 14 days after service of the subpoena or before the time specified for compliance if such time is less than 14 days after service, serve upon the party or attorney designated in the subpoena written objection to producing any or all of the designated materials or inspection of the premises — or to producing electronically stored information in the form or forms requested. If objection is made, the party serving the subpoena shall not be entitled to inspect, copy, test, or sample the materials or inspect the premises except pursuant to an order of the court by which the subpoena was issued. If objection has been made, the party serving the subpoena may, upon notice to the person commanded to produce, move at any time for an order to compel the production, inspection, copying, testing, or sampling. Such an order to compel shall protect any person who is not a party or an officer of a party from significant expense resulting from the inspection, copying, testing, or sampling commanded.

(3) (A) On timely motion, the court by which a subpoena was issued shall quash or modify the subpoena if it

(i) fails to allow reasonable time for compliance;

(ii) requires a person who is not a party or an officer of a party to travel to a place more than 100 miles from the place where that person resides, is employed or regularly transacts business in person, except that, subject to the provisions of clause (c)(3)(B)(iii) of this rule, such a person may in order to attend trial be commanded to travel from any such place within the state in which the trial is held;

(iii) requires disclosure of privileged or other protected matter and no exception or waiver applies; or

(iv) subjects a person to undue burden.

(B) If a subpoena

(i) requires disclosure of a trade secret or other confidential research, development, or commercial information, or

(ii) requires disclosure of an unretained expert's opinion or information not describing specific events or occurrences in dispute and resulting from the expert's study made not at the request of any party, or

(iii) requires a person who is not a party or an officer of a party to incur substantial expense to travel more than 100 miles to attend trial, the court may, to protect a person subject to or affected by the subpoena, quash or modify the subpoena or, if the party in whose behalf the subpoena is issued shows a substantial need for the testimony or material that cannot be otherwise met without undue hardship and assures that the person to whom the subpoena is addressed will be reasonably compensated, the court may order appearance or production only upon specified conditions.

(D) DUTIES IN RESPONDING TO SUBPOENA.

(1) (A) A person responding to a subpoena to produce documents shall produce them as they are kept in the usual course of business or shall organize and label them to correspond with the categories in the demand.

(B) If a subpoena does not specify the form or forms for producing electronically stored information, a person responding to a subpoena must produce the information in a form or forms in which the person ordinarily maintains it or in a form or forms that are reasonably usable.

(C) A person responding to a subpoena need not produce the same electronically stored information in more than one form.

(D) A person responding to a subpoena need not provide discovery of electronically stored information from sources that the person identifies as not reasonably accessible because of undue burden or cost. On motion to compel discovery or to quash, the person from whom discovery is sought must show that the information sought is not reasonably accessible because of undue burden or cost. If that showing is made, the court may nonetheless order discovery from such sources if the requesting party shows good cause, considering the limitations of Rule 26(b)(2)(C). The court may specify conditions for the discovery.

(2) (A) When information subject to a subpoena is withheld on a claim that it is privileged or subject to protection as trial-preparation materials, the claim shall be made expressly and shall be supported by a description of the nature of the documents, communications, or things not produced that is sufficient to enable the demanding party to contest the claim.

(B) If information is produced in response to a subpoena that is subject to a claim of privilege or of protection as trial-preparation material, the person making the claim may notify any party that received the information of the claim and the basis for it. After being notified, a party must promptly return, sequester, or destroy the specified information and any copies it has and may not use or disclose the information until the claim is resolved. A receiving party may promptly present the information to the court under seal for a determination of the claim. If the receiving party disclosed the information before being notified, it must take reasonable steps to retrieve it. The person who produced the information must preserve the information until the claim is resolved.

(e) CONTEMPT. Failure of any person without adequate excuse to obey a subpoena served upon that person may be deemed a contempt of the court from which the subpoena issued. An adequate cause for failure to obey exists when a subpoena purports to require a nonparty to attend or produce at a place not within the limits provided by clause (ii) of subparagraph (c)(3)(A).

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

U.S. PHILIPS CORPORATION, Plaintiff, v. EASTMAN KODAK COMPANY, Defendant.	)	Civil Action No. 06-00251-GMS

EASTMAN KODAK COMPANY'S SUBPOENA *AD TESTIFICANDUM* AND *DUCES TECUM* OF AT&T CORP.

PLEASE TAKE NOTICE THAT, pursuant to Rules 30 and 45 of the Federal Rules of Civil Procedure, Eastman Kodak Company ("Kodak"), by and through its counsel, will take the deposition of AT&T Corp. ("AT&T"), on August 3, 2007 at 9:00 a.m. at the offices of DLA Piper US LLP, 1717 Main Street, Suite 4600, Dallas, Texas 75201-4629, (214) 743-4500 (or at a time and place as otherwise agreed by the parties).

The deposition, on the deposition topics listed in Attachment A, will take place upon oral examination pursuant to the Federal Rules of Civil Procedure before an officer duly authorized by law to administer oaths, and will continue from day to day until completed. Some or all of the deposition may be recorded stenographically and may be recorded by videotape. Some or all of the deposition testimony may involve real-time computer connection between the deposition-taker and stenographer using software such as "LiveNote."

Before the deposition, on July 20, 2007 at the offices of DLA Piper US LLP, 401 B Street, Suite 1700 San Diego, CA 92101, (619) 699-2700 (or at the time and place as otherwise agreed by the parties), AT&T must produce documents in its possession, custody and control in response to the document requests listed in Attachment A. Fed.R.Civ.P. 45(a)(1)(C).

Dated: June 29, 2007

Nicole Wyll

John Allcock
Sean C. Cunningham
Tim Lohse
John D. Kinton
Jesse Hindman
Nicole Wyll
DLA PIPER US LLP
401 B Street, Suite 1700
San Diego, CA 92101-4297
Tel: 619.699.2700
Fax: 619.699.2701
Attorneys for Defendant
Eastman Kodak Company

ATTACHMENT A

TO THE SUBPOENA FOR AT&T CORP.

DEFINITIONS

1. "AT&T," "you" or "your" means AT&T Corp., including without limitation all of its corporate locations, predecessors, affiliates, successors, subsidiaries, divisions or parts thereof; all past or present directors, officers, agents, representatives, employees, consultants, attorneys, and entities acting in joint-venture or partnership relationships with AT&T; and others acting on behalf of AT&T.

2. The term "Philips" means U.S. Philips Corporation including its predecessors, successors, assigns, past and present subsidiaries, past and present corporate parents, past or present subsidiaries of its corporate parents, divisions, affiliates, officers, directors, agents and employees, attorneys and other persons acting on its behalf, including Royal Philips Electronics, Philips Kommunikations Industrie AG (Germany), Philips Research Laboratories (United Kingdom), Philips Data Systems, Philips International B.V. (Netherlands), Philips Telecommuicatie Industrie (Netherlands), Philips Consumer Electronics B.V. (Netherlands), Philips Patentverwaltung GmbH, N. V. Philips Gloeilampenfabrieken and Peter Vogel.

3. The term "Widcom" means Widcom, Inc., including its predecessors, successors, assigns, past and present subsidiaries, past and present corporate parents, divisions, affiliates, officers, directors, agents and employees, attorneys and other persons acting on its behalf, including Widergren Communications, Inc., and Robert Widergren.

4. "Widcom VTC-56" and "VTC-56" mean the coder-decoder, or "codec," that was developed by Widcom and was claimed to be capable of compressing a video image and

transmitting it over 56 kilobits-per-second lines or links, and includes all of its prototypes as well as its previous, predecessor, and subsequent models.

5. "Widcom's Personal Videoconferencing Station" and "PVS" mean the videoconferencing system developed by Widcom and referred to by those names, and includes all of its prototypes as well as its previous, predecessor, and subsequent models.

6. As used herein, "product" refers to any system, apparatus, method, device, software, encoder, decoder, codec, and/or process using Image Coding Technology.

7. "Image Coding Technology" means any technology for compressing and/or decompressing data representing moving images, facsimile images and/or still images, including but not limited to the coding methods described in the JPEG Standard and MPEG Standards.

8. "Standard" or "Standards" or "Recommendation" refers to any documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and/or services are fit for their purpose.

9. "Standards Body" refers to any organization that maintains and/or issues documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and/or services are fit for their purposes. To illustrate, the ITU-T (formerly CCITT) and ISO are Standards Bodies.

10. "ITU-T" means the International Telecommunication Union Telecommunication Standardization Sector and all predecessors, including the International Telephone and Telegraph Consultative Committee ("CCITT").

11. "ISO" means the International Organization for Standardization.

12. "IEC" means the International Electrotechnical Commission.

13. The term "participation" refers to involvement of any kind relating to the activity (such as development, consideration, or adoption of a standard) and/or organization (such as the ISO and/or CCITT) referred to in the request.

14. The phrase "JPEG Standard" refers to the Standard Recommendation ISO/IEC 10918-1|ITU-T Rec. T.81, entitled "Information Technology — Digital Compression and Coding of Continuous-Tone Still Images — Requirements and Guidelines."

15. The phrase "MPEG Standards" refer collectively to one or more of the ISO/IEC 11172-2 Standard for "MPEG-1," the ISO/IEC 13818-2 Standard for "MPEG-2," and the ISO/IEC 14496-2 Standard for "MPEG-4."

16. "Pirsch patent" means United States Patent No. 4,420,771, entitled "Technique for Encoding Multi-Level Signals" to Peter Pirsch, issued December 13, 1983, attached hereto as Exhibit A.

17. "Pirsch article" means the article entitled "Adaptive Intra-Interframe DPCM Coder," published in The Bell System Technical Journal, Vol. 61, No. 5, pp. 747-64 (May-June 1982), attached hereto as Exhibit B.

INSTRUCTIONS

1. This subpoena calls for you to produce all documents identified below that are within your possession, custody, or control, or are otherwise available to you.

2. Electronic records and computerized information that are produced must be readable with standard commercial software or must be accompanied by a description of the system from which it was derived sufficient to render such materials readable.

3. If, after exercising due diligence to secure the information requested, you cannot fully comply with a specific Request, or any part thereof, please state the reason(s) for your inability to reply and respond to the fullest extent possible.

DOCUMENT REQUESTS- GENERAL

1. All documents authored by AT&T or by any of its employees concerning Image Coding Technology prior to September 11, 1987, including such documents by Peter Pirsch.

2. Documents sufficient to fully describe the methods for image encoding and decoding used by any Image Coding Technology for: (1) any projects or products that led to the inventions of the Pirsch patent and/or Pirsch article; (2) any projects or products that were based on the inventions of the Pirsch patent and/or Pirsch article prior to September 11, 1987; (3) any other products made by or at the direction of AT&T that used Image Coding Technology prior to September 11, 1987; and (4) any other projects of AT&T that used Image Coding Technology prior to September 11, 1987.

3. Documents sufficient to describe any AT&T projects or products disclosed in documents produced in response to Document Request Number 2, including documents sufficient to describe the project and/or product names and model numbers.

4. Documents sufficient to demonstrate the public use, knowledge, availability, offer for sale and/or sale of any project or product disclosed in documents produced in response to Document Request Number 2 and which occurred prior to September 11, 1987, including documents sufficient to demonstrate such use, knowledge, availability, offer for sale and/or sale that occurred prior to September 11, 1987.

5. All documents concerning patent licenses from AT&T for (1) U.S. Patent No. 4,973,961, to Christos Chamzas and Donald L. Duttweiler, entitled "Method and Apparatus for

Carry-over Control in Arithmetic Entropy Coding," issued November 27, 1990; and (2) U.S. Patent No. 5,025,258 to Donald L. Duttweiler, entitled "Adaptive Probability Estimator for Entropy Encoding/Decoding," issued June 18, 1991; and (3) the Pirsch Patent, regardless of whether a license was actually executed, including patent licenses for each patent.

6. All documents concerning any device, apparatus or product made by or otherwise associated with Widcom, including but not limited to the Widcom VTC-56 and/or the Widcom Personal Videoconferencing Station, and including but not limited to documents concerning the research, development, conception, reduction to practice, use, sale, offer for sale, distribution, display, and marketing thereof by any person or entity.

7. All documents concerning any video conferencing codec, video compressor or video compression system or technology developed by Widcom for other companies, including without limitation Bell & Howell and Defense Advanced Research Projects Agency ("DARPA"), including but not limited to documents concerning the research, development, conception, reduction to practice, use, sale, offer for sale, distribution, display, and marketing thereof by any person or entity.

8. All documents concerning papers, articles, publications, public statements, reviews, announcements, and news reports by anyone regarding any product associated with Widcom, including but not limited to the Widcom VTC-56, the Widcom Personal Videoconferencing Station, any video compressor developed by Widcom for Bell & Howell, or any video conferencing codec developed by Widcom for DARPA, including but not limited to all drafts and final versions of any such materials and any memoranda or correspondence between co-authors or others concerning either the work reflected in the materials or the publications themselves.

DOCUMENTS REQUESTED- STANDARDS SETTING

9. All documents relating to any disclosure by You of Image Coding Technology to the ISO and/or CCITT Standards Bodies concerning the JPEG Standard prior to February 15, 1994, including (but not limited to) any proposal, submission, description, contribution and/or technical specification.

10. All documents relating to any participation by AT&T, or any employee of AT&T, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994, with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification, including participation by Barry G. Haskel, Nikil S. Jayant, James J. Johnston, R. Aravind, Hermant Behda, Guido Bertocci, Christos Chamzas, Stephen A. Corteselli, William R. Daumer, T.C. Dodd, Donald L. Duttweiler, Adrian Ligtenberg, David Malah, Ms. G. Mascavich (or Mascarich), David S. Nelson, Atal Puri, Sharad Singhal, D. Stanwyck, Ming-Ting Sun, Donald R. Thelen, Tony G. Tao, and Richard Yackel.

11. All Communications between AT&T and any of its employees and the ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee relating to the JPEG Standard and/or MPEG Standards, including communications to or from those employees listed in Request Number 10.

12. All documents relating to any participation by Philips, or any employee of Philips, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994, with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification, and including participation by current or former Philips employees, including but not limited to Loeck Zeckendorf, Gerald Weth, Ulf Rothgordt, Peter Vogel, Karl Heinz Wenzel, Dr. Cord Brandis, Jan van der Meer, Norman Richards, John Morris, Simon Turner, Hendrick Eisema, Louis Van Loon, Erik Schylander, P. De Wit, Burkhard Krupa and Joachim Spiedel.

DEPOSITION TOPICS- GENERAL

1. The authenticity of any documents produced by AT&T in response to this subpoena.

2. The search for and collection of any documents relevant or responsive to this subpoena.

3. The methods for image encoding and decoding used by any Image Coding Technology for: (1) any projects or products that lead to the inventions of the Pirsch patent and/or Pirsch article; (2) any projects or products that were based on the inventions of the Pirsch patent and/or Pirsch article prior to September 11, 1987; (3) any other products made by or at the direction of AT&T that used Image Coding Technology prior to September 11, 1987; and (4) any other projects of AT&T that used Image Coding Technology prior to September 11, 1987.

4. Any AT&T projects or products disclosed in documents produced in response to Document Request Numbers 2-3, including project and/or product names and model numbers.

5. The public use, knowledge, availability, offer for sale and/or sale of any project or product disclosed in documents produced in response to Document Request Number 4 and which occurred prior to September 11, 1987, including such use, knowledge, availability, offer for sale and/or sale that occurred prior to September 11, 1987.

6. Patent licenses or attempts by AT&T to license (1) U.S. Patent No. 4,973,961, to Christos Chamzas and Donald L. Duttweiler, entitled "Method and Apparatus for Carry-over Control in Arithmetic Entropy Coding," issued November 27, 1990; (2) U.S. Patent No. 5,025,258 to Donald L. Duttweiler, entitled "Adaptive Probability Estimator for Entropy Encoding/Decoding" issued June 18, 1991; and (3) the Pirsch Patent, regardless of whether a license was actually executed, including patent licenses for each patent and discussions concerning licensing of such patents.

7. Documents authored by AT&T or by any of its employees concerning Image Coding Technology prior to September 11, 1987.

8. AT&T's purchase or knowledge of any Widcom product, including without limitation the Widcom VTC-56 and the Widcom Personal Videoconferencing Station

DEPOSITION TOPICS- STANDARDS RELATED

9. Participation by AT&T, or any employee of AT&T, in any ISO and/or CCITT Standards Bodies prior to February 15, 1994 with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification, including

those documents currently or formerly in the possession of the employees listed in Document Request Number 10 above.

10. Communications between AT&T and any of its employees and the ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies relating to the JPEG Standard and/or MPEG Standards, including those employees listed in Document Request Number 10 above.

11. Participation by Philips, or any employee of Philips, including those employees listed in Document Request 9 above, in any ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies including but not limited to the ISO Working Group 8, ISO Working Group 10, the CCITT Study Group VIII, the CCITT Study Group XV, and/or the joint ISO/CCITT JPEG Committee, prior to February 15, 1994 with respect to Image Coding Technology concerning the JPEG Standard and/or MPEG Standards, including but not limited to any communication, proposal, submission, contribution, and/or technical specification.

12. Communications between Philips and any of its employees and the ISO and/or CCITT Standards Bodies and any subgroup of those Standards Bodies relating to the JPEG Standard and/or MPEG Standards, including those employees listed in Document Request Number 12 above.

EXHIBIT A

United States Patent [19]

Pirsch

[11] **4,420,771**

[45] **Dec. 13, 1983**

[54] **TECHNIQUE FOR ENCODING MULTI-LEVEL SIGNALS**

[75] Inventor: **Peter Pirsch**, Hanover, Fed. Rep. of Germany

[73] Assignee: **Bell Telephone Laboratories, Incorporated**, Murray Hill, N.J.

[21] Appl. No.: **232,701**

[22] Filed: **Feb. 9, 1981**

[51] Int. Cl.³ H04L 3/00; H04N 7/12

[52] U.S. Cl. ..... 358/261; 340/347 DD; 358/133

[58] Field of Search ... 340/347 DD, 347 M, 347 AD; 358/133, 260, 261, 135

[56] **References Cited**

U.S. PATENT DOCUMENTS

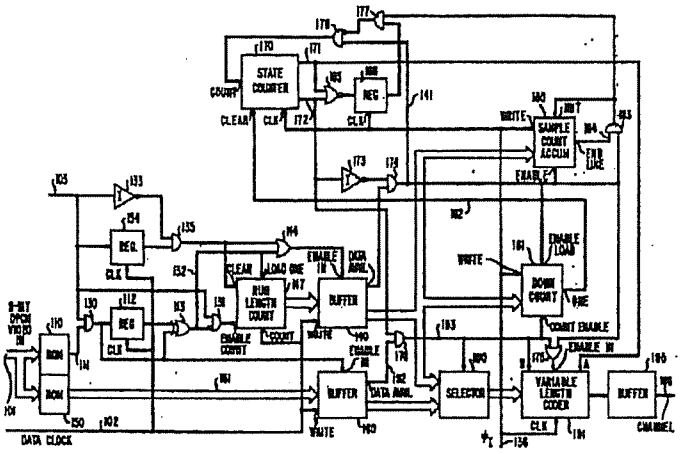
3,483,317	12/1969	De Groat	358/261 X
4,121,259	10/1978	Preuss et al.	358/261
4,163,260	7/1979	Hisao et al.	358/261
4,168,513	9/1979	Haina et al.	358/261

Primary Examiner—T. J. Sloyan
Attorney, Agent, or Firm—Barry H. Freedman

[57] **ABSTRACT**

A series of words each representing one of at least M ($M \geq 3$) different values is encoded by grouping the words into alternate runs of (1) a first, frequently occurring value, and (2) all other values. A coded representation of each run length as well as a representation of the non-frequent values which make up every other run are then combined in a predetermined sequence. Different code tables may be used to generate code words for each type of information.

17 Claims, 3 Drawing Figures



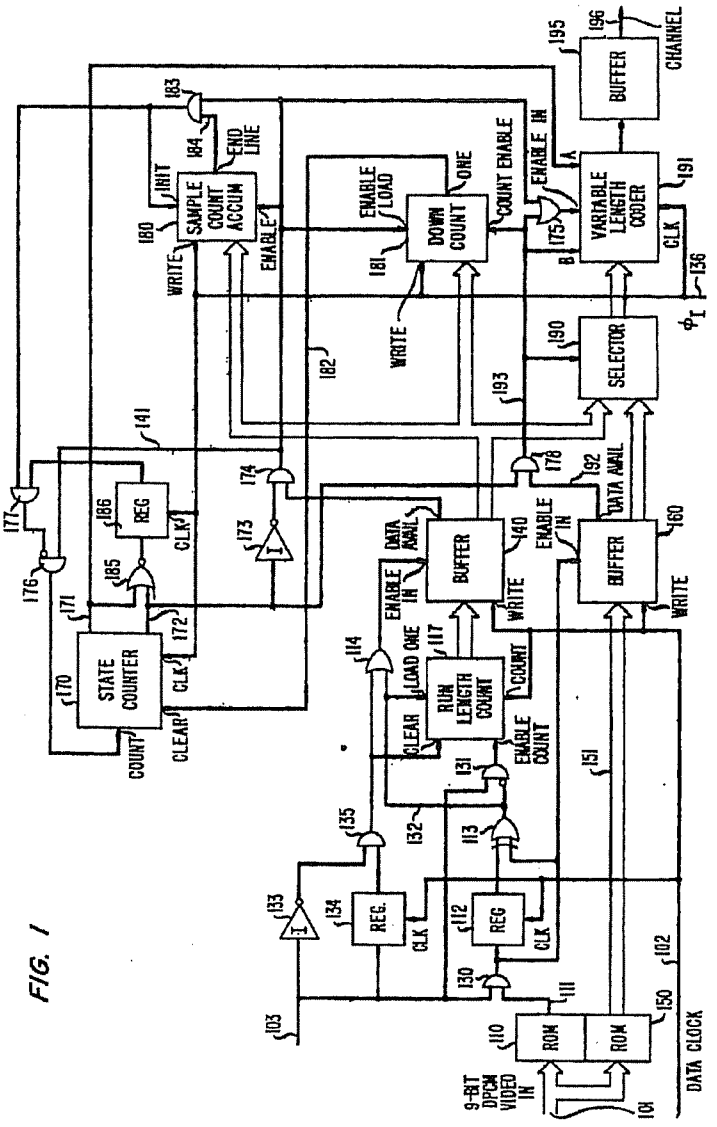

BUFFER
CHANNEL
VARIABLE LENGTH CODER
CLK
SELECTOR
DOWN COUNT
ENABLE LOAD
ONE
COUNT ENABLE
ENABLE IN
SAMPLE COUNT ACCUM
INIT
END LINE
ENABLE
WRITE
DATA AVAIL
BUFFER
REG
CLK
STATE COUNTER
COUNT
CLEAR
LOAD ONE
RUN LENGTH COUNT
REG.
ROM
9-BIT DPCM VIDEO IN
DATA CLOCK

FIG. 1

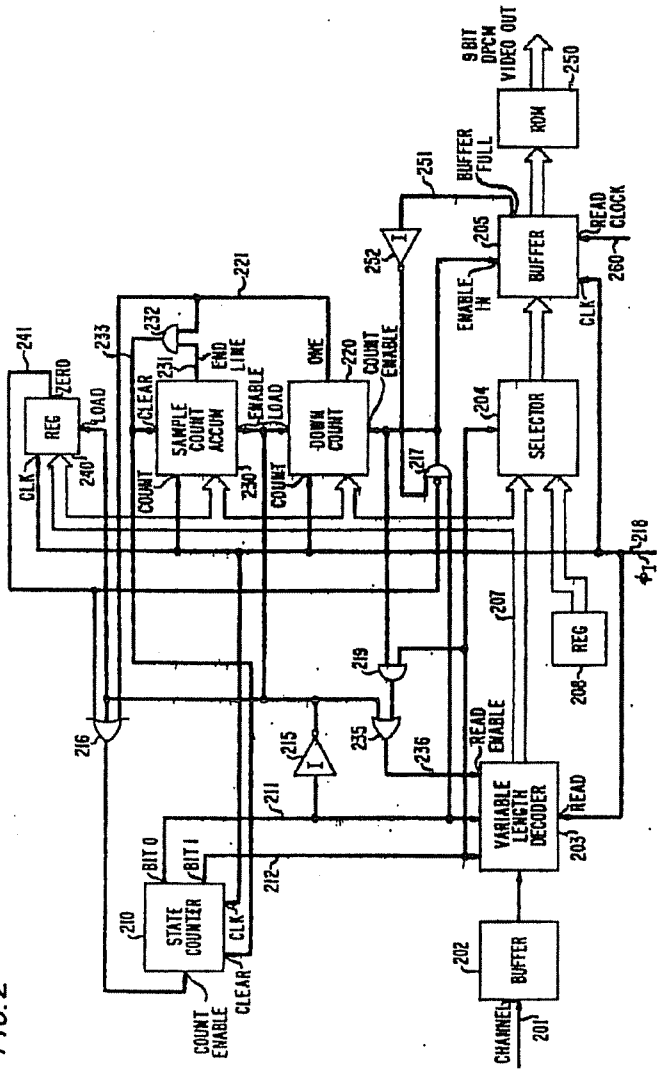

9 BIT DPCM VIDEO OUT
250
ROM
BUFFER FULL
251
READ CLOCK
260
BUFFER
205
ENABLE IN
CLK
252
221
241
233
232
231
END LINE
ONE
220
COUNT ENABLE
REG
ZERO
LOAD
CLEAR
SAMPLE COUNT ACCUM
ENABLE LOAD
DOWN COUNT
SELECTOR
204
CLK
240
COUNT
230
COUNT
217
218
207
REG
208
219
READ ENABLE
216
215
235
236
211
VARIABLE LENGTH DECODER
READ
203
BIT 0
BIT 1
212
210
STATE COUNTER
CLK
CLEAR
COUNT ENABLE
202
BUFFER
CHANNEL
201

FIG. 2

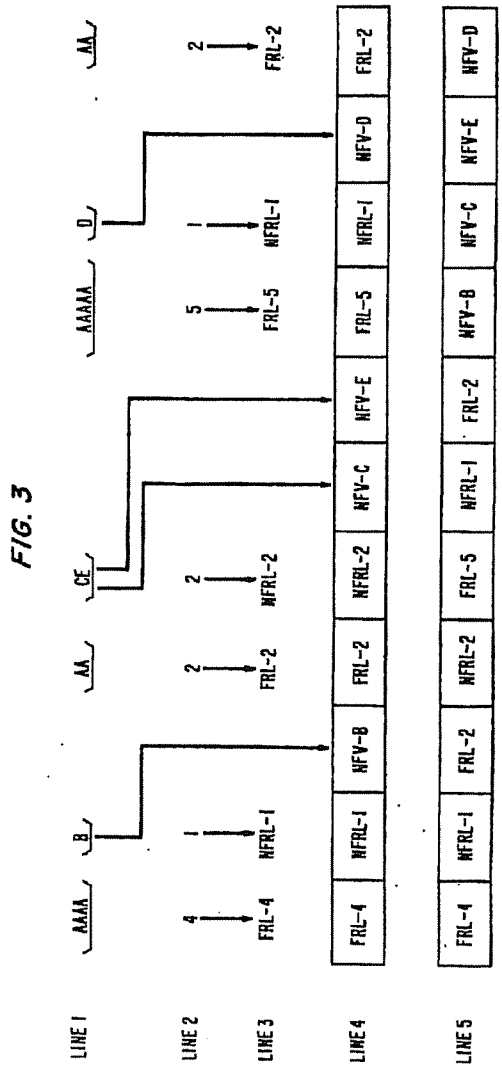

LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
AAAA
B
AA
CE
AAAAA
D
AA
4
FRL-4
1
NFRL-1
2
FRL-2
2
NFRL-2
5
FRL-5
1
NFRL-1
2
FRL-2
FRL-4
NFRL-1
NFV-B
FRL-2
NFRL-2
NFV-C
NFV-E
FRL-5
NFRL-1
NFV-D
FRL-2
FRL-4
NFRL-1
FRL-2
NFRL-2
FRL-5
NFRL-1
FRL-2
NFV-B
NFV-C
NFV-E
NFV-D

FIG. 3

TECHNIQUE FOR ENCODING MULTI-LEVEL SIGNALS

TECHNICAL FIELD

The present invention relates generally to a technique for encoding multi-level signals and, in particular, to a form of run-length coding of such signals when one of the levels occurs much more frequently than all of the other possible levels.

BACKGROUND OF THE INVENTION

Numerous advances have been made in the efficiency with which still or moving picture information may be encoded, based upon the recognition that much information represented in a scene, in graphics or in text, is highly correlated. For example, it is possible to very accurately predict the intensity value (or other characteristics) of a particular portion of a picture, based perhaps upon spatial or temporal relationships, or other known properties of pictures, such as luminance/chrominance relations or by using velocity/displacement compensation. These predictions, in turn, are usually compared with the actual picture signal, so that only the error values need be encoded. With more sophisticated predictions, the error values statistically tend to be smaller, allowing further processing which takes advantage of the increased entropy; numerous techniques have been developed for this purpose. For example, the error signal can be non-uniformly quantized, so that inconsequential values are discarded or de-emphasized and important values emphasized. Since errors usually occur in spurts (near "non-predictable" edges, for example) various adaptive techniques can also be used, depending upon which area of the picture is being processed. Alternatively, the error signal can be processed to take advantage of spatial redundancy by a technique known as bit-plane encoding, wherein one or more of the most significant bits of each error sample, representing a group of contiguous picture elements, are coded as a series of run lengths, rather than individually. However, it is noted that the lower order bits of each error value are not usually highly correlated, reducing the effectiveness of the bit-plane approach.

Despite the improvements which the foregoing techniques provide, it nevertheless remains that the efficiency with which multi-level signals having a statistically "preferred" value are encoded may be increased, by reducing redundancy still further and thus reducing the amount of data needed to represent the signal. Accordingly, the broad object of the present invention is to enable more efficient encoding of a multi-level signal wherein one particular level or value occurs much more frequently than any other value. One such multi-level signal very well suited for encoding in accordance with the present invention is an error signal derived from a predictive picture encoder.

SUMMARY OF THE INVENTION

In accordance with the present invention, a series of words which represent samples of a multi-level signal to be encoded are first processed or grouped so as to form run length code words which indicate the sequence of occurrence of (1) words of a first frequently occurring predetermined value and (2) words of all remaining, less frequent values. The lengths of each type of run, called "frequent value" runs and "non-frequent value" runs, respectively, are preferably encoded using a variable length code. The magnitudes or values of words which make up the "non-frequent value" runs are also encoded, again preferably using variable length coding. However, a different code assignment can be used, if desired. The run-length codes and the non-frequent value codes are then combined in a predetermined sequence for transmission to a remote location where the original signal can be recovered.

BRIEF DESCRIPTION OF THE DRAWING

The present invention will be more fully understood from the following detailed description when read in light of the accompanying drawing in which:

FIG. 1 is a block diagram of an encoder for multi-level input signals constructed in accordance with the present invention;

FIG. 2 is a block diagram of a decoder arranged to recover the original signal encoded by the apparatus of FIG. 1; and

FIG. 3 is a diagram illustrating possible sequences for the code words generated in accordance with the invention.

DETAILED DESCRIPTION

The present invention is, in effect, a hybrid technique allowing the advantages of run-length encoding usually used only in conjunction with binary signals, to be achieved with a multi-level signal. It is most effective when the signal to be encoded contains a significant proportion (say >50%) of words of a single value called the "frequent" value, which statistically is expected to occur most often. All other values are referred to collectively as "non-frequent" values. A particularly suitable application of the invention is in processing the output of a video encoder in which the value of each picture element is predicted using previously encoded information, and in which each prediction is compared with the true value to yield an error value. This type of predictive video encoder produces a large number of predictions which will be accurate, and thus a large number of error values will have values at or near ZERO. These form the "frequent value" inputs to the encoder of the present invention, while remaining error values form the "non-frequent" values. However, it is to be clearly understood that multi-level signals derived from other sources may be processed in accordance with the instant invention, and that advantageous results will be achieved as long as a heavy bias exists in favor of a given most frequent value.

FIG. 1 is a block diagram showing an encoder arranged to encode a multi-level input signal in accordance with the present invention. A multi-level input signal such as a multibit digital signal in parallel format is applied simultaneously on line 101 to first and second read-only memories (ROM's) 110 and 150, which are shown and may be implemented as portions of a signal ROM. A clock pulse is provided on line 102 at the occurence of each input word on line 101, and a blanking signal provided on line 103 remains high during the entire period (e.g., the active portion of each line interval) when input signals are being processed. ROM 110 is arranged to provide a one-bit output on line 11, indicating whether each input word has a "frequent" value, on the one hand, or any of a plurality of remaining possible "non-frequent" values, on the other hand. Continuing the example of an input signal derived from a predictive video encoder, each word may contain 9 bits

which can have only a limited number (say 16) of possible values. This limitation occurs because a quantizer is typically included within the video encoder in order to associate different groups of error values with a small number of corresponding representative values. The nine-bit word length used in the quantizer is one bit longer than the 8 bit word length typical of each input to the video encoder, to provide a sign indication, since each error can be positive or negative. The quantizer output is used in forming succeeding predictions (usually represented as 8 bit words) which in turn are compared with the corresponding 8 bit input word to yield the next 9-bit error value. A particular one of the nine-bit quantizer output words, representing the group of smallest error values, may be the "frequent value" that occurs most often, and ROM 110 may be arranged to generate a "0" (ZERO) in response to this input. On the other hand, all of the remaining 15 possible 9-bit words cause ROM 110 to generate an output of "1" (ONE). indicating a "non-frequent" value.

As long as a high blanking signal is maintained on line 103, the output of ROM 110 on line 111 is applied via AND gate 130 to a one-bit shift register 112 and to the first input of an exclusive OR gate 113 which receives its second input from the output side of register 112. The purpose of gate 113 is to signal the end of each "run" of bits output from ROM 110 by detecting a difference between its present output and the immediately preceding output stored in register 112 at the occurence of each clock pulse on line 102. When the inputs to exclusive OR gate 113 are the same, its low output is applied to an inverting input of AND gate 131, the output of which is high provided that the input on line 103 is high. The output of gate 131 in turn enables a run-length counter 117 to count clock pulses on line 102, so as to accumulate an indication of the length of each run of line values output from ROM 110. Counter 117 must have a capacity sufficient to count the number of words contained in each interval during which the blanking signal is high. For a typical line interval, a 9-bit capacity would be appropriate.

At the transition point between runs, the inputs to exclusive OR gate 113 are different, whereupon the output of gate 113 goes high. A logic signal is thus applied via OR gate 114 to the enable input of a run-length buffer 140 which is now ready to receive the run length stored in counter 117 at the instant that a clock pulse is applied to the write input of the buffer via line 102. The high output of gate 113 on line 132 also causes a "ONE" to be loaded in counter 117, so that the next run of input words may be processed.

When the blanking signal on line 103 goes low, indicating that active input signals are no longer present (at the end of each line, for example), inverter 133 and one-bit register 134 both provide high outputs, enabling AND gate 135. Its output, in turn, is passed through OR gate 114 and used to enable transfer of the multi-bit run length accumulated in counter 117 to buffer 140 when a clock pulse is received at the write input via line 102. The high output of AND gate 135 also clears counter 117 in preparation for the next series of input words.

When the output of ROM 110 is a ONE, indicating an input with a non-frequent value, the high output of AND gate 130 enables a code word output from ROM 150 to be entered in a non-frequent value buffer 160 when a clock pulse is applied to the write input of the buffer via line 102. ROM 150 is arranged to code the input word applied on line 101 by generating a fixed length (say 4-bit) code word on line 151 to represent each possible non-frequent input value. This coding removes the redundancy introduced in the input signal by virtue of the previously described quantization process. However, the code conversion performed in ROM 150 is not an essential feature of the present invention. If an input signal derived from other sources is to be encoded, ROM 150 may be eliminated and the input words applied directly to line 151. In either event, the capacity of buffer 160 must be sufficient to store the number of code words (or input words) expected in each active interval, until the blanking signal on line 103 goes low.

Information in buffers 140 and 160 is read out and encoded, on a first-in/first-out basis, under control of internal timing pulses ϕ_1 input on line 136. These pulses are provided by an internal clock circuit, not shown, which preferably operates at a rate somewhat higher than the data clock on line 102, so that information can be encoded at least as fast as it is received in the encoder of FIG. 1. The protocol or sequence of encoding the run length information in buffer 140 and the non-frequent value information in buffer 160 must be preselected and thus known at both the encoder and decoder. For purposes of explanation, it is assumed that the desired sequence is (1) run lengths of frequent values, followed by (2) run lengths of non-frequent values, followed by (3) the non-frequent values themselves.

Supervision of the encoding sequence is provided by a two-bit state counter 170 which is initialized to provide a low output on both lines 171 and 172 at the beginning of each active interval. In this state, called the 0—0 state, the frequent value run length is to be coded. To accomplish this, the low output on line 172 is inverted in an inverter 173, and used to raise the output of AND gate 174 as long as buffer 140 has available data, as indicated by a high output on line 141. The high output of AND gate 174 serves several purposes: first, a sample count accumulator 180 is enabled to accept and accumulate the run length stored in buffer 140 at the occurrence of a ϕ_1 pulse at its write input. Simultaneously, the same run length is entered in a down counter 181, but this counter is not otherwise used during the 0—0 state. Second, a high logic signal is applied to the "enable input" terminal of a variable length encoder 191 via OR gate 175, so that the run length held in buffer 140 may be passed through a selector switch 190 and applied to that encoder, which will be described more fully hereinafter. Finally, the high output of gate 174 enables the output of AND gate 176 to go high, causing state counter 170 to advance to its next state, called the 0–1 state, in which the run length of non-frequent values is to be encoded.

In the 0–1 state, the output of counter 170 on line 171 is high while its output on line 172 remains low. When the logic level of line 171 is raised, variable length coder 191 is signalled, enabling a change in the coding assignment used to encode the run length information. This aspect of the invention will be discussed more fully below. Since the logic level on line 172 remains low in the 0–1 state, the output of AND gate 174 remains high. In this condition, the length of the non-frequent value run is added to the previous count in accumulator 180 and also entered in down counter 181, overwriting the previous count stored therein. The high output of gate 174 again causes counter 170 to advance, this time to the third state, called the 1–0 state, in which line 172 goes high while line 171 goes low.

In the 1–0 state, the non-frequent value code words stored in buffer 160 are to be encoded. When the logic level on line 172 goes high, the output of AND gate 178 likewise goes high, since at least one non-frequent value code word has been entered in buffer 160 so as to raise the logic level on its data available output (line 192). The high output of gate 178 is passed through OR gate 175 and, as in the 0–1 state, enables coder 191 to receive inputs. However, the high output from gate 178 causes switch 190 to reposition itself so as to apply the non-frequent value codes in buffer 160 to coder 191.

While state counter 170 is in the 1–0 state allowing read out of the non-frequent value codes held in buffer 160, the outputs of AND gates 174 and 176 are low, keeping the counter from changing state. Down counter 181, which has received the run length indicating the number of non-frequent words that must be applied to coder 191 now supervises circuit operation, and the count contained therein is decreased by one each time a ϕ_1 clock pulse is applied on line 136, indicating that another non-frequent value word has been processed. When the count in counter 181 is decreased to ONE, the logic level on line 182 rises, resetting state counter 170 to its 0—0 state with the next ϕ_1 clock pulse, so that the entire cycle described above may be repeated.

Accumulator 180 is used to indicate the end of a processing interval which can include words derived from a single video scan line in the example given above. At the end of such a processing interval, it may be the case that the last information being processed is a frequent value run length; since it is desired that the next processing interval also start in the same state, it is thus necessary to inhibit the advancement of state counter 170. Inhibition is accomplished if counter 170 was most recently in its 0—0 state, since both inputs to NOR gate 185 are then low, producing a high output which sets a register 186 at the next ϕ_1 pulse. This clock pulse advances counter 170 to its 0–1 state. Now, both inputs to AND gate 174 (which is high in the 0–1 state) and the other (on line 184) being high when the count in accumulator 180 indicates that the end of the processing interval has been reached. The high outputs of AND gate 183 and register 186 serve to produce a high output from AND gate 177, which is inverted, keeping the output of AND gate 176 low. This assures that counter 170 will not advance, but will remain in the 0–1 state at the beginning of the next processing interval.

When counter 170 is in a state other than the 0—0 state at the instant when accumulator 180 indicates the end of a processing interval by a high output on line 184, the inhibition mentioned above does not occur. In such case, the outputs of OR gate 185 and thus register 186 are low, so that the output of AND gate 177 is also low. This allows the output of gate 176 to go high and thus advance counter 170 as appropriate.

The output of AND gate 183 is also used to raise the logic level at the "initialize" input to accumulator 180 at the end of each processing interval. In this condition, the next run length output from buffer 140 is loaded directly into accumulator 180, rather than being added to the previous accumulator contents.

As mentioned previously, variable length encoder 191 may comprise a read-only memory capable of storing three complete sets of code words, two for representing the run lengths accumulated in buffer 140 and the third for representing the non-frequent value words accumulated in buffer 160. The particular code set or dictionary used depends upon the type of information (i.e., run length or value) that is being encoded, and for the run lengths, which type of run; i.e., frequent or non-frequent values, is involved. The actual selection of a code table or dictionary depends upon a two-bit code word coupled to coder 191 on lines 171 and 193. For convenience, the bit derived from state counter 170 on line 171 is called the "A" bit, while the bit derived from the output of AND gate 178 on line 193 is called the "B" bit. With bits A and B low, coder 191 is arranged to code frequent value run lengths, while with bit A high and bit B low, non-frequent value run lengths are coded. Finally, with bit B high and bit A low, the non-frequent value words themselves are encoded. For each case, a different code dictionary may be constructed using statistics derived from expected or typical signals. In general, short codes are assigned to most frequent expected values, and longer codes are used for less frequent values. Since the length and rate of the code words output from encoder 191 varies, its output is entered in a first in-first out buffer 195 prior to transmission to a remote location or a storage medium via line 196.

The encoding sequence just described will be more fully appreciated by consideration of the following example. In Table 1 below; a 9-bit multi-level input as might be applied on line 101 is shown in column 1. Each word can assume any of 16 possible values, determined by the characteristics of the quantizer contained in the predictive video encoder which is the source of the input signal. Each non-frequent 9-bit word is quantized or converted to a corresponding 4-bit word in ROM 150, as shown in column 2 of Table 1, using a quantizing strategy which forms no part of the present invention. In this example, input value 000000000 is the frequent value which is not quantized or encoded. This input value causes ROM 110 to produce a ZERO output, while all other input values produce a ONE output from ROM 110, as shown in column 3.

TABLE 1

Input (Line 101)	Output Of ROM 150	Output Of ROM 110	output of Counter 117	Output of Coder 191 Using Table 2	Table 3	Table 4
000000000		0	1	110		
000011001	0010	1				000
000101010	0011	1	2		10	011
000000000		0				
000000000		0				
000000000		0	3	10		
000000101	0001	1				10
000011001	0010	1				000
010130111	1010	1	3		011	01001
000000000		0				
000000000		0	2	000		
000000101	0001	1	1		000	000
000000000		0	1	111		
000000101	0001	1	1		000	10
000000000		0				
000000000		0				
000000000		0				
000000000		0				
000000000		0	5	0101		

The output of ROM 110 is assembled into runs by counter 117, which essentially counts consecutive like outputs from ROM 110. Each multibit word (shown in decimal form in column 4) output from counter 117 is applied to buffer 140. In the example of Table 1, the output of ROM 110 begins with a run of value 0 and

length 1, and is followed by a run of value 1 and length 2. The values of succeeding runs must, by definition, alternate, the run lengths being 3, 3, 2, 1, 1, 1, and 5 as seen from column 4.

The code word assignments used by coder 191 to represent the run lengths in column 4 can be different variable length codes for runs of value ZERO and runs of value ONE, and this arrangement is particularly useful when the run length statistics for the two values are likely to be different. However, the totality of code words used must be such that the receiver can distinguish each word from the next word. Examples of code dictionaries for ZERO and ONE run lengths are given in Tables 2 and 3 below, in which it is seen that the code words are the same but the assignments are different. The appropriate code words determined by using Tables 2 and 3 have been entered in columns 5 and 6 of Table 1. It is to be noted that a code must be provided for a run length of ZERO in Table 2 (for frequent value runs) to account for the possibility that a series of input words will begin with a non-frequent value run. In that event, the first frequent value run is assigned the code representing the ZERO length.

TABLE 2

"ZERO" RUNS

Run Length	Code
0	111
1	110
2	000
3	10
4	011
5	0101
.	
.	
.	

TABLE 3

"ONE" RUNS

Run Length	Code
1	000
2	10
3	011
4	0101
5	110
.	
.	
.	

The code words generated by coder 191 to represent each non-frequent value are also selected using a variable length Huffman code, another example of which is shown in Table 4 below. Appropriate code words for the example in Table 1 generated using Table 4 have been entered in column 7.

TABLE 4

Value	Code
0001	10
0010	000
0011	011
0100	110
0101	111
0110	0101
0111	00100
1000	00101
1001	01000
1010	01001
1011	00110
1100	001110
1101	00111101
1110	00111110

TABLE 4-continued

Value	Code
1111	00111111

From columns 5, 6 and 7 of Table 1, it will be seen that the input words of column 2, which contain a total of 76 bits, are represented by code words having a total of 47 bits, for a compression ratio of approximately 3:2. Where the present invention is used for coding actual error value sequences generated using a predictive video encoder, a compression ratio of approximately 5:2 can be expected. The compression ratio depends upon the type of input sequence.

A multi-valued digital signal encoded with the apparatus of FIG. 1 may be returned to its original form using the decoder shown in block diagram form in FIG. 2. Incoming code words received from the transmission channel 201 are entered in a first-in, first-out buffer 202 which equalizes the transmission channel rate with the processing rate of the decoder circuitry. Data stored in buffer 202 is applied to a variable length decoder 203 which recognizes words from the serial data stream and, like encoder 191 of FIG. 1, contains a read-only memory. Decoder 203 advantageously is arranged to perform run length and value decoding under control from a state counter 210, which insures that the same protocol used in the encoder is used to recover the original signal.

Upon initialization, counter 210 is in the first of four possible states, called the 0—0 state. In this state, the outputs on lines 211 and 212 are both low, and the frequent value run length code word applied to decoder 203 is converted to the corresponding run length output on lines 207 using, for example, the code word to run length conversion in Table 2. The low output from counter 210 on line 211 is inverted by an inverter 215, and applied to the enable inputs of a down counter 220 and a sample count accumulator 230 each of which receives the frequent value run length. A register 240, for purposes described below, is also enabled to load the frequent value run length. The high output of inverter 215 is also passed through OR gate 216 to the count input of state counter 210, advancing it to its second state, called the 0–1 state, upon the ocurrence of a clock pulse ϕ_f applied on line 218. In this state, the output on line 211 is high while the output on line 212 stays low.

Decoder operation is now controlled by down counter 220, which enables assembly of the desired number of words each having the frequent value stored in register 208, in a second buffer 205. Entry into the buffer is enabled by a high output from an AND gate 217. Clocking is again provided by a series of clock pulses ϕ_f applied on line 218, generated by an internal clock not shown, which is also used to synchronize various other circuit elements.

At each occurrence of a ϕ_f pulse on line 218, the count in down counter 220 is decreased by one, provided that the count enable input received from AND gate 217 is high. Concurrently, a word representing the frequent value, permanently held in a register 209, is passed through selector switch 204 and entered in buffer 205. This procedure is repeated until the count in counter 221 decreases to ONE, at which time the last frequent value word is being processed, and the last such word is input to buffer 205.

When the count in counter 220 reaches ONE, a high output on line 221 is passed through OR gate 216, enabling state counter 210 to advance to its third state, called the 1–0 state, in which line 212 is high and line 211 is low. In this state, a non-frequent value run length code word is taken from buffer 202 and applied to variable length decoder 203. The corresponding run length, obtained from Table 3, is entered in down counter 220. Concurrently, the run length on line 207 is added to the count in accumulator 230.

As the run length is entered, the high output of inverter 215 is passed through OR gate 216, causing state counter 210 to advance to its final state (called the 1—1 state) in which both outputs on lines 211 and 212 are high. In the 1—1 state, the code words representing the non-frequent values are output from buffer 202 and applied to decoder 203. The non-frequent values are recovered, using the code conversion of Table 4, and applied to buffer 205 via selector switch 204, which has been repositioned by virtue of the high signal on line 212. Down counter 220 again decrements at each occurrence of an internal clock pulse on line 218, signifying entry of a non-frequent value word into buffer 205. When the count in counter 220 again reaches one, indicating that the last non-frequent value word is about to be processed, the output on line 221 goes high, resetting counter 210 to its initial 0—0 state so that the cycle described above can be repeated.

When the count in accumulator 230 indicates that the number of words processed is equal to the number of words contained in one processing interval, the output on line 231 goes high. This enables the output of AND gate 232 to go high when down counter 220 has completed its cycle and its output on line 221 also goes high. The output from AND gate 232 on line 233 is used to clear state counter 210, whereupon the entire decoding cycle will be repeated for the next processing interval. It is noted that the arrangement described starts counter 210 in its 0–0 state independent of the preceding state just completed. The output of AND gate 232 also serves to clear accumulator 230 in preparation for the next processing interval.

Register 240 is provided in the decoder of FIG. 2 in order to take account of a special situation in which the length of the first frequent value run is zero, i.e., when the first run in a processing interval is a run of non-frequent values. In this case, entry of frequent values into buffer 205 must be inhibited, and state counter 210 must be advanced. When register 240 receives the value ZERO, state counter 210 is advanced to the 0–1 state and the zero output of register 240 on line 241 goes high. This changes the state of counter 210 to the 1–0 state, since the count enable input from OR gate 216 is thus high. The high output on line 241 also serves to disable AND gate 217, preventing data from entering buffer 205 under the conditions specified above.

Variable length decoder 203 is arranged to provide the next output word each time the level on an output enable lead 236 is raised to indicate that another word is needed. Thus, each time a run length code is loaded in counter 220 and accumulator 230 using an enable input from inverter 215, the same signal is passed from inverter 215 through OR gate 235 to provide a high input on line 236. Each time a non-frequent value is entered in buffer 205, enable line 236 is raised by the action of AND gate 219. One input of gate 219 is high in the 1–0 or 1—1 state; its second input is controlled by the output of AND gate 217, which output is also used to enable buffer 205 to receive the frequent or non-frequent value. If buffer 205 becomes full, it is desirable that the operation of the decoder circuit be temporarily halted. Upon such an occurrence, a "buffer full" output on line 251 goes high, whereupon the low output of an inverter 252 acts to disable AND gate 217. Otherwise, the output of gate 217 is high in the 1—1 and 0–1 states, as desired.

The code table or dictionary used by decoder 203 is directly determined by the state of counter 210. In the 0—0 state, frequent value run-length code words are decoded using Table 2, while in the 1–0 state, non-frequent value run length code words are decoded in accordance with Table 3. The non-frequent value codes are decoded using Table 4 when counter 210 is in its 1—1 state. It is to be noted that decoder 203 does not operate in state 0–1, at which time the frequent value word held in register 208 is being entered in buffer 205.

Read-only memory 250, like read-only memory 150 of FIG. 1, is simply a digital code translator arranged, for example, to re-expand the code words output from buffer 205 to match the word length used in the display connected to the decoder output. Continuing the example discussed above, ROM 250 would be arranged to convert each 4-bit word output from buffer 205 to a corresponding 9-bit word to be applied to a DPCM decoder. Data is read out of buffer 205 by clock pulses applied on line 260 at a suitable rate.

Selector switch 204, like switch 190 of FIG. 1, may be a conventional series of semiconductor logic gates arranged to couple a selector and pass to its output one of two multibit input words, depending upon the logic state of a control input. For switch 204, the control input is derived from the output of counter 210 on line 212, and the input on line 207 is coupled to its output when the control input is high. The switch is repositioned to couple the input received from register 208 to its output when the control input is low.

The coding sequence used in the encoder and decoder of FIGS. 1 and 2, respectively, is illustrated in FIG. 3. A series of multibit input words having values A, B, C, D or E is input to the encoder, as shown in line 1. The words in a processing interval are grouped into runs, as shown in line 2, to separate the most frequent value A from all other less frequent values B-E. The run lengths for both frequent and non-frequent values are next coded (line 3), where FRL-n represents a code word for run length "n" derived from a first (frequent value) conversion table, while NFRL-n represents a code word for a run of length "n" derived from a second (non-frequent value) conversion table. Each set of run length codes is followed, as shown, on line 4, by a code word NFV-x, which represents the value of word "x" using yet a third conversion table.

If desired, a different order for the code words on line 4 of FIG. 3 can be used. However, the run length code words which indicate the sequence of occurrence of frequent and non-frequent values in the signal being encoded should usually precede the code words which specify the particular non-frequent values in the signal. For example, all of the run length code words for an entire processing interval can precede the non-frequent value code words for the interval, as shown in line 5 of FIG. 3. In this example, the frequent and non-frequent run length code words alternate, and are decoded at the receiver to give a count of the number of words that have been processed. When codes representing the number of words in each active interval have been

accumulated, the receiver is then switched so as to decode the non-frequent value codes which follow the run length codes for that interval. These changes in the encoding protocol may necessitate corresponding changes in both the encoder and decoder circuitry, such as an increase in the buffer storage capacity. For this reason, the encoding order described above in connection with FIGS. 1 and 2 is preferred.

While the encoder and decoder in the above examples processed an input signal derived from a predictive video encoder, it is to be clearly understood that the present invention is useful with any series of input samples or words capable of having at least M different values, where M≧3. Where a binary (M=2) signal is to be encoded, numerous prior art run length coding techniques are available. To obtain the maximum advantage, the statistical likelihood that one of the M values will occur should be significantly greater than the statistical likelihood that any other of the values will be present. It is also to be observed that a multiplicity of code assignments can be used for the different run lengths and non-frequent values, other than the codes in Tables 2, 3 and 4.

The definition of a "run" stated previously, namely, a series of consecutive inputs having the same value, can also be modified, if desired, without diminishing the advantages of the present invention. Specifically, an alternative definition of a run may be a series of consecutive words of like value as well as the next (subsequent) word of different value. For example, for a binary input of ONE's and ZERO's, a run of ZERO's would include the ONE bit following any group of successive ZERO's, and a run of ONE bits would include the ZERO bit immediately following the successive ONE's. This alternate definition is to be understood to be within the scope of the present invention, and the means used to determine the length of successive runs including exclusive OR gate 113 and run length counter 117 of FIG. 1 would be modified accordingly. If such a different definition were used, variable length encoder 191 and decoder 203 would also be modified appropriately.

What is claimed is:

1. Apparatus for compressing samples of a multilevel communication signal, each sample having one of at least M (M≧3) different values, wherein the statistical likelihood of occurrence of a particular one of said values is much greater than the statistical likelihood of occurrence of any other of said values, including

means for grouping said samples in runs of a first or a second type, wherein said first type runs include one or more samples of said particular value and said second type runs include one or more samples of said other values, and

means for encoding the lengths of said runs and a representation of said other values.

2. The invention defined in claim 1 wherein said encoding means includes means for assigning a code word to each run length of said first type run using a different code dictionary than used to code the run lengths of said second type runs.

3. The invention defined in claim 2 wherein said encoding means is arranged to encode a representation of said other values using a variable length code wherein the shortest code word is associated with the most frequently occurring one of said other values.

4. Apparatus for encoding a series of samples of a multilevel communication signal each having one of at least three different values, including:

means for generating a first series of run length code words which indicate the sequence of occurrence of (1) ones of said samples having a first one of said different values which occurs more frequently than any other of said different values, and (2) other ones of said samples having other ones of said values, and for generating a second series of code words indicating the values of said other ones of said samples, and

means for combining said first and second series of code words.

5. The invention defined in claim 4 wherein said multilevel communication signal is an error signal derived from a predictive encoder.

6. The invention defined in claim 5 wherein run length code words in said first series which represent runs of samples of said first value are derived from a different code table than run length code words which represent samples of said other values.

7. The invention defined in claim 6 wherein said generating means is arranged to generate a variable length code word to represent the value of each of said other ones of said samples, such that the shortest code word is representative of the most frequently occurring one of said samples.

8. Apparatus for encoding a series of samples of a multi-level error signal derived from a predictive encoder, including

means for grouping said samples so as to form alternate runs of (1) samples having the most frequently occurring predetermined value and (2) samples having all remaining, less frequently occurring values, and

means for encoding (a) the length of each of said runs, and (b) the values of only said samples having said remaining, less frequently occurring values.

9. The invention defined in claim 8 wherein said grouping means includes (1) means for generating a one-bit code indicating the occurrence of samples with said predetermined values, and (2) means responsive to said generating means for counting consecutive like occurrences of said one-bit code.

10. A method of encoding samples of a multilevel communication signal, each sample having one of at least M (M≧3) different values, wherein the statistical likelihood of a particular one of said values is much greater than the statistical likelihood of any other of said values, including the steps of:

grouping said samples in runs of a first or a second type, wherein said first type runs include one or more consecutive samples of said particular value and said second type runs include one or more consecutive samples of said other values, and

generating an output signal representing said multilevel signal by encoding the lengths of said runs and a representation of said other values.

11. The method defined in claim 10 wherein said encoding step includes assigning a code word to each run length of said first type run using a different code dictionary than used to code the run lengths of said second type runs.

12. The method defined in claim 11 wherein said encoding step further includes the step of encoding a representation of said other values using a variable length code wherein the shortest code word is associ-

ated with the most frequently occurring one of said other values.

13. The method of encoding a series of samples of a multilevel communication signal, each sample having one of at least three different values, including the steps of:

generating a first series of run length code words which indicate the sequence of occurrence of (1) ones of said multilevel samples having a first one of said different values which occurs more frequently than any other of said different values, and (2) other ones of said samples, and generating a second series of code words indicating the values represented by said other ones of said samples, and

generating an output signal by combining said first and second series of code words.

14. The method defined in claim 13 wherein said multilevel signal is an error signal derived from a predictive encoder.

15. A method of encoding a series of words which represent samples of a multi-level signal representing pictorial information, including the steps of:

grouping said words so as to form alternate runs of (1) words having the most frequently occurring predetermined value and (2) words having all remaining, less frequently occurring value, and

generating an output signal which represents said pictorial information by encoding (a) the length of each of said runs, and (b) the values of only said words having said remaining, less frequently occurring values.

16. The invention defined in claim 1 wherein said multilevel communication signal is an error signal derived from a predictive encoder.

17. The method defined in claim 10 wherein said multilevel communication signal is an error signal derived from a predictive encoder.

* * * * *

EXHIBIT B

Copyright © 1982 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 5, May–June 1982
Printed in U.S.A.

Adaptive Intra-Interframe DPCM Coder

By P. PIRSCH

(Manuscript received August 4, 1981)

Adaptive prediction schemes provide lower transmission rates than those obtained by simple previous frame prediction. In this paper, we measure the entropy of prediction errors for two types of adaptive intra-interframe prediction algorithms. In the first case, that predictor which results in the least prediction error for previously transmitted neighboring pels is selected from a set of predictor functions. In the second case, prediction is a weighted sum of previous frame and intraframe predictions, where the weights are changed from pel to pel by gradient techniques. We also investigate various modifications of the basic methods. Further, a new type of variable length encoding in which the locations of the nonzero prediction errors are coded by horizontal run lengths is discussed. Compared with the pel entropy of previous frame prediction, the run length coding gives a gain of 2 to 16 percent, depending on the scene. Compared to simple previous frame prediction the first type of adaptive scheme in combination with horizontal run length coding provides a gain in entropy of 18 to 29 percent, whereas the second type of adaptive scheme provides a gain of 20 to 32 percent.

I. INTRODUCTION

The bit rate required for digital transmission of television pictures can be significantly reduced by interframe DPCM encoding. The coding method which has been widely proposed for video-telephone and video-conference application is conditional replenishment.[1,2] In conditional replenishment, each frame of a television sequence is segmented into changed and unchanged areas. Various methods can be used for encoding the changed parts of a frame. Intraframe predictive coding is very efficient for these parts.[3,4] In conditional replenishment, no information about the unchanged areas is transmitted. At the receiver, the unchanged areas are reconstructed by repeating from the previous frame. However, it is necessary to transmit address information that

indicates the location of the changed areas. Several modifications and improvements of the basic method of conditional replenishment have been made. Most by them are described in a survey by Haskell.[6]

This paper describes adaptive intra-interframe prediction. It is obvious that stationary background of a frame is best predicted from a pel in the previous frame which has the same position as the pel to be predicted, whereas parts of a frame with moving objects are better predicted by an intraframe predictor. Therefore, a prediction scheme which provides automatic switching between the two types of predictors, depending upon the part of the picture, will result in lower bit rates. To avoid the transmission of additional predictor control information, the adaptive prediction schemes described here are based on previously transmitted reconstructed pels. Further, no forward segmenter like that of conditional replenishment is used. Therefore, only the quantized prediction error has to be coded and transmitted.

A block diagram of such a DPCM encoder with adaptive prediction is shown in Fig. 1. The investigations in this paper concern a comparison of the performance of two types of adaptive predictors. The first one, denoted by predictor selection, is a scheme where one predictor is selected from a set of predictors. In the second scheme, the predictor is a weighted sum of predictors and the prediction coefficients are changed continuously by a gradient algorithm. As a measure of predictor performance, the entropy of the quantized prediction error is used. For three different television scenes an estimate of the entropy is obtained from DPCM simulations. The necessary measures against buffer overflow and underflow, in case of variable length encoding, have not been considered here.

This paper is organized as follows. Section II gives a detailed description of the two basic algorithms and their modifications. Section III describes a variable length encoding scheme which is especially suited for DPCM coders that have improved prediction. Results of simulations on real scenes are given in Section IV.

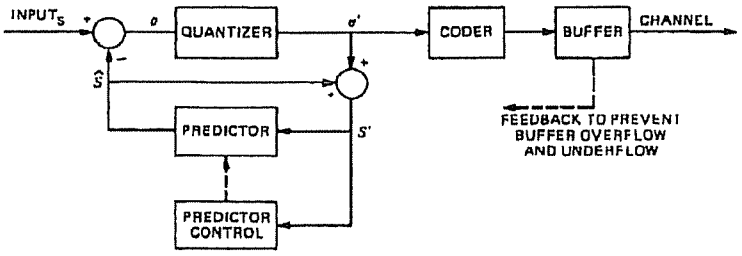
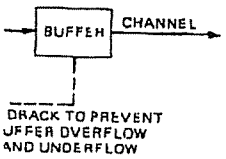


Fig. 1—Block diagram of a DPCM coder with adaptive predictor.

eral modifications and
al replenishment have
rvey by Haskell.
ie prediction. It is ob-
best predicted from a
sition as the pel to be
ing objects are better
, a prediction scheme
e two types of predic-
ill result in lower bit
redictor control infor
ied here are based on
ther, no forward seg-
used. Therefore, only
nd transmitted.
adaptive prediction is
concern a comparison
lictors. The first one,
iere one predictor is
scheme, the predictor
ction coefficients are
As a measure of pre-
d prediction error is
imate of the entropy
ıry measures against
ble length encoding,

II gives a detailed
iodifications. Section
which is especially
ediction. Results of



ve predictor.

II. DESCRIPTION OF THE PREDICTION ALGORITHMS

Let f_i be one of M predictor functions. If each f_i is a linear predictor function, then

$$f_i = \sum_{j=1}^{N} a_{ij} s'_j, \tag{1}$$

where a_{ij} are the weighting coefficients and s'_j are previously transmitted pels. The prime in s'_j indicates that these are reconstructed pels which are known at the receiver. The subscripting for pels neighboring the present pel s_0 is shown in Fig. 2. The predictor functions f_i, $i = 1, 2, \cdots M$, are linear combinations of N pels, s'_j, $j = 1, 2, \cdots N$, which form a vector

$$\mathbf{s}' = \begin{vmatrix} s'_1 \\ s'_2 \\ \vdots \\ s'_N \end{vmatrix}. \tag{2}$$

In vector notation, equation (1) can be written as

$$f_i = \mathbf{a}_i^T \mathbf{s}'. \tag{3}$$

Here the superscript T denotes the transpose of a vector or matrix, and $\mathbf{a}_i$ is the vector formed by the coefficients a_{ij}, $j = 1, 2, \cdots N$. The prediction value $\hat{s}_0$ is a weighted sum of all predictor functions,

$$\hat{s}_0 = \sum_{i=1}^{M} b_i f_i. \tag{4}$$

If $\mathbf{f}$ denotes the vector of elements f_i, $i = 1, 2, \cdots M$, and $\mathbf{b}$ denotes the vector of elements b_i, $i = 1, 2, \cdots M$, then

$$\hat{s}_0 = \mathbf{b}^T \mathbf{f}. \tag{5}$$

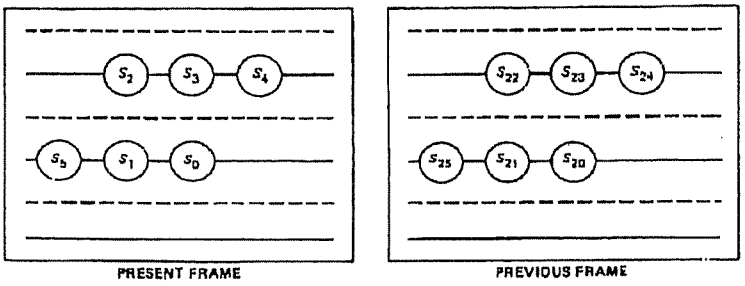


Fig. 2—Configuration and subscripting of picture elements. Pel s_0 is the present pel to be predicted. Dotted lines denote scan lines from previous fields.

This description is general and includes switched prediction by allowing special values of $\mathbf{b}$, such as $b_k = 1$ and $b_i = 0$ for all $i \neq k$. Combining (3) and (5) it follows that

$$\mathbf{f} = \mathbf{A}\mathbf{s}' \tag{6}$$

$$\hat{s}_0 = \mathbf{b}^T\mathbf{A}\mathbf{s}', \tag{7}$$

where

$$\mathbf{A} = \begin{vmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_M^T \end{vmatrix} \tag{8}$$

is an $M \times N$ matrix. The set of predictor functions is described by the matrix $\mathbf{A}$, with the coefficient vectors $\mathbf{a}_i$ chosen such that a particular predictor function provides a good prediction for a specific area of a television scene, like stationary background, moving objects, etc. The algorithm then seeks to automatically adapt the vector $\mathbf{b}$ to various areas of a scene so as to minimize the prediction error.

In this investigation, the set of predictor functions is restricted to a previous frame predictor

$$f_1 = s'_{20} \tag{9}$$

and an intraframe predictor

$$f_2 = a_{21}s'_1 + a_{22}s'_2 + a_{23}s'_3. \tag{10}$$

The following prediction algorithms are described for two predictor functions, but they can easily extend to more than two.

2.1 Predictor selection schemes

From a set of predictor functions, the predictor which results in the least prediction error for previously transmitted neighboring pels is selected as the predictor for the present pel. For each predictor function, a decision function u_i is defined, which is the sum of the amount of the prediction errors for each pel in a small window of neighboring pels. The predictor which has the smallest value for the decision function is chosen as predictor. This criterion was also used by Stuller et al.[5] for gain and displacement compensation. The basic selection rule for two predictor functions is as follows:

$$\hat{s} = \begin{cases} f_1 & \text{if} \quad u_1 \leq u_2 \\ f_2 & \text{if} \quad u_1 > u_2, \end{cases} \tag{11}$$

where

$$u_i = \sum_{k \in W} |s'_k - f_i(s'_k)|. \tag{12}$$

ıed prediction by allow-
for all $i \neq k$. Combining

(6)

(7)

(8)

ions is described by the
ı such that a particular
for a specific area of a
oving objects, etc. The
he vector b to various
ın error.
ctions is restricted to a

(9)

(10)

ibed for two predictor
ıan two.

or which results in the
ed neighboring pels is
l. For each predictor
ich is the sum of the
in a small window of
smallest value for the
riterion was also used
npensation. The basic
ollows:

(11)

(12)

MAY–JUNE 1982

The subscript k denotes a pel in a neighborhood W. The window W is chosen such that s'_k is known at the receiver. The decision function u_i can be evaluated at the receiver without transmission of additional information about predictor selection. For previously transmitted nearest neighbors, the index set W is

$$W_n = \{1, 2, 3, 4\}. \tag{13}$$

For real-time implementation the choice of W_n creates problems, because of the use of the pel s'_1. The time constraint for calculation of u_i can be reduced by using the index set

$$W_\beta = \{2, 3, 4, 5\} \tag{14}$$

or

$$W_\gamma = \{2, 3, 4\} \tag{15}$$

instead of W_n. The window W_γ is also used by Stuller et al.[6]

A further simplification for hardware implementation can be obtained by introducing a quantizer function $Q_s[\cdot]$ in (12). Then the decision functions u_i are given by

$$u_i = \sum_{k \in W} Q_s \quad [|s'_k - f_i(s'_k)|]. \tag{16}$$

A modification which leads to a simpler implementation than the basic selection rule (11) can be described as follows. Choose the predictor function f_i which has within a window W most frequent minimum magnitude of the difference

$$d_{ik} = s'_k - f_i(s'_k). \tag{17}$$

In the case of two predictor functions at each position k, a binary variable v_k which describes which predictor function is better, is defined as follows,

$$v_k = \begin{cases} 1 & |d_{1k}| \leq |d_{2k}| \\ 0 & |d_{1k}| > |d_{2k}|. \end{cases} \tag{18}$$

The decision functions u_i are now given by

$$u_1 = \sum_{k \in W} v_k$$

$$u_2 = \sum_{k \in W} \bar{v}_k, \tag{19}$$

where $\bar{v}_k$ is the complement of v_k. The predictor with smallest value u_i is chosen. The selection rules as discussed above require that one predictor function be chosen even if both decision functions u_i are identical. An improvement can be obtained by using a "soft-predictor

switch," i.e., the prediction value is a weighted sum of predictor functions, as given by (4), with the weights b_i being proportional to the frequency of preference of the predictor function f_i. Hence, for two predictor functions,

$$b_1 = \frac{1}{n} \sum_{k \in W} v_k$$

$$b_2 = \frac{1}{n} \sum_{k \in W} \bar{v}_k, \tag{20}$$

where n is the number of pels in the window W. To avoid the division by 3, for $W = W_n$ the contribution of the pel at position 3 to (20) is doubled, and n is chosen to be 4 for this special case.

2.2 Adaptive prediction based on a steepest descent method

The steepest descent[7] is a mathematical method which has been often used for optimization. One advantage of this method is its simplicity. This method has been used frequently for adaptive systems. It is also proposed by Netravali and Robbins[8] and Stuller et al.[9] for motion-compensated prediction. Here it will be applied to adaptive intra-interframe prediction.

Let us assume that the prediction value is a weighted sum of predictor functions as given by (5). Then the prediction error is given by

$$e = s - \mathbf{b}^T\mathbf{f}. \tag{21}$$

In the following, the present pel is denoted by s, rather than s_0. The variance of the prediction error e is a quadratic function in $\mathbf{b}$.

$$F(\mathbf{b}) = E[(s - \mathbf{b}^T\mathbf{f})^2], \tag{22}$$

where $E[\cdot]$ is the expected value. The gradient with respect to $\mathbf{b}$ is given by

$$\mathbf{g} = \nabla_b F(\mathbf{b}) = -2E[(s - \mathbf{b}^T\mathbf{f})\mathbf{f}] \tag{23}$$

$$= -2E[e\mathbf{f}].$$

The steepest descent is an iterative method, where starting from an initial guess the vector $\mathbf{b}$ is modified recursively according to,

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - \gamma^{(m)}\mathbf{g}^{(m)}, \tag{24}$$

The adjustment of the vector $\mathbf{b}^{(m)}$ is made in the direction of the negative gradient. The scalar $\gamma^{(m)}$ has to be optimized by a one-dimensional search scheme at each step m. However, real-time appli-

cations are performed with a constant value of γ. The best value of γ depends on the type of data. In addition, the value of γ influences the stability and the speed of convergence of $\mathbf{b}$.

From eqs. (23) and (24), it follows that an adaptive prediction scheme which, based on a gradient method, is given by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} + 2\gamma E_w[\mathbf{ef}]^{(m)}, \tag{25}$$

where $E_w[\cdot]$ is the expected value within a small window of neighboring pels as given by (13), (14), or (15). The coefficient vector $\mathbf{b}$ is updated on a pel-by-pel basis along the scanning direction, i.e., if $\mathbf{b}^{(m+1)}$ is the coefficient vector at the present pel, $\mathbf{b}^{(m)}$ is the coefficient vector at the previous pel. At the beginning of each line, an initial estimate of $\mathbf{b}$ is used, e.g., the mean of $\mathbf{b}$ at the previous line. Simulations indicate that because of a fast adjustment an initial vector $\mathbf{b}$ with elements $b_i = 1/M$, $i = 1, 2, \cdots M$ is appropriate.

In this study, several modifications of the recursion given by (25) have been investigated. The various algorithms will be compared with respect to prediction gain and cost of implementation. A high prediction gain requires an appropriate value of γ in (25). Simulations with several values of γ indicate that for video signals with normalized range [0, 1] the optimum value of γ is about one. In such a case the adjustment from pel to pel is relatively small, and the transition from one predictor function to another takes several pels. By introducing an additional constraint

$$\sum_{j=1}^{M} b_j = 1, \tag{26}$$

the value of optimum γ is increased to about 64. The increased value of γ provides a shorter transition from one predictor function to another and the constraint (26) improves the stability of the algorithm.

With the constraint of (26), the steepest descent method has to be modified to minimize the augmented function of (22)

$$\Phi(\mathbf{b}, \lambda) = E[(s - \mathbf{b}^T\mathbf{f})^2] + \lambda(\mathbf{b}^T\mathbf{o} - 1), \tag{27}$$

where $\mathbf{o}$ is a vector with all elements equal to 1. The coefficient vector $\mathbf{b}$ is updated recursively by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - \gamma(-2E[\mathbf{ef}]^{(m)} + \lambda^{(m)}\mathbf{o}). \tag{28}$$

Using (26) to eliminate $\lambda^{(m)}$ from (28), and replacing $E[\cdot]$ by $E_w[\cdot]$, then

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} + 2\gamma\,\mathbf{C}\,E_w[\mathbf{ef}], \tag{29}$$

where $\mathbf{C}$ is an $M \times M$ matrix given by

$$\mathbf{C} = \mathbf{U} - \frac{1}{M}\mathbf{o}\mathbf{o}^T = \begin{vmatrix} 1-\frac{1}{M} & -\frac{1}{M} & \cdots & -\frac{1}{M} \\ -\frac{1}{M} & 1-\frac{1}{M} & \cdots & -\frac{1}{M} \\ \vdots & \vdots & & \\ -\frac{1}{M} & -\frac{1}{M} & \cdots & 1-\frac{1}{M} \end{vmatrix} \tag{30}$$

and $\mathbf{U}$ is the unit matrix. Because of (26), it follows that

$$s - \mathbf{b}^T\mathbf{f} = \mathbf{b}^T(s\mathbf{o} - \mathbf{f}) = \mathbf{b}^T\mathbf{d}, \tag{31}$$

where $\mathbf{d}$ is a vector of differences similar to (17). This leads to an equivalent recursion of (29), given by

$$\mathbf{b}^{(m+1)} = \mathbf{b}^{(m)} - 2\gamma\mathbf{C}E_W[e\mathbf{d}]. \tag{32}$$

In the recursions given above, the coefficient vector $\mathbf{b}$ at the previous pel is updated by an adjustment to obtain the coefficient vector at the present pel. However, a picture is two-dimensional in nature, the values of $\mathbf{b}$ for pels from the previous line in the immediate neighborhood of the present pel are quite close to that of the present pel. This idea results in a modification of (25) which is given below.

$$\mathbf{b}^{(m+1)} = E_W[\mathbf{b}]^{(m)} + 2\gamma\, E_U[e\mathbf{f}]^{(m)}. \tag{33}$$

Let us assume that the samples s and the predictor functions f_i are represented by 8 bits. In such cases, in the recursions given above at each position within the window, a product of two 8-bit numbers has to be calculated. A reduction in the cost of implementation can be achieved by using the three-level quantizer, shown in Fig. 3, for the prediction error e and the differences d. These investigations show that a three-level quantizer with a dead zone is more advantageous than the signum function used by Netravali and Robbins.[8]

The algorithm (29) and (33) for the case of two predictor functions, in combination with a three-level quantizer Q_D, results in the following recursive scheme,

$$\begin{aligned} b_1^{(m+1)} &= E_W[b_1]^{(m)} + \gamma E_W[Q_D(e)Q_D(f_1 - f_2)]^{(m)} \\ b_2^{(m+1)} &= E_W[b_2]^{(m)} - \gamma E_W[Q_D(e)Q_D(f_1 - f_2)]^{(m)}, \end{aligned} \tag{34}$$

with the constraints

$$\begin{aligned} b_1 + b_2 &= 1 \\ 0 &\le b_1 \\ 0 &\le b_2. \end{aligned} \tag{35}$$

$$\cdots -\frac{1}{M} \quad \cdots -\frac{1}{M} \qquad (30)$$

$$\cdots 1-\frac{1}{M}$$

lows that

d. (31)

(17). This leads to an

]. (32)

ector b at the previous
oefficient vector at the
nsional in nature, the
e immediate neighbor-
of the present pel. This
ven below.

$$']^{(m)}. \qquad (33)$$

dictor functions f_i are
ursions given above ar
two 8-bit numbers has
mplementation can be
own in Fig. 3, for the
se investigations show
is more advantageous
l Robbins."
vo predictor functions,
results in the following

$$f_1 - f_2)]^{(m)}$$
$$f_1 - f_2)]^{(m)}, \qquad (34)$$

(35)

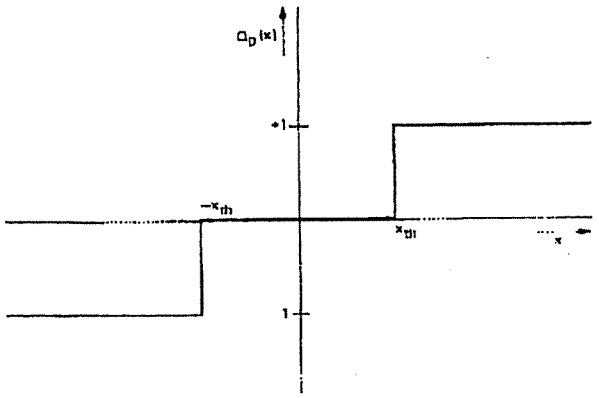


Fig. 3—Three-level quantizer for gradient quantization.

The latter two constraints of (35) were introduced to avoid negative weighting coefficients.

III. VARIABLE LENGTH ENCODING BY HORIZONTAL RUN LENGTH

An adaptive prediction scheme leads to many predictable pels. A pel is described as predictable if its quantized prediction error is represented by the level zero. To obtain a low transmission rate, the quantized prediction error is coded by a variable length code. There is always a loss in mean transmission rate compared to the entropy if not all of the negative logarithm of the probability of the prediction error representative levels are integer. This loss is especially high if one level has a probability much larger than 0.5. For adaptive prediction schemes, this is true for the quantizer level zero. To overcome this problem, block coding is frequently used. For the application described, a special coding scheme is proposed.

From each frame, a two-level picture is generated which indicates where the pels with zero code words (ZCW) and where the pels with nonzero code words (NZCW) are located. This new picture can be coded by known one-dimensional and two-dimensional coding techniques for two-level pictures. The NZCWs are coded in parallel by a variable-length code like a Huffman code and multiplexed with the code words of the two-level picture such that the receiver can decide between the two types of data. A block diagram of such a coder is shown in Fig. 4.

For a horizontal run length code, the set of symbols to be coded is listed in Fig. 5. For each of the sets, i.e., zero runs (ZR), nonzero runs (NZR) and nonzero code words (NZCW), a variable length code can be determined independently and matched to the probability of the

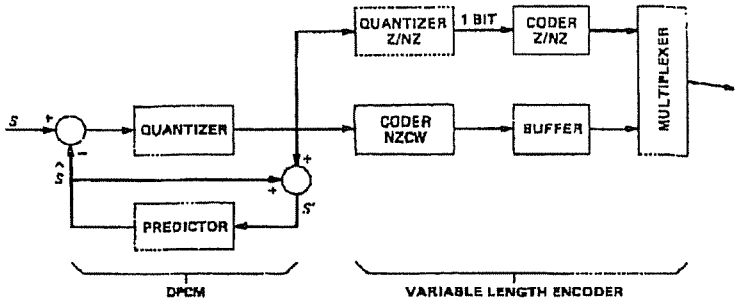


Fig. 4—Block diagram of a new type of variable length encoding.

REGULAR SET OF CODE WORDS

$\{0, 1, 2, \ldots, k\}$

NEW SETS FOR CODING

(i) Set of nonzero code words (NZCW)

$\{1, 2, \ldots, k\}$

(ii) Set of zero runs (ZR)

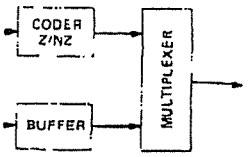
i	ZR
0	1
1	01
2	001
3	0001
⋮	⋮ ⋮ ⋱
n	0000...01
$n+1$	0000...00

(iii) Set of nonzero runs (NZR)

i	NZR
0	0
1	10
2	110
3	1110
⋮	⋮ ⋮ ⋱
m	1111...10
$m+1$	1111...11

Fig. 5—Set of symbols for horizontal run length coding.

symbols of that particular set (e.g., Huffman code). The type of runs are chosen so as to allow a wrap-around coding from line to line. Wrap-around coding means that a run is not terminated at the end of a line but continued in the next line. Furthermore, the longest run to be



NGTH ENCODER

length encoding.

›

›gth coding.

·de). The type of runs
·om line to line. Wrap-
ed at the end of a line
.he longest run to be

MAY–JUNE 1982

coded could be shorter than one line. The code words must be transmitted in a sequence so that the receiver always knows which code table must be used for decoding. Fig. 6 gives an example in which a ZR is transmitted at the beginning of a line. In this example, it is also assumed that the NZCWs are transmitted just after the corresponding run.

The entropy

$$H = -\sum_i p_i \log p_i \tag{36}$$

is used as an estimate for the mean code word length, with p_i being the relative frequency of the ith code word derived from the DPCM simulation of a TV sequence. The variable length code described above consists of three independent codes. Hence, the entropy H_{RUN} in bits per sample is given by

$$H_{RUN} = \frac{n_{NZCW}}{n_{PEL}} H_{NZCW} + \frac{n_{ZR}}{n_{PEL}} H_{ZR} + \frac{n_{NZR}}{n_{PEL}} H_{NZR}, \tag{37}$$

where n is the number of events specified by the subscript.

An advantage of the type of run length coding presented here is that in the case of statistically independent symbols, the overall entropy is not changed ($H_{PEL} = H_{RUN}$). In the case of interframe coding, the zeros and nonzeros are grouped together because they are related to the picture content. In this case, a decrease in entropy is achieved by the horizontal run length coding.

IV. SIMULATION RESULTS

Computer simulations were performed for the prediction algorithms given above using three different television sequences. These sequences are the same as those used in Refs. 6 and 8. Each sequence

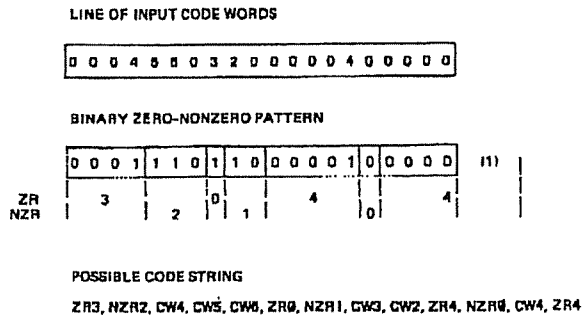


Fig. 6—Example of a horizontal run length code.

consists of 60 frames obtained by sampling a video signal of 1-MHz bandwidth, at the Nyquist rate. Each sample was quantized to 8 bits. One frame of each sequence is shown in Fig. 7.

One scene, called Judy, is a head-and-shoulders view of a person engaged in active conversation. The second scene, John and Mike, shows two people entering the camera field of view and walking briskly around each other. The third sequence, Mike and Nadine, is a panned view of two people always in view of the camera.

Even though the quantizer characteristic of a DPCM coder should be designed according to the prediction scheme, for simplification in these investigations, the same 35-level quantizer shown below was used for all simulations. The quantizer has the following positive representative levels: 0, 5, 12, 19, 28, 37, 46, 57, 68, 79, 90, 103, 116, 129, 142, 155, 168, 181. This quantizer was chosen since it gave good picture quality, although the quantization error was visible in specific picture areas under short viewing distance. The decision levels are always in the middle between two succeeding levels. The performance of the predic-

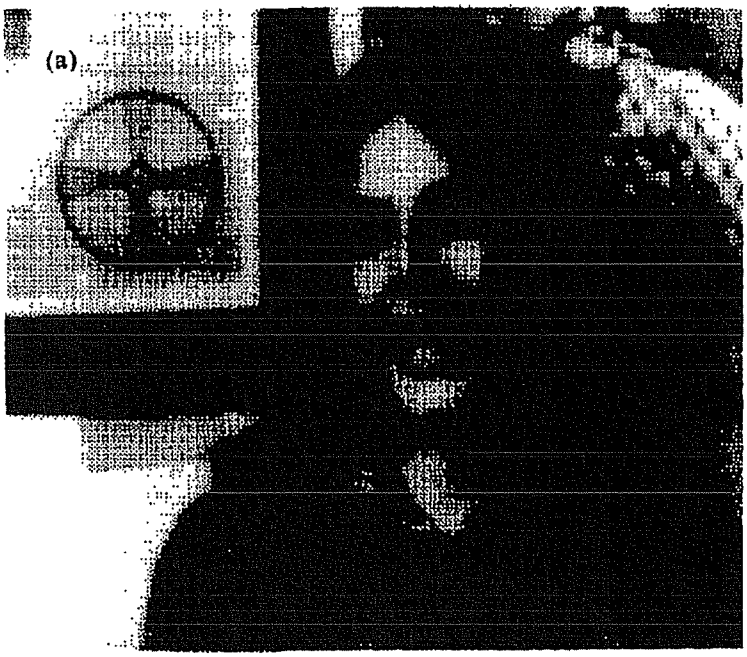
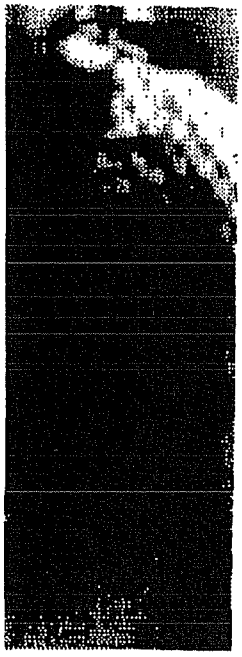


Fig. 7a One frame out of each sequence—Scene Judy.

video signal of 1-MHz
as quantized to 8 bits.

ders view of a person
cene, John and Mike,
ew and walking briskly
id Nadine, is a panned
.

DPCM coder should be
simplification in these
vn below was used for
positive representative
116, 129, 142, 155, 168,
good picture quality,
specific picture areas
els are always in the
ormance of the predic-



cene Judy

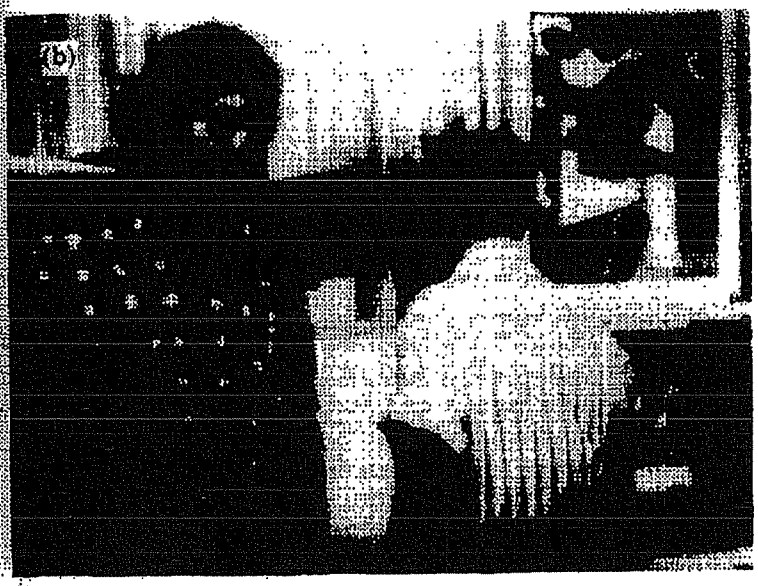


Fig. 7b One frame out of each sequence—Scene John and Mike.

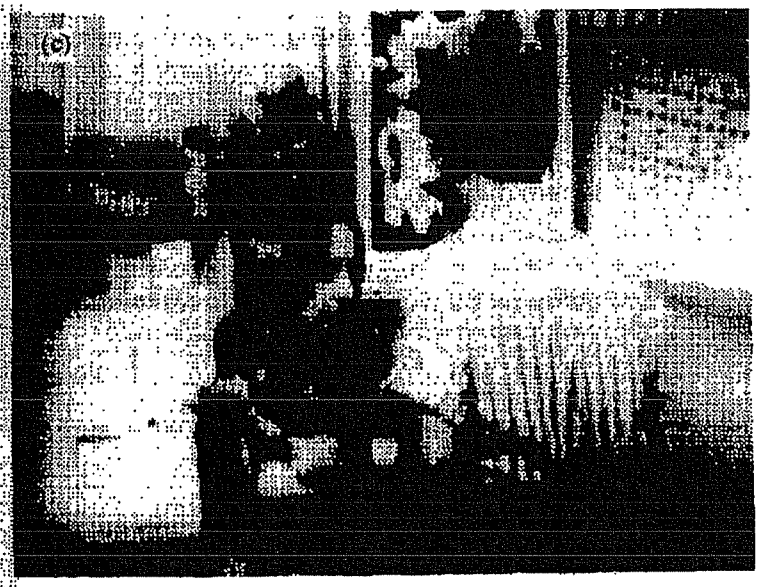


Fig. 7c One frame out of each sequence—Scene Mike and Nadine.

tion schemes was evaluated by computing the pel entropy, the entropy of a horizontal run length code, and the variance of the quantized prediction error.

For comparison of adaptive and nonadaptive schemes, results for four nonadaptive predictors were obtained. The nonadaptive prediction schemes which were used are given below.

$$\hat{s} = s'_{20} \tag{38}$$

$$\hat{s} = s'_1 - s'_{21} + s'_{20} \tag{39}$$

$$\hat{s} = \frac{3}{4}s'_1 - \frac{2}{4}s'_2 + \frac{3}{4}s'_3 + \frac{3}{4}s'_{20} - \frac{2}{4}s'_{21} + \frac{1}{4}s'_{22} - \frac{2}{4}s'_{23} \tag{40}$$

$$\hat{s} = \frac{7}{8}s'_1 - \frac{5}{8}s'_2 + \frac{6}{8}s'_3. \tag{41}$$

The first predictor (38) is simple previous frame prediction. The prediction scheme given by (39) is frequently proposed for interframe coding.[8,9] The predictor (40) is a three-dimensional predictor proposed by Klie[10] for moving areas of a picture. Equation (41) describes an intraframe predictor which minimizes the variance of the prediction error.[11]

The results of the nonadaptive predictors are shown in the upper part of Tables Ia, b, and c. These investigations show that previous frame prediction (38) is advantageous for sequences with not much motion (Judy), while the intraframe predictor (41) and the predictor (40) are better for sequences with rapidly moving objects (Mike and Nadine). An additional decrease in entropy can be obtained by using the horizontal run length coding scheme. This gain is especially high (16 percent) for the sequence Judy where ZR and NZR are better grouped.

Table Ia—Entropy per pel and variance of the prediction error for nonadaptive and adaptive predictors—Scene Judy.

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
1.035	0.875	16.6	Previous frame, eq. (38)
1.120	0.963	8.5	2-D Interframe, eq. (39)
1.349	1.297	9.1	3-D Interframe, eq. (40)
1.840	1.760	31.7	2-D Intraframe, eq. (41)
0.808	0.765	5.3	Predictor selection, eq. (11), (12), W_α
0.781	0.718	4.8	Predictor selection with soft switch eq. (18), (20), W_α
0.783	0.730	4.9	Gradient algorithm, eq. (34), W_α

el entropy, the entropy
iance of the quantized

ve schemes, results fo
he nonadaptive predic

(38)

(39)

$$+\frac{1}{4}s'_{22}-\frac{2}{4}s'_{2a} \qquad (40)$$

(41)

frame prediction. The
roposed for interframe
nal predictor proposed
tion (41) describes an
ance of the prediction

re shown in the upper
ns show that previous
uences with not much
(41) and the predictor
ing objects (Mike and
1 be obtained by using
gain is especially high
R and NZR are better

e prediction error for
–Scene Judy.

tion Scheme
(38)
(39)
(40)
(41)
eq (11), (12), W_0
with soft switch eq. (18)
eq. (34), W_α

MAY–JUNE 1982

Table Ib—Entropy per pel and variance of the prediction error for nonadaptive and adaptive prediction—Scene John and Mike

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
2.393	2.190	142.1	Previous frame, eq. (38)
2.400	2.286	114.1	2-D Interframe, eq. (39)
2.154	2.094	61.5	3-D Interframe, eq. (40)
2.397	2.323	88.9	2-D Intraframe, eq. (41)
1.795	1.711	39.6	Predictor selection, eq. (11), (12), W_0
1.774	1.687	36.7	Predictor selection with soft switch eq. (18), (20), W_α
1.724	1.629	34.2	Gradient algorithm, eq. (34), W_α

Table Ic—Entropy per pel and variance of the prediction error for nonadaptive and adaptive predictors—Scene Mike and Nadine.

Entropy in Bit Per Pel		Variance	
H_{PEL}	H_{RUN}	$E[e^2]$	Prediction Scheme
2.859	2.809	194.9	Previous frame, eq. (38)
3.008	2.982	250.0	2-D Interframe, eq. (39)
2.537	2.504	108.0	3-D Interframe, eq. (40)
2.546	2.506	117.1	2-D Intraframe, eq. (41)
2.386	2.353	87.4	Predictor selection, eq. (11), (12), W_0
2.370	2.336	80.8	Predictor selection with soft switch eq. (18), (20), W_α
2.325	2.284	77.2	Gradient algorithm, eq. (34), W_α

Adaptive prediction schemes as given in Section II were simulated with (38) and (41) as predictor functions. The average bit rate per pel for three schemes are shown in the lower part of Tables Ia, b, and c. The adaptive schemes give an additional decrease in entropy if the horizontal run length coding technique is used; this improvement depends upon the type of picture.

Compared to the case of simple previous frame prediction, the predictor selection in combination with horizontal run length coding results in reductions of 18 to 29 percent. The corresponding reductions for the more sophisticated gradient method are 20 to 32 percent. The minimum and maximum entropy of a single frame within a sequence are reduced by about the same amount as the average entropy of the sequence. This can be recognized for the gradient method in Fig. 8, which shows the entropy per pel of each frame versus frame number.

In Section II, several modifications of the basic methods, to obtain a simpler hardware implementation, were presented. Most of these modifications have only a small influence on the entropy. The basic predictor selection scheme requires the summations of 8-bit numbers for determination of the decision functions (12). A coarse four-level quantizer

DPCM CODER 761

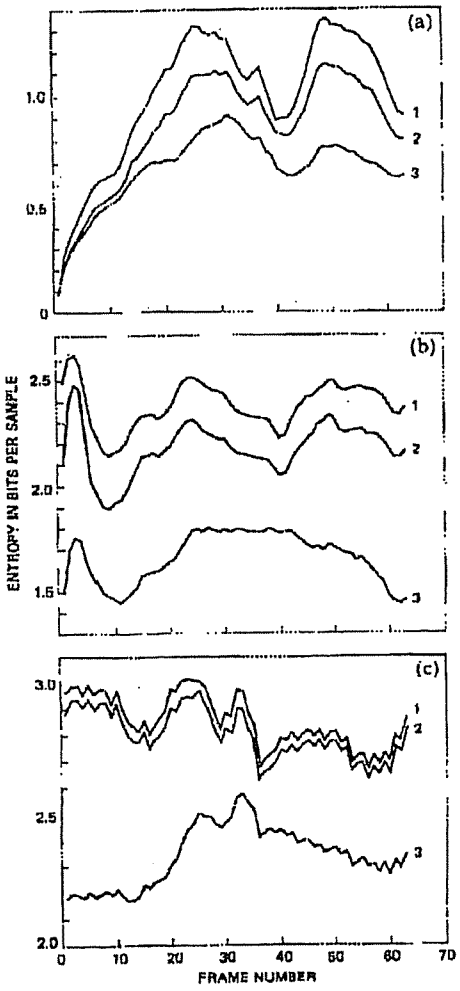
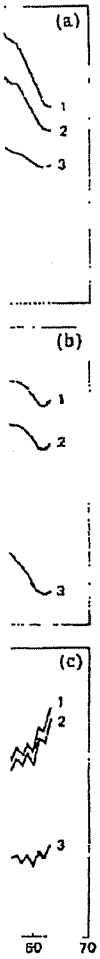


Fig. 8—Plots of entropy per pel versus frame number for each sequence. Configuration one shows the pel entropy H_{pel} of previous frame prediction; two shows the horizontal run length entropy H_{RUN} of previous frame prediction; and three shows the horizontal run length entropy H_{RUN} of the gradient algorithm (33) with the constraint (26). (a) Scene Judy. (b) Scene John and Mike. (c) Scene Mike and Nadine.



ich sequence. Configuration
; two shows the horizontal
three shows the horizontal
ith the constraint (26). (a)
Nadino.

$$Q_s(x) = \begin{cases} 0 & 0 \le |x| < 6 \\ 1 & 6 \le |x| < 18 \\ 2 & 18 \le |x| < 36 \\ 4 & 36 \le |x| \end{cases} \tag{42}$$

for determination of the decision function (16) increases the entropy by about 1 percent.

The use of a binary variable v_k, equation (18), which indicates which predictor function is advantageous at the position k, in combination with the soft-switch algorithm of equation (20) is to be preferred. Compared to the predictor selection scheme (11), (12), this algorithm provides a reduction of up to 7 percent in entropy. In addition, it is easier to implement.

For the gradient method the algorithm (34) which incorporates several modifications of the original method is useful concerning cost of implementation and the reduction in entropy. The constraint (26) is especially advantageous. For the algorithm (34), a three-level quantizer with thresholds at ±4 was used. The optimum value of γ was found to be 1/4. Each line started with initial values $b_1 = 1/2$ and $b_2 = 1/2$ for b. As long as the weighting coefficients b_j are represented by more than 4 bits, the gradient method provides a small gain in entropy compared to the predictor selection schemes.

In these investigations, three windows W_α, W_β, and W_γ were used. The window W_β provides results very close to that of W_α, whereas W_γ provides an increase of about 2 percent in entropy.

Further, it was found that using three predictor functions (the intraframe predictor is now split into two functions, one for horizontal prediction and one for vertical prediction) is not better. Besides the intraframe predictor function (39), the predictor function

$$f_2 = \frac{1}{2}s'_1 + \frac{1}{2}s'_3 \tag{43}$$

was also used. This resulted in an increase of 4 to 5 percent in the entropy.

It is of interest to know how these adaptive schemes perform in comparison with conditional replenishment and displacement compensation schemes. The results published in Ref. 6 (Table I, page 1235) based on the same source data are of some interest in this context. Hence, a comparison is possible, but it should be noted that the 35-level quantizer used in this investigation is a modification of the one used in Refs. 6 and 8. Further in this investigation, an additional thresholding of prediction error is not performed.

Compared to conditional replenishment the adaptive schemes provide a reduction in entropy of 19 to 38 percent, depending upon the

scene. For active scenes like John and Mike and Mike and Nadine, the adaptive schemes provide a data rate close to that of displacement compensation (within ±5 percent range). The run length coding scheme provides an additional reduction in entropy for sequences with low activity. For the sequence Judy, this reduction is 26 percent compared to conditional replenishment in case of previous frame prediction in combination with run length coding.

V. CONCLUSION

The performance of two types of adaptive intra-interframe predictors in combination with horizontal run length coding was studied. The gain in entropy of the predictor selection scheme is nearly as high as that of an adaptive scheme which is based on a gradient technique. Various modifications of the two basic methods which were investigated provided only small changes in entropy. Therefore, the adaptive algorithm which has the lowest cost of implementation should be chosen.

Further investigations are necessary for the quantizer design and the buffer control in a fixed rate system. A combination of the described adaptive intra-interframe algorithms with motion compensation will result in a more sophisticated system which provides further entropy reduction.

REFERENCES

1. F. W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment," B.S.T.J., *48*, No. 7 (September 1969), pp. 2545–54.
2. B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Coding of Videotelephone Pictures," Proc. IEEE, *60*, No. 7 (July 1972), pp. 792–800.
3. J. O. Limb, R. F. W. Pease, and K. A. Walsh, "Combining Intraframe and Frame-to-Frame Coding for Television," B.S.T.J., *53*, No. 6 (August 1974), pp. 1137–73.
4. H. Wendt, "Interframe-Codierung für Videosignale," Internationale Elektr. Rundschau (1973), pp. 2–7.
5. B. G. Haskell, "Frame Replenishment Coding of Television," in *Image Transmission Techniques*, W. K. Pratt, ed., New York: Academic Press, 1979.
6. J. A. Stuller, A. N. Netravali, and J. D. Robbins, "Interframe Television Coding Using Gain and Displacement Compensation," B.S.T.J., *59*, No. 7 (September 1980), pp. 1227–40.
7. P. R. Adby and M. A. H. Dempster, *Introduction to Optimization Methods*, London: Chapman and Hall, 1974.
8. A. N. Netravali and J. D. Robbins, "Motion-Compensated Television Coding: Part I," B.S.T.J., *58*, No. 3 (March 1979), pp. 631–70.
9. H. Yasuda et al., "Transmitting 4-MHz TV Signals by Combinational Difference Coding," IEEE Trans. Commun., *COM-25*, No. 5 (May 1977), pp. 508–16.
10. J. Klie, "Codierung von Fernsehsignalen für niedrige Übertragungsbitraten," Ph.D. Dissertation, University of Hannover, West Germany, 1978.
11. P. Pirsch and L. Stenger, "Statistical Analysis and Coding of Color Video Signals," Acta Electronica, *19*, No. 4 (1976), pp. 277–87.