

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and AMAZON.COM SERVICES LLC,
Petitioner

v.

NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2024-00725

U.S. Patent No. 8,204,134

DECLARATION OF JOSEPH P. HAVLICEK, PH.D.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: 5 April 2024

By: 

Joseph P. Havlicek, Ph.D.

TABLE OF CONTENTS

I.	BACKGROUND AND QUALIFICATIONS	1
A.	Educational Background	1
B.	Professional Experience	2
C.	Publications	6
II.	MATERIALS CONSIDERED	7
III.	LEGAL STANDARDS	7
A.	Priority Date	8
B.	Claim Interpretation	9
C.	Prior Art.....	9
D.	Obviousness.....	9
IV.	TECHNOLOGY BACKGROUND.....	13
V.	'134 PATENT.....	18
A.	Specification.....	18
B.	Claims.....	20
C.	Prosecution History	20
VI.	LEVEL OF ORDINARY SKILL	21
VII.	CLAIM CONSTRUCTION	21
VIII.	OVERVIEW OF THE CITED REFERENCES	22
1.	Yagasaki (U.S. Patent No. 5,835,672).....	22
2.	Oliver (U.S. Patent No. 6,064,693).....	36
3.	Lyon (International Publication No. WO 91/06182)	39
4.	Ran (U.S. Patent No. 5,768,533)	39
IX.	DETAILED EXPLANATION OF GROUNDS.....	41
A.	Ground 1: Yagasaki Renders Obvious Claims 1-5, 9-12, and 16-19.....	41

1.	Claims 1-5	41
2.	Claims 9-12	69
3.	Claims 16-19	73
B.	Ground 2: Yagasaki and Oliver Render Obvious	
	Dependent Claims 5, 7, 13, 15, 20, and 22	80
1.	Claim 5: “A method according to claim 1, further comprising entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”	80
2.	Claim 7: “A method according to claim 1, wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.”	87
3.	Claim 13: “A video decoder according to claim 9, wherein the video decoder is further configured to enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”	89
4.	Claim 15: “A video decoder according to claim 9, wherein the video decoder is further configured to configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.”	90
5.	Claim 20: “A computer program product according to claim 16, the computer program product further comprising: a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”	90
6.	Claim 22: “A computer program product according to claim 16, the computer program product further comprising: a computer program code for configuring the	

	buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.”	91
C.	Ground 3: Yagasaki, Oliver, and Lyon Render Obvious Dependent Claims 6, 14, and 21	92
1.	Claim 6: “A method according to claim 5, wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.”	92
2.	Claim 14: “A video decoder according to claim 13, wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.”	96
3.	Claim 21: “A computer program product according to claim 20, the computer program product further comprising: a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.” ..	97
D.	Ground 4: Yagasaki and Ran Render Obvious Dependent Claim 8	97
1.	Claim 8: “A method according to claim 2, further comprising detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.”	97
APPENDIX A: MATERIALS CONSIDERED		103
APPENDIX B: LISTING OF CHALLENGED CLAIMS		106

I, Dr. Joseph P. Havlicek, do hereby declare:

I. BACKGROUND AND QUALIFICATIONS

1. I have been retained by Petitioner Amazon.com, Inc. and Amazon.com Services LLC (“Amazon” or “Petitioner”) to provide my opinions regarding claims 1-22 of U.S. Patent No. 8,204,134 (“the ’134 patent,” EX-1001).

2. I am currently the Gerald Tuma Presidential Professor of Electrical and Computer Engineering at the University of Oklahoma.

3. Attached as EX-1004 is a copy of my current curriculum vitae, which provides further details about my academic and professional background and qualifications.

4. I am being compensated for my services and reimbursed for reasonable expenses that I may incur while working on this matter. I have no financial interest in the outcome of this matter or any other matter that may exist between Petitioner and Nokia Technologies Oy (“Nokia” or “Patent Owner”). This compensation does not depend in any way on the conclusions I reach.

A. Educational Background

5. 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

6. 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.

7. 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.

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

9. Since January 1997, I have been a regular faculty member in the School of Electrical and Computer Engineering at the University of Oklahoma, Norman, Oklahoma. 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 from 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.

10. 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).

11. For over 35 years, I have conducted research and taught classes in the field of image processing and image 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 11 Ph.D. students to completion and I am currently supervising two Ph.D. students. 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 67 additional doctoral committees and 68 additional M.S. committees.

12. 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).

13. 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.

C. Publications

14. A complete listing of my publications is found in my curriculum vitae, attached as EX-1004. I highlight some of the publications relevant to the subject matter of the '134 patent below. I have published numerous peer-reviewed scholarly journal articles, book chapters, and conference papers, including 24 journal articles, 9 book chapters, and 100 conference papers; the following are representative:

- J.P. Havlicek, T.N. Arian, H. Soltani, T. Przebinda, and M. Özaydın, “A preliminary case for Hirschman transform video coding,” in Proc. IEEE Southwest Symp. Image Anal. & Interp., Santa Fe, NM, Mar. 29-31, 2020, pp. 104-107.
- E. Vorakitolan, J.P. Havlicek, R.D. Barnes, and A.R. Stevenson, “Simple, Effective Rate Control for Video Distribution in Heterogeneous Intelligent Transportation System Networks,” in Proc. IEEE Southwest Symp. Image Anal. & Interp., San Diego, CA, Apr. 6-8, 2014, pp. 37-40.

- V. DeBrunner, J.P. Havlicek, T. Przebinda, and M. Özaydın, “Entropy-Based Uncertainty Measures for $L_2(\mathbb{R}^n)$, $l_2(\mathbb{Z})$, and $l_2(\mathbb{Z}/N\mathbb{Z})$ with a Hirschman Optimal Transform for $l_2(\mathbb{Z}/N\mathbb{Z})$,” IEEE Trans. Signal Process., Aug. 2005, vol. 53, no. 8, pp. 2690-2699.
- O. Alkhoul, V. DeBrunner, and J. Havlicek, “Hirschman Optimal Transform (HOT) DFT Block LMS Algorithm,” in Adaptive Filtering, L. Garcia, ed., ISBN: 978-953-307-158-9, InTech, Sep. 2011, pp. 135-152.
- V. DeBrunner, M. Özaydın, T. Przebinda, and J. Havlicek, “The optimal solutions to the continuous- and discrete-time versions of the Hirschman uncertainty principle,” in Proc. IEEE Int’l. Conf. Acoust., Speech, Signal Process., Istanbul, Turkey, Jun. 5-9, 2000, vol. 1, pp. 81-84.

II. MATERIALS CONSIDERED

15. In developing my opinions below relating to the ’134 patent, I have considered the materials cited herein, including those itemized in the table of “Materials Considered” attached as Appendix A to this Declaration.

III. LEGAL STANDARDS

16. I am not an attorney and offer no legal opinions, but in the course of my work, I have had experience studying and analyzing patents and patent claims from the perspective of a POSITA.

17. For the purposes of this Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Priority Date

18. I understand that, to gain benefit of the filing date of an earlier patent application, each application in the chain leading back to the earlier application must comply with the statutory written description requirement.

19. I understand that, to satisfy the written description requirement, a patent's specification must reasonably convey to a POSITA that the inventor had possession of the claimed subject matter when the patent application was filed. I understand that, to show "possession" of the invention, the disclosure must describe the invention, including all of its claimed limitations. It is not enough that a claimed invention is an obvious variant of that which is disclosed in the specification.

20. I further understand that entitlement to a filing date extends only to subject matter that is disclosed and not to subject matter that is not disclosed but would be obvious over what is expressly disclosed. Thus, a prior application itself must describe an invention, and do so in sufficient detail that a POSITA can clearly conclude that the inventor invented the claimed invention as of the filing date sought.

21. I also understand that a patent's priority date is a legal determination based on underlying factual issues. For that reason, I am not rendering a legal opinion on the ultimate legal question of priority date. Rather, my testimony

addresses the underlying facts and factual analysis that would support a legal determination of the priority date.

B. Claim Interpretation

22. I understand from Counsel that during an IPR proceeding, claims are to be construed in light of the specification as would be read by a POSITA at the time the application was filed. I understand that claim terms are given their ordinary and customary meaning as would be understood by a POSITA in the context of the entire disclosure. A claim term, however, will not receive its ordinary meaning if the patentee acted as his or her own lexicographer and clearly set forth a definition of the claim term in the specification. In that case, the claim term will receive the definition set forth in the patent.

C. Prior Art

23. I understand from Counsel that certain patents and printed publications can be considered prior art in an IPR proceeding. I understand that prior art must predate the effective filing date of the claims of the patent-at-issue.

D. Obviousness

24. I understand that a claim is unpatentable if the claim limitations, as arranged in a claim, would have been obvious to a person of ordinary skill in the art as of the date of invention. For example, a combination of familiar elements

according to known methods is likely to be obvious if the combination does no more than produce predictable results.

25. In determining obviousness, I understand that the analysis may take into account multiple prior art references, as well as the common knowledge possessed by a POSITA (as discussed above). I further understand that the POSITA is not an automaton but one having common sense and ordinary creativity. I also understand that a claim is not obvious simply because a combination of prior art references discloses each claim element collectively.

26. In determining obviousness, I understand that one should consider the following.

- The scope and content of the prior art;
- The differences between the claimed invention and the prior art;
- The level of ordinary skill in the pertinent art; and
- Secondary considerations of obviousness.

27. I understand that a POSITA is assumed to have knowledge of all prior art references. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists so as to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be rigidly applied, but that the test can be important to avoiding such hindsight.

28. I understand that a POSITA can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine the references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a POSITA would employ to combine the known elements in the prior art in the manner claimed by the patent-at-issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. This does not mean, however, that a POSITA, for purposes of the obviousness inquiry, does not have recourse to common sense.

29. I understand that the following are examples of approaches and rationales that may be considered in determining whether a prior art reference could have been combined with other prior art or with other information within the knowledge of a POSITA:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of known techniques to improve similar devices, methods, or products in the same way;

- Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;
- That it was “obvious to try”—i.e., that it was one choice from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art; and
- Some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art reference or to combine multiple prior art reference teachings to arrive at the claimed invention.

30. I also understand that obviousness is a legal conclusion based on the underlying factual issues of the scope and content of the prior art, the differences between the claimed invention and the prior art, the level of ordinary skill in the art, and any objective indicia of non-obviousness. For that reason, I am not rendering a legal opinion on the ultimate legal question of obviousness. Rather, my testimony addresses the underlying facts and factual analysis that would support a legal conclusion of obviousness, and when I use the term obvious, I am referring to the perspective of a POSITA at the time of invention.

IV. TECHNOLOGY BACKGROUND

31. The '134 patent is generally concerned with decoding video sequences from which image frames have been intentionally removed by the encoder.¹ EX-1001, Abstract, 4:21-24. In particular, the '134 patent describes conducting memory management in the decoder to maintain synchronicity with the encoder. *See* EX-1001, Abstract, 4:8-17.

¹ The challenged claims of the '134 patent use the term “image frames,” so I follow that convention here. The '134 patent uses the terms “image frame,” “frame,” “video frame,” and “picture” interchangeably. *See* EX-1001, 4:8-17 (“A similar problem is encountered when buffering and indexing reference pictures in the encoder and the decoder. As a part of the indexing, the encoder may intentionally remove a plurality of image frames, which are unnecessary as reference image frames for the temporal prediction process of any remaining image frames. ”), 8:24-27 (“FIG. 2 further shows a next I-frame 202 of the INTRA layer, the frame thus being also an independently determined video frame that has been determined without using motion-compensated temporal prediction.”), 10:24-28 (“If there are temporally overlapping image frames on the same layer, the frames are interpreted as alternative presentations of the same image content and therefore any mode of image presentation can be used.”).

32. Digital video, like analog video (and film) comprises a rapid, time-ordered sequence of image frames. Sullivan, 74. Digital video in its raw format is highly redundant. Each video frame comprises a regular, rectangular array of pixels. Transmission or storage of raw digital video data requires huge volumes of data and very fast data rates. Sullivan, 74-76. Compression and decompression refer to techniques to take one form of data and convert it into another form that takes up less space, i.e., a fewer number of bits, in a memory. Compression is the process of taking data and making it occupy less space in memory, and decompression is the process of restoring the data to its original form. An encoder compresses data, and a decoder reverses the process for display. Riley, 15, 18-19 (“[D]igital video information must be compressed (encoded) prior to transmission and decompressed (decoded) before displaying it at the receiver.”).

33. Image frames in a video sequence may depend on information from other frames. This dependency arises from the use of a compression technique called “temporal prediction,” which takes advantage of “temporal redundancy” across image frames. “[T]emporal redundancy refers to the changes taking place in specific image objects in subsequent image frames.” EX-1001, 2:1-3. Because image frames in a video sequence are typically captured in quick succession, many aspects of the scene being captured do not change, or change in predictable ways, from one image frame to the next. These aspects are considered to be temporally redundant.

34. Temporal prediction leverages temporal redundancies across image frames by encoding only the changes that occur between frames. EX-1001, 1:65-4:17; Wiegand, 1:46-2:30; Sullivan, 76; Aravind, 82-86. Temporal prediction allows image frames to be encoded using less data because redundant information is excluded. Instead, the encoded information contains some information about the depicted scene and refers to another image frame for any missing information.

35. A specific type of temporal prediction discussed in the '134 patent is “motion-compensated temporal prediction,” sometimes called simply “motion-compensated prediction” or “motion compensation.” EX-1001, 2:18-24, 20:12-16. Motion-compensated prediction works by predicting the location of an object in a given image frame based on its location in other image frames and information about its movement. EX-1001, 2:16-23 (describing “motion-compensated temporal prediction” as follows: “the contents of some (typically most) of the image frames in a video sequence are predicted from other frames in the sequence by tracking changes in specific objects or areas in successive image frames”). The widely used video coding standards developed by the Moving Picture Experts Group (MPEG) utilize an adaptive combination of motion-compensated prediction and transform coding of the remaining spatial information to achieve high data compression. Aravind, 82-88; Riley, 38-47; Sullivan, 75-77.

36. When an image frame is encoded using motion-compensated temporal prediction, additional information outside of the encoded image frame is needed to decode it, namely, the “reference frame” on which it depends. *See* EX-1001, 2:31-33 (“If a frame is lost, frames dependent on it can no longer be correctly decoded.”). For each block in a current frame, a search is made to find a best match to the block in a reference frame that has already been coded. Aravind, 76-78; Sullivan, 78. The best-matched block is used as a prediction for the current block. The difference between the pixel values of the current block and the pixel values of the prediction block is called the residual data. The location of the best match block in the reference frame relative to the location of the current block is called a motion vector. The residual data block and the motion vector are coded, and typically comprise a small fraction of the amount of data needed to code the current block alone.

37. Image frames are given “types” based on their dependence on other image frames. *See, e.g.*, Riley, 40; EX-1001, 3:50-52 (“Further, the reference picture selection can be generalized to cover all temporally predicted frame types, including B-frames.”). “Intra-frames” or “I-frames” are coded independently and do not depend on information from other image frames. EX-1001, 2:23-29; Riley, 40. “Inter-frames,” also called “predicted frames” or “P-frames,” depend on information from previous I- or P-frames. EX-1001, 2:27-31; Riley, 40. “Bi-directionally

predicted frames,” or “B-frames,” depend on both a preceding and succeeding reference frame. EX-1001, 2:43-48; Riley, 28-30, 40.

38. Motion prediction requires both the encoder and decoder to have access to reference frames, which are stored in memory. Riley, 28-29. By the earliest priority date of the '134 patent, it was further known that multiple reference pictures could be stored in buffer memory. *See* Wiegand, 8:12-26; Long-Term, 3; Fast-Search, 619-620; Girod, 5:37-61; Yagasaki, 14:54-15:67, Figures 14-16.

39. Synchronizing these buffers is crucial to ensure the accurate reconstruction of motion between frames. *See* Riley, 28; Wiegand, 8:12-26; Long-Term, 3; Fast-Search, 619-620; Girod, 1:60-63. The main reason for this is that, at the decoder, each block of an image frame that was encoded using motion compensated temporal prediction must be reconstructed by adding the residual data block that is decoded from the encoded bitstream to the motion-compensated prediction block that was used to predict the block *at the encoder*. Thus, it is critical for the prediction blocks obtained at the decoder to match precisely the ones that were used for temporal prediction at the encoder – otherwise the decoded image frames obtained at the decoder will not match the ones that were used for encoding the bitstream at the encoder. Since these decoded image frames will generally be used themselves as temporal predictions for subsequent temporally predicted image frames, any disagreement, mismatch, or loss of synchronization between the

reference frames stored in buffer memory at the encoder and decoder will tend to cause rapid divergence of the decoded image frames obtained at the decoder, resulting in visually significant degradation of the decoded video sequence. For at least this reason, it was known before the earliest priority date of the '134 patent to notify the decoder when the encoder deliberately excludes a frame from the encoded signal. *See* Casavant, Abstract, 1:8-13, 1:56-67, 2:13-35, 2:48-50, 3:33-54, 4:1-33, 6:60-7:38, Figures 3, 6; Yagasaki, Abstract, 3:25-35, 3:54-67, 4:8-15, 4:30-41, 4:61-5:16, 5:38-65, 11:1-6, 11:37-40, 15:12-67, 18:25-32, Figures 15-16, claims 1-2, 9-12.

V. '134 PATENT

40. I have reviewed the '134 patent and provide the following summary.

A. Specification

41. The '134 patent is titled "Grouping of Image Frames in Video Coding" and relates to buffer memory management in a video decoder. EX-1001, Abstract, 4:8-17, 20:22-27. According to the '134 patent, video "encoder[s] may intentionally remove a plurality of image frames" from a compressed video sequence, for example, if the image frames "are unnecessary as reference image frames for the temporal prediction process of any remaining image frames." EX-1001, 4:9-13. When the encoder does so without "[providing] the decoder a reason for the removal," however, "the [decoder] may unnecessarily interpret these intentional

removals as protocol errors,” triggering “unnecessary error correction actions.” EX-1001, 4:13-17, 4:65-5:4.

42. The '134 patent addresses this problem by including an “indication” in the compressed video sequence that “enable[s] the decoder to take into account...image frames the encoder has intentionally removed” from the video sequence. EX-1001, 4:21-33. The '134 patent further explains that “[t]he missing image frames cause discontinuities in the numbering of the image frames.” EX-1001, 4:33-47. The decoder decodes the indication from the compressed video sequence during the decoding process. EX-1001, 4:31-36.

43. As part of the decoding process, the decoder enters image frames into a buffer memory. EX-1001, 4:31-33; *see also id.*, 4:8-13, 5:1-4, 19:54-20:56. “[I]n response to the indication, the [decoder’s] buffer memory is configured to provide a number of image frames corresponding to [the] discontinuity in the numbering of the image frames.” EX-1001, 4:36-39. Finally, “the image frames in the buffer memory are used in the decoding process.” EX-1001, 4:39-40.

44. In sum, the '134 patent describes a memory management technique that performs synchronization of the decoder’s buffer memory when video frames are intentionally removed by the encoder. EX-1001, Abstract, 4:8-17, 20:22-27. As I discuss above in ¶39 of the Technology Background, such techniques were known before the '134 patent’s earliest priority date. Accordingly, the Challenged Claims

were obvious in view of the prior art before the '134 patent's earliest priority date, as I explain in Section IX below.

B. Claims

45. A labelled list of the Challenged Claims is attached as Appendix B to this Declaration.

C. Prosecution History

46. I have reviewed the '134 patent's prosecution history and provide the following summary.

47. The Examiner issued multiple § 102 and § 103 rejections based on U.S. Patent No. 7,103,669 ("Apostolopoulos"; EX-1009). EX-1002, 1373-1377, 1830-1842, 2127-2140, 2442-2451. In response, the applicant made numerous amendments and cancellations. For example, the applicant amended the claims to recite "an intentional discontinuity of numbering of the image frames." EX-1002, 2094-2095, 2097 (added language indicated by underline in original). In its accompanying argument, the applicant admitted that Apostolopoulos discloses "discontinuities of numbering[.]" EX-1002, 2100; *see also id.*, 2435 (repeating admission).

48. The Examiner eventually allowed the claims based on applicant's arguments and amendments. EX-1002, 2489. None of the prior art references cited

in this Declaration was before the Examiner during prosecution of the '134 patent or during prosecution of the application that resulted in U.S. Patent No. 7,894,521.

VI. LEVEL OF ORDINARY SKILL

49. I understand that when both interpreting the claims and assessing the prior art, I must do so from the perspective of a POSITA as of the relevant priority date. My understanding is that the earliest possible priority date of the '134 patent is January 23, 2002.

50. In my opinion, as of the earliest possible priority date, a POSITA would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa.

51. I was at least a POSITA as of the earliest possible priority date of the '134 patent, and still am.

VII. CLAIM CONSTRUCTION

52. I have given all the claim terms of the Challenged Claims their plain and ordinary meaning, as would have been understood by a POSITA, at the time of the alleged invention.

VIII. OVERVIEW OF THE CITED REFERENCES

53. For my analysis, I have been asked to assume that the following references are prior art.

1. Yagasaki (U.S. Patent No. 5,835,672)

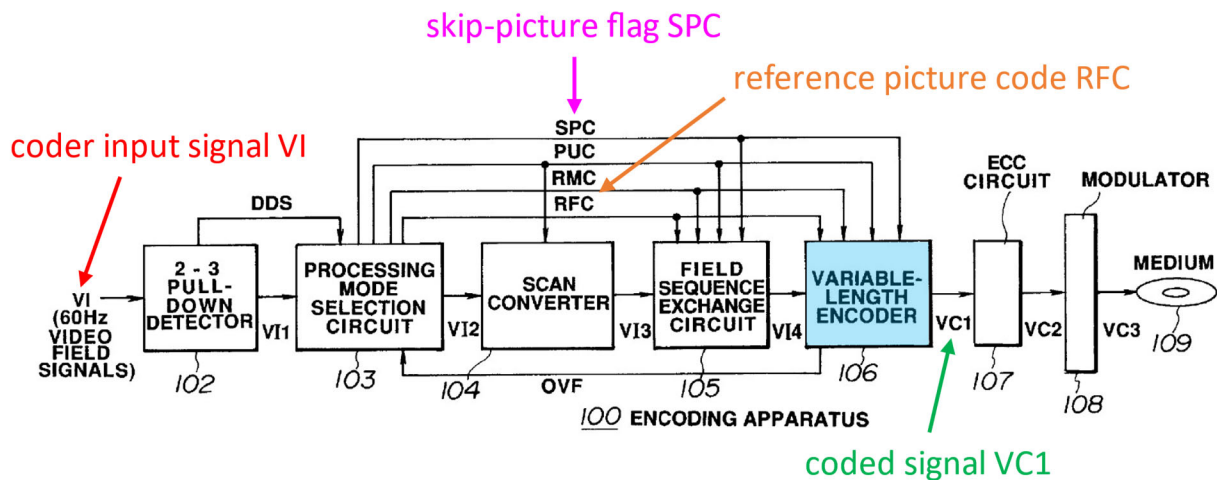
54. Yagasaki relates to coding and decoding of video signals. Yagasaki, Abstract, 1:8-12, 5:38-65. Specifically, Yagasaki teaches including a “skip-picture flag” in a coded video signal to indicate a picture that has been intentionally eliminated from the signal by the encoder. Yagasaki, Abstract, 3:25-35, 3:54-67, 4:8-15, 4:30-41, 4:61-5:16, 5:38-65, 11:1-6, 11:37-40, 15:12-67, 18:25-32, Figures 15-16, claims 1-2, 9-12. The decoder replaces the eliminated picture with a picture retrieved from buffer memory, which Yagasaki refers to as field memories 1207 through 1210. Yagasaki, 14:54-15:67, Figures 14-16.

a. First embodiment

(i) Coding apparatus 100

55. Figure 3 of Yagasaki, shown below with annotations, illustrates coding apparatus 100. Yagasaki, 6:66-7:2. Coding apparatus 100 includes...processing mode selecting circuit 103 which generates...control signals and flags for each picture in the coder input signal VI [annotated **red**.]” Yagasaki, 7:11-34; *see also id.*, 8:46-9:4, 10:56-11:10, Figures 6-8. I note that coding apparatus 100 is mislabeled coding apparatus 101 at 7:11 of Yagasaki; 101 is the numeral assigned

to the decoding apparatus of Figure 3, as noted in Yagasaki at 6:66-67 and 7:66. The “control signals and flags” include “skip picture flag SPC [annotated pink]” and “reference picture code RFC [annotated orange]” that indicates, when the picture is a skip picture, the picture...that will be used as the reference picture for reproducing the skip picture.” Yagasaki, 7:17-18, 7:30-33; *see also id.*, 8:50-53, 8:65-67, 11:11-12:36, Figures 7-8.

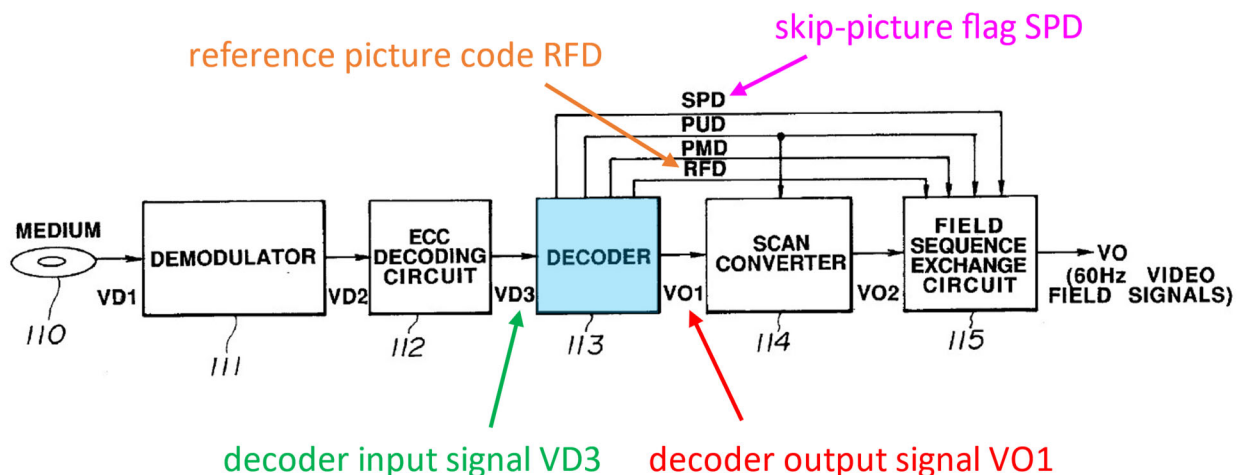


Yagasaki, top Figure 3 (annotated).

56. In response to the control signals and flags, encoder 106 (shaded blue) generates coded signal VC1 (annotated green) by encoding input signal VI4 together with the control signals and flags. Yagasaki, 7:56-60, 9:21-26, 11:37-40. Coded signal VC1 has “error correction codes” appended and “is modulated to provide the recording signal VC3[.]” Yagasaki, 7:56-65.

(ii) **Decoding apparatus 101**

57. The bottom half of Figure 3, shown below with annotations, illustrates decoding apparatus 101. Yagasaki, 6:66-7:2 The input provided to decoding apparatus 101 is signal VD1, which is the same as recording signal VC3. Yagasaki, 7:66-8:7. As I discuss above in ¶32, the decoder reverses the encoding process to decode the compressed video signal, meaning that the output of the encoding process is the input to the decoding process. Decoder input signal VD3 (annotated **green**) is fed to decoder 113 (shaded **blue**) after demodulation and error decoding have been applied. Yagasaki, 7:66-8:7. Decoder input signal VD3 is the same as coded signal VC1, also for the reasons explained in ¶32.

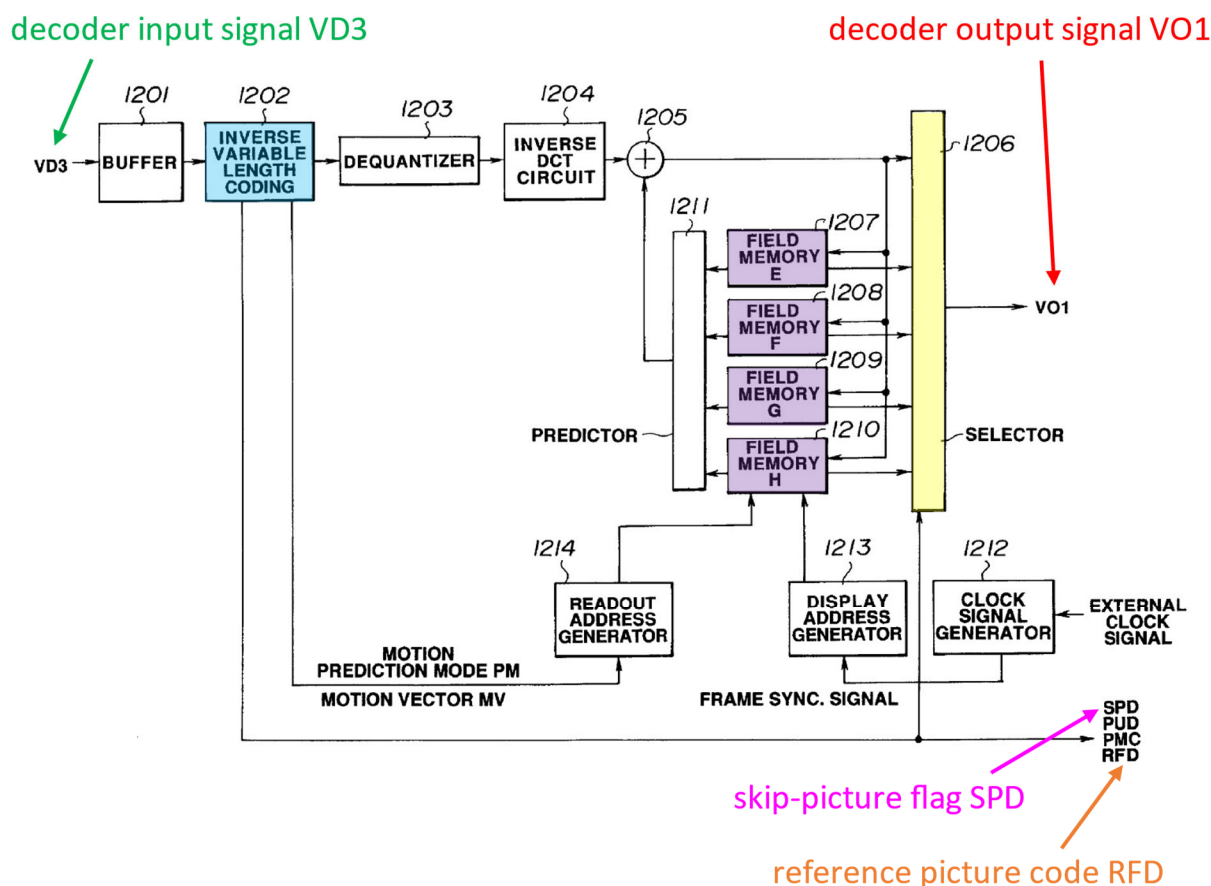


Yagasaki, bottom Figure 3 (annotated).

58. Decoder 113 extracts from decoder input signal VD3 “flags and codes” generated during the encoding process—which are the “control signals and flags” I discuss above in Section VIII.1.a(i), including skip picture flag SPD [annotated

pink] and reference picture code RFD [annotated orange]—and “decodes the coded picture signals in the decoder input signal VD3 in response to these flags and codes to provide the decoder output signal VO1 [annotated red].” Yagasaki, 8:7-17. I note that Yagasaki uses a naming convention ending with “C” for flags/codes in the encoder and a naming convention ending with “D” for flags/codes in the decoder. Flags/codes with similar names between the encoder and decoder correspond to one another.

59. Figure 14 of Yagasaki, shown below with annotations, provides a detailed view of decoder 113. Yagasaki, 14:36-37. “The signal for each picture in...decoder input signal [VD3] [annotated green]...is fed to the variable-length decoder 1202 [shaded blue].” Yagasaki, 14:37-41. Variable-length decoder 1202 “reverses the variable length coding applied...in the encoder [106 during the encoding process] and extracts various control signals and flags,” including “reference picture code RFD [annotated orange] and...skip picture flag SPD [annotated pink].” Yagasaki, 14:39-44; *see also id.*, 15:7-8.

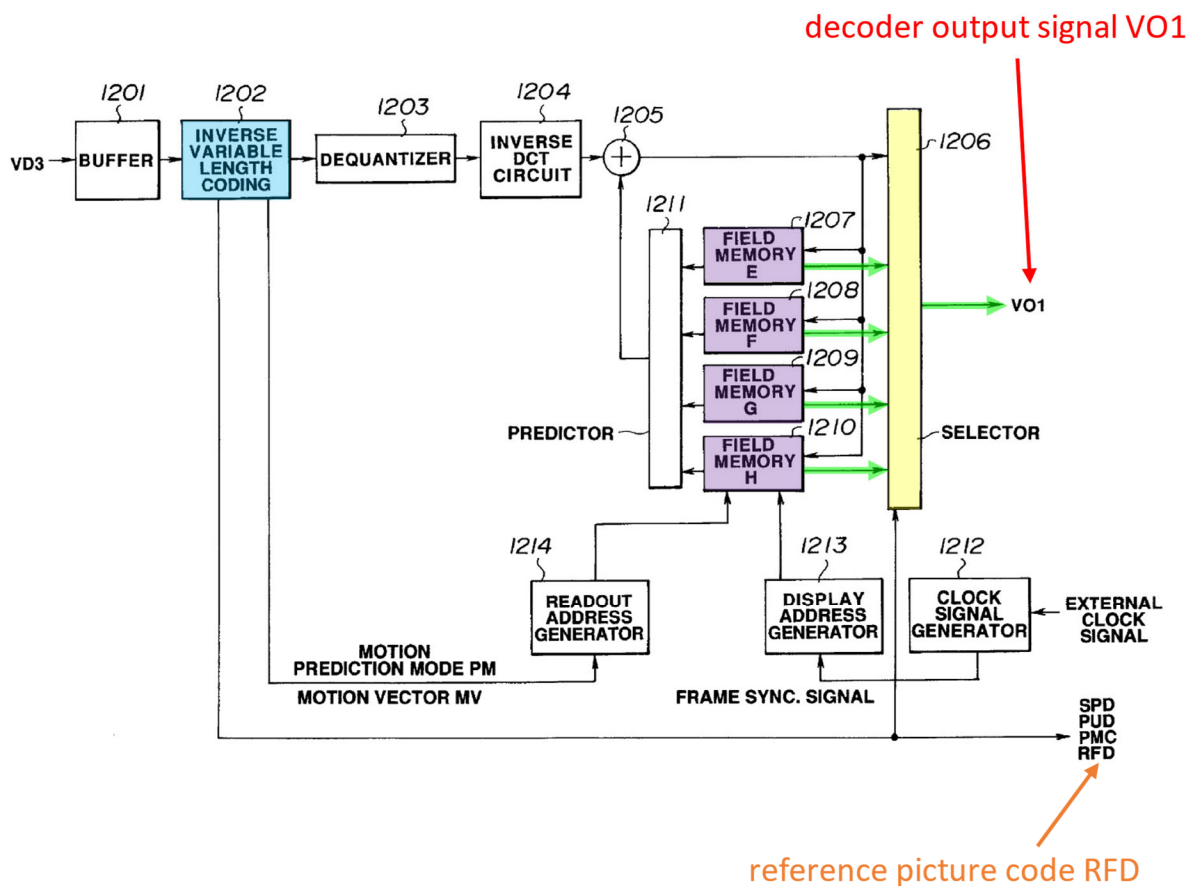


Yagasaki, Figure 14 (annotated).

60. In the following paragraphs, I explain the operation of decoder 113 (a) when the skip-picture flag indicates a skipped picture and (b) when the skip-picture flag does not indicate a skipped picture.

61. **(a) When skip-picture flag indicates a skipped picture:** Referring to annotated Figure 14 below, “when [the skip picture flag SPD] indicates a skip picture, ...the selector 1206 [annotated yellow]...provide[s] the picture indicated by the reference picture code [annotated orange] as a picture of the decoder output signal VO1 [annotated red].” Yagasaki, 15:8-11, 15:12-14 (“[T]he decoder of the

present embodiment copies reference pictures to provide skip pictures to replace the skip pictures that were eliminated in the coding process.”). Selector 1206 retrieves the reference pictures from field memories 1207 through 1210 (shaded purple). Yagasaki, 15:1-67, Figures 14-16. The data path from field memories 1207 through 1210 to selector 1206 and then decoder output signal VO1 is annotated green.

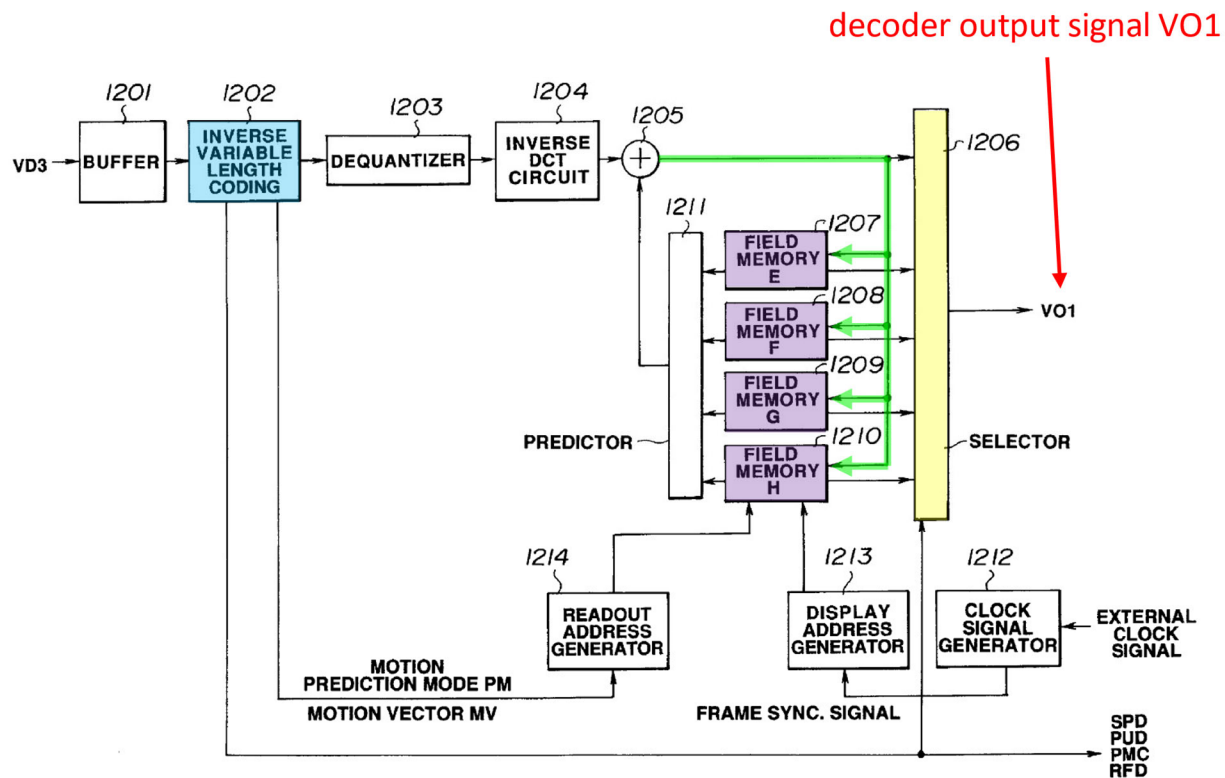


Yagasaki, Figure 14 (annotated).

62. (b) When skip-picture flag does not indicate a skipped picture:

When the skip-picture flag does not indicate a skip picture, as illustrated in annotated Figure 14 below, adder 1205 combines macroblocks of the current picture signal's

motion prediction errors with reference macroblocks generated “from one or more reference pictures stored in the field memories 1207 through 1210” (shaded purple). Yagasaki, 14:39-57, 14:62-67. The resulting “reconstituted current picture” is then fed “to the field memories 1207 through 1210” and to selector 1206 (shaded yellow) for output to decoder output signal VO1 (annotated red). Yagasaki, 14:54-60. The data path is annotated green.



Yagasaki, Figure 14 (annotated).

63. Decoder 843 thus decodes each picture signal in decoder input signal VD3 and stores it in field memories 1207 through 1210 in a decoded format,

ensuring a steady stream of decoded images is available for use in the field memories.

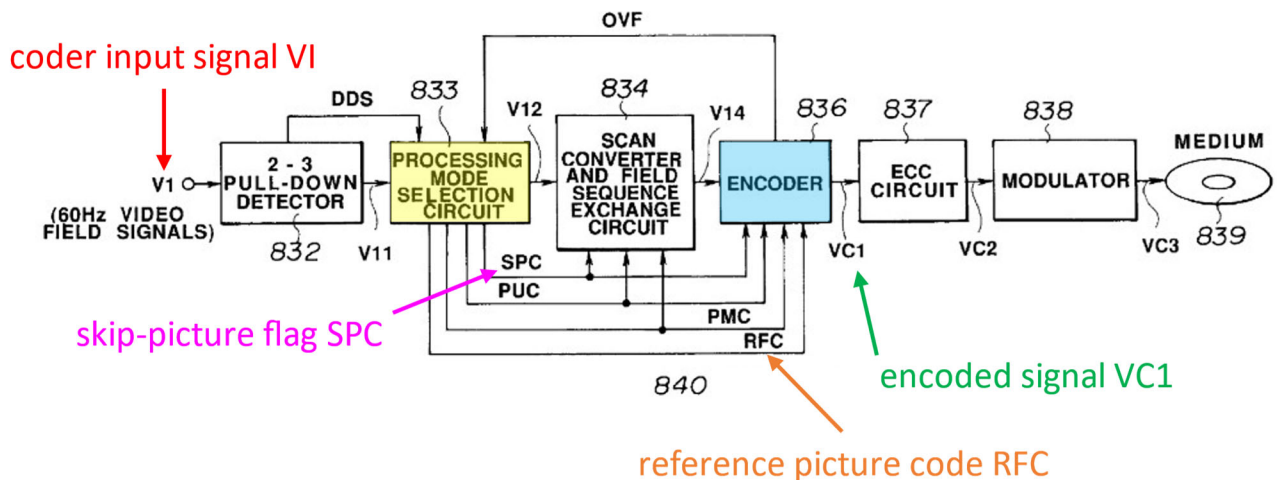
b. Second embodiment

64. Yagasaki's second embodiment builds upon the first embodiment's teachings by introducing a "temporal_reference code," which "is a 10-bit code that indicates the display sequence of the picture in a group of pictures (GOP)." Yagasaki, 19:20-22. Encoder 836 of coding apparatus 840 includes the temporal_reference code in the coded signal—together with the skip-picture flag and reference picture code, as was done in the first embodiment—for extraction by decoder 843 of decoding apparatus 850. Yagasaki, 19:15-20, 19:27-37, 19:44-50, Figures 20-21. I note that Yagasaki mistakenly refers to encoder 836 as encoder 863 at 19:10, 19:16, and 19:18. This Declaration applies the correct designation, encoder 836, when referencing these instances.

(i) Coding apparatus 840

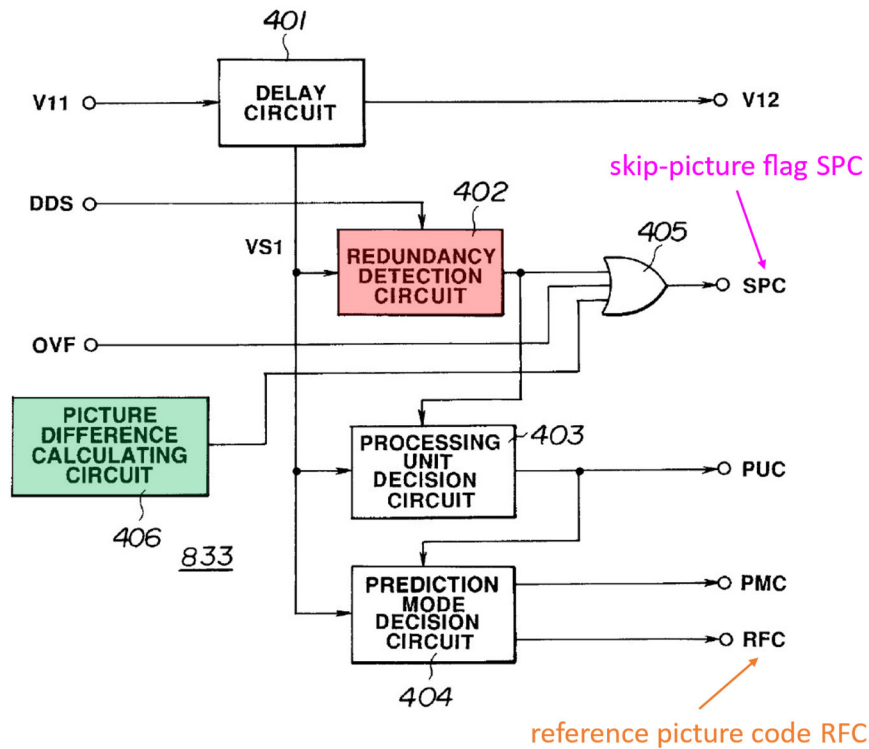
65. The top half of Figure 17, shown below with annotations, illustrates coding apparatus 840. Coding apparatus 840 generates control signals and flags "in the same manner as described" for the first embodiment. Yagasaki, 16:3-35. As with its first embodiment counterpart, "encoder 836 [shaded blue]...includes the skip picture flag SPC[] [annotated pink]...and the reference picture code RFC

[annotated **orange**]...in the encoder output signal VC1 [annotated **green**]” “in lieu of each skip picture[.]” Yagasaki, 16:53-63, 18:25-32.



Yagasaki, top Figure 17 (annotated).

66. Processing mode selection circuit 833 (annotated **yellow** in Figure 17 above) is further detailed in Figure 18, shown below with annotations. Yagasaki, 6:45-47, 17:51-54. “The processing mode selection circuit 833 generates [control signals and flags] for each picture represented by the video signal VI1[] in the same manner as described” for the first embodiment. *See* Yagasaki, 16:9-31, 17:53-54. Video signal VI1 is the same as coder input signal VI (annotated **red** in Figure 17 above) only “delayed by a delay time.” Yagasaki, 16:9-12. I further note that Figure 17 shows VI as **VI**, but a POSITA would have understood that they are meant to be the same. Processing mode selection circuit 833 therefore generates control signals and flags for each picture represented by coder input signal VI.



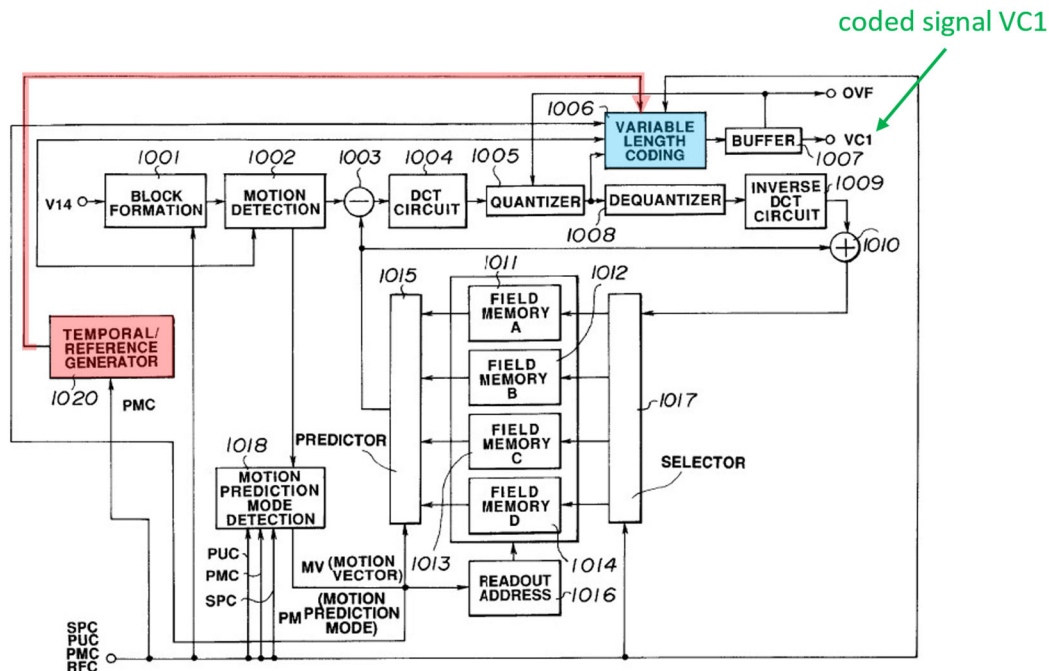
Yagasaki, Figure 18 (annotated).

67. Remaining with annotated Figure 18 above, “redundancy detection circuit 402 [annotated **red**] detects duplicate pictures (field or frame),” which Yagasaki defines as “a picture that exhibits little or no change relative to another picture.” Yagasaki, 17:64-67. “The redundancy detection circuit 402 generates the skip picture flag SPC [annotated **pink**] in response to each [detected] duplicate picture.” Yagasaki, 17:67-18:3. “The picture difference calculating circuit 406 [annotated **green**] calculates the pixel-by-pixel difference between the current picture and plural other pictures in the video signal VS1,” which is the video signal V11 subjected to “different multiples of a field period and” fed out as “multiple delayed signals.” Yagasaki, 17:55-58, 18:4-6. “If the sum of the differences

between the current picture and one of the other pictures is not greater than a certain threshold value, it is deemed that the picture exhibits little or no change relative to that other picture, and the picture difference calculating circuit 406 generates a skip picture flag SPC.” “In addition, the prediction mode decision circuit 404 generates a reference picture code RFC [annotated orange] indicating the other picture as the reference picture for reproducing the skip picture.” Yagasaki, 18:6-14. “The threshold value may be increased when it is desired to increase the number of duplicate pictures skipped[] and may be reduced when it is desired to increase the picture quality.” Yagasaki, 18:14-17.

68. Encoder 836 is further detailed in Figure 20, shown below with annotations. Yagasaki, 6:41-43, 6:50-52, 16:1-5, 17:1-2, 19:16. Yagasaki explains that encoder 836 “is substantially similar to” encoder 106 of the first embodiment, discussed above in Section VIII.1.a(i), except that encoder 836 “additionally includes the temporal reference generating circuit 1020 [shaded red], which generates[]...a temporal_reference code for each picture.” Yagasaki, 19:16-21. A POSITA would have understood that Yagasaki, which describes its invention in the context of “MPEG coding” (Yagasaki, 2:47), assigns temporal_reference codes at the frame level. *See* EX-1026, 50 (“The temporal_reference is a 10-bit unsigned integer associated with each coded picture. ... When a frame is coded as two field pictures, the temporal_reference associated with each coded picture shall be the

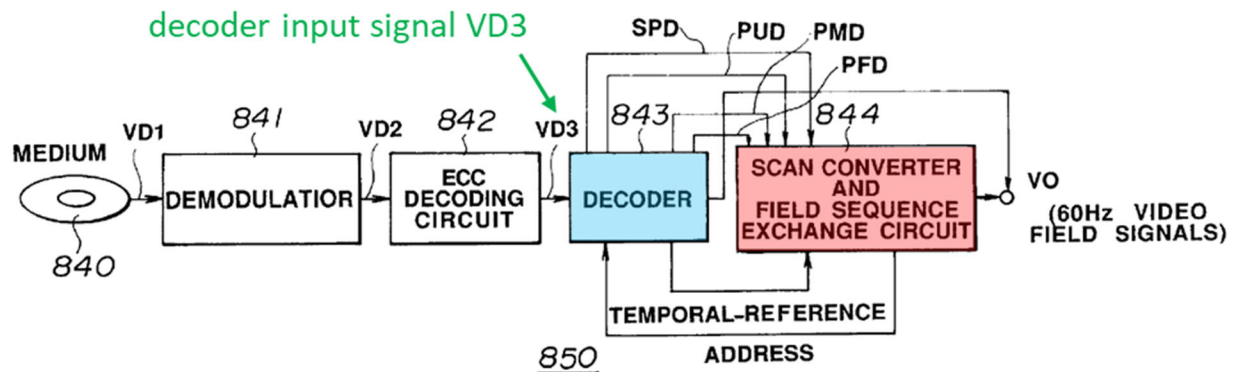
same. The temporal_reference of each coded frame shall increment by one modulo 1024 when examined in display order at the output of the decoding process, except when a group of pictures header occurs. After a group of pictures header, the temporal_reference of the first frame in display order shall be set to zero.”); EX-1033, 44 (“The temporal-reference is a 10-bit unsigned integer associated with each input picture.”). “The temporal reference generator feeds the temporal reference code to the variable-length coder 1006 [shaded blue] for inclusion in the coding apparatus output signal VC1 [annotated green].” Yagasaki, 19:27-29. The data path is annotated red.



Yagasaki, Figure 20 (annotated).

(ii) Decoding apparatus 850

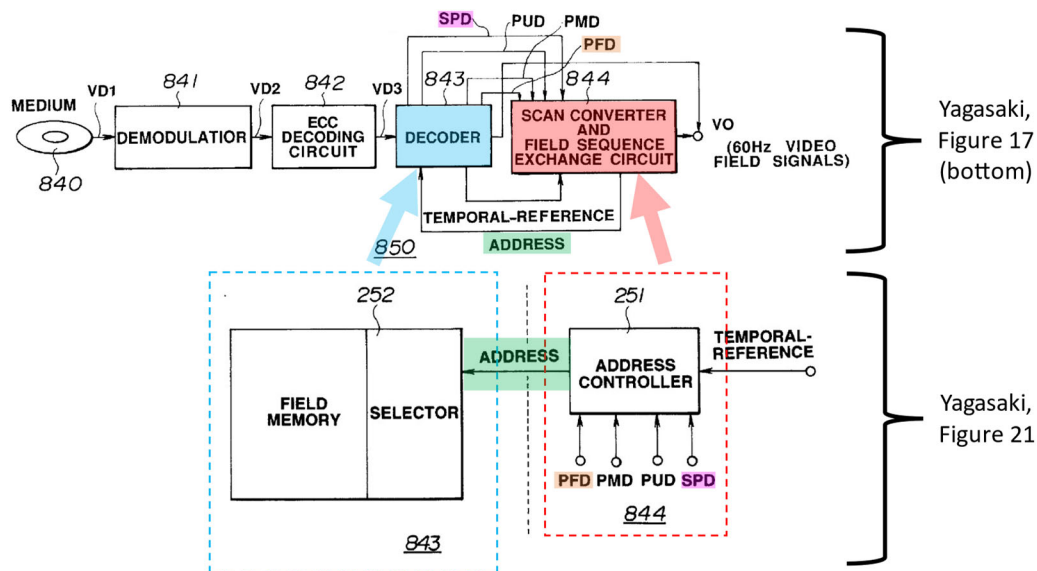
69. Decoding apparatus 850, illustrated in annotated Figure 17 below, feeds decoder input signal VD3 (annotated **green**) to decoder 843 (shaded **blue**) after demodulation and error decoding, mirroring the process used in the first embodiment. Yagasaki, 17:1-8. Yagasaki explains that “construction of the decoder 843 is similar to that” of its first embodiment counterpart—decoder 113, discussed above in Section VIII.1.a(ii)—other than “the relation between the decoder 843 and the field sequence exchanging and scan converting circuit 844 [annotated **red**].” Yagasaki, 19:38-43. Decoding apparatus 850 outputs “decoding apparatus output signal VO[, which] is a video signal that is suitable for display on a monitor after conversion to an analog signal.” Yagasaki, 17:42-45.



Yagasaki, bottom Figure 17 (annotated).

70. As illustrated in annotated Figures 17 and 21 below, the relation between decoder 843 (annotated **blue**) and field sequence exchanging and scan converting circuit 844 (annotated **red**) concerns the processing of flags and codes

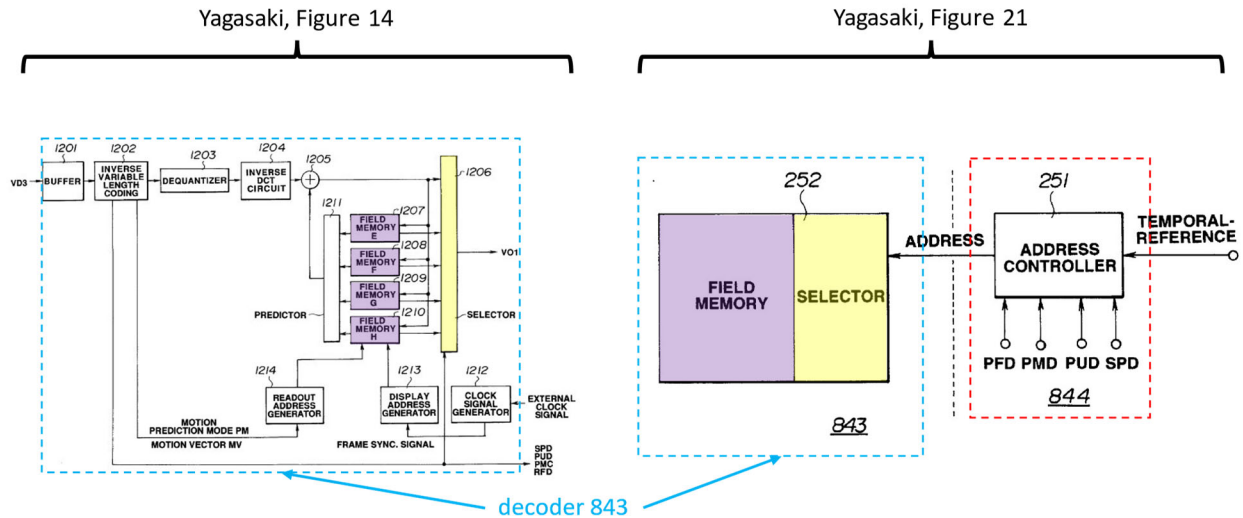
from input signal VD3 for use in the decoding process. Yagasaki, 19:38-63, Figure 21. Field sequence exchanging and scan converting circuit 844 “receives [*inter alia*] the skip picture flag SPD [annotated pink] [and] the reference picture code RFD [annotated orange]” and “[i]n response..., feeds readout addresses [annotated green] to the field memory set 252...to cause the field memory set 252 to read out the field of the decoding apparatus output signal VO indicated by each readout address.” Yagasaki, 19:55-63. I note that Figures 17 and 21 both mislabel reference picture code RFD as PFD; this is apparent from the discussion of those figures in the specification. See Yagasaki, 17:19, 17:41, 19:48, 19:54, 19:57.



Yagasaki, bottom Figure 17 and Figure 21 (annotated).

71. As illustrated in annotated Figures 14 and 21 below, “field memory set 252 is provided by the field memories 1207 through 1210 [annotated purple] and

the selector 1206 [annotated **yellow**] shown in [Figure] 14[,]” discussed above in Section VIII.1.a(ii) for the first embodiment. Yagasaki, 19:63-65.



Yagasaki, Figures 14 and 21 (annotated).

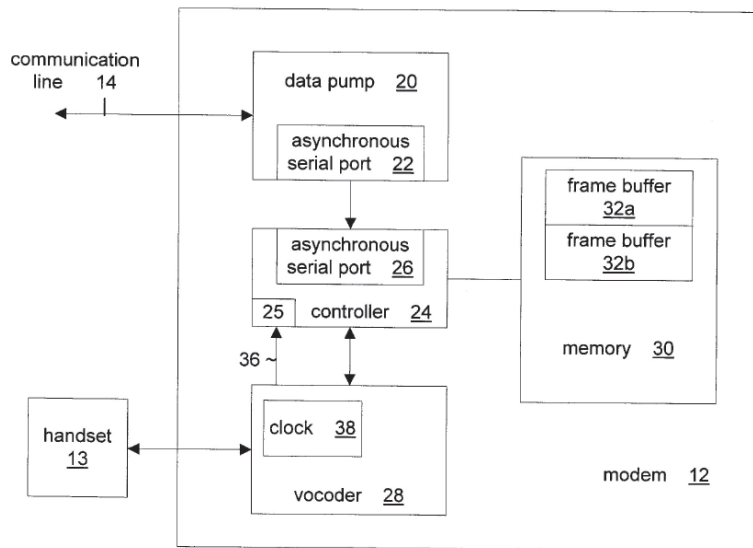
2. Oliver (U.S. Patent No. 6,064,693)

72. Oliver relates to a method of handling “data underrun” by providing “filler frames” during decoding in place of missing frames. Oliver, Abstract, 2:45-63, 3:25-28, 3:45-49, 4:52-55. Data underrun occurs when a buffer is fed with data at a lower speed than the data is being read from the buffer. Oliver, 2:45-51, 5:66-6:8. In 2002, a POSITA would have been well aware of the issue of buffer data underrun, as it was a common challenge in the field of data streaming and storage technologies. Buffer underrun (also called underflow) can lead to interruptions in data processing or transmission, presenting a challenge that was widely recognized and addressed in various computing and networking contexts at that time. *See, e.g.,*

MPEG-1, 66 (“It is a requirement of the encoder...that the bitstream it produces will not cause the VBV input buffer to either overflow or underflow.”); MPEG-2, 144 (“VBV buffer underflow shall not occur when the low_delay flag is equal to 0. This requires that all picture data for the n-th picture shall be present in the VBV buffer at the decoding time, t_n ”).

73. “A filler frame is defined...as a frame inserted into a stream of data frames to compensate for lack of reception of one or more frames in the transmission.” Oliver, 4:52-55. Oliver’s method substitutes filler frames—which “have little or no impact on the system”—for frames expected in, but absent from, the decoder’s buffer memory. Oliver, 4:55-56; *see also id.*, 3:30-45.

74. Figure 2, below, illustrates modem 12 in an embodiment. Oliver, 3:56-59, 4:64-65. Controller 24 receives a series of “compressed...frames[] ...and provide[s] the data frames to a vocoder 28” for decoding. Oliver, 5:17-20, 5:48-51. Vocoder 28 “may comprise any device which decodes encoded data, such as [an] MPEG video decoder.” Oliver, 9:31-32; *see also id.*, 4:47-52 (“[T]he present invention...is applicable to any digital data which is transferred in frames, such as video data[.]”). “[M]emory [30] comprises two or more buffers [32] for buffering frames” “before being provided to the vocoder 28.” Oliver, 6:9-17.



Oliver, Figure 2.

75. “[C]ontroller 24 retriev[es] frames from the buffers 32 and provid[es] the frames to the vocoder 28 in a manner which handles data underrun[.]” Oliver, 8:13-17. Specifically, “controller 24...handle[s] data underrun of [received] frames...by providing filler frames, preferably either ‘quiet’ frames or repeat frames[.]” Oliver, 5:27-30. During operation, “controller 24 maintains a [memory pointer] to...a [next] data frame...to provide to the vocoder 28.” Oliver, 8:17-23. Upon receipt of a frame, “controller 24 determines if [frame buffers 32] has a valid frame[.]” Oliver, 8:45-47. “If the next data frame [in the series of data frames] has not been received, the method provides a filler frame[...]to the decoder.” Oliver, 3:32-36, 8:52-55. The method “repeatedly performs these steps for each of the data frames in the series of frames.” Oliver, Abstract.

3. Lyon (International Publication No. WO 91/06182)

76. Yagasaki cites Lyon on its face. Yagasaki, [56]. Lyon describes a process for editing video signals. Lyon, Abstract, 5. Specifically, Lyon describes an editing process during which “artificial” video frames are generated. Lyon, 5-6, 11. Sometimes these artificial frames can include “visually disturbing artefacts” when an editing cut juxtaposes the last field of one edited sequence with the first field of the next sequence.² Lyon, 5-6. Such frames disrupt the smooth visual flow of the video. Lyon, 6.

77. Lyon explains that “steps should be taken to ensure that any [such] ‘artificial’ intermediate video frames...are identified...and processed accordingly.” Lyon, 5-6; *see also id.*, 11 (“[S]teps are preferably taken to ensure that a frame having fields from either side of a cut, is not treated in the same manner as a normal decoded frame.”).

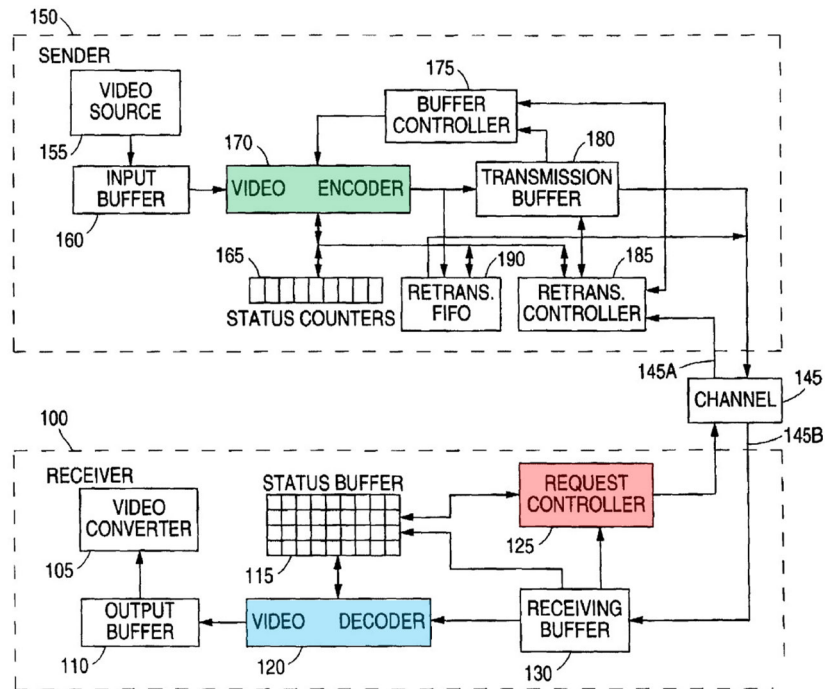
4. Ran (U.S. Patent No. 5,768,533)

78. Ran “relates...to coding, transmission, and display of video in telecommunication systems having bursty channel errors[,]” *i.e.*, high error rates.

² In interlaced video, each video frame comprises two interleaved fields – one containing the odd-numbered lines of the image, and the other containing the even-numbered lines.

Ran, 1:8-10. Figure 1, below, illustrates a communication system in an embodiment.

Ran, 3:28-29, 4:15-16. Sender 150, comprising video encoder 170 (annotated **green**), transmits to receiver 100, comprising video decoder 120 (annotated **blue**), “a series of video data packets...represent[ing] a sequence of frames which form a moving image.” Ran, 3:38-41, 4:15-38. When receiver 100 detects an error, “request controller 125 [shaded **red**] sends a retransmission request to sender 150” requesting retransmission of the packet that contained the error. Ran, 7:59-8:9; *see also id.*, 3:46-48.



Ran, Figure 1 (annotated).

79. When “all data needed for a frame has been received without errors, the frame is ‘displayable’, i.e.,] ready to be decoded and/or displayed.” Ran, 3:56-59;

see also id., 3:9-10. Ran teaches that a frame that contains **inter coding** is “displayable” if said frame has been received without errors and the “previous data” (*i.e.*, the reference picture) on which it relies has also been received without errors. Ran, 3:3-9, 9:62-10:17; *see also id.*, 6:3-32. Put differently, if a reference picture required to decode an inter-coded frame “was never received correctly” by the decoder, an error is detected and a retransmission request is made because the frame “cannot be correctly inter decoded.” Ran, 9:58-10:46.

80. I note that Ran describes its invention in terms of “frame segments,” each being “part of a frame of the moving image.” Ran, 2:35-38. This is consistent with the ’134 patent, which explains that “[t]emporal prediction is typically carried out in video coding methods block- or macroblock-specifically, instead of image-frame-specifically.” EX-1001, 2:67-3:2.

IX. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Yagasaki Renders Obvious Claims 1-5, 9-12, and 16-19

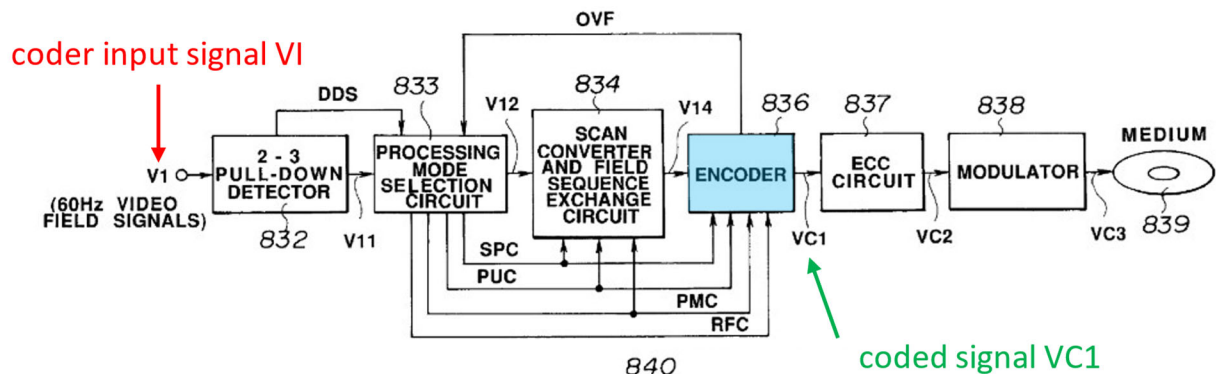
1. Claims 1-5

a. Independent Claim 1

- (i) 1[pre]: “A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:”**

81. In my opinion, Yagasaki discloses or renders obvious 1[pre]. Yagasaki discloses “a method for decoding a coded signal to provide an output video signal.”

Yagasaki, 3:36-37; *see also id.*, 3:3-5, 3:54-57, 4:41-42, 4:61-63, 8:33-35, 21:62-64. Yagasaki's "coded signal" teaches the claimed "*compressed video sequence*."³ The top half of Figure 17, reproduced below with annotations, illustrates coding apparatus 840 of Yagasaki's second embodiment. Yagasaki, 16:1-5. Coding apparatus 840 generates flags and codes for each picture signal in coder input signal VI (annotated **red**) and codes the picture signal—along with the generated flags and codes—to produce coded signal VC1 (annotated **green**). Yagasaki, 16:5-67; *see also id.*, 5:45-56. Specifically, encoder 836 (shaded **blue**) predictively codes each picture signal as an I-, P-, or B-picture. Yagasaki, 16:23-28, 16:53-57.



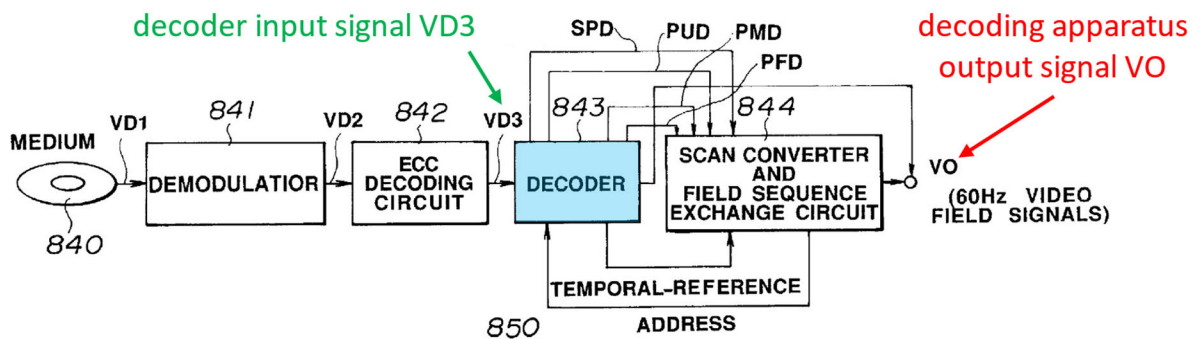
Yagasaki, top Figure 17 (annotated).

82. Yagasaki's coded signal therefore teaches the '134 patent's "*compressed video sequence*," which is a sequence of I-, P-, and/or B-frames. EX-1001, 2:5-55, 21:22-33, Figures 1-2. As I have previously discussed in ¶¶35-37, a

³ Claim language is indicated with italics throughout this Declaration.

POSITA in 2002 would have understood a “*compressed video sequence*” to commonly refer to a sequence of I-, P-, and B-frames due at least to the widespread adoption of the MPEG standards, as well as other early digital video compression standards that employed coding schemes that utilized such frame types.

83. Yagasaki further discloses “*decoding*” the coded signal (“*compressed video sequence*”). The bottom half of Yagasaki’s Figure 17 (below) illustrates decoding apparatus 850 of the second embodiment. Yagasaki, 16:1-3, 17:1-2. Yagasaki explains that decoding apparatus 850 produces “a video signal that is suitable for display on a monitor after conversion to an analog signal.” Yagasaki, 17:42-45. Decoder 843 (shaded blue) extracts from decoder input signal VD3 (annotated green) “flags and codes” originally generated by coding apparatus 840 and “decodes the picture signal in the decoder input signal VD3 to provide the decoding apparatus output signal VO [annotated red].” Yagasaki, 17:9-22.



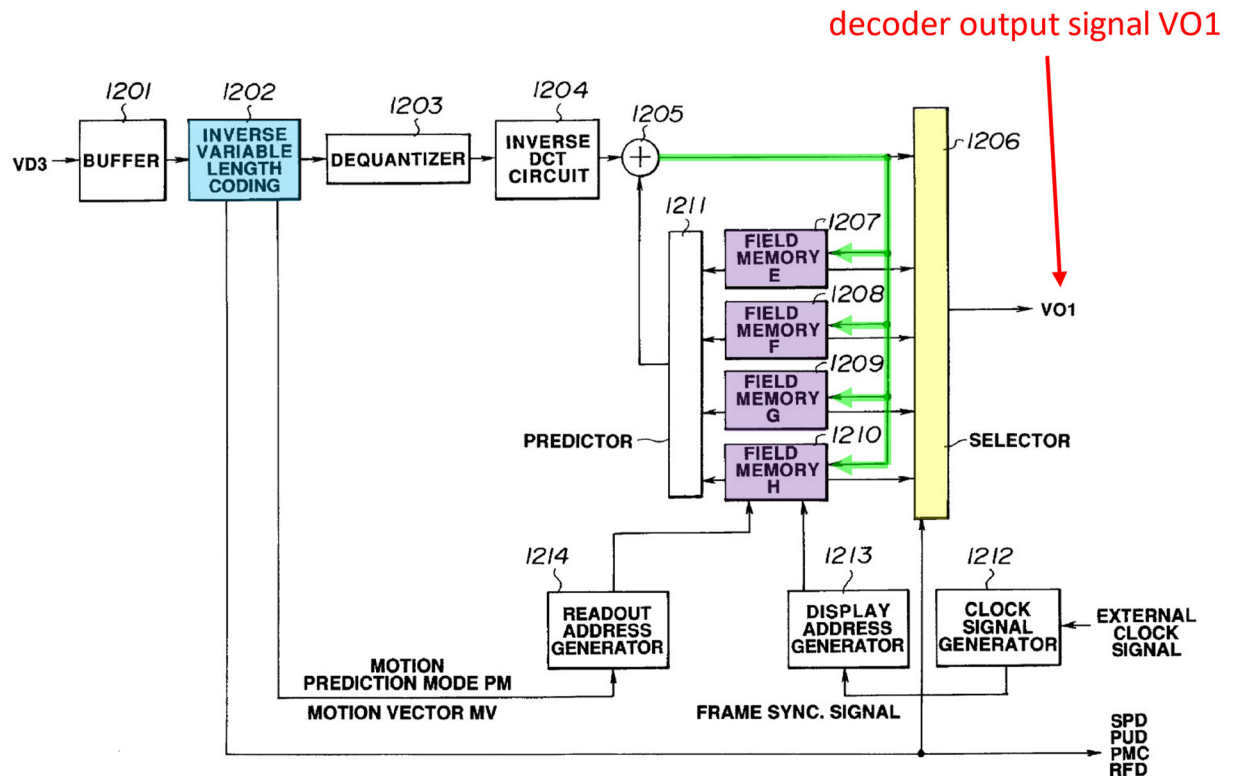
Yagasaki, bottom Figure 17 (annotated).

84. As I discuss above in ¶¶32-36, these steps would have been commonly known and accepted by a POSITA as steps for decompressing (decoding) a compressed video sequence. Yagasaki therefore discloses “[a] method for decoding a compressed video sequence.”

85. Furthermore, in Yagasaki’s method, “image frames are entered into a buffer memory,” as described in the ’134 patent and recited in 1[pre]. For example, as discussed below, Yagasaki’s decoder 843 uses field memories 1207 through 1210 to store previously decoded pictures for use as reference pictures. Yagasaki, 14:54-67, Figure 14. As I explain below, this process is analogous to the ’134 patent’s use of a buffer memory during decoding to store the last M coded frames in decoded form for use as reference images. EX-1001, 19:54-20:1.

86. As I explain above in Section VIII.1.b(ii), Figure 14 of Yagasaki shows the relevant construction of decoder 843 in Yagasaki’s second embodiment. Figure 14 illustrates how field memories 1207 through 1210 (shaded purple) are used in the decoding process. As I discuss above in Section VIII.1.a(ii), for each picture signal in input signal VD3, “reference pictures stored in the field memories 1207 through 1210” are used to “generate[] motion-compensated macroblocks” for “reconstitut[ing a] current picture.” Yagasaki, 14:60-67. “The resulting...picture is fed...to the field memories 1207 through 1210” for storage (data path highlighted

green) and is also fed to selector 1206 (shaded yellow) for output to the decoder output signal VO1 (annotated red). Yagasaki, 14:54-60.



Yagasaki, Figure 14 (annotated).

87. In Yagasaki, the term “reconstitute” is used interchangeably with “decode,” meaning to reconstruct a decoded video picture from the encoded video bitstream (*see, e.g.,* Yagasaki, 1:26-28 (“the video signal representing the current picture (a frame or a field) is predicted from the *decoded and reproduced*

*(reconstituted)*⁴ video signal representing a reference picture”); Yagasaki, 14:54-58 (“Each macroblock of motion prediction errors is fed from the inverse DCT circuit 1204 to the adder 1205, where it is added to a corresponding reference macroblock from the predictor 1211. ***The resulting macroblock of the reconstituted current picture*** is fed to the selector 1206...”)), or in some contexts meaning to perform one or more steps of the decoding process for a video picture (*see, e.g.*, Yagasaki, 13:39-45 (“The quantized DCT coefficients are also fed from the quantizer 1005 to the dequantizer 1008, where the quantizing applied by the quantizer 1005 is reversed, and thence to the inverse DCT circuit 1009, which performs an inverse orthogonal transform complementary to that performed by the DCT circuit 1004. ***Each resulting macroblock of reconstituted motion prediction errors*** is fed to the adder 1010...”).

88. Each picture signal is thus decoded and stored in field memories 1207 through 1210 for use as reference pictures for decoding the remaining picture signals in input signal VD3. *See* Yagasaki, 14:57-60. Field memories 1207 through 1210 are a “*buffer memory*” because they are used to temporarily store these decoded pictures for subsequent use as temporal prediction reference pictures, including for

⁴ All emphases within quotations in this Declaration have been added unless otherwise stated.

retrieval and manipulation in the decoding and motion compensation calculation processes. Riley, 28 (“A motion-compensated reference frame is created using the previous decoded frame and the motion vectors for the current frame. The current frame is reconstructed by adding the difference frame to this reference frame. This frame is displayed *and is also stored in the decoder frame store*”), 29. A POSITA would have understood that Yagasaki’s field memories 1207 through 1210, like Riley’s frame store (*see also* Riley at Figure 3.7), are a frame buffer memory as that term is used in the ’134 patent because they are used to store decoded (reconstructed/reconstituted) video pictures for use as reference pictures for decoding subsequent pictures that will be received in the encoded video signal.

89. Yagasaki’s decoder 843 uses the decoded pictures stored in field memories 1207 through 1210 to generate motion-compensated macroblocks for adding to motion prediction error macroblocks of a current picture, *i.e.*, for purposes of performing motion compensation to decode the current picture of the coded signal. Yagasaki, 14:54-67. As I previously discuss in ¶36, these steps are standard in a motion-compensated prediction decoding process. The decoded pictures are thus entered into field memories 1207 through 1210 “*in connection with the decoding*” because the decoded pictures serve as reference pictures for subsequent decoding operations.

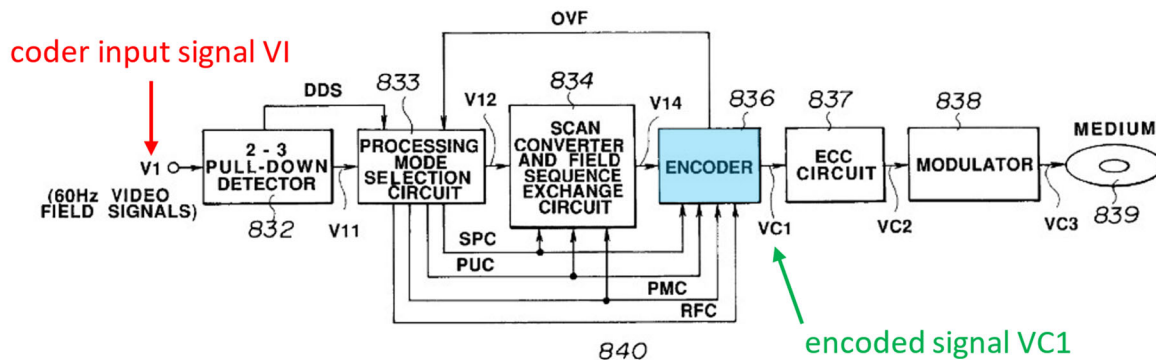
90. Thus, Yagasaki teaches a method for decoding a coded signal (“*a compressed video sequence*”) using field memories 1207 through 1210 (“*a buffer memory*”) to store decoded pictures (“*image frames*”).

(ii) 1[a]: “decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;”

91. In my opinion, Yagasaki discloses or renders obvious 1[a]. As I discuss above in Section VIII.1, the coded signal in Yagasaki’s first and second embodiments includes skip-picture flags in place of eliminated pictures. Yagasaki, Abstract, 3:18-35, 4:8-15, 4:30-41, 4:61-5:16, 5:45-64, 15:12-67, 18:25-32, Figures 15-16, claims 1-2, 9-12. As I explain below, in Yagasaki’s second embodiment, each eliminated picture corresponds to a discontinuity in “temporal_reference code[s]” assigned to every picture in the coded signal. I further note that the antecedent basis for “*the image frames*” in 1[a] is “*image frames...entered into a buffer memory*” in 1[pre]. Yagasaki’s skip-picture flag thus “*inform[s] an intentional discontinuity of numbering*” of the decoded pictures stored in field memories 1207 through 1210 (“*the image frames*”), as described in the ’134 patent and recited in 1[a].

92. The top half of Figure 17, below, illustrates coding apparatus 840 in Yagasaki’s second embodiment. Yagasaki, 16:1-5. Encoder 836 (shaded blue) generates a “temporal_reference code...for each consecutive picture in the coder

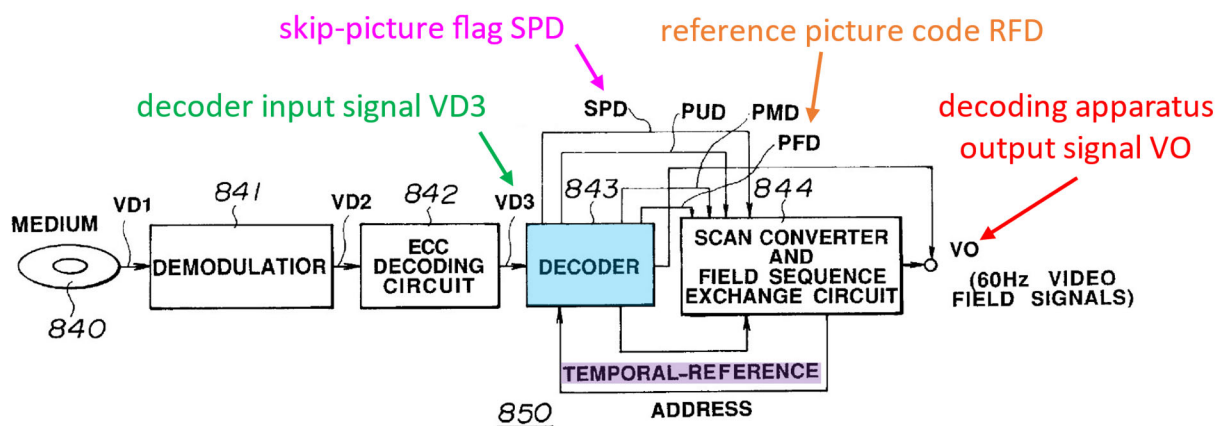
input signal VI [annotated **red**].” Yagasaki, 19:18-26, Figure 20. “The temporal_reference code...is generated by *incrementing a counter by one* for each consecutive picture in the coder input signal VI[.]” Yagasaki, 19:21-26, Figure 20. Yagasaki’s temporal_reference code is therefore a “*numbering*” as described in the ’134 patent and recited in 1[a]. EX-1001, 8:17-20 (“[T]he video frames of the sub-sequence are given image numbers, using for example *consecutive numbering that starts with the number 0*[.]”); see also *id.*, 3:63-4:3, 7:63-67, 8:34-41. Encoder 836 includes each temporal_reference code in coded signal VC1 (annotated **green**). Yagasaki, 19:28-30, Figure 20.



Yagasaki, top Figure 17 (annotated).

93. The bottom half of Figure 17, below, illustrates decoding apparatus 850 in Yagasaki’s second embodiment. Yagasaki, 6:41-43, 17:1-2. As I discuss above for 1[pre], during the decoding process, each picture signal in input signal VD3 (annotated **green**) is decoded and stored in field memories 1207 through 1210 of decoder 843 (shaded **blue**). Yagasaki, 14:37-67, Figure 14. Each decoded picture

in field memories 1207 through 1210 is therefore associated with a temporal_reference code. Yagasaki, 19:18-30, Figure 20. Yagasaki's temporal_reference code is thus a “numbering” applied to each of the decoded pictures stored in Yagasaki's field memories 1207 through 1210, *i.e.*, “the images.” Yagasaki therefore teaches “numbering of the image frames.”



Yagasaki, bottom Figure 17 (annotated).

94. Yagasaki also teaches “*decoding from the compressed video sequence an indication.*” As illustrated above, “decoder input signal VD3 [annotated green] is fed into the decoder 843 [shaded blue], which extracts from the decoder input signal VD3 the skip picture flag [annotated pink], ...the reference picture code [annotated orange],⁵ and the temporal reference [annotated purple]” “originally

⁵ As I note above in Section VIII.1.b(ii), Figure 17 mislabels reference picture code RFD as *PFD*.

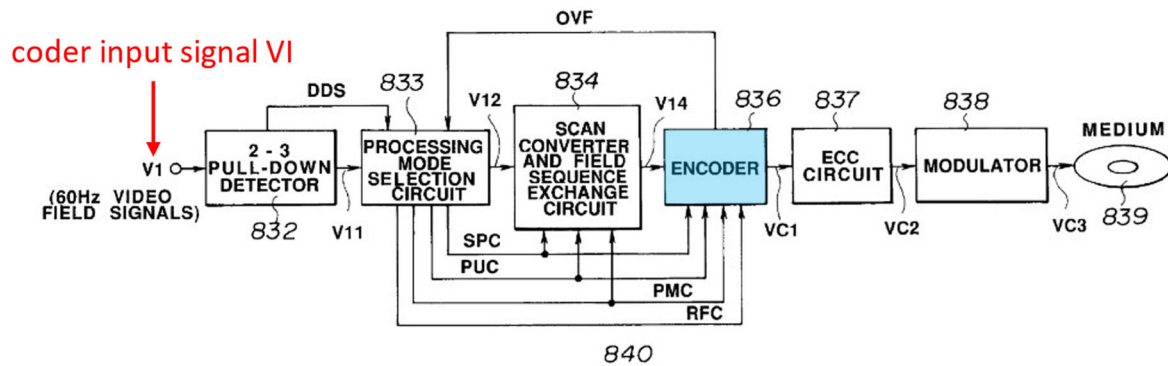
generated...in the coding apparatus 840” for each picture in decoder input signal VD3. Yagasaki, 17:9-16. Specifically, variable-length decoder 1202 within decoder 843 “extracts” *inter alia* the skip-picture flag by “***revers[ing] the variable length coding*** applied by the variable length coder 1006 in the encoder [106].” Yagasaki, 14:39-44, Figure 14. Variable-length coder 1006 applies “variable-length coding such as Hufmann [sic] coding[] or run-length coding.” Yagasaki, 13:33-35.

95. Reversing a variable-length coding scheme such as Huffman coding or a run-length coding scheme is “*decoding*” because such processes involve translating encoded data back into its original form or a form that is readily understandable by humans or machines. Variable-length coding schemes such as Huffman coding compress data by assigning shorter codes to more frequently occurring symbols and longer codes to less frequent symbols. Run-length coding schemes, on the other hand, compress data by reducing sequences of the same value to a single value and count. Decoding, therefore, involves the reverse process: interpreting these specialized coding schemes to reconstruct the original data.

96. In Huffman coding, decoding is the process of following the code-to-symbol mappings defined in the Huffman tree to translate the compressed sequence of bits back into the original symbols. For run-length coding, decoding translates the compressed pairs of values and counts back into sequences of data. A POSITA would have recognized these operations as decoding because they are standard

procedures for converting data from a compressed encoding back into a usable or original form, which is the essence of what decoding entails in the realm of data compression and transmission. Yagasaki thus teaches decoding from decoder input signal VD3 (“*the compressed video sequence*”) a skip-picture flag (“*an indication*”).

97. In Yagasaki’s system, the elimination of a skip picture by the encoder results in a discontinuity in the sequence of temporal_reference codes. As Yagasaki explains, and as illustrated in the top half of Figure 17 below, a temporal_reference code is generated “for each consecutive picture in the coder input signal VI” (annotated **red**), *i.e.*, for each picture in the video signal that is provided as the initial input to coding apparatus 840. Yagasaki, 19:18-26. Skip pictures are eliminated during “the predictive coding step.” Yagasaki, claim 1 (“wherein the predictive coding step includes eliminating...fields from the input video signal”), claims 3-6. The predictive coding step takes place in encoder 836 (shaded **blue**) because that is where the video signal is processed and compressed to produce the encoded output signal VC1. Yagasaki, 14:47-67, 19:38-43, Figure 14. I note that although the circuit that generates the temporal_reference code (temporal reference generator 1020) for inclusion in the encoded signal is located in encoder 836 (Yagasaki, 17:9-16, 19:17-20), the temporal_reference codes are logically assigned as soon as coder input signal VI enters the coding apparatus 840. Yagasaki, 19:22-25.



Yagasaki, top Figure 17 (annotated).

98. Because temporal_reference codes are logically assigned to each picture in the input video signal at the beginning of the processing chain (i.e., before skip pictures are eliminated), each skipped picture corresponds to a “discontinuity” in temporal_reference codes, as described in the ’134 patent and recited in 1[a].

99. I further note that the temporal_reference is defined in both the MPEG-1 and MPEG-2 standards. See MPEG-1, 21, 27; MPEG-2, 30, 50. In fact, MPEG-2 expressly states that when “the encoder...skip[s] one or more frames, “[t]his action is indicated by a *discontinuity in the value of temporal_reference* for the next picture.” MPEG-2, 150. Yagasaki is discussed within the context of MPEG. Yagasaki, 2:47-63. A POSITA would have therefore understood that a skipped picture causes a discontinuity in the sequence of temporal_reference codes because the MPEG-2 standard explicitly accounts for the situation in which the encoder skips a frame and explains how temporal_reference values are to be managed in such situations.

100. Finally, I note that during examination of the '134 patent, the applicant admitted that Apostolopoulos discloses “discontinuities of numbering[.]” EX-1002, 2100, 2435 (repeating admission). For this element, the Examiner identified disclosure in Apostolopoulos describing a sequence of video frames that has been separated into subsequences of even frames and odd frames, with no indication that the frames are expressly numbered. EX-1002, 1831 (citing Apostolopoulos, 7:9-37), 2128 (same). The applicant successfully distinguished Apostolopoulos by arguing that the subsequences of even and odd frames in Apostolopoulos were “discontinuities of numbering *caused by errors*,” and were thus not “intentional.” EX-1002, 2100-2101, 2435. This admission supports my opinion that the “*discontinuity of numbering of the image frames*” element of 1[a] requires discontinuity of the frames themselves, i.e., absent/missing frames. Furthermore, in my opinion, Yagasaki discloses more than Apostolopoulos for the “*discontinuity of numbering*” element of 1[a] because Yagasaki expressly discloses frame numbers in the form of temporal_reference codes, as discussed above, as well as discontinuity of the frames themselves (i.e., missing frames).

101. Thus, Yagasaki teaches decoding from the coded signal (“*decoding from the compressed video sequence*”) a skip-picture flag (“*an indication*”) informing an intentional discontinuity of temporal_reference codes (“*informing an*

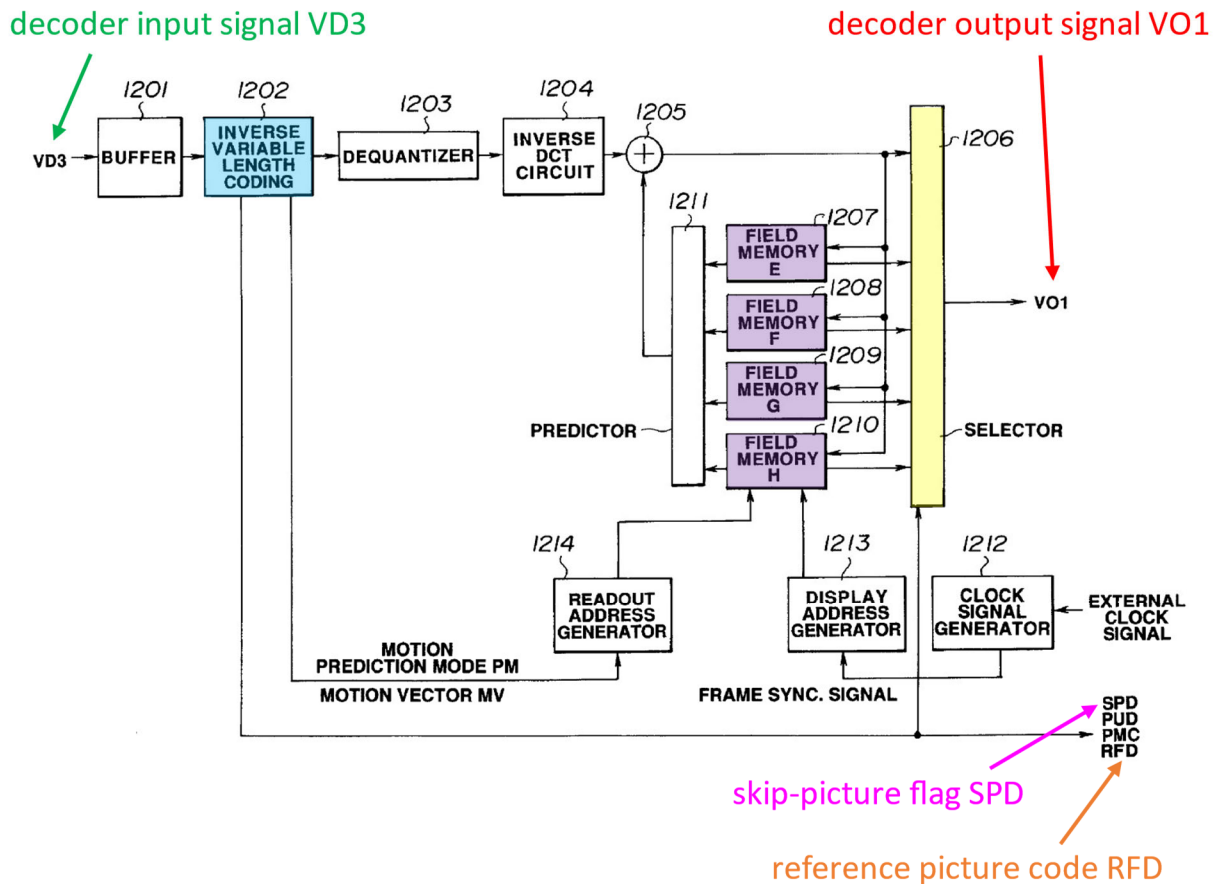
intentional discontinuity of numbering”) of the stored decoded pictures (“*the image frames*”).

- (iii) **1[b]: “configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and”**

102. In my opinion, Yagasaki discloses or renders obvious 1[b]. As I discuss above in Section VIII.1.a(ii), in response to a skip-picture flag (“*an indication*”; 1[a]), Yagasaki’s system “provide[s] the picture indicated by the reference picture code as a picture of the decoder output signal VO1.” Yagasaki, 15:10-11, Figure 14; *see also id.*, 15:12-67, 19:38-65, Figures 15-16, 21. As I explain below, “the picture indicated by the reference picture code” “*correspond[s] to at least one discontinuity in the numbering of the image frames*” because it replaces the eliminated skip picture. Furthermore, Yagasaki teaches decoder 843 “*configuring*” field memories 1207 through 1210 (“*the buffer memory*”) to provide the picture in response to the skip-picture flag, as described in the ’134 patent and recited in 1[b].

103. As I discuss for 1[a] above, during the decoding process, decoder 843 extracts *inter alia* the skip-picture flag and the reference picture code from decoder input signal VD3. Yagasaki, 14:39-44, 17:9-16, 19:38-43, Figures 14, 17. Specifically, variable-length decoder 1202, shaded blue in Figure 14 of Yagasaki (below), “extracts the reference picture code RFD [annotated orange] and the skip

picture flag SPD [annotated **pink**] which, when it indicates a skip picture, causes the selector 1206 [shaded **yellow**] to provide the picture indicated by the reference picture code as a picture of the decoder output signal VO1.” Yagasaki, 15:7-11.

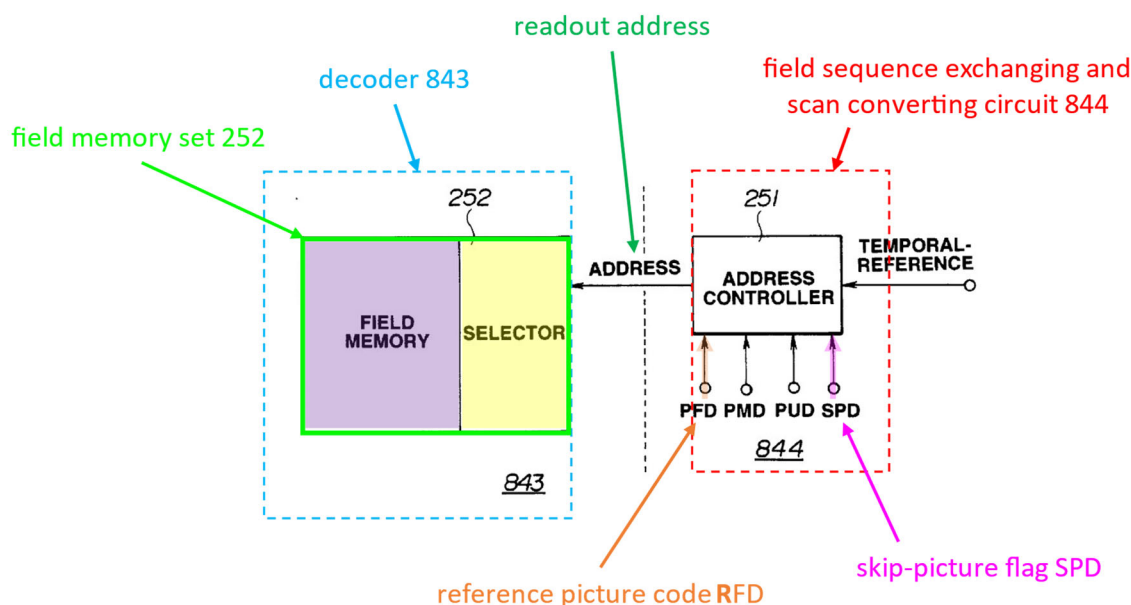


Yagasaki, Figure 14 (annotated).

104. The skip picture flag SPD and reference picture code RFD are fed to field sequence exchanging and scan converting circuit 844, annotated **red** in Yagasaki’s Figure 21 (below). Decoder 843 (annotated **blue**) transmits *inter alia* skip-picture flag SPD (annotated **pink**) and reference picture code RFD (annotated

orange)⁶ to field sequence exchanging and scan converting circuit 844 after extraction from decoder input signal VD3. Yagasaki, 19:38-63. “In response[,]. . . field sequence exchanging and scan converting circuit 844 feeds readout addresses [annotated **green**] to the field memory set 252 [annotated **lime green**] with selector in the decoder 843[.]” Yagasaki, 19:55-61. Field memory set 252 comprises field memories 1207 through 1210 (shaded **purple**) and selector 1206 (shaded **yellow**). Yagasaki, 19:63-65, Figure 14. Selector 1206 “read[s] out the field of the decoding apparatus output signal VO indicated by each readout address” as a picture of the decoder output signal VO1. Yagasaki, 19:61-63; *see also id.*, 15:7-11, Figure 14. In other words, the decoded picture indicated by the readout address is output as a picture of the decoder output signal VO1.

⁶ As I note above in Section VIII.1.b(ii), Figure 21 mislabels reference picture code RFD as **PFD**.



Yagasaki, Figure 21 (annotated).

105. In sum, in response to the skip-picture flag, variable-length decoder 1202 causes field memories 1207 through 1210 to provide a picture—indicated by a reference picture code—to decoder output signal VO1. Yagasaki, 15:7-11, 19:38-65, Figures 14, 21. The reference picture code “indicates, when the picture is a skip picture, the picture that is to be used as the reference picture for reproducing the skip picture.” Yagasaki, 16:29-31. The picture indicated by the reference picture code is thus “*a number of image frames*,” *i.e.*, one frame. The picture indicated by the reference picture code further corresponds to the skip picture (“*at least one discontinuity in the numbering of the image frames*”; 1[a]) because it “replace[s]...the skip picture[]...eliminated in the coding process.” Yagasaki, 15:12-14.

106. Yagasaki thus teaches, in response to the skip-picture flag (*“the indication”*), causing field memories 1207 through 1210 (*“configuring...the buffer memory”*) to provide a picture (*“a number of image frames”*) to replace the skipped picture (*“corresponding to [the] discontinuity”*).

(iv) 1[c]: “using the image frames in the buffer memory in the decoding process.”

107. In my opinion, Yagasaki discloses or renders obvious 1[c]. As I discuss for 1[b] above, Yagasaki’s “reference picture code...indicates...the picture that is to be used as the reference picture for reproducing the skip picture.” Yagasaki, 16:29-31. When decoder 843 encounters a skip-picture flag, the picture indicated by the reference picture code is read out from field memories 1207 through 1210 to the decoding apparatus’s output signal. Yagasaki, 19:55-65, Figure 21; *see also id.*, 15:7-11, Figure 14. A POSITA would have understood that the picture indicated by the reference picture code is *“used in the decoding process”* of Yagasaki because decoder 843 utilizes the picture to bypass the computational overhead associated with generating a motion-compensated picture, which allows for efficient handling of skip-picture scenarios during the decoding process. Yagasaki, 14:54-15:16, Figure 17. Yagasaki thus teaches *“using the image frames in the buffer memory in the decoding process.”*

b. Claim 2: “A method according to claim 1, further comprising using the number of image frames in the buffer memory in

place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.”

108. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for 1[b] above, in response to the skip-picture flag, decoder 843 configures field memories 1207 through 1210 to read out a picture indicated by reference picture code, which is a “*number of image frames in the buffer memory.*” Yagasaki, 16:29-31, 19:55-63. For the reasons I discuss for 1[a] above, the skip-picture flags in Yagasaki correspond to pictures eliminated by the encoder, which are thus “*image frames that do not exist at a decoder.*” Yagasaki, 3:18-24, 18:28-32. For the reasons I discuss for 1[b] above, Yagasaki teaches “*using*” the picture indicated by reference picture code “*in place of*” the eliminated picture because the picture indicated by reference picture code is copied to the decoder output signal to replace the eliminated picture. Yagasaki, 15:7-11, 19:55-65, Figures 14, 21. Yagasaki further teaches that the “*using*” occurs “*due to the at least one discontinuity in the numbering of the image frames*” because it is performed in response to detection of a skip-picture flag (*i.e.*, an eliminated skip picture), which is the cause of the “*discontinuity [in the] numbering of the image frames*” for the reasons I discuss in 1[a]. Yagasaki, 15:7-11.

c. Claim 3: “A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.”

109. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for 1[pre] above, during the decoding process, Yagasaki’s decoder 843 sequentially stores decoded pictures of input signal VD3 in field memories 1207 through 1210. Yagasaki, 14:54-67, 19:38-65, Figures 14, 21. A POSITA would have understood field memories 1207 through 1210 to have a small capacity in relation to the length of the video stream because the field memories are used in the video decoding process to store reference frames for motion compensation. As I discuss above in ¶33, reference frames are essential for constructing current frames by adding motion prediction errors to them. The process does not require storing the entire video stream but only a limited number of reference frames that are immediately necessary for decoding successive frames. This temporal locality principle means that the capacity of these memories can be minimized to hold just a few frames at any given time.

110. The technical constraints of the time, including memory speed and the cost of high-capacity memory, would have led a POSITA to understand that field memories were designed to be small. High-capacity memories were more expensive and consumed more power, which was less desirable for most applications. *See* Riley, 9 (recognizing steep hardware costs). Furthermore, the decoding process

benefits from fast access to stored reference frames, a feat easier to achieve with smaller memories. Consequently, given these constraints, products aimed at the mass consumer market would prioritize compactness and energy efficiency over large memory capacity, aligning with consumer expectations for affordable retail prices, portability and longer battery life in their devices.

111. A POSITA would have further understood field memories 1207 through 1210 to comprise the most recently decoded pictures of input signal VD3 because they are specifically intended for temporarily holding decoded reference frames for motion compensation, not for long-term storage of video content. Their role in the decoding process involves quick read and write operations to facilitate real-time video playback. A POSITA would have understood that, given this specific role, there is no need for large-capacity memories since the data turnover is high, with old reference frames quickly replaced by new ones as the video plays.

112. Although I note above in ¶38 that by 2002 it was known to store multiple reference pictures in buffer memory, the existence of long-term reference picture buffers, which are designed to hold select reference frames for extended periods, does not contradict the understanding that field memories 1207 through 1210 are not intended for long-term storage of video content but rather complements it by providing a separate mechanism for storing important reference frames that are needed over longer sequences for effective motion compensation. Such buffers are

not for long-term storage of video content; the data turnover remains high, with old reference frames quickly replaced by new ones as the video proceeds. In addition, I note that Yagasaki is discussed within the context of MPEG (Yagasaki, 2:47-63), and neither MPEG-1 nor MPEG-2 expressly supports the concept of long-term reference pictures.

113. In sum, a POSITA in 2002, considering the principles of video decoding, hardware design constraints, and the nature of motion compensation, would naturally understand field memories 1207 through 1210 to have relatively small capacity, tailored to the immediate needs of the decoding process rather than extensive long-term storage. Furthermore, the well-known need for encoder/decoder synchronization, discussed above in ¶39, would ensure that the encoder and decoder buffers are the same size. Buffers having such characteristics were known as “sliding window” buffers and their benefits were well known by the ’134 patent’s earliest priority date. *See* Wiegand, 8:12-26 (sliding window approach “speed[s] up the motion estimation”), 13:16-23 (sliding window “improves prediction efficiency of...frame selection”); *see also id.*, 11:62-12:4; Riley, 62-63; Long-Term, Abstract, 3; Fast-Search, 619-620.

114. The ’134 patent’s description of “sliding window[ing]” matches the description above. “In sliding windowing, M image frames that were coded last are...[stored] in the buffer memory...in a decoded and reconstructed form, which

allows them to be used as reference images in the coding.” EX-1001, 19:61-64. “As the coding proceeds, the image frame buffering functions on the basis of the FIFO principle (First-In-First-Out).” EX-1001, 19:64-66. For the reasons explained above, a POSITA would have applied this conventional process to Yagasaki’s disclosure, which does not specify the memory management policies for the field memories. In 2002, the sliding window buffer technique was one of a limited number of options available to a POSITA for effectively managing the storage and retrieval of reference pictures in video decoding. Given this backdrop, the choice to implement a sliding window buffer approach would have been an obvious solution to try among the finite options available. The predictability of the solution’s success, due to the mature state of video processing technologies at that time. A POSITA would have had an expectation of success in applying a sliding window reference picture memory management policy to Yagasaki because the sliding window approach was well known and well understood at the time. *See* Wiegand, 8:12-26, 11:62-12:4, 13:16-23; Riley, 62-63; Long-Term, Abstract, 3; Fast-Search, 619-620.

115. Thus, in Yagasaki’s system, as new pictures are decoded from input signal VD3, they replace older pictures in field memories 1207 through 1210, following the FIFO principle. This means that the oldest stored frames in the field memories are the first to be replaced by newer frames, maintaining a continuous cycle of the most recent “M” frames, as described in the ’134 patent specification

regarding “sliding windowing” and as recited in this claim regarding “*sliding window buffering*.”

- d. **Claim 4: “A method according to claim 1, further comprising continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.”**

116. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for 1[b] above, when decoder 843 encounters a skip-picture flag, field memories 1207 through 1210 provide a decoded picture corresponding to a reference picture code associated with the skip-picture flag (“*the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames*”). Yagasaki, 15:7-11, 19:55-63. Yagasaki further teaches that when a skip-picture flag is not encountered, decoder 843 decodes and stores the resulting picture into field memories 1207 through 1210. Yagasaki, 14:54-67, Figure 14. I note that if the skip-picture flag is encountered in successive frames, then for each such frame, Yagasaki’s system operates as I describe for 1[b] above.

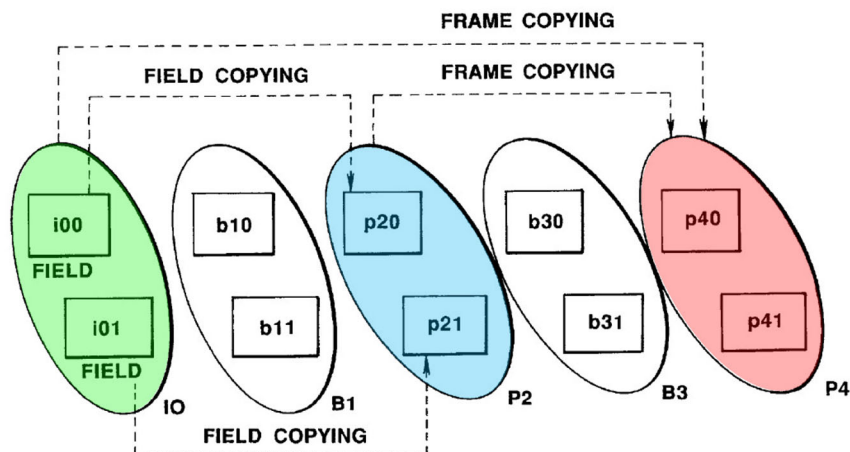
117. As I discuss for 1[a] above, each picture stored in field memories 1207 through 1210 is associated with a temporal_reference code that represents “the display sequence of the picture in a group of pictures (GOP)” and “is generated by

incrementing a counter by one for each consecutive picture in the coder input signal VI[.]” Yagasaki, 19:21-26. As I further discuss in 1[a] above, the temporal_reference is defined in both the MPEG-1 and MPEG-2 standards. See MPEG-1, 21, 27; MPEG-2, 30, 50. As I discuss above in ¶33, the MPEG video coding standards were in wide use as of 2002. These standards were meticulously specified and widely adopted frameworks for video compression that included detailed definitions of various components involved in encoding and decoding processes, including the syntax and semantics of frame numbering. As a result, a POSITA would have understood the temporal_reference defined in the MPEG standards to be the “correct image number” because it adheres to these specific, standardized guidelines, ensuring interoperability and consistency across devices and applications. A POSITA familiar with these standards would recognize the importance of adhering to such specifications for successful encoding and decoding of video streams.

118. Accordingly, when a skip-picture flag is immediately followed by a picture without a skip-picture flag, Yagasaki “*continu[es] the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames,*” as recited by this claim.

- e. **Claim 5: “A method according to claim 1, further comprising entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”**

119. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. Figure 15 of Yagasaki, annotated below, illustrates duplication in the decoder of skip-picture P4 (shaded red) that is a P-picture. Yagasaki, 6:35-37, 15:12-17. Yagasaki explains that “if the frame P4, a P-picture, ...is a skip picture, its reference frame being the frame P2 [shaded blue], the frame P4 is reproduced by copying the frame P2.” Yagasaki, 15:24-27. If, on the other hand, “the reference frame for the frame P4 is the frame I0 [shaded green], the frame P4 is reproduced by copying the frame I0.” Yagasaki, 15:27-29.



Yagasaki, Figure 15 (annotated).

120. As I discuss above in Section VIII.1.b(i), during encoding, Yagasaki teaches “calculat[ing] the pixel-by-pixel difference between the current picture and plural other pictures in the video signal” and “generating a reference picture code

RFC indicating the other picture as the reference picture for reproducing the skip picture” when the current picture and another picture “exhibit[] little or no change.” Yagasaki, 17:51-18:17, Figure 18; *see also id.*, Abstract. A POSITA would have understood that in the Figure 15 example shown above, when reproducing skip-picture P4, the decoder of Yagasaki copies reference frame P2 (or IO) into the buffer position of P4 because, even though it is a skip picture, P4 may subsequently be identified as a reference picture for calculating another picture in the video sequence. For example, as I discuss above in ¶¶66-67, Yagasaki at 17:51-18:17 (and with reference to Figure 18) discusses a process by which picture difference calculating circuit 406 “calculates the pixel-by-pixel difference between the current picture and plural other pictures in the video signal” in order to determine “the reference picture for reproducing the skip picture.” Yagasaki does not indicate that any effort is made to ensure that the reference picture that is identified for reproducing the skip picture is not itself a skip picture. Therefore, a POSITA would have understood that a skip-picture like P4 may be used as a reference picture.

121. Given that P4 can subsequently be used as a reference picture, a POSITA would have understood that in a sliding window buffer like field memories 1207 through 1210 – see claim 3 above – the reference frame P2/IO must be copied into the buffer position of P4 because there will come a time when P4 exists in field memories 1207 through 1210, but P2/IO will not. This is because, as I discuss

above in claim 3, P2/IO is being overwritten by newer frames as the sliding window advances.

122. Finally, I note that the '134 patent does not expressly define “*filler frames*.” See EX-1001, 4:54-59, 20:34-56. A POSITA would have understood that in Yagasaki, the frame copied into the P4 position (from either P2 or IO) as described above is a “filler” frame because it “fills” the gap left by eliminated skip-picture P4.

123. Accordingly, Yagasaki discloses “*entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames*.”

2. Claims 9-12

a. Independent Claim 9

124. Claim 9 is substantively the same as claim 1, except it is written in terms of “[a] video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding.”

- (i) 9[pre]: “A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to”**

125. In my opinion, Yagasaki discloses or renders obvious 9[pre]. As I discuss for 1[pre] above, Yagasaki teaches “*decoding a compressed video sequence*.” Yagasaki also teaches the additional elements of 9[pre]. For example,

Yagasaki discloses decoder 843 “decod[ing] the picture signal in the decoder input signal VD3 to provide the decoding apparatus output signal VO.” Yagasaki, 17:20-22. Yagasaki’s decoder 843 therefore teaches “[a] video decoder for decoding a compressed video sequence.”

126. Yagasaki further discloses decoder 843 “being connected to” field memories 1207 through 1210, which corresponds to the claimed “buffer memory,” as discussed for 1[pre]. Decoder 843 encompasses field memories 1207 to 1210, *i.e.*, field memories 1207 through 1210 are contained within decoder 843. Yagasaki, 19:38-65, Figure 21; *see also id.*, 14:54-15:11, Figure 14. Decoder 843 is therefore “connected to” field memories 1207 to 1210.

(ii) 9[a]: “decode from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;”

127. In my opinion, Yagasaki discloses or renders obvious 9[a]. As I discuss for 1[a] above, Yagasaki teaches “decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames.” Yagasaki also teaches the additional elements of 9[a]. For example, Yagasaki teaches decoder 843 decoding the skip-picture flag from the coded signal. Yagasaki, 14:39-44, Figure 14; *see also id.*, 13:33-35. Yagasaki’s decoder 843 is therefore configured to practice 9[a].

- (iii) **9[b]: “configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and”**

128. In my opinion, Yagasaki discloses or renders obvious 9[b]. As I discuss for 1[b] above, Yagasaki teaches “*configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames.*” Yagasaki also teaches the additional elements of 9[b]. For example, Yagasaki teaches variable-length decoder 1202 of decoder 843 configuring, in response to the skip-picture flag, field memories 1207 through 1210 to provide a picture indicated by a reference picture code as a picture of decoder output signal VO1. Yagasaki, 15:7-11, 19:38-65, Figures 14, 21. Yagasaki’s decoder 843 is therefore configured to practice 9[b].

- (iv) **9[c]: “use the image frames in the buffer memory in the decoding process.”**

129. In my opinion, Yagasaki discloses or renders obvious 9[c]. As I discuss for 1[c] above, Yagasaki teaches “*using the image frames in the buffer memory in the decoding process.*” Yagasaki also teaches the additional elements of 9[c]. For example, Yagasaki teaches that decoder 843 uses the picture indicated by the reference picture code in the decoding process. Yagasaki, 14:54-15:16, Figure 17. Yagasaki’s decoder 843 is therefore configured to practice 9[c].

- b. **Claim 10: “A video decoder according to claim 9, wherein the video decoder is further configured to use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.”**

130. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for claim 2 above in Section IX.A.1.b, the decoded pictures addressed by reference picture code are copied to the decoder output signal to replace skipped pictures that do not exist at decoder 843 (*“us[ed]...in place of image frames that do not exist at a decoder”*). Yagasaki discloses that decoder 843 copies the reference picture to the decoder output signal. Yagasaki, 19:38-65, Figure 21. Yagasaki’s decoder 843 is therefore configured to practice this claim.

- c. **Claim 11: “A video decoder according to claim 9, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.”**

131. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim for the reasons discussed for claim 3.

- d. **Claim 12: “A video decoder according to claim 9, wherein the video decoder is further configured to continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the**

number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.”

132. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for claim 4 above in Section IX.A.1.d, when a skip-picture flag is not encountered immediately after one has been encountered, decoder 843 decodes and stores the resulting picture into field memories 1207 through 1210 from a correct temporal_reference number. Yagasaki, 14:54-67, 19:38-65, Figures 14, 21. Yagasaki’s decoder 843 is therefore configured to practice this claim.

3. Claims 16-19

a. Independent Claim 16

133. Claim 16 is substantively the same as claim 1, except that it is written in terms of “[a] computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.”

- (i) 16[pre]: “A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:”**

134. In my opinion, Yagasaki discloses or renders obvious 16[pre]. As I discuss for 9[pre] above, Yagasaki teaches decoder 843 “for decoding a compressed

video sequence.” A POSITA would have found it obvious to implement decoder 843 as “[a] computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.” Software-implemented decoders were well known in the art before the earliest priority date of the ’134 patent. For example, C-Cube’s CL480PC MPEG-1 audio/video decoder decoded video using a combination of processors. EX-1020, 3. One of the CL480PC’s processors used “microcode” stored in a DRAM that was loaded into on-chip memory. EX-1020, 3. A POSITA would have understood that this microcode was “a computer program product...executable in a data processing device, for decoding a compressed video sequence” and that the DRAM on which it was stored was “a non-transitory computer readable medium.” Moreover, a POSITA would have understood the configuration of C-Cube’s CL480PC to be a common and obvious configuration existing in the art before the earliest priority date of the ’134 patent.

135. A POSITA in 2002 would have further understood a software implementation to be cost effective. *See, e.g.,* Riley, 9 (“Increases in PC processing power are enabling high-performance video CODECs to be implemented in software, causing the cost of the systems to fall rapidly.”). By 2002, a POSITA would have observed that the exponential growth in processing power, as described by Moore’s Law, was not only theoretical but had practical applications in enhancing

software capabilities. This meant that tasks previously requiring dedicated hardware could be efficiently handled by software running on general-purpose computers, significantly reducing costs. Furthermore, unlike hardware solutions, which are often difficult and expensive to upgrade, software implementations can be easily updated or scaled to accommodate new formats or improved algorithms. A POSITA in 2002 would have recognized that this inherent flexibility further contributes to cost-effectiveness of software-based implementations over time, as updates and enhancements can be deployed without the need for physical modifications or replacements.

136. A POSITA would have also found it obvious and been motivated to implement decoder 843 as “*a computer program product*” stored on a personal computer. For example, U.S. Patent No. 7,263,270, filed on November 25, 2000, teaches implementing an MPEG-1 video decoder on software stored in memory and executed using a processor. EX-1012, 6:59-65, 7:53-58. Likewise, in 1995, engineers wrote about an entry-level enterprise workstation that could “achieve[] real-time MPEG-1...video decompression via software running on a general-purpose processor.” EX-1021, 60. A POSITA would have understood that this software was “*a computer program product*” and that this it would have been “*stored on a non-transitory computer readable medium and executable in a data processing device.*” As another example, in 1993, engineers implemented a decoder in 12,000

lines of C code using the X Windowing System (EX-1022, 3) which a POSITA would have understood to comprise a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device.

137. Accordingly, a POSITA would have found it obvious and would have been motivated to implement decoder 843 as “*a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence.*”

- (ii) **16[a]: “a computer program code for decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;”**

138. This limitation is substantively identical to 1[a], except that it is directed to “*a computer program code for*” decoding an indication. As I explain for 1[a] above, Yagasaki teaches “*decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames.*” And as I explain for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using “*a computer program code.*” Yagasaki therefore renders obvious 16[a].

- (iii) **16[b]: “a computer program code for configuring, in response to the indication, the buffer memory to provide a**

number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and”

139. This limitation is substantively identical to 1[b], except that it is directed to “*a computer program code for*” configuring, in response to the indication, the buffer memory. As I explain for 1[b] above, Yagasaki teaches “*configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames.*” And as I explain for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using “*a computer program code.*” Yagasaki therefore renders obvious 16[b].

(iv) 16[c]: “a computer program code for including the image frames in the buffer memory in the decoding process.”

140. This limitation is substantively identical to 1[c], except that it is directed to “*a computer program code for*” including the image frames. As I explain for 1[c] above, Yagasaki teaches “*using the image frames in the buffer memory in the decoding process.*” A POSITA would have understood “*including the image frames...in the decoding process*” to be subsumed within “*using the image frames...in the decoding process*”—and therefore taught by Yagasaki for the reasons I explain for 1[c] above—because “*image frames*” cannot be “*us[ed]...in the decoding process*” if they are not “*includ[ed]...in the decoding process.*” And as I explained for 16[pre] above, it would have been obvious to a POSITA to carry out

the recited step using “*a computer program code.*” Yagasaki therefore renders obvious 16[c].

- b. **Claim 17: “A computer program product according to claim 16, the computer program product further comprising: a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.”**

141. This claim is substantively identical to claim 2, except that it is directed to “[*a*] computer program product...comprising[*] ...a computer program code for*” including the image frames. As I explain for claim 2 above, Yagasaki teaches “*using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.*” And as I explain for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using “*a computer program code.*” Yagasaki therefore renders this claim obvious .

- c. **Claim 18: “A computer program product according to claim 16, the computer program product further comprising a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.”**

142. This claim is substantively identical to claim 3, except that it is directed to “[*a*] computer program product...comprising *a computer program code for*” using the number of image frames in the buffer memory in sliding window buffering

of image frames. As I explain for claim 3 above, Yagasaki teaches “*wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.*” And as I explained for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using “*a computer program code.*” Yagasaki therefore renders this claim obvious.

- d. **Claim 19: “A computer program product according to claim 16, the computer program product further comprising a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.”**

143. This claim is substantively identical to claim 4, except that it is directed to “[a] computer program product...comprising[]...a computer program code for” continuing the entering of the image frames. As I explain for claim 4 above, Yagasaki teaches “*continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.*” And as I explain for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using “*a computer program code.*” Yagasaki therefore renders this claim obvious.

B. Ground 2: Yagasaki and Oliver Render Obvious Dependent Claims 5, 7, 13, 15, 20, and 22

- 1. Claim 5: “A method according to claim 1, further comprising entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”**

144. As I discuss for claim 5 in Ground 1 above, a POSITA would have understood that Yagasaki discloses, in response to the skip-picture flag, copying into field memories 1207 through 1210 (*“entering into the buffer memory”*) the picture indicated by the reference picture code (*“a number of filler frames”*)⁷ to provide a picture (*“a number of image frames”*) to replace the skipped picture (*“corresponding to the at least one discontinuity in the numbering of the image frames”*).

145. In the alternative, the Yagasaki-Oliver combination renders obvious the additional limitations of this claim. Yagasaki teaches that “the coding apparatus output signal VC1 may be broadcast using a suitable broadcast system, cable distribution system, telephone system, etc.” Yagasaki, 21:34-40. A POSITA would have understood that broadcast, cable, and telephone systems circa 2002 were susceptible to transmission errors that led to dropped frames, which reduced performance. *See* Riley, 4 (“A communication network offers a particular quality

⁷ I note that Oliver explicitly teaches that a filler frame can be a “previous or repeated frame.” Oliver, 4:57-59.

of service...to traffic transmitted through the network....Decoded video frames must be presented to the viewer at a constant rate, and if a particular frame is delayed too long then it cannot be displayed and is ‘lost.’ ***Errors and losses have a particularly severe effect on the quality of decoded video.***”); Girod, 1:60-63 (“Transmission errors cause different results of the decoding process at encoder and decoder, thereby leading to differently and inconsistently reconstructed frames.”). A POSITA would have thus been motivated to improve the system of Yagasaki by implementing known techniques to enhance overall performance of the system by mitigating and ameliorating the impact of communication network transmission delays and errors including, inter alia, frames dropped due to bottlenecks at various intermediate network nodes and frames lost due to long transmission delays or unrecoverable transmission errors.

146. Oliver provides such an error-correction technique. Oliver teaches “an improved method and system for handling underrun and overrun of digital data frames”—*e.g.*, MPEG video frames—in a communications system such as a telephone system. Oliver, 2:61-63, 4:47-52, 9:31-32; *see also id.*, Abstract, 2:45-61. Data underrun can occur, explains Oliver, when a decoder is fed with data at a lower speed than the data is being read from the buffer. *See* Oliver, 2:31-3:7, 5:66-6:8. “If the next data frame [in the series of data frames] has not been received, the method [of Oliver] provides a filler frame[.]...to the decoder.” Oliver, 3:30-35.

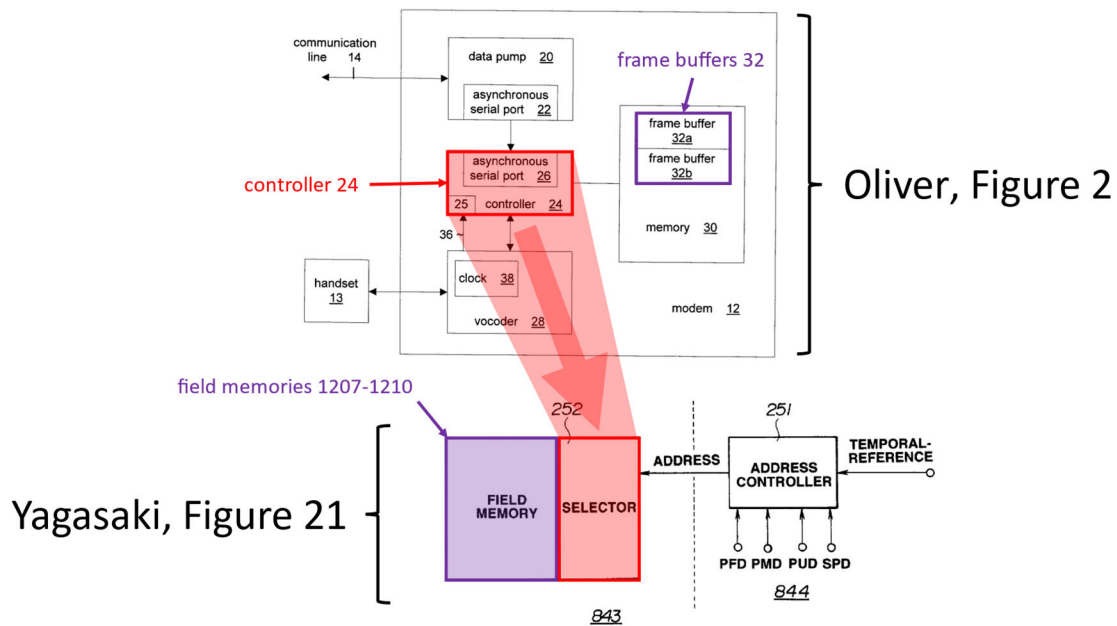
147. Oliver's filler frame is "a frame inserted into a stream of data frames to compensate for lack of reception of one or more frames in the transmission." Oliver, 4:52-55. A POSITA would have understood that incorporating this teaching of Oliver into Yagasaki's decoder would prevent attempted accesses of reference pictures missing due to transmission errors, which could compromise the stability of Yagasaki's system. For example, the decoding process could crash, hang, or terminate unexpectedly upon encountering a reference to a missing or nonexistent reference frame. Alternatively, the decoding process might continue to run, but it may produce incorrect results, exhibit erratic behavior, or cause data corruption. As I discuss above in ¶33, in predictive video coding, frames are often predicted based on previously decoded frames. A frame incorrectly reconstructed due to a missing reference frame can propagate to subsequent frames that are predicted from it, perpetuating distortion in the video across a plurality of frames and degrading the viewing experience. Riley, 4 ("Video coding schemes achieve compression by removing redundant information from the video data, which means that all of the coded data is significant and hence a transmission error is likely to have a significant impact on video quality.').

148. A POSITA would have understood that filler frames, as taught in Oliver, were a known solution to this problem. Oliver's controller 24 provides a filler frame in place of a frame that does not exist in frame buffers 32. Oliver, 3:32-

36, 8:45-55. I note that although the frames stored in Oliver's frame buffers 32 are compressed, a POSITA would have understood that Oliver's teachings are applicable to Yagasaki's field memories 1207 through 1210, which store decoded frames, because the essential role of buffering does not change with the data state (compressed or decoded) and buffer memories are therefore agnostic to the type of data they store. Buffers work at the byte level, generally without concern for the higher-level interpretation of those bytes, which can vary based on the application. This is crucial for ensuring flexibility in handling various types of data and is a foundational concept that a POSITA would have understood. A POSITA would have had no trouble understanding that the buffer-related teachings of Oliver could easily be used to store decompressed data in Yagasaki's system. In the context of an audio application, Oliver explains that the filler frame can be "[a] 'quiet frame'...in which sound is essentially absent," *i.e.*, "a frame in which the volume parameter...has been assigned a zero value." Oliver, 7:53-60, 7:61-63. In Oliver's video application embodiment (Oliver, 4:47-52, 9:31-32), a POSITA would have understood that a filler frame could similarly be generated by assigning all pixel values of the frame to the same value, the choice of which would dictate the color of the frame. For example, the system could easily generate a monochromatic, neutrally colored frame that maintains visual continuity without introducing distracting elements.

149. Implementing the above functionality of controller 24 of Oliver in Yagasaki's system would ensure that if decoder 843 attempts to access a reference picture lost during transmission, a filler frame would be provided in lieu of the missing reference picture. A POSITA would have been motivated to make this combination to mitigate errors during decoding. In the proposed combination, Oliver's filler frame would be stored in field memories 1207 through 1210 and provided to the output video signal in the same manner as Yagasaki's reference pictures.

150. Oliver's Figure 2 and Yagasaki's Figure 21 illustrate the proposed combination below. "[C]ontroller 24 [annotated red]...provid[es] filler frames to the vocoder 28 when a data underrun condition occurs," *i.e.*, when the system attempts to access a picture from frame memories 32 (annotated purple) that does not exist. In the combined system, if decoder 843 attempts to access a missing reference picture from field memories 1207 through 1210 (shaded purple), selector 1206 [also annotated red] provides a filler frame to the decoder output signal. A POSITA would have found it obvious to store the filler frame in field memories 1207 through 1210 so that the filler frame is read out in precisely the same manner as any reference picture in Yagasaki.



Yagasaki, Figure 21 and Oliver, Figure 2 (annotated).

151. A POSITA would have found it obvious to incorporate Oliver's method of swapping a filler frame for a nonexistent buffer frame into Yagasaki's decoder. A POSITA would have first recognized that Yagasaki and Oliver are combinable because both are directed to coding/decoding of data frames, in particular, video frames. Yagasaki, Abstract, 1:8-12; Oliver, 4:47-52, 9:31-32.

152. The Yagasaki-Oliver combination merely amounts to using a known technique (implementing buffer memory management in a decoder using filler frames, as taught by Oliver) to improve a similar known device (Yagasaki's decoder 843) in the same way (providing buffered frames as pictures of the output signal). As I discuss above in ¶39, buffer memory management techniques in which the decoder is notified when the encoder deliberately excludes a frame from the encoded

signal were known by the '134 patent's earliest priority date. Oliver teaches a particular implementation of such a technique that uses filler frames, and that would have been compatible with (and beneficial to) Yagasaki's system.

153. A POSITA would have additionally recognized that the proposed combination would have amounted to no more than applying a known technique (Oliver's use of filler frames for decoder buffer memory management) to a known device ready for improvement (Yagasaki's decoder 843) to yield predictable results (decoder buffer memory having increased error resilience). This approach would enhance error resilience by ensuring continuous operation of the decoder even in the event of missing frames, preventing it from encountering errors that could disrupt video playback.

154. A POSITA would have had a reasonable expectation of success in making the combination. A POSITA could have easily modified Yagasaki's decoder 843 to enable the provision of the filler frames disclosed in Oliver. For example, a filler frame could be stored in field memories 1207 through 1210 for retrieval via readout address. Alternatively, the filler frame could be generated on-the-fly by predictor 1211, allowing for additional flexibility. Either modification would have been trivial for a POSITA at the relevant time because the underlying principles of video decoding and frame handling were well understood, and the techniques for manipulating and storing frames in memory, as well as generating frames using

predictive algorithms, were established practices in the field of digital video processing.

2. **Claim 7: “A method according to claim 1, wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.”**

155. In my opinion, the Yagasaki-Oliver combination renders obvious the additional limitations of this claim. As I discuss for claim 5 above, in the combined system the filler frame is provided in place of frames missing at the decoder. A POSITA would have understood that an efficient way of incorporating this teaching into decoder 843 is with “pointers,” which are identifiers that indicate the locations of data in memory. *See* EX-1019, 514 (defining “pointer” as “[a] data element that indicates the location of another data element” and “[a]n identifier that indicates the location of an item of data”). As a preliminary matter, I note that although the ’134 patent does not expressly define “*memory indications*,” a POSITA would have understood them to be pointers based on their description in the ’134 patent because they indicate the locations of data in buffer memory. *See* EX-1001, 4:60-64, 20:34-56.

156. In 2002, memory pointers were well known to a POSITA because they were fundamental concepts in computer programming and software development, essential for accessing and manipulating memory locations directly. They had been

extensively used in various programming languages, including C and C++, for decades by that time, enabling efficient data structure management and memory allocation tasks critical in software engineering and computer science.

157. Oliver teaches the use of pointers to indicate different buffer memories. Oliver, 6:59-65. This disclosure is applicable to using a pointer to reference a frame within a particular buffer memory because pointers offer a universal method for accessing and managing memory locations, regardless of the specific type of data or memory buffer involved. This flexibility makes pointers a natural choice for efficiently implementing the referencing of any data type within buffer memory, including filler frames, without necessitating physical relocation or duplication of the data itself.

158. Using a pointer to reference a filler frame would reduce memory usage by referencing a pre-defined filler frame stored once in memory, avoiding the need to insert an additional filler frame for each missing frame. By leveraging the inherent capability of pointers to serve as direct links to specific memory addresses, the Yagasaki-Oliver system would have minimized the operational overhead associated with managing multiple instances of the same data, thereby conserving memory resources and optimizing the decoder 843's performance.

159. A POSITA would have further recognized that the filler frame could be stored independently of the sliding window, ensuring it remains available and is not

discarded when new frames enter the buffer. The concerns I elucidate in claim 3 above regarding data turnover would therefore not be an issue. This independent storage – facilitated by pointers – not only guarantees the availability of the filler frame regardless of the sliding window’s state but also enhances the system’s flexibility in handling frame loss scenarios. It allows for a dynamic and responsive approach to error recovery wherein decoder 843 can seamlessly substitute missing frames with the filler frame to maintain video playback continuity without the risk of the filler frame being prematurely removed or overwritten.

160. A POSITA would have understood this operation to be compatible with Yagasaki’s system. Yagasaki teaches configuring field memories 1207 through 1210 to provide pictures “indicated by each readout address.” Yagasaki, 19:44-65, Figure 21. A POSITA would have understood Yagasaki’s readout addresses to be “*memory indications*” because they are references to specific locations in buffer memory where data is stored to be read out for processing or display. Yagasaki, 19:55-65, Figure 21; *see also id.*, 14:27-31, Figure 12; EX-1001, 4:60-64, 20:34-56. Yagasaki’s system would therefore “read out” the filler frame from field memories 1207 through 1210 in the same manner as is done for reference frames: by using the readout address to direct decoder 843 to a specific data location in buffer memory.

3. Claim 13: “A video decoder according to claim 9, wherein the video decoder is further configured to enter into the buffer

memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”

161. The Yagasaki-Oliver combination renders obvious the additional limitations of this claim for the reasons discussed for claim 5, Ground 2. As I explain in claim 5 above, in the proposed combination, Yagasaki’s decoder 843 would perform this step.

4. **Claim 15: “A video decoder according to claim 9, wherein the video decoder is further configured to configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.”**

162. In my opinion, Yagasaki discloses or renders obvious the additional limitations of this claim. As I discuss for claim 7, Ground 2 above, variable-length decoder 1202 of Yagasaki’s decoder 843 configures, in response to the skip-picture flag, field memories 1207 through 1210 to provide a picture indicated by a reference picture code as a picture of decoder output signal VO1 through the use of address controller 251 in field sequence exchanging and scan converting circuit 844. Yagasaki, 15:7-11, 19:38-65, Figures 14, 21. In the combined Yagasaki-Oliver system, Yagasaki’s decoder 843 is therefore configured to practice this claim.

5. **Claim 20: “A computer program product according to claim 16, the computer program product further comprising: a computer program code for entering into the buffer memory a number of**

filler frames corresponding to the at least one discontinuity in the numbering of the image frames.”

163. This claim is substantively identical to claim 5, except that it is directed to “[a] computer program product...comprising[] a computer program code for” entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames. As I explain for claim 5, Ground 2 above, the combined Yagasaki-Oliver system teaches “*entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.*” And as explained for 16[pre], a POSITA would have recognized that the “*entering into*” would be carried out by “*a computer program code*” in Yagasaki. Oliver similarly discloses that “*entering into*” would be carried out by “[a] computer-readable storage medium comprising program instructions.” Oliver, claim 22. The combined Yagasaki-Oliver system therefore renders this claim obvious.

- 6. Claim 22: “A computer program product according to claim 16, the computer program product further comprising: a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.”**

164. This claim is substantively identical to claim 7, except that it is directed to “[a] computer program product...comprising[] ...a computer program code for” configuring the buffer memory to provide a number of image frames by memory

indications. As I explain for claim 7, Ground 2 above, Yagasaki teaches *“configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.”* And as I explain for 16[pre] above, it would have been obvious to a POSITA to carry out the recited step using *“a computer program code.”* Oliver similarly discloses that *“entering into”* would be carried out by *“[a] computer-readable storage medium comprising program instructions.”* Oliver, claim 22. The combined Yagasaki-Oliver system therefore renders this claim obvious.

C. Ground 3: Yagasaki, Oliver, and Lyon Render Obvious Dependent Claims 6, 14, and 21

1. Claim 6: “A method according to claim 5, wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.”

165. In my opinion, the Yagasaki-Oliver-Lyon combination renders obvious the additional limitations of this claim. As I discuss for claim 5, Ground 2 above in Section IX.B.1, the filler frames of Oliver are provided in lieu of reference pictures for purposes of maintaining the stability, security, and performance of Yagasaki’s system. The filler frames are subsequently provided to the output video signal, or alternatively used as reference pictures to temporally predict other frames in the video sequence, in the same manner as Yagasaki’s reference pictures. *See* Yagasaki,

15:54-67, Figure 14. As I also discuss above in the context of claim 5 in Ground 2, a POSITA reading Oliver would have found it obvious to use a filler frame in which all pixel values have been assigned the same value, in view of Oliver's audio embodiment and Oliver's stated applicability to video frames. Oliver, 4:47-52, 7:53-63, 9:31-32. A POSITA would have known that using a frame that does not belong to the original video sequence as a reference frame to predictively code other frames could cause visual distortions and decrease video quality.

166. Lyon provides a solution to this problem. As discussed above in Section VIII.3, Lyon teaches "ensur[ing] that any 'artificial' intermediate video frames...are identified as such and processed accordingly" so that such frames are "not treated in the same manner as a normal decoded frame." Lyon, 6, 11. Specifically, Lyon describes a "line interpolation" algorithm that converts the intermediate video frames between formats and explains that "since the line interpolation algorithm assumes both fields of the intermediate video frame to represent the same image, the application of this algorithm to a pair of fields which are adjacent only through the coincidence of editing, may produce visually disturbing artefacts." Lyon, 5-6.

167. A POSITA would have recognized that using filler frames that do not belong to the original video sequence as reference frames would similarly produce "visually disturbing artefacts" in the predicted frames because motion prediction,

like the line interpolation algorithm described in Lyon, relies on the assumption that adjacent frames are related. A POSITA would have thus been motivated to identify the filler frames in the Yagasaki-Oliver system because the filler frames, like the artificial frames in Lyon, should “not [be] treated in the same manner as a normal decoded frame” because they are not part of the original video sequence and therefore do not carry meaningful video information for purposes of performing motion prediction. Lyon, 11.

168. Lyon thus supplements and is entirely consistent with the disclosure in Oliver, and a POSITA would have found it obvious to incorporate Lyon’s teaching of identifying artificial video frames into the Yagasaki-Oliver system. Like Yagasaki and Oliver, Lyon is directed to coding/decoding of video data. Lyon, 1-2; Yagasaki, Abstract, 1:8-12; Oliver, 4:47-52, 9:31-32. The relied-on teaching of Lyon would operate the same in the combined system as it does alone by indicating to the decoder when a frame does not belong to the original video sequence and should not be used for subsequent processing. Lyon, 5-6, 11.

169. Lyon provides an express motivation that would have led a POSITA to modify the Yagasaki-Oliver system: avoiding the “visually disturbing artefacts” that result from the inclusion of inappropriate frames. Lyon, 6. As discussed for claim 5 above in Section IX.B.1, it would have been obvious in the Yagasaki-Oliver combination to store the filler frames in field memories 1207 through 1210 and

provide them via selector 1206 as pictures of the output video signal in the same manner as Yagasaki's reference pictures. Differentiating filler frames from video sequence frames would allow the decoder to ensure filler frames are not "treated in the same manner as a normal decoded frame" (Lyon, 11), resulting in a higher quality video signal because the filler frames are not being used as reference frames and are therefore not introducing visual artifacts not present in the original sequence.

170. A POSITA would have understood that incorporating Lyon's teaching would realize this benefit (i.e., higher quality output video) in the Yagasaki-Oliver system because Lyon differentiates "artificial" frames for the same underlying reason that a POSITA would have been motivated to differentiate the filler frames in the Yagasaki-Oliver system: because using the artificial frame in Lyon's line-interpolation algorithm would result in lower quality output video. Lyon, 5-6. The same result would occur if the filler frame was used as input to Yagasaki's motion prediction algorithm. A POSITA would recognize this similarity. Accordingly, Lyon's rationale for differentiating artificial frames (i.e., to exclude them from being input to the line interpolation algorithm described in Lyon) is analogous to the reason for incorporating that teaching into the Yagasaki-Oliver combination, namely, to exclude filler frames from being input to the motion prediction algorithm. A POSITA would further conclude that the same type of "visually disturbing artefacts"

that are avoided in Lyon would also be avoided in the resulting combined system. Lyon, 5-6, 11; Yagasaki, 1:15-55, 12:6-33; Oliver, 4:47-52, 9:31-32.

171. A POSITA would have further understood that Yagasaki already includes much of the functionality needed to incorporate Lyon's teaching. Specifically, Lyon's filler frames could easily be identified using the same types of flags and codes that Yagasaki's decoder 843 already uses to designate different types of pictures in the coded video signal, such as reference pictures (RFD) and skip-pictures (SFD). *See* Yagasaki, 18:28-32.

172. A POSITA would have had a reasonable expectation of success in making the proposed combination. As I previously explained, Yagasaki teaches the use of identifiers to designate different types of pictures in the coded video signal, for example, reference pictures and skip-pictures. Yagasaki, 18:28-32. Modifying the Yagasaki-Oliver combination to identify filler frames in the same manner would have been a trivial modification within the skill of a POSITA at the relevant time.

2. Claim 14: "A video decoder according to claim 13, wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence."

173. The Yagasaki-Oliver-Lyon combination renders obvious the additional limitations of this claim for the reasons discussed for claim 6. As I explain in claim 6 above, in the proposed combination, Yagasaki's decoder 843 would perform this step.

3. **Claim 21: “A computer program product according to claim 20, the computer program product further comprising: a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.”**

174. This claim is substantively identical to claim 6, except that it is directed to “[a] computer program product...comprising[] a computer program code for” designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence. As explained for claim 6, Yagasaki teaches “wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.” And as explained for 16[pre], a POSITA would have recognized that the “designating” would be carried out by “a computer program code.” The combined Yagasaki-Oliver-Lyon system therefore renders this claim obvious.

D. Ground 4: Yagasaki and Ran Render Obvious Dependent Claim 8

1. **Claim 8: “A method according to claim 2, further comprising detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.”**

175. The Yagasaki-Ran combination teaches the additional limitations of this claim. As a preliminary matter, I note that based on the language and structure of claim 8, a POSITA would have interpreted the term “reference” to mean “referred to” and not “reference” as in a “reference picture” for use in motion compensation

prediction. Although the term “reference” holds a specific meaning in the context of motion compensated prediction, it is typically paired with another term like “picture” or “frame” to clarify its use, which is how the term is consistently used in the ’134 patent. Accordingly, in my opinion a POSITA would have understood “*a reference in the decoding process to an image frame*” to broadly mean referring to an image frame in the decoding process.

176. As I discuss above in Section VIII.4, Ran teaches sending a request for retransmission of a reference picture upon detection that the reference picture is missing at the decoder. Ran, 3:46-48, 7:59-8:9; 9:58-10:46. Yagasaki’s system, on the other hand, relies on reference pictures in the decoder 843 but does not account for the possibility of a missing reference picture, e.g., lost during transmission to the decoder. Yagasaki, 14:54-57, 14:60-67. In the combined system, the “*detecting [an] error*” would occur when decoder 843 tries to retrieve a missing reference frame from field memories 1207 to 1210. The reference picture code associated with the missing reference picture is “*a reference in the decoding process to an image frame,*” that in turn corresponds to the skipped picture (“*corresponding to the at least one discontinuity in the numbering of image frames*”).

177. A POSITA would have understood the necessity of handling transmission errors in Yagasaki’s system. Yagasaki explains that “the coding apparatus output signal VC1 may be broadcast using a suitable broadcast system,

cable distribution system, telephone system, etc.” Yagasaki, 21:37-40. A POSITA would have understood that broadcast, cable, and telephone systems as of the earliest priority date of the '134 patent were prone to various types of interference and signal degradation. For instance, broadcast signals could be affected by atmospheric conditions, physical obstructions, or electromagnetic interference. Cable systems, while being less susceptible to external interference compared to airwave broadcasts, may nonetheless suffer from signal degradation over long distances or due to poor infrastructure. Telephone lines, especially those used for DSL Internet, were limited by the quality and length of the copper wire, leading to potential loss of signal integrity.

178. Furthermore, by 2002, the transition from analog to digital broadcasting was underway. Although digital video offers superior quality over analog in terms of resolution and clarity, it is susceptible to packet loss and errors. The analog/digital transition increased the need for robust error-handling mechanisms to maintain the integrity of digital video content. Accordingly, a POSITA would have been motivated to combine Ran's solution of requesting retransmission of reference pictures missing at the decoder with the system of Yagasaki because such teachings would have improved Yagasaki's system by making it more error resilient. Specifically, Ran's teachings would have permitted Yagasaki's decoder 843 to

attempt proper decoding even when a reference picture is missing at the decoder due to a transmission error.

179. Ran's retransmission technique mirrors the '134 patent's description of initiating "error correction actions" during the decoding process when "a reference to an image number is detected which is then found to indicate a filler frame located in the buffer." EX-1001, 20:51-53. As a threshold matter, a POSITA would have understood that initiating an error correction action requires first detecting the error that initiated the action. Filler frames, the '134 patent explains, "correspond[] to...missing image numbers," *i.e.*, reference images that are missing from the buffer. EX-1001, 20:35-42. The error correction action described in the '134 patent can include "asking the encoder to **re-transmit the reference image** in question." EX-1001, 20:51-56. In other words, the '134 patent's "error correction action[]" involves requesting re-transmission of a reference image when the reference image is missing from the decoder's buffer memory. That is precisely what Ran teaches. Ran, 3:3-9, 6:3-32, 9:58-10:46.

180. Ran provides an express teaching, suggestion, and motivation that would have led a POSITA to incorporate its teachings into Yagasaki's system. Ran explains that "errors in video data...result in video displayed with flawed areas containing incorrect colors and/or intensities[, which] can be very noticeable and distracting." Ran, 2:3-6. Ran notes further that such errors are especially

problematic in “video compression processes [that] use...‘inter’ coding”—like Yagasaki’s system (Yagasaki, 14:34-67, Figure 14)—because “new inter coded data builds new frames on frames containing flaws, and the receiver may preserve and repeat flaws caused by an uncorrectable error.” Ran, 2:7-17. As I discuss above in ¶¶32-39, this aspect of inter-coding (potential for error propagation) was well known by the ’134 patent’s earliest priority date. Ran’s retransmission technique “eliminates the flawed areas...and permits use of inter coding.” Ran, 2:21-22. A POSITA would have recognized that incorporating Ran’s retransmission technique into Yagasaki’s system would have yielded the expected, predictable result of improved inter coding.

181. A POSITA would have recognized that the Yagasaki-Ran combination would have amounted to no more than applying a known technique (Ran’s retransmission technique for requesting missing reference frames) to a known device ready for improvement (Yagasaki’s decoder 843) to yield predictable results (decoder having increased resilience to transmission errors). The principles of error recovery and frame retransmission were already established in the field of digital communication and video processing, and applying these techniques to enhance the robustness of video decoders was a logical step within the expertise of a POSITA circa 2002. *See* Wiegand, 2:55-3:39; Sullivan, 75, 78; Long-Term, 1.

182. A POSITA would have had a reasonable expectation of success in combining the retransmission technique of Ran with Yagasaki's decoding apparatus 850. Ran explains that its retransmission technique "can be implemented using a digital signal processor (DSP) which executes decoding and encoding software" or as "software executed by the main CPU of [a personal] computer." Ran, 11:33-52. A POSITA would have recognized a DSP and software as conventional and known in the field. *See* Riley, 7-9; Wiegand, 6:20-23. Thus, a POSITA would have found it simple to implement the retransmission technique of Ran in the system of Yagasaki.

APPENDIX A: MATERIALS CONSIDERED

Exhibit	Reference
1001	U.S. Patent No. 8,204,134 (“the ’134 patent”)
1002	File History of the ’134 patent
1003	Declaration of Dr. Joseph P. Havlicek (“Havlicek”)
1004	Curriculum Vitae of Dr. Joseph P. Havlicek
1005	U.S. Patent No. 5,835,672 (“Yagasaki”)
1006	U.S. Patent No. 6,064,693 (“Oliver”)
1007	International Publication No. WO 91/06182 (“Lyon”)
1008	U.S. Patent No. 5,768,533 (“Ran”)
1009	U.S. Patent No. 7,103,669 (“Apostolopoulos”)
1010	U.S. Patent No. 6,807,231 (“Wiegand”)
1011	U.S. Patent No. 6,205,177 (“Girod”)
1012	U.S. Patent No. 7,263,270 (“Lapstun”)
1013	[Reserved]
1014	Martyn J. Riley and Iain E.G. Richardson, <i>Digital Video Communications</i> , Artech House, Inc. (1997) (“Riley”)
1015	Gary J. Sullivan and Thomas Wiegand, <i>Rate-Distortion Optimization for Video Compression</i> , IEEE Signal Processing Magazine, pp. 74-90 (November 1998) (“Sullivan”)
1016	[Reserved]
1017	Thomas Wiegand, Xiaozheng Zhang, and Bernd Girod, <i>Motion-Compensating Long-Term Memory Prediction</i> , Proceedings of International Conference on Image Processing, Vol. 2 (IEEE, 1997) (“Long-Term”)
1018	Thomas Wiegand, Bo Lincoln, and Bernd Girod, <i>Fast Search For Long-Term Memory Motion-Compensated Prediction</i> , Proceedings 1998 International Conference on Image Processing, ICIP98, Cat. No. 98CB36269 (IEEE, 1998) (“Fast-Search”)

Exhibit	Reference
1019	<i>IBM Dictionary of Computing</i> , Berkeley: McGraw-Hill Osborne Media (1994)
1020	CL480PC MPEG-1 System Decoder Specification, <i>available at</i> https://web.archive.org/web/19970128081644/http://www.c-cube.com/prdctlst/cl480pc.pdf
1021	Ruby B. Lee et al., <i>Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7100LC Processors</i> , Hewlett-Packard Journal (Apr. 1995)
1022	Ketan Patel, et al., <i>Performance of a Software MPEG Video Decoder</i> , Multimedia '93: Proceedings of the First ACM Int'l Conf. on Multimedia, August 2-6, 1993
1023	[Reserved]
1024	Rangarajan Aravind, Glenn L. Cash, Donald L. Duttweiler, Hsueh-Ming Hang, Barry G. Haskell, and Atul Puri, <i>Image and Video Coding Standards</i> , AT&T Technical Journal, (January/February 1993) ("Aravind")
1025	[Reserved]
1026	ITU-T H.262 (07/95) Transmission of Non-Telephone Signals, Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video (July 1995) ("MPEG-2")
1027	[Reserved]
1028	U.S. Patent No. 5,426,464 ("Casavant")
1029	[Reserved]
1030	[Reserved]
1031	First Amended Complaint, <i>Nokia Corp. v. Amazon.com, Inc.</i> , No. 1:23-cv-01232 (D. Del.), ECF No. 5.
1032	Complaint, <i>Nokia Techs. Oy v. HP Inc.</i> , No. 1:23-cv-01237 (D. Del.), ECF No. 1.

Exhibit	Reference
1033	ISO/IEC 11172-2 (1st ed.) Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video ("MPEG-1")

APPENDIX B: LISTING OF CHALLENGED CLAIMS

Reference	Claim Limitation
Claim 1	
1[pre]	A method for decoding a compressed video sequence, wherein image frames are entered into a buffer memory in connection with the decoding, the method comprising:
1[a]	decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;
1[b]	configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and
1[c]	using the image frames in the buffer memory in the decoding process.
Claim 2	
2	A method according to claim 1, further comprising using the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.
Claim 3	
3	A method according to claim 1, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.
Claim 4	
4	A method according to claim 1, further comprising continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.

Reference	Claim Limitation
Claim 5	
5	A method according to claim 1, further comprising entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.
Claim 6	
6	A method according to claim 5, wherein said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.
Claim 7	
7	A method according to claim 1, wherein configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames is carried out by memory indications, whereby no data is entered into the buffer memory.
Claim 8	
8	A method according to claim 2, further comprising detecting an error in the compressed video sequence on a basis of a reference in the decoding process to an image frame corresponding to the at least one discontinuity in the numbering of image frames.
Claim 9	
9[pre]	A video decoder for decoding a compressed video sequence, the video decoder being connected to a buffer memory wherein image frames are entered in connection with the decoding, the video decoder being configured to
9[a]	decode from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;

Reference	Claim Limitation
9[b]	configure, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and
9[c]	use the image frames in the buffer memory in the decoding process.
Claim 10	
10	A video decoder according to claim 9, wherein the video decoder is further configured to use the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.
Claim 11	
11	A video decoder according to claim 9, wherein the number of image frames in the buffer memory are used in sliding window buffering of image frames.
Claim 12	
12	A video decoder according to claim 9, wherein the video decoder is further configured to continue the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.
Claim 13	
13	A video decoder according to claim 9, wherein the video decoder is further configured to enter into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.
Claim 14	
14	A video decoder according to claim 13, wherein

Reference	Claim Limitation
	said filler frames are designated by an identifier indicating that the filler frames do not belong to the compressed video sequence.
Claim 15	
15	A video decoder according to claim 9, wherein the video decoder is further configured to configure the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames by memory indications, whereby no data is entered into the buffer memory.
Claim 16	
16[pre]	A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, image frames of said video sequence being entered into a buffer memory in connection with the decoding, the computer program product comprising:
16[a]	a computer program code for decoding from the compressed video sequence an indication informing an intentional discontinuity of numbering of the image frames;
16[b]	a computer program code for configuring, in response to the indication, the buffer memory to provide a number of image frames corresponding to at least one discontinuity in the numbering of the image frames; and
16[c]	a computer program code for including the image frames in the buffer memory in the decoding process.
Claim 17	
17	A computer program product according to claim 16, the computer program product further comprising: a computer program code for including the number of image frames in the buffer memory in place of image frames that do not exist at a decoder due to the at least one discontinuity in the numbering of the image frames.

Reference	Claim Limitation
Claim 18	
18	A computer program product according to claim 16, the computer program product further comprising a computer program code for using the number of image frames in the buffer memory in sliding window buffering of image frames.
Claim 19	
19	A computer program product according to claim 16, the computer program product further comprising a computer program code for continuing the entering of the image frames comprised by the compressed video sequence into the buffer memory from a correct image frame number after the buffer memory has been configured to provide the number of image frames corresponding to the at least one discontinuity in the numbering of the image frames.
Claim 20	
20	A computer program product according to claim 16, the computer program product further comprising: a computer program code for entering into the buffer memory a number of filler frames corresponding to the at least one discontinuity in the numbering of the image frames.
Claim 21	
21	A computer program product according to claim 20, the computer program product further comprising: a computer program code for designating said filler frames by an identifier indicating that the filler frames do not belong to the compressed video sequence.
Claim 22	
22	A computer program product according to claim 16, the computer program product further comprising: a computer program code for configuring the buffer memory to provide a number of image frames corresponding to the at least one discontinuity in the numbering of the image frames

Reference	Claim Limitation
	by memory indications, whereby no data is entered into the buffer memory.

166038037.12