

**UNITED STATES INTERNATIONAL TRADE COMMISSION
WASHINGTON, D.C.**

**Before The Honorable Doris Johnson Hines
Administrative Law Judge**

In the Matter of

**CERTAIN VIDEO CAPABLE
ELECTRONIC DEVICES, INCLUDING
COMPUTERS, STREAMING DEVICES,
TELEVISIONS, CAMERAS, AND
COMPONENTS AND MODULES
THEREOF**

Investigation No. 337-TA-1379

RESPONDENTS' RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. ARGUMENT	1
A. The Intrinsic Record Shows “Skip Coding Mode” Must Be Capable of Providing Residual Information.....	1
1. The Claims Show “Skip Coding Mode” Includes the Capability to Provide Residual Information.	1
2. The Specification’s Description of the Invention and Embodiments Reinforces the Scope of “Skip Coding Mode.”	4
B. Complainants Failed to Show a Clear Intent to Exclude Residual Information.	5
1. Complainants’ Alleged Lexicography Is Silent on Residual Information.	5
2. Complainants Cannot Fill the Patentee’s Silence With Expert Testimony Contradicting the Intrinsic Record.....	6
(a) Complainants’ Alleged Lexicography Distorts the Independent Claims and Invalidates Dependent Claims.	7
(b) Contrary to Complainants’ Arguments, the Specification Describes the “Present Invention” As Including Residual Information.	8
C. Complainants’ And Staff’s Claim Differentiation Arguments Are Flawed.	9
D. If the ALJ Agrees with Complainants’ Claim Differentiation Argument, the ALJ Should Find the “First Segment” Can Have Residual Information Somewhere.....	12
III. CONCLUSION.....	13

TABLE OF AUTHORITIES

	Page
CASES	
<i>AK Steel Corp. v. Sollac & Ugine</i> , 344 F.3d 1234 (Fed. Cir. 2003).....	1
<i>Baxalta Inc. et al. v. Genentech, Inc.</i> , 972 F.3d 1341 (Fed. Cir. 2020).....	5
<i>Clearstream Wastewater Sys., Inc. v. Hydro-Action, Inc.</i> , 206 F.3d 1440 (Fed. Cir. 2000).....	3
<i>Cont’l Cirs. LLC v. Intel Corp.</i> , 915 F.3d 788 (Fed. Cir. 2019).....	9
<i>Curtiss Wright Flow Control Corp. v. Velan, Inc.</i> , 438 F.3d 1374 (Fed. Cir. 2006).....	11
<i>Dippin’ Dots, Inc. v. Mosey</i> , 476 F.3d 1337 (Fed. Cir. 2007).....	11, 12
<i>Genentech, Inc. v. Chiron Corp.</i> , 112 F.3d 495 (Fed. Cir. 1997).....	11
<i>Littelfuse, Inc. v. Mersen USA EP Corp.</i> , 29 F.4th 1376 (Fed. Cir. 2022)	6
<i>Ortho-McNeil Pharm., Inc. v. Mylan Labs., Inc.</i> , 520 F.3d 1358 (Fed. Cir. 2008).....	8
<i>Pfizer, Inc. v. Ranbaxy Lab’ys Ltd.</i> , 457 F.3d 1284 (Fed. Cir. 2006).....	2, 4, 7
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	6, 7
<i>Ruckus Wireless, Inc. v. Innovative Wireless Sols.</i> , LLC, 824 F.3d 999 (Fed. Cir. 2016).....	8
<i>SciMed Life Sys. v. Advanced Cardiovascular Sys.</i> , 242 F.3d 1337 (Fed. Cir. 2001).....	4
<i>Tate Access Floors, Inc. v. Interface Architectural Res., Inc.</i> , 279 F.3d 1357 (Fed. Cir. 2002).....	1

STATUTES

35 U.S.C. 112.....	8
35 U.S.C. § 112.....	2, 4, 7, 11

OTHER AUTHORITIES

MPEP § 608.01(n)	2, 7
U.S. Patent No. 7,532,808.....	passim

I. INTRODUCTION

Respondents’ opening brief demonstrates that “skip coding mode” must be capable of providing residual information. Dependent claims recite *not* providing residual information in a bitstream. Under the doctrine of claim differentiation, “skip coding mode” in the independent claims cannot exclude residual information. The specification consistently describes the invention and several embodiments as including residual information.

Complainants base their proposed construction of “skip coding mode” that excludes residual information on an alleged lexicography and extrinsic expert testimony. Complainants fail to meet the “exacting” standard for lexicography because the intrinsic record never excludes residual information from the invention. For the same reason, Complainants cannot rebut the presumption of claim differentiation.

Accordingly, the ALJ should construe “skip coding mode” to include the capability of providing residual information.

II. ARGUMENT

A. The Intrinsic Record Shows “Skip Coding Mode” Must Be Capable of Providing Residual Information.

The two most important guides for claim construction—the claims and specification—support Respondents’ position that “skip coding mode” must be capable of providing residual information.

1. The Claims Show “Skip Coding Mode” Includes the Capability to Provide Residual Information.

Claim construction always begins with the claim language. *Tate Access Floors, Inc. v. Interface Architectural Res., Inc.*, 279 F.3d 1357, 1370 (Fed. Cir. 2002). “Under the doctrine of claim differentiation, dependent claims are presumed to be of narrower scope than the independent claims from which they depend.” *AK Steel Corp. v. Sollac & Ugine*, 344 F.3d 1234, 1242 (Fed.

Cir. 2003). Moreover, a dependent claim that fails to specify a further limitation of the subject matter is invalid for failing to comply with 35 U.S.C. § 112, ¶ 4. *See Pfizer, Inc. v. Ranbaxy Lab 'ys Ltd.*, 457 F.3d 1284, 1292 (Fed. Cir. 2006); MPEP § 608.01(n) at § III (“[N]on-compliance with 35 U.S.C. 112(d) renders the claim unpatentable just as non-compliance with other paragraphs of 35 U.S.C. 112 would.”). Here, claim differentiation between the independent claims (claims 1, 7, 10, and 16) and their respective dependent claims (claims 30, 41, 49, and 60) shows that “skip coding mode” must allow residual information because the only difference in the dependent claims is that they specifically recite not providing that information.

Independent claim 1 (below), for example, recites a method for encoding a video sequence. Notably, the claim is silent about the presence or absence of residual information.¹

1. A method of encoding a video sequence, the method comprising:
assigning a skip coding mode to a first segment of a first frame of the sequence;
assigning either a zero motion vector or a predicted non-zero motion vector for the skip coding mode for the first segment based at least in part on the motion information of a second segment neighboring the first segment; and
forming a prediction for the first segment with respect to a reference frame based at least in part on the assigned motion vector for the skip coding mode, wherein the assigned motion vector is one of the zero motion vector and the predicted non-zero motion vector; and
providing in an encoded bitstream an indication of the skip coding mode, wherein no further motion vector information for the first segment is coded in the encoded bitstream.]

'808 patent at cl. 1 (highlighting added). The Staff agrees with Respondents that the independent claims allow for residual information. Staff Op. Br. at p. 9 (noting that claim “uses permissible

¹ The patentee knew how to specify that information would not be sent in independent claims. For example, in claim 1, the patentee expressly stated that “no further motion vector information... is coded in the... bitstream.” No such limitation exists as to “skip coding mode.”

language for the use of prediction error [(i.e., residual)] information” and the intrinsic record “nowhere” explicitly disclaims use of residual information.). The Staff disagrees with Complainants’ position that residual information can never be provided for any skip coding mode. *See* Declaration of Dr. Michael Orchard in Support of Nokia and ITC Staff’s Proposed Construction for “Skip Coding Mode” in U.S. Patent No. 7,532,808 at ¶ 25 (hereinafter “Orchard Decl.”) (stating that “residual information can never be provided for any skip mode macroblock”), ¶ 20 (“residual information can never be provided for any skip mode macroblock associated with a ‘non-active’ motion vector”), ¶ 21 (“[N]o residual information is involved in the reconstruction of skip mode macroblocks associated with ‘active’ motion vectors.”); Staff Op. Br. at p. 9.² Respondents and the Staff agree on the scope of the independent claims.

Dependent claim 30 specifies an embodiment in which “no residual information is provided for the first segment”:

30. A method according to claim 1, wherein no residual information is provided for the first segment in the encoded bitstream.

’808 patent at cl. 30 (highlighting added). Because dependent claim 30 specifies that no residual information is provided, independent claim 1 must be capable of providing residual information. *Clearstream Wastewater Sys., Inc. v. Hydro-Action, Inc.*, 206 F.3d 1440, 1446 (Fed. Cir. 2000) (“Under the doctrine of claim differentiation, it is presumed that different words used in different claims result in a difference in meaning and scope for each of the claims.”). A POSITA³

² If the ALJ adopts Complainants’ position, it should modify Complainants’ construction to make clear that “skip coding mode” “cannot include residual information” because there is a claim construction dispute that requires resolution and Complainants’ construction, as written, does not resolve that dispute.

³ Complainants and the Staff propose a different level of ordinary skill in the art than Respondents. But, for purposes of this claim construction dispute, Respondents do not believe there is a material difference between the parties’ POSITA definitions.

would understand the dependent claims to mean “the ‘first segment’ that has been assigned a ‘skip coding mode’ in the independent claims can have residual information.” Ex. 1, Declaration of Dr. Joseph Havlicek (“Havlicek Decl.”) at ¶ 34.

If claim 1 cannot include residual information, then claim 30 does not add a narrowing limitation, and it is invalid. 35 U.S.C. § 112 ¶ 4 (a “claim in dependent form shall contain a reference to a claim previously set forth and then specify a further limitation”); *see, e.g., Pfizer*, 457 F.3d at 1292 (invalidating dependent claim that failed to specify a further limitation).

2. The Specification’s Description of the Invention and Embodiments Reinforces the Scope of “Skip Coding Mode.”

Additionally, the specification supports Respondents’ construction. The specification describes coding modes with residual information, as Respondents explained in their opening brief. Resp. Op. Brief at pp. 12-15. For example, the invention uses INTER coding modes, which indisputably include residual information,⁴ when defining “the present invention”⁵ for the encoder in Figure 6:

As the *present invention* relates to the encoding of video frames in INTER-coded format and more particularly to details of the motion-compensated prediction performed as part of the INTER coding process, description of encoder 600 in INTRA-coding mode will be omitted and the following sections will concentrate on the operations performed by the encoder in INTER-coding mode. In INTER-coding mode... combiner 116 forms corresponding blocks of prediction error [(i.e. residual information)].

’808 patent at 16:15-31 (emphasis added). Similarly, the invention uses INTER coding modes when defining “the present invention” for the decoder in Figure 7:

Furthermore, as the *present invention* affects the decoding of video frames in INTER-coded format, description of the operations performed by decoder 700 in

⁴ Nokia’s expert admits INTER modes include residual information. *See* Orchard Decl. at ¶ 18; *see also* Havlicek Decl. at ¶ 34.

⁵ *See, e.g., SciMed Life Sys. v. Advanced Cardiovascular Sys.*, 242 F.3d 1337, 1340-45 (Fed. Cir. 2001) (noting that language in the specification of “the present invention,” “defines [the patentee’s] invention”).

connection with the decoding of INTRA-coded frames will be omitted.... [A]n INTER-coded macroblock may compromise ... prediction error information.
Id. at 20:39-49 (emphasis added).

These statements show the patentee not only explicitly included INTER-coded modes with residual information in the scope of its invention, *see* Havlicek Decl. at ¶ 34, but that it also knew how to omit modes, e.g., INTRA-coded modes, that are unrelated to the invention.

Given the above evidence from the claims and specification, the ALJ should adopt Respondents’ construction.

B. Complainants Failed to Show a Clear Intent to Exclude Residual Information.

Unable to overcome the intrinsic record in support of Respondents’ construction, Complainants argue “skip coding mode” excludes residual information because the patentee redefined the term to exclude it in a couple of select specification excerpts. But lexicography requires a “clear intent” to exclude claim scope. *Baxalta Inc. et al. v. Genentech, Inc. et al.*, 972 F.3d 1341, 1349 (Fed. Cir. 2020) (“[T]he standard for lexicography is exacting, requiring the patentee to clearly express an intent to redefine a term.”) (internal quotation marks and citations omitted). Complainants gloss over that exacting legal standard and cherry-pick excerpts from the specification that are silent about residual information while ignoring contrary examples. Here, lexicography must be so clear that it also overcomes the presumption of claim differentiation. None of Complainants’ arguments or citations justify this extraordinary result. Indeed, the intrinsic record as a whole shows there is no lexicography to exclude residual information.

1. Complainants’ Alleged Lexicography Is Silent on Residual Information.

Complainants’ lexicography argument hinges on two excerpts in the specification that allegedly constitute a “redefinition” of skip coding mode. The two excerpts are silent about

residual information, vitiating Complainants’ ability to show “clear intent” to exclude claim scope from the start.

As shown below, these excerpts focus on motion information, not residual information, which is used in a separate step after accounting for motion. Compl. Op. Brief at p. 16. As Dr. Joseph Havlicek explains in his declaration, those specific excerpts do not mention whether residual information is sent or not and certainly do not say the invention can never send residual information. Havlicek Decl. at ¶¶ 40-41.

According to the invention, the coding modes used by encoder 600 correspond to those provided by JM1 of the JVT codec (shown in Table 3), with the exception that the SKIP mode is redefined to allow representation of global and regional motion. More specifically, the SKIP mode is modified in such a way that a zero (non-active) motion vector or a non-zero (active) motion vector is associated with each skip mode macroblock, depending on the characteristics of the motion in image segments surrounding the macroblock in question. In the following this type of motion vector will be referred to as a “skip mode motion vector”. (’808 patent at 17:43-53)⁶	In order to overcome, or at least mitigate to a large extent the problems associated with the coding of global motion in prior art video coding systems, the present invention is based on a redefinition of the skip mode concept used in JM1 of the JVT codec. The method according to the invention not only provides an improvement in coding efficiency in the presence of global motion (i.e. motion affecting the entire area of video frame), but also enables regional motion to be represented in an efficient manner. According to the invention, the skip mode concept is redefined in such a way that a macroblock assigned to skip mode is either associated with a zero (non-active) motion vector, in which case it is treated in the same way as a conventional skip mode macroblock and copied directly from the reference frame, or it is associated with a non-zero (active) motion vector. The decision as to whether a macroblock should be (’808 patent at 14:14-29)
---	--

That silence is fatal to Complainants’ lexicography argument. *Littelfuse, Inc. v. Mersen USA EP Corp.*, 29 F.4th 1376, 1379-82 (Fed. Cir. 2022) (finding no lexicography where “[n]othing in the specification state[d] that a fastening stem [could not] be present in a single-piece apparatus.”).

2. Complainants Cannot Fill the Patentee’s Silence With Expert Testimony Contradicting the Intrinsic Record.

Using expert testimony to speak for the patentee is itself suspect. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1318 (Fed. Cir. 2005) (expert testimony is “generated at the time of and for the

⁶ If the ALJ allows Complainants to tie their construction to the JM1 “redefinition” based on this part of the specification, the Construction should also include the other mentioned JM1 coding modes in Table 3, not just the redefinition part Complainants find favorable.

purpose of litigation and thus can suffer from bias that is not present in intrinsic evidence”). Here, it is especially problematic when it conflicts with the claims and specification. *See Phillips*, 415 F.3d at 1319 (expert testimony “is unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of the intrinsic evidence”).

Complainants try to fill the patentee’s silence with expert testimony that the patentee actually meant to limit “skip coding mode” to JM1 Skip Mode, which excludes residual information. Compl. Op. Br. at pp. 17-18 (“As Dr. Orchard explains in his declaration ... a POSITA would not understand the inventor to be replacing JM1’s SKIP Mode with a skip coding mode that does not itself skip ‘the coding transmission and decoding of residual information’ for skip mode macroblock.”). But Complainants’ expert relies on cherry picking certain excerpts and stretching simple phrases like “generate the pixels” to mean “generated directly ... and thus no ‘residual information’” when the specification teaches adding residual information to “generated” pixels. Havlicek Decl. at ¶¶ 46-47; *see also* Havlicek Decl. at ¶¶ 39, 41, 43 45, 50, and 52. Complainants’ expert does not discuss the claims, their use of the term “residual information,” or that Figures 6 and 7, and their descriptions, teach the use of residual information as part of “the present invention.” *See generally* Orchard Decl.

(a) Complainants’ Alleged Lexicography Distorts the Independent Claims and Invalidates Dependent Claims.

As explained above in Section A.1, the claims demonstrate that the patentee did not define “skip coding mode” to exclude residual information. Complainants’ construction renders dependent claims 30, 41, 49, and 60 invalid because they do not add a narrowing limitation as required under 35 U.S.C § 112 ¶ 4. *See, e.g., Pfizer*, 457 F.3d at 1292 (invalidating dependent claim that failed to specify a further limitation); MPEP § 608.01(n) at § III (“[N]on-compliance with 35 U.S.C. 112(d) renders the claim unpatentable just as non-compliance with other

paragraphs of 35 U.S.C. 112 would.”); *see also Ruckus Wireless, Inc. v. Innovative Wireless Sols., LLC*, 824 F.3d 999, 1004 (Fed. Cir. 2016) (stating principle that claims “should be construed to preserve [their] validity.”); *see also Ortho-McNeil Pharm., Inc. v. Mylan Labs., Inc.*, 520 F.3d 1358, 1363 (Fed. Cir. 2008) (construing claims to avoid the “nonsensical result” of “rendering several other dependent claims meaningless”).

Consequently, Complainants’ and Staff’s construction cannot be correct because it renders dependent claims meaningless.

(b) Contrary to Complainants’ Arguments, the Specification Describes the “Present Invention” As Including Residual Information.

Complainants also failed to show the patentee limited the invention to JMI Skip mode and disclaimed residual information in the specification. To the contrary, the patentee expressly said the invention was not limited to a modification of JM1 Skip Mode.

The patentee expressly stated that their invention was not limited to a supposed “redefinition” of JM1 Skip Mode, as Complainants contend, but could be used in other modes. ’808 patent at 25:4-8 (“The invention can also be used to generate motion compensated prediction for other Macroblock modes *in addition to or instead of the SKIP mode*. It is also not limited by the macroblock structure but can be used in any segmentation based video coding system.”) (emphasis added). Also, the patentee stated that, in an embodiment, there is no skip mode whatsoever—“the macroblock mode table is completely regenerated depending on the output of the surrounding motion analysis block, for example, *by removing the skip*.” ’808 patent at 24:57-60 (emphasis added). Complainants’ expert again ignores these portions of the specification.

Additionally, the patentee was specific when it wanted to exclude residual information. For example, the patentee expressly excluded residual information from being sent in an embodiment disclosed in connection with Figure 7. ’808 patent at 20:26-55 (“Operation of a video decoder 700

according to an embodiment of the invention will now be described with reference to FIG. 7 ... If a macroblock is encoded in skip mode, no prediction error or motion vector information relating to the macroblock is included in the bit-stream.”). If “skip coding mode,” by definition, did not include residual information, the patentee would not have needed to specify not providing residual information in connection with this embodiment.

Moreover, the patentee clearly stated when an embodiment was implemented according to JM1, so it follows that not all embodiments are so limited. ’808 patent at 23:13-17 (“In the example illustrated in FIG. 10, the video codec is implemented according to JM1 of the JVT codec, with appropriate modifications to implement the modified SKIP mode concept according to the invention in both the encoder and the decoder of the video codec.”).

In all, these examples show the patentee did not intend their invention to be limited to JM1 or for “skip coding mode” to exclude residual information.⁷ *See Cont’l Circs. LLC v. Intel Corp.*, 915 F.3d 788, 798 (Fed. Cir. 2019) (finding no lexicography where intrinsic record was inconsistent).

C. Complainants’ And Staff’s Claim Differentiation Arguments Are Flawed.

Complainants argue claim differentiation does not apply because the dependent claims recite no residual information “*in the encoded bitstream*,” and somehow “in the encoded bitstream” is a separate limitation. Compl. Op. Brief at p. 22 (emphasis added). This is incorrect for several reasons.

First, “in the encoded bitstream” is not a separate limitation. It simply clarifies that there is

⁷ Complainants’ alleged lexicography also lacks support in the provisional application to which the ’808 patent claims priority. Specifically, none of the excerpts on which Complainants rely for lexicography is found in the provisional application. *Compare* ’808 patent at 17:43-53 and 14:14-29 *with* Ex. 3 (’808 patent provisional application). Those excerpts and the embodiments underlying Complainants’ construction, including JM1, were not added until later.

no residual information anywhere. Indeed, the phrase follows the same style and language found in other claims. *See, e.g.*, '808 patent at cl. 1 (“wherein no further motion vector information for the first segment is coded in the encoded bitstream.”). So “no residual information ... in the encoded bitstream” just means “no residual information” for purposes of the claims. For the independent claims, this also means there can be residual information somewhere in order for the dependent claim to be narrowing.

Second, the specification is clear that the coding mode defines whether a macroblock can or cannot have residual information. Havlicek Decl. at ¶¶ 35-36. In other words, the “skip coding mode” defines whether there can be residual information in the bitstream or not, which, as Dr. Havlicek explains “is necessary so the decoder knows whether to look for residual information in the bitstream for that macroblock or not to look.” *Id.* For example, “[d]epending on the coding mode, the compressed video data included in the bit-stream for an INTER-coded macroblock may comprise ... prediction error information for each block....” '808 patent at 20:46-49 (emphasis added). Complainants’ expert also agrees with Respondents that the coding mode signals whether the macroblock can have residual information. Orchard Decl. at ¶¶ 17-19 (comparing macroblock modes and acknowledging that “reconstruction in INTER modes” can have “residual information” and a prior art skip mode cannot).

Lastly, Complainants ignore an essential part of the claim language: the fact that the dependent claims specify “no residual information is provided *for the first segment [i.e., the skip coded segment]* in the encoded bitstream.” *Id.* (emphasis added). Contrary to Complainants’ argument, the dependent claims do not broadly claim that residual information is not provided *for some other segment* in the encoded bitstream, but rather add a specific limitation that residual

information is not provided *for the first segment [i.e., the skip coded segment]* in the encoded bitstream.⁸

Staff argues that “comprising” in the preambles of the independent claims means the term “skip coding mode” does not need to allow or not allow residual information. Staffs Brief. at pp. 10-11. Staff misapplies the law on “comprising.” The term “comprising” in a preamble raises a presumption that the list of claim limitations is nonexclusive. But “‘comprising’ is not a weasel word with which to abrogate claim limitations.” *Dippin’ Dots, Inc. v. Mosey*, 476 F.3d 1337, 1343 (Fed. Cir. 2007); *see also Genentech, Inc. v. Chiron Corp.*, 112 F.3d 495, 501 (Fed. Cir. 1997).

In *Dippin’ Dots*, the Federal Circuit rejected a similar argument. There, the patentee argued the term “beads” could include irregularly shaped beads because the preamble used “comprising,” even though the intrinsic record disclosed spherical beads. *Dippin’ Dots*, 476 F.3d at 1342-43. The Federal Circuit rejected the argument because while the transitional term “comprising” rendered the enumerated steps open-ended and thus allowed for additional steps, the “presumption raised by the term ‘comprising’ does not reach into each of the six steps to render every word and phrase therein open-ended.” *Id.* at 1343.

⁸ Complainants cite *Edwards Lifesciences LLC v. Cook Inc. et al.*, 582 F.3d 1322 (Fed. Cir 2009). *Edwards* is distinguishable. Here, the parties have not “agreed at trial” to a construction (*i.e.*, waiver). Further, “every embodiment described in the specification and shown in the drawings” of the ’808 does not exclude residual information, the ’808 patent does not use “the present invention” language to exclude residual information, every claim does not have “a requirement that” having no residual information “perform[s] [a certain] function,” and here the dependent claims do not “add further limitations.” *Id.* at 1331-32. *Curtiss Wright Flow Control Corp. v. Velan, Inc.*, 438 F.3d 1374, 1380 (Fed. Cir. 2006) does not support Complainant’s claim construction either. In *Curtiss Wright*, the Federal Circuit applied a different standard because the comparison was between independent claims, not dependent claims. *Curtiss Wright* noted that if it was a dependent claim, “reading an additional limitation from a dependent claim into an independent claim ... might render the dependent claim invalid” under 35 U.S.C. § 112 ¶ 4. *Id.* at 1380.

Here, as in *Dippin’ Dots*, “comprising” is a transitional term in the preamble, indicating that the claim limitations are nonexclusive, and leaving the claim open to additional limitations that are consistent with the stated limitations. But “comprising” does not reach into each of the recited limitations, like “skip coding mode,” and render them open-ended.

Staff’s position is also incorrect from a technical perspective: the “skip coding mode” in the independent claims cannot be defined as both a skip coding mode with residual and a skip coding mode without residual. The parties agree that coding modes are designed to be either capable *or* incapable of providing residual information. *See* Orchard Decl. at ¶¶ 17-19 (comparing coding modes and acknowledging that “reconstruction in INTER modes” can have “residual information” and a prior art skip mode cannot); Havlicek Decl. at ¶ 37 (“a macroblock coding mode defines whether the macroblock can have residual information or cannot”). Coding modes that are capable of providing residual information tell the decoder to look for residual information. *See* Havlicek Decl. at ¶ 37 (“This is necessary so the decoder knows whether to look for residual information in the bitstream for that macroblock or not to look.”). This does not mean a macroblock coded under this mode must provide residual information, but the coding mode must provide that capability. *See* Havlicek Decl. at ¶ 37 (“a macroblock coding mode can allow a macroblock to have residual information and still decide not to send residual information.”). By contrast, coding modes that are incapable of providing residual information do not have such a code because they do not need one. *Id.* at ¶ 35.

D. If the ALJ Agrees with Complainants’ Claim Differentiation Argument, the ALJ Should Find the “First Segment” Can Have Residual Information Somewhere.

If the ALJ agrees with Complainants’ claim differentiation argument that “the entire coded bitstream does not contain residual information for the first segment,” the ALJ should still find that the independent claims 1, 7, 10 and 16 require that the “first segment *can* have residual

information somewhere in the encoded bitstream.” In other words, under Complainants’ theory, the dependent claims necessitate that whatever the coding mode of the first segment is, the first segment must be capable of having residual information somewhere in the bitstream and infringement of the independent claims requires a showing of this capability.

III. CONCLUSION

For the reasons stated herein, the ALJ should adopt Respondents’ position and construe skip coding mode as: “a coding mode in which a zero (nonactive) motion vector or a nonzero (active) motion vector is associated with each skip mode macroblock, depending on the characteristics of the motion in image segments surrounding the macroblock in question, and residual information can be provided for each skip mode macroblock.”

GROUND RULE 1.7 CERTIFICATION

Pursuant to Ground Rule 1.7, on behalf of Respondents Amazon.com, Inc., Amazon.com Services LLC, and HP, Inc., the undersigned hereby certifies that Respondents’ Opening Claim Construction Brief complies with the word count limitations and contains 3,247 words according to the word-processing program used to prepare the brief (Microsoft Word). The word count includes footnotes. The word count does not include captions, the signature block, or the certificate of service.

DATED: February 23, 2024

Respectfully submitted,

/s/ Thomas N. Millikan

Veronica S. Ascarrunz

Perkins Coie LLP

700 13th Street, NW, Suite 800

Washington, DC 20005-3960

(202) 654-6200

Stefani E. Shanberg

Robin L. Brewer

Perkins Coie LLP
505 Howard Street, Suite 1000
San Francisco, California 94105-3204
(415) 344-7000

Thomas N. Millikan
Perkins Coie LLP
11452 El Camino Real, Suite 300
San Diego, CA 92130-2080
(858) 720-5700

Stephen S. Korniczky
Martin R. Bader
Sheppard, Mullin, Richter & Hampton LLP
12275 El Camino Real, Suite 100
San Diego, CA 92130-4092
(858) 720-8900

*Counsel for Respondents Amazon.com, Inc. and
Amazon.com Services LLC*

/s/ Stephanie L. Roberts

Eric S. Namrow
Stephanie L. Roberts
Kandis C. Gibson
Manu Bansal
Kevin J. Spinella
Emily K. Burge
Eunjean Je
Morgan, Lewis & Bockius LLP
1111 Pennsylvania Ave, N.W.
Washington, D.C. 20004-2541
Telephone: (202) 739-3000
Facsimile: (202) 739-3001

Jason E. Gettleman
Ehsun Forghany
Morgan, Lewis & Bockius LLP
1400 Page Mill Road
Palo Alto, CA 94304
Telephone: (650) 843-4000
Facsimile: (650) 843-4001

Rick L. Rambo
Thomas R. Davis
Morgan, Lewis & Bockius LLP

1000 Louisiana Street, Suite 4000
Houston, TX 77002-5005
Telephone: (713) 890-5000
Facsimile: (713) 890-5001

Elizabeth M. Chiaviello
Morgan, Lewis & Bockius LLP
1717 Main Street, Suite 3200
Dallas, TX 75201-7347
Telephone: (214) 466.4000
Facsimile: (214) 466.4001

Counsel for Respondent HP Inc.

**UNITED STATES INTERNATIONAL TRADE COMMISSION
WASHINGTON, D.C.**

**Before the Honorable Doris Johnson-Hines
Administrative Law Judge**

**CERTAIN VIDEO CAPABLE
ELECTRONIC DEVICES, INCLUDING
COMPUTERS, STREAMING DEVICES,
TELEVISIONS, CAMERAS, AND
COMPONENTS AND MODULES
THEREOF**

Investigation No. 337-TA-1379

**EXPERT DECLARATION OF JOSEPH P. HAVLICEK, PH.D, REGARDING CLAIM
CONSTRUCTION OF U.S. PATENT NO. 7,532,808**

I, Joseph P. Havlicek, Ph.D., declare as follows:

I. INTRODUCTION

1. My engagement is the same as that specified in my first declaration in paragraph 1. For this second declaration, I have been asked to provide my opinions regarding the understanding a person having ordinary skill in the art would have had regarding certain language in U.S. Patent No. 7,532,808 (“the ’808 patent” or “the Asserted Patent”). This declaration sets forth my opinions relating to the claim construction issues addressed below.

2. As explained in paragraph 1 of my first declaration, my name is Joseph P. Havlicek, Ph.D., and I am currently Gerald Tuma Presidential Professor of Electrical and Computer Engineering at the University of Oklahoma. I have been retained by Respondents Amazon.com, Inc. and Amazon.com services LLC (collectively, “Amazon”) as an independent expert consultant regarding *Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, Cameras, and Components and Modules Thereof*, Inv. No. 337-TA-1379. My opinions are based on my education, experience, and background in the fields discussed below and

on my analysis as also described below. In forming my opinions, I have analyzed the '808 Patent, its prosecution history, and all other materials identified in this declaration.

A. Educational Background

3. I received a Bachelor of Science Degree in electrical engineering with minors in mathematics and computer science from Virginia Tech in 1986. I also received a Master of Science degree in electrical engineering from Virginia Tech in 1988. I received a Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1996. My Ph.D. research was in the field of image processing.

B. Professional Experience

4. From December 1984 to May 1987, I was a software engineer at Management Systems Laboratories, Blacksburg, VA. My main job responsibilities included developing software for nuclear materials management under contract with the United States Department of Energy.

5. From June 1987 to January 1997, I was an electrical engineer at the United States Naval Research Laboratory. For the period of June 1987 through August 1989, I was an on-site contractor affiliated with SFA, Inc, Landover, Maryland. From August 1989 through January 1997, I was a regular government employee. I was on leave without pay from August 1987 through July 1988 while completing my Master of Science Degree. I was also on leave without pay for much of the period from August 1990 through January 1997 while I completed my Ph.D. degree. My main job responsibilities at the United States Naval Research Laboratory included designing digital and analog circuits to process real-time video signals and designing and implementing target detection, tracking, and identification algorithms. I developed missile warning receivers for Navy aircraft including the Navy's first two-color infrared system. I was a recipient of the 1990

Department of the Navy Award of Merit for Group Achievement for this work. The production version of this system was deployed on Navy helicopters and saw combat in Afghanistan and Iraq.

6. From January 1993 through December 1993, I was an on-site contractor at International Business Machines (IBM) Corporation, Austin, TX. My main job responsibilities included designing and implementing image compression and decompression algorithms (CODECs) for IBM products.

7. Since January 1997, I have been a faculty member in the School of Electrical and Computer Engineering at the University of Oklahoma, Norman, OK. I was an Assistant Professor from January 1997 through June 2002. I was promoted to the rank of Associate Professor and granted tenure in July 2002. I was promoted to the rank of Professor in July 2007. I was appointed to the Williams Companies Foundation Presidential Professorship in April 2009. In April 2017 I was appointed to the Gerald Tuma Presidential Professorship. My main job responsibilities at the University of Oklahoma include conducting academic research in electrical and computer engineering, teaching graduate and undergraduate courses in electrical and computer engineering, and performing professional and institutional service.

8. I am a member of several professional societies and organizations, including the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Signal Processing Society, the IEEE Computer Society, and the IEEE Intelligent Transportation Society. I am a Senior Member of the IEEE. From November 2015 through February 2018, I served as a Senior Area Editor for the IEEE Transactions on Image Processing. I was formerly an Associate Editor for the IEEE Transactions on Image Processing from December 2010 through October 2015. I served as a Technical Area Chair for the IEEE International Conference on Image Processing in the area of Image & Video Analysis, Synthesis, and Retrieval (2012, 2013) and have served on the organizing

committee of that conference (2007). I have also served as a Technical Area Chair for the IEEE International Conference on Acoustics, Speech, and Signal Processing in the area of Image, Video, and Multidimensional Signal Processing (2012-2014).

9. For over 35 years, I have conducted research and taught classes in the field of image and video processing and analysis. My main scholarly contributions have been in the areas of modulation domain image models and image processing (AM-FM image models), video target tracking, and distributed control of video networks for intelligent transportation systems. I have supervised or co-supervised 12 Ph.D. students to completion and I am currently supervising two Ph.D. students. I have been a member of 67 additional doctoral dissertation committees. I have supervised 27 M.S. students to completion and I am currently supervising two additional M.S. students. I have served as a committee member on 68 additional M.S. committees. A listing of my Ph.D. and M.S. supervisions and committee memberships may be found in my curriculum vitae.

10. I teach a variety of courses at the University of Oklahoma, including the required junior-level Signals and Systems course ECE 3793 (taught 21 times), the graduate level Digital Image Processing course ECE 5273 (taught 25 times), and the graduate level Digital Signal Processing course ECE 5213 (taught 18 times). I have received a number of teaching awards, including the University of Oklahoma College of Engineering Outstanding Faculty Advisor Award (2005-2006) and the University of Texas Engineering Foundation Award for Exemplary Engineering Teaching while Pursuing a Graduate Degree (1992).

11. Since joining the University of Oklahoma in January 1997, I have been Principal Investigator or Co-Principal Investigator on over 110 externally funded grants and contracts with a total value of \$27,061,653. My main research contributions have been in the areas of signal, image, and video processing, video target tracking, and intelligent transportation systems. I have

been author or coauthor on over 130 scholarly publications in these areas. I was a recipient of the 1990 Department of the Navy Award of Merit for Group Achievement for my work in video target tracking. My research group at the University of Oklahoma originated the Virtual Traffic Management Center concept featured in a December 2014 FHWA technical report (Guidelines for Virtual Transportation Management Center Development) and a November 2014 FHWA national webinar with the same title. I am co-founder and director of the University of Oklahoma Center for Intelligent Transportation Systems. Under my supervision, the Center has collaborated with the Oklahoma Department of Transportation since 1998 to design and implement the Oklahoma Statewide Intelligent Transportation System, including a geographically distributed video network that is currently deployed on major highways and interstates across the entire State of Oklahoma.

12. I have also published numerous peer reviewed scholarly journal articles, book chapters, and conference papers, including 24 journal articles, 9 book chapters, and 100 conference papers. Exhibit A is a copy of my current curriculum vitae, which provides further details about my academic and professional background and qualifications, including publications.

II. LEGAL PRINCIPLES

13. I am not a legal expert or a lawyer and offer no opinions on the law. However, I have been instructed on certain aspects of patent law relevant to these proceedings by counsel. My understanding of the relevant legal principles as they apply to these proceedings are summarized below.

A. Person of Ordinary Skill in the Art

14. I understand that a “person of ordinary skill in the art” (“POSITA”) refers to a hypothetical person who is familiar with the relevant art at the time of the invention. Many factors may determine the level of ordinary skill in the art, including: (1) the type of problems encountered

in the art, (2) prior art solutions to those problems, (3) the rapidity with which innovations are made, (4) the sophistication of the technology, and (5) the educational level of active workers in the field. I understand that a POSITA is a person of ordinary creativity, not an automaton, meaning that a POSITA may employ inferences and creative steps in their work.

15. I believe that a POSITA at the time of the alleged invention of the '808 patent would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study, and (2) approximately two to three years of practical experience with video and/or image processing or coding. Additional experience can substitute for the level of education, and vice-versa.

16. Based on my education, training, and professional experience in the field of the '808 Patent, I possessed the abilities and knowledge of at least a POSITA at the time of the claimed inventions of the '808 Patent, which I understand Nokia contends to be March 15, 2002. My opinions would be the same if the priority date were around the filing date of the '808 Patent, which I understand to be March 14, 2003. In addition, based on my experience I am able to assess the '808 Patent as well as the prior art from the vantage point of a POSITA at the time of the claimed inventions of the '808 Patent.

B. Claim Construction

17. I understand that the terms of a patent claim are generally given their ordinary and customary meaning, which is the meaning the claim terms would have to a POSITA at the time of the invention after reading the patent and its prosecution history. I understand that a POSITA is deemed to read a particular claim term not only in the context of the claim in which the term appears, but also in the context of the entire patent.

18. I understand that other sources should also be consulted to ascertain the meaning of the terms used in the patent. Those sources include, in addition to the claims themselves, the specification of the patent, its prosecution history, and dictionaries, treatises, standards documents, or other evidence extrinsic to the patent.

19. I understand that the claims of a patent define the boundaries of the patentees' exclusive rights. To determine the scope of those claims, courts often define the meaning of claim terms that the parties dispute. I understand that claim terms should generally be given their plain and ordinary meaning as understood by one of ordinary skill in the art at the time of the invention. But, I understand that claim terms must also be defined consistent with the intrinsic evidence, which includes the patent's specification and the patent's prosecution history.

20. I understand the language of the claims helps guide the construction of claim terms. For example, the context in which a claim term is used in the claims can be highly instructive about the meaning of the claim term. I understand that the presence of a dependent claim that adds a particular limitation gives rise to a presumption that the limitation in question is not present in the independent claim. I also understand that, beyond the wording in the claims, the specification of the patent is the best guide to the meaning of a disputed claim term. However, I have been informed that the limitations of preferred embodiments and examples appearing in the specification should not be read into the claims.

21. I understand that use of the phrase "the present invention" can also define an invention. I understand that a patentee may also define claim terms in the specification to give a claim term a different meaning than it would ordinarily possess or to limit the scope of the term.

22. I understand that the prosecution history is another important source of evidence in the claim construction analysis. I understand that the prosecution history is the record of the

proceedings before the U.S. Patent and Trademark Office (PTO), including communications between the patentee and the PTO. I understand that the prosecution history can inform the meaning of the claim language by demonstrating how the patentee and the PTO understood the invention and whether the patentee limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be. I understand that a patentee may also define a term during the prosecution of the patent. I also understand that a patentee is precluded from proposing a construction for a claim term that would recapture claim scope that the patentee clearly and unambiguously disclaimed or disavowed during prosecution.

23. I understand that courts may also consider extrinsic evidence when construing claims. Extrinsic evidence is any evidence that is outside the patent and the prosecution history corresponding to a claim term. I understand that examples of extrinsic evidence include technical dictionaries, treatises, and expert testimony. I understand that intrinsic evidence is given more weight when determining the meaning of claim language.

24. I understand that a process or method claim covers an act or a series of acts or steps. I also understand that an apparatus or system claim covers a concrete thing, consisting of parts, or of certain devices and combinations of devices.

III. OPINION REGARDING CLAIM CONSTRUCTION

25. In providing the opinions below, I have applied the principles above, and the level of a POSITA, at the time of the alleged invention.

26. I have reviewed the '808 patent and its file history.

27. I have reviewed the references incorporated in the '808 patent.

28. I have reviewed the U.S. Provisional Application No. 60/365,072, to which the '808 patent claims priority.

29. I have reviewed Respondents' opening claim construction briefing.

30. I have reviewed Complainants' opening claim construction briefing.

31. I have reviewed the Staff's opening claim construction briefing.

32. Dr. Orchard appears to agree with me that a "POSITA would understand "prediction error information" and "residual information" in the '808 patent to mean the same thing." Cf. Havlicek Opening Decl. at paras. 30-31 and Declaration of Dr. Michael Orchard in Support of Nokia and ITC Staff's Proposed Construction for "Skip Coding Mode" in U.S. Patent No. 7,532,808 at para 18 (hereinafter "Orchard Decl.").

33. I disagree with Dr. Orchard that the '808 Patent only teaches a "'skip coding mode' that provides no 'residual information' for any skip mode macroblocks." Orchard Decl. at paras. 15-16. In my opinion, a person of ordinary skill in the art would understand that the '808 Patent as a whole teaches a "skip coding mode" that can have "residual information" for skip coding mode macroblocks. It was known in the art to use skip modes for interpolated frames (*i.e.* skipped frames) and predicted frames that also could send residual information. *See, e.g.*, Exhibit 1, MPEG 89/182 at Sec. 2.4.3; see also, Exhibit 2, MPEG91/005 at pg. 25 ("Macroblock Types for P-frames").

34. For example, dependent claims 30, 41, 49, and 60 teach "wherein no residual information is provided for the first segment," which a POSITA would understand, by applying, claim differentiation, to mean the "first segment" that has been assigned a "skip coding mode" in the independent claims can have residual information." '808 at cls. 30, 41, 49 and 60. As another example, the specification discusses that "the present invention relates to the encoding of video frames in INTER-coded format," not a "skip coding mode" as distinct from any INTER-coded mode. '808 at 16:15-21. Rather, the '808 patent blurs the lines between traditional INTER-coded

modes and skip mode. Dr. Orchard agrees that INTER-coded formats can have residual information. Orchard Decl. at para. 18. Similarly, and as another example, the specification discusses that “*the present invention* affects the decoding of video frames in INTER-coded format,” not just “skip coding mode.” ’808 at 20:39-42. Lastly, Figures 6 and 7 show and their descriptions discuss including prediction error (i.e. residual information). ’808 at Figs. 6 and 7. Notably, Dr. Orchard does not address these examples in his opening declaration. See, generally, Orchard Decl. In my opinion, a POSITA would understand these examples to mean that the ’808’s redefined skip coding mode can have residual information.

35. In my opinion, a POSITA would understand that a macroblock coding mode defines whether the macroblock can have residual information or not. This is necessary so the decoder knows whether to look for residual information in the bitstream for that macroblock or not to look. For example, the GMVC_COPY macroblock signaled that no residual information can be sent. VCEG-O20 at 3-4. So a GMVC decoder would not expect to receive residual information for a GMVC_COPY macroblock. *Id.* Conversely, a GMVC_16 mode signals to the decoder that residual information can be sent. So the GMVC decoder should look for residual information. As another example, JM1 states, “Macro block type (MB_Type) Refer to TABLE 7. There are different MB-Types for Intra and Inter frames.” JM1 at 17. Table 7 shows whether each macroblock type can have residual information (*i.e.*, a Coded Block Pattern). JM1 at 57. As discussed below, a macroblock mode can signal to the decoder that a macroblock can have residual information and then another code can signal that no residual information is present.

36. Dr. Orchard appears to agree with me that macroblock modes identify whether a macroblock uses prediction error or not. See Orchard Decl. at paras. 18-19.

37. I disagree with Dr. Orchard's opinion that Respondents proposed construction would mean a skip coding mode would necessarily include residual information. Orchard Decl. at para. 26. A POSITA reading the '808 patent would understand that a macroblock coding mode can provide the option for a macroblock to have some residual information and still decide not to send residual information. For example, JM1 describes codes that signal whether "one or more of the 4 4x4 blocks contain coefficients" and that a "0 in position n of the CBP... means that the corresponding 8x8 block *has no coefficients*." In other words, a 0 can signal no coefficients and no residual information. Here is the excerpt from JM1:

3.4.6 Coded Block Pattern (CBP)

The CBP contains information of which 8x8 blocks - Luminance and chrominance - contain transform coefficients. Notice that an 8x8 block contains 4 4x4 blocks meaning that the statement '8x8 block contains coefficients' means that 'one or more of the 4 4x4 blocks contain coefficients'. The 4 least significant bits of CBP contain information on which of 4 8x8 luminance blocks in a macroblock contains nonzero coefficients. Let us call these 4 bits CBPY. The ordering of 8x8 blocks is indicated in FIGURE 3. A 0 in position n of CBP (binary representation) means that the corresponding 8x8 block has no coefficients whereas a 1 means that the 8x8 block has one or more non-zero coefficients.

For chrominance we define 3 possibilities:

- nc=0: no chrominance coefficients at all.
- nc=1 There are nonzero 2x2 transform coefficients. All chrominance AC coefficients = 0. Therefore we do not send any EOB for chrominance AC coefficients.
- nc=2 There may be 2x2 nonzero coefficients and there is at least one nonzero chrominance AC coefficient present. In this case we need to send 10 EOBs (2 for DC coefficients and 2x4=8 for the 8 4x4 blocks) for chrominance in a macroblock.

The total CBP for a macroblock is: **CBP = CBPY + 16xnc**

The CBP is signalled with a different codeword for Inter macroblocks and Intra macroblocks since the statistics of CBP values are different in the two cases.

38. I agree with Dr. Orchard's opinion that one embodiment of the '808 patent at 14:16-29 teaches a skip-mode based on modifying the motion information for JM1's skip mode to associate zero or non-zero motion vectors. Orchard Decl. at para. 20.

39. I disagree with Dr. Orchard's opinion that the embodiment of '808 patent at 14:16-29 is a "definition of 'skip mode'" and that the embodiment is a basis to say "residual information can never be provided for any skip mode macroblock associated with a 'non-active' motion vector"

as Dr. Orchard opines. Orchard Decl. at para. 20. That specific embodiment does not say residual information cannot be sent.

40. I agree with Dr. Orchard's opinion that the embodiment of the '808 patent at 14:16-29 is an example where the '808 redefined "skip mode" macroblock signals that it will not use residual information in conjunction with a zero (non-active) motion vector. Orchard Decl. at para. 20. I disagree with his conclusion that "residual information can never be provided for any skip mode macroblock associated with a 'non-active' motion vector" as Dr. Orchard opines. Orchard Decl. at para. 20. That excerpt does not say it can never use residual information.

41. I agree with Dr. Orchard's opinion that one embodiment of the '808 patent at 14:32-42 teaches a skip-mode based on modifying a motion vector relative to JM1. Orchard Decl. at para. 22.

42. I disagree with Dr. Orchard's opinion that the embodiment of '808 patent at 14:32-42 not mentioning residual information is an omission that leads to an inference that only motion vectors are modified and "no such additional modifications are intended." Orchard Decl. at para. 22. That specific embodiment does not mention whether residual information is sent or not sent.

43. I agree with Dr. Orchard's opinion that one embodiment of the '808 patent at 15:63-67 states that "motion parameters conforming to the motion are generated... In this way, no additional information needs to be transmitted." Orchard Decl. at para. 23.

44. But I disagree with Dr. Orchard's opinion that the embodiment of the '808 patent at 15:63-67 excludes residual information. This embodiment is discussing motion information when it says "no additional information needs to be transmitted," not residual information. *Id.* at 15:63-67. And, other embodiments, unaddressed by Dr. Orchard, state that other motion information (i.e. "side information") can be sent. "Additional side information can also be sent to

guide the generation of motion parameters.” ’808 at 24:65-7; see also 24:46-50 (“Instead of using the surrounding motion analysis block to indicate the classification information, the indication is provided by other means (for example as side information in macroblock, slice, picture or sequence levels).”)

45. I agree with Dr. Orchard’s opinion that one embodiment of the ’808 patent at 15:63-67 states that a “non-zero motion vector is used to generate the pixel values for the macroblock.” Orchard Decl. at para. 24.

46. But I disagree with Dr. Orchard’s modification of this language to say it means pixels are “generated directly... and thus no ‘residual information’ is provided.” The ’808 patent explains in another embodiment that pixel values are “generated” by motion compensation and can be added to prediction error (i.e. residual information). *See* ’808 at 16:65-17:03 (“In INTER-coding mode... combiner 112 also receives predicted pixel values for each block of the macroblock, generated by motion-compensated prediction block 650. The combiner 112 combines each of the locally decoded blocks of prediction error values with a corresponding block of predicted pixel values to produce reconstructed image blocks...”). Therefore, a POSITA would clearly understand that the word “generate” as used in the ’808 patent specifically includes the possibility of adding residuals to the prediction values to generate the reconstructed pixel values.

47. I disagree with Dr. Orchard’s opinion that including residual in the redefined skip mode of the ’808 patent would be an oxymoron. Orchard Decl. at para. 26. As discussed above in paragraphs 33 and 34, the ’808 patent discusses changing skip mode and describes in the claims and embodiment with residual information as “the present invention”.

48. I agree with Dr. Orchard’s opinion that the ’808 patent at 15:7-11 states that [i]n contrast to both the global motion compensation and global motion vector coding methods... no

additional information must be transmitted in the video bit-stream to enable correct decoding of the video data.” Orchard Decl. at para. 27.

49. But I disagree with Dr. Orchard’s interpretation of this language to say it means “no additional information, such as residual information, can be transmitted.” Dr. Orchard’s opinion incorrectly changes “no additional information... must be” to “additional information... cannot be”. The excerpt is unclear in what the “additional information” is in addition to. It appears to be comparing the “additional information” to the prior art global motion compensation and global motion vector coding methods, which transmitted motion information. And, while the excerpt says “no additional information *must* be transmitted,” as discussed above, para. 50, the patent describes embodiments where additional information is explicitly transmitted. ’808 at 24:46-49 and 66-67.

50. I agree with Dr. Orchard’s opinion that the ’808 patent at 15:11-14 states that “a minimal amount of modification is required to incorporate the method according to the invention into existing video coding systems that employ the concept of skip mode macroblocks.”” Orchard Decl. at para. 28.

51. But I disagree with Dr. Orchard’s assertion that, “it would require a substantial and inefficient modification to implement residual information within a skip coding mode...” Orchard Decl. at para. 28. No substantial modification is necessary. The ’808 patent describes Figures 6 and 7 as implementing skip coding mode and prediction error (*i.e.*, residual information). ’808 at 15:38-39 and 15:40-42. I also disagree with Dr. Orchard that Table 3 of the ’808 Patent “would lack any coding mode in which no residual information is provided.” Orchard Decl. at para. 28. As discussed above in paragraph 36, the skip mode and other modes could signal that they have no residual information.

This declaration as executed on February 23, 2024, in Norman, Oklahoma.

A handwritten signature in blue ink, appearing to read "JP Havlicek", is written over a horizontal line.

Joseph P. Havlicek

EXHIBIT 1

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC2/WG8
CODED REPRESENTATION OF PICTURE AND AUDIO INFORMATION**

ISO-IEC/JTC1/SC2/WG8

MPEG 89/

October 1989

**ISO/IEC JTC1/SC2/WG8
MPEG 89/182**

Title: MCPIC - Motion Compensated Predictive-Interpolative Coding.

Object: Proposal for MPEG Coding Standard

Author: D. Le Gall, Bellcore



Date: **October 1, 1989**

From: **C. T. Chen**

Subject: **MCPIC: Bellcore's ISO/MPEG Video Coding
Proposal**

F. C. Jeng

D. J. Le Gall

K. M. Uz

A. Wong

K. M. Yang

1. Introduction

The Bellcore proposal described in this document is aimed at fulfilling the requirements of the ISO/MPEG proposal package description MPEG 89/128 while providing compatibility with the CCITT videoconferencing standard. The Bellcore proposal is actually a superset of the CCITT standard and provides more flexibility in the following ways:

- Frequent periodic reset of the temporal predictor (to provide random access)
- Interpolative coding of every second frame
- Flexibility of providing adaptive Huffman code tables
- 1/2 pel accuracy in motion estimation
- Optional quantization matrix (according to JPEG standard)

The basic algorithm is a hybrid predictive-transform coding loop with motion compensation nearly identical to the CCITT Reference Model 8 (RM8), this hybrid coding loop is used to code the original at 15 frame/s, the additional frames are coded with motion compensated interpolation and additional DCT coding of the prediction error. The required additional performances are fulfilled quite painlessly within this framework. Because the coding algorithm relies both on motion compensated prediction (MCP) and motion compensated interpolation (MCI), the proposal was named MCPIC (Motion Compensated Predictive/Interpolative Coding).

2. Description of the Proposed Bellcore Algorithm

2.1 Overview of MCPIC

In order to achieve the target compression with an acceptable quality, various techniques have been combined and the appropriate parameters optimized. The compression algorithm has two phases: preprocessing and encoding (Fig.1). Preprocessing transforms the video source signal from its original CCIR 601 4:2:2 format into a progressively scanned format

more suitable for high compression encoding. The encoder operates on a progressive format at 30 frames/sec and each frame is an array of 352x240 samples for the luminance (Y) and 176x120 samples for each of the chrominances (U & V) (Fig.2). During preprocessing, filtering and subsampling are performed on the original signal to reduce the amount of information to be coded without objectionable aliasing errors. Details of the preprocessor are given in section 2.1.

Most of the compression is obtained during the second phase: encoding. The core of the encoder is similar to the algorithm proposed by the CCITT Study Group XV for video telephony. The structure of the CCITT reference model (RM8) has been used as guidelines, and efforts made to ensure compatibility with the syntax of the CCITT Standard. To improve the compression, the hybrid predictive/transform technique of the CCITT algorithm is used on every other frame and the skipped frames are coded with an interpolative scheme (Fig. 3). In addition some JPEG ideas for quantization and coding have been adopted. The principal compression techniques used in the encoder include the 2-dimensional Discrete Cosine Transform (DCT), Motion Compensation (MC), and Variable Length Coding (VLC). Details of the compression algorithm in the hybrid coding loop are given in section 2.3. The interpolative mode helps preserve an acceptable quality even with difficult video sources like those chosen by the ISO committee, while at the same time permitting fast random access and reverse playback. Details about the interpolative mode are given in section 2.4.

The decoder is designed to be capable of decoding both the bit stream of the CCITT Study Group XV emerging standard for video telephony and the bit stream of the ISO/JPEG still image coding standard. The variable length codes of our proposal have been designed to cover at the same time the CCITT codes and the JPEG codes (both algorithms are based on 8x8 DCT). Issues of variable length codes are addressed in section 2.5. Complexity of the decoder is addressed in another document.

2.2 Preprocessing

To achieve good performance at low bit rates, the video source is presented to the coder in a modified "CIF" format instead of the original CCIR 601 format, the reduced format has roughly four times less samples. The simplest way to convert from CCIR 601 to the intermediate format is by dropping every other field in the original image, and performing decimation and filtering only in the horizontal direction. The resulting "CIF" sequence will suffer from aliasing in vertical and temporal directions (Y-T space) and show annoying artifacts.

In order to reduce the aliasing effect and the noise, a 2-D diamond-shaped lowpass filter is applied to the CCIR 601 sequence in the Y-T space before subsampling. Although the linear filtering results have less aliasing in the Y-T space, the spatial resolution has also been reduced significantly when rapid motion is present and this can be visually unpleasant. In order to improve the sharpness of the image, we modified the filter slightly by introducing a non-linearity. Based on the local activity of the images, i.e. the absolute value of difference between original data and linear filtering data, our pre-processed picture elements could be one of these three possibilities: the original data, a weighted sum of the original data and the linear filtering result, the linear filtering result alone.

First we choose two thresholds, T_1 and T_2 where $T_1 < T_2$. If D , the absolute value of difference between original data and linear filtering data, is less than T_1 , then the linear filtering result is used. This step is to reduce the aliasing effect and noise. If D is higher

than T2, then the original data is kept. This step is to keep the sharpness of the image. Finally, if D is located between T1 and T2, then the linear weighted sum of the linear filtering data and the original data is used. This step is to smooth the transition region and hence reduce the visibility of false contours.

By trial and error, the values for two thresholds are set at 30 and 70 in our current implementation. This nonlinear filter has been applied to several image sequences and the experimental results show that aliasing effect and noise has been reduced significantly while good image quality can still be obtained.

2.3 The Hybrid Coding Loop

The hybrid coding loop shown in Figure 4. summarizes the core of the compression algorithm. This hybrid coding loop is based on temporal predictive coding with motion compensation and transform coding. A simple mechanism in the loop allows the encoder to switch between the intraframe and the interframe modes. The intraframe mode, represented by the data path in bold face, realizes data compression within one frame, i.e. compression along the spatial dimension. The intra-mode is used to encode the first frame in a scene and during periodic reset of the prediction loop. The interframe mode, represented by the fine data path, realizes data compression between two consecutive frames, i.e. compression along both the spatial and temporal dimension. This mode is used to code the rest of the video sequence where the decoded previous frame is used as the prediction for the current frame, and only the change between the two is encoded. The encoding procedure in either mode becomes clear by following the block diagram in Figure 4.

2.3.1 Intraframe Coding

The basic processing unit in the encoder is a data structure called macro block. The macro block has a 16x16 luminance block and two 8x8 chrominance blocks. Figure 2 shows how a frame is partitioned into various data structures. The encoder processes a video sequence frame by frame, and within each frame, macro block by macro block. In the intraframe mode, each macro block (four 8x8 subblocks of luminance and the two corresponding 8x8 blocks of chrominance) of the current frame is processed using an 8x8 Discrete Cosine Transform (DCT). The result of the DCT is a 2-dimensional array of coefficients for each subblock. The DC coefficients (the first coefficient of each subblock) are each represented by a byte, and the AC coefficients are quantized by a matrix of visually adapted uniform quantizers (Figure 5). The quantizer matrix for luminance is finer than that for the chrominance because the human vision system is less sensitive to chrominance detail. The output of the quantizer is then entropy coded, multiplexed with other side information, and stored in the output buffer. The output buffer empties itself periodically to model a constant output. It is important to faithfully code the intraframe picture such that the interframe pictures which are coded based on the reconstruction of the previous frame may have better predictions. The feedback path at the output of the quantizer serves to reconstruct the currently encoded frame. This decoded frame is stored in a buffer memory for comparing with the next input frame in the interframe mode.

2.3.2 Interframe Prediction

The filtered input signal is compared with the reconstructed picture in the frame memory via the motion estimator/compensator. The motion compensator maps the input macro block within a search window in the frame memory to locate an image block that matches closely with the input. The best match is determined by the minimal absolute block difference. The range of the search window allows a maximum of +/- 15 pixel displacement. The prediction

accuracy is further enhanced by a half pixel precision after an integral pixel full search; therefore, the maximum allowable displacement is ± 15.5 pixels. The motion vector, indicating the displacement of the image, is differentially coded according to a VLC with the prediction being the motion vector of the previous macro block. The amplitude of the differential motion vector is limited to ± 15 . After the best match is located and filtered, a decision is made whether this best match is good enough (measured by a variance criterion). If the match is poor, the encoder switches to intraframe mode to process this particular macro block. Otherwise, it is used as the prediction for the current macro block.

The prediction error is taken to be the difference between the input signal and the best match from the motion compensation. This prediction error is DCT coded. Variable thresholding keeps the most significant 20 AC coefficients and sets the rest of the AC coefficients to zero in each 8x8 subblock. The result is quantized by the adjusted quantizer matrix, and afterward entropy coded. The quantized prediction error is again decoded, and added to the previous prediction to update the predictor in the frame memory.

2.3.3 Loop Filter

As in the CCITT px64 standard, a loop filter is used, however in consideration of the increased accuracy of the motion estimation (half pixel), it was found that there was no need to use a filter with as strong a low pass characteristic, thus the loop filter used (LF(2)) is a compromise between the filter of the CCITT px64 coder (LF(1)) and no filter at all:

$$LF(2) = 1/2 [1 + LF(1)].$$

2.3.4 Rate equalization

While this is by no mean mandatory for the MPEG proposal, Bellcore's team was concerned that the encoding could be performed in a relatively unsupervised way. While we recognize that a fixed quantizer could be used throughout a scene, and iteratively adjusted until the desired bit-rate is obtained, it was considered interesting to investigate more automatic procedures.

In both modes (intra, inter), a buffer rate strategy is used to control the bit rate and quality tradeoff by dynamically adjusting the input filter and the quantizer parameters. If the percentage of fullness in the output buffer is high, the buffer will signal the input filter to reduce the signal bandwidth by low-pass filtering the incoming data, and will increase the stepsize of the quantizer matrix to reduce the instantaneous bit rate. The percentage of buffer fullness determines the shape of the low pass filter and the stepsize, i.e. the multiplicand of the quantizer matrix. Since intraframe coding cannot take advantage of the temporal correlation, it takes a larger amount of bits to obtain a picture of acceptable quality. Therefore, there is a peak in the instantaneous bit rate during intraframe coding. The frames following an intraframe mode are likely to experience a drop in quality due to the rate buffer strategy. To preserve a relatively constant quality, the output buffer is split into an intraframe buffer and an interframe buffer. The rate buffer strategy is applied to the interframe buffer only. In this way the excess number of bits needed for the intraframe modes is shared equally among all frames in the sequence.

2.4 Interpolative Mode

The hybrid coding loop is used to code every other frame in the source i.e. 15 frames/s. The remaining frames are interpolated at the decoder, using additional information sent by the encoder. The decision of transmitting the interpolative motion vector as side information was based on the following reasons.

- Need for a high quality motion vector field (smoothness and precision).
- Difficulty in sharing a common motion vector for prediction and interpolation (inversion problem).
- Desire not to impose any particular motion estimation scheme as part of a standard.

The process of interpolative coding can be divided into three parts:

- Motion estimation
- Interpolation
- Coding of interpolation error and displacements.

2.4.1 Motion Estimation and Coding of the Motion Field

The motion estimation algorithm is based on a restrictive motion model. The image is assumed to be composed of rigid objects having a translational motion on a plane. Furthermore, the velocity is assumed to be constant between three frames. The goal is to obtain a displacement field that resembles the "true" motion, unlike the motion estimator in the predictor loop, which simply finds a "good predictor" for a given block.

A hierarchical block-matching scheme [3] is used to compute the displacement field. The sequence is spatially lowpass filtered and subsampled by two in each dimension. This is repeated to obtain two lower resolution versions I_2 and I_1 from the original frame I_0 . A full-search block-matching is performed on I_2 . The block size is 8×8 and maximum displacement is 3 in each direction.

The next step is computing the displacement field for I_1 . The displacements from the lower level I_2 are used as initial estimates for I_1 after a resampling by bilinear interpolation. The procedure is repeated to obtain the final displacement field for I_0 . The block size and the maximum displacement are kept constant throughout.

The full-search block-matching algorithm is slightly modified for interpolation. A block b is compared to a block displaced by $+d$ in the previous frame, and to a block displaced by $-d$ in the following frame. The displacement that minimizes the total displaced block difference is chosen as the optimal displacement. It is observed that constraining the allowable displacements in this manner results in a smooth motion field with high temporal and spatial correlation.

2.4.2 Interpolation

The interpolated frame is obtained by displacement and averaging of the previous and following frames. A special case occurs at the frame boundaries and is of particular relevance for images exhibiting a global motion (zoom and/or pan). A block close to the boundary may be interpolated outside the previous or the following frame. In that case, only the other frame is used for interpolation. This is very useful in a scene consisting of pure pan, since the frame can be interpolated almost perfectly. Note that the estimation scheme has also been modified for this special case.

2.4.3 Coding of interpolation error and displacements

Motion compensated Interpolation is, in general, not perfect. Therefore an additional signal, interpolation error, must be transmitted for a higher quality image. However, this error signal differs from the prediction error in the hybrid coding loop in a significant way. While the prediction is of IIR nature and any past errors tend to accumulate, interpolation is of FIR nature and errors do not propagate. Thus, a larger degree of freedom is allowed in coding of the interpolation error.

Prior to coding, the interpolation error is thresholded (actually masked) based on a heuristic principle: distortions are more likely around the edges of a moving objects, and are likely to be visually more objectionable. Therefore, if the local gradient of the motion field exceeds a threshold, the adjacent blocks are coded. Otherwise, they are set to zero. This procedure eliminates coding small distortions in texture which are hardly visible. The remaining signal is coded using DCT. The threshold is a choice parameter for rate control. In fact, the interpolation error can be simply discarded to prevent a buffer overflow. In view of the very stringent requirements of coding the test sequences at less than 0.9 Mbits/s, the interpolation error was never actually transmitted when coding the test sequences.

The motion vectors corresponding to a 8x8 block are DPCM coded; The DPCM loop is initialized at the first vector at the upper left of the frame and the predictor is given by the motion vector of the previous block. The differential vectors are coded by a Variable Length Code, typically requiring 1.2 to 4.3 bits per vector. The horizontal and vertical displacements are coded with a 2-D VLC with 1024 symbols. The less likely event are coded with a prefix and PCM codes into a code word that does not exceed 16 bits. Typical bitrate for the displacement is thus 3.5 Kbits/frame.

2.5 Entropy Coding

The symbols resulting from the coding phase include:

- The nonzero amplitudes and run-lengths of zeros for the quantized DCT coefficients,
- The differential motion vectors (DMV) for motion-compensated macro blocks
- Macro block type in the interframe coding,
- Quantizer index for each group of blocks (GOB),
- Run-lengths of fixed macro blocks in the intermode

The quantizer indices and the intraframe replenishment frequency are PCM-coded. The DC coefficients for intramode MB's are allocated 8 bits (PCM) to ensure the intensity continuity between adjacent blocks. All the other DCT coefficients are coded with a 2-D variable length code (VLC), where each nonzero amplitude is associated with a preceding run-length of zeros, and jointly they are treated as a single symbol.

The differential motion vectors are VLC-coded, with VLC optimized for a training set of image sequences. The macro block types are also VLC-coded, where an address relative to the preceding fixed macro blocks is sent out, in addition. The run-length of the fixed macro blocks accounts for such an relative address which is VLC-coded.

For the DCT coefficients which are encoded with 2-D VLC, two approaches were studied. The first VLC tested is the one given in the Reference Model 8 of the CCITT. The second VLC is similar to the one proposed by JPEG, where nonzero amplitudes are split into 8 groups, and then run-length of zeros and nonzero amplitude groups are combined into a 8-bit

symbol. A 1-D VLC is then generated for those 8-bit symbols; for each class of amplitude a fix number of additional bits is transmitted to specify exactly the amplitude. The advantage of JPEG's approach is that the VLC can be optimized locally according to the symbol statistics, whereas CCITT's 2-D VLC is fixed and less flexible. Some information regarding the codebook information needs to be sent along when using the JPEG's approach. In practice, such an overhead requires at most 2.2 kbits per scene (in practice much less), and is therefore trivial. The proposed decoder will be able to decode both the CCITT and the JPEG variable length code.

Simulation results have shown that using the JPEG's VLC approach results in a substantial savings on the bitrate, as compared with the one using the CCITT's 2-D VLC. This is due essentially to the scene adaptive nature of the JPEG VLC.

3. Additional Performances

The MCPIC coder is designed with the additional performances of document MPEG 89/128 in mind. The "Group of Frames" is a data structure for stored video that allows a satisfactory level of flexibility in Random Access and Reverse Playback. A group of frame starts with an intra-coded frame: such break in the interframe dependency makes random access possible and at the same time provides additional robustness to channel errors. In a group of frame all frames beyond the first are either predictively or interpolatively coded. The set of frames coded predictively constitute a video sequence at half the nominal frame rate and this reduces by a factor of two the amount of buffer required for reverse playback. A graphic representation of a "Group of Frames" is shown in figure 6. If a new scene is detected prior to the 15th frame, the encoder starts a new "Group of Frames".

3.1 Reverse Playback

The interpolative mode, together with the periodic reset to the "intra" mode allows reverse playback with a minimum amount of buffer and small delay. In our proposal the number of frame in a "Group of Frames" is 15. If a "intra" mode is invoked every 15 frame, and since the interpolation information is bidirectional, the amount of buffering necessary to play in reverse is the memory required to store the initial intra-coded frame the 7 predictively coded frames plus one additional frame memory to store the result of the interpolation, i.e. a total of 9 frames which amounts to less than 9 Mbits. Timing is critical in decoding a video sequence for reverse playback, more timing details are given in the attached document: "Demonstration of VLSI Implementability".

3.2 Fast Searches

Fast search is achieved by playing the sequence at a reduced frame rate and decoding only the intraframe coded pictures at the decoder's end. Fast search is essentially constrained by the amount of information to be read from the storage medium since the intraframe coded frames require significantly more bits than when motion compensated prediction or interpolation is used. Typically the number of bits required by an intra-coded picture is about 100 kbits. When the amount of information to be read from the storage medium is about 1 Mbit/second, AT MOST TEN INTRA-CODED FRAMES CAN BE DECODED EVERY SECOND. For a segment of 10 seconds, if a speed up factor of 10 is required, then, only ten frames of the segment can be displayed. This suggest a frame repeat factor of three. Because the amount of bits required by the intra-coded frames is highly variable, the

decoder will be forced to skip frames to account for excess data. A simple algorithm for fast searches with a speed up of 10 is then the following: while the cumulative bit rate is below a certain limit display the intra-coded frame of every second "Group of frames" three times, if the cumulative bitrate exceeds a threshold skip the next intra-coded frame and repeat the current one (three times).

3.3 Random Access

Random access is achieved by periodic reset to the intraframe coding mode. Since intraframe coding does not rely on the previous frame for decoding, a frame coded in this mode is readily accessible. The degree of random accessibility depends on the frequency at which the intraframe mode is invoked. When a frame coded in interframe mode is requested, the decoder will start decoding from the last frame coded in intraframe mode until the frame of interest is reached. Presently our encoder invokes the intraframe mode once every 15 frames (Group of Frames (GOF) see figure 6). The amount of time to reach a given frame is given by the amount of bits representing the predictively coded frames of the GOF anterior to the given frame. The worst case Random access time is thus given by the time it takes to read the entire predictive part of a GOF structure.

3.4 Still Image Mode

Coding a frame of full resolution CCIR 601 images is required for still image mode. Since the human visual system is not extremely sensitive to the color details, the resolution of the chrominance components is reduced, before compression, by a factor of four, both horizontally and vertically, as compared with that of the luminance component. The coding algorithm is a DCT-based algorithm similar to the video intraframe coding in the above and essentially follows JPEG. Specifically, the difference image between the original and the interpolated (enlarged) decoded image from the video coding is first decomposed into 8x8 disjoint blocks. DCT is then applied to each block. The transformed coefficients are thresholded, quantized, and zigzag scanned into a 1-D sequence. The run-length of zeros and nonzero amplitudes are finally entropy-coded by using the JPEG's VLC approach. The high-resolution still image coding is constrained so as not to exceed 270 kbits.

To improve the image quality under the constraint of 270 kbits which corresponds to a 0.8 bits per pixel, some alternative of DCT coefficient thresholding and quantization is considered. Specifically, a certain number of the DCT coefficients in each block are retained, and then a quantization strategy is applied to the remaining coefficients. It is found that such an alternative gives a better visual image quality under the bitrate constraint.

3.5 Ways to provide direct transcoding of CCITT p x 64

By design of the proposed algorithm a binary stream from a CCITT px64 coder can be decoded by

- (1) setting the controller to either the quarter CIF or the full CIF format.
- (2) setting the inverse quantizer to the (default) CCITT mode, and
- (3) selecting the (default) CCITT Variable Length Code for the DCT AC coefficients: (the symbols [Run, Amplitude]).

3.6 Ways to provide direct transcoding of JPEG baseline system

Also by design of the proposed algorithm, a "JPEG baseline encoded" still image can be decoded by

- (1) Setting the controller to the appropriate (720x486 4:2:2) "JPEG" format
- (2) Loading the quantization matrices (Luminance and Chrominance).
- (3) Computing and loading the JPEG variable length codes (also used as option by this proposal)

References

- [1] "Description of Reference Model 8 (RM8)" Document 525, CCITT SG XV, Working Party XV/4, Specialist Group for Visual Telephony.
- [2] B. Girod, "Displacement Estimation by Multi-field Matching", presented to the Picture Coding Symposium, Stockholm, June 1987.
- [3] M. Bierling, "Displacement Estimation by Hierarchical Block Matching", 3rd SPIE Symposium on Visual Communications, Nov. 1988, Cambridge, Massachusetts.

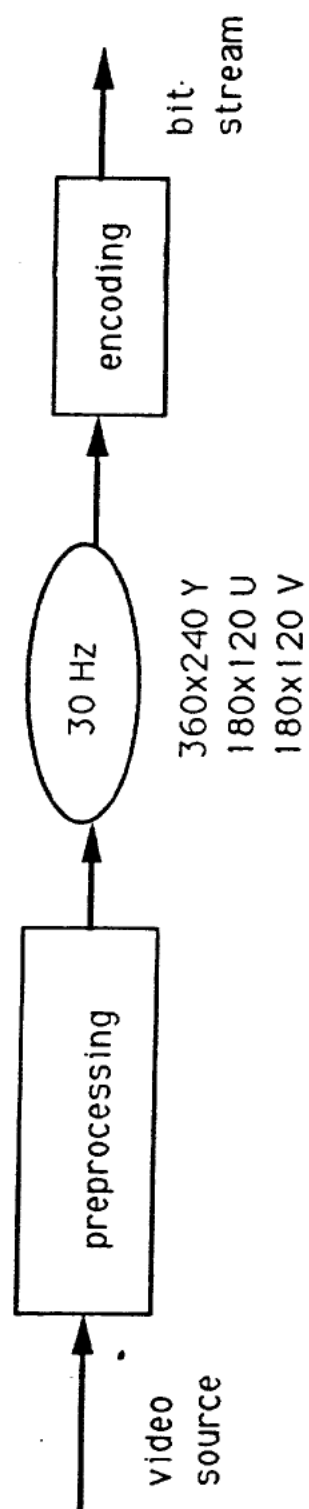
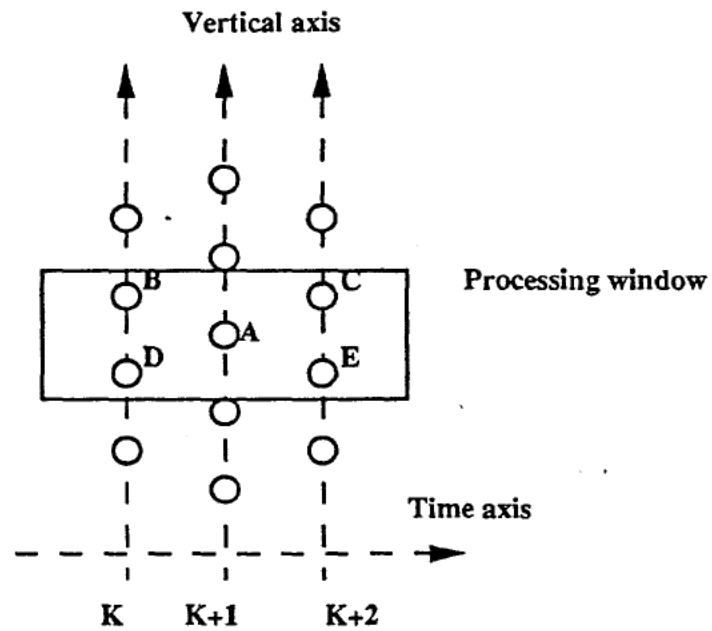


Figure 1



$$X_{lin} = 1/2 A + 1/8 (B+C+D+E)$$

$X =$	X_{lin}		if $ X_{lin} - A < T1$
	$\frac{(T2 - X_{lin} - A)}{(T2 - T1)} X_{lin} + \frac{(X_{lin} - A - T1)}{(T2 - T1)} A$		if $T1 < X_{lin} - A < T2$
	A		if $ X_{lin} - A > T2$

Figure 2: Non Linear Filter for Preprocessing

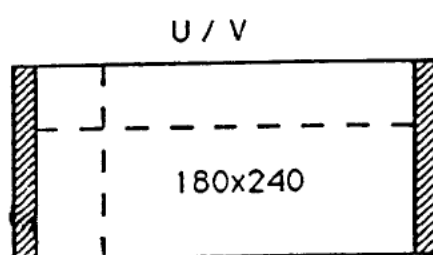
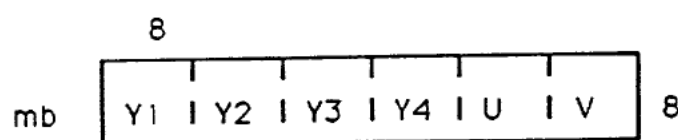
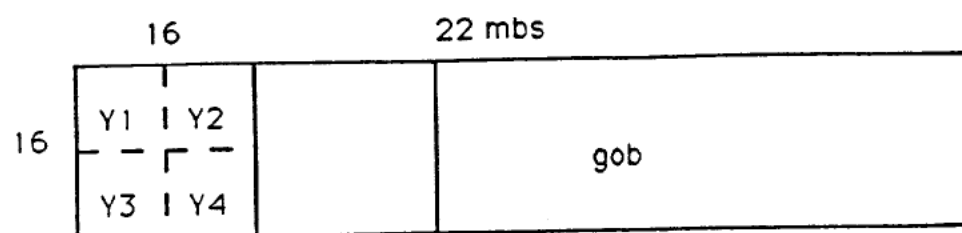
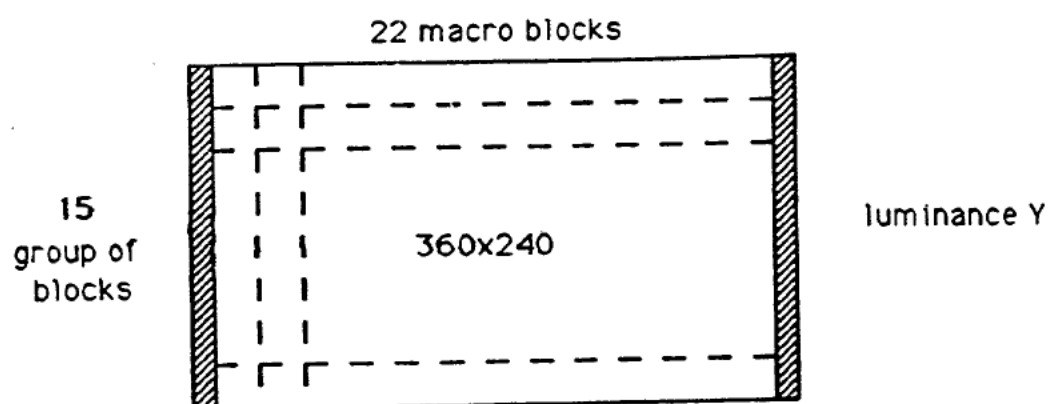
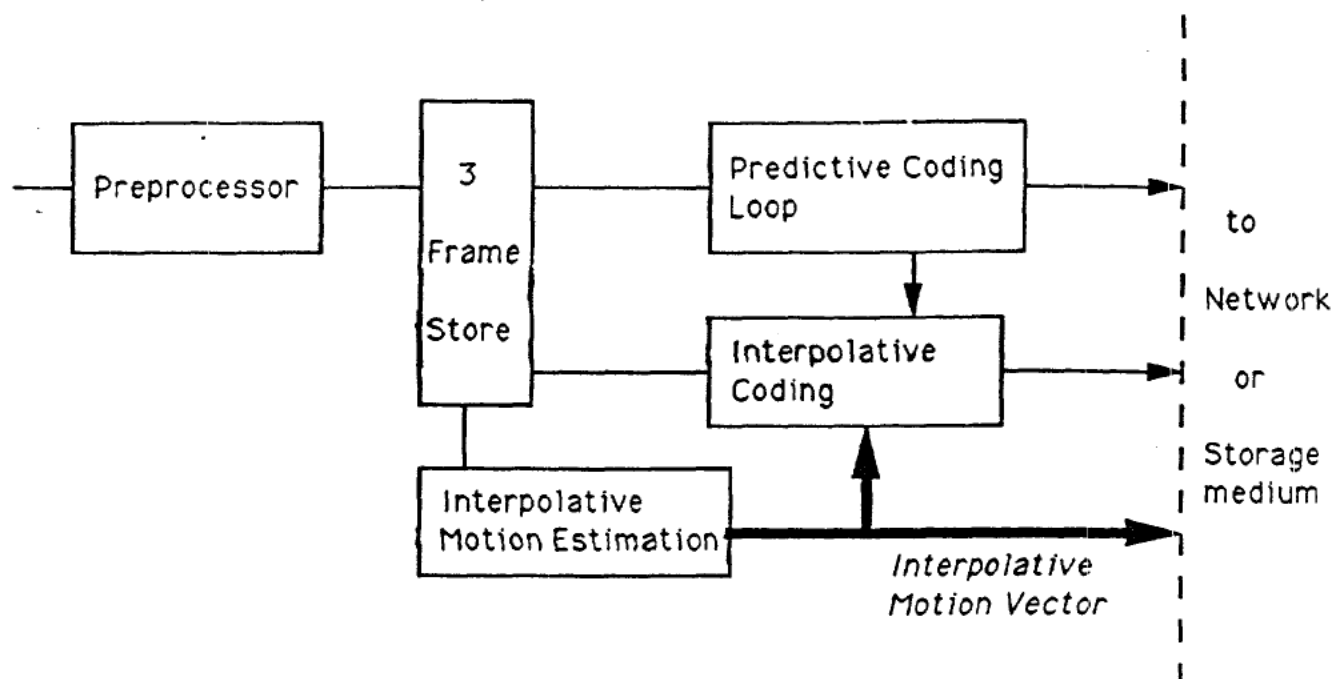


Figure 2a
Blocking of a frame



Motion Compensated Predictive/Interpolative Coding: Encoder

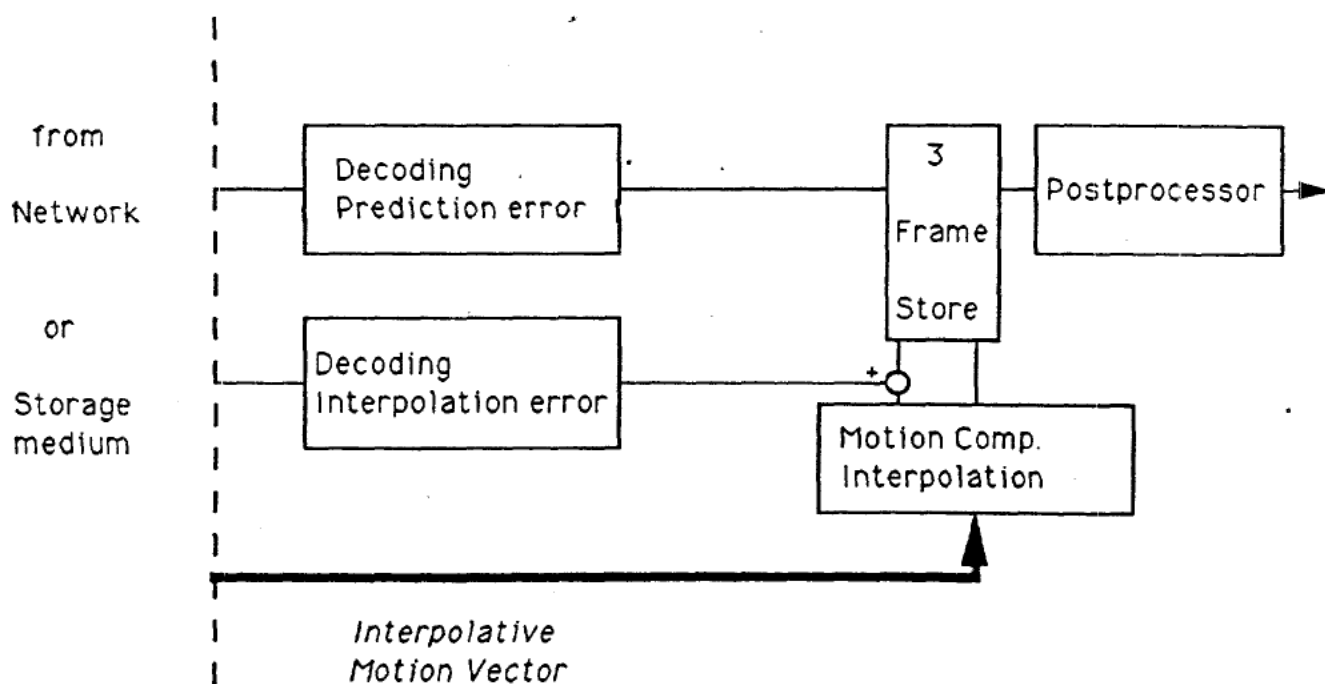


Figure 3

Motion Compensated Predictive/Interpolative Coding: Decoder

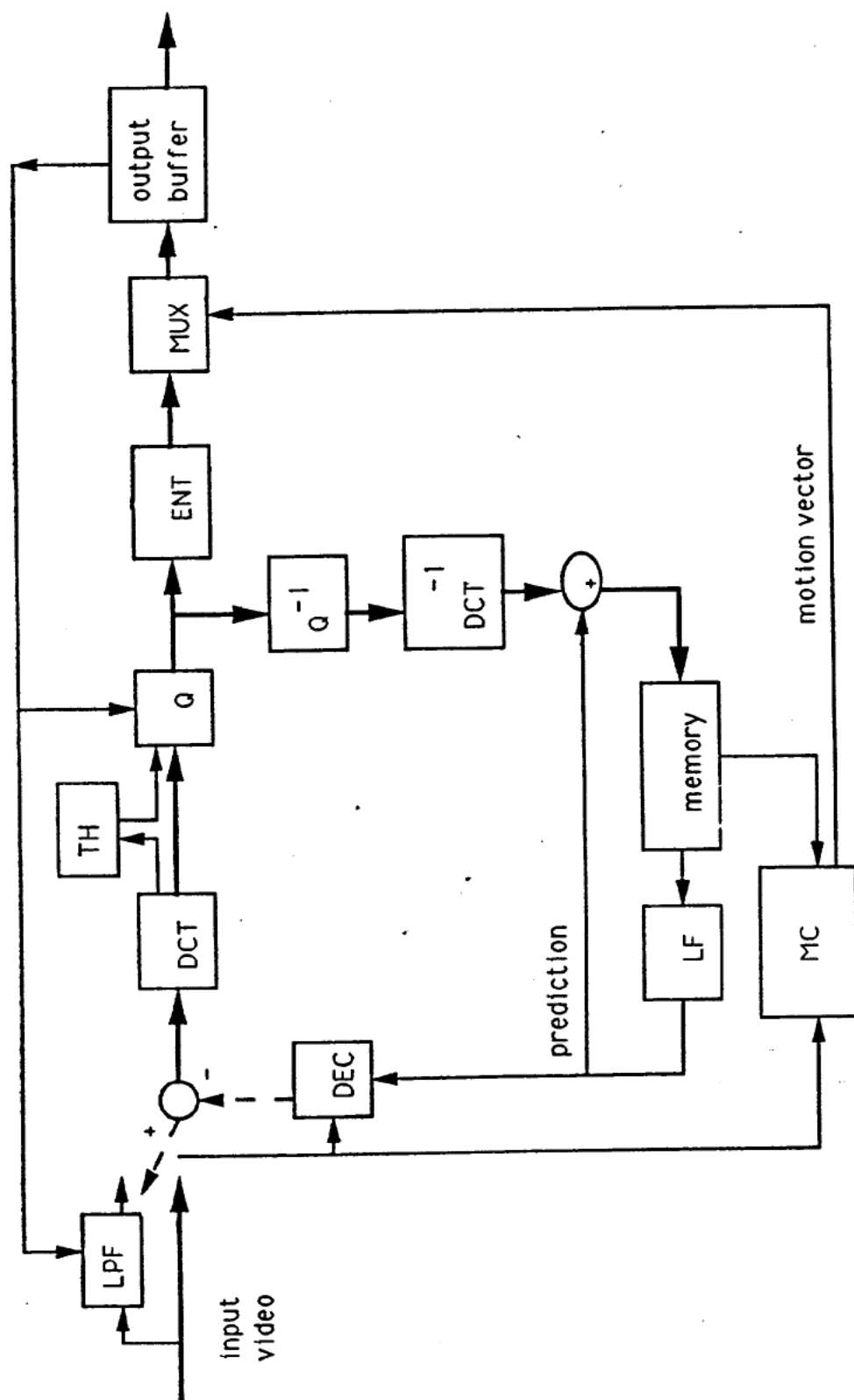


Figure 4
Hybrid Coding Loop

(a)	10	10	10	11	11	12	14	16
	12	12	12	12	12	14	16	17
	12	12	12	12	14	15	16	18
	12	12	12	13	14	15	16	19
	13	13	13	13	14	15	17	22
	13	13	13	14	15	16	20	24
	14	14	14	15	16	19	22	25
	14	14	15	16	17	21	22	29
(b)	16	18	20	20	26	36	47	47
	18	20	24	24	26	36	47	47
	20	24	24	24	36	47	50	66
	24	24	26	36	47	56	70	80
	26	26	36	47	56	70	80	99
	36	36	47	56	70	80	99	99
	47	47	56	66	80	99	99	99
	47	56	66	80	99	99	99	99

Figure 5

Quantizer matrix for (a) luminance and (b) chrominance

Predictive Coding

Intra
Frame
Coding

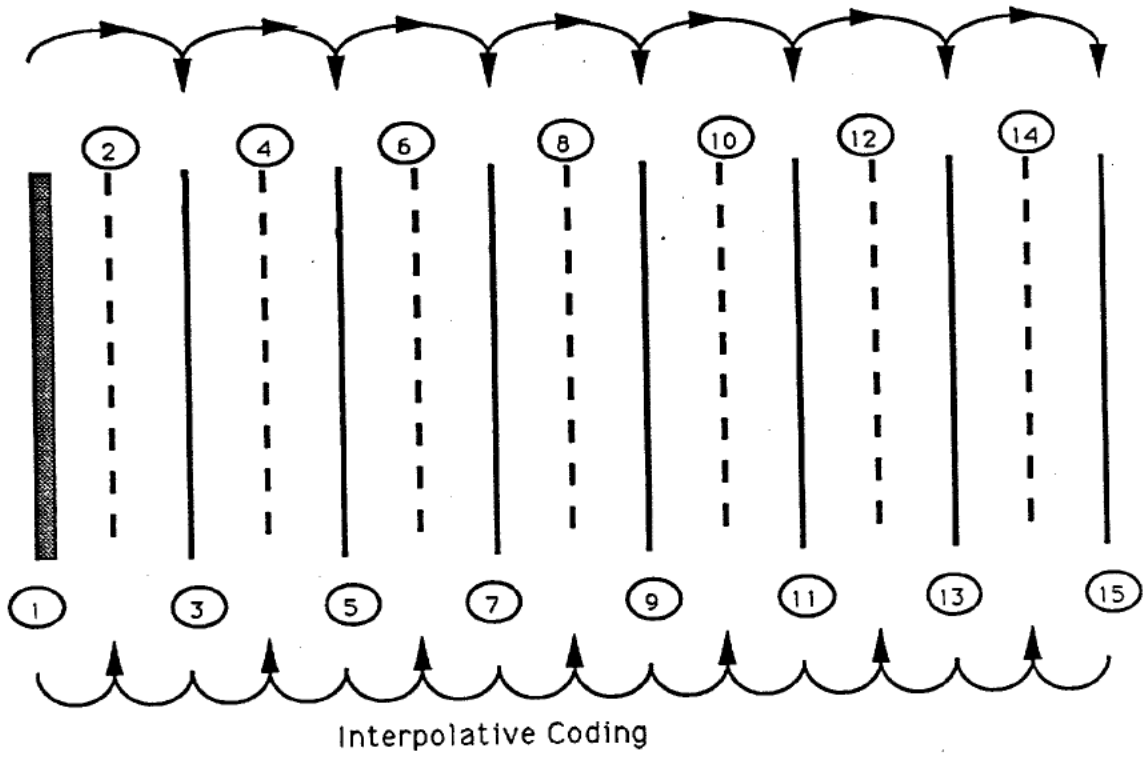


Figure 6 : A data structure suitable for Reverse Playback and Random Access

**** Macro Block Type ****

MB type	code
MC coded	1
MC not coded	01
no MC, coded	000
intra mode	001

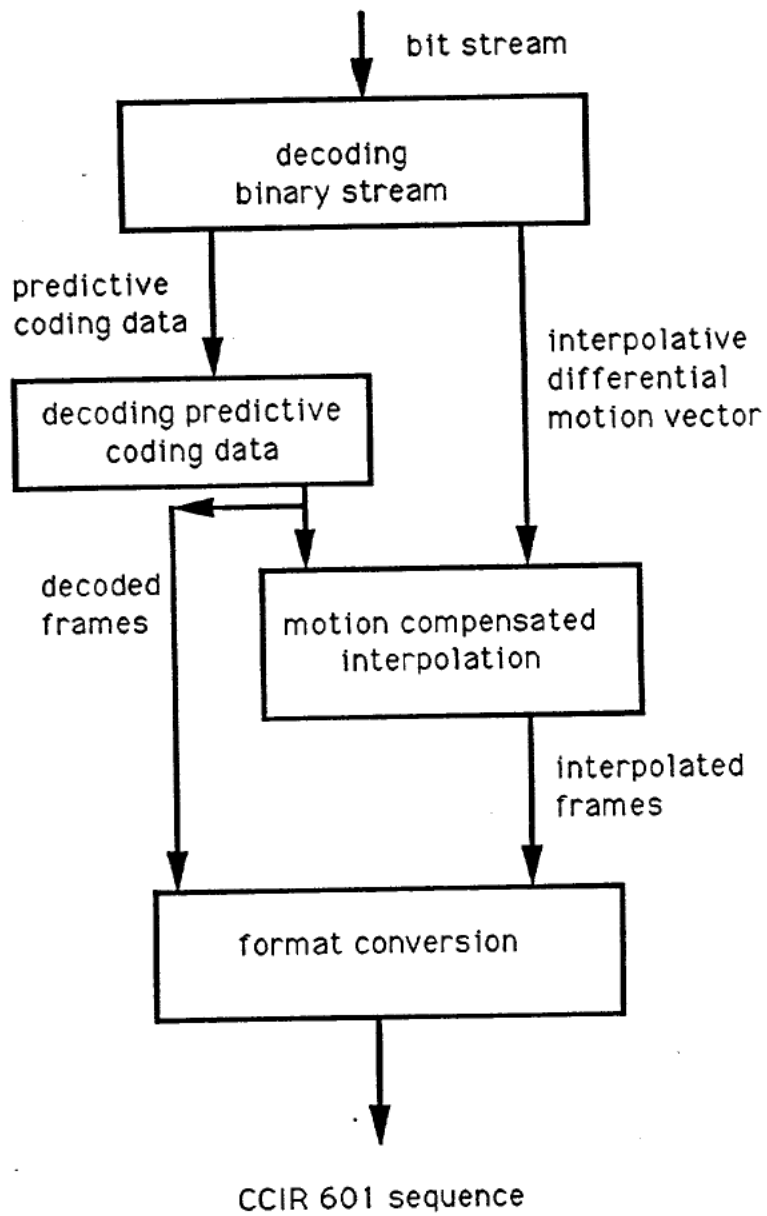
**** Run-length of Fixed Macro Blocks ****

run-length	code
0	1
1	010
2	011
3	0010
4	0011
5	00010
6	00011
7	000010
8	000011
9	0000010
10	0000011
11	00000010
12	00000011
13	000000010
14	000000011
15	000000001000
16	000000001001
17	000000001010
18	000000001011
19	000000001100
20	000000001101
21	000000001110
22	000000001111

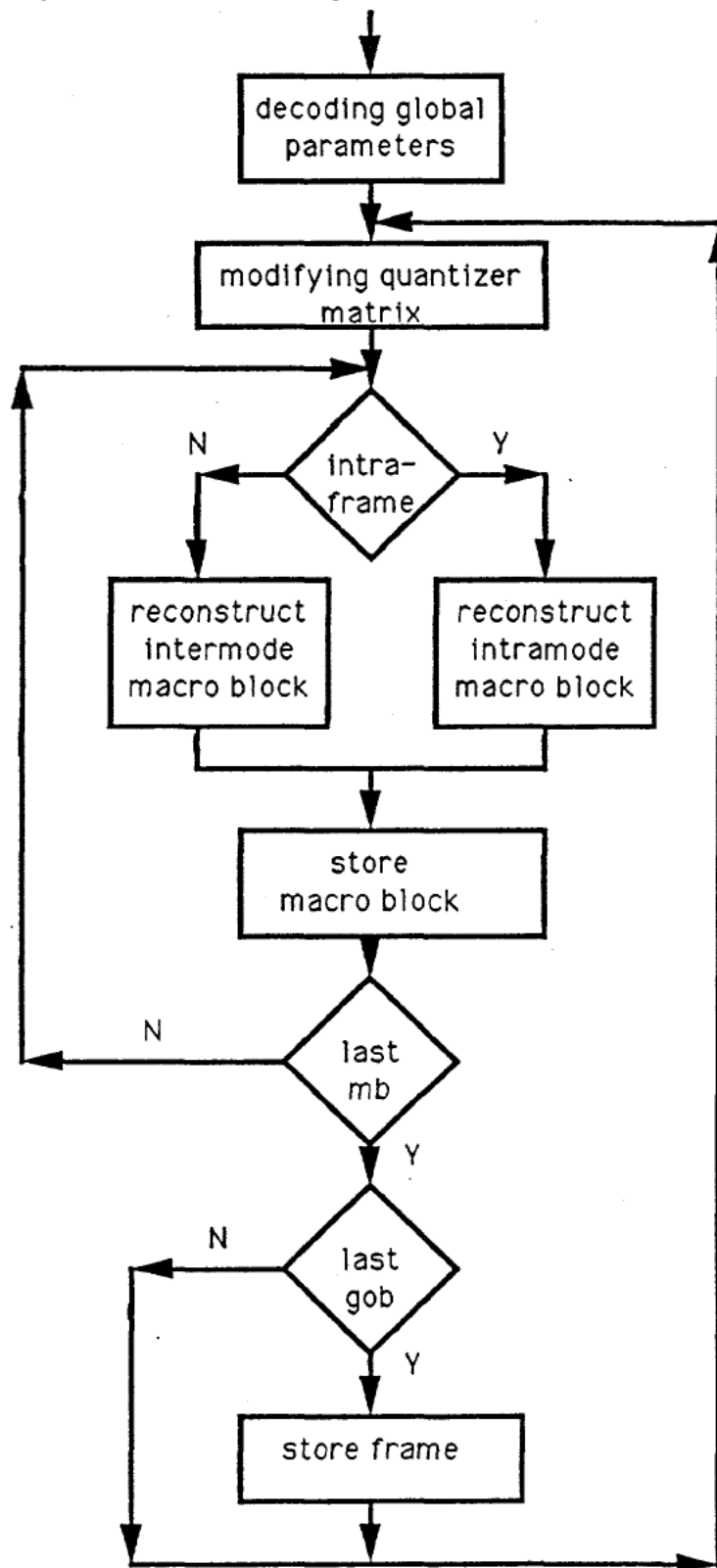
**** Differential Motion Vectors ****

DMV	code
0	00
0.5	10
1	010
1.5	11100
2	11110
2.5	01100
3	01110
3.5	111010
4	111110
4.5	110100
5	110000
5.5	110001
6	011010
6.5	011011
7	011110
7.5	011111
8	1101010
8.5	1101011
9	1101100
9.5	1101101
10	1101110
10.5	1101111
11	11101100
11.5	11101101
12	11101110
12.5	11101111
13	11111100
13.5	11111101
14	11111110
14.5	11111111
15	11001

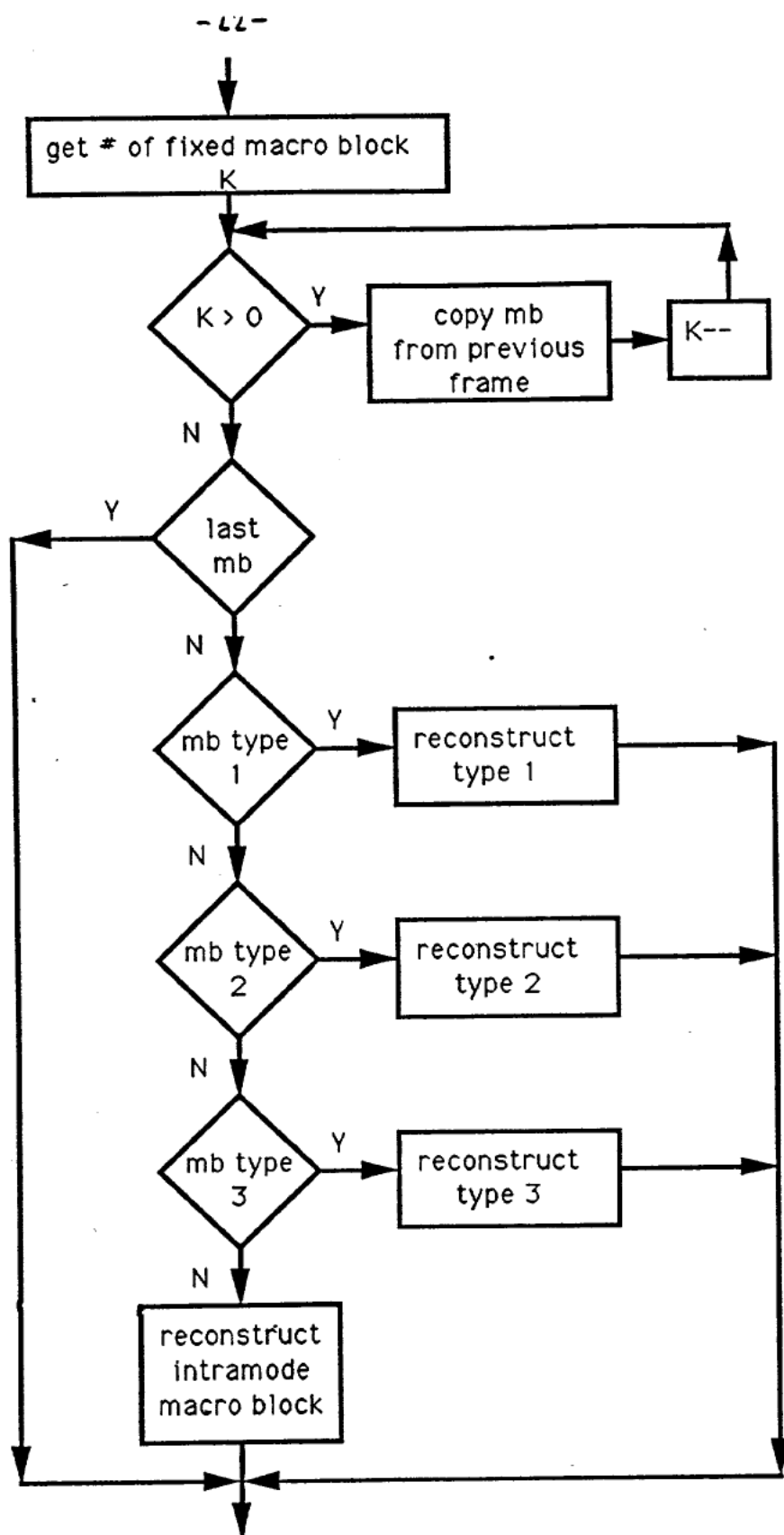
4. *Block diagrams and Flowcharts*



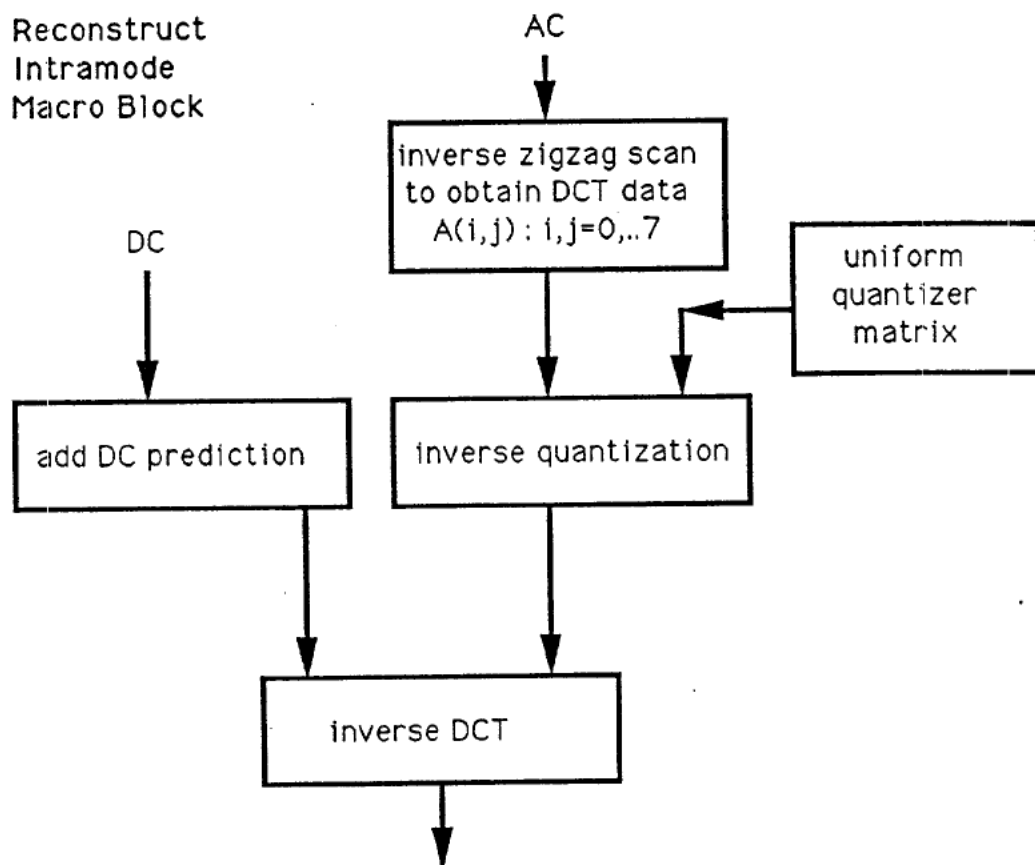
Decoding Predictive Coding Data



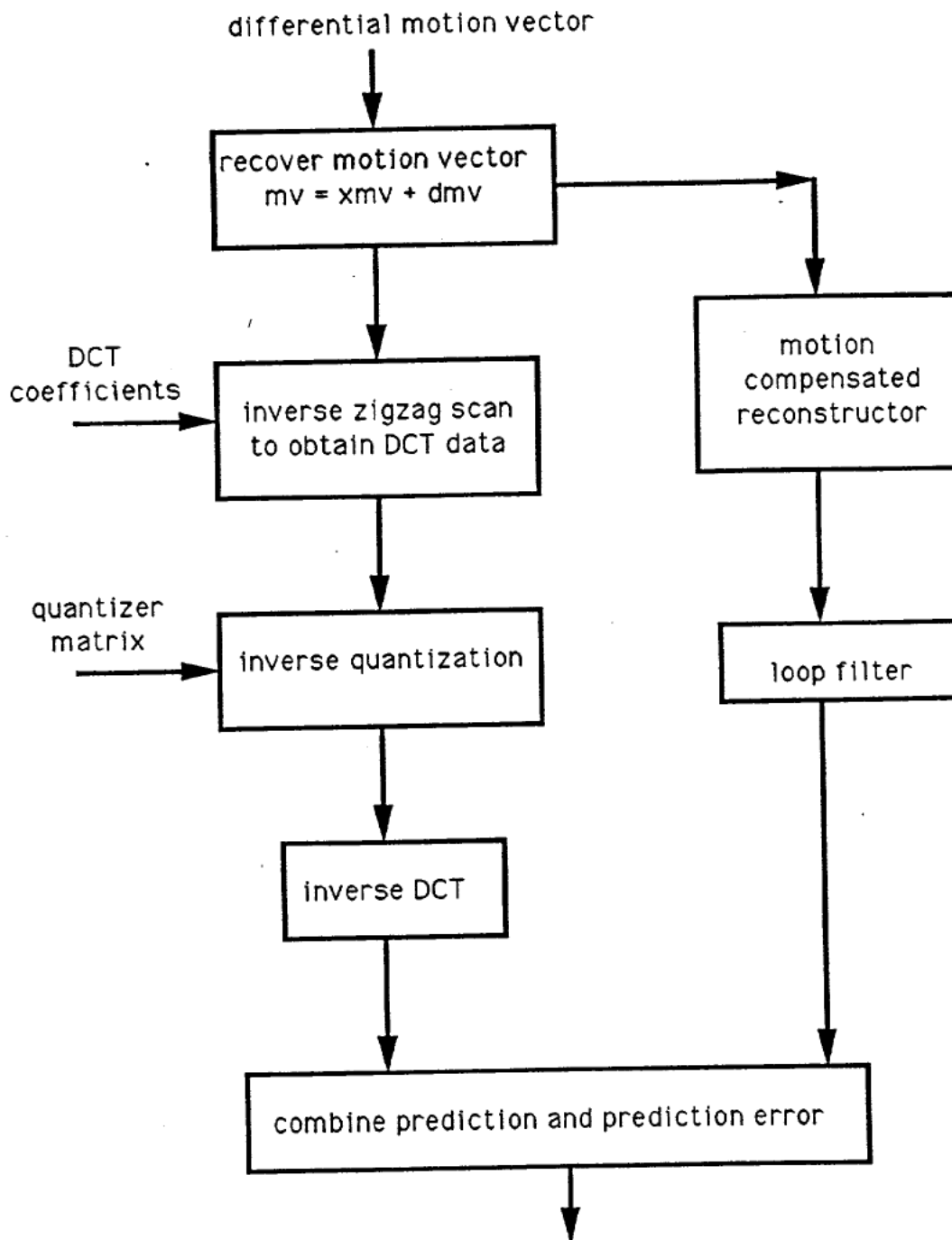
Reconstruct
Intermode
Macro
Block



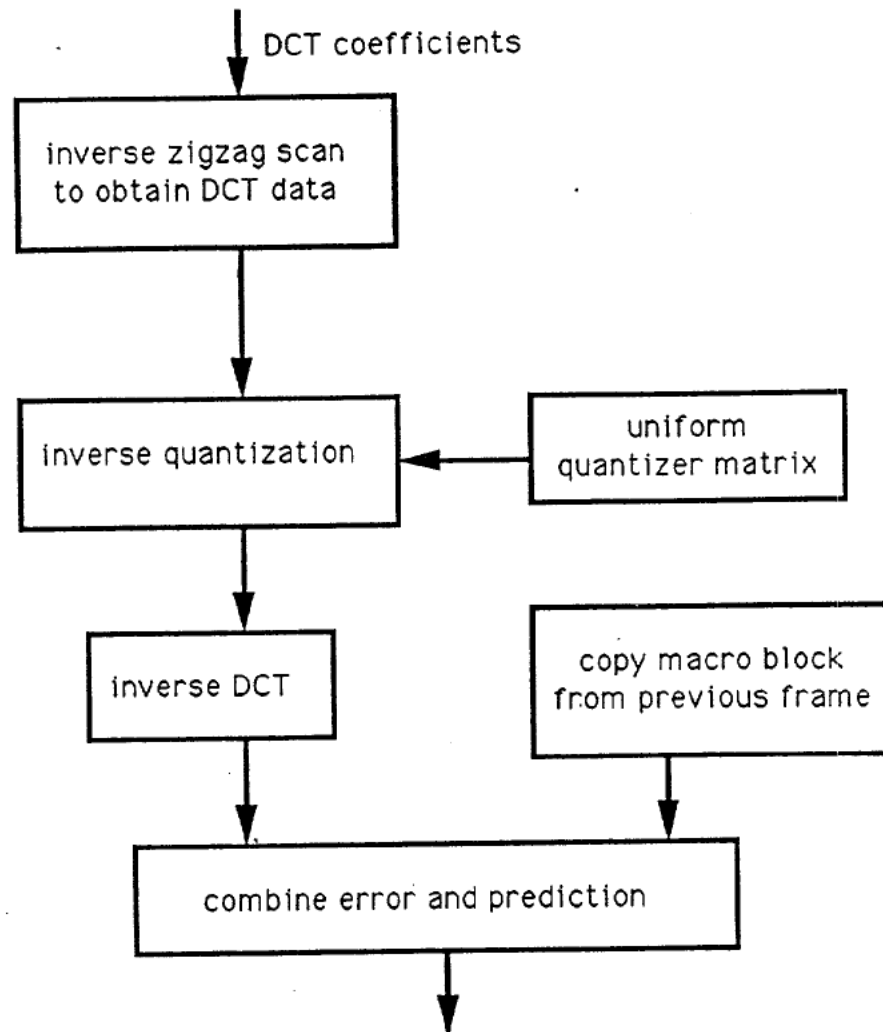
Reconstruct
Intramode
Macro Block



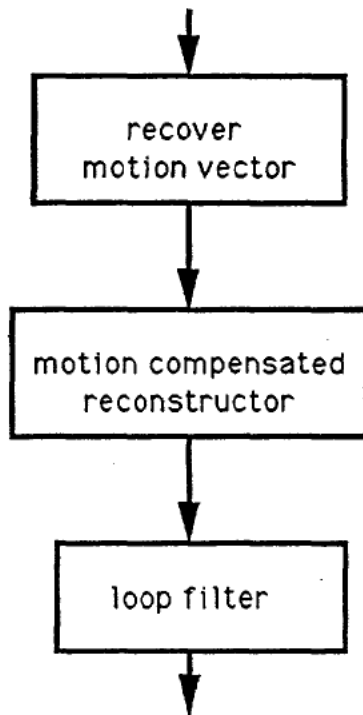
Reconstruct Type 1 Macro Block



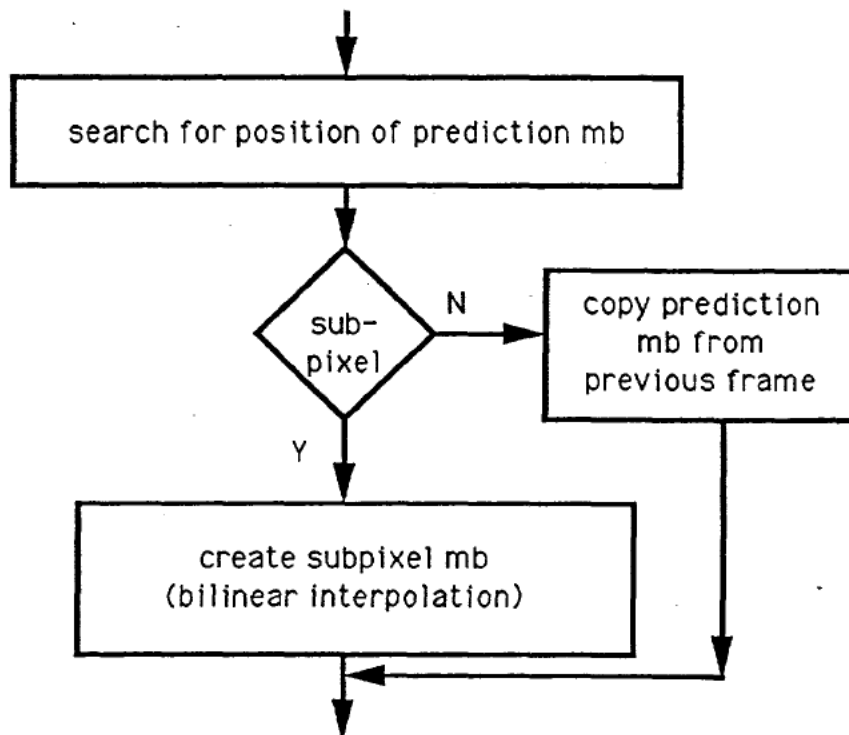
Reconstruct Type 3 Macro Block



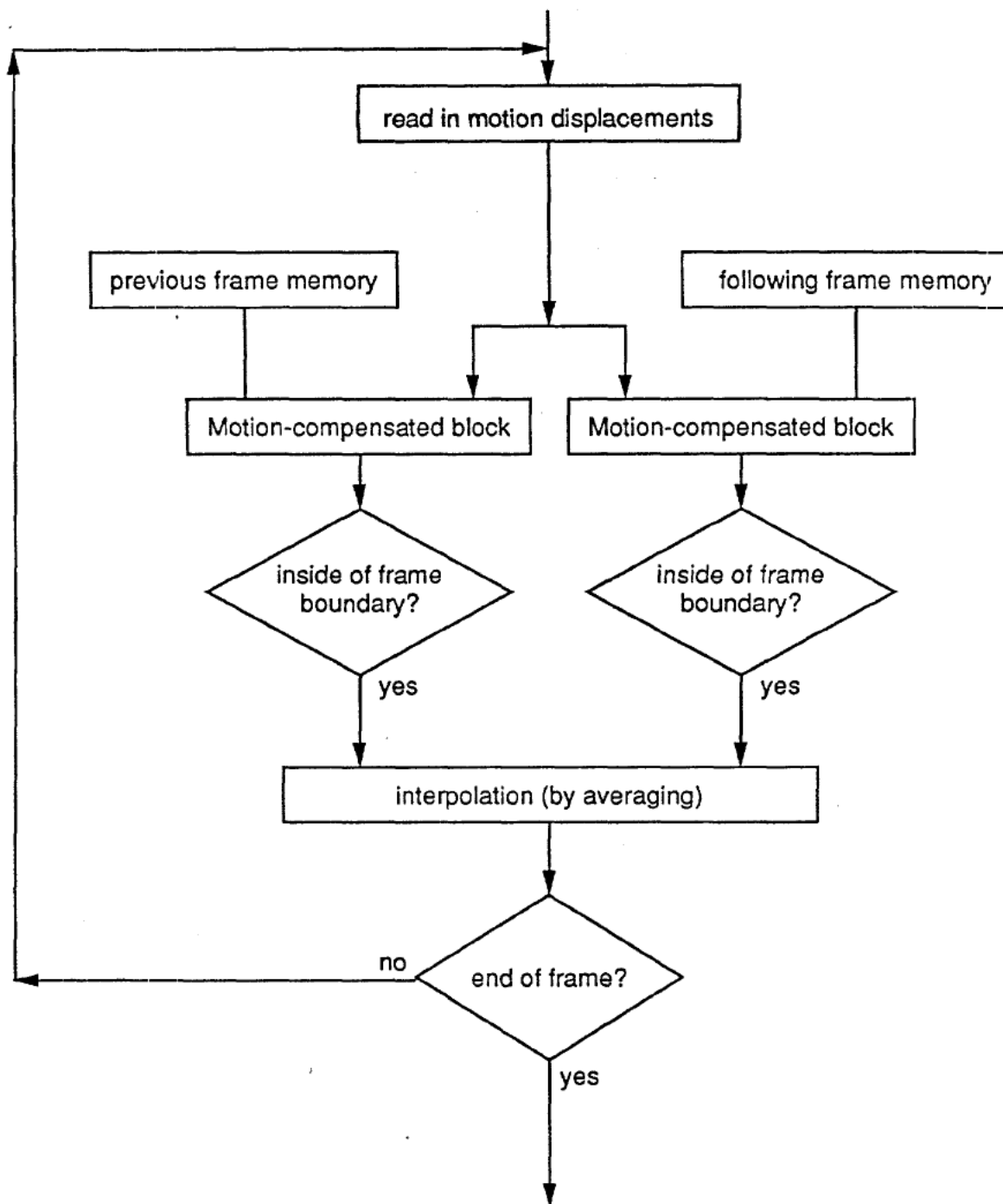
Reconstruct Type 2 Macro Block



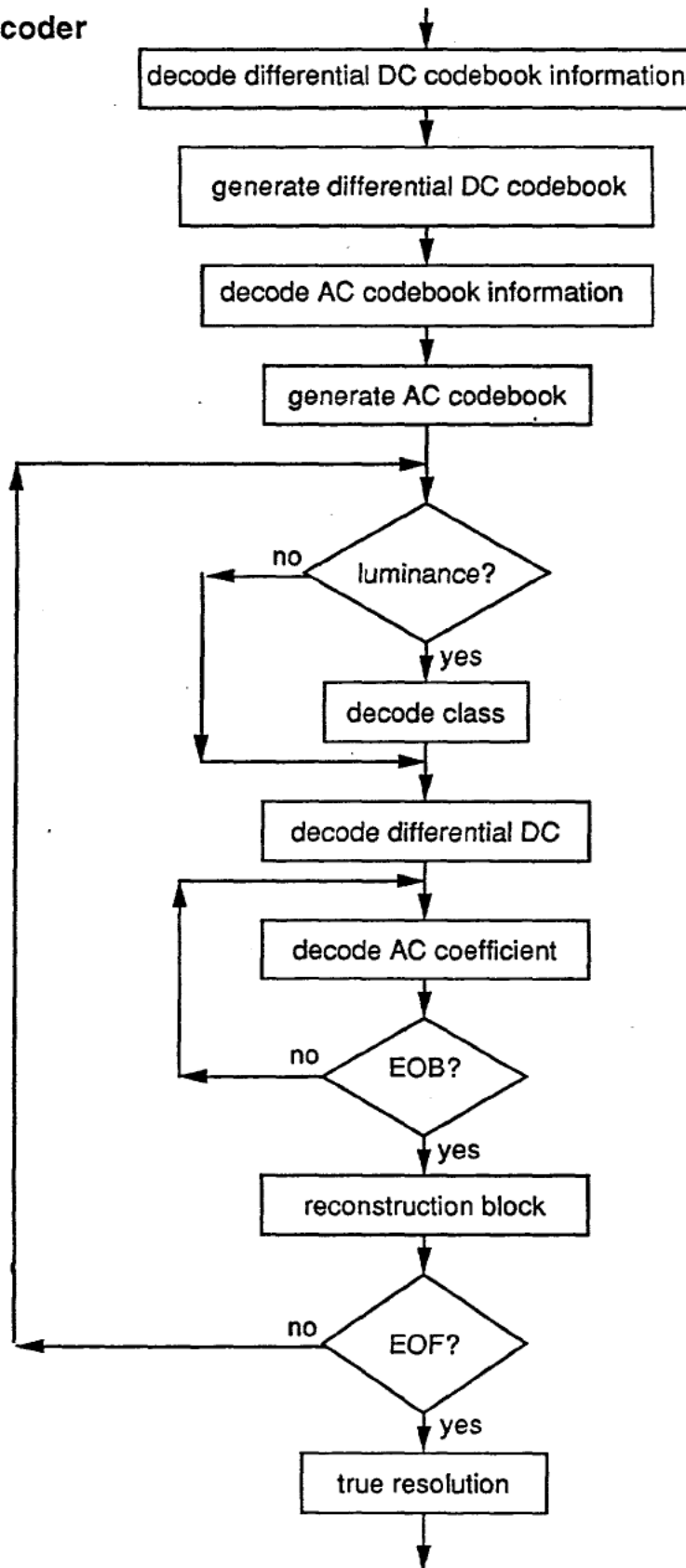
Motion Compensated Reconstructor



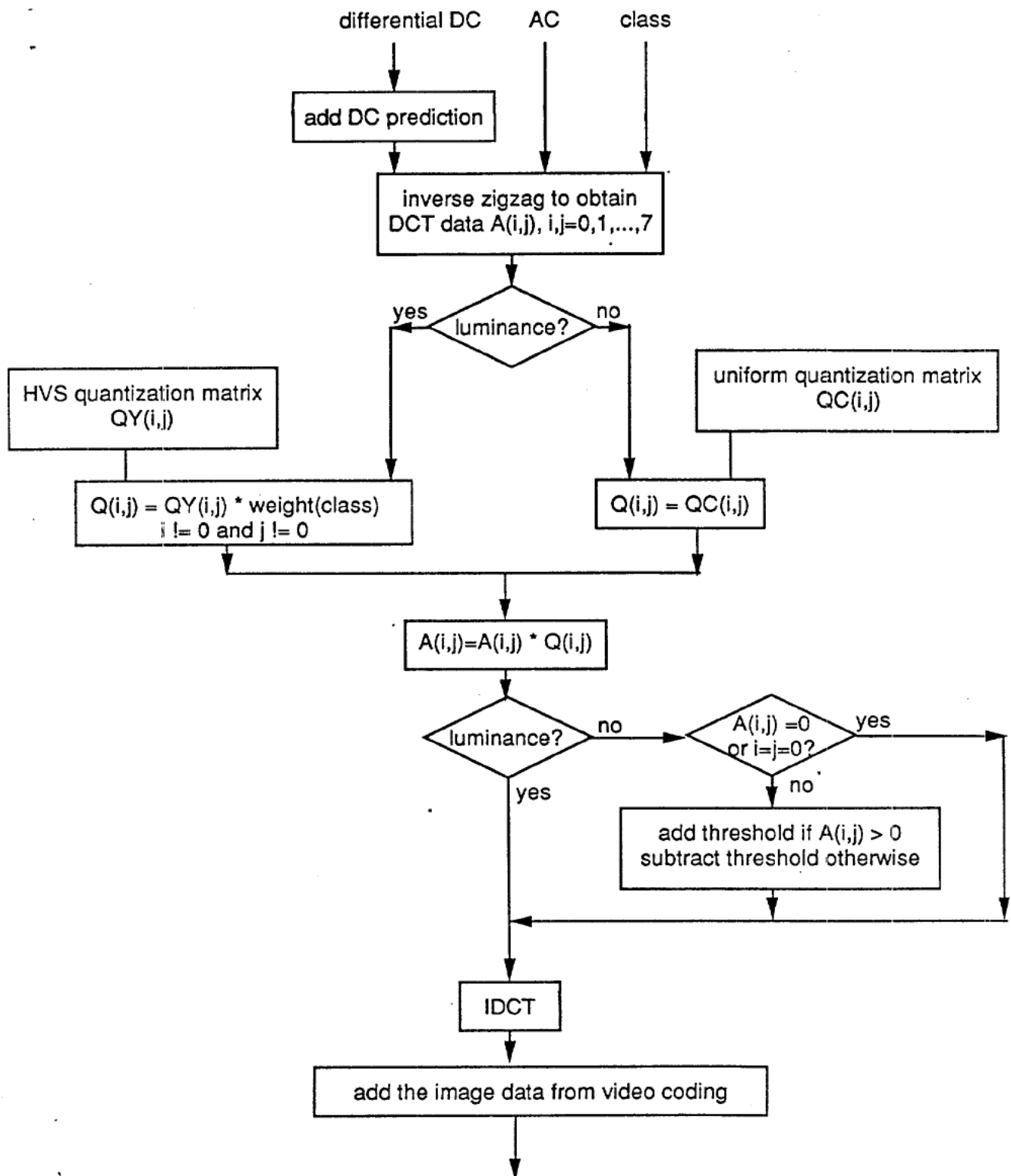
Interpolative Frame



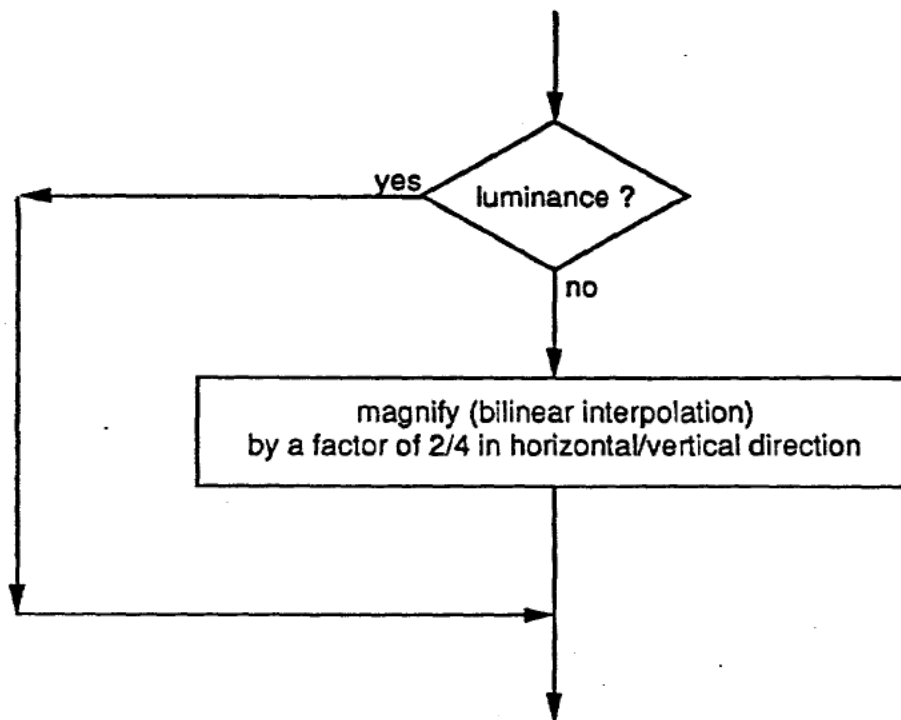
Still Image Decoder



Reconstruction block



True resolution



5. Coded Bitstream and Executable code

This file contains the exact command lines to decode the ISO coded sequences. 'decshell' is the shell script that invokes the decoding procedure step by step.

To decode the pp1 sequence (scene 1 of table tennis --- pp1.0 to pp1.88):

==> decshell dslsym.pp1.pass2.enc.b opp1 0.6 47 0

Note: The output sequence after format conversion is :

opp1.CCIR.0 to opp1.CCIR.88.

The last frame of this scene is supposed to be interpolated. Since there is no 90th frame, we must repeat opp1.CCIR.87 for opp1.CCIR.88.

To decode the pp2 sequence (scene 2 of table tennis --- pp2.0 to pp2.59) :

==> decshell dslsym.pp2.pass2.enc.b opp2 0.55 31 0

Note: The output sequence after format conversion is :

opp2.CCIR.0 --- opp2.CCIR.59.

The last frame of this scene is supposed to be interpolated. Since there is no 60th frame, we must repeat opp2.CCIR.57 for opp2.CCIR.58.

To decode the pp3 sequence (scene 3 of table tennis --- pp3.0 to pp3.151) :

==> decshell dslsym.pp3.pass2.enc.b opp3 0.55 81 0

Note: The output sequence after format conversion is :

opp3.CCIR.0 --- opp3.CCIR.151.

The last frame of this scene is supposed to be interpolated. Since there is no 152nd frame, we must repeat opp3.CCIR.150 for opp3.CCIR.151.

To decode diva sequence (diva.0 to diva.149) :

==> decshell dslsym.diva.pass2.enc.b odiva 0.55 80 0

Note: The output sequence after format conversion is :

odiva.CCIR.0 --- odiva.CCIR.149.

To decode flower garden sequence (fgpan.0 to fgpan.149) :

==> decshell dslsym.fgpan.pass2.enc.b ofg 0.55 80 0

Note: The output sequence after format conversion is :

ofg.CCIR.0 --- ofg.CCIR.149.

To decode windmill sequence (still.0 to still.29) :

=> decshell dslsym.still.pass2.enc.b ostill 0.6 16 1

Note: The last argument of the decshell command '1' invokes the still mode decoder.

The output sequence after format conversion is :

ostill.CCIR.0 --- ostill.CCIR.29

To decode one frame of CCIR resolution image, the shell script is "still.com". Given the luminance and chrominance binary stream "dlsym.CCIR.y.enc.b" and "dlsym.CCIR.uv.enc.b", respectively, and the low resolution CIF image "ostill.29" (obtained from decompression of still sequence), the command is

==> still.com dslsym.CCIR.y.enc.b dslsym.CCIR.uv.enc.b ostill 29 still.090.rec

The output will be still.090.rec.yuv.

***** binary stream files *****

rwxr-xr-x3879/3879	2580	Oct	9	15:19	1989	ds1sym.CCIR.uv.enc.b
rwxr-xr-x3879/3879	31146	Oct	9	15:19	1989	ds1sym.CCIR.y.enc.b
rwxr-xr-x3879/3879	533177	Oct	10	15:46	1989	ds1sym.diva.pass2.enc.b
rwxr-xr-x3879/3879	557247	Oct	10	15:46	1989	ds1sym.fgpan.pass2.enc.b
rwxr-xr-x3879/3879	347572	Oct	10	15:46	1989	ds1sym.ppl.pass2.enc.b
rwxr-xr-x3879/3879	156274	Oct	10	15:46	1989	ds1sym.pp2.pass2.enc.b
rwxr-xr-x3879/3879	563035	Oct	10	15:46	1989	ds1sym.pp3.pass2.enc.b
rwxr-xr-x3879/3879	112053	Oct	10	15:46	1989	ds1sym.still.pass2.enc.b

***** executdble files *****

rwxr-xr-x3879/3879	0	Oct	11	11:20	1989	exesun4/
rwxr-xr-x3879/3879	24576	Oct	9	15:28	1989	exesun4/aastilldecl
rwxr-xr-x3879/3879	24576	Oct	9	15:28	1989	exesun4/abstilldecl
rwxr-xr-x3879/3879	24576	Oct	9	16:21	1989	exesun4/abyuvfr
rwxr-xr-x3879/3879	24576	Oct	9	15:35	1989	exesun4/btoa
rwxr-xr-x3879/3879	32768	Oct	10	15:30	1989	exesun4/cJdecoder1
rwxr-xr-x3879/3879	40960	Oct	9	15:22	1989	exesun4/cif2ccir_an_b1
rwxr-xr-x3879/3879	24576	Oct	9	16:02	1989	exesun4/cosmetics
rwxr-xr-x3879/3879	40960	Oct	10	15:30	1989	exesun4/eJdecoder1
rwxr-xr-x3879/3879	24576	Oct	9	15:22	1989	exesun4/layers
rwxr-xr-x3879/3879	24576	Oct	9	15:25	1989	exesun4/magnify
rwxr--r--3879/3879	40	Oct	9	15:25	1989	exesun4/magnify.com
rw-r--r--3879/3879	129600	Oct	9	15:30	1989	exesun4/ostill.29
rwxr--r--3879/3879	90112	Oct	9	15:23	1989	exesun4/reccr
rwxr--r--3879/3879	96	Oct	9	15:27	1989	exesun4/reccr.com
rwxr-xr-x3879/3879	98304	Oct	9	15:23	1989	exesun4/recy
rwxr-xr-x3879/3879	24576	Oct	9	15:20	1989	exesun4/smlavg2
rwxr-xr-x3879/3879	699840	Oct	9	16:28	1989	exesun4/still.090.rec.yuv
rwxr--r--3879/3879	787	Oct	10	16:47	1989	exesun4/still.com
rw-r--r--3879/3879	2307	Oct	11	11:05	1989	exesun4/README
rwxr-xr-x3879/3879	49152	Oct	11	10:38	1989	exesun4/decdsl
rwxr-xr-x3879/3879	1731	Oct	11	10:38	1989	exesun4/decshell
rwxr-xr-x3879/3879	98304	Oct	11	10:38	1989	exesun4/dfa
rwxr-xr-x3879/3879	131072	Oct	11	10:38	1989	exesun4/image2rm5
rwxr-xr-x3879/3879	73728	Oct	11	10:38	1989	exesun4/rm52s
rwxr-xr-x3879/3879	81920	Oct	11	10:38	1989	exesun4/scalev
rwxr-xr-x3879/3879	16384	Oct	11	10:38	1989	exesun4/xt

6. Figure of merit Random Access

The "Group of Frames" (GOF) data structure is the key to assessing random access in MCPIC. The maximum number of bits needed to fully decode a given frame is given by the cumulative bit count for "intra" and "predictive" mode (the interpolative information - both motion information and interpolation error - can be ignored). Out of the four sequences submitted (Table Tennis, Flower Garden, Diva and Still), the worst case happened in the first GOF of the sequence Table Tennis. Much is to be learned by the careful Analysis of the GOF data.

The cumulative bit count for GOF(1).table_tennis is 658949 bits. At an average reading rate of 0.9 Mbits/s, this correspond to a latency of:

$$x = 0.73 \text{ seconds}$$

the Figure of merit for random access is $FM = 5 \cdot (2-x)$, i.e.

$$FM_{\text{Random Access}} = 6.35.$$

With the same system parameters (GOF) this figure of merit for random access could be much higher if less bits had been used for the first half second of table tennis. Because of temporal masking this could have been done without affecting the visual quality.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO-IEC/JTC1/SC2/WG8
CODED REPRESENTATION OF PICTURE AND AUDIO INFORMATION

ISO-IEC/JTC1/SC2/WG8
MPEG 89/
October, 1989

Source: Kun-Min Yang, Didier J. LeGall (Bellcore)
Title: Demonstration of VLSI implementability

1. Introduction

The Bellcore proposed ISO-MPEG decoder has the capability of performing the following features:

- Forward normal playback.
- Reverse normal playback.
- Transcoding of CCITT RM8 [1].
- Transcoding of JPEG Base Line System [2].
- Still image build up with high resolution.
- Random access.

The decoder is partitioned into functional modules such that the hardware components can be shared for performing different features. The partition is shown in Figure 1.

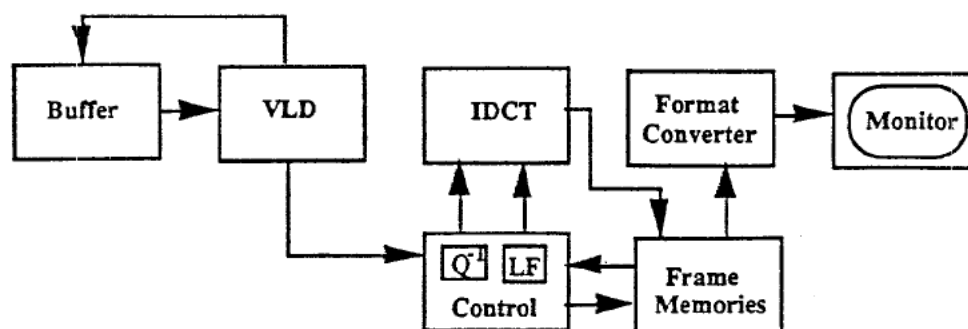


Figure 1. Schematic diagram of the decoder.

The functions that each component performs are explained as follows:

1.1 Input Buffer

The input buffer is a 256k-bit DRAM Ring Buffer with control counters to point the read/write addresses.

1.2 Variable Length Decoder (VLD)

The VLD chip receives bit stream from the input buffer and decodes it into information including the format of the syntax and the coded video information. It decodes CCITT RM8 Variable Length Codes (VLCs) and JPEG VLC. Since the throughput of the VLD chip is varying in time, a feedback signal is used to control the input bit stream from the buffer.

1.3 System Control

This block does the inverse quantization, performs the loop filtering, generates address sequences for read/write of the frame memories and generates control signals. In the inverse quantizer sub-module, there are two RAMs holding two quantization matrices for transcoding the JPEG bit stream. There is a multiplier to calculate the transformed coefficients from their representations and the current quantization step. Depending on the block type and motion vector, the control signals which are derived from the format information trigger the generation of address sequences for fetching data from frame memories. This data stream is fed through a Loop Filter (LF) if it is a motion compensated block, otherwise the LF is bypassed.

1.4 Inverse Discrete Cosine Transform (IDCT)

To achieve the inverse DCT function we can use a commercially available chip (IMS A121 from Inmos [3]). This chip transforms 8×8 DCT coded information into 8×8 spatial information. An adder is provided in the last stage of the chip to add the previous frame information with the inverse transformed information to reconstruct the current block. The next version of A121 will include inverse Zig-Zag scan procedure, therefore no extra component is necessary to perform this.

1.5 Frame Memories

The memory requirement is set to be 9 frames (about 9M bits) according to the most demanding mode, the reverse playback mode. The frame memories are organized to be shared for all different modes. Each frame requires one Mega bits for accommodating Y, U and V components in CIF size. An extra memory space of 150k bits might be needed to save the interpolative-mode motion vectors for reverse playback.

1.6 Format Converter

This chip implements an interpolation filter to convert the CIF format into CCIR 601 525/60 format. It includes a module for displaying JPEG images, and combining a reconstructed video frame with some extra decoded information for building up a still image with high resolution.

The following table summarizes the decoder cost¹.

	Quantity	Area (mm ²)	Pins	s _{1i}	s _{2i}	s _{3i}	s _{4i}	Cost
IDCT	1	36	44	25	0	1	0.44	12.77
VLD	1	40	64	25	0	1	0.64	13.37
System Control	1	45	84	35	0	1	0.84	16.87
Format Converter	1	9	31	18	0	1	0.31	10.35
1M DRAM	1	60	34	39	1	1	0.34	26.53
4M DRAM	2	240	34	342	8	2	0.68	189.62
Buffer	1	9	18	18	.256	1	0.18	12.46
Total				502	9.256	8	3.43	281.97

The total cost is estimated to be 281.97.

In the following sections, the design of each functional component in Figure 1 will be described in more details. The IMS A121 chip is commercial available and will not be discussed furthermore.

2. Variable Length Decoder (VLD)

Two key functions of this chip are the decoding of the VLC used in JPEG and the other set of VLCs used in RM8, and the deformatting of the syntax embedded in the bit stream. The Bellcore proposed MPEG coder uses the same syntax as that in RM8, and the same VLC as that in JPEG. There are five VLC tables used in RM8, i.e. macro block types (TYPE 3), macro block relative addressing (MBA), coded block pattern (CBP), differential motion vector (DMV) and 2-D VLC for run-lengths and levels of T-coefficients (2D-T). (For detail code tables, refer to [1].) One VLSI architecture for decoding

1. s₁: Cost of VLSI chip (w₁ = 0.29)
s₂: No. of Mbits DRAM (w₂ = 10)
s₃: No. of VLSI chips (w₃ = 4.2)
s₄: No of pins in the unit of 100 (w₄ = 3)

multiple-table VLC is described in [4]. This architecture will be adopted to decode the RM8 VLCs. An extra table plus some interface circuitry for decoding JPEG VLC is shown in Figure 2.

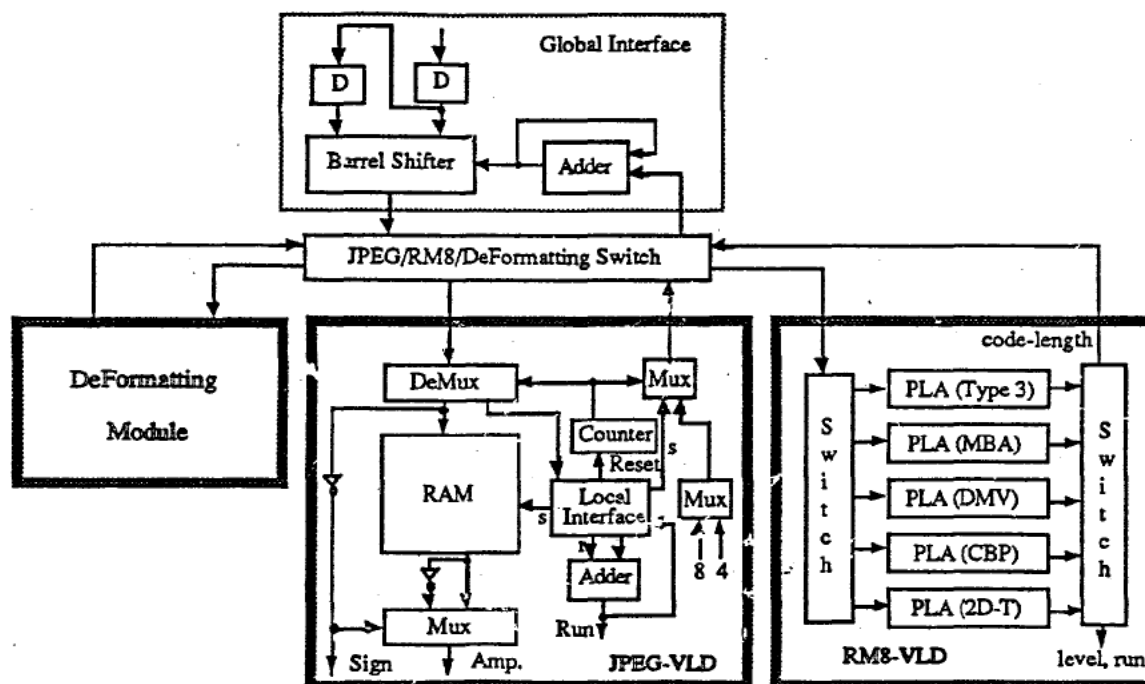


Figure 2. Schematic diagram of Variable Length Decoder.

In JPEG coder, either AC or differential DC coefficients are transformed into code-pairs. For AC coefficients, the [run-length, size] of each pair is modeled as an 8-bit code called Fixed Length Code (FLC). The first four bits of a FLC specify the run-length (r) ranging from 0 to 15. The last four bits specify the size (s) of a VLC, i.e. number of bits following a FLC that represents a coefficient amplitude. The following table gives the sizes for different amplitude ranges.

AC Coefficient Amplitude	Size
-1, 1	1
-3, -2, 2, 3	2
-7... -4, 4... 7	3
-15... -8, 8... 15	4
-31... -16, 16... 31	5
-63... -32, 32... 63	6
-127... -64, 64... 127	7
-255... -128, 128... 255	8
-511... -256, 256... 511	9
-1023... -512, 512... 1023	10
-2047... -1024, 1024... 2047	11

In the coded bit stream, there are FLCs and VLCs occur alternatively except for the situation that r is larger than 15. When $r = n \times 15 + r'$, where $0 < r' < 15$, then $n+1$ FLCs are used and Sizes' in the first n FLCs are set to zero. Therefore, in the JPEG-VLD circuit module, a 1-bit counter is used to control a MUX for selecting 8 or s , which will be added to the accumulator in the global interface circuit and hence treated as a control signal to the barrel shifter. If s is equal to zero, then the counter will be reset to choose 8, and r in the next byte of the bit stream will be accumulated to run-length. If s is non-zero, then s bits following the FLC are calculated as amplitude. An example is given in the following table to translate the binary codes to sign-and-magnitude representations when $s = 3$.

Binary codes	Sign-and-magnitude
000	-111 (-7)
001	-110 (-6)
010	-101 (-5)
011	-100 (-4)
100	100 (4)
101	101 (5)
110	110 (6)
111	111 (7)

Obviously, if the first bit of a binary code is 0, then the resulting amplitude is negative and the magnitude is one's complement of the binary code. If the first bit is 1, then the resulting amplitude is the same as the binary code. An 11×11 RAM is used to store the bits in different shifted versions. One out of eleven contents will be chosen according to the value of s . The first bit is also used as a sign bit, and depending on its value, the magnitude of the s bits or its one's complement is used as the amplitude.

The Bellcore proposed coder uses 5 bits of FLC for run-length (r) and the rest 3 bits for size (s). The dynamic range of the VLC for AC amplitudes is within that of JPEG VLC. Therefore the circuitry for decoding JPEG VLC can be used to decode the proposed VLC.

For differential DC coefficients, the FLC part has only 4 bits for representing the size of VLC. A maximum of 12 bits are used to represent the amplitude in a signed integer format. The circuitry for differential DC coefficients is a subset of that of AC coefficients.

3. System Control

3.1 Inverse Quantization(Q^{-1})

The inverse quantizer translates a transformed level, i.e. x , back to a transformed coefficient, i.e. y , according to the equation:

$$\begin{aligned} y &= (x + 0.5) \times g && \text{if } x > 0 \\ y &= 0 && \text{if } x = 0, \text{ for RM8,} \\ y &= (x - 0.5) \times g && \text{if } x < 0 \end{aligned} \quad (1-a)$$

$$y = x \times g, \quad \text{for JPEG,} \quad (1-b)$$

where g is the quantization step. In RM8, g is fixed within a macro block while it is frequency dependent in JPEG coder. The default quantization matrices for luminance and chrominance are shown in Figure 3.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)

17	18	24	47	66	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

Figure 3. (a) Default quantization matrix of luminance.
(b) Default quantization matrix of chrominance.

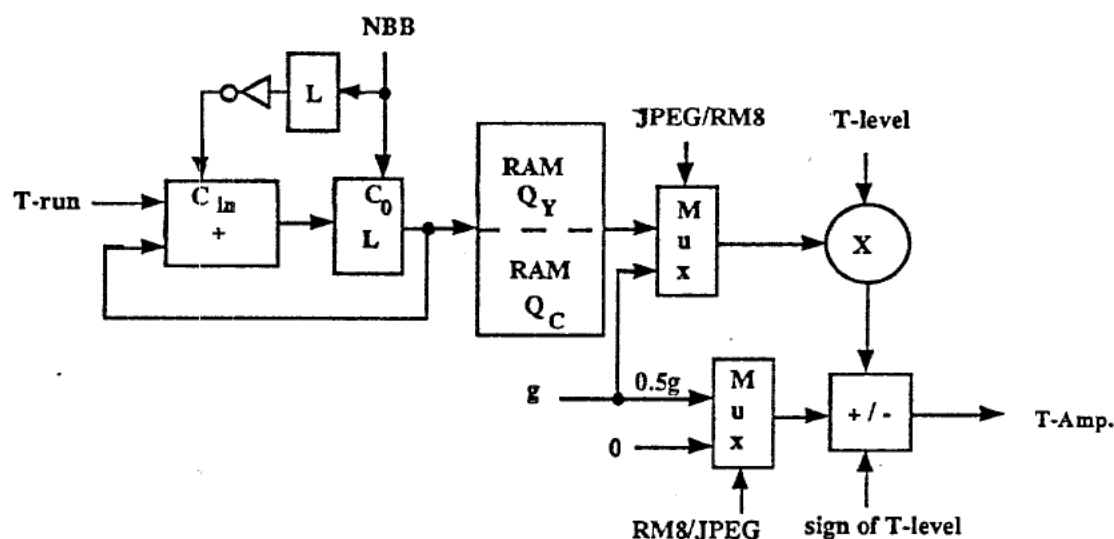


Figure 4. Schematic diagram of the inverse quantizer.

The schematic diagram of a inverse quantizer is shown in Figure 4, there are two RAMs (each has 64 entries with 7-bit in depth) implemented to hold the JPEG Q-matrices. Since the incoming transformed levels are in Zig-Zag scanned order, thus the quantization step values in the Q-matrices are arranged in the same order. They can be customer-programmed by loading new matrices before the decoding process starts. A 5-bit adder is used to accumulate the run-lengths. The accumulation result is in terms used as an address to locate the corresponding g value, in the Q-matrices, for the current non-zero amplitude. The accumulator is reset to zero at the beginning of every block when a NBB (New Block Begin) signal is detected. The next non-zero amplitude position is (run-length + 1) pixels away from its previous one. Therefore the carry-in (C_{in}) of the accumulator is set to 1 except for the beginning cycle of a block. There is a multiplier to perform the multiplication in Eq. (1-a). Two inputs to it are the transformed level (T-level) and g which is either from the bit stream (for RM8) or from the Q-matrices (for JPEG). In last stage, an adder/subtractor is implemented to either add or subtract $0.5 \times g$, as Eq. (1-a), for RM8. Zero is added, as Eq. (1-b), for JPEG. Then the computation of T-Amplitude is complete.

The Bellcore proposed MPEG coder uses custom-made Q-matrices. A multiplier is needed to multiply the matrices with the quantization setpsize.

3.2 Loop Filter

This module is designed to perform loop filtering both for the proposed MPEG coder (LF_1) and RM8 (LF_2). The filter impulse responses are shown in Figure 5.

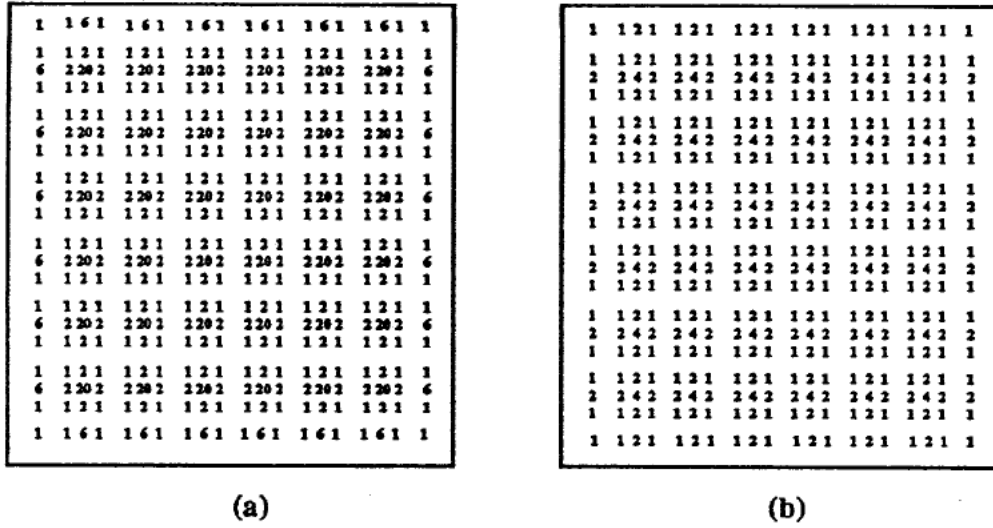


Figure 5. (a) LF_1 . (b) LF_2 .

Both of loop filters have 3×3 kernels except for the pixels that are along the boundary of an 8×8 block. The pixels along the boundary are 1-dimensional filtered, and those in the corners are repeated by themselves. The central weight of LF_1 is 20 instead of 4 in LF_2 , which results in sharper filtering responses. However LF_1 is non-separable. In order to minimize the circuitry redundancy, LF_1 is manipulated as follows:

$$LF_1 = \frac{1}{32} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 20 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{2} \left\{ \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} = \frac{1}{2} \left\{ LF_2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} \quad (2)$$

The first term on the right hand side of the equality is identical to LF_2 , and is a separable kernel. In order to let the filtered outputs be in raster scanned format, within an 8×8 block, to match to the same format of IDCT outputs, a 2-pass 1-dimensional filter architecture is used. In Figure 5, Similar circuitry structures are used for both horizontal filtering and vertical filtering. The only difference is the length of delay elements.

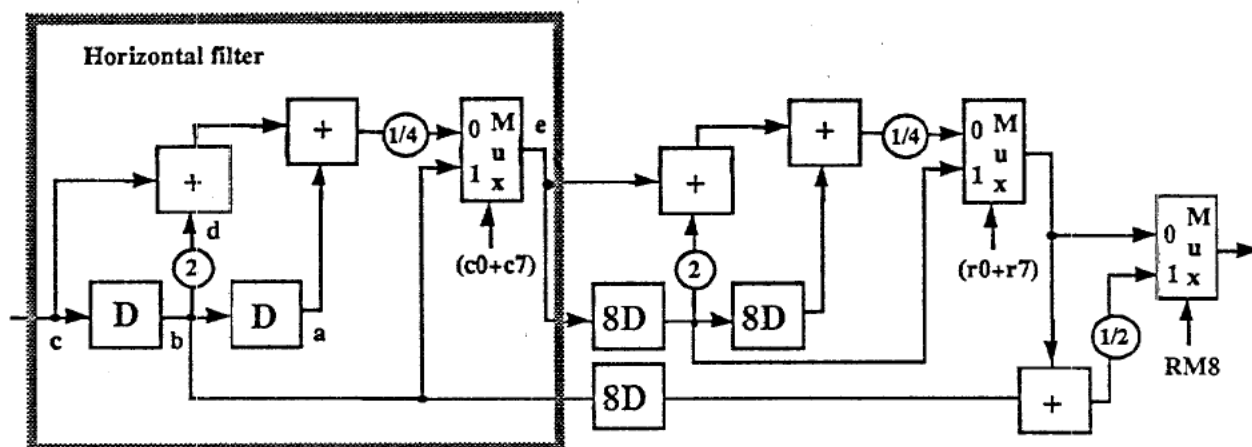


Figure 5. Schematic diagram of the loop filter.

The block in the left part of Figure 5 does the horizontal filtering. The calculations at several locations are demonstrated in Figure 6, where r_0 , r_7 , c_0 and c_7 represent the first row, the last row, the first column and the last column in an 8×8 block. Data are input from location c and outputs are obtained at location e . The MUX before location e , controlled by $(c_0 + c_7)$, serves for boundary condition. The signals c_0 , c_7 , r_0 and r_7 can be derived from the address generator which will be described in the next sub-section. The last term in Eq. (2) has only one entry, a scaled value of an input pixel itself. In the implementation, a matching delay is used to add the scaled pixel value to the filtered result if LF_1 is on. By suitable insertion of latches between critical computation path to form pipeline stages can reduce the cycle period and increase the throughput rate. The matching delays along different pathes then need to be added at the same time. However, the maximum throughput rate can occur is less than 4 M pixels/sec. This is easy to achieve using current VLSI technology.

t	a	b	c	d	e
0			x_{00}		
1		x_{00}	x_{01}	$2x_{00}$	x_{00}
2	x_{00}	x_{01}	x_{02}	$2x_{01}$	$1/4(x_{00} + 2x_{01} + x_{02})$
3	x_{01}	x_{02}	x_{03}	$2x_{02}$	$1/4(x_{01} + 2x_{02} + x_{03})$
⋮			⋮		⋮
7	x_{05}	x_{06}	x_{07}	$2x_{06}$	$1/4(x_{05} + 2x_{06} + x_{07})$
8	x_{06}	x_{07}	x_{10}	$2x_{07}$	x_{07}
9	x_{07}	x_{10}	x_{11}	$2x_{10}$	x_{10}
10	x_{10}	x_{11}	x_{12}	$2x_{11}$	$1/4(x_{10} + 2x_{11} + x_{12})$
⋮					⋮

Figure 6. Some calculations at locations a, b, c, d and e of horizontal filter.

3.3 Address Generator

The address generator has to generate a set of address sequences for read/write of the frame memories to accomplish the following functions:

3.3.1 Within One Video Frame

Store Current Frame: The current frame is reconstructed and written in a frame memory according to the macro block sequence shown in Figure 7.

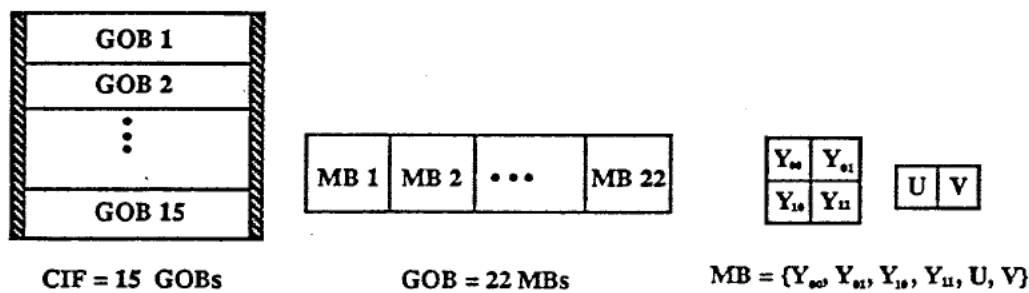


Figure 7. Macro block syntax.

Within a 16×16 macro block, there are six 8×8 blocks arranged as $\{Y_{00}, Y_{01}, Y_{10}, Y_{11}, U, V\}$. Let

the upper-left corner of a block be the base address for that block, then the base address sequence for the six blocks within a macro block is a regular one. The base address sequence for macro blocks within a frame is also a regular one. Therefore, the combined base address sequence for all blocks can be generated by a counter. Since the blocks may be of different block types, the timing for the base address sequence is triggered by control signals derived from the format information. Once the base (starting) address of a block is triggered, a regular raster scanned sequence will be generated, one address at a time ranging from 0 to 7 in both x and y coordinates. The raster scanned sequence will be added to the base address to obtain the physical location in a frame memory where the reconstructed data will be written into. This address sequence is also suitable for motion compensated interpolation mode whose 8×8 blocks are all of block types within a subset of RM8 block types, i.e. either motion-compensated/not-coded or no-motion/not-coded.

Note that, although the macro block syntax, the GOB arrangement, used in RM8 has been changed from that of RM5 to a new one and the frame size might be different, the address generator can be designed to be re-configurable to satisfy different GOB arrangement and several possible frame sizes, i.e. 352×240 , 352×288 , 176×144 . Since the address generator is implemented by a set of counters, only the counters related to GOB layer and above need to be programmable. The extra logic circuitry for this programmability is simple.

Fetch Reference Frame Data: During inter-mode, a block, offset by a motion vector (zero if no motion), is read from the previous reconstructed frame memory to reconstruct the current frame. The address sequence is identical as that of the current frame except for the offsets by motion vectors. Therefore an adder is needed to add the motion vector to the "*current frame*" address sequence, discussed in the last paragraph, to form a new sequence. For motion compensated interpolation mode, blocks from both the previous frame and the next frame are needed. One is offset by the motion vector while the other one is offset by negative of the motion vector. If one block is out, or partially out of a frame boundary, then only the other frame is used. Other than that, an average of the two is used. The circuit in Figure 8 has two multiplexors controlled by the out-of-bound conditions, a_out and b_out , to either take an average or just one value.

Display: The video data stored in the frame memories is further processed by a format converter for display. The address sequence must be designed to match to the operations performed in the format converter. The format converter as well as the address sequence for display will be discussed in the next section.

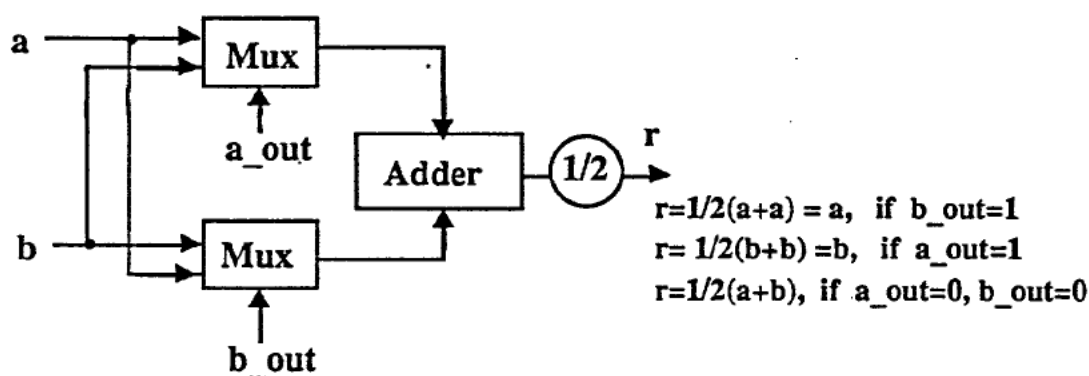


Figure 8. Circuit for boundary conditions in interpolative mode.

3.2 Organization of Frame Memories

The reverse playback mode has the most demanding of memories. Based on the proposed coder, there is an intra-frame mode every 15 frames. Within these 15 frames, every other frame is motion-compensation interpolated, i.e. 1st frame is intra-mode, 3rd, 5th, 7th, ..., 15th are inter-mode, and 2nd, 4th, 6th, ..., 14th are interpolative-mode. Therefore, nine frame memories, i.e. 9M bits, are needed. Eight of them are used for storing the inter-mode decoded frames in a forward sequence starting from a intra-mode frame. In the implementation two 4M DRAMs and one 1M DRAM are used. The arrangement of memories, in the unit of video frame, is demonstrated in Figure 9. A "white triangle" denotes that a decoded frame is written into a memory at a certain time slot and a "shaded triangle" denotes that the stored frame is read out for display. Note that a memory can be written again only after a previous stored frame has been display. But the same frame can be written into the memory and displayed at the same time slot through a dual-port memory. The forward playback is also demonstrated in Figure 9.

For each mode, there is a cycle for locating the memories. These sequences can be implemented by a finite-state-machine using PLA or a look-up-table. There are periods that two frames are reconstructed at the same time slot, e.g. 15 and 41 frames. However, this occurs only when one is intra-mode frame which does not require reference frame, and the other one is motion compensated interpolation frame which does not require IDCT. They both can use the same "current-frame" address sequence, and they were written into different memories. All the blocks in these two kinds are of block types that are

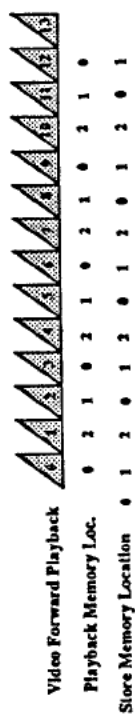


Figure 9. Organization of frame memories.

reconstructed one pixel at a cycle without break. The only difference is that they have different initial delays at the beginning of a frame. A suitable matching can be derived and inserted for reconstructing the motion compensated interpolation mode which has zero delay.

4. Format Converter

The function of this chip is to convert the CIF format into the CCIR 601 525/60 format for display. The sizes for different components of both formats are described as follows:

CIF	CCIR 601 525/60
Y : 352×240 non-interlaced	$\rightarrow 704 \times 480$ interlaced
C : 176×120 non-interlaced	$\rightarrow 352 \times 480$ interlaced

The CIF luminance component is 1-to-2 horizontally interpolated into 704×240 as a field of CCIR 601 and then the same field is repeated once to form a frame. The interpolation filter is given by

$$x_{i+\frac{1}{2}} = \frac{1}{16} (-x_{i-1} + 9x_i + 9x_{i+1} - x_{i+2}), \quad (3)$$

and is implemented as low-left part of Figure 9.

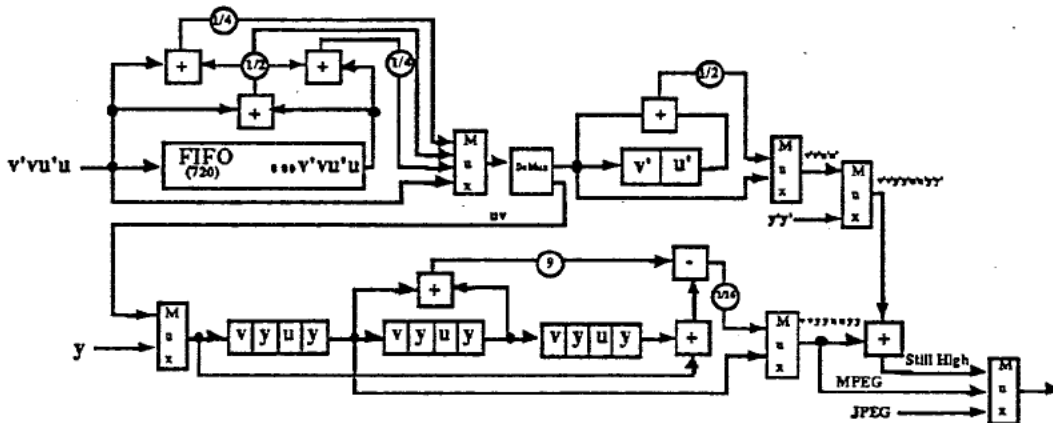


Figure 9. Schematic diagram of a format converter.

The CIF chrominance component is interpolated with the same filter as Eq. (3) horizontally, but it is bilinear interpolated vertically. The resulting 352×240 forms a CCIR 601 field and again it is repeated

to make a frame. The vertical filtering is processed before horizontal filtering, therefore a FIFO holding one line of U and V, interlaced with each other, is implemented in the vertical filter. Note that the size of the FIFO is related to the horizontal dimension of a video frame. But there are only a limit set of possible sizes might be considered, i.e. 176, 180, 320, 352, 360. Therefore the size of this FIFO can be designed to be programmable for the above mentioned sizes using a very simple multiplexing circuitry.

The format converter also implements the still image build-up, i.e. extra 270K bits are decoded as residue information and added to a reconstructed video frame to obtain a high resolution image. The residue information has a format as follows:

Y': 704 × 480 , C': 176 × 120 , non-interlaced.

The Y' component has full CCIR 601 resolution, thus no further processing is needed. The chrominance component is 1-to-2 bilinear interpolated horizontally and 1-to-4 bilinear interpolated vertically. However CCIR format is interlaced, therefore only two out of four vertically interpolated lines of U'/V' are displayed for each field. A MUX is used to control this function. To further reduce the complexity, the residue frame is processed simultaneously with the conversion of a video frame. Their results are added together, by an adder in the last stage in the format converter, to form a still high. This is made possible by suitably interlacing two kinds of data stream and by analyzing of the sampling rate as follows:

MPEG	Still High
$Y: 360 \times 240 \times 60 = 5M$	$Y': 720 \times 480 \times 30 = 10M$
$U/V: (180 \times 120 \times 2) \times 2 \times 60 = 5M$	$U'/V': (180 \times 120 \times 2) \times 2 \times 60 = 5M$

The input data rate for the combined U/V and U'/V' stream is 10M which is matched to that of combined Y and U/V. The speed requirement for most of the calculations is the same as the input data, 10 M, which is achievable. Only the MUXs and the adder in the last stage require 20 M speed, which again is achievable. Since the two output streams have their YUV arranged in exactly the same order, they can be added directly and then separated into three components. If the Still High mode is not on, only the reconstructed video information is output.

The address sequences for inputs are in a raster scanned order to match to that of the CCIR 601 display format. Each line of U/V and/or U'/V' is input twice consecutively except for the first line in a frame. Each CIF frame is input twice to make two fields of a CCIR frame.

By considering the advances in Mixed IC technology, a YUV-to-RGB matrix and a D/A converter can be implemented in this module. Then the outputs can be connected to a monitor directly.

5. Conclusion

In this proposal, we designed a JPEG VLD module, a loop filter module, an inverse quantizer module, an address generator module and a format converter chip for the Bellcore proposed MPEG decoder. All the modules are JPEG and/or RM8 compatible. We also described the organization of frame memories and procedures for integrating all the modules to perform different tasks.

References

- [1] "Description of Reference Model 8 (RM8)" Document 525, CCITT SGXV, Working Party XV/4, Specialists Group for Visual Telephony, 1989.
- [2] G. Wallace, "Description of the JPEG Baseline System to be Used for Evaluation of JPEG-Compatibility by MPEG", Document of ISO-IEC/JTC1/SC2/WG8, August, 1989.
- [3] "IMS A121 2-D Discrete Cosine Transform Image Processor; Advance Information", Inmos, April 1989.
- [4] Ming-Ting Sun, Kun-Min Yang and Kou-Hu Tzou, "High-Speed Programmable ICs for Decoding of Variable-Length Codes", SPIE's 33rd Annual International Technical Symposium on Optical & Optoelectronic Applied Science and Engineering, San Diego, California, 1989, pp.1153-04.

EXHIBIT 2

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE D E , NORMALISATION
ISO/IEC JTC1/SC2/WG11
CODED REPRESENTATION OF PICTURE AND AUDIO INFORMATION

ISO-IEC/JTC1/SC2/WG11
MPEG 91/
November, 1991

Title: Hybrid Extended MPEG (HEM) Coding Algorithm (proposal registration no. 5)
Purpose: Proposal for MPEG Phase II Co'ding Standard
Subgroup: Video, Test, Requirements, and Implementation
Source: Bellcore and NTT

1. Introduction

This document describes the Hybrid Extended MPEG (HEM) coding system, where the coder, decoder, various functionalities and their complexities are analyzed in detail. This coding system is being proposed as a standard for compression and decompression of moving pictures as addressed by the second phase of the MPEG work. This document is aimed at fulfilling the requirements of the ISO/MPEG proposal package description (MPEG 91/100).

The basic algorithm shares many similar features with the video coding system specified in the MPEG Simulation Model (SM) [1]; therefore, the syntax is also very similar to the one described in the CD 11172-2 [2]. HEM exceeds the SM quality in 'coding pictures' at higher rates by modifying the SM algorithm. In particular we believe that the following new features contribute significantly, to quality over that achievable by the SM:

- Hybrid field-frame based approach (field-based motion compensation and frame-based Discrete Cosine Transform).
- Optimized bit allocations and quantizer assignment.
- Adaptive Huffman code tables, (according- to JPEG standard).

The HEM has a hybrid coding structure with field-based motion compensation (MC) and frame-based Discrete Cosine Transform, (DCT). Motion estimation is first performed on pairs of 4:2:0 subsampled fields. After motion compensation, odd and even fields are merged by interleaving the rows into 'one frame, and the DCT is applied to the residual, signal on a frame basis. Bit allocation is done 'for the entire frame, and the quantizers are assigned on the Macroblock level based on the bit allocation. After quantization, the DCT values are encoded using the adaptive Huffman code table as-in the JPEG standard.

2. Description of the Hybrid Extended MPEG Algorithm

2.1 Overview of HEM

The compression algorithm starts with preprocessing where each CCIR601 field is converted to a 4:2:0 subsampled field with 240x720 pixels of Y and 120x360 pixels of Cr and Cb. The encoder operates on the subsampled sequence at 60 fields/sec. Details of the preprocessor are given in section 2.2.

Since each subsampled frame (fields merged) has dimensions four times the size of a SIF picture, the processing complexity is slightly more than four times that of MPEG I. HEM performs source coding in intraframe, predictive, and bidirectionally predictive modes. It also has a similar, yet modified set of data structures as compared to MPEG I. The data structures include Groups of Pictures (GOPs), Super Slices (SSs), Super Macroblocks (SMBs), and Macroblocks (MBs). Each GOP has 12 frames with two bidirectionally coded pictures between two predicted frames (M=3) (Fig.1). A Super Slice consists of one row of Super Macroblocks, and each Super Macroblock has two Macroblocks spatially oriented in the vertical dimension (Fig.2).

The structure of the encoder is similar to the algorithm in the SM with the parameters set differently, affecting the controls of various coding operations. The principal compression techniques used in the encoder are still @dimensional Discrete Cosine Transform; Motion Compensation, and Variable Length Coding (VLC). To improve motion prediction, motion estimation is performed field by field on the same parity; i.e., even fields are used to predict even fields, and odd for odd. The DCT is still applied on each 8x8 block after the two motion compensated fields have been merged. The Macroblock Quantizers (MQANTs) are selected through an iterative process such that the total bit count generated in a frame is within some margin of the bit allocation. Finally the DCT quantized values are encoded using a Huffman table adapted to the statistics of the quantizer outputs. The decoded pictures are converted back to the CCIR601 format through postprocessing.

Details of each module of the compression algorithm are described in the following sections. The issues of compatibility to existing standards; complexity of the coder-decoder, and other functionalities are covered in subsequent sections. The algorithm block diagrams, flow charts, VLC tables, and syntax diagrams are also given. Finally, the listing of bit stream files and the coding statistics, are appended.

2.2 Preprocessing

The source video is converted from the CCIR601 format into a sequence of 4:2:0 subsampled fields through preprocessing. The sampling rate of the chrominance signals is reduced vertically by a factor of 2. Since each CCIR601 field contains 243 lines of luminance and chrominance signals, the last three lines out of those 243 in all three components are dropped. Each of Cr and Cb arrays of size 240x360 is then vertically decimated to get arrays of size 120x360 which are used in the coding. The decimation is carried out by processing the Cr and Cb input arrays along each column with a symmetric eleven-tap FIR filter and computing only 120 output samples per column. The output (column) signal $y[n]$ is given by:

$$y[n] = 1/5 \sum_{i=-5}^5 h[i]x[2n-i], \quad 0 \leq n < 120, \quad (1)$$

where the input signal $x[n]$ is extended by five samples at each end. The filter coefficients are given by:

$$h = [3, 0, -25, 0, 150, 256, 150, 0, -25, 0, 3];$$

and the scaling factor $S = 512$.

1.3 Postprocessing

The decoded video is converted from the 4:2:0 subsampled format back to the CCIR601 format through postprocessing. The sampling rate of the chrominance signals is increased vertically by a factor of 2. Each of Cr and Cb decoded-field arrays of size 120x360 is vertically interpolated to get arrays of size 240x360. The interpolation is carried out by processing the Cr and Cb input arrays along each column to insert a sample after every input sample. The interpolated output (column) sample $y_o[n]$ is given by:

$$y_o[2n+1] = 1/S_g \sum_{i=-1}^2 g[i]x_d[n-i], \quad 0 \leq n < 120, \quad (2)$$

where the decoded signal $x_d[n]$ is extended by one sample at the beginning and two samples at the end of the column. Note that $y_o[2n] = x_d[n]$, $0 \leq n < 120$. The filter coefficients are given by:

$$g = [-1, 9, 9, -1];$$

and the scaling factor $S_g = 16$. Since each CCIR601 field contains 243 lines of luminance and chrominance signals, three (black) lines are appended to the 240 lines in all three components.

2.4 Predictive/Transform Coding Loop

Figure 3 summarizes the core of the compression algorithm. It shows tremendous resemblance in structure to the SM hybrid coding loop. This coding loop is based on temporal predictive coding with motion compensation and transform coding. It also contains an embedded decoder to recover the coded pictures. A simple mechanism in the loop allows the encoder to switch among the intraframe, the predictive, and the bidirectionally predictive modes. One distinctive feature of the HEM which is different from the SM is that motion estimation/compensation is performed field by field on the same parity; while all other operations including the DCT and quantization are performed frame-based (Fig.4). A Macroblock is the basic processing unit in most operations, where the MBs are scanned within the SMBs down each Super Slice throughout the entire frame.

2.4.1 Intraframe Coding

Intraframe coding is used to encode the first frame (encoding order) of each GOP. It is done like the SM intra-coding where DCT is performed once on every 8x8 block in the source picture. After DCT, an MQANT is assigned to every MB based on the statistics of the DCT outputs. Then quantization is done utilizing the MQANTs and a non-uniform 8x8 matrix. The process of MQANT assignment and quantization is reiterated until certain requirements are met. Then a VLC table is generated based on the statistics of the quantizer

outputs to encode the quantized DCT coefficients. The intraframe (I-frame) is reconstructed through inverse quantization and inverse DCT. The decoded image is stored in a coded picture memory as reference picture for future motion compensation.

2.4.2 Predictive Coding

Predictive coding is used once every three frames except on the first frame of a GOP. This coding scheme requires motion prediction from a previously coded I-frame or P-frame (predicted frame). Motion estimation is performed on the original luminance components of both the current picture and a reference picture. Since motion estimation is more accurate when performed on the same parity field, the even field of the reference frame is used to predict the even lines of the SMBs in the current frame, and the odd field is used to predict the odd lines. The motion compensated even and odd lines are then interleaved to produce a motion compensated frame. Then the intra/non-intra decision is made for each MB in the P-frame based on the criterion in the SM algorithm [1]. According to this decision, either the prediction error or the original data (in the case of an intra MB) is encoded through the DCT. DCT is actually performed twice, once on the luminance of the original picture, and once on the difference picture. The result of the first transform is used to evaluate MQUANTS. Again, the assignment of MQUANTS and quantization are repeated until some criterion is reached. A new VLC table is then generated to encode the quantized output. Each P-frame is also reconstructed by adding the prediction back to the decoded difference picture. This recovered picture is used to predict the B-frames (bidirectionally predicted frames). In addition to the P-frame Macroblock types used in the MPEG I standard, a new MB type is defined. A non-coded MB with zero motion vectors, formerly considered as skipped, is now defined with a 7-bit codeword. More on Huffman coding will be described in a separate section.

2.4.3 Bidirectionally Predictive Coding

Bidirectionally predictive coding is used twice consecutively between two non-B-frames. In this coding mode, motion estimation is performed twice, with forward prediction referencing a past frame, and backward prediction referencing a future frame, where both reference pictures must have been previously coded and decoded in either the intraframe or the predictive mode. Again, motion estimation is field-based on the same parity. MB type decisions for the B-frame are the same as the SM and their VLCs apply. As in predictive coding, DCT is performed once on the luminance of the original picture, and once on all three components of the difference frame. The result of the first transform is used to evaluate MQUANTS. After some iterations of MQUANT assignment and quantization, a new VLC table is generated for the DCT quantized coefficients.

2.4.4 Motion Estimation

Motion estimation is performed on the luminance component of the original picture. Specifically, it is applied to each 16x16 block in a field. If the current frame is a P- or B-frame, the 16x16 block in the even (odd) field is found a best-matched block from the even (odd) field of the previous I- or P-frame, and a set of motion vectors (horizontal and vertical components) is obtained correspondingly. Note that there are two sets of motion vectors for each 32x16 block in a frame where the odd and even lines are block-matched independently. For a B-frame, an additional set of motion vectors is obtained from the next I- or P-frame with the same process. All the motion vectors are with half-pixel resolution, and are

obtained by full search. The maximum search range is related to the field distance. Given a field distance of F , the search range used in the proposal is $[-(\max\{21, 7F\}+0.5), \max\{21, 7F\}+0.5]$ subject to the constraint of image boundary.

2.4.5 MQUANT Assignment

MQUANTs are assigned to all MBs in a way to stabilize the compressed quality in the spatial-temporal domain. Their values are also adjusted iteratively to keep the bit rate output under control. In this section, only the MQUANT assignment is described. The rate control is left for a following section.

As described previously, all three coding modes require DCT to be performed on the luminance component of the original picture. For every 16x16 luminance Macroblock, the value $ACsrc$ is evaluated according to the following:

$$ACsrc = ACav * \frac{128}{\max(DCav, 96)} \quad (3)$$

where $ACav$ is the absolute AC average over the 4 8x8 blocks of DCT coefficients, $DCav$ is the average DC coefficient, 128 is a normalizing factor, and 96 is used to decrease the sensitivity of the DC term for $DCav$ values below 96. $ACsrc$ is a factor that explores the invisible distribution of a MB. MQUANT is proportionally assigned based on this value:

$$MQUANT = 2^c * ACsrc, \text{ for } I \text{ and } P \text{ frames;} \quad (4)$$

$$MQUANT = 2^{c+g} * ACsrc, \text{ for } B \text{ frames;} \quad (5)$$

where c is determined by rate control, and g is a preset value so that the quantizers for B-frames are slightly larger. According to this equation, MBs with higher texture are assigned larger MQUANTs and vice versa. Since every new MQUANT comes with an overhead in MB type and extra bits to specify the new value, some measure is taken to limit the occurrence of new MQUANTs. After DCT has been applied on the original picture, $ACsrc$ is calculated for every 16x16 block of the entire frame. Based on the statistics of the $ACsrcs$, one particular value ($ACflat$) is selected where the $ACsrc$ of a MB is changed to $ACflat$ if any one of its neighbors has a value less than or equal to $ACflat$. This step attempts to assign MQUANTs of flat regions to the surrounding edges. Then again based on the statistics of $ACsrcs$, $ACmax$ and $ACmin$ are selected to limit the range of $ACsrc$, with $ACmin$ greater than $ACflat$. Any value outside of this range is cropped by the max or the min. This step attempts to fix MQUANT in regions with $ACsrcs$ that fall outside of the range. These two steps not only control the usage of MQUANT, but also help to reduce the artifacts associated with coarse quantization of edges.

2.4.6 Quantization and Inverse Quantization

Quantization and inverse quantization are done very similarly as in the SM. They utilize both the Macroblock Quantizer and a Quantization Matrix (QM). As in the MPEG I standard, the QM is downloadable in the sequence layer, and separate matrices can be used for intra- and non-intra-quantization. The QM used in the simulations takes on intermediate shape between the two default matrices in the SM. The only different feature introduced is the expansion of end-of-block area before the intra-quantization. Each 8x8 block of an intra MB is zig-zag

scanned to obtain a one-dimensional array of increasing frequencies. Then a variable threshold (Th) is applied on every element starting from the last coefficient and ending on the fourth coefficient of lower frequency. The variable thresholding is performed this way:

```
for (i=63; i>=3; i-1) {
    Th[i] = (2*MQUANT*QM[i])/16;
    if (DCT[i]>Th[i] || DCT[i]<-Th[i]) break;
    DCT[i]=0;
}
```

where MQUANT is the Macroblock Quantizer assigned to the current MB, QM[i] is the *i*th entry of the Quantization Matrix, and 16 is a normalizing factor. Any element in the zig-zag scanned array with an absolute value lower than its corresponding threshold will be set to zero. This thresholding proceeds from the coefficient of highest frequency and terminates upon an element that exceeds the threshold. The purpose of this thresholding is to expand the end-of-block area of each 8x8 block within an intra MB to cut down the number of bits used for high frequency signals.

2.4.7 Rate Control

Rate control is achieved by adjusting the value of *c* which is used to evaluate MQUANTs (Eq. 3). Large *c*s contribute to coarse quantization and lowering of current bit rate. The process of MQUANT assignment and quantization reiterates during the coding of a frame until an optimal *c* is selected for the frame. This value of *c* is carried over to the next frame so that a stable quality can be achieved in the temporal domain.

This rate control scheme begins by assigning target bits for I-, P-, and B-frames such that these target values meet the bit rate constraint. The variable *c* is initialized for the first iteration in the MQUANT-quantization loop. It is also re-adjusted in each iteration in order that the final number of bits produced by the current frame is within some δ of the target value. This δ is a small fraction of the desired bit rate. To prevent indefinite looping, some terminating conditions are used and are summarized in a flow chart (Fig.5). If none of those conditions applies, *c* must be adjusted before the next iteration. The adjustment of *c* is summarized in another flow chart (Fig.6).

After the last iteration, *c* should have been changed more or less from its initial value. Since this new *c* will become the initial value for the following frame, the projected value of bits generated for each frame type can be calculated based on this new *c*. If these projected values were to be used as target bits, the resulting bit rate will eventually deviate from the desired bit rate. Therefore the target bits used in the terminating conditions must be updated based on the rate of change in bits due to the change in *c*. Also, at the end of each GOP, the buffer occupancy must return to the initial state at the beginning of the GOP to achieve a fixed delay. Since the actual number of bits generated per frame varies in time, the buffer state at the end of a GOP is indeterminate before the coding of the entire GOP is complete. Therefore, the projected difference in this final buffer state must be monitored and the target bit updating should absorb the buffer difference in advance.

In predictive coding, a simple switch is used to detect a scene change. If the number of intra MBs in a P-frame is greater than half the total number of MBs in a picture, a scene change is

signaled. The initial value of c will be raised by a small step for the current P- and the following two B-frames to accommodate the increase of bit consumption. This switch helps to speed up the iteration process that seeks for an optimal c .

2.5 Parameter Selection

The assignment of parameters is automatic, and no single parameter in the simulations has been optimized for any particular video sequence. However, there are three small sets of parameters whose values differ for bit rates above and below 6 Mbits/sec. The first set handles the initial assignment of target bits among different picture types. The second set determines the shapes of the intra- and non-intra-quantization matrices. The third set selects the max and the min $ACsrc$ based on the distribution of the $ACsrc$ on the current frame.

2.6 Entropy Coding

Because of the great resemblance in structure between this algorithm and the MPEG I simulation model, the set of symbols produced is coded in a similar manner; we only describe the differences below.

Macroblock Addressing and Coded Block Pattern from the MPEG I standard are not used since not many MBs are skipped at high bit rate coding. The start codes and the header information used in different layers remain the same. Those symbols that are handled differently include the following:

- The intra DC coefficients are differentially coded as in CD 11172-2, while the zero-runs and non-zero amplitudes for all other quantized DCT coefficients are coded using the JPEG approach. This approach optimizes the VLCs locally according to the symbol statistics in each frame. Some data regarding the codebook information will be sent along for the decoder to reconstruct the customized VLC table, and this information is inserted as the extension data in the picture layer.
- Two pairs of motion vectors must be coded for each motion-compensated Super Macroblock because the odd and even lines of an SMB are motion estimated independently. The differential motion vectors (DMVs) are encoded as in CD 11172-2 with the prediction being the motion vector of the most recently motion-compensated even MB. In other words, the prediction for an even MB motion vector will be that of the previous even MB and the prediction for the odd MB motion vector will be that of the current even MB. The motion vector prediction is reset at the beginning of a Super Slice. For a P-frame, it is again reset if both MBs of an SMB are of types No MC or intra, or if either one has type No MC while the other has type intra. For a B-frame, it is only reset if both MBs are intra. No change in syntax from CD 11172-2 is required to encode the DMVs.
- Macroblock types VLC tables for the three coding schemes remain the same except that a new MB type is introduced in the P-frame. A No-MC and Not-Coded MB will be encoded as type "0000001".
- The Macroblock Quantizer of the first MB in a Super Slice will be coded as the Super Slice Quantizer. This value will be used by the Super Slice until it is over-ridden by a different MQQUANT. All Super Slice Quantizers as well as the new MQQUANTs are coded with 5 bits as in the CD 11172-2.

3. Compatibility Feature

The coder and decoder of the HEM algorithm shares many common elements with the MPEG I system. These elements include the 8x8 DCT, zig-zag scan, quantization, inverse quantization, 16x16 motion estimation/compensation, and most of the VLC tables. This system cannot claim total compatibility with MPEG I because the data structure has been changed in the SMB level (see syntax diagrams). The MBs are scanned in a slightly different order, and the motion vectors are predicted differently. However, these two systems are compatible up to the building blocks. Furthermore, the issues of forward, backward, upward, and downward compatibility have become less important due to the implementation of programmable chips which can operate as different systems by loading the corresponding firmware. Given such technology, the key issue now seems to rest upon the commonality of processing elements.

4. Random Access Feature

Random access is achieved by periodic reset to intraframe coding. Since intraframe coding does not rely on the previous frame for decoding, an intraframe is readily accessible. The degree of random accessibility depends on the frequency at which intraframe coding is invoked. If a P- or a B-frame is requested, the decoder will start decoding from the previous intraframe until the requested frame is reached, ignoring all the intermediate B-frames. Presently our encoder invokes the intraframe mode once every 12 frames which is determined by the GOP structure. The amount of time to reach a given frame is given by the time to read the amount of bits representing the predictively coded frames of the GOP anterior to the given frame. The worst case of random access time is thus given by the time it takes to read the entire predicted part of a GOP structure.

5. Coding/Decoding D&Y

Fig.7 shows the complexity block diagram of the Hybrid Extended MPEG (HEM) encoder. Each module is labeled with a block number. Block number 1 down-samples the incoming pictures from 4:2:2 to 4:2:0 using an 11-tap filter, and the results are being stored into frame memories 2, 3, 4, and 5. Blocks 2 and 3 are used to store B-frames, and blocks 4 and 5 are used for I- and P-frames which will be used as reference pictures for motion estimation. Blocks 6 and 7 perform backward and forward motion estimation respectively. Blocks 22 and 23 store the corresponding odd-field motion vectors while block-matching on the even fields is being performed. Block 8 performs DCT on the luminance signal and block 10 stores the output of block 9, the ACs of each luminance Macroblock (MB). Blocks 19 and 20 perform motion compensation based on the motion vectors and the reconstructed reference frames (blocks 18 and 17). Then block 11 performs DCT on the prediction error; the coefficients are saved in block 12. These coefficients are quantized and the adaptive VLCs are generated in an iterative process with a maximum number of 5 iterations. The final set of quantized values are reconstructed through blocks 15 and 16. The timing requirement of each module is shown in Fig.8. In Fig.8 the number attached to the arrow identifies the processed frame number and the letter 'e' or 'o' represents its field parity. The length of the arrow implies the time duration for processing. The dash arrow associated with a memory implies the reading of data being held. Since a B-frame has the longest delay among the three frame types, the delay of the first B-frame (frame 1) is analyzed. Due to the nature of bi-directional prediction, a delay of M frames is imposed. Besides this M-frame delay, there is also a computational delay. The delay associated with the preprocessor is 1/20 of a frame.

Since the HEM encoder does not perform quantization until DCT of the entire frame is finished, a one-frame delay is expected, before quantization. Furthermore, since this algorithm uses a frame adaptive VLC, VLCs cannot be assigned until quantization of the entire frame is done. Assuming that the quantization iterations of the entire frame are completed within 1/4 frame, the total computational delay is, 1.3 (1/20+1+1/4) frame. Therefore, the encoding delay is $M+1.3$ frames (with $M \geq 2$).

Fig.9 shows the block diagram of the HEM decoder. The structure looks, exactly like the MPEG I decoder except for the merge and separate fields modules which are simply address generators. The post-processor performs chrominance vertical, up-sampling from 4:2:0 to 4:2:2 using a 4-tap filter and converts each frame into interlaced fields. Each module in this block diagram performs its operation according to the timing requirements shown in Fig.10. The expected delay is 1/2 of a frame due to the conversion from the interleaved structure to the interlaced structure.

Since we are assuming an encoder output buffer of 6 frames, the worst case delay is 360 ms.

$$\begin{aligned} \text{Overall Delay} &= \text{Encoder Delay} + \text{Buffer Size} + \text{Decoding Delay} \\ &= 3 + 1.3 + 6 + 0.5 \\ &= 10.8 \text{ frames} \end{aligned}$$

6. Coder/Decoder Complexity

Each module in the encoder timing diagram (Fig.8) has its required performance/capacity. Table 1 shows the number of operations in the worst case for each Macroblock. Motion estimation is performed twice per MB (forward and backward directions). The 8x8 DCT is performed 10 times for each MB (4 times on the luminance MB in block 8 of Fig.7 and 6 times on the prediction error in block 11). Quantization of 8x8 DCT is performed 30 times due to the worst case iteration. VLC assignment is done 5 times (worst case iteration). Inverse quantization and 8x8 IDCT are performed 6 times each for each 8x8 block in a MB. Table 3 shows the requirements of each module. The pre-processor contains memory for 26 lines of the CCIR 601 input picture. Blocks 2, 3, 4, 5, 19, and 20 each contains 1 frame memory (240x720+120x360+120x360 pixels per frame). Block 10 contains 1350 floating points for ACsrcs, one for each MB. Block 12 contains all the DCT coefficients of an entire frame. Blocks 22 and 23 contain 675 pairs of motion vectors (x and y components) of a field. Blocks 6 and 7 complete motion estimation for one field in 1/2 a frame time or 1/60 of a second. Block 8 performs DCT on luminance only, while block 11 performs DCT on all 3 components. Each must be completed in 1/2 a frame. Block 13 has to complete 5 iterations of quantizations of 8x8 DCT on an entire frame in a 1/4 frame or 1/120 of a second. Blocks 15 and 16 perform 8100 (1350 MBs x 6 8x8 blocks) operations in 1/2 a frame.

Similarly, the number of operations of each module per MB in the decoder (Fig.9) is shown on Table 2. VLC decoding on each MB is done once. Inverse quantization and inverse DCT of 8x8 block are performed 6 times in each MB. Table 4 lists the corresponding required performance/capacity. The inverse-operations in blocks 3 and 4 on the entire frame are completed in one frame time. The decoder requires 2 frame memories to store the reconstructed pictures and 1 frame memory in the post-processor to hold the output frame for scanning of odd and even lines.

7. Additional Functionalities

7.1 Fast Searches

Fast search is achieved by playing the sequence at a reduced frame rate and decoding only the intraframes at the decoder end. Fast search is essentially constrained by the amount of information to be read from the channel since the coded intraframes require significantly more bits than when motion compensated prediction, or interpolation is used. At 4 Mbits/sec, the number of bits required by an intra-coded picture is typically less than 600 kbits. Therefore, at most 6 intraframes can be decoded every second given this bit rate. A factor of 4 speedup is achieved by decoding every other I-frame and displaying it 6 times. This amounts to decoding 5 intraframes per second with the resulting bit rate of 3 Mbits/sec. Fast search in either forward or reverse direction is implemented in the same way. The demo tape shows a video sequence in f&t-forward and fast reverse with a speedup factor of 4. The DC-picture mode from MPEG 1 could also be used to implement fast search without adding complexity.

7.2 Low Delay Mode

Shorter delay may be achieved by changing some parameters in the HEM encoder. In the low delay mode, there is only one B-frame coded between two P-pictures (M-2). To reduce the necessary buffer size, the periodic reset of the I-frame is eliminated. The GOP header is still inserted every 12 frames to report the time codes, but the first frame of a GOP will no longer be an I-frame except for the first GOP of a sequence. To terminate predictive error propagation, every 2 out of 15 Super Slices in P-frames are coded in intramode; i.e., each P-frame will have 2 Super Slices coded without temporal prediction, and these MBs are coded with type intra. Hence there is no need to send extra information to define these "Intra-Slices". This implementation will sacrifice random accessibility, but such functionality is usually not necessary where low delay pictures are required.

Scene change is detected in P-frames only. When such condition occurs, no DCT coefficient will be sent for the following B-frame until the next P-frame is coded. Also, if the buffer occupancy exceeds three quarters of the buffer capacity, the same measure will be applied for the following B-frame.

Assuming a buffer size of 2 frames for the low delay mode, the overall delay is 193 ms.

$$\begin{aligned}\text{Overall Delay} &= \text{Encoding Delay} + \text{Buffer Size} + \text{Decoding Delay} \\ &= 2 + 1.3 + 2 + 0.5 \\ &= 5.8 \text{ frames}\end{aligned}$$

If the first I-frame is not available to the decoder as in the case of channel-hopping or delayed access, the decoder will fill up its reconstructed frame memories with the value 128 (the default reference DC values in intramode), and start decoding and displaying the reconstructed pictures once synchronization is established. The image quality will gradually build up in about 1/2 a second because the picture will be fully replenished by intracoding in 8 P-frames.

7.3 Cell Loss Resilience

The cell loss resilience characteristic of the algorithm has been demonstrated by the following simulation. The bit stream of a coded sequence is packetized into cells and each cell contains 48 bytes. No link between the cell boundary and the bit stream syntax is made.

A uniform distribution of packet loss is simulated by simply dropping a packet at a ratio of 10^{-4} . When a cell loss occurs, the decoder continues to decode the bit stream until it cannot go on. The decoder then looks for unique codewords (marks) to resynchronize. The undecodable blocks will be replaced with the corresponding segments from the previous frame according to the encoding order.

References

- [1] "MPEG Video Simulation Model Three", Document MPEG 90/41, ISO/MPEG Simulation Model Editorial Group.
- [2] "Coding of Moving Pictures and Associated Audio - for Digital Storage Media at up to about 1.5 Mbit/s - Part 2 Video", CD 11172-2.

8. Figures, Tables, Flowcharts, and Syntax Diagrams

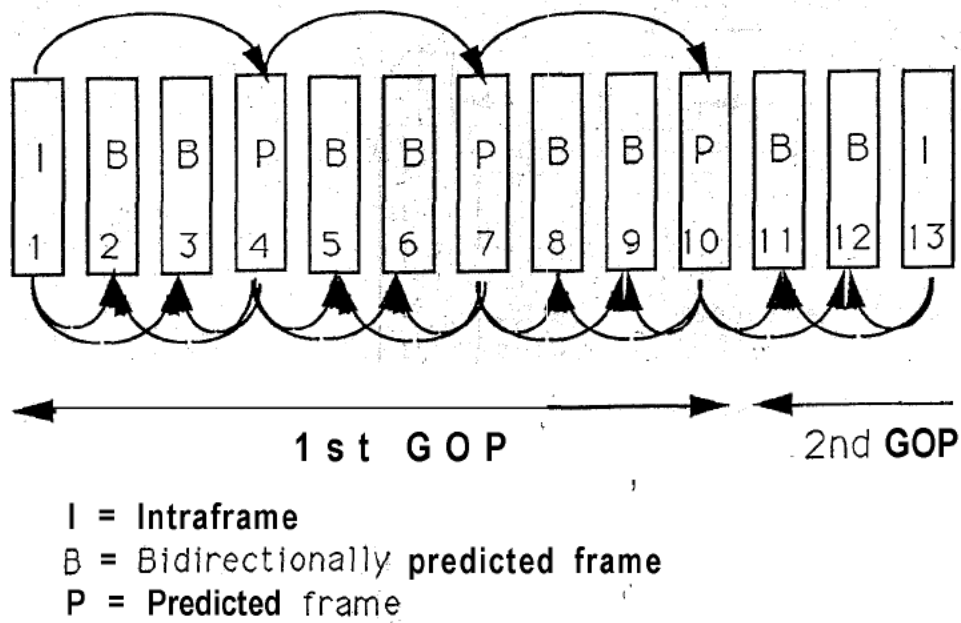


Fig. 1 A Group of Pictures (display order)
 The first GOP has 10 frames; only.
 Other GOPs have 12 frames each,

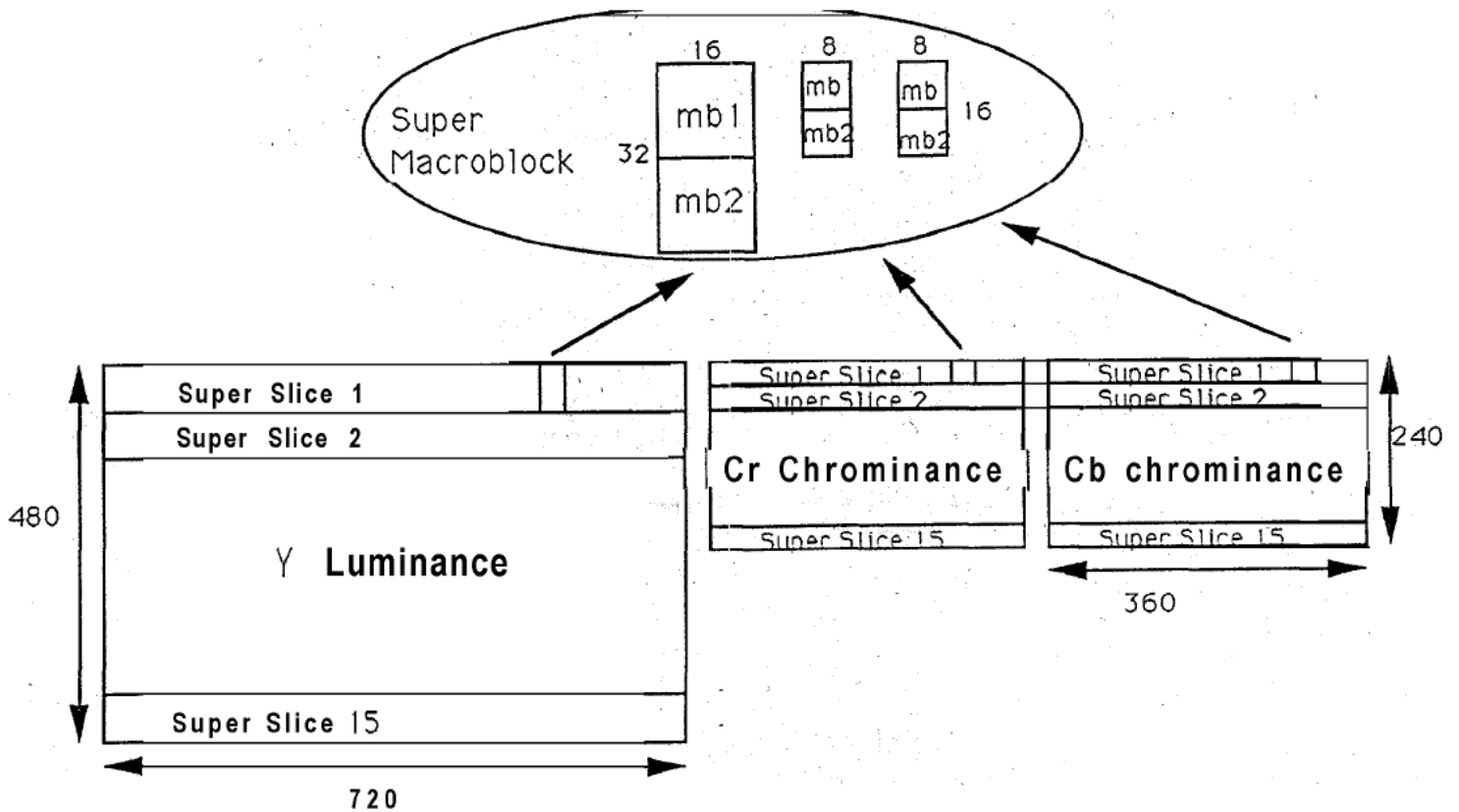


Fig.2 Data Structures --- Super Slice, Super Macroblock, Macroblock

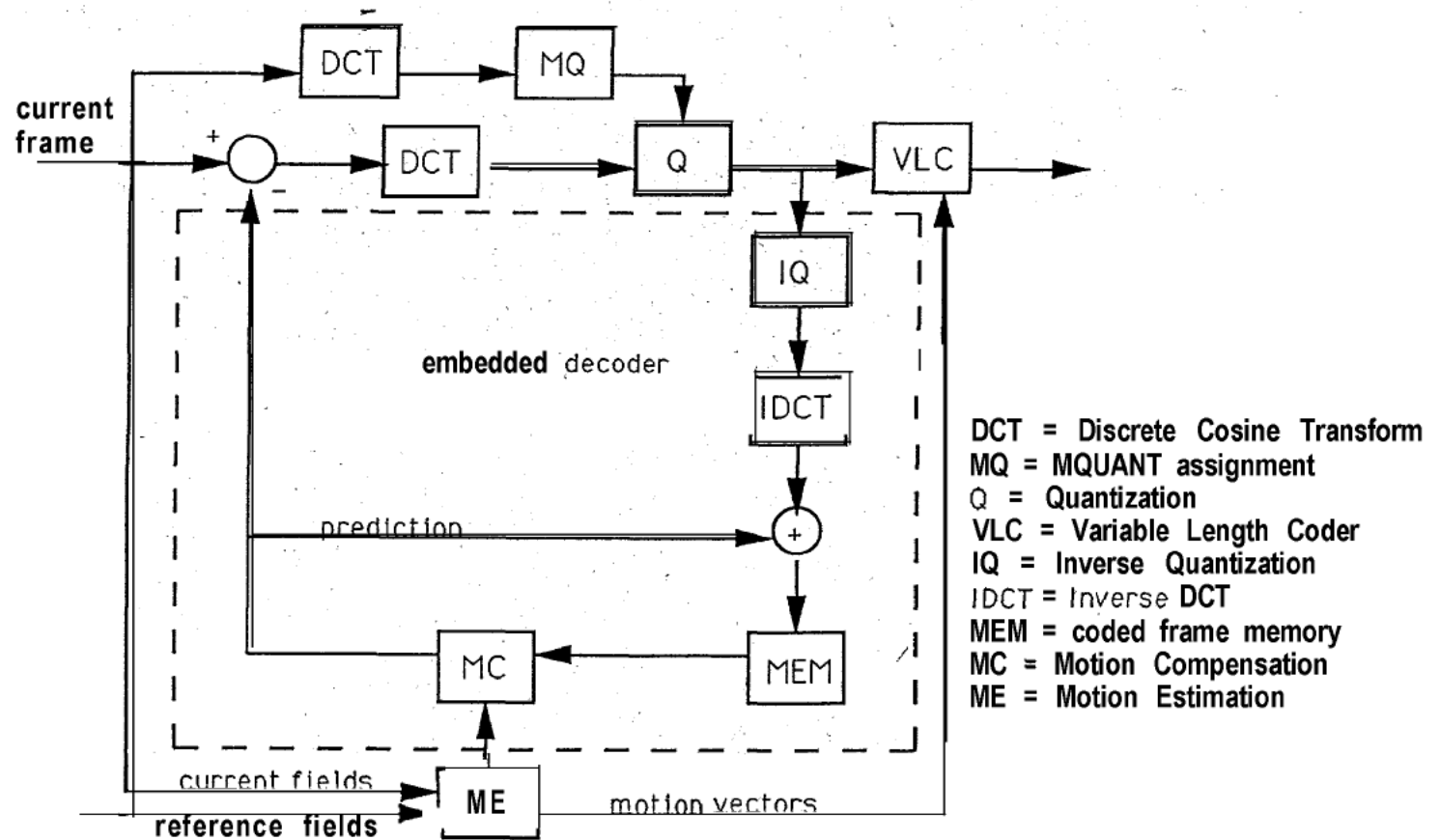


Fig.3 Predictive/Transform Coding Loop

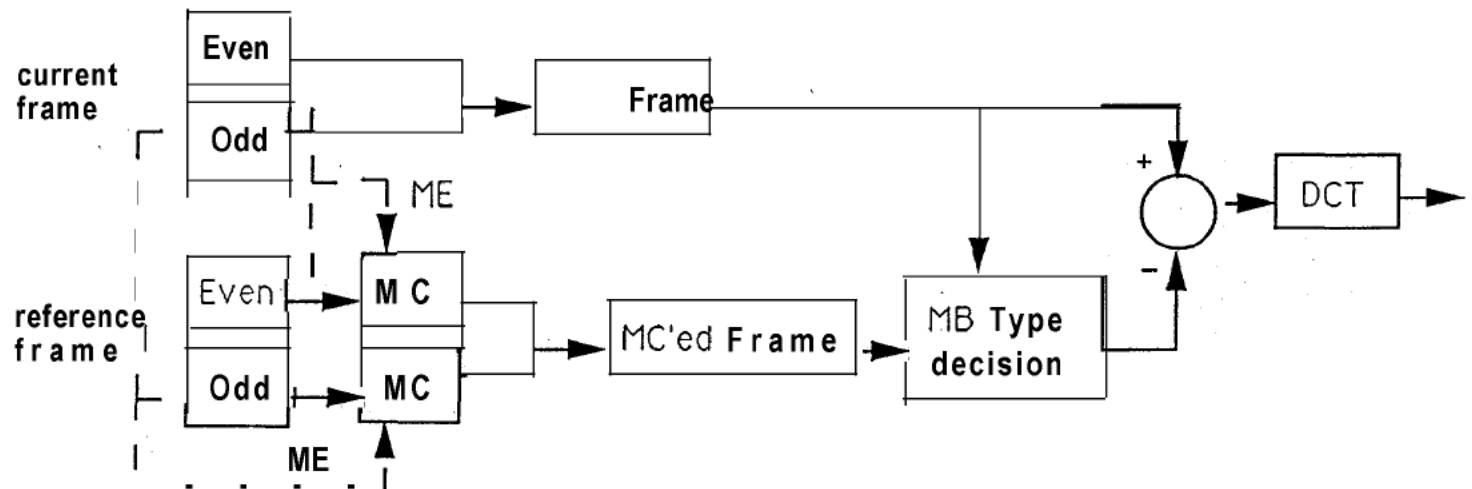


Fig.4 Field-based motion estimation/compensation

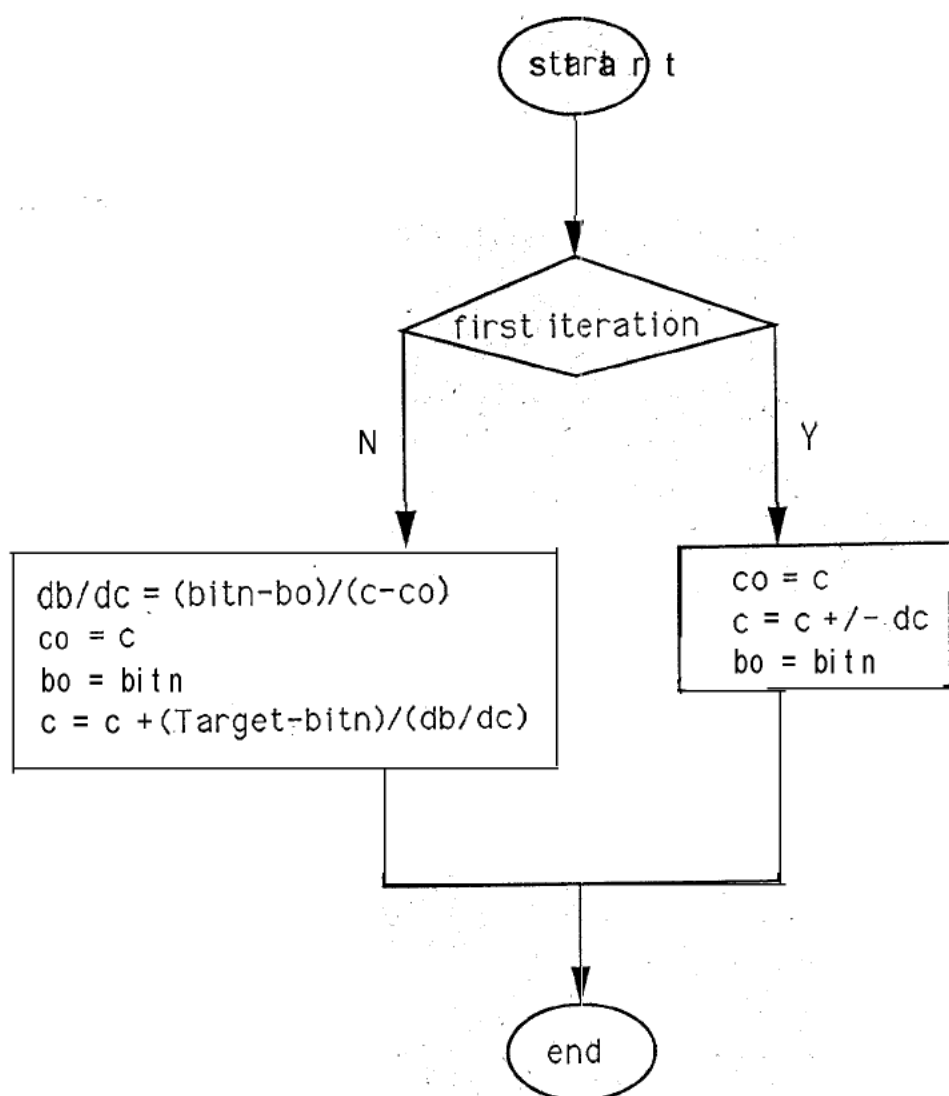


Fig. 6 Adjustment of c

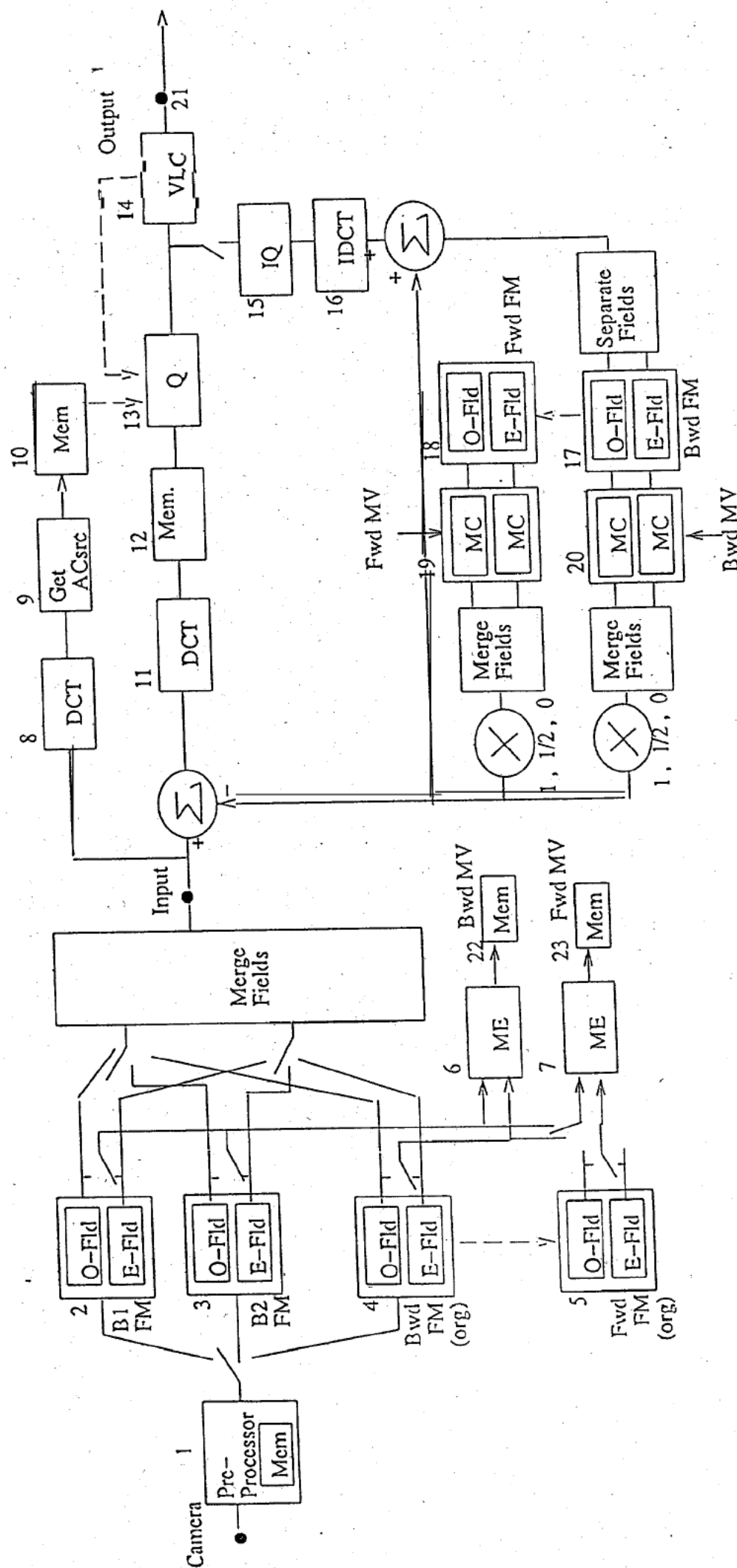


Fig.7 Encoder Complexity Block Diagram

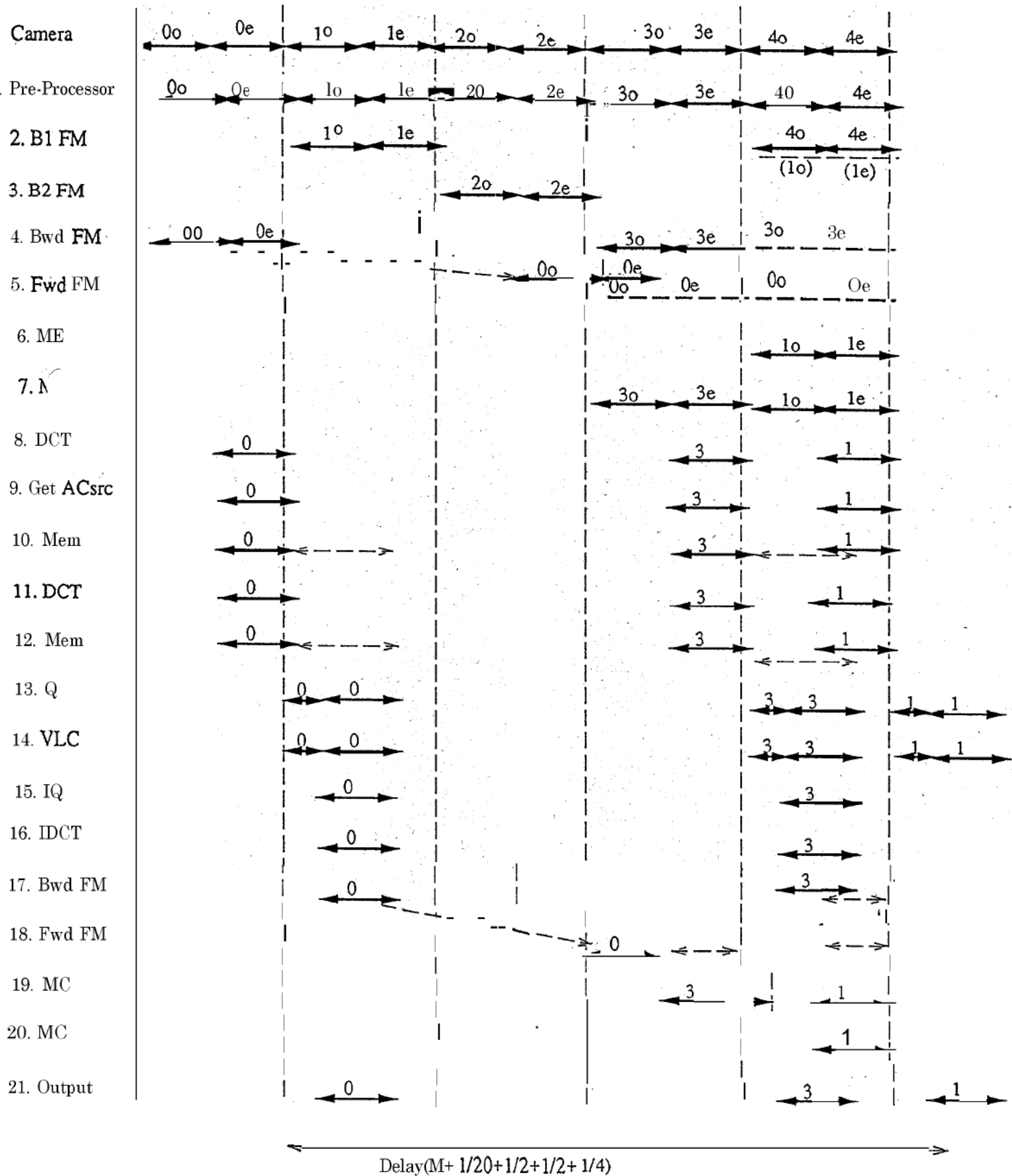


Fig8 Encoder Timing Chart

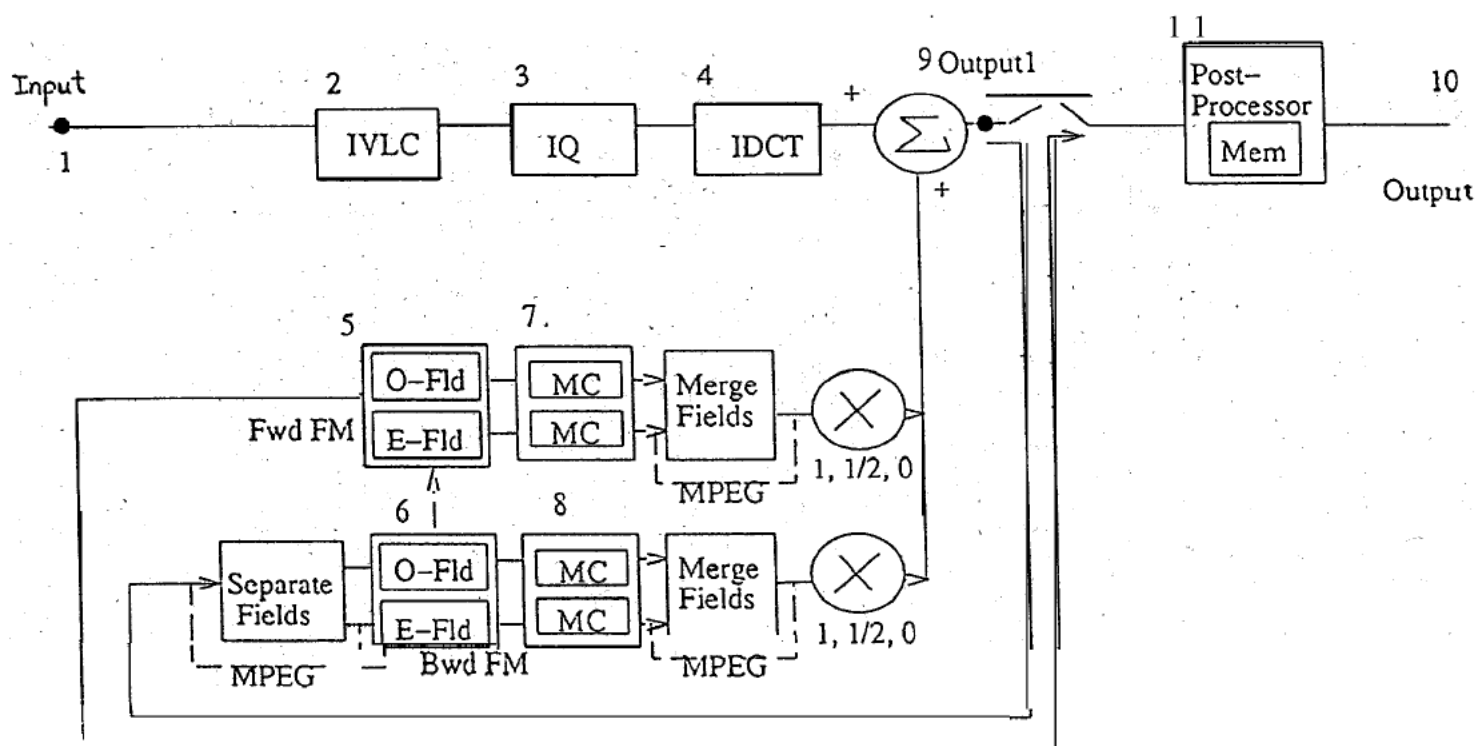


Fig. 9 Decoder Complexity Block Diagram

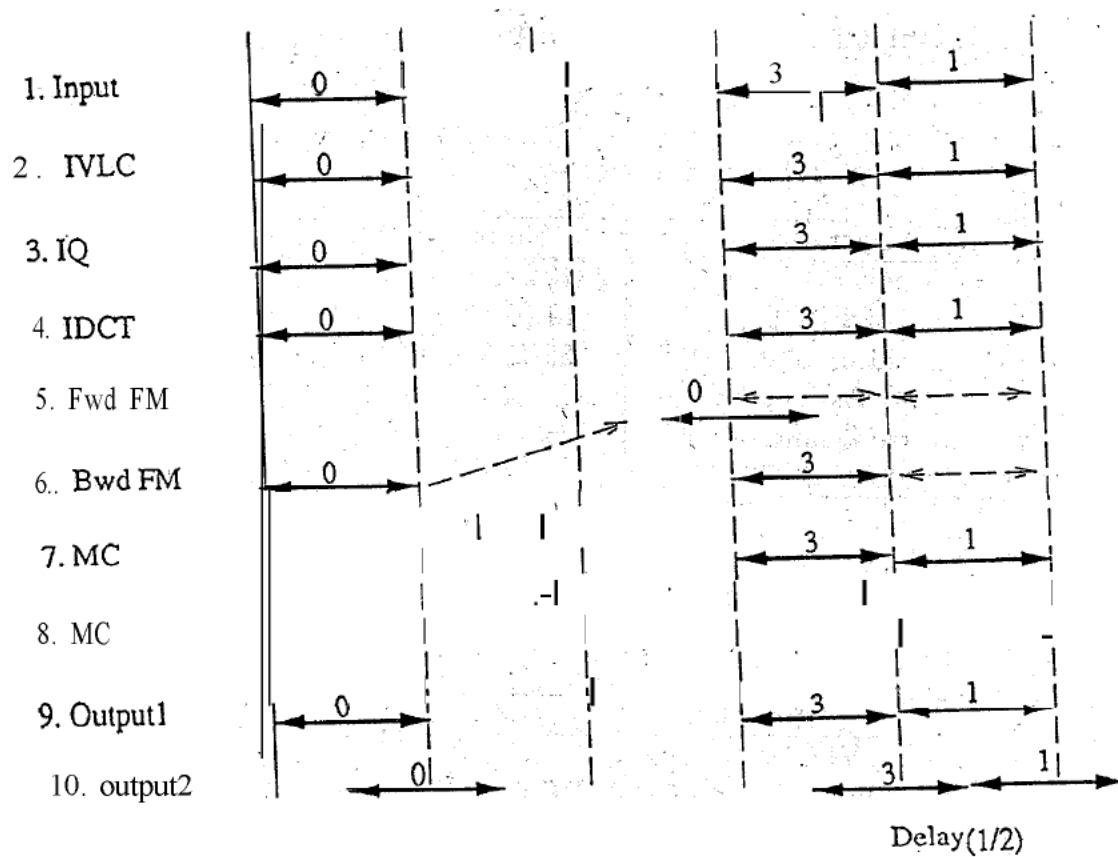


Fig. 10 Decoder Timing Chart

The number of operations per macro' block

Table 1: Encoder

Operation	Times
Motion Estimation	2 ($1/2 \times 2 \times 2$)
8x8 DCT	10 (6 + 4)
Quantization of 8x8 DCT	30 (6 x 5)
VLC Assignment	5
Inverse Quantization	6
8x8 I D C T	6

Table 2: Decoder

Operation	Times
VLC Decoding	1
Inverse Quantization of 8x8 IDCT	6
8 x 8 IDCT	6

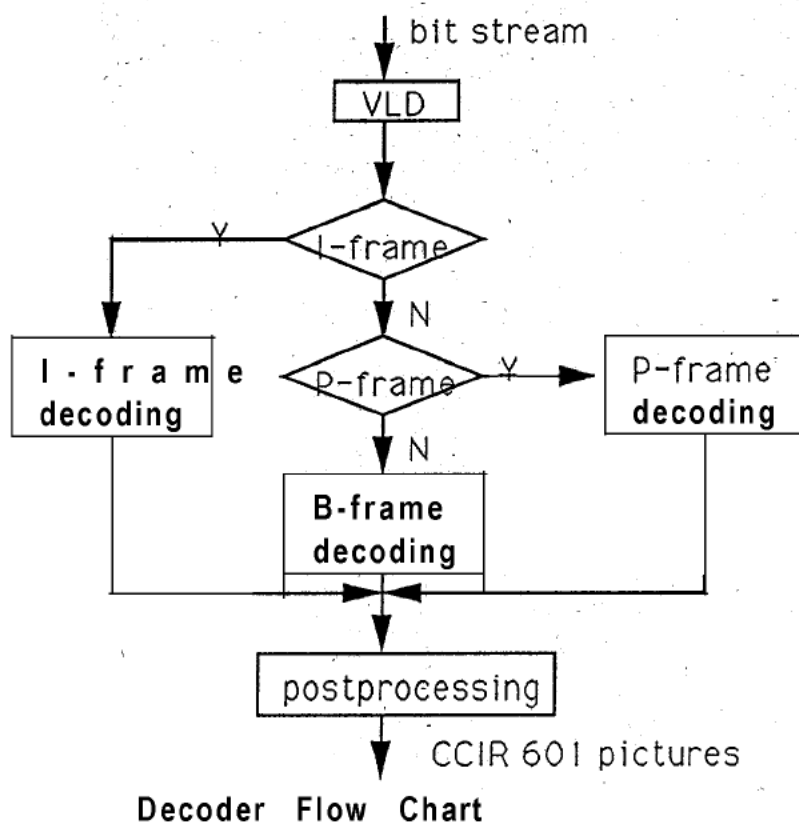
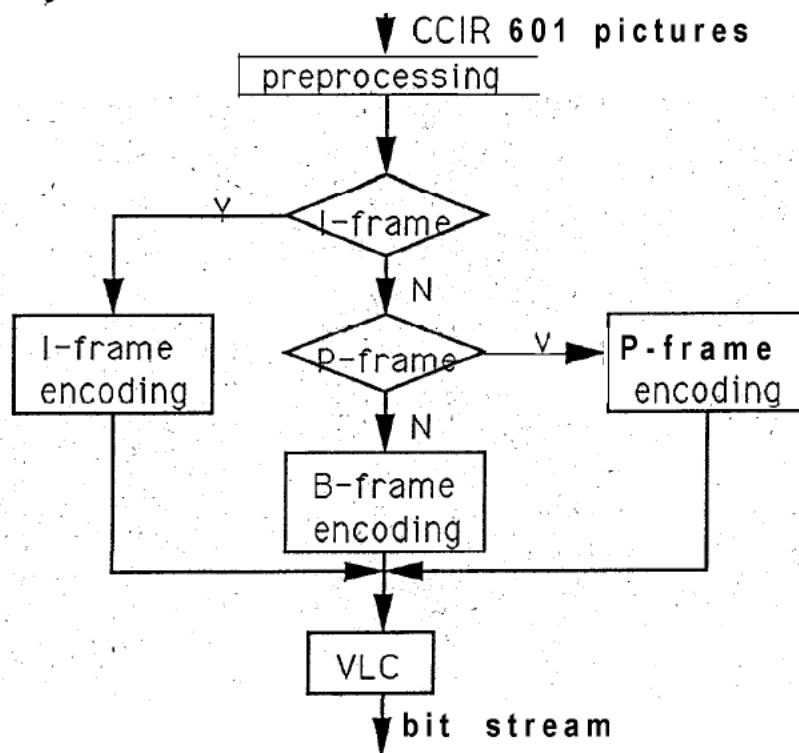
The performance/capacity requirements for each block

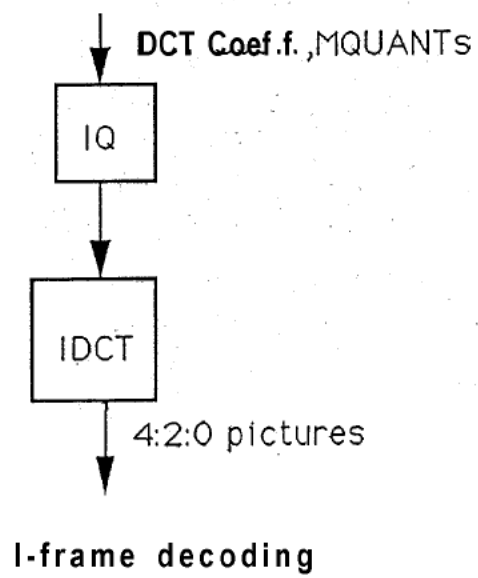
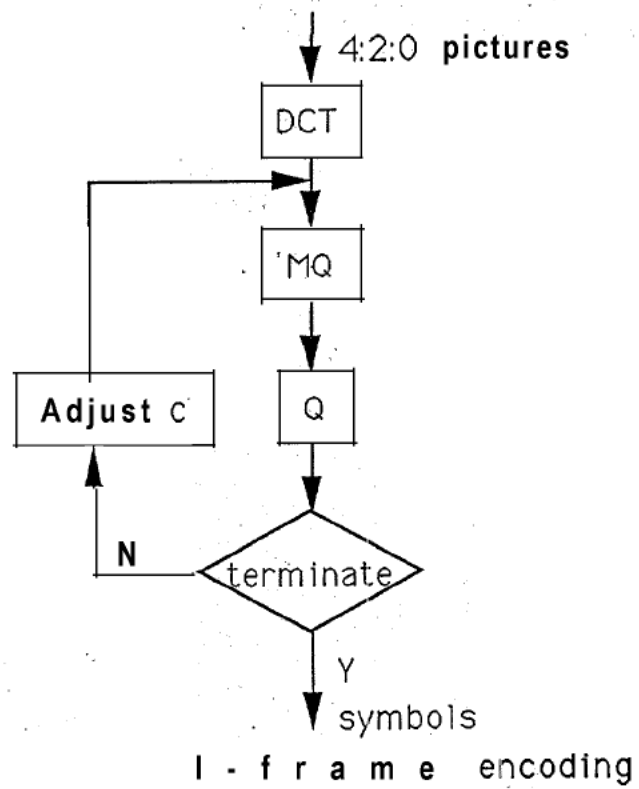
Table 3: Encoder

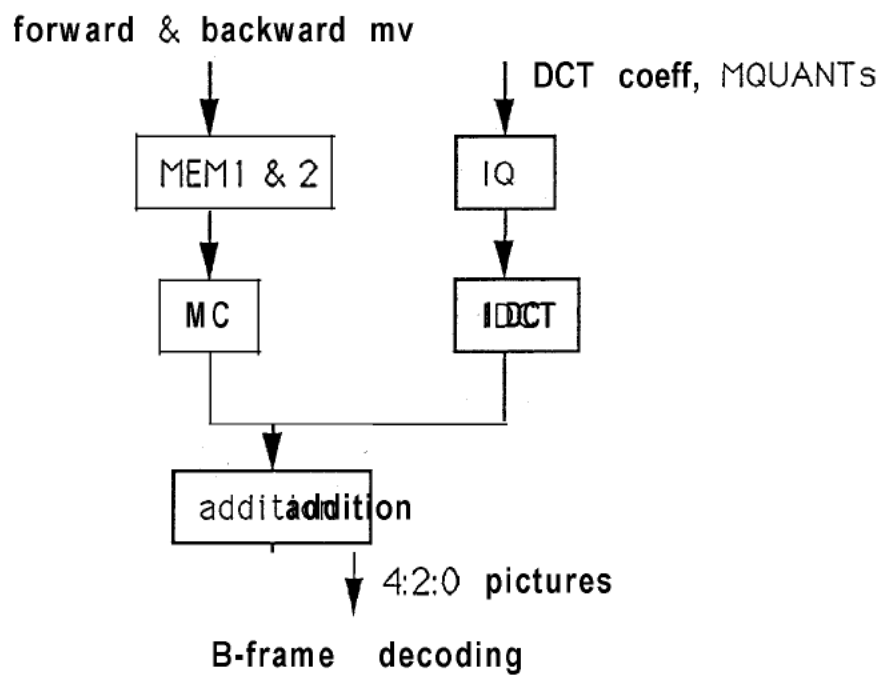
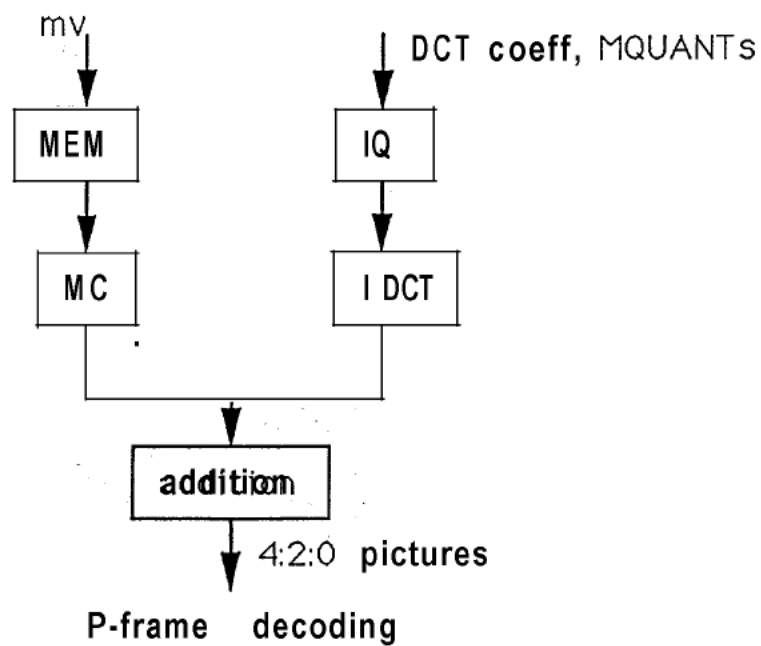
Block No.	Item	Performance/Capacity Requirements
1	Memory in Pre-Processor	37440 x 8[bit]
2	Memory	259200 x 8[bit] x 2
3	Memdry	259200 x 8[bit] x 2
4	Memory	259200 x 8[bit] x 2
5	Memory	259200 x 8[bit] x 2
6	Motion Estimation of 16x16 block	675 in 1/60[sec]
7	Motion Estimation of 16x16 block	675 in 1/60[sec]
8	8x8 DCT	5400 in 1/60[sec]
10	Memory	1350 x 32[bit]
11	8x8 DCT	8100 in 1/60[sec]
12	Memory	518400 x 12[bit]
13	Quantization of 8x8 DCT	40500 in 1/120[sec]
15	Inv-Quantization of 8x8 DCT	8100 in 1/60[sec]
16	8x8 Inv-DCT	8100 in 1/60[sec]
19	Memory	259200 x 8[bits] x 2
20	Memory	259200 x 8[bits] x 2
22	Memory	675 x 16[bit]
23	Memory	675 x 16[bit]

Table 4: Decoder

Block Ho.	Item	Performance/Capacity Requirements
3	Inv-Quantization of 8x8 DCT	8100 in 1/30[sec]
4	8x8 Inv-DCT	8100 in 1/30[sec]
5	Memory	259200 x 8[bit] x 2
6	Memory	259200 x 8[bit] x 2
11	Memory	518400 x 8[bit]







Macroblock Types for P-frames				
VLC code	New MQ	Forward Motion	Coded	Intra
1	0	1	1	0
01	0	0	1	0
001	0	1	0	0
00011	0	0	0	1
00010	1	1	1	0
00001	1	0	1	0
000001	1	0	0	1
0000001	0	0	0	0

9. Unix Listing of Coded Bitstreams and Statistics

total	29066			
drwxr-xr-x	2	ctc	512	Oct 15 09:30 ./
drwxrwxrwx	6	ctc	1024	Oct 15 09:29 ../
-rwxr-xr-x	1	ctc	2470312	Oct.14 17:23 GARDEN4*
-rwxr-xr-x	1	ctc	5575523	Oct.14 17:44 GARDEN9*
-rwxr-xr-x	1	ctc	2475177	Oct 14 17:59 MOBILE4*
-rwxr-xr-x	1	ctc	5568348	Oct 14 18:32 MOBILE9*
-rwxr-xr-x	1	ctc	5561873	Oct 14 18:47 POPPLE9*
-rwxr-xr-x	1	ctc	2473335	Oct 14 17:40 TENNIS4*
-rwxr-xr-x	1	ctc	5554939	Oct 14 18:13 TENNIS9*

**** FILE = GARDEN4.stat ****

Index	Type	Y S N R (d B)	Frame bits	Cumu. bits
=====				
0	I	34.01	568872	568872
1	P	30.72	21'6416	785288
2	B	30.78	43592	828880
3	B P	29.48	235360	1064240
4	B	29.16	35136	1136928
5	B	28.84	32760	1169688
6	P	29.33	240760	1410448
7	B	29.26	37264	1447712
8	B	28.85	35968	1483680
=====				
10	I	31.39	491000	491000
11	B	28.27	53232	544232
12	B	28.53	56264	600496
13	P	29.35	23'5592	836088
14	B	29.06	35552	871640
15	B	28.80	34000	905640
16	P	28.64	238872	1144512
17	B	27.74	45248	1189760
18	B	27.39	47856	1237616
19	P	28.35	236256	1473872
20	B	28.20	32536	1506408
21	B	27.86	32296	1538704
=====				
22	I	30.47	440368	440368
23	B	2.8	29752	470120
24	B	28.41	32848	502968
25	P	30.56	280304	783272
26	B	29.02	57288	840560
27	B	29.63	42496	883056
28	P	29.81	253584	1136640
29	B	29.45	34632	1171272
30	B	28.96	34032	1205304
31	P	29.54	265048	1470352
32	B	29.06	35096	1505448
33	B	28.69	40352	1545800
=====				
34	I	31.25	500608	500608
35	B	29.08	40760	541368
36	B	29.93	31088	572456
37	P	30.16	263736	836192
38	B	30.11	35128	871320
39	B	29.63	39688	911008
40	P	29.04	259312	1170320
41	B	29.50	33152	1203472
42	B	28.71	32224	1235696
43	P	28.78	251544	1487240
44	a	27.85	40480	1527720
45	B	27.66	40752	1568472

46	I	30.44	495224	495224.
47	B	28.92	29928	525152
48	B	29.48	27944	553096
49	P	29.79	262000	815096
50	B	28.99	37536	852632
51	B	28.72	40752	893384
52	P	29.36	224624	1118008
53	B	27.13	57256	1175264
54	B	27.314	56824	1232088
55	P	28.65	222424	1454512
56	B	28.24	35704	1490216
57	B	28.20	33744	1523960
58	I	29.83	440536	1440536
59	B	29.02	25760	466296
60	B	29.43	25176	491472
61	P	29.23	260792	752264
62	B	28.18	47376	799640
63	B	28.05	45008	844648
64	P	28.67	238664	1083312
65	B	29.24	37848	1121160
66	B	28.47	40080	1161240
67	P	29.65	257944	1419184
68	B	28.93	32944	1452128
69	B	29.08	33712	1485840
70	I	31.44	547272	547272
71	B	30.54	36288	583560
72	B	30.05	37488	621048
73	P	29.74	262544	883592
74	B	29.57	40088	923680
75	B	29.41	33144	962824
76	P	29.79	2158736	1221560
77	B	29.38	37440	1259000
78	B	29.68	315624	1294624
79	P	29.82	235928	1530552
80	B	29.61	35544	1566096
81	B	29.30	34672	1600768
82	I	31.69	550504	550504
83	B	29.27	42424	592928
84	B	29.85	36792	629720
85	P	30.02	243440	873160
86	B	30.54	34296	907456
87	B	30.00	36960	944416
88	P	29.80	239544	1183960
89	B	30.04	35376	1219336
90	B	29.74	35792	1255128
91	P	28.97	244864	1499992
92	B	29.34	28888	1528880
93	B	28.65	-31632	1560512
94	I	31.16	537320	537320
95	B	29.73	32496	569816
96	B	29.27	38832	608648

97	P	29.38	278824		887.472
98	B	27.87	51200		938672
99	B	27.67	-48896		987568
100	P	28.19	230496		1218064
101	B	27.02	51792		1269856
102	B	27.02	56144		1326000
103	P	27.12	207320		1533320
104	B	26.43	50072		1583392
105	B	26.00	58920		1642312
=====					
106	I	28.61	424120		424120
107	B	26.33	55632		479752
108	B	26.69	45040		524792
109	P	27.98	237608		762400
110	B	27	39056		8.01456
111	B	27.19	39680		841136
112	P	27.61	234944		1076080
113	B	26.60			1122680
114	B	26.79	36912		1159592
115	P	27.51	231600		13.91192
116	B	26.90	39408		1430600
117	B	26.67	40232		1470832
=====					
118	I	29.63	480224		480224
119	B	28.57	25840		506064
120	B	28	25648		531712
121	P	29.22	279624		811336
122	B	28	40304		851640
123	B	28.49	38960		890600
124	P	29.44	260776		1153376
125	B	29.11	3332-8		1184704
126	B	29.00	34272		1218976
127	P	28.39	256576		1475552
128	B	27.82	34808		1510360
129	B	27.56	32960		1543320
=====					
130	I	29.90	506048		506048
131	B	28.98	23872		529920
132	B	29.16	24032		553952
133	P	29.46	267448		821400
134	B	29.30	34520		855920
135	B	29.13	35304		891224
136	P	30.07	266016		1157240
137	B	29.56	-34024		1191264
138	B	29.65	35720		1226984
139	P	30.07	267776		1494760
140	B	29	32440		1527200
141	B	29.11	35400		1562600
=====					
142	I	-31.20	571464		571464
143	B	30.85	25240		596704
144	B	30.45	24824		621528
145	P	28.78	251944		873472
146	B	28.92	30208		903680
147	B	28.52	29672		933352
148	P	29.74	270640		1203992

149	B	29.59	21704	1225696
150	B	29.74	10000	1235696

=====

For I pictures
total motion vector bits = 0
total luminance bits = 5697720
total chrominance bits = 785790
total misc. overhead bits = 70055)

For B pictures
total-motion vector bits = 1150888
total luminance bits = 1802183
total chrominance bits = 127603
total misc. overhead-bits = 656390

For P pictures
total motion vector bits = 312402
total luminance bits = 8161497
total chrominance bits = 741791
total misc. overhead bits = 256182

, -GRAND TOTAL BIT COUNT = 19762496

**** FILE = GARDEN9.stat****

Index	Type	YSNR(dB)	Frame bits	Cumu. bits
=====				
0	I	37.89	760216	760216
1	P	35.39	392704	1152920
2	B	35.55	196248	1349168
3	B	35.27	187744	1536912
4	P	35.17	473048	2009960
5	B	34.56	168624	2178584
6	B	34.45	174120	2352704
7	P	36.25	533168	2885872
8	B	35.45	193544	3079416
9	B	35.26	171280	3250696
=====				
10	I	37.62	824088	824088
11	B	33.71	178920	1003008
12	B	33.91	198064	1201072
13	P	34.19	411768	1612840
14	B	34.52	175656	1788496
15	B	34.22	175416	1963912
16	P	34.95	516400	2480312
17	B	32.80	163160	2643472
18	B	32.89	180552	2824024
19	P	33.84	447280	3271304
20	B	33.16	119200	3390504
21	B	32.88	123856	3514360
=====				
22	I	36.48	771136	771136
23	B	33.69	127328	898464
24	B	33.77	163816	1062280
25	P	36.15	493680	1555960
26	B	34.89	229088	1785048
27	B	35.11	181472	1966520
28	P	35.15	478208	2444728
29	B	34.59	154536	2599264
30	B	33.81	141256	2740520
31	P	34.83	462392	3202912
32	B	33.91	141368	3344280
33	B	33.68	165040	3509320
=====				
34	I	36.53	789176	789176
35	B	33.90	166816	955992
36	B	34.61	137176	1093168
37	P	34.93	475480	1568648
38	B	34.64	139472	1708120
39	B	34.41	159464	1867584
40	P	34.98	538072	2405656
41	B	34.55	149776	2555432
42	B	34.28	162296	2717728
43	P	34.72	511880	3229608
44	B	33.61	193568	3423176
45	B	33.48	198728	3621904

=====					
-46	I	36.46	825560.	825560	
47	B	33.90	1 4 3 2 3 2	968792	
48	B	34.24	122032	1090824	
49	P	35.80	543056	1633880	
50	B	34.33	1'71512	1805392	
51	B	33.55	-144360	1949752	
52	P	35.76	513544	2463296	
53	B	31.22	144424	2607720	
54	B	31.29	138904	2746624	
55	P	32.96	3 6 9 2 3 2	3115856	
56	B	-33.42	146976	3262832	
57	B	33.31	1 4 8 3 3 6	3411168	
=====					
58	I	35.23	716168	716168	
59	B	34.15	1154304	870472	
60	B	34.33	139224	1009696	
61	P	34.70	515488	1525184	
62	B	33.45	1134848	1720'032	
63	B	33.76	205984	1926016	
64	P	34.53	4'74208	2400224	
65	B	34.36	144576	2 5 4 4 8 0 0	
66	B	33.96	1 6 4 2 5 6	2709056	
67	P	35.60	512632	3221688	
68	B	34.16	143328	3365016	
69	B	34.26	1 4 0 7 8 4	3505800	
=====					
70	I	36.17	8 1 2 9 7 6 0	8.29760	
71	B	3 4 . 7 6	125912	955672	
7 2	B	3 4 . 2 9	148008	1103680	
7 3	P	3 4 . 2 9	464128	-1567808	
74	B	3 3 . 7 3	1'73112	1740920	
75	B	33.29	152192	1893112	
76	P	34.49.	4'71664	2364776	
77	B	33.44	x 3 9 1 9 2	2503968	
78	B	33.77	122296	2626264	
7 9	P	35.68	504040	3130304	
80	B	34.86	186048	3316352	
81	B	3 4 . 9 2	193136.	3509488	
=====					
82	-1	37.40	8 1 3 7 3 0 4	887304	
83	B	34.73	1-99208	1086512	
84	B	35.11	198304	1284816	
85	P	3 4 . 7 7	4 4 4 9 0 4	1729720	
86	B	3 4 . 9 0	128424	1858144	
87	B	34.45	134712	1992856	
88	P	34.56	426800	2419656	
89	B	34.40	129408	2549064	
90	B	34.28	131568	2680632	
91		34.49	488360	3168992	
9 2		34.21	131176	3300168	
		33.46	130288	3430456	
-----i-i-=====					
94	I	36.49	837504	837504	
95	B	34.36	'1136488	973992	
96	B	34.17	1'73400	1147392	

97	P	34.59	505688	1653080
98	B	33.47	214232	1867312
99	B	33.24	-211304	2078616
100	P	33.69	471928	2550544
101	B	31.97	185560	2736104
102	B	32.05	186320	2922424
103	P	31.97	395296	3317720
104	B	31.57	183696	3501416
105	B	31.33	-214592	3716008

106	I	33.55	669640	669640
107	B	31.17	-193504	863144
108	B	31.22	167432	1030576
109	P	32.60	418800	1449376
110	B	31.73	140608	1589984
111	B	32.77	145200	1735184
112	P		466888	2202072
113	B	31.61	-194888	2396960
114	B	31.63	15952.8	2556488
115	P	32.77	455840	3012328
116	B	31.83	160336	3172664
117	B	32.06	186184	3358.848

118	I	35.00	773896	773896
119	B	34.73	244272	1018168
120	B	34.03	165632	1183800
121	P	-34.66	546152	1729952
122	B	33.26	153536	18.83488
123	B	33.43	151896	203.5384
124	P	34.38	469600	2504984
125	B	34.01	151248	2656232
126	B	33.86	155632	2811864
127	P	33.39	472336	3284200
128	B	32.45	152328	3436528
129	B	32.23	144384	3580912

130	I	34.71	771016	771016
131	B	33.41	129896	900912
132	B	33.91	134520	1035432
133	P	35.37	572976	1608408
134	B	34.45	192992	1801400
135	B	34.48	181760	1983160
136	P	35.27	461736	2444896
137	B	34.55	162848	2607744
138	B	34.57	150600	2758344
139	P	35.31	490280	3248624
140	B	34.50	168296	3416920
141	B	34.32	186472	3603392

142	I	35.87	863328	863328
143	B	35.16	143504	1006832
144	B	34.85	136.128	1142960
145	P	34.02	500520	1643480
146	B	33.19	148432	1791912
147	B	33.08	147416	1939328
148	P	34.97	480744	2420072

149	B	34.64	146368	2566440
150	B	35.03	25392	2591832

=====

For I pictures'
total motion vector bits = 0
total luminance bits = 8770239.
total chrominance bits = 1486024
total misc. overhead bits = 62529

For B- pictures'
total motion vector bits = 1128323
total luminance bits = 13296391
total chrominance bits = 971413
total misc. overhead bits = 7.18345

For P pictures'
total motion vector bits = 313035
total luminance bits = 15884984
total chrominance bits = 1751736.
total misc. -overhead bits = 221171

GRAND TOTAL BIT COUNT = 44604184

**** FILE = MOBILE4.stat ****

Index	Type	YSNR(dB)	Frame bits	Cumu. bits
=====				
0	I	28.75	560856	560856
1	P	26.78	210552	771408
2	B	26.03	43184	814592
3	B	25.65	42920	857512
4	P	25.91	194376	1051888
5	B	25.64	50216	1102104
6	B	25.34	50384	1152488
7	P	26.27	224224	1376712
8	B	25.36	52952	1429664
9	B	25.40	52056	1481720
=====				
10	I	26.90	446144	446144
11	B	25.96	42296	488440
12	B	26.10	41392	529832
13	P	27.93	256312	786144
14	B	26.95	63472	849616
15	B	26.89	64064	913680
16	P	27.85	186848	1100528
17	B	27.19	41880	1142408
18	B	27.16	42392	1184800
19	P	27.79	201512	1386312
20	B	27.14	40936	1427248
21	B	27.08	41200	1468448
=====				
22	I	30.19	644072	644072
23	B	28.62	40832	684904
24	B	28.39	44816	729720
25	P	29.71	237896	967616
26	B	29.09	38808	1006424
27	B	28.88	39608	1046032
28	P	29.46	233560	1279592
29	B	28.72	40552	1320144
30	B	28.81	39680	1359824
31	P	29.64	205320	1565144
32	B	27.93	56504	1621648
33	B	28.01	51120	1672768
=====				
34	I	29.72	623328	623328
35	B	28.00	43104	666432
36	B	28.21	41920	708352
37	P	28.84	187408	895760
38	B	27.73	49600	945360
39	B	27.57	53400	998760
40	P	27.78	164824	1163584
41	B	27.62	35176	1198760
42	B	27.59	38944	1237704
43	P	27.64	182424	1420128
44	B	27.17	37272	1457400
45	B	27.15	35848	1493248

```

=====
46      I      29.11      571920      571920
47      B      27.59      41040      612960
48      B      27.59      39592      652552
49      P      28.48      202552      855104
50      B      27.84      43488      898592
51      B      27.63      52856      951448
52      P      28.34      2135696      1157144
53      B      27.33      58744      1215888
54      B      27.27      56680      1272568
55      P      27.84      209768      1482336
56      B      27.13      60000      1542336
57      B      26.98      50680      1593016
=====
58      I      28.67      550128      550128
59      B      27.54      37256      587384
60      B      27.39      36160      625544
61      P      27.57      203104      828648
62      B      27.31      35800      864448
63      P      27.22      33816      898264
64      P      27.22      234808      1133072
65      B      26.97      37488      1170560
66      B      26.77      36560      1207120
67      P      27.23      234672      1441792
68      B      26.63      43064      1484856
69      B      26.42      45824      1530680
=====
70      I      28.56      545232      545232
71      B      27.17      44296      589528
72      B      27.14      45584      635112
73      P      27.27      223,040      858152
74      B      26.74      45952      904104
75      B      26.64      48312      952416
76      P      27.33      212112      1164528
77      B      26.32      51888      1216416
78      B      26.24      50736      1267152
79      P      27.90      185224      1452376
80      B      26.86      51616      1503992
81      B      26.86      45568      1549560
=====
82      I      29.65      618088      618088
83      B      27.74      51128      669216
84      B      27.62      50712      719928
85      P      28.72      1,64512      884440
86      B      27.65      51680      936120
87      B      27.67      53200      989320
88      P      28.72      1,88232      1177552
89      B      27.55      56160      1233712
90      B      27.29      54424      1288136
91      P      27.86      3,72920      1461056
92      B      27.01      48664      1509720
93      B      26.97      48144      1557864
=====
94      I      28.96      563184      563184
95      B      27.70      47920      611104
96      B      27.80      43224      654328
=====

```

88	P	28.78,	209320	863648
99	B	28.19	48256	911904
100	P	28.16	47952	959856
101	B	28.26	215232	1175088
102	B	28.08	-38048	1213136
103	P	27.86	34944	1248080
104	B	28.18	217008	1465088
105	B	27.98	37120	1502208
106	B	27.74	41472	1543680
=====				
106	I	29.00	555848	555848
107	B	27.70	4.2112	-597960
108	B	28.04	41168	639128
109	P	29.00	217984	857112
110	B	27.82	52520	909632
111	B	27.88	52672	962304
112	P	28.98	211768	1174072
113	B	27.94	57680	1231752
114	B	27.66	50528	1282280
115	P	29.11	192528	1474808
116	B	28.11	50376	1525184
117	B	-27.85	56928	1582112
=====				
118	I	29.60	606088	606088
119	B	28.79	35072	641160
120	B	28.85	35816	676976
121	P	29.53	206400	883376
122	B	28.32	55464	938840
123	B	28.60	53096	991936
124	P	29.16	164488	1156424
125	B	28.01	49048	1205472
126	B	28.3'5	44752	1250224
127	P	29.23	159352	1409576
128	B	28.21	44320	1453896
129	B	28.33	43552	1497448
=====				
130	I	30.43	637688	637688
131	B	29.46	45104	68279.2
132	B	28.79	49248	732040
133	P	30.22	196904	928944
134	B	29.53	44352	973296
135	B	29.29	42968	1016264
136	P	29.97	185152	1201416
137	B	29.38	43952	1245368
138	B	29.40	43872	1289240
139	P	29.24	165712	1454952
140	B	28.98	36104	1491056
141	B	28.73	39272	1530328
=====				
142	I	30.69	705896	705896
143	B	29.70	37944	743840
144	B	29.28	40856	784696
145	P	30.13	208200	992896
146	B	29.54	43392	1036288
147	B	29.64	43320	1079608
148	P	28.99	169408	1249016

150 ' B 28.990 B9648 1200864

=====

For I pictures

total motion vector bits = 10
total luminance bits = 6302990
total chrominance bits = 1230860
total misc. overhead bits = 94622

For B pictures

total motion vector bits = 815123
total luminance bits = 2.825184
total chrominance bits = 194888
total misc. overhead bits = 696397

For P pictures

total motion vector bits = 197006
total luminance bits = 6283192
total chrominance bits = 813606
total misc. overhead bits = 347548

GRAND TOTAL BIT COUNT = 19801416

**** FILE = MOBILE9.stat ****

Index	Type	YSNR(dB)	Frame	bits	Cumu.	bits
=====						
0	I	32.87		760640	760640	
1	P	31.04		410520	1171160	
2	B	30.19		188112	1359272	
3	B	29.98		191528	1550800	
4	P	30.97		432264	1983064	
5	B	30.05		187080	2170144	
6	B	29.91		190648	2360792	
7	P	31.19		422160	2782952	
8	B	29.96		185376	2968328	
9	B	29	93	186920	3155248	
=====						
10	I	32.38		748152	748152	
11	B	30.44		166120	914272	
12	B	30.71		177232	1091504	
13	P	33.69		497144	1588648	
14	B	32.03		219120	1807768	
15	B	31.97		226960	2034728	
16	P	32.91		371400	2406128	
17	B	32.20		174528	2580656	
18	B	32.16		167944	2748600	
19	P	32.81		399600	3148200	
20	B	32.07		174104	3322304	
21	B	32.06		173592	3495896	
=====						
22	I	35.14		952320	952320	
23	B	33.15		171936	1124256	
24	B	33.07		188552	1312808	
25	P	33.64		397456	1710264	
26	B	33.29		155400	1865664	
27	B	33.06		164872	2030536	
28	P	33.87		417216	2447752	
29	B	32.88	161680		2609432	
30	B	32.85		151616	2761048	
31	P	35.17		498032	3259080	
32	B	32.13		184920	3444000	
33	B	32.20		177088	3621088	
=====						
34	I	35.17		952792	952792	
35	B	32.96		229424	1182216	
36	B	33.05		221408	1403624	
37	P	33.22		322712	1726336	
38	B	32.21		197648	1923984	
39	B	32.14		206960	2130944	
40	P	32.51		363112	2494056	
41	B	31.68		123888	2617944	
42	B	31.87		138304	2756248	
43	P	32.86		396264	3152512	
44	B	32.13		167728	3320240	
45	B	32.05		175448	3495688	

```

=====
46      I      -34.29'      864120      864120
4  7    B      32.28      185920      1050040
48      B      32.27      180440      1230480
49      P      33.09      370528      1601008
50      B      32.48      170584      1771592
51      B      32.39      192112      1963704
52      P      32.92      -367040      2330744
53      B      32.21      -218976      2549720
54      B      32.18      209488      2759208
55      P      32.21      392360      3151568
56      B      31.20      176136      3327704
57      B      31.28      166152      3493855
=====
58      I      33.29      779808      779808
59      B      32.18      186992      966800
60      B      32.04      197368      1164168
61      P      32.86      450968      1615136
62      B      32.07      191176      1806312
63      B      32.05      181016      1987328
64      P      32.81      480568      2467896
65      B      32.05      189216      2657112
66      B      31.91      184504      2841616
67      P      32.78      476816      3318432
68      B      31.93      207416      3525848
69      B      31.01      165616      3691464
=====
70      I      33.19      780184      780184
71      B      31.50      158808      938992
72      P      31.82      168080      1107072
73      B      31.82      410280      1517352
74      B      30.94      173448      1690800
75      B      30.85      176144      1866944
76      P      31.76      377184      2244128
77      B      30.77      1135824      -2429952
78      B      30.66      1135952      2615904
79
80      B P      32.25 31.06 3154168 163576      2970072 3133648
81      B      31.71      199464      3333112
=====
82      I      34.28      862688      862688
83      B      32.24      189512      1052200
84      B      32.14      206272      1258472
85      P      33.94      414848      1673320
86      B      32.93      255576      1928896
87      B      32.89      250992      2179888
88      P      33.00      332184      2512072
89      B      32.12      207304      2719376
90      B      31.98      204880      2924256
91      P      32.38      364888      3289144
92      B      31.13      174320      3463464
93      B      31.14      168960      3632424
=====
94
95      B I      33.18 31.70 140168 781712      781712 921880
96      B      32.23      179080      1100960
=====

```

97	P	33.78	460256	1561216
98	B	33.11	184376	1745592
99	B	33.11	175168	1920760
100	P	33.88	471352	2392112
101	B	33.33	164824	2556936
102	B	33.19	160144	2717080
103	P	33.97	47.5048	3192128
104	B	33.29	152176	3344304
-105	B	33.32	164072	3508376
=====				
106	I	35.26	961352	961352
107	B	32.51	172472	1133824
108	B	32.60	16872-8	1302552
109	P	34.11	425896	1728448
110	B	32.92	-213488	1941936
111	B	32.91	213024	2154960
112	P	33.03	340544	2495504
113	B	32.31	180560	2676064
114	B	32.13	191432	2867496
115	P	32.98	318376	-3185872
116	B	32.31	168056	3353928
117	B	32.09	193592	3547520
=====				
118	I	34.30	850544	850544
119	B	33.42	178768	10219312
120	B	33.47	169296	1198608
121	P	34.06	371264	1569872
122	B	32.99	232592	1802464
123	B	33.18	209176	2011640
124	P	33.31	3135.84	2325224
125	B	32.42	188480	2513704
126	B	32.62	161008	2674712
127	P	34.08	369808	3044520
128	B	32.84	146160	3190680
129	B	33.00	163112	3353792
=====				
130	I	34.96	930120	930120
131	B	-33.71	170736	1100856
132	B	33.31	203144	1304000
133	P	34.21	340104	1644104
134	B	33.72	180568	1824672
135	B	34.76	-192072	2016744
136	P	...	362392	2379136
137	B	33.81	187720	2566856
138	B	33.83	173720	2740576
139	P	34.16	382432	3123008
140	B	33.77	161784	3284792
141	B	33.56	175904	3460696
=====				
142	I	36.11	1001368	1001368
143	B	34.25	149808	1151176
144	B	33.97	178696	1329872
145	P	34.91	402104	1731976
146	B	34.14	154984	1886960
147	B	34.19	149424	2036384
148	P	34.75	453584	2489968

149	B	33.48	224768	2714736
150	B	34.75	-42888	2757624

For I pictures

total motion vectorbits = 0
total luminance bits= 9302807
total chrominance bits = 1841605
total misc. overhead bits = 81388

For B pictures

total motion vector bits = 813334
total luminance bits = 15344079
total chrominance bits= 1109422
total misc. overhead bits = 847693

For P pictures

total motion vector bits = 196913
total luminance bits = 13096124
total chrominance bits = 1619312
total misc. overhead bits = 294107

GRAND TOTAL BIT COUNT = 44546784

**** FILE = POPPLE9.stat ****

Index	Type	YSNR(dB)	Frame bits	Cumu. bits
=====				
0	I	39.47	851064	851064
1	P	37.29	432048	1283112
2	B	35.47	213024	1496136
3	B	35.43	213216	1709352
4	P	36.37	383832	2093184
5	B	35.41	221824	2315008
6	B	35.38	217344	2532352
7	P	36.39	402432	2934784
8	B	35.35	221888	3156672
9	B	35.38	220296	3376968
=====				
10	I	37.55	615528	615528
11	B	35.45	209816	825344
12	B	35.02	195200	1020544
13	P	36.40	384352	1404896
14	B	35.41	217384	1622280
15	B	35.34	215896	1838176
16	P	36.30	387248	2225424
17	B	35.38	215736	2441160
18	B	35.34	211816	2652976
19	P	36.25	388752	3041728
20	B	34.84	194552	3236280
21	B	34.76	193392	3429672
=====				
22	I	37.50	624360	624360
23	B	34.90	194976	819336
24	B	34.92	188560	1007896
25	P	36.36	384320	1392216
26	B	34.95	196648	1588864
27	B	34.87	190720	1779584
28	P	36.37	394736	2174320
29	B	34.88	190064	2364384
30	B	34.89	188496	2552880
31	P	37.31	446576	2999456
32	B	36.02	245104	3244560
33	B	35.98	239408	3483968
=====				
34	I	38.44	676032	676032
35	B	36.10	237600	913632
36	B	36.10	240944	1154576
37	P	36.50	382536	1537112
38	B	35.50	211720	1748832
39	B	35.50	1219792	1968624
40	P	36.47	389392	2358016
41	B	35.49	214320	2572336
42	B	35.44	219744	2792080
43	P	36.47	396448	3188528
44	B	35.39	214592	3403120
45	B	35.38	215696	3618816

```

=====
46      I      36.97      '464648      4 6 4 6 4 8
47      B      35.50      204808      -669456
48      B      35.49      203280      872736
49      P      37.41      429880      1302616
50      B      36.14      -232624      1535240
51      B      36.11      234184      1769424
52      P      37.33      436136      2205560
53      B      36.08      236016      2441576'
54      B      36.06      234816      2676392
55      P      36.44      '382776      '3059168
56      B      35.48      213496      3272664
57      B      34.96      193168      3465832
=====
58      I      37.64      '625328      625328
59      B      35.46      204768      830096
60      B      35.53      206128      1036224
61      P      36.46      396304      1432528
62      B      35.46      213152      1645680
63      B      35.43      211728      1857408
64      P      36.48      412264      2269672
65      B      35.37      226424      2496096
66      B      35.37      233832      2729928
67      P      36.02      324608      3054536
68      B      35.33      221632      3276168
69      B      35.36      223312      3499480
=====
70      I      37.69      '609288      609288
71      B      35.45      228680      837968
72      B      35.47      228720      1066688
73      P      36.54      405328      1472016
74      B      35.44      256656      1728672
75      B      35.47      263008      1991680
76      P      36.14      329432      '2321112
77      B      34.72      196816      2517928
78      B      34.74      1'89632      2707560
79      P      36.57      400840      3108400
80      B      35.53      246016      '3354416
81      B      35.50      240664      359508-0
=====
82      I      37.78'      576720      576720
83      B      35.61      232320      809040
84      B      35.68      226888      1035928
85      P      37.44      416880      1452808
86      B      35.87      222616      1675424
87      B      35.89      216.528      1891952
88      P      37.39      424032      2315984
89      B      35.95      211752      2527736
90      B      35.89      216568      2744304
91      P      37.43      430824      3175128
92      B      35.94      222032      33.97160
93      B      35.97      231984      3629144
=====
94      I      37.95      553768      553768
95      B      35.98      2'27128      780896
96      B      35.99      2.35440      1016336
=====

```

97	P	35.77	313984	1330320
98	B	35.37	182688	1 5 1 3 0 0 8
99	B	35.28	189224	1702232
100	P	35.59	3 4 4 4 4 8	2046680
101	B	35.03	204976	22516.56
102	B	34.98	220064	2 4 7 1 7 2 0
103	P	34.76	3 5 6 8 4 8	2 8 2 8 5 6 8
104	B	34.54	218264	3046832
105	B	34.34	2 3 9 1 2 8	3285960
=====				
106	I	35.83	516272	516272
107	B	33.90	2 3 7 4 3 2	753704
108	B	3 3 . 8 0	249328	1003032
109	P	34.01	343768	1 3 4 6 8 0 0
110	B	33.88	266056	1612856
111	B	33.13	252304	1865160
112	P	33.41	3 4 4 7 7 6	2209936
113	B	33.00	256160	2466096
114	B	3 2 . 4 7	-262752	2728848
115	P	33.54	3 5 8 5 9 2	3087440
116	B	32.74	2 6 5 1 7 6	3352616
117	B	32.25	-261488	3614'104
=====				
118	I	34.72	5 0 8 6 5 6	508656
119	B	3 2 . 8 0	265248	773904
-120	B	32.24	241048	1014952
121	P	33.61	320936	1335888
122	B	32.61	249712	1585600
123	B	32.36	258512	1844112
124	P	33.49	326608	2170720
125	B	3 2 . 6 7	239536	2410,256
126	B	32.50	253840	2664496
127	P	33.05	335360	29994'56
128	B	3 2 . 5 1	253984	3253440
129	B	31.94	2 4 0 3 6 8	3493808
=====				
130	I	34.26	553504	553504
131	B	31.85	2 4 1 1 2 8	794632
132	B	-31.54	245400	1040032
133	P	32.75	329992	1 3 7 0 0 2 4
134	B	31.84	261400	1631424
135	B	31.44	255280	1 8 8 6 7 0 4
136	P	32.50	332808	2219512
137	B	31.54	243944	2463456
138	B	31.10	241080	2704536
139	P	32.80	337576	3042112
140	B	31.51	242896	3285008
141	B	31.18	241544	3526552
=====				
142	I	33.78	531648	531648
143	B	31.46	245608	7.77256
144	B	31.48	254544	1031800
145	P	3 2 . 4 4	334688	1366488
146	B	31.40	252680	1619168
147	B	31.00	242032	1861200
148	P	32.12	386784	2247984

149:	B	31.04	217368	2465352
150	B	32.12	10248	2475600

=====

For I pictures
total motion vector bits = 0
total luminance bits = 6050935
total chrominance bits = 1584343
total misc. overhead bits = 71538

For B pictures
total motion vector bits = 3338761
total luminance bits = 137233110
total chrominance bits = 4647090
total misc. overhead bits = 749863

For P pictures
total motion vector bits = 694735
total luminance bits = 10616436
total chrominance bits = 2770002
total misc. overhead bits = 247971

/-

GRAND TOTAL BIT COUNT = 44494984

**** FILE = TENNIS4.stat ****

Index	Type	YSNR(dB)	Frame	bits	Cumu. bits
=====					
0	I	28.39	563568		563568
1	P	27.44	201568		765136
2	B	27.25	38176		803312
3	B	27.26	39368		842680
4	P	26.50	248304		1090984
5	B	26.83	23800		1114784
6	B	26.02	27376		1142160
7	P	25.60	250496		1392656
8	B	25.58	26504		1419160
9	B	25.26	29032,		1448192
=====					
10	I	27.55	507592		507592
11	B	25.18	34824		542416
12	B	25.87	26696		569112
13	P	26.29	276896		846008
14	B	25.95	33472		879480
15	B	25.97	33672		913152
16	P	26.14	24888,0		1162032.
17	B	25.62	28768		1190800
18	B	25.46	29424		1220224
19	P	26.27	237440		1457664
20	B	25.65	29992		1487656
21	B	25.68	30320		1517976
=====					
22	I	29.53	559016		559016
23	B	25.02	43616		602632
24	B	24.34	58848		661480
25	P	27.49	163944		825424
26	B	25.57	45216		870640
27	B	26.44	43336		913976
28	P	29.14	226632		1140608
29	B	27.40	48760		1189368
30	B	27.59	44792		1234160
31	P	30.54	234048		1468208.
32	B	28.61	44376		1512584
33	B	29.06	46128		1558712
=====					
34	I	33.49	447920		447920
35	B	30.18	55344		503264
36	B	30.94	48896		552160
37	P	33.41	240224		792384
38	B	31.72	53176		845560
39	B	31.99	56552		902112
40	P	34.20	250832		1152944
41	B	32.64	59560		1212504
42	B	32.76	56680		1269184
43	P	34.18	234784		1503968
44	B	32.94	53920		1557888
45	B	32.97	58408		1616296

46	I	35.75	386856	386856
47	B	33.29	58568	445424
48	B	33.53	56928	502352
49	P	34.86	245704	748056
				816048
51	50 B B	33.75, 33.58	71544 67992	887592
52	P	34.35	217920	1105512
53	B	33.23	57984	1163496
54	B	33.28	57576	1221072
55	P	34.25	215544	1436616
56	B	33.23	56120	1492736
57	B	33.30	54408	1547144
58	I	34.75	350352	350352
59	B	33.10	47800	398152
60	B	33.36	55200	453352
61	P	34.27	221968	675320
62	B	33.43	53864	729184
63	B	33.36	51128	780312
64	P	34.19	233432	1013744
65	B	33.32	55152	1068896
66	B	33.17	54784	1123680
67	P	30.40	240664	1364344
68	B	30.04	30176	1394520
69	B	30.42	47016	1441536
70	I	34.32	430896	430896
71	B	33.67	97760	528656
72	B	33.73	96568	625224
73	P	35.22	212904	838128
74	B	34.38	81176	919304
75	B	34.37	74112	993416
76	P	35.17	174816	1168232
77	B	34.42	69456	1237688
78	B	34.48	67264	1304952
79	P	35.12	162456	1467408
80	B	34.57	63512	1530920
81	B	34.52	66704	1597624
82	I	37.52	611880	611880
83	B	35.77	81080	692960
84	B	35.80	77008	769968
85	P	36.39	179800	949768
86	B	35.54	71800	1021568
87	B	35.48	72288	1093856
88	P	35.85	144824	1238680
89	B	35.40	69248	1307928
90	B	35.42	67120	1375048
91	P	35.71	141656	1516704
92	B	35.18	57592	1574296
93	B	35.18	58504	1632800
94	I	36.59	559184	559184
95	B	35.53	46368	605552
96	B	35.46	47248	652800

97	P	26.99	179192	831992
98	B	26.86	-59408	891400
99	B	27.39	67208	958608
100	P	28.57	143392	1102000
101	B	28.87	63456	1165456
102	B	28.80	55664	1221120
103	P	31.34	262544	1483664
-104	B	30.44	28800	1512464
105	B	30.59	27216	1539680
=====				
106	I	33.11	490488	490488
107	B	32.29	22312	512800
108	B	32.27	21280	534080
109	P	34.79	257848	791928
110	B	34.00	77776	869704
111	B	33.91	64816	934520
112	P	33.99	197136	1131656
113	B	34.11	62952	1194608
114	B	33.74	65184	1259792
115	P	33.89	200968	1460760
116	B	33.81	66688	1527448
117	B	33.28	40232	1567680
=====				
118	I	36.36	648640	648640
119	B	34.08	-28920	677560
120	B	-33.95	27776	705336
121	P	32.92	167576	872912
122	B	33.00	31104	904016
123	B	32.84	33048	937064
124	P	32.87	188472	11125536
125	B	30.54	44168	1169704
126	B	31.86	33208	1202912
127	P	32.92	191536	1394448
128	B	30.38	54520	1448968
129	B	31.22	45728	1494696
=====				
130	I	34.09	572312	572312
131	B	32.25	41896	614208
132	B	32.22	38264	652472
133	P	33.48	208296	860768
134	B	32.47	-47280	908048
135	B	31.58	65712	973760
136	P	33.58	212640	1186400
137	B	31.42	63056	1249456
138	B	32.25	50080	1299536
139	P	33.32	224928	1524464
140	B	32.47	47008	1571472
141	B	31.45	57392	1628864
=====				
142	I	34.08	584344	584344
143	B	30.84	48784	633128
144	B	31.13	43720	676848
145	P	32.66	190648	867496
146	B	31.16	44656	912152
-147	B	30.55	49944	962096
148	P	31.78	185240	1147336

149	B	30.11	38720	1186056
150	B	31.78	-9424	1195480

=====

For I pictures . . .
total motion vector bits = 0
total luminance bits = -6033847
total chrominance bits = 605255.
total misc. overhead bits = 73946

For B pictures
total motion vector bits = 1179809
total luminance bits = 2878105
total chrominance bits = 277133
total misc. overhead bits = 722433

For P pictures
total motion vector-bits = 279295
total luminance bits = 6824579
total chrominance bits = 640092
total misc. overhead bits = 272186

GRAND TOTAL BIT COUNT = 19786680

**** FILE = TENNIS9.stat ****

Index	Type	YSNR(dB)	Frame bits	Cumu. bits
=====				
0	I	33.87	792256	792256.
1	P	31.97	384.592	1176848
2	B	31.82	-189072	1365920.
3	B	31.82	196736.	1562656
4	P	31.22	-446824	2009480
5	B	31.59	166432	2175912
6	B	31.08	215480	2391392
7	P	30.31	462792	2854184
8	B	29.39	125416	2979600
9	B	29.40	176680	3156280
=====				
10	I	32.24	696272	696272
11	B	29.16	157688	853960
12	B	36.20	173768	1027728
13	P	31.29	502744	1530472
14	B	30.32	216688	1747160
=====				
-15	B	30.53	259904.	2007088
17	B	29.89	148200	2576992
18	B	29.87	175456	2752448
19	P	31.23	434224	3186672
20	B	30.19	177848	3364520
21	B	30.20	-175032	3539552
=====				
22	I	34.20	768504	768504
23	B	29.85	244600	1013104
24	B	28.62	260992	1274096.
25	P	30.52	316984	1591080
26	B	28.19	148984	1740064
27	B	29.81	181424	1921488
28	P	32.76	400784	2322272
29	B	30.73	177752	2500024.
30	B	30.92	180800	2680824
31	P	34.16	418592	3099416
32	B	31.87	170312	3269728
33	B	32.40	171456	3441184
=====				
34	I	36.70	588216	588216
35	B	33.35	181256	769472
=====				
36	B	34.96	223632	1449248
38	B	34.92	207040	1656288
39	B	35.56	225768	1882056
40	P	37.33	445440	2327496
41	B	35.87	211768	2539264
42	B	36.13	245272	2784536
43	P	37.64	460512	3245048
44	B	36.41	245368	3490416
45	B	36.48	250128	3340544

```

=====
46,47 B I 38.49 592232 592232
48 B 35.89 166816 759048
49 P 36.27 178008 937056
P 37.4136 -1352592
36.55

50 B 3 B 6 21879253 22745013841798784
52 P 37.46 418824 2217.608.
53 B 2 364 .162 209696 3 0 4
54 B 36.28 211912 2639216
455 P 37.47 3056040
56 B 36.26 210672 3266712
57 B 36.70 227760 34944.72
=====
59 I B 37.85 585064 585064
B 36.27 209064 794128

60 P 37.0434 216416 404160 14114544
62 B 36.40 216376 1631080
63 B 36.38 214776 1845856
64 P 37.02 418848 2264704
65 B 36.29 225400 2490104
66 B 36.00 175608 2665712
67 P 34.13 406008 3071720
68 B 33.86 1.29096 3200816
69 B 34.56 154680 3355496
=====
70 I 38.20 696592 696592
71 B 37.20 228696 925288
72 B -37.23 229776 1155064
73 P 37.80 403480 1558544
74 B -36.71 1.79536 1738080
75 B 36.66 176208 1914288
2368912 P 38.89 454624

77 B 36.85 181152 2550064
78 B 36.93 175040 2725104
79 P 39.17 466032 3191136
80 B 37.43 1.89752 3380888
81 B 37.39 1.92720 3573608
=====
82 I 40.12 921288 921288
83 B 37.69 -188448 1109736
84 B 37.67 183544 1293280
85 P -39.11 440656 1733936
86 B 37.47 192488 1926424
87 B 37.46 1.95472 2121896
88 P 39.08 428504 2550400
89 I 3 37.48 188184 2738584
90 B 37.53 184280 2922864
91 P 39.06 424680 3347544
92 B 37.48 1.86936 3534480
93 B 37.50 3.87072 3721552
=====
94 I 40.06 928384 928384
95 B 37.75 174624 1103008
96 B 37.74 1.79736 1282744
=====

```

97	P	32.60		1672600
99	B	31.14	389856	1750016
	B	32.92,	774168,	1885864
100	P	35.61	-389008	2274872
101	B	34.88,	163520	2438392
102	B	34.79	154600	2592992
103	P	37.04	448640	3041632
104	B	36.47	-222720,	3264352
105	B	36.34	222656	3487008
=====				
106	I	38.28	801424	801424
107	B	36.57,	123320,	924744
108	B	36.54	123896	1048640
109	P	38.18	429504	1478144
110	B	37.42	201904	1680048
111	B	37.35	203136	1883184
112	P	38.15	428096	2311280
113	B	37.44	-201328	2512608
114	B	37.26	209432	2722040
115	P	37.25	366608	3088648
116	B	37.21	211624	3300272
117	B	37: 09	225640	3525912
=====				
118	I	38.55	831360	831360
119	B	37.11	197152	1028512
120	B	37.02	195624	1224136
121	P	37.23	391608	1615744
122	B	35.63	145248	1760992
123	B	35.78	141320	1902312
124	P	36.36	370296	2272608
125	B	35.83	232096	2504704
126	B	35.47	150608	2655312
127	P	36.39	382920	3038232
128	B	35.21	210416	3248648
129	B	35.25	176808	3425456
=====				
130	I	36.78	775016	775016
131	B	35.61	151536	926552
132	B	35.37	153456	1080008
133	P	36.36	373912	1453920
134	B	35.37	157168	1611088
135	B	35.28	202656	1813744
136	P	36.37	375432	2189176
137	B	35.66	25648.0	2445656
138	B	35.68	225448	2671104
139	P	36.24	386880	3057984
140	B	36.37	24.4520	3302504
141	B	35.24	181344	3483848
=====				
142	I	37.65	839328	839328
143	B	35.34	191824	1031152
144	B	34.73	161272	1192424
145	P	36.32	375752	1568176-
146	B	34.69	162512	1730688
147	B	34.59	180784	1911472
148	P	36.27	393776	2305248

149	B	34.65	180024	2485272
150	B	36.28	9328	2494600

=====

For I pictures

'total motion vector'bits = 0
total luminance bits= 8941863
'total chrominance bits = 801939
'total'misc. overhead bits = 72134'

For B pictures

total motion vector bits = 1178389
total luminance bits = 15870203
total chrominance bits = 971806
total misc. overhead bits = 805890

For P pictures

total motion vector bits = 282098
total luminance bits = 14280388.
total chrominance bits = 968654
total misc. overhead bits = 266148

GRAND TOTAL BIT COUNT = 44439512

CERTIFICATE OF SERVICE

I hereby certify that a copy of **RESPONDENTS' RESPONSIVE CLAIM CONSTRUCTION BRIEF** has been served on February 23, 2024, as indicated, on the following:

The Honorable Lisa R. Barton Secretary to the Commission U.S. International Trade Commission 500 E. Street, S.W., Room 112A Washington, DC 20436	☐ First Class Mail ☐ Overnight Delivery ☒ EDIS ☐ Email
Honorable Doris Johnson Hines Administrative Law Judge U.S. International Trade Commission 500 E Street, SW Washington, DC 20436 Pyong Yoon, Esq. (Attorney Advisor)	☐ First Class Mail ☒ Overnight Delivery ☒ EDIS ☒ Email ☒ Upload to Box account <i>JohnsonHines1379@usitc.gov</i> <i>Pyong.Yoon@usitc.gov</i>
Marissa Ducca, Esq. Jeffrey Hsu, Esq. Office of Unfair Import Investigations U.S. International Trade Commission 500 E. Street, S.W., Suite 401 Washington, DC 20436 Phone: (202) 205-2044	☐ First Class Mail ☐ Overnight Delivery ☒ Email ☒ Upload to Box account <i>Marissa.Ducca@usitc.gov</i> <i>Jeffrey.Hsu@usitc.gov</i>

Adam D. Swain

Alston & Bird LLP
950 F Street NW
Washington, DC 20004
Telephone: (202) 239-3300

John D. Haynes
Nicholas T. Tsui
Lindsay C. Church
Shawn P. Gannon
Paul Michael Haley
Alston & Bird LLP
1201 West Peachtree Street
Atlanta, GA 30309
Telephone: (404) 881-7000

Theodore Stevenson
Alston & Bird LLP
2200 Ross Avenue, Suite 2300
Dallas, TX 75201
Phone: (214) 922-3400

Caleb J. Bean
Alston & Bird LLP
333 South Hope Street
16th Floor
Los Angeles, CA 90071
Phone: 213-576-1000

August R. Meny
Alston & Bird LLP
90 Park Avenue
New York, NY 10016-1387
Phone: 212-210-9400

Warren Lipschitz
Alexandra F. Easley
MCKOOL SMITH, P.C.
300 Crescent Court Suite 1200
Dallas, TX 75201
Telephone: (214) 978-4000

R. Mitch Verboncoeur
MCKOOL SMITH, P.C.
303 Colorado Street Suite 2100
Austin, TX 78701
Telephone: (512) 692-8700

- ☐ First Class Mail
☐ Overnight Delivery
☒ Email

NKAmazonITC@alston.com

Nokia_HP_Amazon@mckoolsmith.com

Joshua J. Newcomer
MCKOOL SMITH, P.C.
600 Travis Street, Suite 7000
Houston, TX 77002
Telephone: (713) 485-7316

Kevin Burgess
MCKOOL SMITH P.C.
104 East Houston St., Suite 300
Marshall, Texas 75670
Telephone: (903) 923-9000

*Counsel for Complainants Nokia Technologies
Oy and Nokia Corporation*

Eric S. Namrow

Stephanie L. Roberts

Kandis C. Gibson

Manu Bansal, Ph.D.

Kevin J. Spinella

Eunjean Je

Morgan, Lewis & Bockius LLP

1111 Pennsylvania Ave, N.W.

Washington, D.C. 20004-2541

Telephone: (202) 739-3000

David J. Levy

Rick L. Rambo

Thomas R. Davis

Morgan, Lewis & Bockius LLP

1000 Louisiana Street, Suite 4000

Houston, TX 77002-5005

Telephone: (713) 890-5000

Jason E. Gettleman

Corey R. Houmand

Morgan, Lewis & Bockius LLP

1400 Page Mill Road

Palo Alto, CA 94304

Telephone: (650) 843-4000

Kevin J. Post

Lance W. Shapiro

Ropes & Gray LLP

1211 Avenue of the Americas

New York, NY 10036

Telephone (212) 596-9000

Matthew J. Rizzolo

Ropes & Gray LLP

2099 Pennsylvania Ave, NW

Washington, DC 20006

Telephone: (202) 508-4600

Larry L. Shatzer

Mark DeBoy

EDELL, SHAPIRO & FINNAN

9801 Washingtonian Blvd., Suite 750

Gaithersburg, MD 20878

Counsel for Respondent HP Inc.

☐ First Class Mail

☐ Overnight Delivery

☒ Email

HP-Nokia-ITC@morganlewis.com

HP-Nokia-Ropes1379@ropesgray.com

ESF_ITC@esfip.com

/s/ Nelo Keith Lang

Nelo Keith Lang

Paralegal

Perkins Coie LLP

700 13th Street, NW., Suite 800

Washington, DC 20005-3960

Email: nkeithlang@perkinscoie.com