

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google LLC,
Petitioner

v.

Withrow Networks Inc.
Patent Owner

IPR2025-00775

U.S. Patent No. 10,771,849 B2
Issue Date: September 8, 2020

Title: MULTIMEDIA SYSTEM FOR MOBILE CLIENT PLATFORMS

DECLARATION OF NATHANIEL POLISH, Ph.D.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND QUALIFICATIONS	6
A. Qualifications and Experience	6
B. Materials Considered.....	10
II. PERSON OF ORDINARY SKILL IN THE ART	15
III. STATEMENT OF LEGAL PRINCIPLES.....	18
A. Claim Construction	18
B. Obviousness.....	20
IV. TECHNOLOGY BACKGROUND.....	25
A. Streaming Media over the Internet.....	25
B. Internet Communication Protocols	29
C. Digital Audio and Video	32
D. Digital Audio and Video Compression.....	33
V. THE '849 PATENT	38
A. Overview of the Specification.....	38
B. Prosecution History [EX 1008].....	42
C. The Challenged Claims	45
VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS	46
A. Brief Summary of Prior Art	47
1. Carmel [EX1003].....	47
2. Mattavelli [EX1004]	54
3. Wang [EX1007]	58
4. Wolters [EX1041]	61
5. Guo [EX1051].....	62
6. Yu [EX1061].....	65

TABLE OF CONTENTS

(continued)

	Page
7. Forecast [EX1016]	67
B. Ground 1: Claims 1-9 Are Obvious Over Carmel in View of Mattavelli.....	69
1. Claim 1 (Independent)	69
(a) “A method comprising:” (claim 1, preamble).....	70
(b) “receiving audio and video segments encoded in a digital encoding format and with an encoding rate;” (claim 1[a.i])	71
(c) “wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;” (claim 1[a.ii]).....	82
(d) “requesting by a multimedia player, transmission of said multimedia objects;” (claim 1[b.i])	100
(e) “wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;” (claim 1[b.ii])	107
(f) “playing back received multimedia objects by the multimedia player,” (claim 1[c.i])	120
(g) “wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,” (claim 1[c.ii])	123
(h) “autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,” (claim 1[c.iii])	138
(i) “and by utilizing optimized decoding processes to maintain quality playback.” (claim 1[c.iv])	153

TABLE OF CONTENTS

(continued)

	Page
(1) Manner and rationales for combining the teachings of Carmel and Mattavelli	162
2. Claim 2: “The method of claim 1, wherein the transmission of the multimedia objects is carried out by protocols that can serve files in a directory.”.....	178
3. Claim 3: “The method of claim 1, wherein the multimedia objects are created from a live multimedia stream that is transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth and converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients.”	183
4. Claim 4: “The method of claim 1, wherein a multimedia file is scanned for an I-frame, and a multimedia segment is split at the I-frame to create the video segment, together with a separate audio segment.”	197
5. Claim 5: “The method of claim 1, wherein multimedia objects are split from multiple MPEG4 composite layer streams by scanning time intervals and splitting the multimedia objects into video multimedia objects at next I-frames.”	205
6. Claim 6 (Independent)	210
7. Claim 7 (Independent)	214
8. Claim 8: “The method of claim 1, wherein the multimedia objects are created from existing multimedia files that have been transcoded into optimal digital formats at an optimal encoding rates reflecting available network bandwidth, and converting them into audio and video segments by splitting the encoded files into intervals for access by wireless clients.”	219

TABLE OF CONTENTS

(continued)

	Page
9. Claim 9: “The method according to claim 1, wherein the object parameters and supplied identification of the first player received multimedia object define a sequence of similar multimedia objects and configure the player accordingly, using object parameters and video decoding optimization, such that fluidity, video and audio quality are maintained.”	221
C. Ground 2: Claims 1-9 Are Obvious Over Carmel in View of Wang.....	225
(1) Manner and rationales for combining the teachings of Carmel and Wang	21938
D. Ground 3: Claims 1-9 Are Obvious Over Carmel in View of Wang and Wolters	246
E. Ground 4: Claims 1-9 are Obvious Over Carmel in View Wang, Wolters, Guo, Yu, and Forecast	249
F. Ground 5: Claims 1-9 are Obvious Over Carmel in View Mattavelli, Guo, Yu, and Forecast	261
VII. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS.....	261
VIII. CONCLUSION.....	262

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

I, Nathaniel Polish, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I have been retained by Google LLC (“Petitioner”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“USPTO”).

2. I am being compensated at a rate of \$750/hour for my services in this proceeding, which is my regular and customary rate.

3. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony or this declaration, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

A. Qualifications and Experience

4. My Curriculum Vitae is attached as Exhibit 1064.

5. I reside in New York City, NY.

6. I am an independent consultant. All of my opinions stated in this declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in streaming media, signal processing, computer network design, and video processing.

7. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. A copy of my current

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

curriculum vitae, which details my education and professional and academic experience, is included as EX1064 in this proceeding. The following provides an overview of some of my experience that is relevant to the matters set forth in this declaration.

8. I expect to testify regarding my background, qualifications, and experience relevant to the issues in this litigation. I have a Ph.D. in Computer Science from Columbia University. I hold the following four degrees from Columbia, spanning the years 1980 to 1993:

- Ph.D. in Computer Science, May 1993, Thesis: Mixed Distance Measures for the Optimization of Concatenative Vocabularies in Speech Synthesis;
- M.Phil. in Computer Science, December 1989;
- M.S. in Computer Science, December 1987; and
- B.A. in Physics, Columbia College, May 1984.

9. For over thirty-five years, I have run a computer technology development firm that I co-founded, called Daedalus Technology Group. My primary business activity is the development of computer-related products. This activity involves understanding the business objectives of customers, designing products to suit their needs, and supervising the building, testing, and deployment of these products. I develop hardware and software as well as supervise others who do

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

so.

10. Also, from time to time I found other companies in order to pursue particular product opportunities. I develop and ultimately sell these companies. Most of my business activity, however, is as a consulting product developer. From time to time, I have also served as an expert witness on computer and software related cases. I am a named inventor on ten United States patents and am a member of several professional societies, including the IEEE and ACM.

11. I have extensive experience in several areas relevant to this case. My doctoral work was in the area of computer speech. As part of that work I developed signal processing systems and algorithms to filter and manipulate sound, especially speech. That work gave me strong practical insight into such topics as the management and transport of streaming media.

12. Much of my professional work over the past thirty-five years has been as a computer technology product developer. Several of those products involved technologies closely related to the topics in the patent at issue in the current matter, including streaming audio and video systems and associated networking. One product in particular was for a company called Instant Video Technologies (IVT) in the mid-1990s. This company later changed its name to Burst.com. The products that I developed for IVT included a family of digital media servers to deliver high

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

quality media over a variety of network connections including Internet connections.

The work resulted in a patent (U.S. Patent No. 5,963,202) on our server technology.

13. In addition to this work for IVT, my firm also designed and developed, under my supervision, a number of products or systems for delivering digital audio and performing associated signal processing. Between 1982-1991, the firm designed and built, under my supervision, all parts of four generations of interactive voice editing systems for The Dun & Bradstreet Corporation. These editing systems produced output used in an extensive, high quality speech synthesis system that was part of a larger information delivery system. The systems were built on networks of computers in a variety of programming languages and systems. The systems involved interactive user interfaces, screen windows, signal processing, and voice file systems.

14. In 2000, I also supervised the development, design, and deployment of an industrial strength interactive voice response (IVR) system that allows users to access streaming media via their cell phones. This system supported up to 96 streams on a single chassis and was designed for a company called Savos. Under my supervision, the firm developed for this system the web-based content management system and interface for customizable user preferences.

15. In addition to the products and systems described above, I have also

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

developed many other systems that utilize various networking techniques for moving data around the Internet in a timely way taking into account spontaneous changes in network capacity and usage.

16. Through these and other experiences, I am very familiar with the technologies at issue in this matter.

B. Materials Considered

17. The analysis that I provide in this Declaration is based on my education, research, and experience, as well as the documents I have considered. In forming my opinions, I have read and considered U.S. Patent No. 10,771,849 B2 (“’849 patent”) [EX1001] and its prosecution history [EX1008]. I have cited to or considered the following documents in my analysis below:

Exhibit No.	Description of Document
1001	U.S. Patent No. 10,771,849 B2 to Mark Sinclair Krebs (filed Nov. 5, 2018, issued Sept. 8, 2020)
1003	U.S. Patent No. 6,389,473 to Sharon Carmel et al. (filed Mar. 24, 1999, issued May 14, 2002) (“Carmel”)
1004	Marco Mattavelli, Sylvain Brunetton, <i>Computational Graceful Degradation for Video Decoding</i> , Signal Processing IX Theories and Applications, Proceedings of Eusipco-98 Ninth European Signal Processing Conference (1998) (“Mattavelli”)
1005	Excerpts from John Watkinson, The MPEG Handbook, MPEG-1, MPEG-2, MPEG-4 (2001) (“Watkinson”)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Exhibit No.	Description of Document
1006	Excerpts from David Gourley et al., HTTP: The Definitive Guide (2002) (“Gourley et al.”)
1007	Shih-Hao Wang et al., <i>A Platform-Based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining</i> , Proc. of the 2003 Joint Conference of the Fourth International Conference on Information, Communications & Signal Processing and Fourth Pacific-Rim Conference on Multimedia Vol. 1, (ICICS-PCM 2003) (“Wang”)
1008	Prosecution History for U.S. Patent No. 10,771,849
1009	U.S. Patent No. 5,841,432 to Sharon Carmel et al. (filed Feb. 9, 1996, issued Nov. 24, 1998) (“Carmel ’432”)
1010	U.S. Patent No. 6,397,230 to Sharon Carmel et al. (filed Aug. 27, 1997, issued May 28, 2002) (“Carmel ’230”)
1011	Microsoft Computer Dictionary (5th Ed. 2002) (excerpted)
1012	U.S. Patent No. 7,359,004 to Keman Yu et al. (filed May 23, 2003, published as U.S. Patent Publication No. 2004/0233844 on Nov. 25, 2004, issued Apr. 15, 2008) (“Yu ’004”)
1013	U.S. Patent Application Publication No. US 2002/0131768 to Robert S. Gammenthaler (filed Mar. 19, 2001, published Sept. 19, 2002) (“Gammenthaler”)
1014	A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC), Tektronix (2000)
1015	Archived Webpage (Internet Archive Wayback Machine) of A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC), Tektronix (June 16, 2002)
1016	U.S. Patent No. 7,096,481 to John Forecast et al. (filed Mar. 31, 2000, issued Aug. 22, 2006) (“Forecast”)
1017	U.S. Patent No. 6,631,163 to Shaomin Peng (filed Nov. 14, 2000, issued Oct. 7, 2003) (“Peng”)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Exhibit No.	Description of Document
1018	Stephan Mietens et al., <i>New Complexity Scalable MPEG Encoding Techniques for Mobile Applications</i> , EURASIP Journal on Applied Signal Processing 2004(2):236-252 (2004) (“Mietens”)
1019	Joseph Y. Hui et al., <i>Client-Server Synchronization and Buffering for Variable Rate Multimedia Retrievals</i> , 14 IEEE Journal on Selected Areas in Communications 226 (Jan. 1996) (“Hui et al.”)
1020	Excerpts from Peter Symes, <i>Digital Video Compression</i> (2004) (“Symes”)
1021	Excerpts from Fernando Pereira, Touradj Ebrahimi, <i>The MPEG-4 book</i> (2002) (“Pereira et al.”)
1022	Excerpts from Craig Hunt and Robert Bruce Thompson, <i>Windows NT TCP/IP Network Administration</i> (1998) (“Hunt et al.”)
1023	Excerpts from Douglas E. Comer, <i>Internetworking with TCP/IP Vol. 1: Principles, Protocols, and Architectures</i> (4th ed. 2000)
1024	U.S. Patent No. 6,408,096 to Thiow Keng Tan (filed Oct. 21, 1998, issued June 18, 2002) (“Tan”)
1025	U.S. Patent Application Publication No. 2005/0102371 to Emre Aksu (filed Nov. 7, 2003, published May 12, 2005) (“Aksu”)
1026	U.S. Patent No. 6,707,944 to Young Kwon Lim et al. (filed Apr. 14, 1998, issued Mar. 16, 2004) (“Lim”)
1027	Thomas Stockhammer et al., <i>H.264/AVC in Wireless Environments</i> , 13 IEEE Transactions on Circuits and Systems for Video Technology 657 (July 2003) (“Stockhammer”)
1028	Excerpts from Iain E. G. Richardson, <i>H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia</i> (2003) (“Richardson”)
1029	Arnaud Bourge and Joel Jung, <i>Low-power H.264 video decoder with graceful degradation</i> , Visual Communications and Image Processing 2004, 5308 Proceedings of SPIE-IS&T Electronic Imaging 372 (2004) (“Bourge”)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Exhibit No.	Description of Document
1030	U.S. Patent No. 5,867,230 to Feng Chi Wang et al. (filed June 30, 1997, issued Feb. 2, 1999) (“Wang et al.”)
1031	U.S. Patent No. 6,637,031 to Philip A. Chou (filed Dec. 4, 1998, issued Oct. 21, 2003) (“Chou”)
1032	Shanwei Cen et al., <i>Flow and Congestion Control for Internet Media Streaming Applications</i> , Proc. SPIE 3310, Multimedia Computing and Networking 1998, 250 (Dec. 29, 1997) (“Cen et al.”)
1033	Jian Lu, <i>Signal Processing for Internet Video Streaming: A Review</i> , Proc. SPIE Image and Video Communications and Processing (Jan. 2000) (“Lu”)
1034	H. Schulzrinne et al., Real Time Streaming Protocol (RTSP), Internet Engineering Task Force, Internet Society (Apr. 1998)
1035	U.S. Patent No. 7,529,806 to Yevgeniy Eugene Shteyn (filed Nov. 4, 1999, issued May 5, 2009) (“Shteyn”)
1036	U.S. Patent No. 5,721,878 to Hal Hjalmar Ottesen et al. (filed June 7, 1995, issued Feb. 24, 1998) (“Ottesen et al.”)
1037	<i>DV Format, Digital Video</i> , Canon U.S.A., Inc. (Oct. 13, 1999) https://web.archive.org/web/19991013131445/http://canondv.com:80/shared/dvinfo/dvinfo2.html
1038	Alan T. Wetzel, Michael R. Schell, <i>Consumer Applications of the IEEE 1394 Serial Bus, and a 1394/DV Video Editing System</i> , IEEE (1996) (“Wetzel et al.”)
1039	Won-Kyu Paik and Sun-Young Hwang, <i>Design of a novel synthesis filter for real-time MPEG-2 audio decoder implementation on a DSP chip</i> , 45 IEEE Transactions on Consumer Electronics 1119 (1999) (“Paik”)
1040	Keun-Sup Lee et al., <i>Software Optimization of the MPEG-Audio Decoder Using a 32-Bit MCU RISC Processor</i> , 48 IEEE Transactions on Consumer Electronics 671 (Aug. 2002) (“Lee”)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Exhibit No.	Description of Document
1041	Martin Wolters et al., <i>A closer look into MPEG-4 High Efficiency AAC</i> , Audio Engineering Society (AES), 115th AES Convention (Oct. 10-13, 2003) (“Wolters”)
1042	Patrick De Smet et al., <i>Do Not Zero-Put: An Efficient Homespun MPEG-Audio Layer II Decoding and Optimization Strategy</i> , Proc. of ACM Multimedia 2004 conference, pp. 376-379 (2004) (“De Smet”)
1043	Patrick De Smet and Ignace Bruyland, <i>Optimized Recursive Subband Synthesis Windowing for Implementing Efficient MPEG Audio Decoders</i> , 10 IEEE Signal Processing Letters 303 (Oct. 2003) (“De Smet & Bruyland”)
1044	MPEG-4 Overview – (V.21 – Jeju Version), International Organisation for Standardisation Organisation International De Normalisation ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio (Mar. 2002)
1045	U.S. Patent No. 5,754,241 to Shigeyuki Okada et al. (filed Nov. 14, 1995, issued May 19, 1998) (“Okada”)
1046	U.S. Patent No. 7,006,755 to Takeo Morinaga (filed Feb. 8, 2001, issued Feb. 28, 2006) (“Morinaga”)
1047	Vladimir Z. Mesarovic et al., <i>Optimized Huffman Decoding for AAC</i> , Silicon for Audio – AES 16th UK Conference, pp. 117-122 (2001) (“Mesarovic”)
1048	U.S. Patent No. 6,963,972 to Yuan-Chi Chang et al. (filed Sept. 26, 2000, issued Nov. 8, 2005) (“Chang”)
1049	U.S. Patent No. 6,490,320 to Anthony Vetro et al. (filed Apr. 11, 2000, issued Dec. 3, 2002) (“Vetro”)
1050	U.S. Patent No. 6,005,600 to Ralph D. Hill (filed Oct. 18, 1996, issued Dec. 21, 1999) (“Hill”)
1051	U.S. Patent No. 7,398,312 to Katherine H. Guo et al. (filed Mar. 29, 2000, issued July 8, 2008) (“Guo”)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Exhibit No.	Description of Document
1052	U.S. Patent No. 8,868,772 to R. Drew Major and Mark B. Hurst (filed Apr. 28, 2005, issued Oct. 21, 2014) (“Major”)
1053	U.S. Patent Provisional Application No. 60/566,831 to Robert D. Major and Mark B. Hurst (filed Apr. 30, 2004)
1054	115th AES Convention (Oct. 10-13, 2003) program and related webpages and links
1055	Complaint filed in <i>Withrow Networks, Inc. v. Google, LLC</i> , No. 5:24-CV-3203 (N.D. Cal. May 28, 2024), ECF No. 1
1056	Returned Summons in a Civil Action filed in <i>Withrow Networks, Inc. v. Google, LLC</i> , No. 5:24-CV-3203 (N.D. Cal. June 11, 2024), ECF No. 19
1057	List of Claims of U.S. Patent No. 10,771,849
1058	U.S. Patent No. 7,535,485 to Moshe Elbaz et al. (filed Aug. 14, 2001, issued May 19, 2009) (“Elbaz”)
1059	U.S. Patent No. 7,502,514 to Yaakov Gringeler et al. (filed Jan. 4, 2005, issued Mar. 10, 2009) (“Gringeler”)
1060	U.S. Patent No. 6,766,373 to Bruce A. Beadle et al. (filed May 31, 2000, issued July 20, 2004) (“Beadle”)
1061	U.S. Patent No. 7,099,389 to Guoyao Yu et al. (filed Dec. 10, 2002, issued Aug. 29, 2006) (“Yu”)
1062	U.S. Patent No. 7,751,628 to Richard Reisman (filed Dec. 23, 2002, issued July 6, 2010) (“Reisman”)
1063	Expert Declaration of Dr. Ingrid Hsieh-Yee
1064	<i>Curriculum Vitae</i> of Dr. Nathaniel Polish
1065	Joint Case Management Statement filed in <i>Withrow Networks, Inc. v. Google, LLC</i> , No. 5:24-CV-3203 (N.D. Cal. Feb. 7, 2024), ECF No. 48

II. PERSON OF ORDINARY SKILL IN THE ART

18. I am informed by counsel that, under the patent laws in effect before

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

the America Invents Act (“AIA”) of 2011, an assessment of claims of a patent filed before the AIA took effect should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date (*i.e.*, the “time the invention was made”). I understand that the ’849 patent claims priority to U.S. patent application no. 11/107,952, which was filed on April 18, 2005.

19. I have been advised that to determine the appropriate level of a person having ordinary skill in the art, I may consider the following factors: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor. I am also informed that a person of ordinary skill in the art is presumed to be aware of all of the pertinent prior art.

20. The ’849 patent purports to disclose methods and systems for multimedia playback and transmission to wireless clients.

21. In my opinion and based on the subject matter of the ’849 patent, a person of ordinary skill in the art at the time of the alleged invention would have had a bachelor’s degree in computer science, computer engineering, or electrical engineering, or the equivalent, and at least two years of work experience in networking or streaming media systems, particularly audio and video, over the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Internet. Additional education could have substituted for professional experience, and significant work experience could have substituted for formal education.

22. My opinions regarding the level of ordinary skill in the art are based on, among other things, my more than 40 years of experience in the field of computer science and data streaming technologies, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, students, and employees, both past and present.

23. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, I qualify as a person of ordinary skill in the art as of the alleged priority date of April 18, 2005, and, in my analysis below, I have adopted the perspective of one of ordinary skill in the art as of April 18, 2005. In addition to being a person of ordinary skill in the art, I have worked closely with many such persons over the course of my career. My opinions and analysis below would remain the same if a person of ordinary skill in the art were found to have somewhat more or less formal education or somewhat more or less work experience in the field.

III. STATEMENT OF LEGAL PRINCIPLES

A. Claim Construction

24. I understand that a purpose of claim construction is to determine what a person of ordinary skill in the art would have understood the claim terms to mean. Claim terms are generally given their ordinary and customary meaning, which is the meaning that the term would have to a person of ordinary skill in the art in question as of the critical date (i.e., either the earliest claimed priority date (pre-AIA) or the effective filing date (AIA)).

25. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification. I understand that the patent specification, under the legal principles, has been described as the single best guide to the meaning of a claim term, and is thus highly relevant to the interpretation of claim terms. Also, I understand for terms that do not have a customary meaning within the art, the specification usually supplies the best context of understanding the meaning of those terms.

26. I further understand that other claims of the patent in question, both asserted and unasserted, can be valuable sources of information as to the meaning of a claim term. Because the claim terms are normally used consistently throughout the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims. Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.

27. I understand that the prosecution history can further inform the meaning of the claim language by demonstrating how the inventors understood the invention and whether the inventors limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be. Extrinsic evidence, such as dictionaries, may also be consulted in construing the claim terms.

28. I understand that, in *Inter Partes* Review (IPR) proceedings, a claim of a patent shall be construed using the same claim construction standard that would be used to construe the claim in a civil action filed in a U.S. district court (which I understand from counsel is called the “*Phillips*” claim construction standard), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the specification and prosecution history pertaining to the patent.

29. I have been instructed by counsel to apply the “*Phillips*” claim construction standard for purposes of interpreting the claims in this proceeding, to the extent they require an explicit construction. The description of the legal principles set forth above thus provides my understanding of the “*Phillips*” standard

as provided to me by counsel. I have applied these principles and standards in my analysis below.

B. Obviousness

30. I understand that a patent claim is obvious if, as of the critical date (i.e., either the earliest claimed priority date (pre-AIA) or the effective filing date (AIA), it would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs.

31. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other facts known as “secondary considerations” such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus,” with the invention claimed in the patent at issue.

32. I understand that a reference qualifies as prior art for obviousness purposes when it is analogous to the claimed invention. The test for determining what art is analogous is: (1) whether the art is from the same field of endeavor,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

regardless of the problem addressed, and (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

33. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art 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 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 person of ordinary skill in the art 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. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense.

34. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

obvious, then the entire patent claim is invalid.

35. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together like the pieces of a puzzle. I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this finite number of predictable solutions leads to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does not more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

36. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. I understand that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem.

37. I understand the following are examples of approaches and rationales

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other market forces if the variations would have been predictable to a person of ordinary skill in the art;
- (3) Combining prior art elements according to known methods to yield predictable results;
- (4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;
- (5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);
- (6) Simple substitution of one known element for another to obtain predictable results; or

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(7) Use of a known technique to improve similar products, devices, or methods in the same way.

38. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether one of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the field of endeavor, would have seen a benefit to selecting the combination claimed.

39. I understand that the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. The test for obviousness, in other words, is not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the teachings of the prior art as a whole.

IV. TECHNOLOGY BACKGROUND

40. The concepts underlying the '849 patent have long been known to ordinary practitioners and scholars in the field of multimedia creation, transmission, and playback.

A. Streaming Media over the Internet

41. Sending streaming media, such as real time audio, over computer

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

networks has been known since the 1980s, at least with Xerox Parc's Etherphone. This technology quickly developed into the beginnings of efficient streaming media data from a source location to client devices, such as user computers in the early 1990s. Streaming media arose as an alternative to the "download-first-and-then-play" technique, which requires a client device to download a file in its entirety and then begin playing the file only after the entire file has been downloaded onto the client device. (*E.g.*, EX1030 at 1:38-49; EX1020 at 340-341.) In contrast, "[i]n streaming, audiovisual data is packetized, delivered over a network, and played as the packets are being received at the receiving end, as opposed to being played only after all the packets have been downloaded." (EX1031 at 2:9-13.)

42. Networks and compression (which I discuss further below) have worked in tandem to enable different sorts of approaches to streaming. Over the years improvements in network performance have enabled better streaming. Improvements in compression, when enabled by sufficient processing power, have further enhanced streaming. None of these improvements come without certain costs. For example, compression means that losing even a small amount of the transmitted data can make all of the data useless for a certain amount of time. Also, at higher bandwidths sometimes network latency can be highly variable. From the early 1990s to the present time, these issues have evolved until they have reached

the current, highly functional, state of development.

43. Streaming increased in prevalence due to the proliferation of the Internet; users typically did not have access to fast Internet connections, which made downloading large media data files cumbersome. (*E.g.*, EX1031 at 2:13-18.) Streaming media data allows users to begin playing/outputting the media, whether that be video or audio, before the entire file is transmitted. (*Id.*) By the late 1990s, “[t]he real-time distribution of continuous audio and video data via streaming multimedia applications account[ed] for a significant, and expanding, portion of the Internet traffic.” (EX1032 at 250.) Many commercial streaming media players were released around this time, including “RealAudio and RealVideo players, Microsoft Netshow, and Vxtreme.” (*Id.*)

44. Streaming media data is typically achieved by breaking the data into segments and sending each segment individually and sequentially to a client device, where each segment can be downloaded and the data played. As I discuss below, many forms of compression start with an initial or source data state and then encode differences or remove redundancy to reduce the amount of data required to be transmitted across the network.

45. Packets are a concept from the field of networking. Usually a packet is the smallest separately transmitted or routed unit of data within a network. In the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Internet Protocol each packet contains an address and a fixed amount of data. Any larger amounts of data that need to be transmitted on the network are broken into multiple packets and separately transmitted.

46. The server that stores the media data “breaks the media into packets that are routed and delivered over the network.” (EX1033 at 1.) “A series of time-stamped packets is called a stream.” (*Id.*) Segmenting media data in smaller parts, including, for example, into packets, for delivery across a computer network was well known in the art before 2005. (*E.g.*, EX1034 at 7 (“When using RTP, a stream consists of all RTP and RTCP packets created by a source”); EX1035 at 1:62-66 (“It is an object of the invention to provide an open architecture solution for content delivery in a download approach that allows for a low or negligible play-out latency. To this end the content file is split into multiple parts. Each part or segment requires a relatively short download time.”); EX1031 at 2:9-13 (“[i]n streaming, audiovisual data is packetized”).)

47. Data segments containing the media data are typically sequentially ordered so the segments can be properly reassembled and the stream can be played in the proper order. (*E.g.*, EX1036 at 3:21-28 (teaching “an apparatus and method for effectuating local reception and processing of source program signals representative of a multimedia program received from a remote multimedia server.

The multimedia server transmits a selected multimedia program as a custom **ordered series** of discrete, digitally compressed program segments”)¹.)

B. Internet Communication Protocols

48. The Hypertext transfer protocol (HTTP) has been called “the common language of the modern global Internet” and is used by clients, servers, and related web applications to communicate and share information. (EX1006 at 3-4 (“Billions of JPEG images, HTML pages, text files, MPEG movies, WAV audio files, Java applets, and more cruise through the Internet each and every day. HTTP moves the bulk of this information quickly, conveniently, and reliably from web servers all around the world” to client devices); *see also* EX1022 at 279, 412.)

49. Web browsers were/are widely-used HTTP clients. As most have experienced:

When you browse to a page, such as “http://www.oreilly.com/index.html,” your browser sends an HTTP request to the server *www.oreilly.com* (see Figure 1-1). The server tries to find the desired object (in this case, “/index.html”) and, if successful, sends the object to the client in an HTTP response, along with the type of the object, the length of the object, and other information.

¹ All bold and/or underline emphasis has been added unless otherwise indicated.

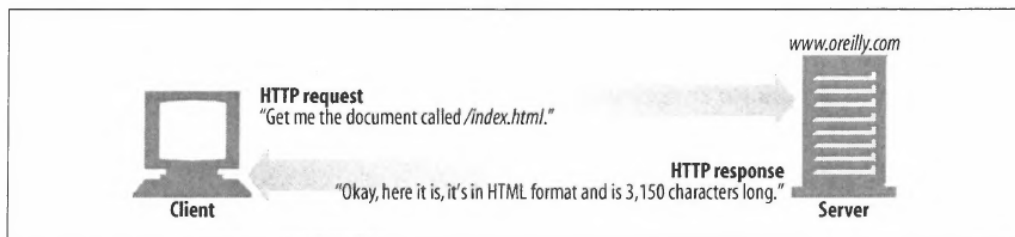


Figure 1-1. Web clients and servers

(EX1006 at 4.) In addition to HTML pages, HTTP can be used to request and obtain other web-based resources hosted by servers, including multimedia video and audio files/data. (*Id.* at 4-6; EX1022 at 413-14.) HTTP transactions consist of request and response messages between the client and server. (EX1006 at 8-9.)

50. Although HTTP is among the, if not the, most-commonly used Internet communications protocol, there are a multitude of others. Another well-known protocol that was (and still is) widely used is the File Transfer Protocol or FTP. FTP predates HTTP and was/is used to upload and download files to/from an FTP server. (See EX1006 at 39.) HTTP and FTP are application protocols that are layered over and use lower-level transport and network protocols, such as TCP/IP, when forming a connection between the client and server. (*Id.* at 12-14; *see also* EX1022 at 412-13; EX1023 at 530 (“*Request/Response*. Once a transport session has been established, one side (usually a browser) must send an HTTP request to which the other side responds.”).)

51. “Each web server resource has a name,”—akin to a postal address—

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

“so clients can point out what resources they are interested in.” (EX1006 at 6.) The uniform resource locator or URL is the most common form of resource identifier. URLs “tell you exactly how to fetch a resource from a precise, fixed location.” (*Id.*; *see also id.* at 23-24 (“Uniform resource locators (URLs) are the standardized names for the Internet’s resources. URLs point to pieces of electronic information, telling you where they are located and how to interact with them.”).) As shown in the exemplary image below, URLs specify the scheme or protocol to be used (e.g., HTTP, FTP), the server on which the resource is located, and the specific path/location and name of the resource on the server.

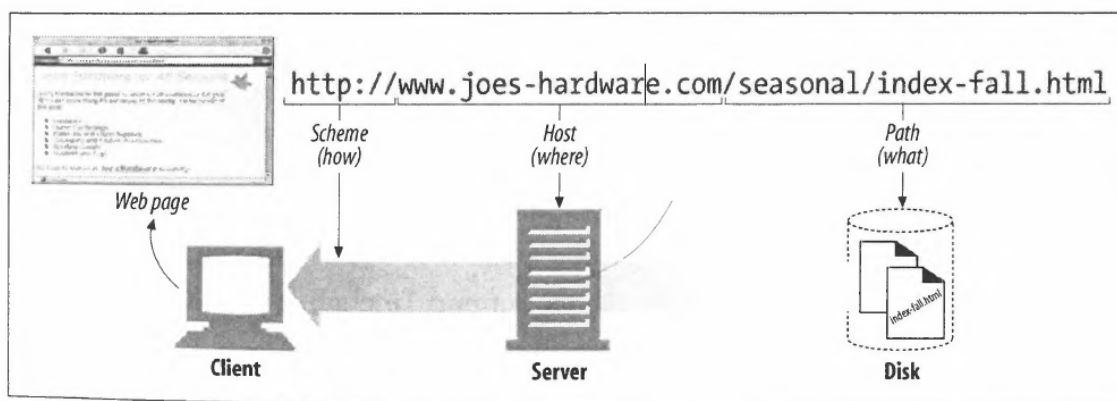


Figure 2-1. How URLs relate to browser, machine, server, and location on the server's filesystem

(*Id.* at 24; *see also id.* at 25 (“Most URLs have the same ‘scheme://server location/path’ structure.”).) The “path” portion of a URL is the local name for the resource on the server. (*Id.* at 26, 28 (“The path component of the URL specifies where on the server machine the resource lives” in a manner resembling a file system

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

path on a local drive).) Exemplary URL formats and syntax for HTTP and FTP schemes are shown below:

Scheme	Description
http	The Hypertext Transfer Protocol scheme conforms to the general URL format, except that there is no username or password. The port defaults to 80 if omitted. Basic form: <i>http://<host>[:<port>]/<path>?<query>#<frag></i> Examples: <i>http://www.joes-hardware.com/index.html</i> <i>http://www.joes-hardware.com:80/index.html</i>
Scheme	Description
ftp	File Transfer Protocol URLs can be used to download and upload files on an FTP server and to obtain listings of the contents of a directory structure on an FTP server. FTP has been around since before the advent of the Web and URLs. Web applications have assimilated FTP as a data-access scheme. The URL syntax follows the general form. Basic form: <i>ftp://<user>[:<password>]@<host>[:<port>]/<path>;<params></i> Example: <i>ftp://anonymous:joe%40joes-hardware.com@prep.ai.mit.edu:21/pub/gnu/</i>

(*Id.* at 38-39.)

C. Digital Audio and Video

52. By the late 1990s, it was well known that video and audio data could be captured by a camcorder or other device and transmitted as digitally encoded data. In the mid-1990s, several commercial video camera manufacturers collaborated to create the DV, or digital video, standard, which was widely adopted by the camera and video industry. (*E.g.*, EX1037 at 3.) Subsequently, video cameras that recorded video in digital formats were common in the late 1990s. For example, “[i]n the fall of 1995 Sony introduced two DV camcorders, the DCR-VX1000 and DCR-

VX700[,]” both of which “meet the HD-DVC requirements for SD ... format digital camcorders.” (EX1038 at 0007.) As with any video compression standard, DV encoded video needed to be decompressed before it could be edited or manipulated and then recompressed, using DV or another video/audio compression method. (*See id.* at 0010-11.)

53. Digital cameras were desirable because they created superior media quality and performance. For example, one commentator observed, “[t]he digital recording and playback, combined with the digital 1394 interface, completely eliminates the noise pick-up and signal losses that occur in analog systems and interconnects.” (EX1038 at 0002.) Furthermore, “[t]he digital record/playback avoids the audio flutter and video picture jitter that is common in present analog systems.” (*Id.*) The field thus recognized the advantages and benefits offered by digital media data.

D. Digital Audio and Video Compression

54. As mentioned above, due to bandwidth restrictions, multimedia data is typically compressed prior to streaming or transmission across a network to an end user/consumer. (*See, e.g.*, EX1020 at 3 (“When we need to transmit a given amount of data, we use a certain bandwidth (measured in bits per second) for the time necessary to transmit all the bits. In the real world, there is always some restriction

on the available bandwidth.”.)

55. Generally speaking, compression works by removing redundancy from the multimedia data. (EX1020 at 322; EX1005 at 9.) “In essence, the same (or nearly the same) information is carried using a smaller quantity or rate of data.... [T]he bit rate is reduced, ideally leaving the dynamics of the signal unchanged.” (EX1005 at 3-4.) When compressing video data, most audiovisual compression standards typically employed/employ techniques that reduce the amount of spatial redundancy—that is, predictable repetition of visual data within a particular image/frame—and techniques that reduce temporal redundancy—that is, visual data that is predictably repeated across a group of multiple images/frames. (*E.g.*, EX1005 at 13-16.)

56. All compression techniques fall into one of two categories: (i) lossless compression or (ii) lossy compression. (EX1005 at 6-7.) In lossless compression, the decoder is able to exactly recreate the original source data. (*Id.*) In contrast, with lossy compression, the data from the decoder is not bit-for-bit identical to the source data. (*Id.*) Lossy compression is used because it can achieve much higher compression factors with (ideally) minimal, if any, effect on the perceived quality of the decoded audiovisual output. (*Id.*) “Successful lossy codecs are those in which the errors are arranged so that a human viewer or listener finds them subjectively

difficult to detect.” (*Id.* at 7.) Accordingly, multimedia compression methods used in streaming applications across the Internet typically employ lossy compression/perceptive coding techniques to achieve higher compression factors.

57. The concept of providing scalability when encoding and decoding multimedia data—and the benefits and advantages of doing so—were also known in the art long before 2005. Scalability refers to an ability of a compression algorithm to adapt to different user capabilities and requests where the quality of the decompressed output multimedia stream is variable/optimized based on resource constraints such as available transmission bandwidth or encoder or decoder processing resources. (*See, e.g.*, EX1020 at 199 (“As an example, if bandwidth and/or computational resources are limited, it may be appropriate to retain full temporal resolution for an important foreground object, but to update to the background at a lower rate.”), 209-210; EX1021 at 294-95 (scalable video coding).)

58. Prior to April 2005 (as well as after), there were many well-known audio and video compression standards in use that were created by different entities and groups in the field. For example, some of the most widely used video (and accompanying audio) compression standards—MPEG-1, MPEG-2, and MPEG-4—were created by the Motion Picture Experts’ Group (MPEG).

59. Many of the MPEG-4 standards were formalized in the late 1990s and

early-2000s with the objective of providing encoding video and audio at very low bit rates with good error resilience to facilitate transmission in limited bandwidth, error-prone environments, including, for example, streaming to mobile devices. (See EX1020 at 196, 199; EX1021 at 4-7, 12-13, 39.) MPEG-4 based compression is based on an object-based audiovisual model where “[a]n object-based scene is built using individual independent objects that have relationships in space and time.” (EX1021 at 11, 39.)

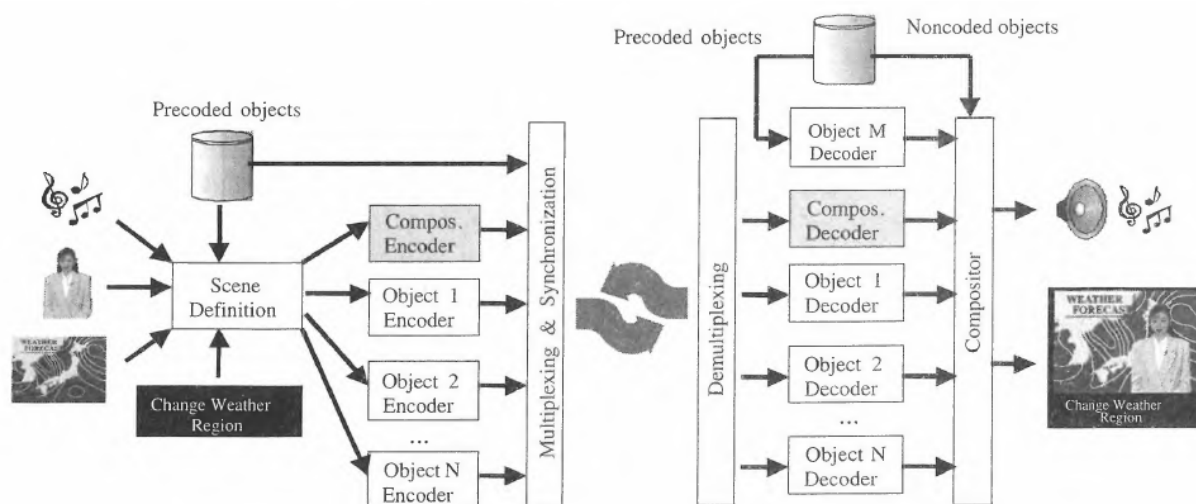


Fig. 1.1 MPEG-4 object-based representation architecture.

(EX1021 at 11.) The following figure provides a walkthrough of some scenarios in which the MPEG-4 standard was used, including for delivery across wireless networks to mobile client devices.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

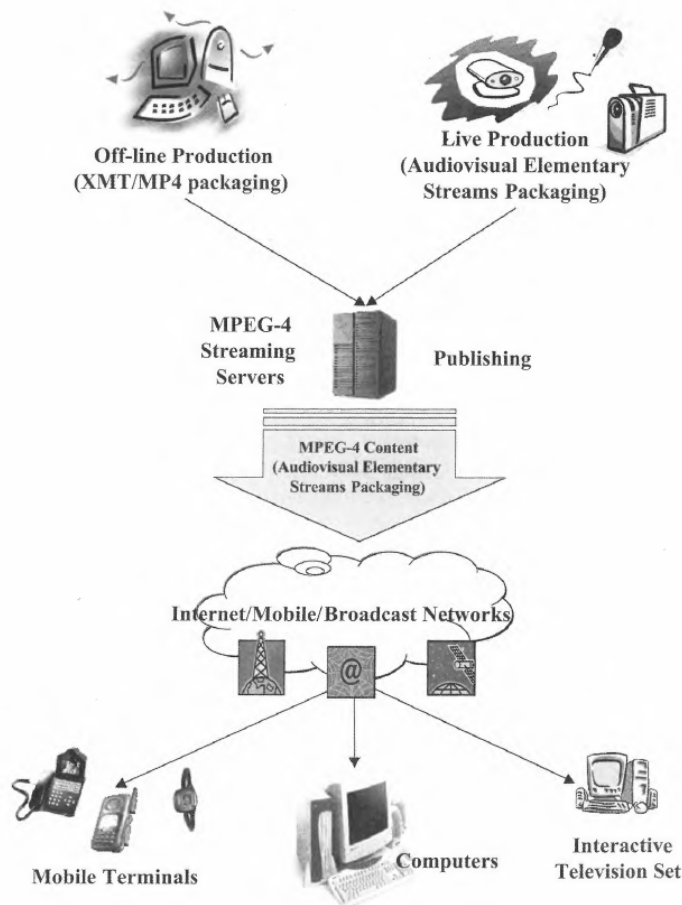


Fig. 2.2 Example MPEG-4 end-to-end walkthrough.

(EX1021 at 41.)

60. The MPEG-4 Advanced Video Coding (AVC) standard (which is sometimes referred to as the H.264 standard) was approved in March 2003. AVC/H.264 was recognized as being particularly well suited for use in mobile applications because of its high compression efficiency, which allowed one to transmit higher quality video using less data/lower bitrates. (*E.g.*, Wang (EX1007) at 1 (“Multimedia services over mobile network[s] are becoming important with the

pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile.”); EX1027 at 657 (“[C]ompression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC [MPEG-4 Part 10] coding an attractive candidate for all wireless applications including multimedia messaging services (MMS), packet-switched streaming services (PSS), and conversational applications.”); EX1028 at xv-xvi, 5-6 (listing some features and improvements in H.264/AVC standard).)

V. THE ’849 PATENT

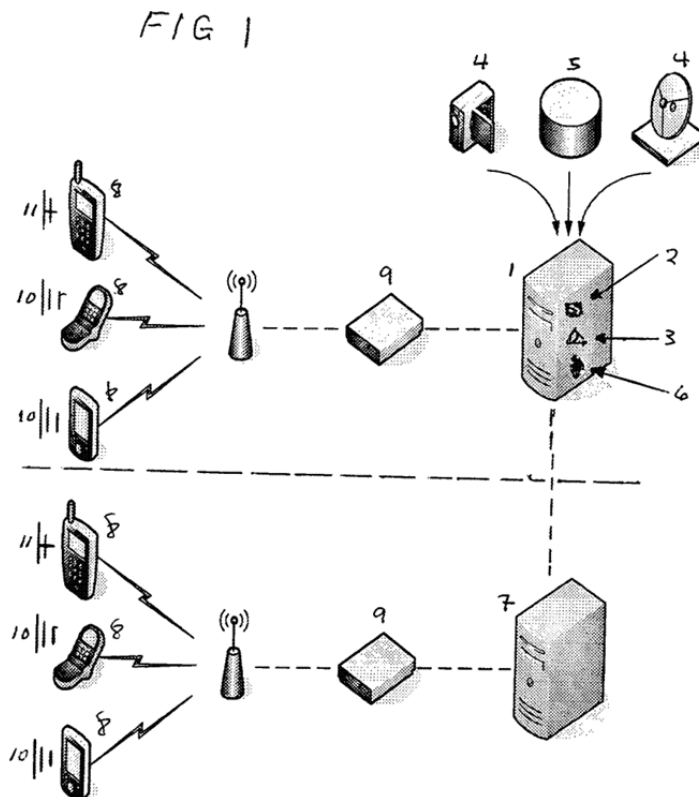
A. Overview of the Specification

61. The ’849 patent states that it “relates to issues of the wireless Internet” and “specifically to methods of multimedia transmission and playback for mobile clients.” (’849 patent at 1:6-8.) According to the ’849 patent “[t]he [then-]current methods of distributing multimedia data over the wireless Internet to mobile clients are constrained by existing wireless bandwidth, and the real-time decoding, processing and displaying of multimedia content with limited hardware capabilities.” (’849 patent at 1:13-17.)

62. The ’849 patent describes several aspects of the purported invention.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

According to the '849 patent, the first aspect “provides a method of efficient multimedia object creation.” ('849 patent at 2:51-53.) The '849 patent describes a system including a “server-based transcoder [2], coupled with a multimedia object creator [3],” that “inputs a standard analog signal or alternative digital signal like MPEG2, and converts this signal into true MPEG4/AAC multimedia objects.” ('849 patent at 2:65-3:2.) The '849 patent states that “[t]he multimedia object creator produces discrete multimedia objects from video and audio segments of a continuous stream.” ('849 patent at 3:7-9.) The multimedia objects are uploaded to one or more content hosting servers where they are made available to download in sequence to clients 8 for playback using multimedia object java applets 10 or embedded or downloaded non-java multimedia players 11. ('849 patent at 3:2-6, 6:18-31, Fig. 1; *see also id.* at 6:32-9:9.)



(’849 patent at Fig. 1.)

63. The second aspect of the purported invention of the ’849 patent purports to “address[] the limitations of large-scale multimedia transmission on cellular networks to wireless clients.” (’849 patent at 2:53-55.) The ’849 patent states that in the case of live content from a live multimedia stream, “the creation [of] multimedia objects is done in real-time and immediately deployed to every content server of the distributed system[,]” which may be geographically spaced and/or take advantage of proxy servers. (’849 patent at 3:63-4:6.) The ’849 patent provides that “[m]edia content converted to multimedia objects must be available to many users and the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

distribution system must be sufficiently robust to allow peaks in demand and have sufficient geographic proximity that network congestion and latency are reduced.” (’849 patent at 3:58-62; *see also id.* at 9:10-62.)

64. According to the ’849 patent, the third aspect “relates to methods of decoding video, sufficiently optimized to be played on a limited wireless client.” (’849 patent at 2:55-57.) The ’849 patent provides that the “the decoder maintains a state information list of short-cuts, related to perceived frame rate in a sorted list starting with those that will decrease output quality the least, to those that will decrease output quality the most but have the most impact on decoding speed.” (’849 patent at 4:10-14.) The ’849 patent continues: “[t]he client player-decoder dynamically adjusts how many shortcuts must be taken. These short-cuts are specifically designed to drastically reduce the number of computations necessary at certain critical steps in the video decoding process at the cost [of] video output quality.” (’849 patent at 4:14-19.) According to the patent, “[t]his allows the video decoder to scale in complexity based on the processing power of the device being used” and “[i]t also allows users to experience multimedia playback despite the limitations of the device they may be using.” (’849 patent at 4:19-23; *see also id.* at 9:63-14:60.) The ’849 patent provides that decoding on constrained or limited devices may introduce additional requirements, stating, for example: “[t]he Chen’s

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

IDCT algorithm is optimized, but the processing time consumed by the standard Chen implementation is to[o] great for real-time decoding on limited devices.” (’849 patent at 12:60-66.)

65. The fourth aspect of the purported invention described in the ’849 patent “relates to methods of decoding of audio, sufficiently optimized to be played on a limited wireless client.” (’849 patent at 2:58-60.) The ’849 patent states that it presents solutions to issues with audio playback on devices with processors that are limited to integer math and have limited audio output abilities, such as an inability to play continuous sound. (*See* ’849 patent at 4:24-43; *see also id.* at 14:61-18:63.)

66. Finally, according to the ’849 patent, in a fifth aspect “the limited mobile handset multimedia object player, for both MPEG4 video decoding and AAC audio decoding, is implemented as a device-hardware-independent java (J2ME) applet.” (’849 patent at 2:60-63; *see also id.* at 4:44-57, 18:64-19:9.)

B. Prosecution History [EX 1008]

67. The ’849 patent was filed on November 5, 2018 and claims priority to a number of earlier filed patent applications. The earliest application to which the ’849 patent claims priority is application No. 11/107,952, which was filed on April 18, 2005.

68. During prosecution of the ’849 patent, the examiner issued a non-final

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

rejection of the then-proposed claims under 35 U.S.C. 112 (pre-AIA), second paragraph and under pre-AIC 35 U.S.C. 103(a) over Chang et al. (U.S. Pat. No. 6,963,972) in view of Vetro et al. (U.S. Pat. No. 6,490,320). (EX1008 at 505-514.)

69. In response, the applicant amended the proposed claims to recite “wherein the multimedia object metadata identifies the encoding rate dependent on the bandwidth selection and other parameters for playback configuration” and argued that “[n]either Chang nor Vetro uses metadata for client-side controls of bandwidth rate adaptation and decoding optimizations as is presently claimed.” (EX1008 at 540-544.)

70. The examiner subsequently issued a final rejection of the then-proposed claims under 35 U.S.C. 112 (pre-AIA), second paragraph, stating “[w]hile ‘meta-data’ is mentioned once on page 21 of Applicant’s specification, there is no description given of what the metadata identifies or contains[,]” and under pre-AIA 35 U.S.C. 103(a) over Chang in view of Vetro and in view of Hemy et al. (U.S. Pat. No. 6,747,991). (EX1008 at 546-558.)

71. In response, the applicant amended the proposed claims to substitute “object parameters and supplied host path identification” in place of “metadata” and to add the term “autonomously” to the “adjusting said selection and playback according to the multimedia object ~~metadata~~parameters and supplied host path”

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

identification” element. (EX1008 at 577-579.) In its remarks, the applicant stated in part that: “Applicant has also clarified that adjusting selection and playback is done autonomously (ie. [*sic*] without server-side intervention).” (EX1008 at 580.)

72. The examiner subsequently issued a notice of allowance for claims 1-9, identifying the limitation that recites “wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification” in the examiner’s reasons for allowance. (EX1008 at 588-590.)

73. I note that the applicant included Carmel (and Carmel ’432 and Carmel ’230) in an Information Disclosure Statement (IDS) that it submitted, but the examiner did not discuss or otherwise acknowledge Carmel (or Carmel ’432 or Carmel ’230) during prosecution. (See EX1008 at 141, 143, 503-535, 545-567.) The Mattavelli and Wang references that I discuss below were not submitted as part of an IDS and were not considered by the examiner during prosecution. As I discuss below, the combinations of Carmel and Mattavelli and Carmel and Wang disclose and render obvious claims 1-9, including the limitation identified by the examiner in the examiner’s reasons for allowance. (See EX1008 at 589.)

C. The Challenged Claims

74. My declaration addresses all of the claims (1-9) of the '849 patent, of which claims 1, 6, and 7 are independent. Representative claim 1 recites:

1. A method comprising:
 - [a.i] receiving audio and video segments encoded in a digital encoding format and with an encoding rate;
 - [a.ii] wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;
 - [b.i] requesting by a multimedia player, transmission of said multimedia objects;
 - [b.ii] wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;
 - [c.i] playing back received multimedia objects by the multimedia player,
 - [c.ii] wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,
 - [c.iii] autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,
 - [c.iv] and by utilizing optimized decoding processes to maintain quality playback.

('849 patent, 19:19-39 (Claim 1; bracketed notations (e.g., [a.i]) added).) Additional details about the claims are provided in my analysis in **Part VI** below.

VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS

75. I have reviewed and analyzed the prior art references and materials listed in Part I.B above. In my opinion the claims of the '849 patent are rendered obvious based on the following prior art.

Ground	Claims	Basis for Challenge under §103
1	1-9	Carmel in view of Mattavelli
2	1-9	Carmel in view of Wang
3	1-9	Carmel in view of Wang and Wolters
4	1-9	Carmel in view of Wang, Wolters, Guo, Yu, and Forecast
5	1-9	Carmel in view of Mattavelli, Guo, Yu, and Forecast

76. For each of these identified obviousness grounds, a person of ordinary skill in the art would have considered the teachings of the identified references further in view of such a person's knowledge, skill, and experience in the art of networking and streaming media. In my opinions below, my discussion and analysis of the teachings of a particular reference or what was known by one of skill in the art is not meant to and should not be understood to be limited to the particular ground, claim, or claim element in which such discussion or analysis appears, but

rather as relevant and applicable to any and all related or applicable grounds, claims, or claim elements.

77. I am informed by counsel that each of the references cited in the grounds above qualifies as prior art to the challenged claims because each reference was filed and/or published before the '849 patent critical date.

A. Brief Summary of Prior Art

1. Carmel [EX1003]

78. Carmel, U.S. Patent No. 6,389,473, is titled “Network Media Streaming.” Carmel “relates generally to network data communications, and specifically to real-time multimedia broadcasting over a network.” (Carmel at 1:10-12.) I am informed Carmel qualifies as prior art to the '849 patent because it was filed on March 24, 1999 and issued on May 14, 2002, which is several years before the '849 patent's apparent April 18, 2005 priority date.

79. Carmel discloses a method for generating a stream of media data and broadcasting the stream via a network (such as the Internet) to client devices. Carmel discloses a system 32 comprising “a transmitting computer 34, which generates the sequence, a plurality of clients 30, and a network server 36, all of which communicate over network 28, preferably using the well-known Internet Protocol[.]” (Carmel at 6:24-35.)

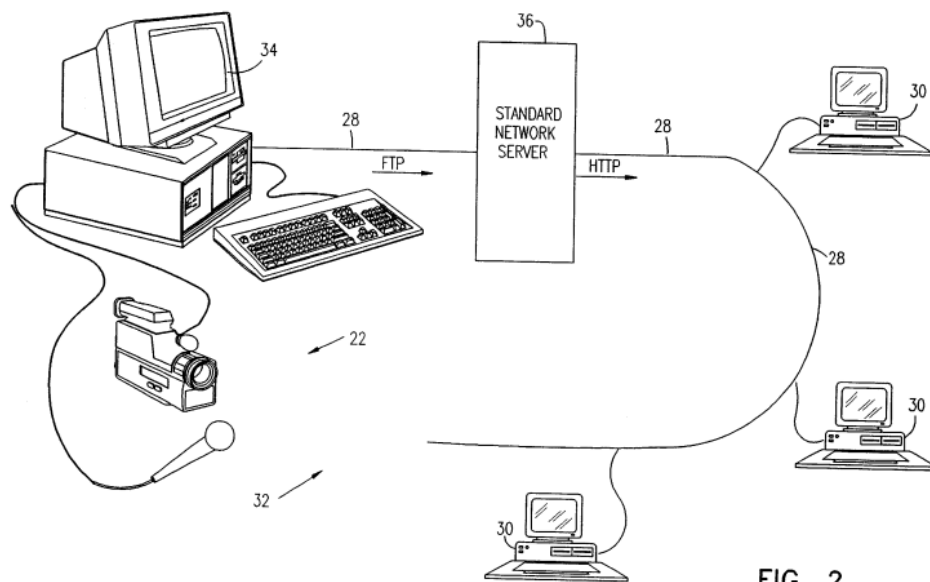
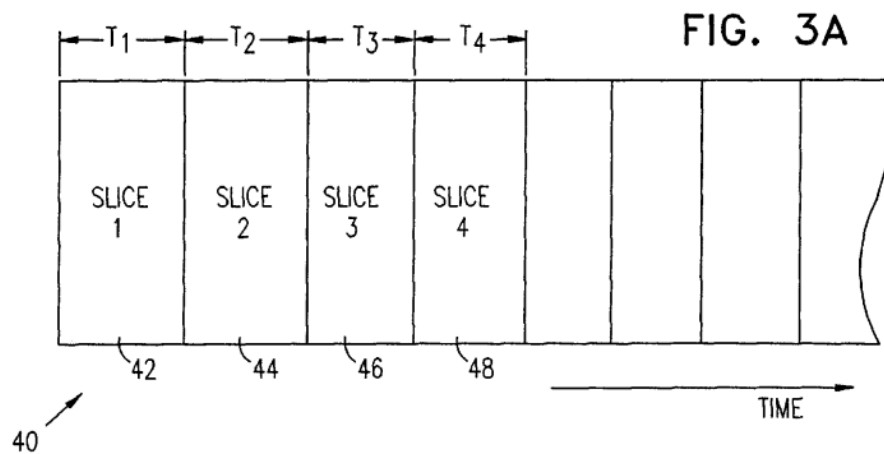


FIG. 2

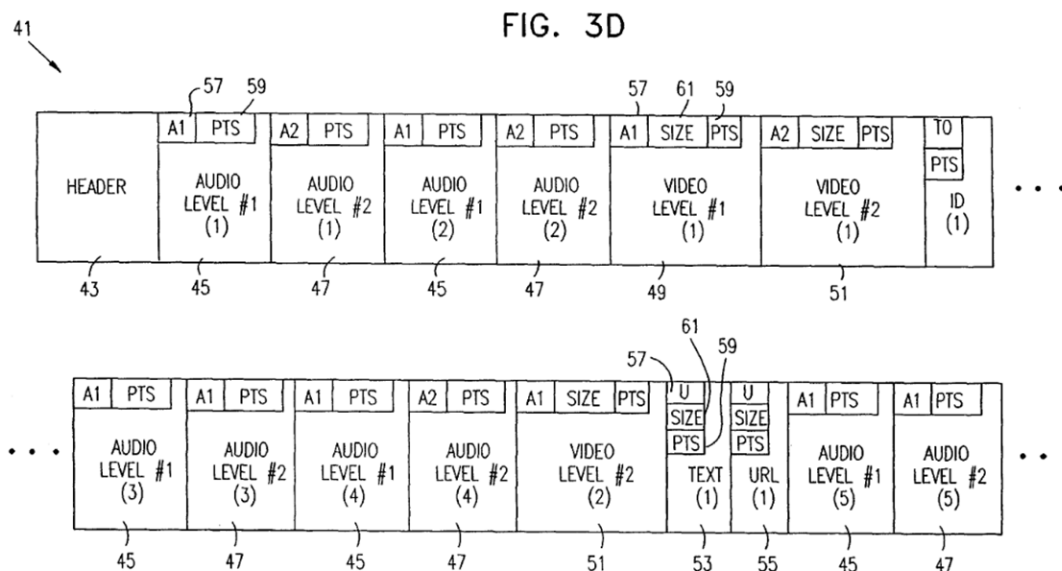
(Carmel at Fig. 2.)

80. The streaming system disclosed by Carmel operates in the following exemplary manner. Carmel teaches computer 34 “receives audiovisual input from input devices 22,” such as a microphone or video camera. (Carmel at 6:32-34.) Computer 34 produces “a stream of broadcast data 40 ... typically corresponding to a multimedia data sequence[.]” (Carmel at 7:18-22.) The data stream 40 “comprises a series of data slices 42, 44, 46, 48, etc.” (Carmel at 7:22-23.) Each slice has a corresponding slice index and is stored in computer 34’s “running slice index 1, 2, 3 ... N.” (Carmel at 7:27-28.)



(Carmel at Fig. 3A.)

81. Carmel discloses a multi-level data stream 41 embodiment, in which the multimedia stream is divided into different slices of audio (45, 47) and video (49, 51) wherein “[e]ach time slice in stream 41 includes multimedia data at multiple quality levels.” (Carmel at 8:42-57.) Each of the clients 30 chooses “the quality level appropriate to the bandwidth of its link on network 28 to server 36.” (Carmel at 9:6-8.)



(Carmel at Fig. 3D.)

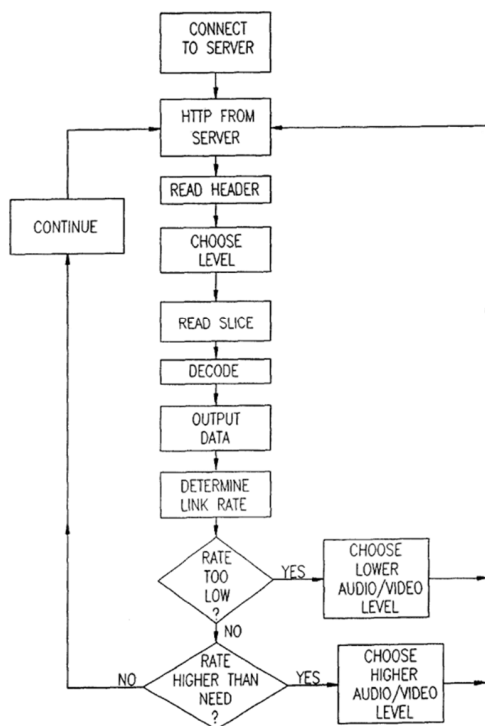
82. Client computers connect to server 36 to “begin[] to download the data stream ... to start receiving the data stream substantially in real time[.]” (Carmel at 8:1-7.) The client can select where in the stream to begin receiving the data: the client can read the index file to determine “the file in data stream 40 that was most recently uploaded by computer 34[;]” or, alternatively, the client can “choose to begin downloading data stream 40 from an earlier point in time[.]” (Carmel at 7:59-8:11.) The index file “contains parameters relating generally to the data stream and/or instructions to computers 30[.]” (Carmel at 8:16-17.)

83. Clients connect to server 36 and begin downloading and consuming data stream 40 using a well-known Internet protocol, such as HTTP, UDP, or other

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Internet protocols. (Carmel at 7:4-17, 10:24-54, 10:67-11:15.) The client “monitors the rate of data coming in over ... its links with server 36.” (*Id.* at 10:55-56.) Carmel teaches multiple means of maintaining a consistent and sufficient transmission rate between server 36 and clients 30. Carmel explains the data rate between server 36 and clients 30 “should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.” (Carmel at Abstract, 2:57-59.) Carmel discloses several methods for achieving this aim, including an option where the client computers 30 select lower quality levels for the slices, which permits faster transmission speed, or if the available bandwidth permits, a higher quality level for a higher quality playback. (Carmel at 11:1-22.)

FIG. 6B



(Carmel at Fig. 6B.)

84. Prior to slicing the data in audio and video segments, Carmel teaches that the multimedia data is preferably compressed (*i.e.*, encoded) by computer 34 “using any suitable compression method known in the art.” (Carmel at 11:23-52, Fig. 7.) Carmel provides examples of suitable compression methods and/or video and audio coding formats that were well-known as of Carmel’s filing date, including GSM 6.10 for audio, H.263 for video, and MPEG standards for audio and video. (Carmel at 11:23-52.)

85. Carmel teaches that each client 30 includes a media player through

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

which the media data can be played. Clients 30 “will typically need only add a Java applet or plug-in to their existing Web browsers ... in order to receive and play back the broadcast.” (Carmel, 7:14-17.) “The operation of client is controlled by a Java applet[.]” (Carmel at 10:27-29.) Carmel provides software for a “sample applet” that can be used to download, decode, and play back data stream 40. (Carmel at 10:33-34, 13:53-14:16.)

86. Carmel also incorporates by reference the disclosure of two other related patents/patent applications by many of the same inventors. Carmel provides: “[s]ome useful techniques for preparing, compressing and transmitting multimedia sequences are described in U.S. Pat. No. 5,841,432 [Carmel ’432, EX1009] and in the above-mentioned U.S. patent application Ser. No. 08/919,027 [Carmel ’230, EX1010], both of which are incorporated herein by reference.” (Carmel at 6:65-7:3; *see also id.* at 2:41-46.) Carmel ’432 and Carmel ’230 disclose, among other things, that its systems and methods include and/or are suitable for networks that have wireless connections utilized by wireless clients. (*See, e.g.*, EX1009 at 1:29-36, 3:21-35, 3:43-53, Fig. 1; EX1010 at 11:50-57.)

87. It is my understanding that Carmel’s disclosure includes the disclosure of Carmel ’432 and Carmel ’230 due to Carmel’s incorporation of these other patents/application by reference. Even if, however, Carmel’s incorporation by

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

reference of Carmel '432 and Carmel '230 does not suffice to incorporate all of the disclosure of Carmel '432 and Carmel '230 into Carmel itself, a person of ordinary skill in the art would still have found it obvious to and would have been motivated to consider the teachings of Carmel, Carmel '432, and Carmel '230 together in view of their shared inventors, their common and overlapping subject matter (*i.e.*, conveying multimedia sequences to a destination computer across a computer network), and Carmel's express reference to and incorporation of the other two Carmel patents.

88. Additional details about Carmel are provided beginning in the analysis of the '849 patent claim limitations, in Part **VI.B** below.

2. Mattavelli [EX1004]

89. **Mattavelli** is an article entitled "Computational Graceful Degradation for Video Decoding" by Marco Mattavelli and Sylvain Brunetton. I understand that Mattavelli was published as part of the materials delivered at the 9th European Signal Processing Conference (EUSIPCO 1998), which was held between September 8-11, 1998. (EX1004.) I understand that these conference materials, including Mattavelli, was additionally published at least by April 22, 1999. (*See* EX1004; EX1063, ¶¶26-42.) I am informed that Mattavelli qualifies as prior art to the '849 patent because it was published several years before the '849 patent's apparent April 18, 2005 priority

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

date.

90. Mattavelli provides “motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding” wherein the “goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources.” (Mattavelli at 2309.) Mattavelli teaches methods where degraded, less computationally intensive decoding algorithms or processes for compressed multimedia data are used during periods where the processing needs are high and exceed the available processing resources, where the visual quality of the video output is temporarily degraded to ensure that the fluidity or real-time continuity of the playback is maintained. (Mattavelli at 2309-10.) Mattavelli states: “[o]bviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as [a] tool able to guarantee real-time performance in [the] presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power.” (Mattavelli at 2312.)

91. Mattavelli identifies several decoding processes, operations or algorithms that can be simplified to save computational resources when necessary, at the expense of potentially degraded, noisier, or less precise results. (Mattavelli at

2310.) These operations include, e.g., “the IDCT [inverse discrete cosine transform], the prediction with and without motion compensation in its various forms (PB/B-frame prediction, overlapped block motion compensation (OBMC), half pixel interpolation, macroblock padding and so on). Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expense[] of imprecise results.” (Mattavelli at 2310.)

92. Mattavelli discloses two strategies or approaches for determining when (and to what extent) such CGD techniques should be employed. Mattavelli states that: “[t]he first is to monitor the execution of the decoding process, to assume that the decoding time of each portion of the frame or VOP [video object plane] is proportional to the image area, and to execute CGD algorithms whenever the current monitored decoding time exceeds the current target decoding time suitable to respect the specified real-time constraints.” (Mattavelli at 2310.) Mattavelli states that:

The second strategy is to exploit the complexity information carried by the frames or VOP headers, like it can be done for MPEG-4 systems [6], and use it in two stages. The first stage predicts the decoding time needed for the current frame or VOP as described in [5]. The second, on the basis of the predicted decoding time and on the complexity information (i.e. statistics of the algorithms necessary to correctly decode the frame), executes only the CGD algorithms that are just necessary to not exceed the target decoding time.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(Mattavelli at 2311.) Mattavelli also provides examples employing each approach, implementing “computationally degraded IDCT and motion compensation algorithms” using the first strategy and “implementing degraded motion compensation algorithms” using the second strategy. (See Mattavelli at 2311-12, Figs. 2 & 3.)

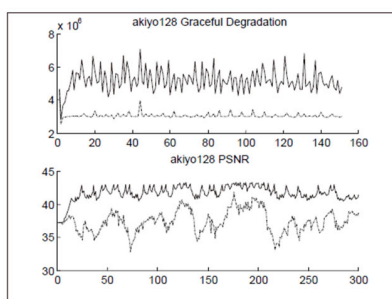


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two PSNR.

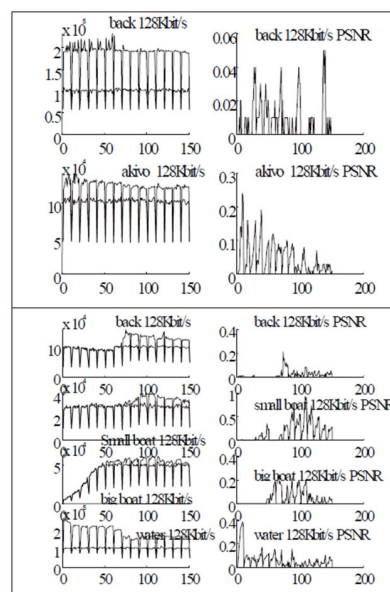


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

(Mattavelli at Figs. 2 & 3.)

93. The second approach taught by Mattavelli uses decoding complexity metadata that is available with the MPEG-4 standard to determine when and by how much to simplify and potentially degrade the video decoding/decompressing in view of the processor constraints on the client/decoder. Mattavelli states that “[o]bviously

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

for any Allocated Decoding Time the goal is to respect the real-time constraints with the lowest degradation in the video/audio content” and “many suitable optimization method[s] or heuristics” can be used to achieve this aim. (Mattavelli at 2311-12.)

94. Additional details about Mattavelli are provided beginning in the analysis of the ’849 patent claim limitations, in Part **VI.B** below.

3. Wang [EX1007]

95. **Wang** is an article entitled “A Platform-Based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining” by Shih-Hao Wang et al. I understand that Wang was published by the IEEE as part of the materials delivered at the Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications, and Signal Processing and Fourth Pacific-Rim Conference on Multimedia (ICICS-PCM 2003), which was held between December 15-18, 2003. (See EX1007.) I am informed that Wang was additionally published at least by May 4, 2004. (See EX1063, ¶¶43-69.) I am informed that Wang qualifies a prior art to the ’849 patent because it was published over a year before the ’849 patent’s apparent April 18, 2005 priority date.

96. Wang presents “a baseline MPEG-4 AVC (Advanced Video Coding) decoder [for mobile platforms] based on an optimized platform-based design methodology.” (Wang at 51.) Wang explains that with this methodology, the authors

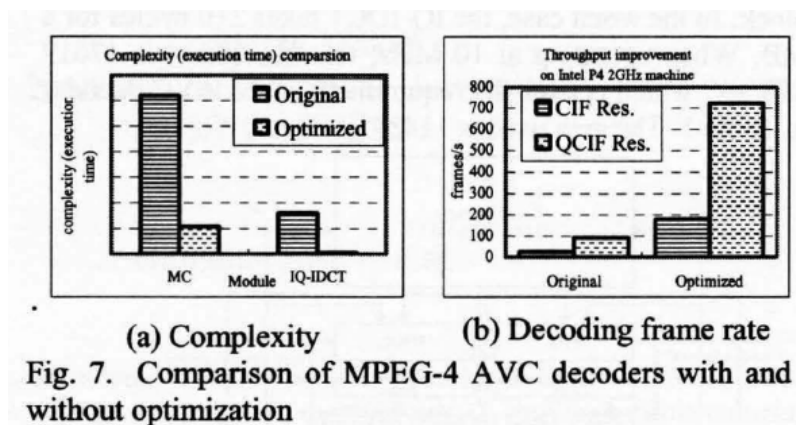
Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

“optimize the software and hardware design of the decoder,” increasing throughput by synchronizing the software and the dedicated co-processors using macroblock-level pipelining. (*Id.*) Wang discloses several techniques and processes by which it “optimize[s] the decoder software” including by enhancing the frame buffer management, boundary padding, and content aware inverse transform and by providing dedicated acceleration modules for the most computationally intensive tasks. (*Id.*; *see also id.* at 53 (“Software Optimization[:] To retain the flexibility and portability across different platforms, our software optimization is mainly from algorithm level improvement. Our optimization techniques are as follows....”).)

97. Wang recognized that “[m]ultimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices.” (Wang at 51.) Wang also teaches that for these kinds of mobile applications over wireless networks the MPEG-4 Advanced Video Coding (AVC) generally provides better image quality “due to its significant improvement on the coding efficiency” compared to other coding formats. (*Id.*) Wang teaches that due to increased power and computational needs, however, “it is challenging to realize [a] real-time AVC encoder and decoder” and that while several optimized AVC modules for general purpose computers had been developed, performance of these general purpose computer-optimized AVC modules is “seriously constrained by the

low computation power and small memory for mobile devices” (*Id.*)

98. “To address such constraints with limited resource[s]” and “increase overall throughput” Wang teaches MPEG-4 AVC decoding processes that are optimized for decoding on mobile devices. (Wang at 51; *see also id.* at 51-55.) Wang states that the optimizations of its prototype MPEG-4 decoder improves throughput by “6 to 7 times” compared to a reference decoder when evaluated on an ARM platform, which was selected as it “is one of the most typical portable devices” (Wang at 51-52; *see also id.* at 54-55.)



2. For the blocks with only nonzero DC coefficient, perform only the inverse DC transform.
3. For the blocks with all zeros, skip the transform.

With our optimization, the entire code size is reduced to 80 kilo-bytes and the complexity for the MC, loop filter and IQ-IDCT are greatly reduced. Fig. 7 (a) shows the complexity comparison in execution time with and without our optimization. Further, Fig. 7 (b) shows the decoding throughput improvement (on an Intel P4 2GHz PC). With our optimization, the overall throughput is improved by 6 to 7 times on the average.

(Wang at 54.)

4. Wolters [EX1041]

99. **Wolters** is paper entitled “A closer look into MPEG-4 High Efficiency AAC” by Martin Wolters et al. I understand that Wolters was published by the Audio Engineering Society (AES) as part of the materials delivered at the 115th AES Convention, held in New York City between October 10-13, 2003. (*See* EX1054; EX1041.) I am informed that was additionally published at least by December 12, 2003. (EX1063, ¶¶70-106.) I am informed that Wolters qualifies as prior art because it was published over a year before the ’849 patent’s apparent April 18, 2005 priority date.

100. Wolters presents an overview of the High-Efficiency Advanced Audio Coding (HE AAC) standard, officially accepted in September 2003, which Wolter’s teaches “[was] not a replacement for [MPEG] AAC, but rather a superset which extends the reach of high-quality MPEG-4 Audio to much lower bitrates.” (EX1041 at 1-2.) Wolters states that HE AAC improved audio coding efficiency by more than 30%. (*Id.* at 1.)

101. Wolters teaches that HE AAC was designed and was especially suited for use in wireless streaming applications to mobile devices. Wolters provides: “[s]pecifically in mobile devices, computational complexity and memory requirements are directly related to power consumption and implementation cost.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

HE AAC was specifically designed to minimize these resources[,]” particularly in certain profiles. (EX1041 at 12; *see also id.* (Example of “Basic configuration for cell phones: Since the listening environment is not very demanding but computational resources might be extremely limited, a down-sampled, low power HE AAC decoder supporting level 2 of the HE AAC profile might be suitable.”), 13 (“Typical scenarios for the Low power version are mobile devices where complexity and thus power consumption is critical while the listening environment is usually not as demanding as in other applications.”).)

102. Wolters also teaches that “optimized source code” was available for license and “[o]ptimized binary implementations” of audio decoders of the HE AAC standard—that is, decoders that utilized optimized audio decoding processes—were available and known in the art. (EX1041 at 12.)

5. Guo [EX1051]

103. **Guo**, U.S. Patent No. 7,398,312, entitled “Method and System for Caching Streaming Multimedia on the Internet,” was filed March 29, 2000. (EX1051.) I am informed that Guo qualifies as prior art because it was filed before the ’849 patent’s apparent April 18, 2005 priority date.

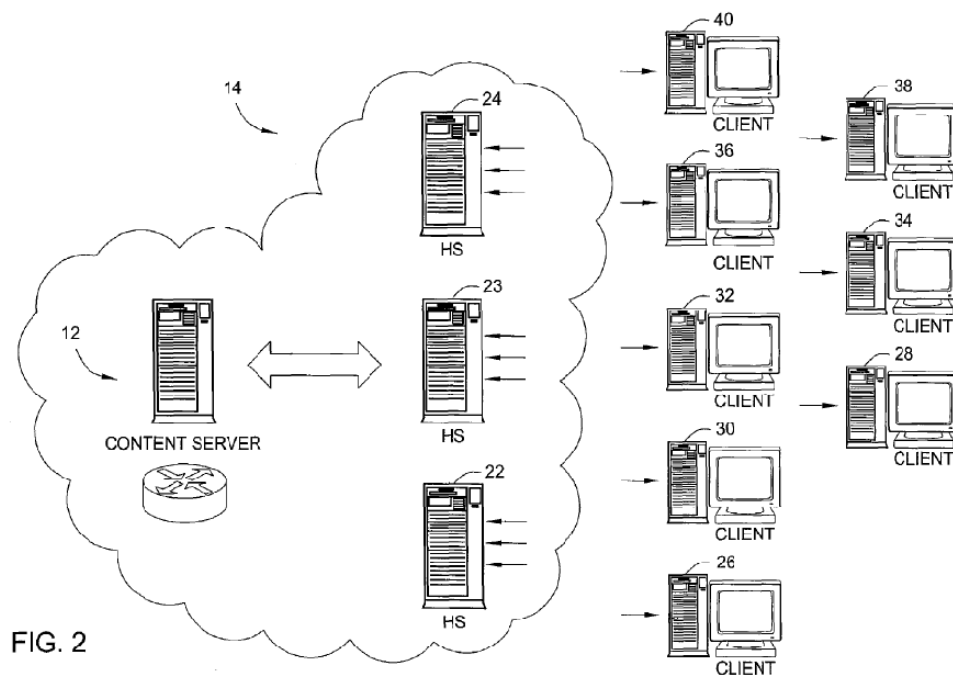
104. Guo teaches streaming systems that use multiple “helper servers” (HSs) “to better support streaming media storage and distribution.” (EX1051 at Abstract,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

2:57-61, Fig. 2.) Guo provides that “[t]he HSs [helper servers] serve to implement several methods specifically designed to support streaming multimedia, including segmentation of streaming multimedia objects (SM objects) into smaller units (i.e., chunks), cooperation of the HSs, and novel cache placement and replacement policies of the constituent units (chunks) which make up the SM objects.” (*Id.* at 2:54-3:8.) Guo teaches that a multimedia stream is divided into smaller chunks/segments of data and that copies of these chunks are stored/cached on multiple helper servers that work together to transmit the full data stream to a client upon request. (*Id.* at 6:24-46 (“In accordance with the object segmentation approach of the present invention, the SM [streaming multimedia] object is divided into multiple chunks where each chunk is made up of a fixed number of segments. The chunks which make up the SM object can then be cached and replaced independently, thereby significantly increasing the utilization of the cache storage.... **A SM object request will likely result in the client receiving some number of segments of the SM object from more than one HS 22-24.**”), 7:11-16 (“In either case, **it may be necessary for a client to get all the constituent segments of a SM [streaming multimedia] object from more than one HS 22-24.** To do so, some form of intelligent pre-fetching is required which pre-fetches subsequent segments while a current segment is being streamed to the client.”); *see*

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

also id. at Abstract, 2:34-50, 3:65-4:6, 4:48-67, 6:2-46, 7:18-32, 11:17-40, 12:50-61, 13:12-19.)



(EX1051 at Fig. 2.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

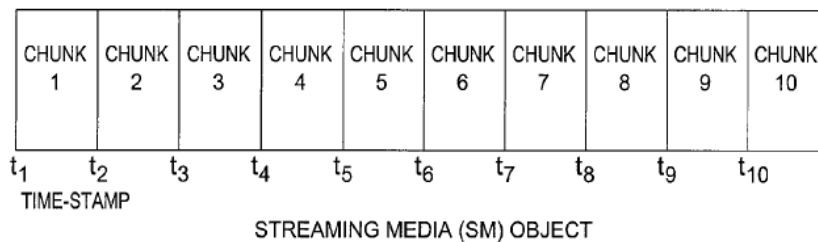


FIG. 5a

LOCAL SERVER CACHE 22	REMOTE CACHE 23	REMOTE CACHE 24	CONTENT SERVER 14
2,3,4,7,8,9	1,2,5,6	1,3,5,6,7,8,9,10	1,2,3,4,5,6,7,8,9,10

CACHE STORAGE TABLE

FIG. 5b

(EX1051 at Figs. 5a and 5b.)

6. Yu [EX1061]

105. **Yu**, U.S. Patent No. 7,099,389, entitled “Rate Control with Picture-Based Lookahead Window,” was filed December 10, 2002. (EX1061.) I am informed that Yu qualifies as prior art because it was filed before the ’849 patent’s apparent April 18, 2005 priority date.

106. Yu discloses systems and methods for providing “rate control with a picture-based lookahead window for encoders/transcoders having mixed codecs in a dual-pass compressed video architecture.” (EX1061 at 1:56-59.) Yu teaches that compression standards have evolved over the years, with newer standards generally

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

providing better, more efficient compression, but they also are usually more complicated or complex to implement. (*Id.*, 1:10-17.) Yu recognizes that there are instances in which one would like to convert a multimedia stream encoded with one codec to a different, typically newer and more sophisticated, standard. Yu teaches that to perform such a conversion, a transcoder consisting of a decoder for the old codec and an encoder for the new codec are used together. Yu provides:

Another reason older codecs continue to be used is that many video streams have already been compressed with the older algorithms, and may easily be decompressed by the older codecs. However where high coding efficiency is desired, there arises the mixed use of both older and newer codecs. In some applications it is desirable to re-transmit video streams compressed with an older codec at a new bit-rate that is lower than the older codec can achieve for the same video quality. Therefore to obtain higher compression efficiency a transcoder having mixed codecs—an older decoder and a newer encoder—is used. A good example is a transcoder that converts MPEG2 compressed video streams to H.264 compressed video streams.

(EX1061 at 1:26-39.)

107. Yu provides an example of such a transcoder that is configured to accept a compressed input stream, decoding it, and then re-encoding it using a desired codec. (EX1061 at 1:56-2:23, 2:44-3:22.)

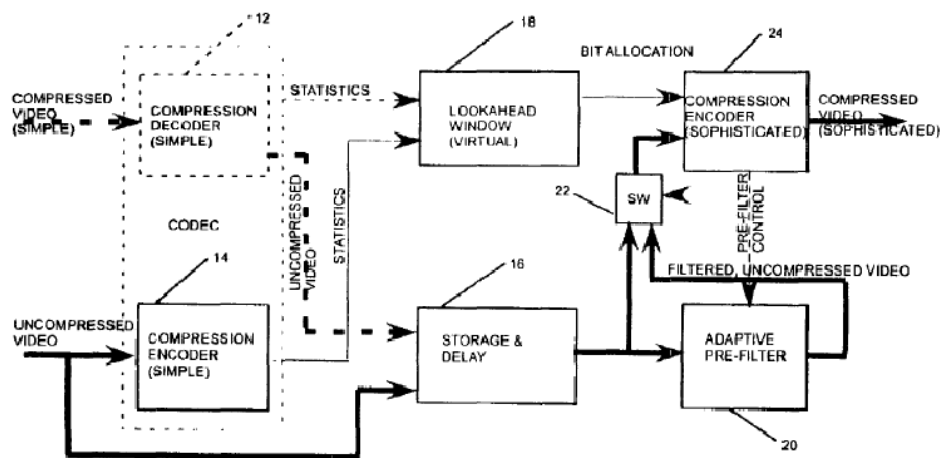


FIG. 1

(*Id.* at Fig. 1.) Yu further teaches that its transcoder is able to extract statistics from the original compressed video input stream and use these statistics to more efficiently re-encode the data with the desired, typically more sophisticated, compression standard. (*See, e.g., id.* at Abstract, 1:6-52, 1:56-2:23, 3:23-26, 9:7-19, Fig. 1.)

7. Forecast [EX1016]

108. **Forecast**, U.S. Patent No. 7,096,481, entitled “Preparation Of Metadata For Splicing Of Encoded MPEG Video and Audio,” was filed March 31, 2000. (EX1016.) I am informed that Forecast qualifies as prior art because it was filed before the ’849 patent’s apparent April 18, 2005 priority date.

109. Forecast discloses techniques for splicing an encoded digital motion stream (such as an MPEG Transport Stream), including (in part) preprocessing that

includes “Group of Pictures (GOP) level pre-processing of splicing In Points”

(EX1016 at Abstract.) Forecast teaches:

[An] encoded MPEG video stream is comprised of a series of groups of pictures (GOPs), and each GOP begins with an independently encoded (intra) I frame and may include one or more following P-frames and B-frames. Each I frame can be decoded without information from any preceding and/or following frame. Decoding of a P frame requires information from a preceding frame in the GOP. Decoding of a B frame requires information from a preceding and following frame in the GOP. To minimize decoder buffer requirements, each B frame is transmitted in reverse of its presentation order, so that all the information of the other frames required for decoding the B frame will arrive at the decoder before the B frame.

(EX1016 at 1:34-47.)

110. Forecast further provides that “[t]he splicing of MPEG encoded audio/visual streams, however, is considerably more difficult than splicing of the uncompressed audio and video. The P and B frames cannot be decoded without a preceding I frame, so that cutting into a stream after an I frame renders the P and B frames meaningless.” (EX1016 at 2:11-19.) Forecast teaches that to avoid such potential decoding errors, the “In Point” or start of a different sequence of frames/pictures is scanned, selected, and split such that it starts with an I frame. (*See, e.g., id.* at 2:11-19, 2:33-50 (SMPTE standard required that the “In Point” of a splice

start with an I frame and no prediction allowed back before the In Point), 12:7-14, 14:40-45 (“Next, in step 122, the splicing procedure finds the closest I frame preceding the desired start frame to be the In Point for splicing.”), 22:53-65 (“There can also be a need for some further reformatting to permit the In Point to be an I frame of an open GOP”).)

B. Ground 1: Claims 1-9 Are Obvious Over Carmel in View of Mattavelli

111. As explained in greater detail below, Claims 1-9 of the ’894 patent are rendered obvious by Carmel in view of Mattavelli.

1. Claim 1 (Independent)

112. As is explained below, claim 1 is obvious over the prior art. For reference, claim 1 recites (with bracketed notations to delineate its limitations):

1. A method comprising:
 - [a.i]** receiving audio and video segments encoded in a digital encoding format and with an encoding rate;
 - [a.ii]** wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;
 - [b.i]** requesting by a multimedia player, transmission of said multimedia objects;
 - [b.ii]** wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

connection;

- [c.i] playing back received multimedia objects by the multimedia player,
- [c.ii] wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,
- [c.iii] autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,
- [c.iv] and by utilizing optimized decoding processes to maintain quality playback.

(’849 patent, 19:19-39 (Claim 1; bracketed notations added).)

(a) “A method comprising:” (claim 1, preamble)

113. The preamble of claim 1 is disclosed by Carmel.

114. Carmel discloses “[a] **method** for real-time broadcasting from a transmitting computer to one or more client computers over a network” (Carmel at Abstract; *see also id.* at 3:24-27 (“There is therefore provided, in accordance with a preferred embodiment of the present invention, a **method** for real-time broadcasting from a transmitting computer to one or more client computers over a network”).) Carmel thus discloses a method.

115. Indeed, as I discuss in detail below, Carmel teaches and suggests

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

methods (and devices and systems for carrying out the disclosed methods) relating to real-time broadcasting and playing of audiovisual data that, alone and in view of the teachings of Mattavelli, teach and render obvious the remaining elements of Claim 1.

(b) “receiving audio and video segments encoded in a digital encoding format and with an encoding rate;” (claim 1[a.i])

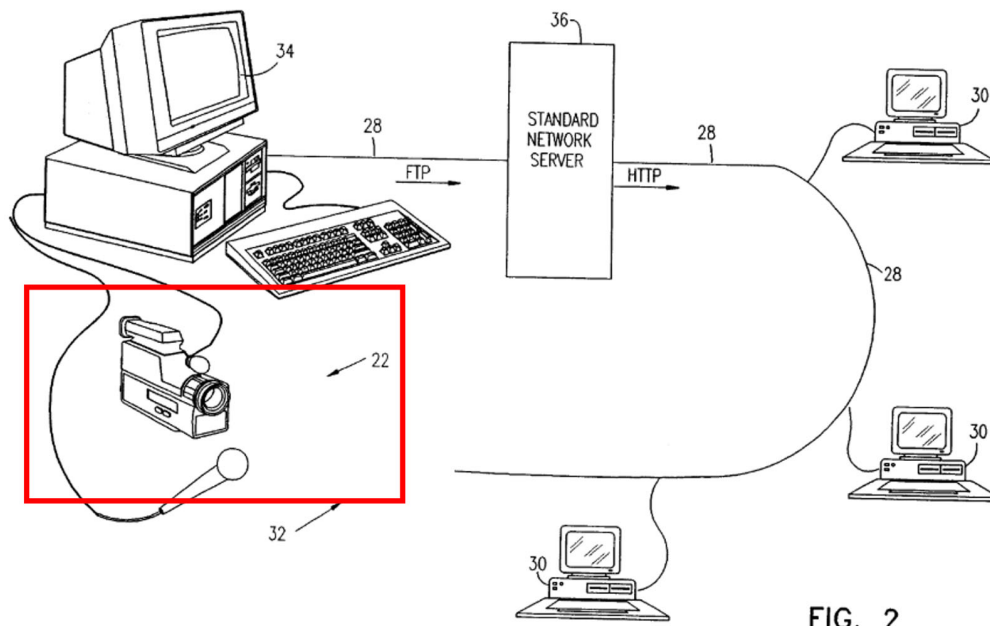
116. Carmel discloses “receiving audio and video segments encoded in a digital encoding format and with an encoding rate.”

117. First, Carmel teaches the creation, transmission, and receipt of “audio and video segments” from a multimedia data stream.

118. As an initial matter, Carmel provides, in accord with what a person of ordinary skill in the art would have understood, that “multimedia data” refers to “data representative of images or sound or a combination thereof” where multimedia image data can (and often does or would) include video data. (Carmel at 2:29-40; *see also* 1:20-22 (“Network broadcasting is most commonly used to stream multimedia data, typically comprising images and sound.”).) For example, Carmel teaches that a multimedia sequence or data stream can be generated from audiovisual input devices 22, which include a video camera (for generating video) and microphone (for generating audio), or can be a prerecorded video and audio

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

sequence. (Carmel at 6:24-35 (“Computer 34 preferably receives audiovisual input from input devices 22, although data inputs of other types may be generated at or by computer 34 using any suitable means known in the art.”), 1:24-28 (“One or more input devices 22 (for example, a video camera and/or microphone) are used to generate a multimedia data stream representing an entertainment or informational program to be transmitted to a plurality of clients 30 via a network 28.”), Figs. 1, 2; *see also id.* at 6:57-60 (“The sequence is preferably generated and compressed in real time, and could comprise, for example, an interview program or an entertainment or sports event, although a prerecorded sequence may similarly be broadcast in this manner.”); 9:64-66 (“Broadcast data are then input to the computer, for example, from input devices 22, or from a video, audio or animation sequence stored on disk or tape.”).)



(Carmel at Fig. 2.)²

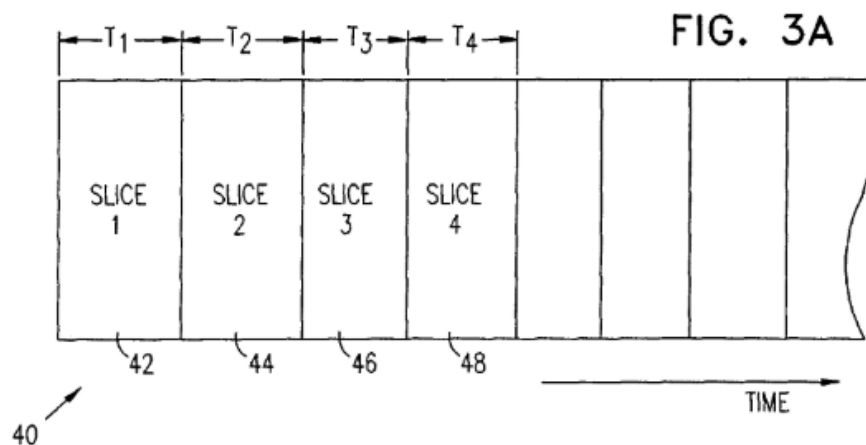
119. Carmel further teaches that, as part of its disclosed process for broadcasting such data, a multimedia data stream that include video and audio data is divided (*e.g.*, by computer 34) into “slices” or segments of video and audio data. For example, in reference to Figure 3A, Carmel provides:

FIG. 3A is a block diagram that schematically illustrates the structure of a stream of broadcast data 40 produced by computer 34, typically corresponding to a multimedia data sequence, in accordance with a preferred embodiment of the present invention. Data stream 40 comprises **a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data**, corresponding to a

² All color annotations have been added unless indicated otherwise.

respective, successive time interval labeled T1, T2, T3, etc.

(Carmel at 7:18-25, Abstract, Fig. 3A.)



(Carmel at Fig. 3A.)

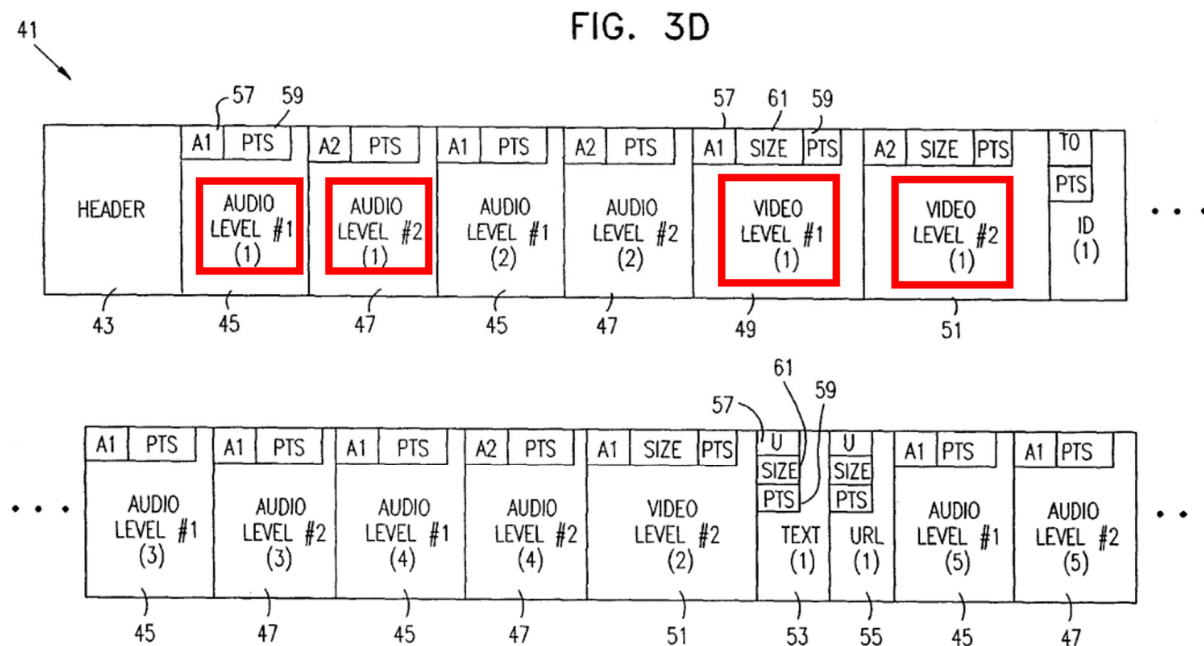
120. Carmel teaches that a multimedia data stream consisting of video and audio data can also be divided into slices or segments having multiple quality levels.

Carmel provides:

FIG. 3D is a block diagram that schematically illustrates a file format of a multi-level data stream 41, in accordance with another preferred embodiment of the present invention. **The data stream is divided into audio slices 45, 47 and video slices 49, 51,** and may also include other data formats, such as a text slice 53 and/or a URL slice 55....

Each time slice in stream 41 includes multimedia data at multiple quality levels. There are two such levels in the example shown in FIG. 3D, identified as level #1 and level #2, but a larger number of levels may also be used.

(Carmel at 8:42-59, Fig. 3D.)



(Carmel at Fig. 3D.)

121. Next, Carmel teaches and suggests that the “audio and video segments” or the sequence/series of slices are “encoded in a digital encoding format with an encoding rate.”

122. Carmel teaches that transmitting computer 34 prepares the input multimedia video and audio data stream prior to dividing it into slices by first encoding or compressing the data. (See Carmel at 2:1-7, 2:41-50, 6:50-7:3, 7:25-26, 9:62-10:1, 11:23-64, Figs. 5 & 7.) For example, referring to Figure 5, Carmel teaches that “[t]he data are **compressed at step 80**, and are then ‘sliced’ at step 82 into files

42, 44, 46, 48, etc., as shown in FIG. 3A.” (Carmel at 9:66-10:1; *see also id.* at 10:19-22 (“Alternatively or additionally, the **encoding/quality level (step 80)** or slicing (step 82) of the data may be adjusted, as described hereinbelow with reference to FIG. 7.”).)

FIG. 5

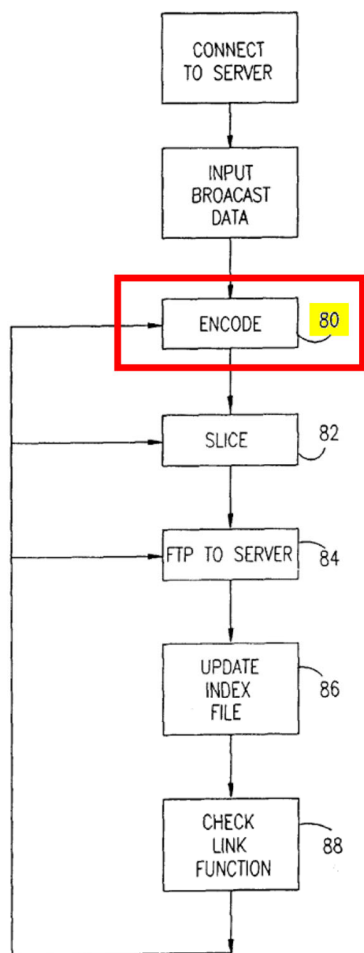
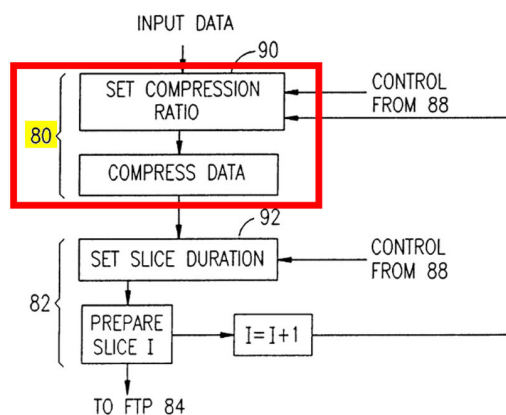


FIG. 7



(Carmel at Figs. 5 & 7.)

123. Carmel further teaches that “[i]n **encoding data stream 40**, computer
34 **preferably compresses the data using any suitable compression method
known in the art.**” (Carmel at 11:23-28.) And, Carmel provides exemplary, well-
known video and audio coding formats for compressing digital audio and video data
that could be used to encode and compress the multimedia data stream. These
include, *e.g.*, GSM 6.10 standard encoding for compressing audio data and H.263
and MPEG coding formats for compressing video and audio data. (Carmel at 11:23-
52.) These standards are digital encoding and compression formats used to compress
digital data so that, for example, it can be transmitted across the Internet using
Internet communication protocols according to the methods taught by Carmel. (*See*,
e.g., Carmel at Abstract, 6:36-7:17, 11:40-52, Fig. 2.)

124. Carmel also teaches and indicates that the audio and video data would
be encoded/compressed with a particular bitrate or encoding rate (often expressed in
terms such as kilobits or kilobytes per second) that corresponds to the compression
rate or ratio used. (*See, e.g.*, Carmel at Abstract; EX1020 at 3, 196, 334, 341, 343-
44.) Carmel repeatedly uses the term “data rate” to refer to the encoding rate or
bitrate of the compressed multimedia data. (*See, e.g.*, Carmel at Abstract (“A method
for real-time broadcasting from a transmitting computer to one or more client
computers over a network, including providing at the transmitting computer a data

stream **having a given data rate**, and dividing the stream into a sequence of slices, each slice having a predetermined data size associated therewith.”), 2:56-59, 3:24-59, 9:31-33, 11:9-22, 14:18-32.) Carmel also teaches that the data rate or bitrate of the compressed data preferably should approximate the available network bandwidth for uploading and/or downloading the audio and video data. (*See, e.g.*, Carmel at Abstract, 3:24-59, 4:49-62, 14:18-32.)

125. Carmel teaches that the compression ratio used by the codec (*e.g.*, an MPEG-4 encoder) can be set or adjusted to achieve this aim of matching the available bandwidth for upload. For example, Carmel provides:

Computer 34 determines a compression ratio by which to compress the data, based on the collective bandwidth of its open links with server 36. Preferably, computer 34 receives an indication of the bandwidths of the links, determined at step 88 in FIG. 5, and adjusts the compression ratio accordingly, at a set compression step 90. For example, the compression ratio may be adjusted by changing compression coefficients (*e.g.*, MPEG coefficients) so as to match the data stream bandwidth to the available link bandwidth.

(Carmel at 11:39-52.) Carmel also teaches that the compression ratio can be varied to match available bandwidth for download by clients 30. (Carmel at 7:44-49 (**“Alternatively or additionally, the compression level of the data is varied, as is likewise described below, so as to adjust the data streaming rate to the available**

bandwidth over one or more channels between computer 34 and server 36, **and/or**
between server 36 and client 30.”.)

126. And, as discussed more below, Carmel teaches that a particular slice or segment or audio or video data can be encoded or compressed at multiple, different quality levels or data rates, such that the client computer can choose what quality level is most suitable for playback in view of the available bandwidth for download. For example, Carmel provides:

In other preferred embodiments, the slices are provided by the server at **multiple resolution or quality levels. Each such level has a different degree of data compression, and thus corresponds to a different data bandwidth requirement.** The client or the server monitors the data transfer rate of a data link opened [between them] and selects the level that is appropriate to the link bandwidth. If the monitored data transfer rate changes during transmission, the quality level is preferably reselected accordingly.

(Carmel at 3:5-13; *see also id.* at 4:39-47, 5:15-18, 8:42-9:9, 10:19-22, 10:64-11:22.)

Carmel thus teaches and suggests (in accord with the understanding of an ordinarily skilled artisan) that the quality level of an encoded segment of audio or video data corresponds to its encoding/compression/data rate or bitrate with higher compression ratios generally resulting in lower bitrates and lower quality levels. Carmel teaches that the techniques it provides for ensuring that the data- or

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

encoding-rate of the sequence of slices matches the available network bandwidth can be used separately or combined and used together. (*See, e.g.*, Carmel at 3:40-50, 4:39-47, 4:63-5:18, 7:36-49, 9:49-52 (“**Alternatively or additionally**, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36.”), 10:19-23 (“**Alternatively or additionally**, the encoding/quality level (step 80) or slicing (step 82) of the data may be adjusted, as described hereinbelow with reference to FIG. 7.”).)

127. Finally, Carmel also teaches the “receiving” of such audio and video segments or slices. For example, Carmel teaches that “[c]lients 30 connect to server 36 and **receive the multimedia sequence** [of slices], substantially in real time[,]” which is played by an applet or multimedia player (Carmel at 7:4-5, 7:12-17, 10:24-34, 14:61-64.) Carmel discloses that the clients 30 receive audio and video slices/segments from server 36.³ (*See, e.g.*, Carmel at 10:64-11:22 (“Responsive to

³ Carmel also discloses that encoded audio and video segments are also received by the server 36 (from transmitting computer 34) to facilitate subsequent broadcast to downstream clients 30. (*See, e.g.*, Carmel at Abstract, 2:1-21, Figs. 2, 4-5.) Carmel also teaches that transmitting computer 34 receives a multimedia sequence from, *e.g.*, input devices 22, and encodes (or transcodes) it into a digital encoding format, slices the multimedia sequence into a series of objects, and associates the slices with object parameters and path identification. (*See, e.g.*, Carmel at Abstract, 6:24-35, 6:50-7:3, Figs. 2, 5, 7-9.)

the selection, server 36 begins to transmit data slices at the chosen quality level. **The slices are received**, decoded and output by the client.”), Figs. 2, 3D and 6B.)

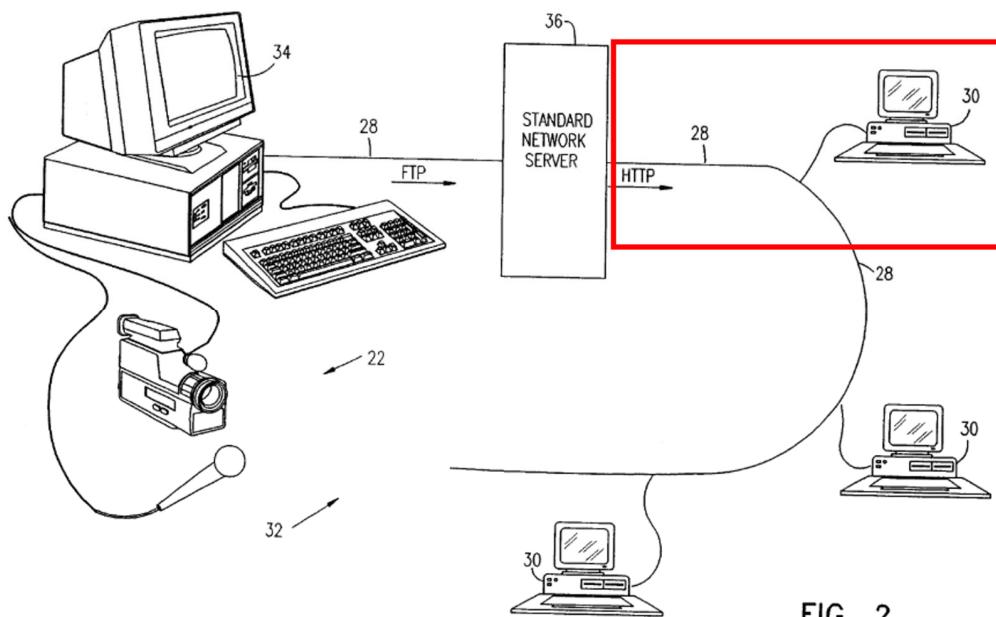


FIG. 2

(Carmel at Fig. 2.)

128. Carmel thus teaches and renders obvious receiving (*e.g.*, by a client 30) audio and video segments (*e.g.*, a sequence of “slices” of a multimedia stream of audio and video data) that are encoded in a digital encoding format (*e.g.*, H.263, MPEG-4, or other suitable compression standards) and with an encoding rate (*e.g.*, a “data rate” or bitrate that corresponds to the quality level and amount by which the video and audio data has been compressed).

(c) **“wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;” (claim 1[a.ii])**

129. Carmel discloses and renders obvious “wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects.” Carmel in view of Mattavelli also renders obvious this claim element.

130. First, Carmel discloses that the “slices” of audio and video data created by transmitting computer 34 and uploaded to server 36 for broadcast and download by network clients 30 form “multimedia objects.” Carmel teaches:

In preferred embodiments of the present invention, a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients. **The data stream is divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed. Each slice is preferably assigned a respective slice index.** The transmitting computer uploads the sequence of slices to the server substantially in real time, preferably using an Internet protocol, most preferably the File Transfer Protocol (FTP), as is known in the art. The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP) The division of the data stream into slices and the inclusion of the slice indices in the data stream to be used by the clients in maintaining

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

synchronization allows the broadcast to go on substantially in real time without the use of special-purpose hardware.

(Carmel at 2:1-21; *see also id.* at Figs. 2, 3A, 3D, 5, 6B.) Carmel further provides that the data stream which is segmented into discrete objects/components/slices preferably consists of multimedia data. (Carmel at 2:29-41 (“In some preferred embodiments of the present invention, the data stream comprises multimedia data captured or generated by the transmitting computer.”).)

131. Carmel also teaches that the slices of a multimedia data stream can be saved as a corresponding sequence of discrete files. (Carmel at Abstract, 2:22-23 (“Preferably, each segment or slice is contained in a separate, respective file.”); *see also id.* at 3:24-39, 5:15-18 (“In a preferred embodiment, the slices are encoded at a plurality of different quality levels, such that **the files corresponding to a given one of the slices** have a different respective data size for each of the quality levels.”), 14:18-32, 15:8-17.) Based on these teachings from Carmel, each respective slice or corresponding multimedia file of audio and video data would have been understood to be a “multimedia object.” (*Compare, e.g.,* Carmel at Abstract, 2:1-41, 5:15-18, Figs. 3A, 3D, & 5, *with* ’849 patent at 3:7-9 (“The multimedia object creator produces discrete multimedia objects from video and audio segments of a continuous stream.”), 3:2-6 (“As multimedia objects they can then be dynamically uploaded to

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

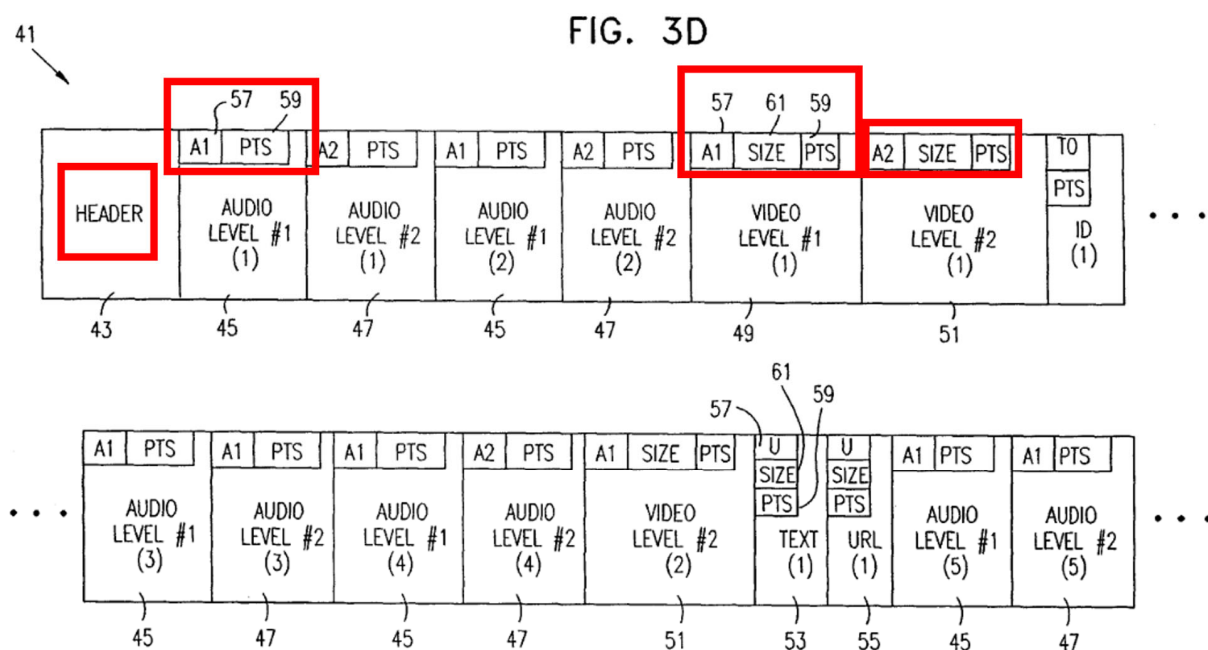
multiple live content hosting web servers, which, through proximate mobile network proxy servers, make live content accessible to mobile clients as consecutive multimedia objects.”.)

132. Next, Carmel further discloses that the audio and video segments or “slices” are associated with object parameters. Carmel teaches that each slice or corresponding file includes certain metadata or parameters pertaining to that slice, including, *e.g.*, a slice index identifier and one or more time stamps/time codes. For example, Carmel provides:

Data stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T_1 , T_2 , T_3 , etc. The data are preferably compressed, as described further hereinbelow. Computer 34 stores each slice as a corresponding file, **having a running slice index 1, 2, 3 . . . N**. Preferably, **each file also includes one or more time stamps**, indicating a real time at which the data in the file were recorded or an elapsed time relative to the beginning of stream 40.

(Carmel at 7:23-31, Fig. 3A; *see also id.* at 2:6-7 (“Each slice is preferably assigned a respective slice index.”), 7:36-49, 3:24-39, 14:18-32, 15:8-17.) For multi-level data streams, Carmel teaches that, in addition to a slice identifier 57 and time stamp 59 (“presentation time stamp (PTS)”), each audio and video slice/file can also include “**size identifier 61**,” as well as a header 43 that “includes data such as the title,

author, copyright and **formats of the data in the stream; the duration of the multimedia sequence represented by the stream; and a description of the available stream levels and associated data sizes.** (Carmel at 8:42-55.) I have highlighted some of these parameters/metadata described by Carmel in Carmel Figure 3D reproduced below.



(Carmel at Fig. 3D.)

133. Carmel teaches and renders obvious embodiments of a multi-level data stream where the time slices of multimedia data at multiple, differing quality levels (and the object parameters (*e.g.*, slice identifiers, time stamps, and size/quality identifiers) associated with each slice) are saved within a single file. (*See, e.g.*,

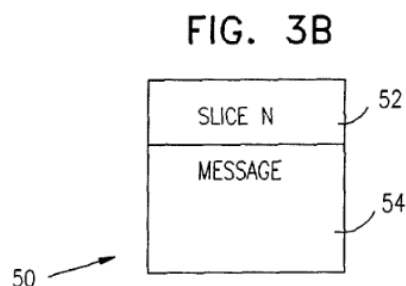
Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Carmel at 8:42-9:5, Fig. 3D.) Carmel also teaches and renders obvious embodiments of a multi-level data stream where each time slice of multimedia data at each differing quality levels (and the object parameters (*e.g.*, slice identifiers, time stamps, and size/quality identifiers) associated with each slice) are saved and uploaded to host server 36 (for download by clients 30) as separate, discrete files. (*See, e.g.*, Carmel at 5:15-18 (“In a preferred embodiment, the slices are encoded at a plurality of different quality levels, **such that the files corresponding to a given one of the slices** have a different, respective data size for each of the quality levels.”); *see also id.* at Abstract, 2:22-23 (“Preferably, each segment or slice is contained in a separate, respective file.”), 14:19-35, 15:8-17, 9:10-10:5, 12:6-12.) Storing multi-level slices of different quality levels as separate files would be particularly suitable and obvious when using the multiple transmission link techniques taught by Carmel. (Carmel at 9:49-52 (“Alternatively or additionally, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36.”).) It further would have been obvious for each file (corresponding to each slice of video or audio data at a particular quality level) to have a header containing any necessary or helpful information/parameters (*e.g.*, a slice index identifier, time stamps, encoding/decoding metadata) for ensuring synchronized

playback in the correct sequence. (*See, e.g.*, Carmel at 2:15-21, 8:47-55, 10:45-54.)

Either way—that is, whether the slices are saved within a single file or as separate files—each audio and video slice has associated object parameters (and, as I discuss below, host path identification) and thus would be understood to form a multimedia object.

134. Carmel also teaches that the audio and video files/slices are accompanied by an index file that contains additional metadata and parameters associated with the audio and video segments. “The index file comprises a slice ID 52, indicating the index of the file in data stream 40 that was most recently uploaded by computer 34. Each time a new file 42, 44, 46, etc., is uploaded, ID 52 in file 50 on server 36 is updated. Preferably, ID 52 holds the file name of the new file, wherein the name typically comprises a string followed by the index of the file.” (Carmel at 7:59-8:1.) Carmel further provides: “Index file 50 may further include a **message 54**, which is read by computers 30 when they connect to server 36 to download data stream 40 or, alternatively or additionally, at any time the message is updated by computer 34. **The message contains parameters relating generally to the data stream and/or instructions to computers 30**” (Carmel at 8:12-17.)



(Carmel at Fig. 3B.) Carmel teaches and suggests that index files can be used with multi-level data stream slices. (*See, e.g.*, Carmel at 14:19-35, 15:8-17.) Further, in view of the similarity of their description and purpose in Carmel, as well as the knowledge in the art, a person of ordinary skill in the art would have understood and found it obvious that these configuration parameters and instructions contained in an index file could additionally or alternatively be included in the header of a multi-level multimedia object/slice/file, as described with reference to Carmel Figure 3D. (*See, e.g.*, Carmel at 8:9-11 (“Further alternatively, stream 40 may be multicast to clients 30, as is known in the art, typically without the use of an index file.”), 8:12-17, 8:47-55, Fig. 3D.) This would be nothing more than the combination of prior art methods according to known methods or a simple substitution of one known element (*e.g.*, use of an index file with parameters and configuration information) for another known element (*e.g.*, use of a header to a multimedia file with parameters and configuration information) to obtain predictable results (*e.g.*, configuring the multimedia player for playback of the sequence of multimedia objects in a manner

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

that maintains playback fluidity and video and audio quality). (*See also, e.g.*, EX1024 at 2:16-3:4 (describing techniques for computational graceful degradation employed by a decoder that utilizes parameters associated with audiovisual objects including parameters relating to the decoding complexity requirements in the headers of audiovisual packets).)

135. In view of these teachings, a person of ordinary skill in the art would have considered this information disclosed by Carmel (*e.g.*, slice identifiers, time stamps, size identifiers, format information, available stream levels and associated data sizes, general parameters related to the data stream, and instructions to computer 30) to be parameters associated with the audio and video segments / “slices.”

136. Further, as I previously discussed, Carmel also discloses that “[i]n **encoding data stream 40**, computer 34 **preferably compresses the data using any suitable compression method known in the art[,]**” and provides exemplary, well-known coding/compression standards, such as H.263 and MPEG (*e.g.*, MPEG-2, MPEG-4), for compressing digital audio and video data that could be used to encode and compress the multimedia data stream. (Carmel at 11:23-52.) Carmel thereby teaches and renders obvious the use of well-known digital compression standards

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

that were available in the art, including, for example, MPEG-4 (visual).⁴ And, as would have been understood by one of ordinary skill in the art, the process of using a standard like MPEG-2 or MPEG-4 to encode/compress video data embeds or includes certain metadata/parameters (such as frame size, resolution, audio sampling rate, etc.) and path identifiers used in decoding/decompressing the data. (*See, e.g.*, EX1005 at 8-9, 13, 20, 38, 49-51, 54 (“MPEG can handle various pixel aspect ratios and allows a control code to be embedded in the sequence header to help the decoder.”), 273 (“Several GOPs form a sequence which begins with a sequence header containing important data to help the decoder.... The sequence header describes the MPEG-2 profile and level, whether the video is progressive or interlaced, whether the chroma is 4:2:0 or 4:2:2, the size of the picture and the aspect ratio of the pixels. The quantizing matrix used in the spatial coder can also be sent.”); *see also id.* at 27-28 (“A transport stream has a complex structure because it needs

⁴ As I discuss further below for claim element 1[b.ii], Carmel teaches and renders obvious applying its techniques in the context of a mobile client using a wireless connection. Based on Carmel’s teachings and his/her knowledge, a person of ordinary skill further would have found it obvious and would have been motivated to employ MPEG-4 in this context, as this standard was designed to be particularly suitable and useful for wireless/mobile applications. (*See, e.g.*, EX1021 at 4-7, 49; *see also* EX1027 at 657 (“[C]ompression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC [MPEG-4 Part 10] coding an attractive candidate for all wireless applications including multimedia messaging services (MMS), packet-switched streaming services (PSS), and conversational applications.”).)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

to incorporate metadata indicating which audio elementary streams and ancillary data are associated with which video elementary stream.”), 329-34; EX1021 at 44-45 (MPEG-4 coded data can contain a number of elementary streams including “scene description data, control data in the form of object descriptors, or metadata describing the content”), 47 (object descriptors “ODs provide information such as the elementary stream available to represent a given media object, the characteristics of the decoders needed to consume the elementary streams, **and the location (possibly a URL) of the elementary streams data.**”), 65-66 (describing object descriptors in MPEG-4); 72-75 (“Each ESD [elementary stream descriptor] provides detailed information about the associated stream, **including its location, type, encoding, bit rate, buffer requirements, and decoder specific information.**”), 76-81, 86-87; EX1028 at 3, 19-20 (frame size and resolution), 103.)

137. Mattavelli teaches that the MPEG-4 video standard includes the option to include decoding “complexity prediction information” along with the video information itself. Mattavelli provides that the computational graceful degradation approach for a video decoder it teaches “**can be virtually applied to any video compression standard and processor based platform**, although the best performances are achieved when **complexity prediction information provided as side information by the video standard** is available. **This option has been**

included in the new [circa 1998] MPEG-4 standard.” (Mattavelli at 2309; *see also id.* at 2311 (“The second strategy is to exploit the **complexity information carried by the frames or VOP headers, like it can be done form MPEG-4 systems [6],** and use it in two stages.”); *see also, e.g.*, EX1024 at 2:16-3:4 (describing including “[p]arameters relating to the [decoding] computational complexity requirements” in the headers of audiovisual packets); EX1021 at 77-79 (describing content complexity descriptors used in MPEG-4).) This type of complexity information provided as “side information” by the MPEG-4 video standard would also be understood as a POSA to comprise object parameters associated with the compressed audio and video segments/slices.

138. A person of ordinary skill in the art would have found it obvious and would been motivated to combine the teachings of Carmel and Mattavelli for the reasons I provide in more detail below in connection with claim element [1.c.iv.], which I incorporate here. Moreover, in view of Carmel’s teaching that MPEG data compression could be used it its disclosed process (Carmel at 11:23-52), it would be obvious that features and options included with the known MPEG-4 coding standard(s) (and identified by Mattavelli) could easily be added to, or were already present in, the compressed slices of video and audio data taught by Carmel. And, in connection with claim element [1.c.iv.] below, I explain why a person of ordinary

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

skill in the art would have found it advantageous and obvious to utilize this complexity information data in Carmel's system in view of Mattavelli's teachings.

139. Next, Carmel discloses and renders it obvious that the audio and video segments/slices are also supplied with "host path identification."

140. As previously discussed, Carmel teaches and suggests that the audio and video slices are preferably saved as a series of separate, corresponding files. (Carmel at 3:24-39, 5:15-18, 7:23-31 ("Computer 34 stores each slice as a corresponding file, having a running slice index 1, 2, 3 . . . N.").) Carmel further teaches that the multimedia slices/files are uploaded, preferably using file transfer protocol (FTP), by the transmitting computer 34 to network server 36. (Carmel at 6:50-7:3 ("After preparing the multimedia sequence, computer 34 uploads the sequence over network 28, preferably using the Internet File Transfer Protocol (FTP)."), 9:10-10:23 ("Alternatively or additionally, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36."), Figs. 2, 4-5.)

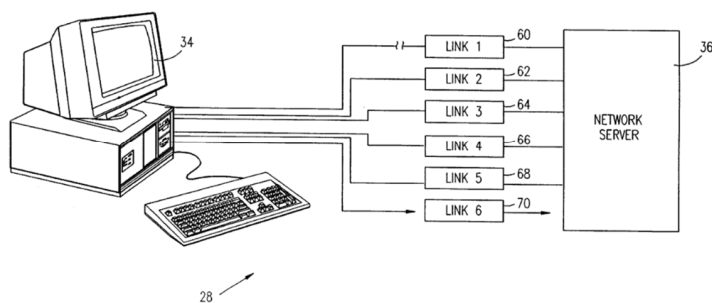


FIG. 4

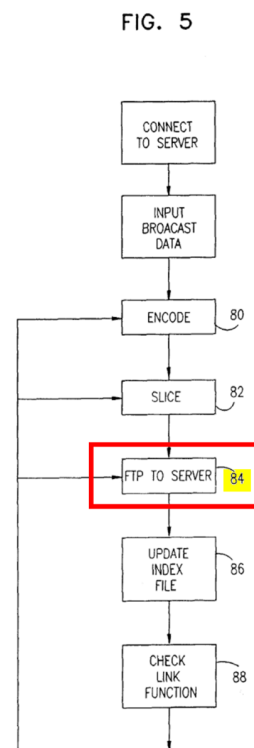


FIG. 5

(Carmel at Figs. 4 & 5.) As shown above in Figure 4, Carmel states that for the purpose of uploading the slices/files via FTP to the server 36, “the computer preferably opens a plurality of links, **each link having its own URL (Uniform Resource Locator).**” (Carmel at 9:17-19; *see also id.* at 9:46-47 (stating that “[a] similar technique is preferably employed by clients 30” for downloading the files).)

141. Once uploaded, Carmel teaches that the server 36 then hosts the files/slices for broadcast to and download to one or more clients 30. Carmel states:

Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. Clients 30 preferably download the sequence using the Hypertext Transfer Protocol

(**HTTP**), although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34. **Since FTP and HTTP are supported by substantially all network servers, server 36 need not include any special-purpose broadcasting hardware or software**, as noted above. Similarly, because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast.

(Carmel at 7:4-17; *see also id.* at 10:36-42.) Carmel similarly provides:

As in the method of FIG. 6A, **each client 30 connects to the server, generally using a single HTTP link**. After reading header 43 and, preferably, making an initial assessment of the link bandwidth, **the client selects one of the available quality levels in the stream. Responsive to the selection**, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.

(Carmel at 10:67-11:8.) Carmel provides: “In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user.”

(Carmel at 10:48-50.)

142. Carmel’s teachings regarding the upload and download of the multimedia slices/files of audio and video data using uniform resource locators

(URLs) and standard Internet protocols, including FTP and HTTP, to do so, discloses the supplying of host path identification to the respective audio and video slices/files—that is, the identification of where on the host server (*e.g.*, in what directory) each respective file in the sequence is to be saved in the case of upload from the transmitting computer 34 and is to be served from in the case of download/broadcast to one or more clients 30.

143. Uniform resource locators (URLs) “are the standardized names for the Internet’s resources. URLs point to pieces of electronic information, telling you where they are located and how to interact with them.” (EX1006 at 23.) URLs include several components including a “path” component, which “specifies where on the server machine the resource lives. The path often resembles a hierarchical filesystem path.” (*Id.* at 28.) In the example URL <<http://www.joes-hardware.com:80/seasonal/index-fall.html>>, the path is “‘/seasonal index-fall.html’”, which resembles a filesystem path on a Unix filesystem. The path is the information that the server needs to locate the resource” and can be divided into segments separated by a slash “/.” (*Id.*) URLs used for FTP and HTTP are similar, except that each is headed by a different scheme (that is, “ftp” vs. “http” before the first colon “:” in the URL. (*Id.* at 27.) I have reproduced exemplary URLs using HTTP and FTP schemes below. Note that both include server host path identification

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

information.

Scheme	Description
http	The Hypertext Transfer Protocol scheme conforms to the general URL format, except that there is no username or password. The port defaults to 80 if omitted. Basic form: <code>http://<host>[:<port>]/<path>?<query>#<frag></code> Examples: <code>http://www.joes-hardware.com/index.html</code> <code>http://www.joes-hardware.com:80/index.html</code>
Scheme	Description
ftp	File Transfer Protocol URLs can be used to download and upload files on an FTP server and to obtain listings of the contents of a directory structure on an FTP server. FTP has been around since before the advent of the Web and URLs. Web applications have assimilated FTP as a data-access scheme. The URL syntax follows the general form. Basic form: <code>ftp://<user>[:<password>]@<host>[:<port>]/<path>[:<params>]</code> Example: <code>ftp://anonymous:joe%40joes-hardware.com@prep.ai.mit.edu:21/pub/gnu/</code>

(EX1006 at 38-39; *see also* EX1023 at 528-29 (URL “*path* is a string that identifies one particular document on the server”); EX1022 at 413-14 (host path identifiers in URLs).)⁵

144. Further, as discussed above, the metadata supplied by the compression codec used also contains information identifying the location of information within

⁵ I note that Patent Owner has indicated that that URLs constitute host path identifiers in its infringement allegations included in its complaint in the related litigation against Google. (*E.g.*, EX1055 at 20 (identifying “BaseURL” as a host path identifier); EX1055, Ex. 2 at 15 (stating “[t]he multimedia multivariant playlist defines several segment streams that are also associated with host path identifiers, such as URLs”), *id.* at 17; EX1055, Ex. 3 at 15, 17.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

the compressed multimedia data. (See EX1021 at 47 (object descriptors “ODs provide information such as the elementary streams available to represent a given media object, the characteristics of the decoders needed to consume the elementary streams, **and the location (possibly a URL) of the elementary streams data.**”), 65-66 (describing object descriptors in MPEG-4); 72-75 (“Each ESD [elementary stream descriptor] provides detailed information about the associated stream, **including its location, type, encoding, bit rate, buffer requirements, and decoder specific information.**”).)

145. To the extent that Patent Owner asserts that the Carmel disclosure concerning URLs and the upload and download of slices/files to and from the server 36 using FTP and HTTP discussed above does not expressly or implicitly disclose the supply of host path identification to the audio and video segments/slices/files, these teachings, at minimum and in view of the knowledge of a person of ordinary skill in the art, fairly suggest and render obvious the supply of host path identification. Indeed, it would be obvious and beneficial to supply host path identification to the audio and video slices/files, as the path identification in a URL when uploading them via FTP so that they are saved in the correct location on the host server and so that these respective files can be correctly identified and located using HTTP when a client 30 is requesting transmission of the files from the server.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(EX1006 at 28 (the “path” component “specifies where on the server machine the resource lives.”); Carmel at 9:49-52 (“Alternatively or additionally, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36.”).) A person of ordinary skill would have been familiar with these basic well-known features of URLs, FTP, and HTTP, and would have found it a straightforward, routine, predictable, and obvious application of these features to supply host path identification to the multimedia files so that they could/would be uploaded and downloaded correctly. This would be nothing more than using known techniques to yield predictable results—that is, the ability to identify and locate the resources/files hosted by the server correctly using these well-known protocols and uniform resource identifiers.

146. Because Carmel discloses and renders obvious for the reasons just discussed the disclosed audio and video segments/“slices” having associated object parameters (*e.g.*, index identifiers, time stamps, size identifiers, quality level, and other control and coding information) and supplying host path identification (*e.g.*, an identification of a host and path information as part of a URL) to the respective slices/files on the host server 36, a person of ordinary skill in the art would understand the audio and video slices/files disclosed by Carmel to form and to be

“multimedia objects,” also as discussed above, and satisfy this claim element. Carmel, alone and in view of Mattavelli, discloses and renders obvious “wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects.”

147. Carmel, and Carmel in view of Mattavelli, thus teach and render obvious wherein, said audio and video segments (*i.e.*, slices/sequence of files of audio and video data, including slices encoded at multiple, different quality levels) are associated with object parameters (*e.g.*, slice indices, time stamps, size/quality level identifiers, “parameters relating generally to the data stream and/or instructions to computers 30” (*see* Carmel at 8:12-17), and/or information associated with encoding/decoding the compressed data), and supplied host path identification (*e.g.*, URLs including host path information for the slices/files so that these resources can be properly located, requested, and received using standard Internet protocols) to form multimedia objects (*e.g.*, slices/segments/a sequence of files of audio and/or video data with associated object parameters and host path identification).

(d) “requesting by a multimedia player, transmission of said multimedia objects;” (claim 1[b.i])

148. Carmel discloses and renders obvious “requesting by a multimedia player, transmission of said multimedia objects.”

149. First, Carmel discloses and renders obvious a multimedia player.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Carmel teaches that the client computers 30 are capable of playing back a multimedia data stream from a server 36 that has been prepared and uploaded according to the methods taught by Carmel. (*See, e.g.*, Carmel at 3:66-4:2 (“Preferably, the one or more client computers decode the sequence and play back the data stream responsive to the indices of the files, at a replay rate generally equal to the data rate.”), 2:15-17 (“The clients use the slice indices of the frames to maintain proper synchronization of the playback.”), 4:3-7, 5:22-25, 10:24-11:22.) Carmel expressly provides: “[i]n the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user.” (Carmel at 10:47-50; *see also id.* at 11:5-8 “Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.”).)

150. Carmel teaches that the multimedia player that receives, decodes, and plays/outputs the sequence of multimedia objects encoding the broadcasted multimedia data stream can be a Java applet or plug-in to the client’s web browser. (Carmel at 7:12-17 (“Similarly, because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast.”), 10:27-35 (“**The operation of client is controlled by a Java applet**, which may be downloaded from server 36, and

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

includes facilities for carrying out the steps shown in FIG. 6A [and 6B], as well as for error detection and, optionally, correction in communications received by the clients and for other functions known in the art. A sample applet of this sort is incorporated herein in the software appendix”), 10:64-11:2 (“FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D) transmitted from server 36, in accordance with another preferred embodiment of the present invention. **As in the method of FIG. 6A**, each client 30 connects to the server, generally using a single HTTP link.”). Carmel provides software for a “sample applet” that can be used to download, decode, and play back data stream 40. (Carmel at 10:33-34, 13:53-14:16.) Carmel thus discloses a media player, such as a Java applet, for receiving, decoding, and playing/outputting a multimedia data stream.

151. These same teachings also render obvious having a media player on the client for performing these functions and “other functions known in the art.” (Carmel at 10:27-35.) Further, in view of these teachings and unless the context indicates otherwise, a person of ordinary skill in the art would understand and interpret Carmel’s references to configurations of or actions taken by the “client 30” or “client computer 30” as configurations of or actions taken by the multimedia player (*e.g.*, a

Java applet) on client computer 30.⁶ (*E.g.*, Carmel at 10:27-28, 10:64-11:22.) At a minimum, these teachings reasonably suggest and render obvious such an interpretation.

152. Next, Carmel further teaches and suggests that the media player on client 30 requests transmission of the multimedia objects (that is, the series of slices of multimedia video and audio data uploaded to server 36 for broadcast). Again, Carmel teaches a multimedia player (such as a Java applet in an exemplary embodiment), and it states that: “[t]he **operation of [the] client is controlled by [the] Java applet**”—i.e., by the multimedia player. (Carmel at 10:27-28.)

153. Carmel further teaches that the client (and multimedia player thereon) chooses and requests transmission of the desired multimedia objects, including, *e.g.*, the multimedia objects corresponding to a particular quality level in the variable bitrate/multi-level data stream embodiment taught by Carmel. For example, Carmel provides:

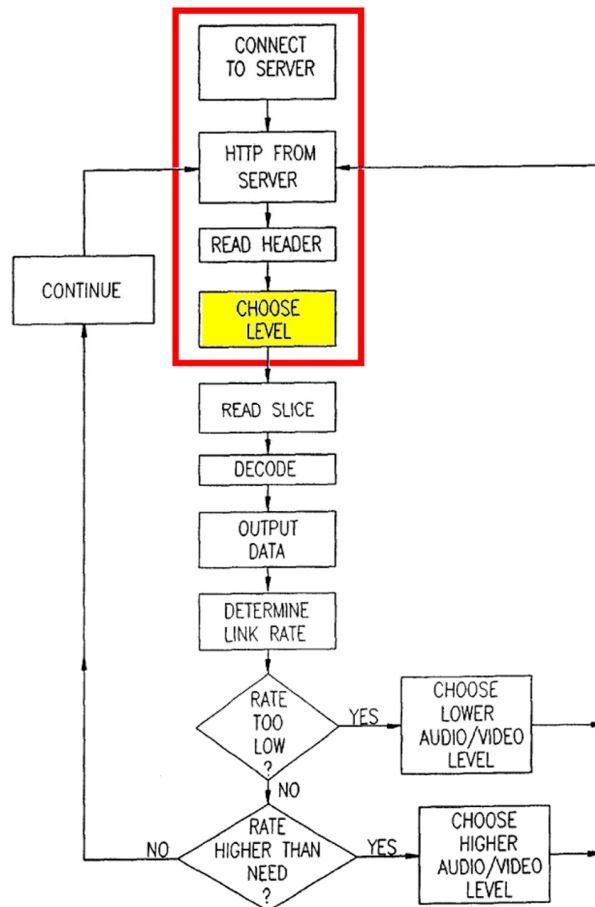
FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D)

⁶ Similarly, in this declaration, I sometimes refer to “client 30” or “client computer 30” as a shorthand way of referring to the multimedia player taught and suggested by Carmel on client computer 30. At other times, I refer to the multimedia player using a lengthier formulation. No implications are intended and no inferences should be drawn from which formulation I use.

transmitted from server 36, in accordance with another preferred embodiment of the present invention. **As in the method of FIG. 6A, each client 30 connects to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an initial assessment of the link bandwidth, *the client selects one of the available quality levels in the stream. Responsive to the selection,*** server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.

(Carmel at 10:64-11:8; *see also id.* at 11:9-22 (“For example, if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, **the client will preferably select** a lower quality level if one is available. On the other hand, if the rate is substantially higher than what is needed to receive the successive slices on time, **the client may select** a higher quality level to take advantage of the available bandwidth.”).)

FIG. 6B



(Carmel at Fig. 6B.)

154. More generally, Carmel repeatedly teaches and suggests that the client 30 (and accordingly the multimedia player thereon) requests the transmission of the desired multimedia objects from the server 36. (*See, e.g.*, Carmel at 2:11-15 (“The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP), or

alternatively, using other protocols, such as UDP or RTP, which are similarly known in the art.”), 3:8-11 (“**The client** or the server monitors the data transfer rate of a data link opened therebetween and selects the level that is appropriate to the link bandwidth.”), 7:4-9 (“Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34.”), 10:35-47 (“Each client 30 connects to server 36, optionally using multiple HTTP links, in a manner similar to that shown and described above with reference to FIG. 4. Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation, but as in the case of transmitting computer 34, a greater number of links may similarly be opened. The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, **so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.**”), Fig. 6A.) Carmel’s teachings regarding the use of HTTP by clients to download the desired files implicitly discloses and renders obvious the use of HTTP GET requests to initiate this transmission. (*See,*

e.g., EX1006 at 4-5, 8-9.)

155. Through these teachings, Carmel thus discloses and renders obvious requesting by a multimedia player (*e.g.*, a Java applet that controls the operations of the client), transmission of said multimedia objects (*e.g.*, requesting transmission of a sequence of multimedia slices (with associated object parameters) encoded at a selected quality level using standard Internet protocols such as HTTP).

(e) **“wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;” (claim 1[b.ii])**

156. Carmel discloses and renders obvious “wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection.”

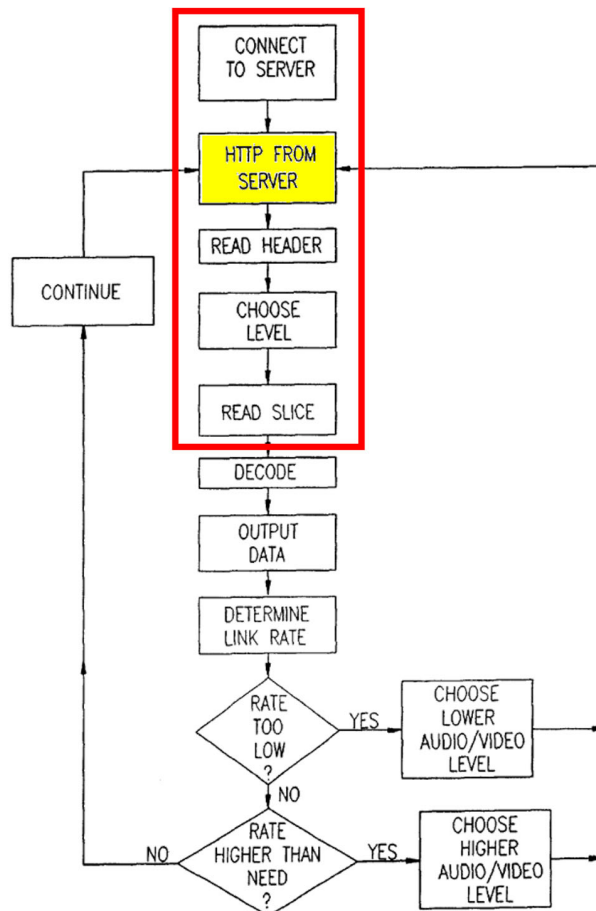
157. First, Carmel teaches and renders obvious locating and downloading the multimedia objects (that is, the slices and/or files of segmented audio and video data taught by Carmel) using HTTP. As previously discussed and incorporated here, Carmel teaches that the clients 30 (and accordingly the multimedia players thereon) preferably use the well-known HTTP protocol to locate and download the desired slices of multimedia data, including in the multi-level data stream embodiment described in connection with Figs. 3D and 6B. For example, Carmel provides:

FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D)

transmitted from server 36, in accordance with another preferred embodiment of the present invention. **As in the method of FIG. 6A, each client 30 connects to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an initial assessment of the link bandwidth, the client selects one of the available quality levels in the stream. Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level.** The slices are received, decoded and output by the client.

(Carmel at 10:64-11:8; *see also id.* at 11:9-22.)

FIG. 6B



(Carmel at Fig. 6B; *see also id.* at Fig. 3D.)

158. Indeed, Carmel repeatedly teaches that the clients preferably use HTTP to connect to the server and locate and download the desired sequence of multimedia slices. (*See, e.g.*, Carmel at 2:11-15 (“**The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP), or alternatively, using other protocols, such**

as UDP or RTP, which are similarly known in the art.”), 7:4-9 (“Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. **Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP)**, although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34.”), 10:35-47 (“**Each client 30 connects to server 36, optionally using multiple HTTP links, in a manner similar to that shown and described above with reference to FIG. 4.** Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation, but as in the case of transmitting computer 34, a greater number of links may similarly be opened. **The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.**”), Figs. 2, 4, 6A.)

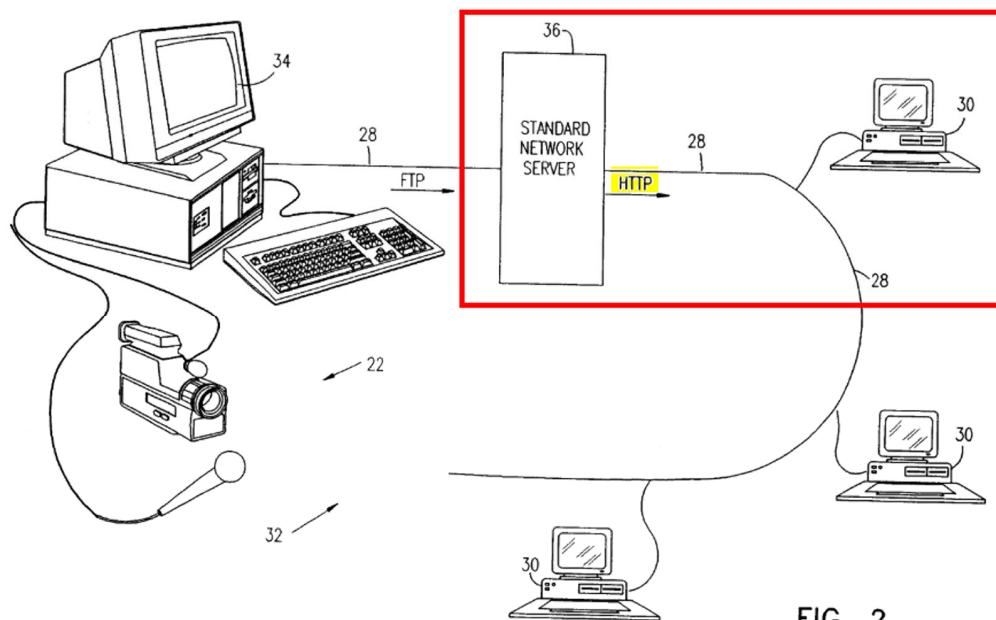


FIG. 2

(Carmel at Fig. 2.)

159. In addition to disclosing the use of HTTP to locate the desired multimedia objects to download, these teachings from Carmel also render this functionality obvious. It would have been obvious to a person of ordinary skill in the art that HTTP commands are involved in all the steps in outlined in red in the Carmel Figure 6B reproduced above. For example, a person of ordinary skill in the art would have understood that HTTP was used by a client 30 to connect to the server 36, to locate, identify and receive the header that is read, and to identify, locate, and receive the slice(s) that are read and downloaded (where they are subsequently decoded and played (“output”) by the client multimedia player). A person of ordinary skill in the art would have further understood that in order to download a particular slice/file of

multimedia data, that particular slice/file must first be identified and located, and it would have been obvious to use of HTTP with uniform resource locators (URLs) to carry out this functionality, as is taught and suggested by Carmel (*see* Carmel at 9:13-27, 9:46-47, 10:35-47, 10:64-11:8) and as was known by those of ordinary skill in the art. (*See, e.g.*, EX1006 at 4-9, 23-29, 38-39 (use of HTTP with URLs to identify and request resources), 120-126; EX1022 at 415.)

160. Next, Carmel also teaches and renders obvious having the multimedia objects (that is, the slices and/or corresponding files of video and audio data) received from servers (*e.g.*, server 36) by the multimedia player (*e.g.*, a Java applet or other multimedia player on a client 30) using a wireless connection.

161. Carmel teaches that client 30 connects to host server 36 via a network 28. Carmel illustrates these connections in Figure 2.

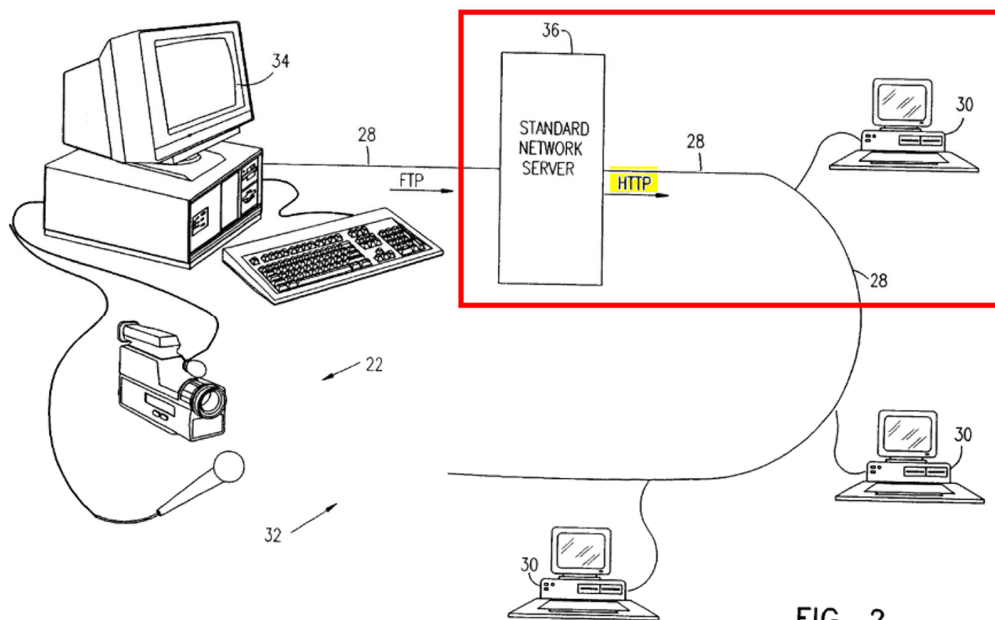


FIG. 2

(Carmel at Fig. 2.) Carmel provides: “Network 28 preferably comprises the Internet, although it may equally comprise a LAN, WAN, intranet or other computer network as is known in the art.” (Carmel at 6:36-38.)

162. Carmel’s teaching of a generic computer network 28 suggests and renders obvious the use of a wireless connection by the client 30, as wireless networks and wireless client devices and connections, including for receiving multimedia, were well-known in the art as of April 18, 2005. (*See, e.g.*, EX1007 at 1 (“Multimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices.”); EX1018 at 236-37 (describing complexity scalable MPEG encoding techniques for a large product family, including in particular wireless mobile devices), Fig. 1; EX1012 at 1:29-31

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(listing examples of “available communications devices” as including “e.g., handheld PCs, personal digital assistants (PDAs), mobile phones, laptops, desktops, etc.”); EX1025 at [0015]-[0016] (network for streaming multimedia content may be, e.g., “a mobile communications network”), Fig. 1; EX1027 at 657 (“The demand for fast and location-independent access to multimedia services offered on today’s Internet is steadily increasing.... Video transmission for mobile terminals is likely to be a major application in emerging 3G systems and may be a key factor in their success.”); EX1021 at 2 (“Moreover, the increasing mobility in telecommunications is a major trend. Mobile connections will not be limited to voice; other types of data, including real-time media, are already emerging.”).)

163. Carmel also expressly teaches the use of a wireless connection to connect a client to a server via its incorporation of Carmel ’432 and Carmel ’230. (See Carmel at 6:65-7:3 (“Some useful techniques for preparing, compressing and transmitting multimedia sequences are described in U.S. Pat. No. 5,841,432 [Carmel ’432, EX1009] and in the above-mentioned U.S. patent application Ser. No. 08/919,027 [Carmel ’230, EX1010], both of which are incorporated herein by reference.”), 2:41-46.)⁷ For example, Carmel ’432 teaches:

⁷ As I previously discussed, if Carmel’s incorporation by reference of Carmel ’432 and Carmel ’230 does not suffice to incorporate all of the disclosure of Carmel ’432 and Carmel ’230 into Carmel itself such that Carmel is considered as a single

FIG. 1 illustrates a typical network on which the invention can be implemented. As noted previously, for purposes of this disclosure, the term “network” shall mean **at least two computers connected for communications through communication lines which can be hard wired or virtual, such as satellite communications**. In a more preferred form, the invention is implemented on global computer networks made up of multiple networks connected by common protocols, such as the network presently known as the “Internet.” Such a network consists of multiple computers, including, for example, but not limited to such devices as servers, and end user computers, such as personal computers, all connected through telephone lines, **or other communication links, both physical and wireless**.

(EX1009 at 3:21-35, Fig. 1; *see also id.* at 3:43-47 (“As illustrated in FIG. 1, a network 11, of the type on which the invention may be implemented, consists of plural network interconnections 17, and multiple computers such as servers 13, identified as servers 13₁ -13_n, and end user computers, such as computer 15”), 3:48-

reference that includes all of the incorporated disclosure, a person of ordinary skill in the art would still have found it obvious to and would have been motivated to consider the teachings of Carmel, Carmel ’432, and Carmel ’230 together in view of their shared inventors, their common and overlapping subject matter (*i.e.*, conveying multimedia sequences to a destination computer across a computer network), and Carmel’s express reference to and incorporation of the other two Carmel patents. If Carmel is not considered a single combined reference, Carmel in view of the teachings of Carmel ’432 and/or Carmel ’230 would have rendered this claim element directed to a wireless connection obvious.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

53 (“A typical end user computer 15, as further illustrated in FIG. 2, will include a main computer unit 19, with peripherals such as a keyboard 21, mouse 25, and display 23 connected thereto. **Computer 19** may be connected by a telephone line 27, either physically **or through wireless connections**, to the network interconnections 17 of FIG. 1.”), 1:29-36.)

164. Carmel ’230 similarly teaches:

Source computer 21 may be connected to a server 24 on which a final product from computer 21 is mounted to form a web site, as is known in the art. A **recipient computer 26 may connect to server 24 via** public phone lines 28. **Alternatively, other connection methods may be used, such as satellite** or cable connections. Preferably, server 24 and computers 26 include suitable modems for the data transmission according to the connection method.

(EX1010 at 11:50-57.) Via these incorporated teachings, Carmel also discloses and renders obvious in this additional manner having the multimedia objects (*e.g.*, the slices and/or corresponding files of video and audio data) received from servers (*e.g.*, server 36) by the multimedia player (*e.g.*, a Java applet or other multimedia player on a client 30) using a wireless connection.

165. Further, to the extent that this claim element requires that the multimedia objects are received by the multimedia player from a plurality of (that is, two or more) servers, Carmel teachings, in view of the knowledge of one of

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

ordinary skill in the art, render such an arrangement obvious. Just as Carmel discloses multiple client computers 30, a person of ordinary skill in the art would have understood and found obvious that the multimedia streaming system taught by Carmel could beneficially include two or more servers 36 hosting copies of the sequence of multimedia objects/slices (either in whole or in part) of a multimedia stream. A person of ordinary skill in the art would have understood and found obvious that such an arrangement with multiple host servers 36 would advantageously provide a faster, more efficient, robust, adaptable system that would be better able to ensure that sufficient bandwidth/download rates are maintained and that the series of multimedia objects are received by a client 30 (and the multimedia player thereon) in the proper sequence “with only a minimal necessary transmission ... delay,” including, *e.g.*, by locating content on servers that are geographically closer to the clients. (Carmel at 10:48-63; *see also id.* at 1:51-54 (“It is an object of some aspects of the present invention to provide substantially continuous, high-bandwidth data streaming over a network using common, existing server and network infrastructure.”), 3:36-39 (download rate by clients 30 should be generally equal to the data rate), 9:10-52, 10:64-11:22, 12:6-12; EX1051 at 5:11-16, 11:24-27 (reducing delay by locating content on servers geographically closer to clients); *cf.* EX1060 at 8:3-7 (connection parameters include “location of server” “since a server

tends to have quicker connection speeds to a web site that is closer than one at a further distance”).)

166. Indeed, systems where multiple servers were used to host and serve copies of or parts of the same sequence of multimedia data to provide system speed, adaptability, scalability, and stable streaming outputs were known in the art. (*E.g.*, EX1051 at Abstract, 2:34-50, 2:54-3:8 (“The HSs [helper servers] serve to implement several methods specifically designed to support streaming multimedia, including segmentation of streaming multimedia objects (SM objects) into smaller units (i.e., chunks), cooperation of the HSs, and novel cache placement and replacement policies of the constituent units (chunks) which make up the SM objects.”), 3:65-4:6, 4:48-67, 6:2-46, 7:11-16 (“In either case, **it may be necessary for a client to get all the constituent segments of a SM [streaming multimedia] object from more than one HS** 22-24. To do so, some form of intelligent pre-fetching is required which pre-fetches subsequent segments while a current segment is being streamed to the client.”), 7:18-32, 11:17-40, 12:50-61, 13:12-19, Figs. 2, 5a-5b; EX1052 at Abstract, 2:39-59 (system configured to alleviate issues of reliability, efficiency, and latency in multimedia streaming), 6:29-33 (“In a further embodiment, the client module 114 is configured **to receive different portions of a content stream from a plurality of locations simultaneously. For example, the**

client module 114 may request and receive content from any of the plurality of web servers 116.”), 7:66-8:13 (“In a further embodiment, the client module 114 may be configured to retrieve streamlets 212 from a plurality of web servers 310.... Thus, a highly scalable solution is provided that is not hindered by massive amounts of client module 114 requests to the web server 116 at any specific location.”), 13:21-62, Fig. 1; EX1053 at [0007], [0014] (“In a further embodiment, the client module 114 may be configured to retrieve streamlets 212 from a plurality of web servers 310, each web server 116 may be located in various locations across the Internet 106.”), Figs. 1-4; EX1050 at Abstract, 3:20-26 (“According to the present invention, **digitized movie data may be accessed from different sources** (e.g., **network servers**, local disks). These sources vary in speed and response time as a result of their capabilities (e.g., limited I/O bandwidth) and transient events (e.g., resource contention). The present invention adapts to these varying conditions in order to maintain a stable output.”), 3:48-56 (“Clips [consisting of a sequence of frames] may be stored in different locations [including on different servers] within the digital network environment 102.”), 11:63-12:2, Fig. 1.)

167. Accordingly, Carmel teaches and renders obvious wherein said multimedia objects are [i] located using http and [ii] received by said multimedia player (e.g., a Java applet that controls the operation of client 30) from servers (e.g.,

from host server 36 and/or a plurality of host servers) [iii] using a wireless connection (*e.g.*, a satellite, cellular, WiFi, or other wireless connection to a mobile client device).

(f) “playing back received multimedia objects by the multimedia player,” (claim 1[c.i])

168. As discussed above for claim element 1[b.1] and incorporated here, Carmel discloses and renders obvious “playing back received multimedia objects by the multimedia player.”

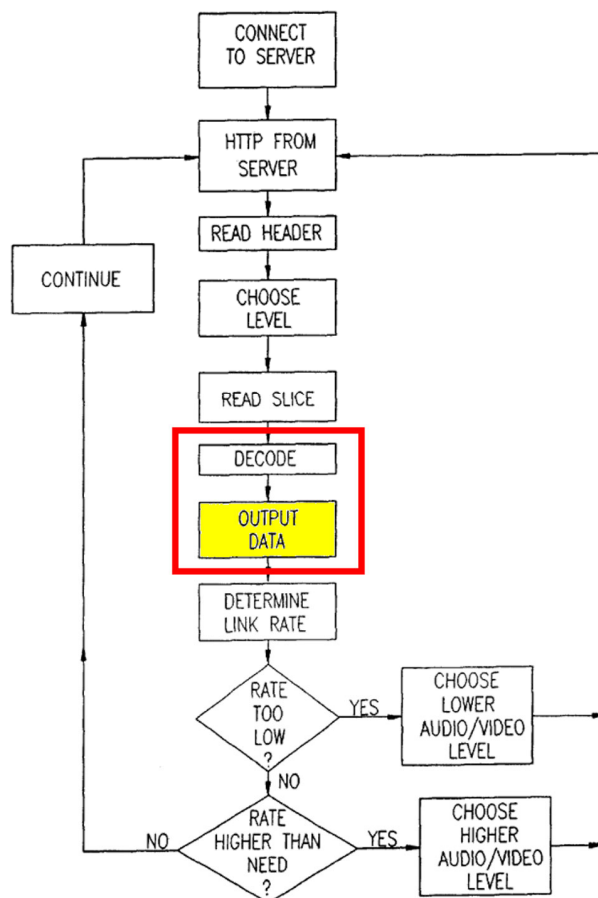
169. Again, Carmel teaches that the client computers 30 are configured to play back a multimedia data stream that from a server 36 that has been prepared and uploaded according to the methods taught by Carmel. (*See, e.g.*, Carmel at 3:66-4:2 (“Preferably, the one or more client computers decode the sequence and play back the data stream responsive to the indices of the files, at a replay rate generally equal to the data rate.”), 2:15-17 (“The clients use the slice indices of the frames to maintain proper synchronization of the playback.”), 4:3-7, 5:22-25, 10:24-11:22.)

170. Carmel teaches that the multimedia player that receives, decodes, and plays/outputs the sequence of multimedia objects encoding the broadcasted multimedia data stream can be a Java applet or plug-in to the client’s web browser. (Carmel at 7:12-17 (“Similarly, because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-

in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast.”), 10:27-35 (“The operation of client is controlled by a Java applet, which may be downloaded from server 36, and includes facilities for carrying out the steps shown in FIG. 6A [and 6B], as well as for error detection and, optionally, correction in communications received by the clients and for other functions known in the art. A sample applet of this sort is incorporated herein in the software appendix”), 10:64-11:22.)

171. Carmel further teaches that the media player plays (which it sometimes refers to as “outputs”) the received and decoded multimedia objects/slices of audio and video data. (*See, e.g.*, Carmel at 11:7-8 (“The slices are received, decoded **and output** by the client.”), Fig. 6B; *see also e.g., id.* at 10:48-54 (“In the case of a multimedia stream, **client 30 reconstructs and outputs the multimedia data for the appreciation of a user.** Time stamps in the data stream are used to synchronize the data, so **that the multimedia sequence is played back just as it was input at computer 34**, preferably with only a minimal necessary transmission and decoding delay.”).)

FIG. 6B



(Carmel at Fig. 6B.)

172. Carmel thus discloses a media player, such as a Java applet, that receives, decodes, and plays/outputs a multimedia data stream (comprised of multimedia objects/slices) “for the appreciation of a user.” These same teachings also render obvious having a media player on the client for performing these functions and “other functions known in the art.” (Carmel at 10:27-35.)

- (g) **“wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,” (claim 1[c.ii])**

173. Carmel discloses and renders obvious “wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth.”

174. As I discussed above for claim element 1[c.i] and incorporate here, Carmel teaches and renders obvious “playing back received multimedia objects by the multimedia player.” Carmel further teaches and renders obvious configuring the multimedia player to play the multimedia objects (*e.g.*, the slices or corresponding files of audio and video data) in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth.

175. First, Carmel teaches playing the multimedia objects or slices of audio and video data in a sequence. For example, Carmel provides with respect to the multi-level data stream embodiment:

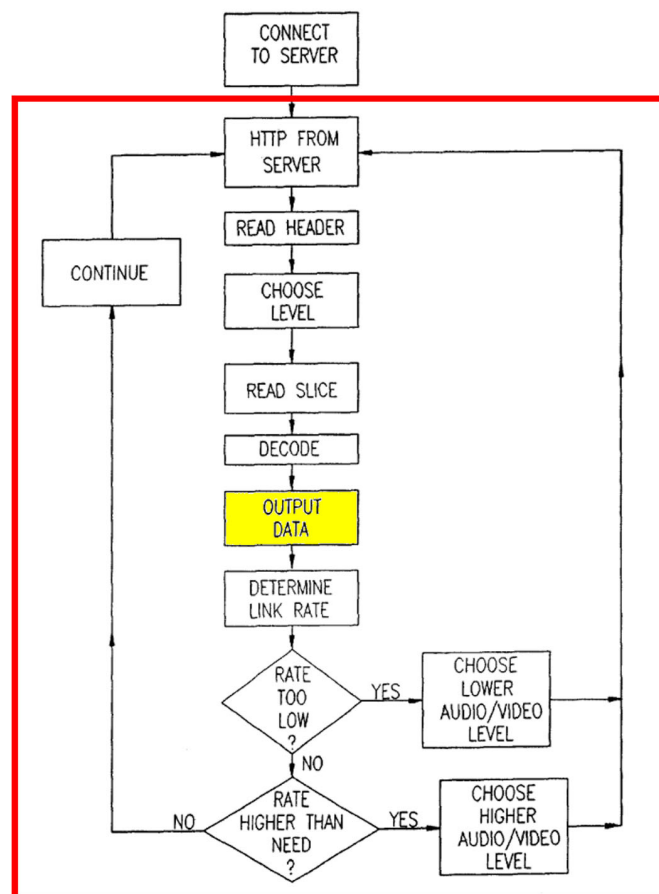
As in the method of FIG. 6A, each client 30 connects to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an initial assessment of the link bandwidth, the

client selects one of the available quality levels in the stream.

Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.

(Carmel at 10:64-11:8.) This process of selecting, downloading, decoding, and playing/outputting the selected sequence of slices is illustrated in Carmel Figure 6B.

FIG. 6B



(Carmel at Fig. 6B.) As I have indicated with the red outline above, this process

repeats itself on a loop as the respective multimedia objects/slices are played in sequence.

176. Carmel repeatedly refers to the series of slices of multimedia data stream as a “sequence” and, as discussed more below, teaches the use of slice indices and timestamps in an attempt to ensure that the slices are played in the correct sequential order. (*See, e.g.*, Carmel at 2:1-21 (“The clients download the data stream from the server, ... The clients use the slice indices of the frames to maintain proper synchronization of the playback. The division of the data stream into slices and the inclusion of the slice indices in the data stream to be used by the clients in maintaining synchronization allows the broadcast to go on substantially in real time without the use of special-purpose hardware.”) 3:24-5:35 (repeated use of “sequence” referring to the series of data slices), 7:4-8 (“**Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time.** Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP, as noted hereinabove with reference to uploading by computer 34.”), 7:18-31 (“FIG. 3A is a block diagram that schematically illustrates the structure of a stream of broadcast data 40 produced by computer 34, **typically corresponding to a multimedia data sequence, in accordance with a preferred embodiment of the**

present invention. Data stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T_1 , T_2 , T_3 , etc.... **Computer 34 stores each slice as a corresponding file, having a running slice index 1, 2, 3 . . . N.** Preferably, each file also includes one or more time stamps, indicating a real time at which the data in the file were recorded or an elapsed time relative to the beginning of stream 40.”), 12:6-12 (“The shorter files 46, 48, etc., that result from the change in slice duration are more likely to reach server 36 in the proper sequence, without being held up by extended bottlenecks. Furthermore, when the slice durations are shorter, the effect of “drop-out” of a slice due to failure of the corresponding link is less marked.”), Figs. 3A, 3D, 6B.)

177. Next, Carmel also teaches configuring the multimedia player to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained. For example, also as previously discussed, Carmel teaches: “In the case of a multimedia stream, **client 30 reconstructs and outputs the multimedia data for the appreciation of a user.** Time stamps in the data stream are used to synchronize the data, so **that the multimedia sequence is played back just as it was input at computer 34,** preferably with only a minimal necessary transmission and decoding delay.” (Carmel at 10:48-54.) Carmel thereby teaches that the media

player on the client 30 is configured to play the multimedia sequence such that fluidity (or continuity) and video and audio quality are maintained—that is, such that it is played back “just as” (or, as a person of ordinary skill in the art would have understood, as close as possible under the particular bandwidth, processor, and other constraints) it was input on the transmitting computer 34. This disclosure further teaches and suggests playing the slices in the correct, sequential order. (*See also* Carmel at 12:6-12 (describing “proper sequence” of files and minimizing the impact on fluidity/continuity from “drop-out”), 13:29-35 (stating teachings relating to upload are applicable to download).)

178. Indeed, the goal of maintaining continuity or fluidity playback along with the highest (or highest feasible or practicable in view of technical and/or economic constraints) quality of audio and video playback when streaming multimedia content to a receiving computer was well-known to those of ordinary skill in the art and something that those in the field sought to achieve. (*See, e.g.*, Mattavelli at 2309 (“When the complexity of the task exceeds the available processing power, real-time services, possibly of degraded quality, are nevertheless guaranteed respecting all real-time constraints.”), 2310-12; EX1012 at 2:20-22, 4:11-5:8 (“Accordingly, and in view of the relative advantages and disadvantages of bi-level video vs. full-color video communications in different bandwidth condition,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

systems and procedures to combine bi-level and full-color video as a function of current and potential network bandwidth conditions are described.”); EX1024 at 1:8-24 (describing method for computational graceful degradation of audiovisual decompression that is useful where “channel capacity is limited and some form of quality of service guarantee is required” and noting “[i]t is common in the case of software decoding to employ some form of graceful degradation when the system resources is not sufficient to fully decode all of the video bitstream”); EX1026 at 1:8-19 (disclosing method for degrading the amount of computation needed to decode video objects using priority information for “optimal graceful degradation in terms of relative[] visual importance between respective video objects”); EX1020 at 199, 209-10, 213-15 (Simple Scalable Visual Profile “is useful for **applications that provide services at more than one level of quality due to bit-rate or decoder resource limitations**, such as Internet use” and Core Scalable Profile “adds support for coding of temporal and spatial scalable arbitrarily shaped objects to the Core Profile [and] is useful for applications such as the Internet, mobile, and broadcast.”), 218-19 (“The *Fine Grain Scalability Profile* (FGS), often known as the *Streaming Profile*, is intended to support applications and environments **where the bandwidth and/or computational power available cannot be predicted and may vary dynamically.**” (italics in original)), 220; EX1021 at 294-95 (scalable video coding).)

A person of ordinary skill in the art would have found it obvious and would have been motivated to achieve this goal, which is fundamental in the field of streaming, to maintain and/or improve the fluidity and/or quality of the playback of streamed multimedia content.

179. Next, Carmel teaches configuring the multimedia player to maintain playback fluidity and video and audio playback quality by selecting a plurality of multimedia objects that reflect available network bandwidth. Indeed, the multi-level data stream embodiment taught by Carmel is configured to do exactly this—that is, to maintain continuity and the highest quality video and audio possible by selecting multimedia slices/files with the quality level that reflects available network bandwidth. Carmel provides:

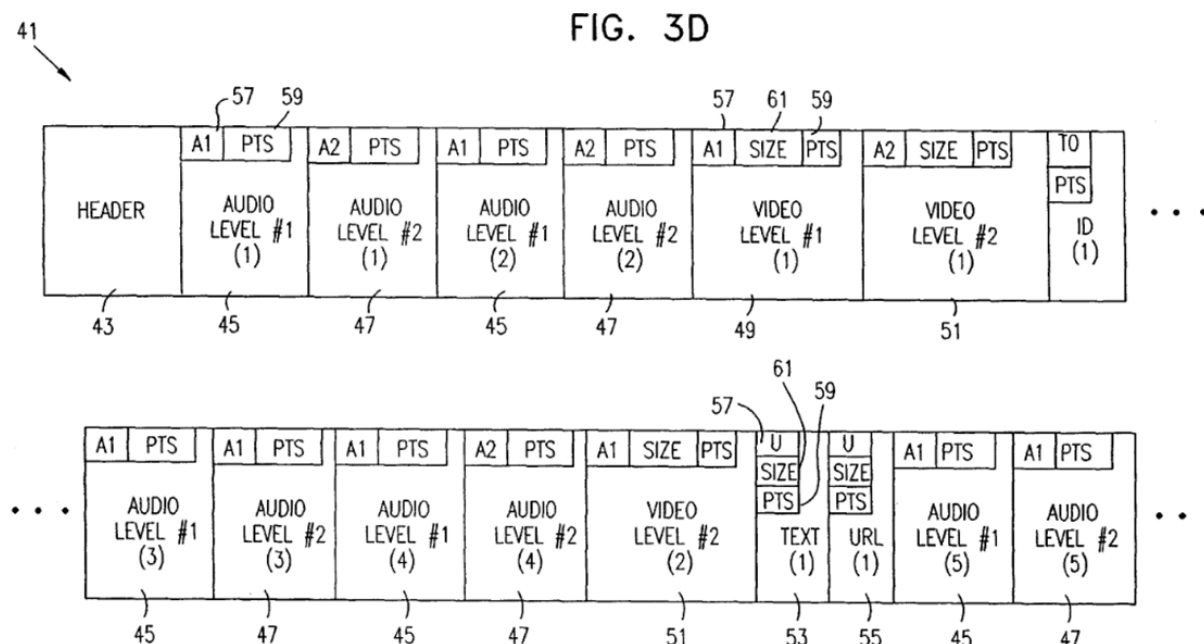
FIG. 3D is a block diagram that schematically illustrates a file format of a multi-level data stream 41, in accordance with another preferred embodiment of the present invention....

Each time slice in stream 41 includes multimedia data at multiple quality levels. There are two such levels in the example shown in FIG. 3D, identified as level #1 and level #2, but a larger number of levels may also be used. Typically, the audio and video data in level #1, contained in slices 45 and 49, are more highly compressed relative to the data in slices 47 and 51 of level #2. **In consequence, the level #1 slices have smaller data volume than the level #2 slices and can therefore be transmitted over a lower-bandwidth data link,**

while maintaining the required slice timing indicated by time stamps 59. The lower data-rate transmission generally comes at the expense of inferior sound and/or image quality. Size identifier 61 describes the size of those slices in stream 41 that have a fixed size associated therewith, wherein typically the size (or the corresponding resolution) of the level #1 video slices is smaller than that of the level #2 slices.

Each of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36. A method for selecting and, as required, varying the level is described hereinbelow with reference to FIG. 6B.

(Carmel at 8:42-9:9.)



(Carmel at Fig. 3D.)

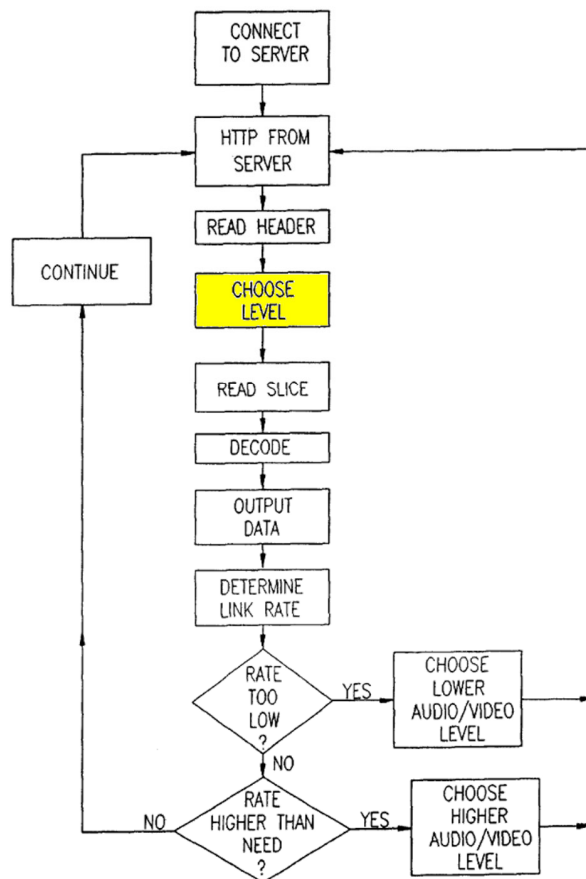
180. In connection with Figure 6B, Carmel again teaches that the client selects an initial quality level for the audio and video data slices (that is, the multimedia objects) based on and in a manner that reflects available network bandwidth. Carmel provides:

FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D) transmitted from server 36, in accordance with another preferred embodiment of the present invention. As in the method of FIG. 6A, **each client 30** connects to the server, generally using a single HTTP link. **After reading header 43 and, preferably, making an initial assessment of the link bandwidth, the client selects one of the available quality levels in the stream. Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level.**

The slices are received, decoded and output by the client.

(Carmel at 10:64-11:8.)

FIG. 6B

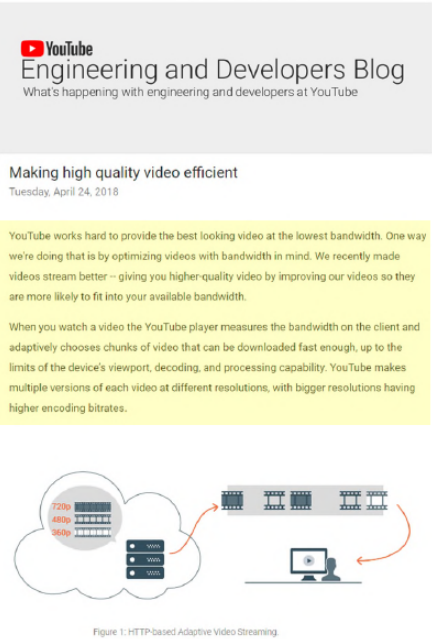


(Carmel at Fig. 6B.)

181. Carmel thus teaches and render obvious “playing back received multimedia objects by the multimedia player ... such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth” via these multi-level streaming teachings, in which multiple versions of a multimedia data stream (comprising slices at multiple quality levels) are provided and selected by a client (controlled by a multimedia

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

player) based on the client's assessment of network bandwidth availability. Patent Owner has signaled (via its infringement allegations made in its complaint) its position that such a configuration or functionality satisfies this claim element. (*See, e.g.,* EX1055, Ex. 2 at 26-28.)

playing back received multimedia objects by the multimedia player, wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification, and by	YouTube playback fluidity, video quality, and audio quality are maintained based on multiple versions of each stream that fit available bandwidth:  The screenshot shows a blog post from YouTube's Engineering and Developers Blog. The title is 'Making high quality video efficient' and it is dated Tuesday, April 24, 2018. The text explains that YouTube works to provide the best looking video at the lowest bandwidth by optimizing videos with bandwidth in mind. It mentions that they recently made videos stream better by giving users higher-quality video by improving their videos so they are more likely to fit into the user's available bandwidth. The post also states that when a user watches a video, the YouTube player measures the bandwidth on the client and adaptively chooses chunks of video that can be downloaded fast enough, up to the limits of the device's viewport, decoding, and processing capability. It concludes by saying that YouTube makes multiple versions of each video at different resolutions, with bigger resolutions having higher encoding bitrates. Below the text is a diagram labeled 'Figure 1: HTTP-based Adaptive Video Streaming' which illustrates a cloud containing video segments being downloaded by a client. Ex. 18, <i>Making high quality video efficient</i>, YouTube Engineering and Developers Blog, https://youtube-eng.googleblog.com/2018/04/making-high-quality-video-efficient.html (Apr. 24, 2018), at 1.
--	--

(EX1055, Ex. 2 at 26.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

playing back received multimedia objects by the multimedia player, wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification, and by	YouTube playback fluidity, video quality, and audio quality are maintained based on multiple versions (green) of each video that reflect available bandwidth (yellow): <pre>#EXTM3U #EXT-X-INDEPENDENT-SEGMENTS #EXT-X-MEDIA:URI="https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/i #EXT-X-STREAM-INF:BANDWIDTH=174282, CODECS="avc1.404815,mp4a.40.2", RESOLUTION=426x240, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="233", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/229/source/yo #EXT-X-STREAM-INF:BANDWIDTH=372198, CODECS="avc1.40481E,mp4a.40.2", RESOLUTION=640x368, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/230/source/yo #EXT-X-STREAM-INF:BANDWIDTH=491883, CODECS="avc1.40481E,mp4a.40.2", RESOLUTION=854x488, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/231/source/yo #EXT-X-STREAM-INF:BANDWIDTH=861742, CODECS="avc1.40481F,mp4a.40.2", RESOLUTION=1280x720, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/232/source/yo #EXT-X-STREAM-INF:BANDWIDTH=158183, CODECS="avc1.40481C,mp4a.40.5", RESOLUTION=256x144, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="233", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/269/source/yo #EXT-X-STREAM-INF:BANDWIDTH=153955, CODECS="vp09.00.11.08,mp4a.40.5", RESOLUTION=256x144, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="233", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/683/source/yo #EXT-X-STREAM-INF:BANDWIDTH=247948, CODECS="vp09.00.20.08,mp4a.40.5", RESOLUTION=426x240, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="233", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/684/source/yo #EXT-X-STREAM-INF:BANDWIDTH=332522, CODECS="vp09.00.20.08,mp4a.40.2", RESOLUTION=426x240, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/684/source/yo #EXT-X-STREAM-INF:BANDWIDTH=587728, CODECS="vp09.00.21.08,mp4a.40.2", RESOLUTION=640x368, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/685/source/yo #EXT-X-STREAM-INF:BANDWIDTH=716029, CODECS="vp09.00.30.08,mp4a.40.2", RESOLUTION=854x488, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/686/source/yo #EXT-X-STREAM-INF:BANDWIDTH=1314966, CODECS="vp09.00.31.08,mp4a.40.2", RESOLUTION=1280x720, FRAME-RATE=24, VIDEO-RANGE=SDR, AUDIO="234", CLOSED-CAPTIONS=NONE https://manifest.googlevideo.com/api/manifest/hls_playlist/expire/1711852684/ei/3HcIZs0BMKKI_9EPKM2h8Aa/ip/173.72.184.227/id/6de6a64b8199e53f/itag/689/source/yo</pre> Video.
--	--

(EX1055, Ex. 2 at 27.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

For adaptive multi-bit-rate playback, HLS requires the player to assess network bandwidth availability to select multimedia objects:

Overview

The Multivariant Playlist describes all of the available variants for your content. Each variant is a version of the stream at a particular bit rate and is contained in a separate playlist. The client switches to the most appropriate variant based on the measured network bit rate. The client's player is tuned to minimize stalling of playback, to give the user the best possible streaming experience.



```
graph LR; Index[Index file] --> AltA[Alternate index file]; Index --> AltB[Alternate index file]; Index --> AltC[Alternate index file]; AltA --> m3u8A[m3u8]; AltB --> m3u8B[m3u8]; AltC --> m3u8C[m3u8];
```

Ex. 13, *HTTP Live Streaming – Creating a Multivariant Playlist*, at 9.

See also,

6.2.4. Providing variant streams

A server MAY offer multiple Playlist files to provide different encodings of the same presentation. If it does so it SHOULD provide a variant Playlist file that lists each variant stream to allow clients to switch between encodings dynamically.

Variant Playlists MUST contain an EXT-X-STREAM-INF tag for each variant stream. Each EXT-X-STREAM-INF tag for the same presentation MUST have the same PROGRAM-ID attribute value. The PROGRAM-ID value for each presentation MUST be unique within the variant Playlist.

Ex. 16, *HTTP Live Streaming Draft*, at 27.

(EX1055, Ex. 2 at 28.)

182. And, as I will discuss in more detail below for the following claim element 1[c.iii], Carmel also teaches that in the multi-level data stream embodiment, in the event that the available network bandwidth is or becomes insufficient for the initial quality level selected by the client and multimedia player thereon (*e.g.*, if lag is detected), the multimedia player is configured to maintain playback continuity or fluidity by choosing a lower audio/video quality level. Conversely, if the client determines that the available network bandwidth is or becomes sufficient to support slice download and playback at a higher quality level while still maintaining

playback fluidity/continuity, the multimedia player is configured to select a higher quality audio and video level. (*See Carmel* at 7:35-49, 8:42-9:9, 11:10-22, Figs. 3D and 6B.)

183. Moreover, Carmel generally teaches and repeatedly emphasizes attempting to match the bitrate or data rate of the slices of multimedia video and audio data with the available network bandwidth for download (and upload) so as to avoid lag and maintain fluidity/continuity of the multimedia playback at the best possible/practicable quality. (*See, e.g., Carmel* at Abstract (“The slices are encoded in a corresponding sequence of files, each file having a respective index, and the sequence is uploaded to a server at an upload rate generally equal to the data rate of the stream, **such that the one or more client computers can download the sequence over the network from the server at a download rate generally equal to the data rate.**”), 2:1-21, 2:56-59, 3:24-39, 3:55-59, 3:66-4:2, 4:20-24 (“Most preferably, opening the plurality of links includes opening links such that the data rates of the links taken together **are sufficient to download the sequence at the download rate generally equal to the data rate.**”), 4:63-5:28, 7:35-41 (“Computer 34 monitors the time codes as file 40 is transmitted, and **clients 30 similarly monitor the time codes as the file is received, in order to ensure that the transmission or reception is ‘keeping up’ with the input of the data to the computer. In the event**

that a lag is detected, steps are taken to increase the data transmission or reception rate, as described further hereinbelow.”), 9:10-51, 10:19-23, 10:48-63 (“Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.... Further preferably, the client compares the times stamped in the data stream to a local real-time clock and, if it determines that there is a significant lag in the time codes relative to the real-time clock, opens additional links with server 36 in order to increase the overall data rate.”), 11:39-52, 11:53-12:17 (“The shorter files 46, 48, etc., that result from the change in slice duration are more likely to reach server 36 in the proper sequence, without being held up by extended bottlenecks. Furthermore, when the slice durations are shorter, the effect of “drop-out” of a slice due to failure of the corresponding link is less marked. On the other hand, if it is determined that the upload time for file 42 (or a subsequent file) is substantially shorter than duration T_1 , the duration of subsequent files may be extended, and/or the compression ratio may be decreased, so as to take better advantage of the available bandwidth.”), 13:23-35 (“Although details of these steps have been described primarily with reference to the uploading process of FIG. 5, it will be understood that similar methods are applicable, *mutatis mutandis*, to the method of downloading the files

from server 36 to clients 30”), Figs. 3A, 4, 5, 7, 8, 9.)

184. These additional teachings, both considered alone and with Carmel’s teachings regarding the multi-level data stream embodiment, additionally disclose and render obvious configuring the multimedia player on client 30 to play multimedia objects (*e.g.*, slices/files of multimedia video and audio data) in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth.

185. Accordingly, Carmel discloses and renders obvious wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity (*e.g.*, real-time playback continuity), video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth (*e.g.*, via the selection of an audio and video quality level for the audio and video slices in a manner that reflects and utilizes the available network bandwidth).

(h) “autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,” (claim 1[c.iii])

186. Carmel discloses and renders obvious “autonomously adjusting said selection and playback according to the multimedia object parameters and supplied

host path identification.”

187. As discussed above for related claim elements 1[c.i] and 1[c.ii] and incorporated here, Carmel teaches and renders obvious “playing back received multimedia objects by the multimedia player” and configuring the multimedia player “to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth.” Carmel further teaches adjusting the selection of multimedia objects (*e.g.*, slices of video and audio data) and their playback in a manner that reflects the available network bandwidth “according to the multimedia object parameters and supplied host path identification.”

188. As previously discussed, Carmel teaches a multi-level data stream embodiment in which each time slice of a multimedia data stream of audio and video is encoded to have multiple quality levels. And, also as discussed above, Carmel teaches and renders obvious that each individual slice—whether stored within a single file or as separate files⁸—includes object parameters, such as a slice

⁸ As discussed above for claim element 1[a.ii], Carmel teaches and suggests that each individual time slice at a particular quality level could be saved as a separate, discrete file that includes these object parameters and host path identification. (*See, e.g.*, Carmel at Abstract, 2:22-23 (“Preferably, each segment or slice is contained in a separate, respective file.”), 3:24-39, 5:15-18 (“In a preferred embodiment, the slices are encoded at a plurality of different quality levels, such that the files corresponding to a given one of the slices have a different respective data size for

index/identifier, one or more timestamps, and a size/quality level parameter. Carmel provides:

FIG. 3D is a block diagram that schematically illustrates a file format of a multi-level data stream 41, in accordance with another preferred embodiment of the present invention. The data stream is divided into audio slices 45, 47 and video slices 49, 51, and may also include other data formats, such as a text slice 53 and/or a URL slice 55. **Each slice is preferably identified by a level identifier 57, a presentation time stamp (PTS) index 59 and, as appropriate, a size identifier 61.** The function of these identifiers and indices is described further hereinbelow. A header 43 includes data such as the title, author, copyright and formats of the data in the stream; the duration of the multimedia sequence represented by the stream; and **a description of the available stream levels and associated data sizes.**

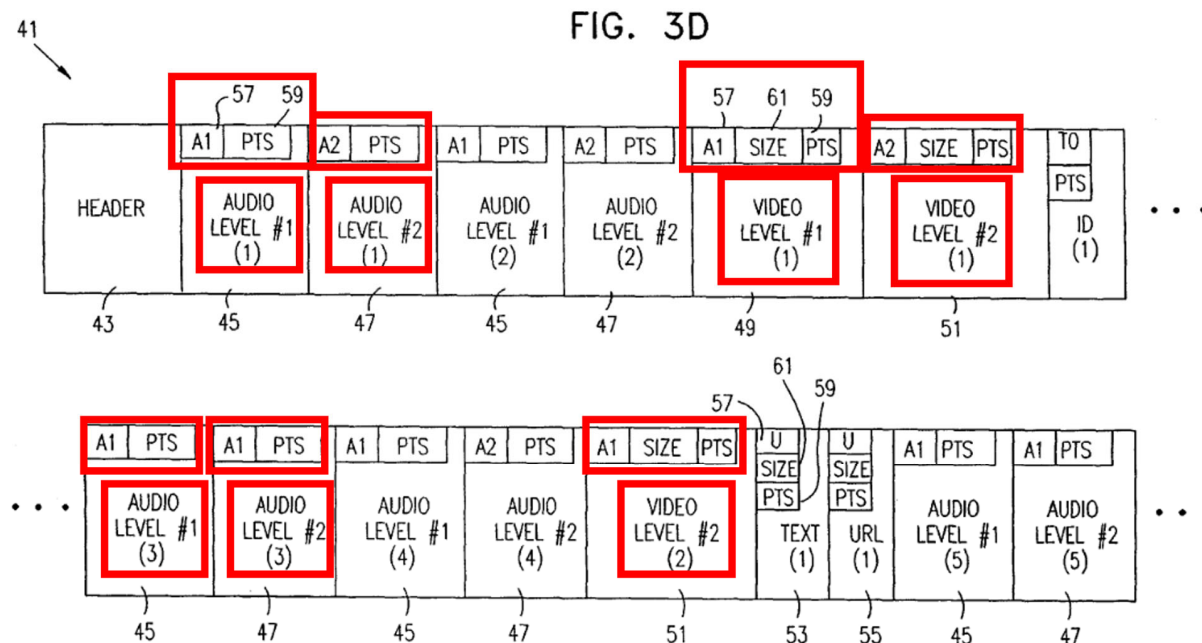
Each time slice in stream 41 includes multimedia data at multiple quality levels. There are two such levels in the example shown in FIG. 3D, identified as level #1 and level #2, but a larger number of levels may also be used. Typically, the audio and video data in level #1, contained in slices 45 and 49, are more highly compressed relative to the data in slices 47 and 51 of level #2. In consequence, the level #1 slices have smaller data volume than the level #2 slices and can

each of the quality levels.”), 8:42-9:5, 9:49-52 (“Alternatively or additionally, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36.”), 14:18-32, 15:8-17.)

therefore be transmitted over a lower-bandwidth data link, while maintaining the required slice timing indicated by time stamps 59. The lower data-rate transmission generally comes at the expense of inferior sound and/or image quality. **Size identifier 61 describes the size of those slices in stream 41 that have a fixed size associated therewith, wherein typically the size (or the corresponding resolution) of the level #1 video slices is smaller than that of the level #2 slices.**

Each of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36. **A method for selecting and, as required, varying the level is described hereinbelow with reference to FIG. 6B.**

(Carmel at 8:42-9:9.) As I have bolded in the last line reproduced above, Carmel teaches a method for selecting and adjusting or varying, as required, the quality level of the downloaded slices in response to the available network bandwidth.



(Carmel at Fig. 3D.)

189. In connection with Figure 6B, Carmel teaches how following an initial selection of the quality level, the client computer 30 is configured to periodically assess the network bandwidth and data transfer over the links and adjust the quality level accordingly of the played back(/output) slices, if necessary (or beneficial). Carmel provides:

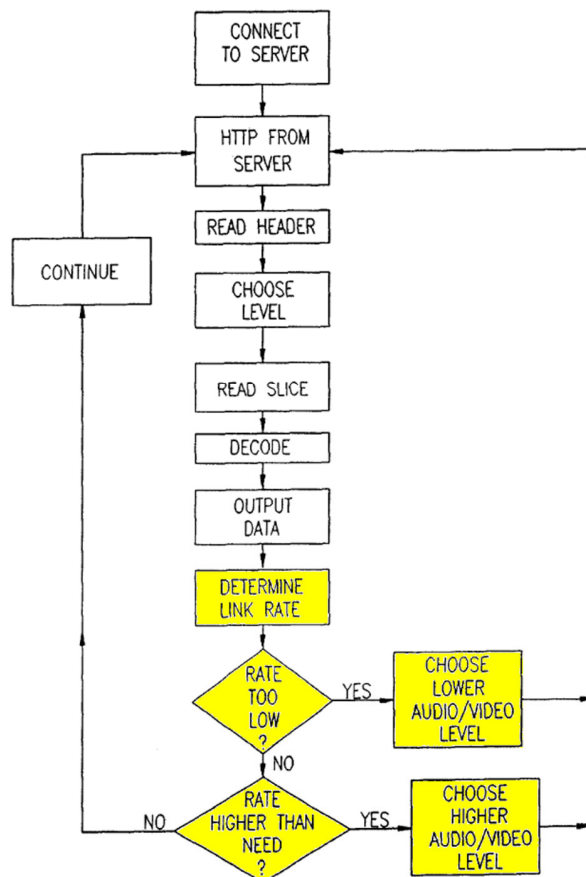
FIG. 6B is a flow chart illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41 (FIG. 3D) transmitted from server 36, in accordance with another preferred embodiment of the present invention. As in the method of FIG. 6A, each client 30 connects to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an initial

assessment of the link bandwidth, the client selects one of the available quality levels in the stream. Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.

Periodically, client 30 makes an assessment of the rate of data transfer over the link from the server and, if necessary, changes the quality level accordingly. For example, **if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, the client will preferably select a lower quality level if one is available. On the other hand, if the rate is substantially higher than what is needed to receive the successive slices on time, the client may select a higher quality level to take advantage of the available bandwidth.** Preferably, upper and lower data rate thresholds, or watermarks, are set dynamically in response to the data rate and are used in determining when a new quality level should be selected.

(Carmel at 10:64-11:22.)

FIG. 6B



(Carmel at Fig. 6B.) I have highlighted the steps in Figure 6B above that involve the adjustment (when necessary or beneficial) to the audio/video slice quality level selection that is made by the client computer (and multimedia player thereon) according to the available network bandwidth.

190. Carmel thereby teaches and suggests that in the event that the available network bandwidth is or becomes insufficient for the initial quality level selected by the client and multimedia player thereon (*e.g.*, if lag is detected), the multimedia

player is configured to maintain playback continuity or fluidity by choosing a lower audio/video quality level. Conversely, if the client determines that the available network bandwidth is or becomes sufficient to support slice download and playback at a higher quality level while still maintaining playback fluidity/continuity, the multimedia player is configured to select a higher quality audio and video level.

191. Further, Carmel teaches that this assessment of whether network bandwidth or the rate of data transfer is insufficient for (or sufficiently in excess of) the currently selected bitrate or quality level of the multimedia slices is performed by the client computer by using the multimedia object parameters, including, in particular, the “time stamps 59” and slice size/quality parameters. (Carmel at 11:11-18 (“For example, if the rate is low, **such that *time stamps 59* indicate that the slices need to be played as fast as or faster than they are being received**, the client will preferably *select a lower quality level* if one is available. On the other hand, if the rate is substantially higher than what is needed *to receive the successive slices on time, the client may select a higher quality level* to take advantage of the available bandwidth.”).) This disclosure regarding selecting slices with a lower or higher quality level also teaches and suggests that the client/multimedia player is configured and is able to identify and locate—using slice identifiers, size/quality level parameters, and associated host path identification for the slices—the

successive slices (having a new quality level) that the media player will receive, decode and output/play. (See Carmel at 10:64-11:22.)

192. Carmel provides additional teachings concerning the use of timestamp information associated with the slices to assess whether the available network bandwidth is sufficient to keep up with the multimedia playback. For example, Carmel teaches:

Preferably, each file also includes one or more time stamps, indicating a real time at which the data in the file were recorded or an elapsed time relative to the beginning of stream 40....

Computer 34 monitors the time codes as file 40 is transmitted, and **clients 30 similarly monitor the time codes as the file is received, in order to ensure that the transmission or reception is “keeping up” with the input of the data to the computer. In the event that a lag is detected, steps are taken to increase the data transmission or reception rate**, as described further hereinbelow. For example, as shown in FIG. 3A, time intervals T_1 , T_2 , T_3 , etc., are not all equal, but rather are adjusted by computer 34 in response to the transmission rate. **Alternatively or additionally, the compression level of the data is varied, as is likewise described below, so as to adjust the data streaming rate to the available bandwidth over one or more channels** between computer 34 and server 36, and/or **between server 36 and client 30.**

(Carmel at 7:28-49.) Carmel similarly teaches elsewhere:

In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user. **Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.**

Client 30 preferably monitors the rate of data coming in over each of its links with server 36. If any of the links is non-operative or is operating unacceptably slowly, that link is closed, and a new link is opened in its place, as described above. **Further preferably, the client compares the times stamped in the data stream to a local real-time clock and, if it determines that there is a significant lag in the time codes relative to the real-time clock, opens additional links with server 36 in order to increase the overall data rate.**

(Carmel at 10:48-63.)

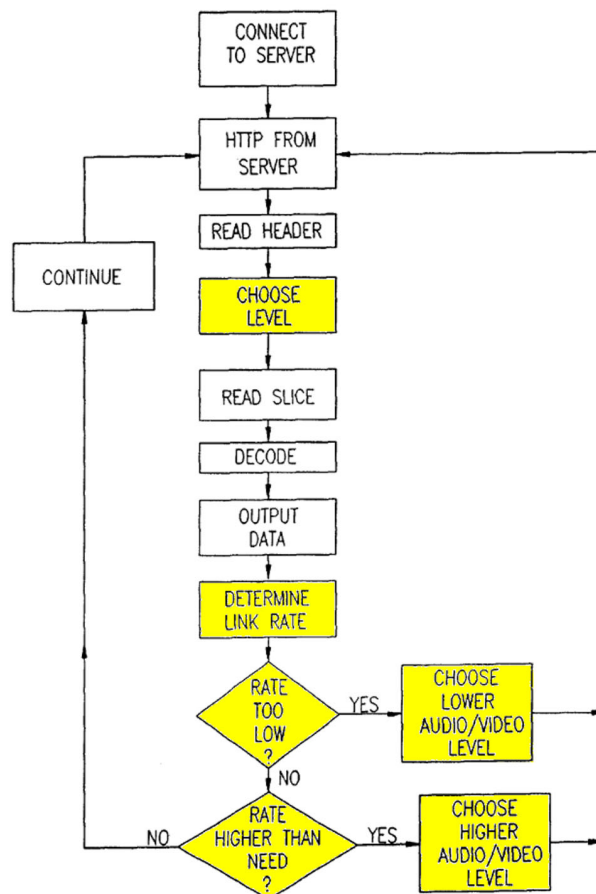
193. These teachings are in accord with Carmel's teachings regarding the multi-level data stream embodiment. They also teach the use of time stamps (that is, a multimedia object parameter) associated with the multimedia data slices to assess the available network bandwidth. The available network bandwidth is assessed to achieve the goal of matching the bitrate or data rate of the multimedia slices with their download rate so as to ensure that the transmission and reception is "keeping up" with the playback rate so as to avoid lag and playback discontinuities to have a

high or, at least, acceptable playback quality, “just as it was input at computer 34.” And, when there is a mismatch between the bitrate/data rate and download rate, Carmel teaches different non-exclusive options to remedy this situation, including adjusting the compression/quality level of the slices, choosing between slices having multiple quality levels, and/or adjusting the available bandwidth. (See Carmel at 7:28-49, 10:48-63, 10:64-11:22; *see also id.* at 9:49-52 (“Alternatively or additionally, the link bandwidth may be accommodated using the multi-level transmission concept illustrated by FIG. 3D, over either a single link or over multiple simultaneous links with server 36.”).) These additional teachings also suggest and render obvious “adjusting said selection and playback according to the multimedia object parameters and supplied host path identification” and further show the obviousness of this claim element when considered alongside Carmel’s teachings regarding the multi-level data stream embodiment. In circumstances where both methods (that is, multiple links and the multi-[quality] level methods) are used to ensure sufficient bandwidth that matches the encoding rate of the data slices, a person of ordinary skill in the art would have appreciated that it would be critical for the client to know which slices/files to request (at what quality level) and to control what slices/files were being requested over each link, and it would have been obvious to use the object parameters and supplied host path information to do so.

194. Further, Carmel also teaches that this adjustment to said selection and playback is performed “autonomously” by the client computer 30. Carmel describes the client computer 30—again, which is controlled by media player thereon—as the entity that chooses, selects, and/or adjusts (when necessary) the quality level of the multimedia slices that it downloads and receives in a manner that reflects the available network bandwidth. (*See, e.g.*, Carmel at 11:1-8 (“each **client 30 connects** to the server, generally using a single HTTP link. After reading header 43 and, preferably, making an initial assessment of the link bandwidth, **the client selects** one of the available quality levels in the stream.... The slices are received, **decoded and output by the client.**”), 11:9-15 (“Periodically, **client 30 makes an assessment** of the rate of data transfer over the link from the server **and, if necessary, changes the quality level accordingly.** For example, if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, **the client will preferably select** a lower quality level if one is available.”), 10:27-31, 10:64-67 (“FIG. 6B is a flow chart **illustrating the operation of clients 30 in downloading and playing back multi-level data stream 41** (FIG. 3D) transmitted from server 36, in accordance with another preferred embodiment of the present invention.”), Fig. 6B; *see also id.* at 8:56-9:9 (“**Each of clients 30 chooses** or is assigned **the quality level appropriate to the bandwidth of its link on**

network 28 to server 36.”), Fig. 3D.)

FIG. 6B



(Carmel at Fig. 6B.) Carmel thus teaches and suggests that the client (and multimedia player thereon) “autonomously”—that is, without server-side direction or intervention—adjust the selection and playback of the multimedia objects in view of the available network bandwidth.

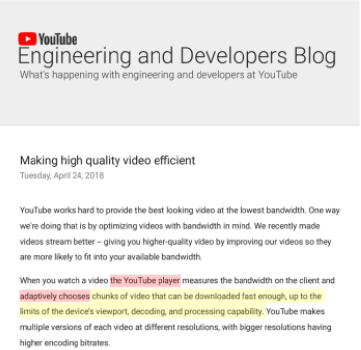
195. These teachings further disclose and render obvious where the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

adjusting of the selection and playback is done autonomously by the client, without any server-side intervention. Indeed, in addition to Carmel's teachings, such client-directed feedback and adjustment mechanisms were well-known in the art and an obvious design option. (*See, e.g.*, EX1019 at 228 (“[T]his protocol emphasizes client QoS control, taking into account that the receiver has better and immediate knowledge of the condition of the end-to-end connection, including the network condition which may not be readily accessible to the transmitter.”), 229 (“[T]he client detects network congestion and requests a downward scaling of the media to relieve the bandwidth bottleneck.”).)

196. I note that in its complaint in the related litigation, the Patent Owner has alleged that the same type of functionality that is disclosed by Carmel—that is, having the multimedia player assess the available bandwidth and adaptively choose (using object parameters like size, resolution/bitrate/quality, etc.) among different quality-level options what slices/segments/chunks of multimedia data (having host path identification such as URLs) can be downloaded fast enough in view of the available bandwidth—satisfies this claim element. (*See, e.g.*, EX1055, Ex. 2 at 29-32.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

playing back received multimedia objects by the multimedia player, wherein the multimedia player is configured to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth, autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification, and by	The YT Player selects a playlist variant based on bandwidth: Using playback statistics, we are able to model the behavior of the player as it switches between resolutions. This allows us in effect to predict when an increased bitrate would be more likely to cause a player to switch to a lower resolution and thereby cancel the effect of bitrate increase in any one resolution. With this model, we are able to find better operating points for each video in the real world.¹ Ex. 18, <i>Making high quality video efficient</i>. YouTube engineering documentation explicitly states that the YouTube player (on the client) autonomously (i.e., without server-side intervention) adjusts the video playback.  <i>Id.</i>
--	--

(E.g., EX1055, Ex. 2 at 29.)

197. Additionally, as discussed in greater detail below for claim element 1[c.iv] and incorporated here, Mattavelli also teaches autonomously adjusting the playback of received multimedia objects according to multimedia object parameters. In short, Mattavelli teaches techniques for computational graceful degradation for decoding in which object parameters (including, e.g., decoding complexity metadata that is available with the MPEG-4 standard) are used to determine when and by how much to adjust the decoding/playback of received video segments in view of processor constraints of the decoder/multimedia player. (See Mattavelli at 2309 (“The presented approach can be virtually applied to any video compression

standard and processor based platform, **although the best performances are achieved when complexity prediction information provided as side information by the video standard is available. This option has been included in the [then-]new MPEG-4 standard.**”), 2311 (“[O]n the basis of the predicted decoding time and on **the complexity information** (i.e. statistics of the algorithms necessary to correctly decode the frame), **executes only the CGD algorithms that are just necessary to not exceed the target decoding time.**”), 2311-12.)

198. Accordingly, Carmel and Carmel in view of Mattavelli teach and render obvious autonomously (*e.g.*, directed by the client (controlled by the media player thereon)) adjusting said selection and playback according to the multimedia object parameters and supplied host path identification (*e.g.*, adjusting the quality level of the audio and video slices selected and the playback of the received slices according to, for example, slice identifiers, time stamps, and decoding complexity metadata, locating the adjusted selection of slices using the appropriate host path identification).

(i) “and by utilizing optimized decoding processes to maintain quality playback.” (claim 1[c.iv])

199. Carmel in view of Mattavelli renders obvious “and by utilizing optimized decoding processes to maintain quality playback.”

200. Carmel in view of Mattavelli teaches and renders obvious configuring

Carmel's multimedia player to utilize optimized decoding processes to play multimedia objects in a sequence such that fluidity, video quality and audio quality are maintained. And, as I discuss below, there are several reasons and rationales that would have motivated a person of ordinary skill in the art to consider and combine the teachings of Carmel and Mattavelli in the manner I have proposed.

201. Mattavelli provides "motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding." (Mattavelli at 2309.) Mattavelli provides that under this CGD approach: "[t]he goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources." (*Id.*) In other words, Mattavelli teaches methods where degraded, less computationally-intensive decoding algorithms and operations for decoding compressed multimedia data are used during periods where the processing needs are high and exceed the available processing resources, where the visual quality of the video output is possibly temporarily degraded to ensure that the fluidity or real-time continuity of the playback is maintained. (*See also, e.g.*, Mattavelli at 2309 ("The main rationale of the Computational Graceful Degradation (CGD) approach is to provide results of poorer quality when the results of the desirable quality cannot be obtained in real time. When the complexity of the task exceeds the available

processing power, real-time services, possibly of degraded quality, are nevertheless guaranteed respecting all real-time constraints.”), 2312 (“Obviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as [a] tool able to guarantee real-time performance in [the] presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power.”).) These types of graceful degradation techniques were known in the art. (*See, e.g.*, EX1029 at 373 (graceful degradation decoding “algorithm can adapt itself to the available resources: depending on CPU utilization, some operations are done, simplified or even skipped. For instances the Inverse DCT can be truncated, or the sub-pixel interpolation filters of the motion compensation can be approximated.”).)

202. Mattavelli states that its goal “is to guarantee real-time video decoding with possible degradations of the image quality, i.e[.,] reductions of the peak signal to noise ratio (PSNR) without losing parts of the image content.” (Mattavelli at 2309.) Mattavelli teaches that there are several decoding operations or algorithms that can be simplified to save computational resources when necessary, at the expense of potentially degraded, noisier, or less precise results. (*Id.* at 2310.) These operations include, e.g., “the IDCT [inverse discrete cosine transform], the prediction with and without motion compensation in its various forms (PB/B-frame

prediction, overlapped block motion compensation (OBMC), half pixel interpolation, macroblock padding and so on). Simplifications of such operations applied to pixels or transform coefficients are always possible but at the expense[] of imprecise results.” (*Id.*)

203. In addition to teaching what types of decoding operations can be simplified when processing resources are exceeded, Mattavelli also discloses two strategies for determining when and to what extent such graceful computational degradation (CGD) techniques should be employed. Mattavelli teaches: “[t]he first is to monitor the execution of the decoding process, to assume that the decoding time of each portion of the frame or VOP [video object plane] is proportional to the image area, and to execute CGD algorithms whenever the current monitored decoding time exceeds the current target decoding time suitable to respect the specified real-time constraints.” (Mattavelli at 2310.) Mattavelli further teaches:

The second strategy is to exploit the complexity information carried by the frames or VOP headers, like it can be done for MPEG-4 systems [6], and use it in two stages. The first stage predicts the decoding time needed for the current frame or VOP, as described in [5]. The second, on the basis of the predicted decoding time and on the complexity information (i.e. statistics of the algorithms necessary to correctly decode the frame), executes only the CGD algorithms that are just necessary to not exceed the target decoding time.

(*Id.* at 2311.) Mattavelli further provides examples employing each approach, implementing “computationally degraded IDCT and motion compensation algorithms” using the first strategy and “implementing degraded motion compensation algorithms” using the second strategy. (*See* Mattavelli at 2311-12, Figs. 2 & 3.)

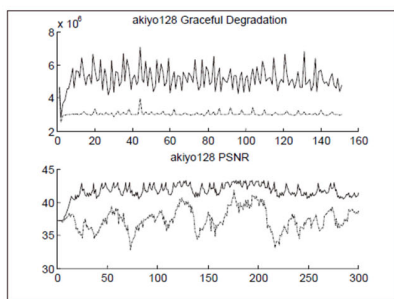


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two PSNR.

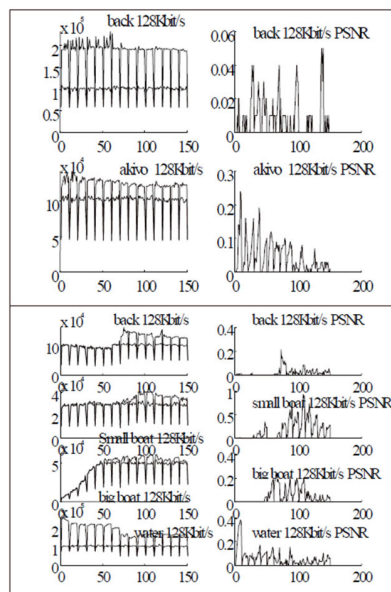


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

(Mattavelli at Figs. 2 & 3.)

204. Under the second strategy taught by Mattavelli, it uses decoding complexity metadata that is available with the MPEG-4 standard to determine when and by how much to simplify and potentially degrade the video decoding/decompressing in view of the processor constraints on the client/decoder.

(See Mattavelli at 2309 (“The presented approach can be virtually applied to any video compression standard and processor based platform, **although the best performances are achieved when complexity prediction information provided as side information by the video standard is available. This option has been included in the [then-]new MPEG-4 standard.**”), 2311 (“[O]n the basis of the predicted decoding time and on **the complexity information** (i.e. statistics of the algorithms necessary to correctly decode the frame), **executes only the CGD algorithms that are just necessary to not exceed the target decoding time.**”).)

Mattavelli further teaches:

Obviously for any Allocated Decoding Time the goal is to respect the real-time constraints with the lowest degradation in the video/audio content. Associating a degradation measure to each ω_i^{CGD} it is possible to minimize the degradation for any given T_{Alloc} . The challenge is to find the set $S_{a,b}$ of a couple of coefficients a_i, b_i , with $i \in S_{\text{CGD}}$, that satisfy minimize video degradations expressed as difference with the non degraded output:

$$S_{a,b} = \operatorname{argmin}_{(a,b)_{i \in S_{\text{CGD}}}} : (\text{PSNR} - \text{PSNR}(a,b)). \quad (5)$$

The determination of the set $S_{a,b}$ can be done by means of many suitable optimization method[s] or heuristics. The described techniques can be easily extended to the case of sequences containing multiple VOP at the same sampling time t .

(Mattavelli at 2311-12.) Mattavelli thereby teaches and renders obvious (i)

maintaining the quality of playback and playback fluidity (*e.g.*, “respect[ing] the real-time constraints with the lowest degradation in the video/audio content”) via (ii) utilizing optimized decoding processes (*e.g.*, Mattavelli’s CGD techniques, including by using “means of many suitable optimization method[s] or heuristics”), including those that employ multimedia object parameters (*e.g.*, decoding complexity prediction information that is included as part of the MPEG-4 standard). (*See* Mattavelli at 2309, 2311-12.)

205. Mattavelli thus teaches and renders obvious configuring a multimedia player (*e.g.*, a multimedia player as taught by Carmel) to utilize optimized decoding processes to playback multimedia objects in a sequence such that fluidity or real-time continuity, and video quality and audio quality are maintained. (*See* Mattavelli at 2312 (“Obviously, the implementation of CGD features are not proposed to be used at their maximum degradation rate but as [a] tool able to guarantee real-time performance in [the] presence of peaks of decoding complexity or as a way to ease the scheduling problems given by temporary fluctuations of the available processing power.”).)

206. Further, to the extent that “optimized decoding processes” is interpreted to mean that both optimized video decoding processes (like those taught by Mattavelli) and optimized audio decoding processes must be employed, Mattavelli’s

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

teachings, in view of the knowledge of a person of ordinary skill in the art, further suggest and render obvious configuring a multimedia player to utilize optimized audio decoding processes as well as optimized video decoding processes. This is so because “audio compression is built on many of the same principles as video compression” (EX1020 at 322) and a person of ordinary skill in the art would have appreciated and found obvious that the same graceful degradation principles taught by Mattavelli would be beneficial to employ in circumstances where the audio decompression demands were high/complex but the decoder processing resources were constrained. (Mattavelli at 2311-12 (“[G]oal is to respect the real-time constraints with the lowest degradation in the video/**audio** content.”).) Moreover, techniques for graceful degradation of audio decoding processes were known in the art. (*See, e.g.*, EX1039 at 1119-20 (“To achieve fast decoding of high quality audio, it is essential to keep the numerical errors to an acceptable level without running into hard real-time constraints of the decoding system.”), 1125 (“We devised two ways of gracefully degrading the audio output: Partial zeroing of the $D(m)$ window coefficients, and eliminating the upper half subband coefficients of the spectral image.”); EX1042 at 379 (scalable optimization techniques that provide a “continuous and soft degradation curve”).)

207. Indeed, many optimized audio decoding processes were known in the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

art, and a person of ordinary skill in the art would have found it advantageous and obvious to employ such optimized decoding techniques for any of the “suitable compression” methods (Carmel at 11:27-28) that were used—particularly in applications involving mobile clients using a wireless connection (as disclosed and rendered obvious by Carmel), which typically had/have relatively more processing and power constraints. (See, e.g., EX1039 at 1119, 1122-25 (teaching optimizations to the synthesis filter and providing a parameter analyzer block to take advantage of a multichannel computing structure for a MPEG-2 audio decoder); EX1040 at 671 (teaching “optimization techniques to implement MPEG audio decoding algorithms [MPEG-1/2 Layer III (MP3) and MPEG-2 AAC (Advanced Audio Coding)] in real-time using [a] general purpose 32-bit MCU RISC processor” that “would be beneficial to the overall power consumption of a portable MPEG player.”); 672-75 (describing optimization techniques for control-intensive and computationally-intensive aspects of audio decoding algorithms, including optimizations for reducing the computational loads of memory access and multiplications); EX1041 at 1, 12-13 (stating that optimized source code and binary implementations for MPEG-4 High Efficiency AAC [HE AAC] audio decoders were available and that “in mobile devices, computational complexity and memory requirements are directly related to power consumption and implementation cost. HE AAC was specifically designed to

minimize these resources”), 14; EX1042 at 376-78, 379 (describing audio decoding optimization techniques that avoid computing non-useful “zero-filled” data and noting that particular techniques (*i.e.*, “cut-off optimizations”) are scalable and can be adjusted to gracefully degrade to meet any real-time decoding deadlines, making these algorithms particularly useful in portable devices); EX1043 at 303-06 (teaching several optimized processes for decoding MPEG audio); EX1047 at 117-18, 120-22 (describing “optimization of the Huffman decoding strategy for AAC decoders” and providing a “new table lookup approach which significantly reduces the computational requirements.”).)

208. Accordingly, Carmel in view of Mattavelli renders obvious utilizing optimized decoding processes to maintain quality playback (*e.g.*, by employing optimized computationally graceful decoding processes when (and only to the extent) needed when the complexity of the decoding tasks exceeds the available processing power in order to maintain the highest perceived quality playback).

(1) Manner and rationales for combining the teachings of Carmel and Mattavelli

209. A person of ordinary skill in the art would have found it obvious and would have been motivated to combine the teachings of Carmel and Mattavelli, predictably resulting in the multimedia transmission and playback system taught by Carmel which uses Mattavelli’s computational graceful degradation (CGD)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

techniques to optimize the decompression and decoding of the received slices of audio and video data for playback. A person of ordinary skill in the art would have understood that using Carmel's method of streaming video and audio data, which has been encoded and compressed "using any suitable compression method known in the art" (Carmel at 11:27-28), to a recipient client device that uses Mattavelli's CGD decoding techniques for media playback would have been successful. (*See, e.g.,* Mattavelli at 2309 ("The presented approach can be virtually applied to any video compression standard and processor based platform"), 2312 ("Simulation results for MPEG-4 video decoding demonstrate the successful implementation of the CGD approach.... Moreover, the presented approach may enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of *worst case decoding* has to be guaranteed." (italics in original)).)

210. Carmel and Mattavelli are both directed towards the processing of multimedia data ultimately for playback "for the appreciation of a user." (*See* Carmel at 10:49-50.) Both references teach methods and techniques for attaining the shared goal of providing the highest perceived audio and video output quality possible or practicable in view of system constraints, while ensuring playback continuity. (*See* Carmel at 10:48-54 ("In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user. Time stamps in the data

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.”), 11:9-18 (“Periodically, client 30 makes an assessment of the rate of data transfer over the link from the server and, if necessary, changes the quality level accordingly. For example, if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, the client will preferably select a lower quality level if one is available. On the other hand, if the rate is substantially higher than what is needed to receive the successive slices on time, the client may select a higher quality level to take advantage of the available bandwidth.”), 7:35-49, Figs. 3D, 6B; Mattavelli at 2309 (“The goal is to provide results of possibly lower visual quality, but respecting the real-time constraints, when the required processing needs exceed the available processing resources.”), 2309-10 (“In our case, the goal is to guarantee real-time video decoding with possible degradations of the image quality, i.e[.] reductions of the peak signal to noise ratio (PSNR), without losing parts of the image content. Such alternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, as can be easily understood, can have catastrophic consequences on prediction based coding systems.”).) As Mattavelli teaches, abrupt interruptions in media playback

“are perceptually very annoying” and, accordingly, “it is evident the importance of techniques” that eliminate or reduce such perceptually annoying outcomes. (Mattavelli at 2310.)

211. Both references also refer to and recognize the encoding and compressing of multimedia data (which requires subsequent decoding) according to well-known coding/compression standards.⁹ (*See, e.g.*, Carmel at 7:25-26, 11:24-52; Mattavelli at 2309, 2310-11.) Both references teach techniques that involve the encoding and/or decoding of multimedia data as part of a process for streaming and playing multimedia content to an end user. Further, Mattavelli teaches a type of dynamically scalable video decoding processes at the decoder that is responsive to varying levels decoding complexity and available processing resources, while

⁹ As discussed above, a POSA would have appreciated that using MPEG-4 compression standards would have been particularly suitable in applications with a mobile client (as taught and rendered obvious by Carmel) due to its high compression efficiency, robustness in error prone environments, and because of the decoding complexity metadata it can provide, which facilitates the computational graceful degradation at the decoder/mobile client taught Mattavelli. (*See, e.g.*, EX1021 at 6-7, 10, 40-41, 77-79, 295; EX1044 at 52 (MPEG-4 provides error robustness and resilience “due to the rapid growth of mobile communications, it is extremely important that access is available to audio and video information via wireless networks. This implies a need for useful operation of audio and video compression algorithms in error-prone environments at low bit-rates (i.e., less than 64 kbit/s).”).) A person of ordinary skill in the art would have found using MPEG-4 compression standards particularly suitable and obvious in a system combining the teachings of Carmel and Mattavelli for these reasons.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Carmel teaches a type of scalable multimedia transmission system where the encoded quality level of the transmitted slices of data is responsive to the available network bandwidth. (See Mattavelli at 2309; Carmel at 10:64-11:22, Fig. 6B.) The concept of, incentives for, and advantages of scalable multimedia coding and decoding were well-known to those of skill in the art. (See, e.g., EX1028 at 142-44 (scalable video coding and decoding capabilities in MPEG-4, noting ability to adapt to constraints including decoder complexity/power and network capacity); EX1020 at 209-11 (acknowledging two types of constraints where scaling features can be implemented in two different ways: “When there are known bandwidth limitations, versions of the bitstream may be used that include only the base layer, or the base layer plus lower-order enhancement layers. Alternatively, all layers may be sent and the scaling decision left to the decoder. If the display device is low resolution, or if computational resources are insufficient, enhancement layers may be ignored.”), 211 (scalability advantageously “provides a wide range of coding and decoding capabilities”); EX1044 at 16 (simple scalable visual profile useful for “useful for applications which provide services at more than one level of quality due to [i] bit-rate[/bandwidth] or [ii] decoder resource limitations, such as Internet use and software decoding”), 64 (bitstream parsing can occur in the decoder); EX1062 at 2:24-27, 2:40-3:15 (“Layering here enables achievement of a high degree of

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

flexibility, in that: 1) if bandwidth is limited, even momentarily, the low-significance layers can be discarded by the transmitter or ignored by the receiver in a highly adaptive fashion, and 2) if the receiver has limited processing or working storage resources for decoding (or a low resolution/low fidelity output device that will not benefit from high quality input), it can limit itself to the high-significance layers, all while minimizing the perceived quality loss.”); EX1018 at 236-37.)

212. In addition to being directed to similar subject matter (and being within the same field of endeavor as the '849 patent), a person of ordinary skill in the art would have appreciated that Carmel and Mattavelli are complementary. Carmel and Mattavelli are primarily focused on mitigating different (though related) bottlenecks or constraints in the chain of transmission and processing of streamed multimedia data from the source to the ultimate user/consumer/player of the video and audio data.

213. As discussed above, Carmel is primarily focused on constraints imposed by the available network bandwidth. It teaches methods for mitigating the effects of these bandwidth constraints on playback continuity and quality by providing slices of multimedia data (including slices of video and audio data that have been encoded at multiple quality levels) for download by a client multimedia player where the bitrate/data rate of the slices generally corresponds to the network

bandwidth available for download. (*See, e.g.*, Carmel at Abstract, 9:6-9 (“Each of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36. A method for selecting and, as required, varying the level is described hereinbelow with reference to FIG. 6B), 9:13-17, 10:64-11:22, 11:40-12:17, Figs. 3D, 6B.) Carmel teaches that the slice download rate can thereby “keep[] up” with the multimedia slice output/play rate for a given (potentially varying) available network bandwidth. (Carmel at 7:36-41.)

214. Mattavelli is primarily focused on a different (though related) constraint in the playback of streamed, encoded multimedia data—namely, limitations on the processing resources available to the client/multimedia player when decoding a compressed bitstream. Also as discussed above, Mattavelli teaches that the processing demands of decoding a compressed multimedia bitstream will fluctuate with the complexity of the encoded bitstream and teaches optimized computational graceful degradation (CGD) techniques that are progressively employed when needed “when the required processing needs exceed the available processing resources.” (Mattavelli at 2309, 2310-12.) The optimized CGD techniques taught by Mattavelli are designed to simplify the decoding processes used to preserve processing resources, when needed, which degrades certain qualities of the video playback in an attempt to avoid or minimize more perceptually noticeable

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

and annoying playback issues such as abrupt interruptions. (Mattavelli at 2309-11, 2312 (“We have introduced the motivations and concepts of the CGD approach for the implementation of video decoders on processor based platforms.”); *see also* EX1024 at 1:8-24 (describing method for computational graceful degradation of audiovisual decompression that is useful where “channel capacity is limited and some form of quality of service guarantee is required” and noting “[i]t is common in the case of software decoding to employ some form of graceful degradation when the system resources is not sufficient to fully decode all of the video bitstream”), 2:27-34, 8:61-9:8; EX1029 at 373 (graceful degradation “algorithm can adapt itself to the available resources: depending on CPU utilization, some operations are done, simplified or even skipped. For instances the Inverse DCT can be truncated, or the sub-pixel interpolation filters of the motion compensation can be approximated.”); EX1039 at 1125 (“We devised two ways of gracefully degrading the audio output: Partial zeroing of the $D(m)$ window coefficients, and eliminating the upper half subband coefficients of the spectral image.” (italics in original)); EX1042 at 379 (scalable optimization techniques that provide a “continuous and soft degradation curve” in audio decoding).) Mattavelli teaches that its “approach is very attractive since it enables a more efficient usage the hardware processing power without needing to implement resources for worst case complexity decoding.” (Mattavelli at

2309; *see also id.* (contrasting processing power needed to guarantee worst case/highest decoding complexity versus processing power needed for average decoding complexity), 2312.)

215. A person of ordinary skill in the art would have been motivated to combine the complementary methods and techniques taught by Carmel and Mattavelli to form a streaming system that is more robust and better able to handle these multiple known potential bottlenecks or constraints in the chain of transmission of a multimedia bitstream from its source to the end user in a way that provides the best possible playback experience. A person of ordinary skill in the art would have recognized that a multimedia streaming system would potentially be subject to constraints on (i) available network bandwidth, (ii) multimedia player processing resources, or (iii) both at the same time, and would have been motivated to apply Mattavelli's teachings regarding CGD decoding techniques to the multimedia player/decoder on the (wireless) client 30 taught by Carmel to mitigate/address both of these known problems or bottlenecks in providing the best possible streaming playback performance.

216. Indeed, Mattavelli expressly teaches that certain techniques for optimizing usage of transmission bandwidth alone are not especially useful (and could even be potentially harmful) in addressing the potential processing resource

limitations faced by the decoder (*i.e.*, after the data has crossed the network and reached the client) that Mattavelli addresses. Mattavelli provides:

In contrast with what could be commonly thought, also the classical multi-resolution coding features such as spatial or temporal scalabilities are not really useful for our goals. In general, they are used to optimize the transmission bandwidth usage enabling the compatibility of services at different [b]itrates and spatial resolutions, but they must be embedded in the original bit-stream at the encoder side. If they are not embedded, the changes of resolution at the decoder size can results more computationally expansive than simple decoding at the given resolution.

(Mattavelli at 2310.) A person of ordinary skill in the art would have understood this disclosure to provide an express motivation to combine the complementary teachings of Carmel and Mattavelli, which address different (though-related) issues—optimizing transmission bandwidth usage for Carmel and optimizing decoder processing usage for Mattavelli—that can separately or together affect the quality and continuity of the playback of the multimedia stream. A person of ordinary skill would have recognized that a system that combined these teachings would advantageously optimize both of these related links (network bandwidth usage and decoder processor usage) in the streaming playback chain and the importance of optimizing both of these links together in a complementary way. (*See*

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Mattavelli at 2310 (“From these arguments[,] the usefulness of CGD and its important features and applications are clear.”).)

217. Indeed, as discussed more in the context of mobile/wireless client devices below, a person of ordinary skill in the art would have appreciated that utilizing optimized decoding processes (as taught by Mattavelli) would also help facilitate the upstream use of more efficient compression standards that, in turn, would facilitate more efficient/optimized use of the available network bandwidth, with better quality audio and video transmitted with less data. A person of ordinary skill would have recognized that Mattavelli’s optimized decoding processes would have allowed a decoder/client/media player to utilize its processing capacity more efficiently and effectively, allowing it to decode more efficient though more computationally demanding compression methods (including during spikes in decoding complexity) without the need for “sustained over-sizing of the processing resources[,]” which Mattavelli notes would not be desirable or economical. (Mattavelli at 2309; *see also id.* (Mattavelli’s “approach is very attractive since it enables a more efficient usage of the hardware processing power without needing to implement resources for worst case complexity decoding.”).) Mattavelli provides that its “approach may **enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of worst[-]case**

decoding has to be guaranteed.” (Mattavelli at 2312 (*italics omitted*).) A POSA would be motivated to implement Mattavelli’s optimized decoding processes in Carmel’s clients 30 to achieve these synergistic benefits, in which more efficient processor usage for decoding would lead to more efficient network bandwidth usage and, ultimately, improved/higher quality playback overall.

218. Carmel also contains disclosure that would have motivated a person of ordinary skill in the art to combine its teachings with those of Mattavelli. Namely, Carmel provides: “Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission **and decoding delay.**” (Carmel at 10:50-54; *see also id.* at 3:66-4:2, 5:22-25, 11:7-8.) Carmel thus expressly acknowledges that there are (at least) two potential sources of delay in playing back a multimedia sequence: (i) transmission delay and (ii) decoding delay. As discussed above, Carmel primarily teaches techniques for minimizing transmission delays. A person of ordinary skill in the art would have been motivated to use Mattavelli’s optimized CGD decoding techniques with the multimedia player taught by Carmel in order to minimize any potential decoding delays or problems referenced by Carmel, while maintaining quality and continuity.

219. Further, a person of ordinary skill in the art would have appreciated that

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

using Mattavelli's CGD decoding optimizations to ensure playback continuity in the face of processor constraints when decoding multimedia content of varying and unpredictable complexity would have been (and would be) particularly advantageous and beneficial in the context of the wireless network and mobile client implementations taught and rendered obvious by Carmel. (*See* claim element 1[b.ii] above; *see also* Carmel at 2:41-46, 6:65-7:3, 6:36-38, Fig. 2; EX1009 (Carmel '432) at 1:29-36, 3:21-35, 3:43-47, 3:48-53, Fig. 1; EX1010 (Carmel '230) at 11:50-57; Mattavelli at 2309 (“**The presented approach can be virtually applied to any video compression standard and processor based platform**”), 2312.) This is so because wireless/mobile client devices were understood to generally have comparatively less processing power and more processing constraints than non-mobile computing devices. (*See, e.g.*, EX1007 (Wang) at 51 (“But the performance of the optimized AVC modules **is seriously constrained by the low computation power** and small memory **for mobile devices** to achieve the least service requirement of QCIF 7.5 fps.”); EX1027 at 658 (“**Mobile devices** are hand-held and **constrained in processing power** and storage capacity.”); EX1021 at 295 (“Not all of the tools are required for every application, and in some cases, it may be preferable if a decoder not support certain tools (e.g., for reasons of complexity or computational load).”).)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

220. And, as suggested by Carmel and as a person of ordinary skill in the art would have appreciated, the (then-)newer, more efficient compression algorithms (e.g., MPEG-4 visual, H.264/MPEG-4 AVC) were/are able to provide higher quality using less data/lower bitrates but typically were generally more complicated and/or computationally expensive to implement and encode/decode. (See, e.g., Carmel at 11:36-38 (computer must be “sufficiently powerful” to use MPEG compression); EX1061 at 1:10-17 (“There are many international standards for digital video compression technologies, such as H.261, MPEG1, MPEG2 (H.262), H.263, H.263+, MPEG4 and H.264 (MPEG4 part 10), that have emerged one after another during the 1990s. Generally[,] algorithms recommended by newer standards are better, but also are usually more complicated to implement. It is easy to find examples of trading algorithm complexity for efficiency.”); see also Wang (EX1007) at 51.) A person of ordinary skill in the art would have been particularly motivated to combine the teachings of Carmel with Mattavelli’s teachings regarding optimized decoding processes in implementations involving a mobile or wireless client for this reason—to facilitate the use of more efficient compression standards (to provide higher quality multimedia playback using less data/lower bitrates) in applications that include processing power-constrained wireless client devices. In other words, a person of ordinary skill in the art would have recognized that utilizing

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Mattavelli's CGD decoding optimizations would have beneficially expanded the set of "suitable" (to use Carmel's terminology) encoding/compression methods—including more efficient, though more computationally expensive, compression methods—for use in implementations that included wireless clients. (Mattavelli at 2309 (Mattavelli's "approach is very attractive since it enables a more efficient usage of the hardware processing power without needing to implement resources for worst case complexity decoding."), 2312 (Mattavelli's "approach may **enable the successful implementation of video decoding for levels and profiles that would be forbidden if the complexity of *worst/-/case decoding* has to be guaranteed.**" (italics in original)).)

221. Additionally, a person of ordinary skill in the art would have considered a system that incorporated the bandwidth optimization techniques taught by Carmel with the decoder optimization techniques taught by Mattavelli to be an obvious combination of known methods to address known issues (*e.g.*, bottlenecks or constraints on network bandwidth and decoder processing resources) to yield predictable results—*i.e.*, a system where a client multimedia player/decoder advantageously optimizes both the available network bandwidth and the available decoding resources to improve or get the best possible the playing/outputting of a multimedia data stream "for the appreciation of a user." (Carmel at 10:48-49.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Similarly, a person of ordinary skill in the art would have found the combined teachings of Carmel and Mattavelli to be an obvious application of a known technique (*e.g.*, the CGD decoding techniques taught by Mattavelli; *see also, e.g.*, EX1024 at 1:8-24; EX1026 at 1:8-19) to a known device, method, or product ready for improvement (*e.g.*, the streaming method and system taught by Carmel) to yield predictable results (*e.g.*, a method and system that optimizes both the bottlenecks of the available network bandwidth and the available decoding resources when playing/outputting a multimedia data stream for the appreciation of a user).

222. Indeed, both network bandwidth/bitrates and decoder processor resource limitations were known issues to be addressed when streaming media. (*See, e.g.*, EX1020 at 209-11 (acknowledging two types of constraints where scaling features can be implemented in two different ways: “When there are known bandwidth limitations, versions of the bitstream may be used that include only the base layer, or the base layer plus lower-order enhancement layers. Alternatively, all layers may be sent and the scaling decision left to the decoder. If the display device is low resolution, or if computational resources are insufficient, enhancement layers may be ignored.”), 213-15 (Simple Scalable Visual Profile “is useful for **applications that provide services at more than one level of quality due to [i] bit-rate or [ii] decoder resource limitations**, such as Internet use” and Core Scalable Profile “adds

support for coding of temporal and spatial scalable arbitrarily shaped objects to the Core Profile ... [and] is useful for applications such as the Internet, mobile, and broadcast.”), 218-19 (“The *Fine Grain Scalability Profile* (FGS), often known as the *Streaming Profile* is intended to support applications and environments **where the [i] bandwidth and/or [ii] computational power available cannot be predicted and may vary dynamically.**” (italics in original)), 220 (“A decoder uses only as much of the bitstream as its resources permit. It may ignore temporal enhancement layers to minimize the number of VOPs to be processed, and it can take only as much data as it can handle from the other layers.”) EX1028 at 142-44 (scalable video coding and decoding capabilities in MPEG-4, noting ability to adapt to constraints including decoder complexity/power and network capacity); EX1062 at 2:24-27, 2:40-3:15 (“flexible scalability” when encoding/decoding data useful in applications with (i) “varying bandwidth” and/or (ii) decoders with different or “limited processing or working storage resources for decoding (or a low resolution/low fidelity output devices that will not benefit from high quality input)”); EX1018 at 236-37.)

2. Claim 2: “The method of claim 1, wherein the transmission of the multimedia objects is carried out by protocols that can serve files in a directory.”

223. Carmel discloses and renders obvious the method of claim 1, “wherein

the transmission of the multimedia objects is carried out by protocols that can serve files in a directory.”

224. Carmel teaches the transmission of multimedia objects (e.g., slices/files of audio and video data) using any one of multiple well-known Internet protocols, including the File Transfer Protocol (FTP) and the Hypertext Transfer Protocol (HTTP). For example, Carmel provides:

In preferred embodiments of the present invention, a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients. The data stream is divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed. Each slice is preferably assigned a respective slice index. **The transmitting computer uploads the sequence of slices to the server substantially in real time, preferably using an Internet protocol, most preferably the File Transfer Protocol (FTP), as is known in the art. The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP), or alternatively, using other protocols, such as UDP or RTP, which are similarly known in the art.**

(Carmel at 2:1-15; *see also id.* at 2:22-28, 3:60-67, 5:19-28.) Carmel teaches that transmitting computer 34 transmits/uploads a sequence of multimedia slices to server 36, preferably using FTP, and that clients 30 download the sequence of

multimedia slices from server 36, preferably using HTTP. (Carmel at 6:50-54, 7:4-9 (also indicating that objects could be downloaded by client and served using FTP), 13:29-35 (teachings directed to upload are applicable to download as well); Fig. 2.)

Carmel further provides:

Since FTP and HTTP are supported by substantially all network servers, server 36 need not include any special-purpose broadcasting hardware or software, as noted above. Similarly, because HTTP is supported by substantially all modern Web browsers, clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers, as described further hereinbelow, in order to receive and play back the broadcast.

(Carmel at 7:9-17.) Carmel teaches and recognizes that these protocols, including FTP and HTTP were known in the art, and a person of ordinary skill in the art would have understood their characteristics, features, and functions, including when supported by or used by a server, like host server 36.

225. Carmel's teachings regarding the use of FTP and HTTP to transmit slices of multimedia data implicitly disclose the transmission of multimedia objects being carried out by protocols that can serve files in a directory. As a person of ordinary skill in the art would have understood, both FTP and HTTP are protocols that can serve files in a directory (when those files are stored in directories on a server). (*See, e.g.*, EX1006 at 25 (referencing changing directories on FTP server),

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

39 (FTP can be used to upload and download files and obtain the contents of a directory structure on an FTP server), 122-23 (discussing directories (and directory URLs) on web/HTTP servers), 126-27, 200-01 (discussing use of HTTP and FTP (and HTTP/FTP gateways) including commands to change to the proper directory on the server), 220-21, 229, 439-41; EX1023 at 528-29 (providing example of URLs (accessed using http) that include directory path information); EX1022 at 293-307 (discussing how to configure an HTTP server/site, including setting up directories), 317-18 (discussing how to set up an FTP site as similar to a (HTTP) website, stating “You first design your FTP directory structure, create the directory tree, and populate it with the files you want available for download”).)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

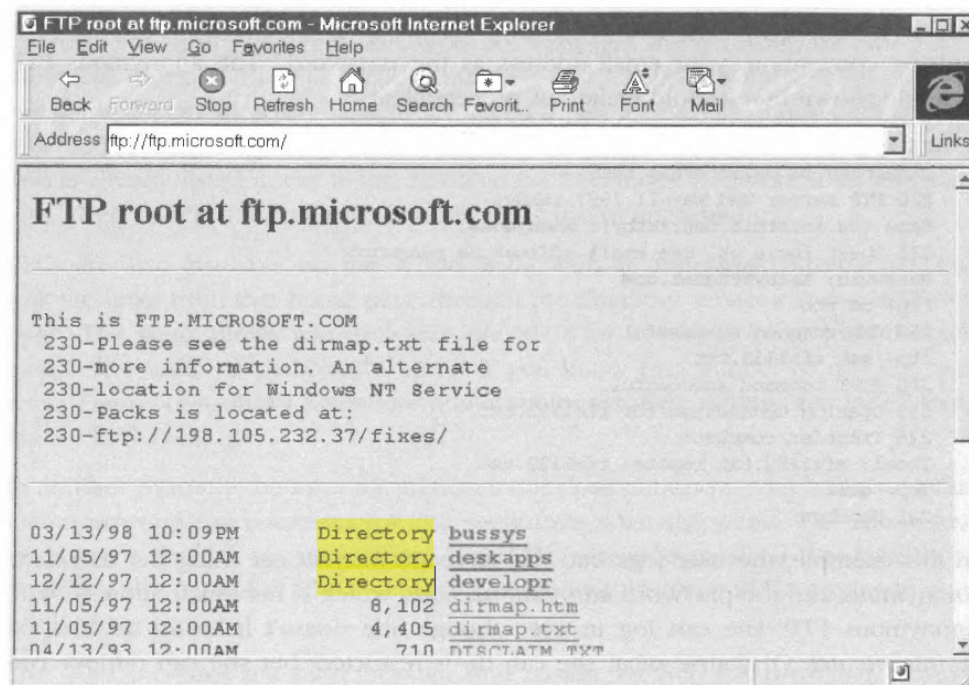


Figure 13-3. Using the Web for file transfer

The URL in Figure 13-3 shows HTTP being used to retrieve a file. FTP can also be the service used to access a resource through the browser. For example, *ftp://ftp.microsoft.com* connects to the anonymous FTP server at Microsoft. Files can be downloaded via HTTP or FTP.

(EX1022 at 415.)

226. Indeed, the '849 patent acknowledges this well-known characteristic of HTTP and FTP. (See '849 patent at 3:30-35 ("The transport mechanism for the multimedia objects is assumed to be **HTTP** for the purposes of illustration, however, other protocols which access content through file and directory structures could be used. For example[,] **FTP**, IMAP4 and NNTP **all have the capability to serve files in a directory structure.**".).)

227. In addition to implicitly disclosing the transmission of multimedia

objects by protocols that can serve files in a directory, these teachings of Carmel, in view of the knowledge of one of ordinary skill in the art, also render this claim element obvious. For the reasons just discussed, a person of ordinary skill in the art would have understood that the well-known FTP and HTTP protocols were (and are) protocols that can serve files in a directory.

3. **Claim 3: “The method of claim 1, wherein the multimedia objects are created from a live multimedia stream that is transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth and converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients.”**

228. Carmel discloses and renders obvious the method of claim 1 “wherein the multimedia objects are created from a live multimedia stream that is transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth and converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients.”

229. First, Carmel teaches and suggests the creation of multimedia objects (e.g., slices or corresponding files of multimedia video and audio data) from a live multimedia stream. Carmel provides that the source multimedia sequence “is **preferably generated and compressed in real time, and could comprise, for**

example, an interview program or an entertainment or sports event, although a prerecorded sequence may similarly be broadcast in this manner.” (Carmel at 6:57-60.) By contrasting it with a prerecorded sequence, Carmel is teaching that a multimedia sequence “generated and compressed in real time” is a live multimedia stream. Carmel also has additional teachings referencing a **“live broadcast transmission”** and the receipt of “audiovisual input from input devices 22” (*e.g.*, a video camera and microphone)—contrasted with data stored on a disk or tape—that further disclose and suggest the creation of multimedia objects from a live multimedia stream. (*See* Carmel at 12:54-58, 9:64-10:1 (“Broadcast data are then input to the computer, for example, from input devices 22, or from a video, audio or animation sequence stored on disk or tape.”), 6:24-35, 1:23-33, Fig. 2.)

230. Next, Carmel also teaches and suggests that the multimedia objects (*e.g.*, slices and/or corresponding files of the multimedia stream) are transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth. For example, Carmel provides:

FIG. 7 is a flow chart that schematically illustrates details of encoding step 80 and slicing step 82 in the method of FIG. 5, in accordance with a preferred embodiment of the present invention. In encoding data stream 40, computer 34 preferably compresses the data using any suitable compression method known in the art.

For example, if data stream 40 comprises audio data, GSM 6.10 standard encoding may be used, as is known in the art, to compress the data by about 10:1. Alternatively or additionally, for video data, H.263 standard compression, similarly known in the art, may be used. These compression standards are advantageous in that common personal computers can perform such compression in real time, in parallel with the other operations illustrated in FIG. 5. Other compression methods known in the art, such as MPEG data compression, may similarly be used, as long as computer 34 is sufficiently powerful.

Computer 34 determines a compression ratio by which to compress the data, based on the collective bandwidth of its open links with server 36. Preferably, computer 34 receives an indication of the bandwidths of the links, determined at step 88 in FIG. 5, and adjusts the compression ratio accordingly, at a set compression step 90. For example, the compression ratio may be adjusted by changing compression coefficients (e.g., MPEG coefficients) so as to match the data stream bandwidth to the available link bandwidth. Methods of adaptively varying the compression ratio of a multimedia data stream that can be used for this purpose are described, for example, in the above-mentioned U.S. patent application Ser. No. 08/919,027.

(Carmel at 12:23-38, 11:23-52; *see also id.* at 9:64-10:1, 10:19-23, 8:56-9:5

(“Typically, the **audio and video data in level #1, contained in slices 45 and 49, are more highly compressed relative to the data in slices 47 and 51 of level #2.**

In consequence, the level #1 slices have smaller data volume than the level #2 slices

and can therefore be **transmitted over a lower-bandwidth data link, while maintaining the required slice timing indicated by time stamps 59.**”), 9:46-52, Fig. 3D.)

FIG. 5

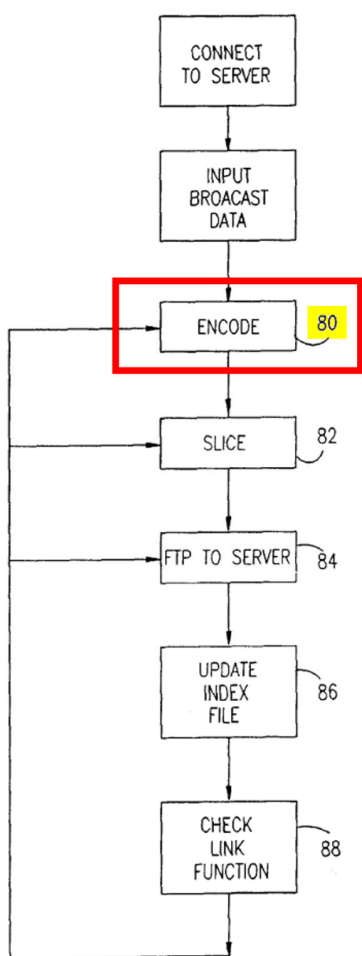
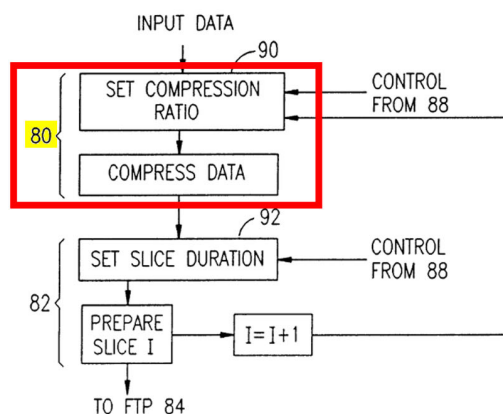


FIG. 7



(Carmel at Figs. 5 & 7.) Carmel teaches and a person of ordinary skill in the art would have found it obvious that Carmel’s teachings regarding selecting the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

compression ratio based on the available bandwidth between the transmission computer 34 and the server 36 would be equally applicable to selecting the compression ratio based on the available bandwidth between the server 36 and client 30. (*See id.* at 2:42-50, 3:63-4:2 (“Preferably, the one or more client computers decode the sequence and play back the data stream responsive to the indices of the files, at a replay rate generally equal to the data rate.”), 4:39-47 (“encoding the slices includes encoding slices at a plurality of different quality levels, such that the files corresponding to a given one of the slices have a different, respective data size for each of the quality levels. Preferably, downloading the sequence includes determining a data bandwidth of the network between the server and the client computer and selecting one of the quality levels responsive to the determined bandwidth.”), 7:44-49 (“Alternatively or additionally, **the compression level of the data is varied**, as is likewise described below, **so as to adjust the data streaming rate to the available bandwidth over one or more channels** between computer 34 and server 36, **and/or between server 36 and client 30.**”).) Indeed, in circumstances where the available bandwidth between the server and client was expected to be the same as or lower than the available bandwidth between the transmission computer and the client, a person of ordinary skill in the art would have been motivated and found it obvious to use a compression ratio that reflected this lower available

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

bandwidth. (*Id.*)

231. Carmel thus teaches and renders obvious encoding/transcoding/compressing the multimedia stream into an optimal digital format (*e.g.*, efficient compression formats resulting from use of the H.263, MPEG-2, MPEG-4,¹⁰ or other suitable compression standards) at an optimal encoding rate reflecting available network bandwidth (*e.g.*, at an encoding rate that reflects the available network bandwidth for uploading the slices/multimedia objects from

¹⁰ As discussed above for claim 1, in view of Mattavelli's teachings regarding the complexity prediction information/metadata that can be included with MPEG-4 formats, as well as the compression efficiency and robustness to transmission errors that can be achieved, which is important for mobile applications, a person of ordinary skill would have found it obvious and have been motivated to utilize these compression standards and understood it to be an "optimal digital format," for such mobile/wireless applications taught and rendered obvious by Carmel and Mattavelli. (*See, e.g.*, EX1021 at 6-7 ("The growth of mobile networks creates an ongoing demand for improved coding efficiency; therefore MPEG-4 set as its target providing subjectively better audiovisual quality compared to existing or other emerging standards.... Particularly, sufficient error robustness shall be provided for low bit-rate applications under severe error conditions" with sample uses including communicating with a mobile terminal); EX1027 at 657 (3G wireless systems first provided video services using integrations of H.263 and MPEG-4 visual simple profile and because of its compression efficiency, H.264/AVC coding was also an attractive candidate for wireless video and multimedia applications).) I note that the '849 patent itself also repeatedly references MPEG-4 encoding stating that it is a particularly efficient and suitable format for mobile/wireless devices and thereby indicating that it is an "optimal" format for such applications. (*See, e.g.*, '849 patent at Abstract, 1:21-32, 1:57-58, 2:50-3:16 (source multimedia data is converted "into true MPEG4/AAC multimedia objects"), 4:44-57, 7:57-8:7, 18:64-19:9, Figs. 1, 3a, 6.)

transmitting computer 34 to server 36 and/or at an encoding rate that reflects the expected network bandwidth(s) clients 30 may have available when downloading the slices from server 36, including in the multi-level data stream embodiment).

232. I note that Carmel uses the term “encode” (or “encoder”) when referring to using well-known coding/compression standards (*e.g.*, H.263, MPEG-4) to compress the multimedia data stream. (*See, e.g.*, Carmel at Figs. 5 & 7.) A person of ordinary skill in the art would have understood that the terms “encode” or “encoder” can be used broadly to refer to any process (or hardware or software) that encodes information—“that is, converts the information to a particular form or format. For example, the Windows Media encoder converts audio and video to a form that can be streamed to clients over a network.” (EX1011 at 192 (definition of “encoder”).) In other words, the terms “encoding” and “transcoding” are often used synonymously to mean any conversion of data from one form to another. (*E.g.*, EX1048 at 1:16-20 (“Transcoding, as it is understood in the art, refers to the technique of transforming multimedia content (*e.g.*, text, images, audio, video and the like) from a first original format, in which the multimedia content was encoded, into a second format.”).) Indeed, the ’849 patent appears to treat these terms as synonyms. (*See, e.g.*, ’849 patent at Abstract (“A host webserver transcodes a live digital or analog audio-visual or audio broadcast signal and splits the input stream

into small multimedia objects of an efficient compression such as MPEG4/AAC.”), 7:57-64 (transcoder 2 encodes multimedia objects with a certain encoding (MPEG4/AAC) from an analog multimedia or alternatively encoded stream).) The ’849 patent does not describe or require that the source or input data stream is compressed, which thereby indicates that “transcoding” does not necessarily require first decoding/decompressing a compressed input stream. Thus, under this understanding of the term “transcoding,” a person of ordinary skill in the art would understand Carmel’s references to “encoding” the input multimedia data stream (e.g. from input devices/video camera and microphone 22 with an original digital or analog format or from a prerecorded file) with a well-known coding/compression standard such as MPEG-4 to constitute “transcoding” this input stream via its conversion from one data format to another.

233. Even if a narrower understanding of the term “transcoding” is used, however, which requires a compressed input stream which is converted from one type of compression to another, Carmel’s teachings, in view of the knowledge of one skilled in the art, render this claim term obvious. This is so because: (i) Carmel’s teachings regarding the video camera input device 22 shown in Figure 2 or using a prerecorded multimedia sequence (*see* Carmel at 6:24-35, 9:64-10:1, Fig. 2) reasonably suggest and render obvious a digital video camera, which were known in

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

the art, and (ii) that a person of ordinary skill in the art would have understood that the video input stream from a digital video camera input device 22 shown in and suggested by Carmel Figure 2 or a prerecorded sequence typically will have been compressed using a conventional video compression process. (*See, e.g.*, EX1012 at 8:35-37 (“Typically, the input video will have been compressed using a conventional video compression process.”), 8:16-29 (describing the process for capturing an input video containing using a video camera, which could be a digital video camera (in which case “the video can be input directly ... into a computing device that is used to encode it”) or a non-digital video camera (where the feed is first “digitized via conventional methods before being input into the computing device for encoding”); EX1013 at [0025] (describing use of digital video camera producing a lightly compressed video stream); EX1021 at 2 (“Together with the first digital video cameras directly recording in MPEG-1 format, this represents a major step for consumer acceptance of digital audiovisual acquisition and compression technology.”); EX1005 at 3-4 (explaining storage and transmission benefits of compression); EX1020 at 3-4 (compression used to reduce transmission and storage costs).)

234. Further, a person of ordinary skill in the art would have understood and found it obvious that encoded/compressed input video and/or audio data should and

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

would first be decompressed (at least to some extent) before it is re-encoded using a different, optimal codec or compression/coding standard at an optimal (also potentially different) encoding rate for streaming across the Internet. (*See, e.g.*, EX1012 at 8:37-41 (“At block 204, the video portion of the frame, which constitutes image frames of the video, is decompressed if necessary. Standard decompression techniques are employed for this task and will depend on the process used to initially compress the input video.”), 8:48-59 (describing decompressing (when necessary) and recompressing the input video and audio), Fig. 2; EX1013 at [0025] (“If a digital video camera is the only camera allowed, the base unit 12 includes only a digital video decompressor 18 that decompresses the lightly compressed AVI signals output by digital video cameras to output a digital format suitable for manipulation of the video image to mix in status data and suitable for [re-]compression.”); EX1049 at 2:6-50, Figs. 1, 2 (transcoding techniques for converting a input bitstream to a different scaled output bitstream that involved fully or partially decoding the input stream before re-encoding the stream were known in the art); EX1058 at 2:1-4, 2:16-31 (compressed input streams are first decompressed before being re-encoded for transmission to match the requirements of the destination terminal (including encoded parameters such as bit rate, frame rate, frame resolution, compression algorithm, etc.)); EX1061 at 1:31-39.) A person of ordinary skill in the art would

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

have understood that was not common or advisable to apply an additional lossy compression technique (*e.g.*, MPEG-4) on already compressed data without first decompressing/decoding it, at least to some extent, before re-encoding/re-compressing the data. (*See id*; *also cf.* EX1059 at 2:1-6 (“It is generally considered ‘impossible’ to meaningfully compress already compressed data. More accurately, perhaps, it should be said that it is considered impractical to compress already compressed data, though it is true that most attempts at compression of already compressed files fails altogether and actually results in an increase in file size.”).)

235. Thus, Carmel’s teachings discussed above, in view of this knowledge of one skilled in the art, render obvious transcoding the input data stream into an optimal digital format at an optimal encoding rate by, when necessary, first decoding a compressed input data stream (using whatever corresponding decoding codec that was used to encode it) and then re-encoding the data stream with the desired “suitable”/optimal coding/compression standard at the desired encoding rate that reflects the available network bandwidth for transmission to wireless/mobile clients. (*See Carmel at 11:23-52, Fig. 7.*)

236. Finally, for the reasons I previously discussed for claim elements 1[a.i], 1[a.ii.], and 1[b.ii] and incorporate here, Carmel discloses and renders obvious “converting the input multimedia stream into audio and video segments by splitting

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

the encoded stream into intervals for access by wireless clients.”

237. In brief summary, Carmel teaches converting an input multimedia stream into audio and video segments by splitting the encoded stream into “slices” or intervals—*e.g.*, of a fixed time length or data size. (*See, e.g.*, Carmel at Abstract (“A method for real-time broadcasting from a transmitting computer to one or more client computers over a network, including providing at the transmitting computer a data stream having a given data rate, and dividing the stream into a sequence of slices, each slice having a predetermined data size associated therewith.”), 7:18-49, 8:42-9:9, 9:53-23, 11:23-12:17, Figs. 2, 3A, 3D, 5, 6B, 7.)

FIG. 5

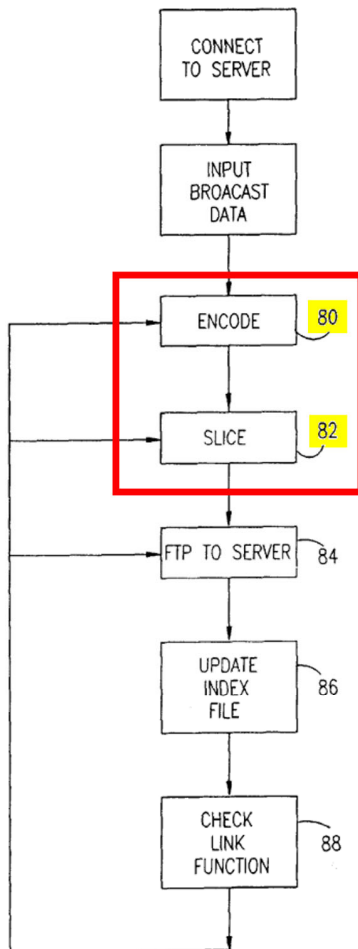
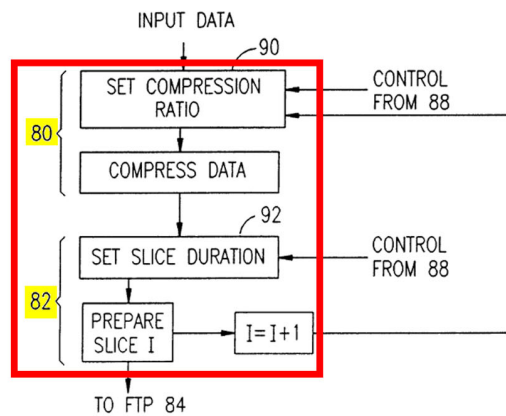
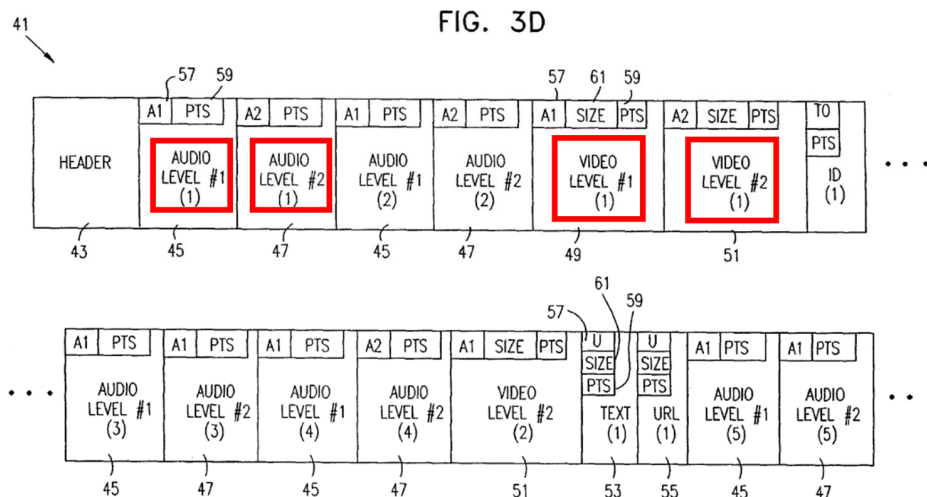


FIG. 7



(Carmel at Figs. 5 & 7.)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2



(Carmel at Fig. 3D.)

238. And, for the same reasons that Carmel teaches and renders obvious client 30 (and the multimedia player thereon) receiving the multimedia objects/slices using a wireless connection that I discussed in claim element 1[b.ii] above, Carmel also teaches and renders obvious client 30 being a wireless client. Carmel's teachings of a generic client computer 30 and its incorporated teachings from Carmel '432 and Carmel '230 teach and render obvious a wireless client recipient. (*See, e.g.*, Carmel at 2:41-46, 6:65-7:3, 6:36-38, Fig. 2; Carmel '432 at 1:29-36, 3:21-35, 3:43-47, 3:48-53 ("Computer 19 may be connected by a telephone line 27, either physically or through wireless connections, to the network interconnections 17 of FIG. 1."), Fig. 1; Carmel '230 at 11:50-57 ("A recipient computer 26 may connect to server 24 via public phone lines 28. Alternatively, other connection methods

may be used, such as satellite or cable connections.”.)

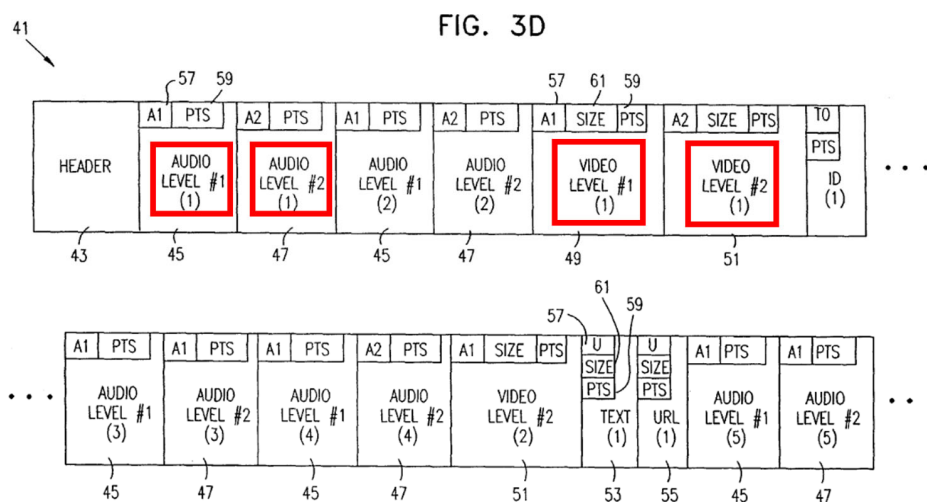
4. Claim 4: “The method of claim 1, wherein a multimedia file is scanned for an I-frame, and a multimedia segment is split at the I-frame to create the video segment, together with a separate audio segment.”

239. Carmel, in view of Mattavelli and knowledge of one of ordinary skill in the art, renders obvious the method of claim 1, “wherein a multimedia file is scanned for an I-frame, and a multimedia segment is split at the I-frame to create the video segment, together with a separate audio segment.”

240. Carmel teaches that multimedia data, “for example, from input devices 22, or from a video, audio or animation sequence stored on disk or tape” are input to transmitting computer 34 and then compressed at step 80. (*See* Carmel at 9:62-67, Fig. 5.) As previously discussed, Carmel further provides that “in encoding data stream 40, computer 34 preferably compresses the data using any suitable compression method known in the art,” and provides examples of such well-known video and audio coding formats such as H.263 and MPEG (*e.g.*, MPEG-2, MPEG-4) that would be suitable for use. (Carmel at 11:23-52, Fig. 7.) Also as previously discussed, and incorporated here, Carmel teaches that after encoding a multimedia file, it is split into “slices” containing video segments together with separate audio segments. (*See, e.g.*, Carmel at 7:18-34 (“Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T_1 ,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

T₂, T₃, etc.”), 8:42-9:5 (“The data stream is divided into audio slices 45, 47 and video slices 49, 51.”), 9:62-10:1 (“The data are compressed at step 80, and are then ‘sliced’ at step 82 into files 42, 44, 46, 48, etc.”); Figs. 3A, 3D, 5.) Further, many of the well-known “suitable compression method[s] known in the art” separately the encode video and audio streams, which are composited or brought back together and synchronized at the decoder. (*See, e.g.*, EX1020 at 216 (describing how video and audio objects are transmitted independently and composited at the decoder).) Carmel’s teachings, combined with the knowledge of a person of ordinary skill in the art regarding these well-known compression standards, further suggests and renders obvious splitting a multimedia stream into separate (though temporally corresponding) video and audio segments that are recombined at the decoder.



(*See Carmel at Fig. 3D.*)

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

241. Further, in view of these teachings, as well as those in Mattavelli and the knowledge of those of ordinary skill in the art, when splitting or “slicing” the compressed/encoded video data, it would have been obvious to scan the file for an I-frame and split the file/stream such that each respective slice of video started with an I-frame.

242. The exemplary formats for compressing video, such as H.263 and MPEG (which includes, e.g., MPEG-4 standards and profiles), that Carmel identifies were indeed known in the art, and it was known by those of ordinary skill that these video compression formats generally employed: (i) intra-coding to exploit spatial redundancy or redundancy within the pixel data within a particular frame/picture combined with (ii) inter-coding to exploit temporal redundancies or redundancies in the pixel data between successive frames or series of frames. (*See* EX1005 at 13-17, 243-48; EX1014 at 3-4 (*see also* EX1015).) It was further known in the by those of ordinary skill in the art that video compression formats/standards including H.263, MPEG-2, and MPEG-4 made (and make) use of predictive coding to exploit these temporal redundancies between successive frames of video, using three different types of frames/pictures that are divided into groups of frames/pictures called a Group of Pictures (GOP). For example, for encoded MPEG video stream:

[An] encoded MPEG video stream is comprised of a series of groups of

pictures (GOPs), and **each GOP begins with an independently encoded (intra) I frame and may include one or more following P-frames and B-frames. Each I frame can be decoded without information from any preceding and/or following frame.** Decoding of a P frame requires information from a preceding frame in the GOP. Decoding of a B frame requires information from a preceding and following frame in the GOP. To minimize decoder buffer requirements, each B frame is transmitted in reverse of its presentation order, so that all the information of the other frames required for decoding the B frame will arrive at the decoder before the B frame.

(EX1016 at 1:34-47; *see also* EX1005 at 15-16, 248-49 (“An *I* picture [which contains ‘picture information in absolute form’] together with all of the [predicted] pictures before the next *I* picture form a *group of pictures* (GOP).”) 248-57; EX1014 at 15 (“In MPEG, three different types of pictures are need to support differential and bidirectional coding while minimizing error propagation. *I* pictures are Intra-coded pictures that need no additional information for decoding.... *P* pictures are forward Predicted from and earlier picture.... *B* pictures are Bidirectionally predicted from earlier or later *I* or *P* pictures.”); EX1017 at 1:34-47 (defining and explaining the characteristics of *I*, *P*, and *B* pictures); EX1018 at 237 (same).)

243. When employing a video coding format/standard such as H.263, MPEG-2, or MPEG-4 (as taught by Carmel) that used temporal coding and predicted

frames, a person of ordinary skill in the art would have been motivated and found it obvious to start each “slice” of a video segment (as taught by Carmel) with an I-frame (that, is the starting point of a GOP) in order to ensure that the slice is able to be decoded properly from the very first frame. If a video segment slice was created arbitrarily, with no attention paid to what type of frames were at the start, this proper decoding from the first frame of the slice could not be guaranteed. A person of ordinary skill in the art would have understood that if a slice were to start with a sequence of predicted frames (P-frames or B-frames), there was a significant risk that the video slice would not be decoded properly at the start because the predicted frames lacked the reference I-frame on which their encoded difference data was based.

244. Indeed, Mattavelli recognized this issue and consequences resulting from predicted frames missing their reference frames. Mattavelli provides:

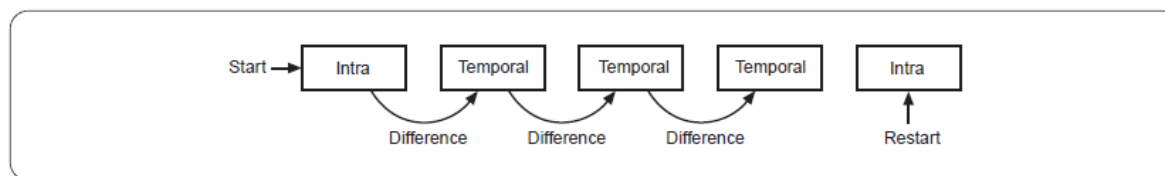
Such alternative is certainly much more desirable than other straightforward options such as cutting image portions or simply skipping frames. These latter options, **as can be easily understood, can have catastrophic consequences for prediction based coding systems. In H.263 decoders, for instance, skipping the decoding of a single frame or portion of it, implies the skipping of all future incoming frames (or portion of them) before the next incoming Intra frame.** The time interval of such skipping operation might cover

many seconds and the information content loss might be very relevant.

Such abrupt interruptions ... are perceptually very annoying.

(Mattavelli at 2309-10.) This issue with frame/data loss in predictive coding was well-known and understood in the art. (*See, e.g.*, EX1005 at 248 (“Predictive (P) coding cannot be used indefinitely, as it is prone to error propagation. **A further problem is that it becomes impossible to decode the transmission if reception begins part-way through.**”); EX1016 at 2:11-19 (“The splicing of MPEG encoded audio/visual streams, however, is considerably more difficult than splicing of the uncompressed audio and video. The P and B frames cannot be decoded without a preceding I frame, **so that cutting into a stream after an I frame renders the P and B frames meaningless.**”); EX1045 at 4:61-5:8 (when a reference frame is dropped “the decode core circuit 104 cannot decode P and B-pictures until the next I-picture is supplied from the bit buffer 102”); EX1014 at 13 (“There are some disadvantages to this simple system [using only differential decoding]. First, as only differences are sent, it is impossible to begin decoding after the start of the transmission.... The solution to these problems is to use a system that is not completely differential. Figure 2-10 shows that periodically complete pictures are sent. These are called Intra-coded pictures or (I-pictures), and they are obtained by spatial compression only. If an error or channel switch occurs, it will be possible to

resume correct decoding at the next I-picture.).)



► Figure 2-10.

(EX1014 at 13, Fig. 2-10.)

245. A person of ordinary skill would have found it obvious to avoid building in this potential for decoding errors at the start of every slice—particularly in circumstances where the previous slice has been delayed or skipped/dropped in transmission (*see* Carmel at 4:34-38, 10:13-17, 12:10-12, 13:21-35 (noting that slices/files may sometimes be dropped due to excessive transmission delay))—by scanning the encoded multimedia file/stream and beginning each slice with an I-frame rather than slicing the video data at an arbitrary location. (*See, e.g.*, EX1005 at 14 (“[A]rbitrary cuts simply cannot be performed on the MPEG bitstream” without risking errors caused by a lack of a reference frame/picture); EX1021 at 255 (“In many sequences of media, some of the samples (frames) can be decoded without knowing any other samples: In video, these are known as I-frames or I-VOPs. **In general, such samples provide useful random access or synchronization points.** The indices of these samples are stored in a table, for each track, to enable random access and editing.”); EX1045 at 5:58-64 (describing scan system that identifies

where I-frames are stored in a video stream); EX1046 at 1:29-35, 1:43-2:2, 4:54-5:18, Fig. 3 (describing system for identifying and indexing I-frames within a video stream to discriminate breaks in groups of pictures to enable skipping/random access points while assuring proper image composition).)

246. Indeed, this solution to this missing reference frame issue (*i.e.*, slicing, splicing, or creating sequences of frames such that the sequence starts with an I-frame) was known in the art. (*See, e.g.*, EX1016 at 2:11-19, 2:33-50 (SMPTE standard required that the “In Point” of a splice start with a I-frame and no prediction allowed back before the In Point), 12:7-14, 14:40-45 (“Next, in step 122, the splicing procedure finds the closest I frame preceding the desired start frame to be the In Point for splicing.”), 22:53-65 (“There can also be a need for some further reformatting to permit the In Point to be an I frame of an open GOP.”); EX1019 at 229 (“The number of frames in a group was chosen so that we may restart from an I frame, if packet loss or error occur during transmission of the group, or from an I frame of the next group.... In a separate pointer file, we store headers for each audio-video group. The header makes it easy for the server to find the I frames if for some reason it needs to decimate the video stream by dropping frames.”); EX1014 at 13, Fig. 2.10 (starting sequence with an Intra frame); EX1012 at 16:18-21 (“Specifically, the first frame of the bi-level video is designated as an I-frame, and then a prescribed

number of frames following the first in sequence are designated as p-frames.”); EX1045 at 13:44-50 (disclosing group of picture structures that start with an I-frame).)

247. Further, the concept of carefully/intentionally (rather than arbitrarily) selecting logical boundary points when segmenting multimedia data for streaming was also known in other related contexts. (*Cf.* EX1021 at 260-61 (when segmenting data for a streaming application the problems caused by the packet loss in transit “may be alleviated by packetizing the media data carefully. If appropriate *boundary points* are found in the media, then the client often can recover all of the data from the packets that successfully arrived.”).) It would have been an obvious, logical, and beneficial design choice to intentionally scan and select for I-frames when determining where to slice the multimedia data stream according to the methods taught by Carmel.

5. Claim 5: “The method of claim 1, wherein multimedia objects are split from multiple MPEG4 composite layer streams by scanning time intervals and splitting the multimedia objects into video multimedia objects at next I-frames.”

248. Carmel, in view of Mattavelli and knowledge of one of ordinary skill in the art, renders obvious the method of claim 1, “wherein multimedia objects are split from multiple MPEG4 composite layer streams by scanning time intervals and splitting the multimedia objects into video multimedia objects at next I-frames.”

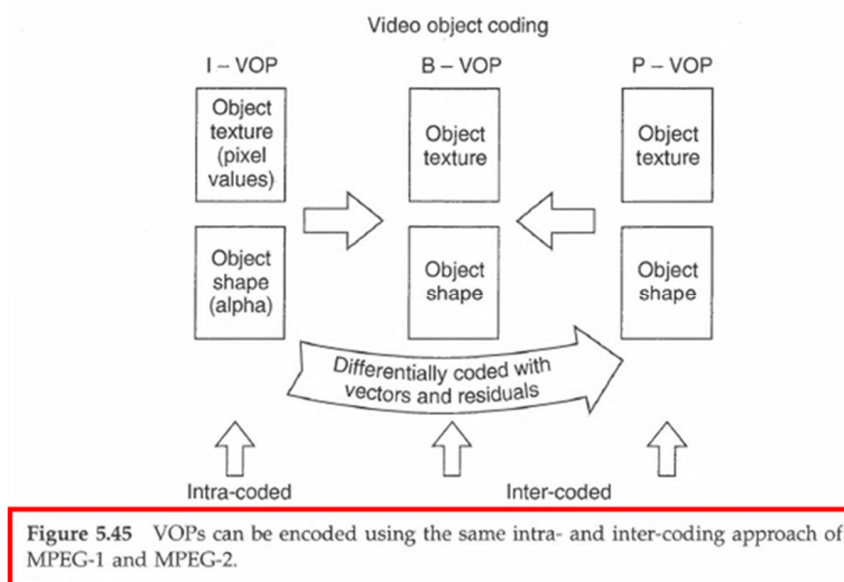
Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

249. As previously discussed for claims 1 and 3 and incorporated here, Carmel teaches that “[i]n encoding data stream 40, computer 34 preferably compresses the data using any suitable compression method known in the art” and provides the MPEG standards as examples of compression methods known in the art that may be used. (Carmel 11:23-52 (“Other compression methods known in the art, such as MPEG data compression, may similarly be used.”).) Carmel thereby teaches and renders obvious encoding data stream 40 using MPEG-4 compression standards, which were known in the art at and prior to April 2005. (*See, e.g.*, EX1005 at 22-25, 31 n.4 (identifying and introducing MPEG-4 standards).)

250. A person of ordinary skill in the art would have understood and appreciated that one of the features or characteristics of MPEG-4 video encoding is the ability to independently identify and separately encode visual objects as separate video object planes (VOPs) that can be brought together or composited by the decoder. (*See* EX1005 at 277 (“MPEG-4 objects are defined as a part of a scene which can independently be accessed or manipulated.... The pictures of conventional imaging become *object planes* in MPEG-4.”), 276-92; EX1020 at 197 (“The most significant departure from conventional transmission systems is the concept of *objects*. Different parts of the final scene can be coded and transmitted separately as video objects and audio objects to be brought together, or composited,

by the decoder.”); EX1021 at 297.)

251. Further, a person of ordinary skill in the art would have understood that the different video objects in a video (that is, sequences of video object planes or VOPs) in the known MPEG-4 standard could be and were encoded using the same sort of intra- (*i.e.*, just spatially coded based on the particular VOP’s pixel values) and inter-coding (*i.e.*, differentially coded based on predicted motion/changes) approach used for whole pictures or frames in other video compression standards. (See EX1005 at 277 (“Where an object intersects an object plane, it can be described by the coding system using intra-coding, forward prediction or bidirectional prediction.”).)



(EX1005 at 279, Fig. 5.45; EX1020 at 198 (Fig. 11.1), 199-209, 210 (“Note that each stream of encoded VOPs forms a video object layer. The base-layer VOL uses

both intra- and intercoding.”).)

252. Carmel further teaches and suggests that different video object streams encoded using MPEG-4 video coding standards are split into slices, just like any other data or data format included in an input multimedia data stream. Carmel provides:

FIG. 3D is a block diagram that schematically illustrates a file format of a multi-level data stream 41, in accordance with another preferred embodiment of the present invention. **The data stream is divided into audio slices 45, 47 and video slices 49, 51, and may also include other data formats, such as a text slice 53 and/or a URL slice 55. Each slice is preferably identified by a level identifier 57, a presentation time stamp (PTS) index 59 and, as appropriate, a size identifier 61.** The function of these identifiers and indices is described further hereinbelow. A header 43 includes data such as the title, author, copyright and **formats of the data in the stream**; the duration of the multimedia sequence represented by the stream; and a description of the available stream levels and associated data sizes.

(Carmel at 8:42-55, Fig. 3D.)

253. Based on these teachings and the knowledge of one of skill in the art, a POSA would have understood Carmel to teach, suggest, and render obvious wherein multimedia objects (*e.g.*, slices of video or video object (VOP) data) are split from multiple MPEG-4 composite layer streams (*e.g.*, multiple VOP layer streams).

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

254. And, for the reasons I discussed above for claim 4 and incorporate here, a person of ordinary skill in the art would also have been motivated and found it obvious to create these slices of separate VOP streams by scanning time intervals and splitting the multimedia objects into video multimedia objects at next I-frames (or, really, next I-VOPs).

255. Briefly, it was well-known that splitting (or splicing or editing) a video sequence that uses predictive coding to remove temporal redundancies (and achieve greater compression) could not be done arbitrarily without risking severe decoding consequences and errors. It was known that such errors would arise when predictively coded frames/VOPs (*i.e.*, P-frames/VOPs and/or B-frames/VOPs) did not have the absolute reference frame(s) (*e.g.*, I-frames/VOPs) on which the predicted and encoded differences were based. As such, it was known and obvious to those of skill in the art, to start with an intra-frame (or I-VOP) when creating, splitting, or splicing a sequence of video frames (or video object planes in the case of a set of MPEG-4 composite video layer streams). (*See, e.g.*, Mattavelli at 2309-10; EX1005 at 248; EX1016 at 2:11-19, 2:33-50, 12:7-14, 14:40-45, 22:53-65; EX1019 at 229; EX1014 at 13; *see also* EX1020 at 201 (“GOVs [group of video object planes] are similar to the groups of pictures (GOPs) in earlier MPEGs, and **provide points in the bitstream where VOPs are encoded independently from**

each other, and so provide random access points into the bitstream.”); EX1021 at 297-98, 304 (display order of group of VOPs begins with I-VOP); EX1045 at 5:58-64 (describing scan system that identifies where I-frames are stored in a video stream); EX1046 at 1:29-35, 1:43-2:2, 4:54-5:18, Fig. 3 (describing system for identifying and indexing I-pictures within a video stream to discriminate breaks in groups of pictures to enable skipping/random access points while assuring proper image composition).)

6. Claim 6 (Independent)

256. Claim 6 is, for purposes of my analysis, materially the same as claim 1 except it is written as a device (*i.e.*, a multimedia player) claim that includes computer executable instructions for performing the same method recited in claim 1. A side-by-side table comparing claim 1 with claim 6 is shown below (with differences noted in underlining). In the table below, I have also added bracketed claim 6 element identifiers (e.g., [a.i]) to match the particular element from claim 6 with the corresponding claim element and bracketed element identifier used for claim 1. Note that some of these element identifiers appear out of sequential order in claim 6 because while claim 6 recites materially the same steps as claim 1, it does so in a slightly different order.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Claim 1 (Method)	Claim 6 (Device with Computer Readable Instructions)
1. A <u>method</u> comprising:	6. A <u>multimedia player</u> comprising:
-	<u>computer executable instructions stored on a non-transitory computer readable medium which when executed by a processor:</u>
[a.i] <u>receiving</u> audio and video segments encoded in a digital encoding format and with an encoding rate;	[b.i] <u>requests</u> transmission of multimedia objects;
[a.ii] wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;	[a.i] <u>receives</u> audio and video segments encoded in a digital encoding format and with an encoding rate;
[b.i] <u>requesting by a multimedia player,</u> transmission of said multimedia objects;	[a.ii] wherein, said audio and video segments are associated object parameters and supplied host and path identification to form multimedia objects;
[b.ii] wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;	[b.ii] and wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;
[c.i] <u>playing</u> back received multimedia objects <u>by the multimedia player,</u>	[c.i] <u>plays</u> back received multimedia objects
[c.ii] <u>wherein the multimedia player is configured to play multimedia objects</u> in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said	[c.ii] in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

multimedia objects that reflect available network bandwidth,	
[c.iii] autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,	[c.iii] autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,
[c.iv] and by utilizing optimized decoding processes to maintain quality playback.	[c.iv] and by utilizing optimized decoding processes to maintain quality playback.

257. For the reasons discussed above with respect to a particular element for claim 1 and incorporated here, Carmel (and/or Carmel in view of the knowledge of a person of ordinary skill in the art and/or in view of Mattavelli, as applicable) teach and render obvious the corresponding element of claim 6 sharing the same element identifier in the chart above. By way of non-limiting example, claim element 6[a.i] in the chart above (“receives audio and video segments encoded in a digital encoding format and with an encoding rate) is taught and rendered obvious for the same reasons provided above for claim element 1[a.i] (“receiving audio and video segments encoded in a digital encoding format and with an encoding rate.”).

258. Further, as I also previously discussed for claim element 1[b.i] and incorporate here, Carmel also teaches and renders obvious a “multimedia player” as is recited in the preamble of claim 6. Briefly, Carmel teaches that the client 30 has a multimedia player on it that enables it to decode and play back the sequence of slices

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

of multimedia data taught by Carmel, “for the appreciation of the user.” (*See, e.g.*, Carmel at 10:47-50; *see also id.* at 2:15-17, 3:66-4:2, 4:3-7, 5:22-25, 10:24-11:22.) Carmel teaches that this decoder and multimedia player can be, for example, a Java applet or plug-in to the client’s web browser. (Carmel at 7:12-17, 10:27-35, 10:64-11:2.) Carmel thus discloses and renders obvious “a multimedia player.”

259. Carmel also discloses and renders obvious a multimedia player that comprises “computer executable instructions stored on a non-transitory computer readable medium which when executed by a processor.” The disclosure I just identified, including *e.g.*, Carmel’s teachings that the multimedia player can be, for example, a Java applet or plug-in to the client’s web browser, teaches and renders obvious the multimedia player on client 30 is software that includes instructions stored on a non-transitory computer readable medium that are executed by a processor on client 30. (*See* Carmel at 7:12-17, 10:27-35; *see also id.* at 2:15-17, 3:66-4:2, 4:3-7, 5:22-25, 10:24-11:22.) Indeed, Carmel even included as part of its disclosure example computer readable and executable program files for carrying out the methods of real-time multimedia streaming that Carmel teaches. (*See* Carmel at 1:3-7, 13:54-14:16, 10:34-35 (“A sample applet of this sort is incorporated herein in the software appendix, as further described hereinbelow.”).) And, as discussed above, Carmel (alone and/or in the proposed combinations) teaches and renders

obvious the computer readable instructions that the multimedia player is configured to carry out, as recited in claim 6. Carmel thus discloses and renders obvious a multimedia player that comprises “computer executable instructions stored on a non-transitory computer readable medium” that are configured to perform the subsequently recited steps of claim 6 when “executed by a processor.”

7. Claim 7 (Independent)

260. Claim 7 is, for purposes of my analysis, materially the same as claim 1 except it is written as a system for performing the same method recited in claim 1. A side-by-side table comparing claim 1 with claim 7 is shown below (with differences noted in underlining). In the table below, I have also added bracketed claim 7 element identifiers (e.g., [a.i]) to match the particular element from claim 7 with the corresponding claim element and bracketed element identifier used for claim 1. Similar to claim 6 above, I note that some of these element identifiers appear out of sequential order in claim 7 because while claim 7 recites materially the same steps/elements as claim 1, it does so in a slightly different order.

Claim 1 (Method)	Claim 7 (System)
1. A <u>method</u> comprising:	7. A <u>system</u> comprising
-	<u>at least one centralized content server</u> <u>transmitting multimedia objects;</u>

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

[a.i] receiving audio and video segments encoded in a digital encoding format and with an encoding rate;	[a.i] & [a.ii] <u>said multimedia objects including segmented audio and video, and parameters and supplied host path identification;</u>
[a.ii] wherein, said audio and video segments are associated with object parameters and supplied host path identification to form multimedia objects;	[b.i] <u>at least one</u> multimedia player <u>configured to:</u> request transmission of multimedia objects;
[b.i] requesting <u>by a multimedia player,</u> transmission of <u>said</u> multimedia objects;	[a.i] receive audio and video segments encoded in a digital encoding format and with an encoding rate;
[b.ii] wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;	[a.ii] wherein, said audio and video segments are associated object parameters and supplied host and path identification to form multimedia objects;
-	[b.ii] <u>and</u> wherein said multimedia objects are located using http and received by said multimedia player from servers using a wireless connection;
[c.i] playing back received multimedia objects <u>by the multimedia player,</u>	[c.i] play back received multimedia objects
[c.ii] <u>wherein the multimedia player is configured to play multimedia objects</u> in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,	[c.ii] in a sequence such that fluidity, video quality and audio quality are maintained by selecting a plurality of said multimedia objects that reflect available network bandwidth,
[c.iii] autonomously adjusting said selection and playback according to the	[c.iii] autonomously adjusting said selection and playback according to the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

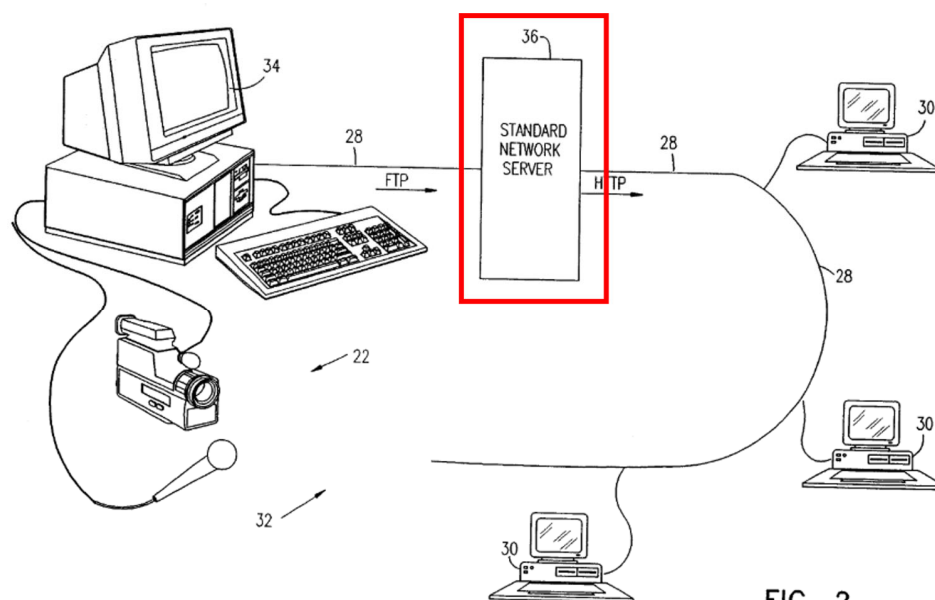
multimedia object parameters and supplied host path identification,	multimedia object parameters and supplied host path identification,
[c.iv] and by utilizing optimized decoding processes to maintain quality playback.	[c.iv] and by utilizing optimized decoding processes to maintain quality playback.

261. For the reasons discussed above with respect to a particular element for claim 1 and incorporated here, Carmel (and/or Carmel in view of the knowledge of a person of ordinary skill in the art and/or in view of Mattavelli, as applicable) teach and render obvious the corresponding element of claim 7 sharing the same element identifier in the chart above. By way of non-limiting example, claim element 7[a.i] in the chart above (“receives audio and video segments encoded in a digital encoding format and with an encoding rate”) is taught and rendered obvious for the same reasons provided above for claim element 1[a.i] (“receiving audio and video segments encoded in a digital encoding format and with an encoding rate.”)

262. Further, Carmel also teaches and renders obvious a “system”—namely, a system for real-time broadcasting of a multimedia sequence over a computer network. (*See, e.g.*, Carmel at Abstract, 6:24-49 (“Reference is now made to FIG. 2, which is a schematic illustration of a computer **system 32** for remote broadcasting of a multimedia sequence over a network 28, in accordance with a preferred embodiment of the present invention. **System 32** comprises a transmitting computer 34, which generates the sequence, a plurality of clients 30, and a network server 36,

all of which communicate over network 28, preferably using the well-known Internet Protocol (IP).”), Fig. 2.)

263. Carmel also teaches a system that includes “at least one centralized content server transmitting multimedia objects.” For example, Carmel teaches a system that includes **network server 36** that communicates over network 28 to transmit multimedia objects to clients 30. (*See Carmel at Abstract, 6:24-49* (“Reference is now made to FIG. 2, which is a schematic illustration of a computer system 32 for remote broadcasting of a multimedia sequence over a network 28, in accordance with a preferred embodiment of the present invention. System 32 comprises a transmitting computer 34, which generates the sequence, a plurality of clients 30, and a **network server 36, all of which communicate over network 28,** preferably using the well-known Internet Protocol (IP).”), 7:4-5 (“Clients 30 connect to **server 36 and receive the multimedia sequence,** substantially in real time.”), Fig. 2.)



(Carmel at Fig. 2.)

264. Carmel (alone and/or in the proposed combinations) also teaches and renders obvious the remaining claim elements of claim 7 for the reasons I previously discussed for their corresponding elements shown in the chart above for claim 1. I note that for the claim 7 element that recites “said multimedia objects including segmented audio and video, and parameters and supplied host path identification” I discuss (and incorporate here) for claim element 1[a.i] how Carmel teaches and renders obvious “multimedia objects including segmented audio and video” and for claim element 1[a.ii] how Carmel teaches and renders obvious multimedia objects including “parameters and supplied host path identification.” I also discuss (and incorporate here) in claim element 1[b.i] how Carmel teaches and renders obvious

“at least one multimedia player” that is configured to (among other things) “request transmission of multimedia objects.”

8. **Claim 8: “The method of claim 1, wherein the multimedia objects are created from existing multimedia files that have been transcoded into optimal digital formats at an optimal encoding rates reflecting available network bandwidth, and converting them into audio and video segments by splitting the encoded files into intervals for access by wireless clients.”**

265. Carmel discloses and renders obvious the method of claim 1, “wherein the multimedia objects are created from existing multimedia files that have been transcoded into optimal digital formats at an optimal encoding rates reflecting available network bandwidth, and converting them into audio and video segments by splitting the encoded files into intervals for access by wireless clients.”

266. Carmel teaches the use of “existing multimedia files” as an input multimedia stream that is encoded/transcoded and sliced by transmitting computer 34 to create multimedia objects. For example, Carmel provides: “[t]he sequence is preferably generated and compressed in real time, and could comprise, for example, an interview program or an entertainment or sports event, **although a prerecorded sequence may similarly be broadcast in this manner.**” (Carmel at 6:50-60.) Carmel also teaches, for example: “Broadcast data are then input to the computer, for example, from input devices 22, **or from a video, audio or animation sequence stored on disk or tape**” before the data is compressed and sliced in subsequent steps.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(Carmel at 9:62-10:1, Fig. 5.) Carmel thus contrasts live multimedia data from, e.g., input devices 22, with “existing multimedia files” that were pre-recorded and stored on, e.g., disk or tape and satisfies this claim element.

267. A person of ordinary skill in the art would have understood found it obvious that a pre-recorded multimedia sequence would typically be “stored on disk or tape” in a compressed form, e.g., to save on storage space/costs. (*See, e.g.*, EX1012 at 8:16-41 (noting that the video capture process “could be either real-time or offline” but regardless of which video capture process was used, “[t]ypically, the input video will have been compressed using a conventional video compression process”); EX1005 at 4-6 (“many systems use compression to offset the high cost of disk storage”); EX1049 at 2:6-50.) As discussed for claim 3 (and incorporated here), a person of ordinary skill in the art would have understood and found obvious that the encoding such input data into a different encoding format (e.g., MPEG-2 encoding to MPEG-4 encoding) and/or rate (e.g., MPEG-encoding at a lower bitrate that matches the available bandwidth for transmission to client 30, as taught by Carmel) to constitute “transcoding” such data.

268. Outside of the “existing multimedia files” limitation, the rest of claim 8 is materially the same as claim 3 of the ’849 patent. (*Compare* ’849 patent at claim 3 (“The method of claim 1, wherein the multimedia objects are created from a live

multimedia stream that is transcoded into an optimal digital format at an optimal encoding rate reflecting available network bandwidth and converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients.”), *with* claim 8 (“The method of claim 1, wherein the multimedia objects are created from existing multimedia files that have been transcoded into optimal digital formats at an optimal encoding rates reflecting available network bandwidth, and converting them into audio and video segments by splitting the encoded files into intervals for access by wireless clients.”) (differences have been underlined).)

269. As I explained in claim 3 and incorporate here, Carmel also teaches and renders obvious the remaining, materially identical elements of claim 8.

9. **Claim 9: “The method according to claim 1, wherein the object parameters and supplied identification of the first player received multimedia object define a sequence of similar multimedia objects and configure the player accordingly, using object parameters and video decoding optimization, such that fluidity, video and audio quality are maintained.”**

270. Carmel discloses and renders obvious the method according to claim 1 “wherein the object parameters and supplied identification of the first player received multimedia object define a sequence of similar multimedia objects and configure the player accordingly, using object parameters and video decoding

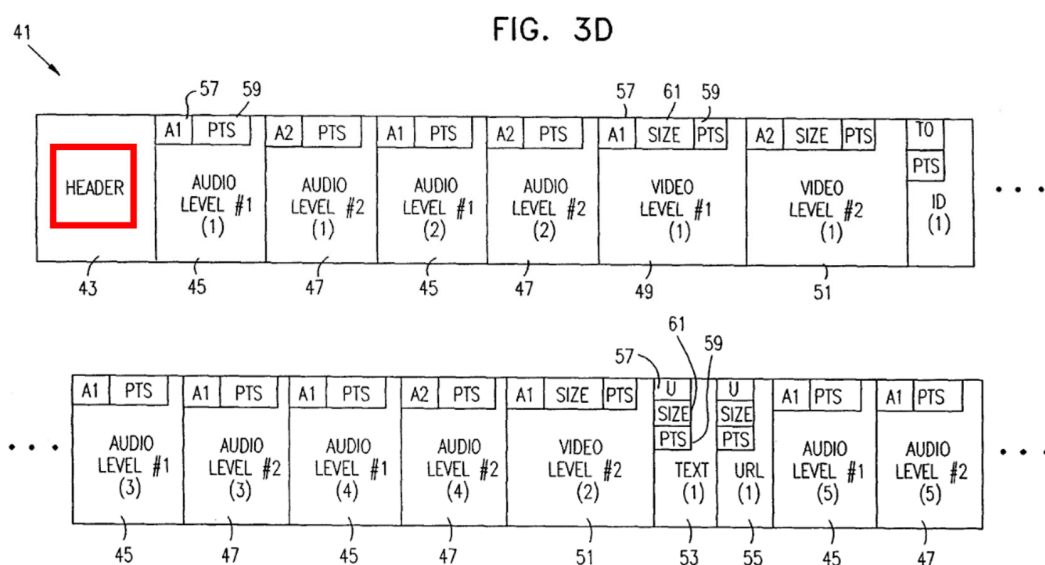
optimization, such that fluidity, video and audio quality are maintained.”

271. As discussed above in claim 1 (including, in particular, claim elements 1[a.i], 1[a.ii], 1[c.ii], 1[c.iii], and 1[c.iv]) and which I incorporate here, Carmel (alone and in the proposed combinations, including in view of the teachings of Mattavelli) teaches and renders obvious, among other things: (i) multimedia objects having object parameters and supplied host path identification (*e.g.*, slice indices/identifiers, timestamps, data size identifiers, host path identifiers, decoding complexity information, and other metadata) and (ii) utilizing object parameters and decoding optimizations to configure the multimedia player to maintain playback fluidity and the highest possible or practicable video and audio quality in view of the network and system (including client) resources and multimedia stream characteristics.

272. Carmel further teaches and renders obvious providing object parameters and metadata with the first object (*e.g.*, an index file) that the client computer (and multimedia player thereon) reads when it begins to download the data stream. (*See* Carmel at 8:1-7 (“When one of computers 30 connects to server 36 and begins to download the data stream, **it first reads the index file** in order to identify at what point in stream 40 to begin and to start receiving the data stream substantially in real time, preferably with only a minimal lag, as it is transmitted from computer

34.”.) Carmel further teaches and suggests that this first file/message that “is read by computers 30 when they to connect to server 36 to download data stream 40” **“contains parameters relating generally to the data stream and/or instructions to computers 30.”** (See Carmel at 8:12-18, Fig. 3B.)

273. Carmel also teaches and suggests an embodiment where a multimedia object includes, in addition to one or more slices of video or audio data, a header containing parameters, metadata, and other configuration information relating to the multimedia stream. For example, Carmel provides: **“A header 43 includes data such as the title, author, copyright and formats of the data in the stream; the duration of the multimedia sequence represented by the stream; and a description of the available stream levels and associated data sizes.”** (Carmel at 8:51-55.)



(Carmel at Fig. 3D.)

274. Carmel thus teaches and renders obvious having the first received multimedia object received by the multimedia player on the client include (*e.g.*, in the header of the first multimedia object) information about and define a sequence of similar multimedia objects (*e.g.*, “parameters relating generally to the data stream and/or instructions to computers 30” and/or metadata regarding “formats of the data in the stream; the duration of the multimedia sequence represented by the stream; and a description of the available stream levels and associated data sizes”) to configure the player on client 30 so that play back fluidity and audio and video quality are maintained (as discussed in claim 1 above).

275. In view of the similarity of their description and purpose in Carmel, as well as the knowledge in the art, a person of ordinary skill in the art would have understood and found it obvious that the configuration parameters and instructions contained in an index file (like that disclosed in Carmel) could additionally or alternatively be included in the header of a multimedia object (including the first multimedia object), like that pictured in Carmel Figure 3D. (*See also* Carmel at 8:9-11 (“Further alternatively, stream 40 may be multicast to clients 30, **as is known in the art, typically without the use of an index file.**”).) This would be nothing more than the combination of prior art methods according to known methods or a simple

substitution of one known element (*e.g.*, use of an index file with parameters and configuration information) for another known element (*e.g.*, use of a header to a multimedia file with parameters and configuration information) to obtain predictable results (*e.g.*, configuring the multimedia player for playback of the sequence of multimedia objects in a manner that maintains playback fluidity and video and audio quality.) (*See also, e.g.*, EX1024 at 2:16-3:4 (describing including parameters relating to the decoding complexity requirements in the headers of audiovisual packets); EX1025 at [0017] (noting that different metadata can refer to the entire file (“file-level meta data”) or to specific media samples “sample-specific meta-data”).) Further, it would have been logical and obvious to include any necessary or desired configuration metadata or instructions to computers 30 relating generally to the data stream (which is comprised of a sequence of similar multimedia objects) with the first object read by the multimedia player on client 30, as configuration is a first/initial step in a process needed for playback. (*See, e.g.*, EX1025 at [0018] (“As media decoding and playback are typically not possible without file-level meta-data, such meta-data appears at the beginning of streaming files as a file header section.”).)

C. Ground 2: Claims 1-9 Are Obvious Over Carmel in View of Wang

276. Ground 2 incorporates Ground 1’s analysis of Carmel’s teachings and adds Wang’s teachings for claim elements 1[a.i], 1[b.ii], 1[c.iv] (and the

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

corresponding elements recited in independent claims 6 and 7, as identified in the charts for claims 6 and 7 above in Ground 1) and dependent claims 3, 4, 5, 8 and 9. Accordingly, only claim elements 1[a.i], 1[b.ii], and 1[c.iv] (and the corresponding elements recited in independent claims 6 and 7) and dependent claims 3, 4, 5, 8, and 9 are addressed in the discussion below. In light of the analysis below of these limitations and Ground 1's analysis of the remaining claim elements and claims, claims 1-9 are obvious based on Carmel in view of Wang.

277. Claim element 1[a.i] recites (in part) encoding audio and video segments “in a digital encoding format.” As discussed in Ground 1, Carmel discloses compressing/encoding the multimedia data stream 40 “using any suitable compression method known in the art” and identifies MPEG compression standards as examples of suitable coding formats. (*See, e.g.*, Carmel at 11:9-39, Figs. 5, 7.) Wang teaches a decoder (*i.e.*, multimedia player) for decoding multimedia data encoded with an MPEG-4 Advanced Video Coding (AVC)¹¹ codec (*see* Wang (EX1007) at 51), which a person of ordinary skill in the art would have recognized as a suitable compression method known in the art. Indeed, as discussed below, the MPEG-4 AVC coding/compression format would have been viewed as particularly

¹¹ The MPEG-4 Advanced Video Coding (AVC) video coding standard is also commonly referred to as H.264.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

well-suited (and an example of “an optimal digital format”) for multimedia streaming implementations involving wireless transmission to a mobile client (as taught and rendered obvious by both Carmel and Wang) due to its high compression efficiency. (See Wang at 51 (“Multimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile.”); EX1027 at 657 (“[C]ompression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC [MPEG-4 Part 10] coding an attractive candidate for all wireless applications including multimedia messaging services (MMS), packet-switched streaming services (PSS), and conversational applications.”).) Accordingly, Carmel in view of Wang renders obvious encoding audio and video segments “in a digital encoding format”—including, in particular, encoding video data using the MPEG-4 AVC format.

278. Claim element 1[b.ii] recites (in part) that the multimedia objects are “received by said multimedia player from servers using a wireless connection.” As discussed in Ground 1, Carmel teaches and renders obvious the use of a wireless connection by a client 30 to receive the multimedia objects from server 36. (See,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

e.g., Carmel at 2:41-46, 6:65-7:3, 6:36-38, Fig. 2; EX1009 (Carmel '432) at 1:29-36, 3:21-35, 3:43-47, 3:48-53, Fig. 1; EX1010 (Carmel '230) at 11:50-57.) Wang also teaches and suggests a multimedia player (*i.e.*, a decoder) receiving compressed multimedia data (encoded with a MPEG-4 AVC codec) via a wireless connection. (*See* Wang at 51 (“Multimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile.”), 52 (emulating the proposed improved AVC decoder architecture on an ARM development board because “the ARM platform is one of the most typical portable devices and takes almost 75% of the market for the mobile devices”).) Accordingly, Carmel in view of Wang also renders obvious having a multimedia player receive multimedia objects from a server “using a wireless connection.”

279. Next, claim element 1[c.iv] recites: “utilizing optimized decoding processes to maintain quality playback.” Wang teaches utilizing optimized decoding processes to maintain quality playback. Indeed, this is a primary focus of Wang.

280. Wang teaches a “platform-based MPEG-4 Advanced Video Coding (AVC) decoder with block level pipelining” that optimizes the decoding hardware

and software so that resource-constrained mobile devices can decode MPEG-4 AVC encoded data fast enough to utilize this compression standard and benefit from the higher image quality it can provide. Wang teaches:

Multimedia services over mobile network are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, **MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile.** Since some of the AVC modules consume considerable power, it is challenging to realize real-time AVC encoder and decoder. Several AVC codec [8][9] are optimized for general purpose processors that have sufficient computation power, large memory, special instruction sets, and wide buses to facilitate the speedup of AVC modules. **But the performance of the optimized AVC modules is seriously constrained by the low computation power and small memory for mobile devices to achieve the least service requirement of QCIF 7.5 fps. To address such constraints with limited resource, we adopt platform based design method to increase overall throughput.**

(Wang at 51.)

281. Wang repeatedly states that its proposed decoder has “optimized” the decoding processes and software it uses in order to achieve this goal of increased throughput/decoding speed of the AVC codec on devices with limited processing resources. For example, Wang provides the following summary in its Abstract:

We present a baseline MPEG-4 AVC (Advanced Video Coding) decoder based on an optimized platform-based design methodology. With this methodology, we jointly **optimize** the software and hardware design of the decoder. Overall decoding throughput is increased by synchronizing the software and the dedicated co-processors. The synchronization is achieved at macroblock-level pipelining. In addition, we **optimize the decoder software** by enhancing the frame buffer management, boundary padding, and content aware inverse transform. To speed up motion compensation and inverse transform, which are the most computationally intensive modules, two dedicated acceleration modules are realized. For comparison, the proposed prototype decoder and MPEG-4 AVC reference decoder are evaluated on an ARM platform, which is one of most popular portable devices. **Our experiments show that the throughput of the MPEG-4 reference decoder can be improved by 6 to 7 times.** On an ARM966 board, **the optimized software** without hardware acceleration can achieve a decoding rate up to 5 frames per second (fps) for QCIF video sequences. With the dedicated accelerators, the overall throughput is increased by about 30% to reach 6.6 fps on the average and is up to 10.3 fps for slow motion video sequences.

(Wang at 51; *see also id.* at 51, 52 (“In this paper, we adopt a platform based design methodology to construct **an optimized AVC decoder** with a novel scheduling to achieve MB-level pipelining.”), 52 (“[T]ask scheduling of the decoding processes is

optimized by a MB engine.”), 53, 54 (“**Software Optimization**” section 4.3 describing some of the optimized decoding processes used, including frame buffer management, boundary padding for motion compensation and content aware IQ-IDCT).) Accordingly, Carmel in view of Wang teaches and renders obvious utilizing optimized decoding processes to provide and maintain quality playback.

282. Based on these same teachings and for the same reasons, Carmel in view of Wang renders obvious using “video decoding optimization, such that fluidity, video and audio quality are maintained” as is recited in claim 9.

283. Further, to the extent that “optimized decoding processes” is interpreted to mean that both optimized video decoding processes (like those taught by Wang) and optimized audio decoding processes must be employed, Wang’s teachings, in view of the knowledge of a person of ordinary skill in the art, further suggest and render obvious configuring a multimedia player to utilize optimized audio decoding processes as well as optimized video decoding processes. This is so because “audio compression is built on many of the same principles as video compression” (EX1020 at 322) and a person of ordinary skill in the art would have appreciated and found obvious that using a combination of optimized video and audio decoding algorithms would beneficially improve the speed and efficiency of the multimedia decoder, which is particularly helpful for mobile clients typically having more processor and

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

power constraints. Indeed, as discussed above, there were numerous optimized audio decoding processes known in the art, and a person of ordinary skill in the art would have found it advantageous and obvious to employ such optimized decoding techniques for any of the “suitable compression method” (Carmel at 11:27-28) that were used, particularly in applications involving mobile clients (as disclosed and rendered obvious by Carmel and Wang). (*See, e.g.*, EX1039 at 1119, 1122-25 (teaching optimizations to the synthesis filter and providing a parameter analyzer block to take advantage of a multichannel computing structure for a MPEG-2 audio decoder); EX1040 at 671 (teaching “optimization techniques to implement MPEG audio decoding algorithms [MPEG-1/2 Layer III (MP3) and MPEG-2 AAC (Advanced Audio Coding)] in real-time using [a] general purpose 32-bit MCU RISC processor” that “would be beneficial to the overall power consumption of a portable MPEG player.”), 672-75 (describing optimization techniques for control-intensive and computationally-intensive aspects of audio decoding algorithms, including optimizations for reducing the computational loads of memory access and multiplications); EX1042 at 376-78, 379 (describing audio decoding optimization techniques that avoid computing non-useful “zero-filled” data and noting that particular techniques (*i.e.*, “cut-off optimizations”) are scalable and can be adjusted to gracefully degrade to meet any real-time decoding deadlines, making these

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

algorithms particularly useful in portable devices); EX1043 at 303-06 (teaching several optimized processes for decoding MPEG audio); EX1047 at 117-18, 120-23 (describing “optimization of the Huffman decoding strategy for AAC decoders” and providing a “new table lookup approach which significantly reduces the computational requirements”).)

284. In particular, for mobile streaming applications utilizing the MPEG-4 AVC video standard for video compression (as taught and rendered obvious by Carmel and Wang), a person of ordinary skill in the art would have found it advantageous and obvious to use High-Efficiency MPEG Advanced Audio Coding (HE AAC) which was an audio compression standard that “extend[ed] the reach of high-quality MPEG-4 Audio to much lower bitrates” enabling higher quality compressed audio with less data. (EX1041 at 1.) “Specifically in mobile devices, computational complexity and memory requirements are directly related to power consumption and implementation cost. HE AAC was specifically designed to minimize these resources,” particularly in certain profiles. (EX1041 at 12; *see also id.* at 11 (Example of “Basic configuration for cell phones: Since the listening environment is not very demanding but computational resources might be extremely limited, a down-sampled, low power HE AAC decoder supporting level 2 of the HE AAC profile might be suitable.”), 13 (“Typical scenarios for the Low power version

are mobile devices where complexity and thus power consumption is critical while the listening environment is usually not as demanding as in other applications.”.)
Indeed, there was also express suggestion in the art to combine use of the MPEG-4 AVC standard with use of the HE AAC standard in circumstances where quality at very low bitrates is needed or desired, like when transmission bandwidth is constrained in streaming over a wireless connection. (EX1041 at 13 (“By coupling HE AAC with MPEG-4 Video, more bits can be allocated to the video signal without degrading the quality of the audio signal.... Combined with the new MPEG/ITU Advanced Video Coding (AVC) standard (included in MPEG-4 as part 10), even more significant gains in quality are possible.”).)

285. And, prior to April 2005, “optimized source code” was available for license and “[o]ptimized binary implementations” of audio decoders of the HE AAC standard—that is, decoders that utilized optimized audio decoding processes—were available and known in the art. (EX1041 at 12.)

286. Next, Claims 3 and 8 each recite (in part) transcoding a multimedia stream “into an optimal digital format” and splitting the stream into multimedia objects “for access by wireless clients.” As discussed above for Ground 1, Carmel discloses and renders obvious these elements (as well as each of the remaining elements of claims 3 and 8). Wang also teaches and suggests these elements. As just

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

discussed for claim elements 1[a.i] and 1[b.ii], Wang teaches that MPEG-4 AVC is a particularly suitable compression format for mobile streaming applications on mobile/wireless devices, as it provides high compression efficiency, allowing high image quality while using lower bandwidth/bitratesize. (Wang at 51, 52; *see also* EX1027 at 657.) And, also as discussed above, Wang teaches an optimized AVC decoder that is able to decode AVC-encoded data with greatly improved throughput on resource-constrained mobile devices. (Wang at 51-55.) Accordingly, Carmel in view of Wang also render obvious transcoding a multimedia stream “into an optimal digital format” and splitting the stream into multimedia objects “for access by wireless clients” as recited in claims 3 and 8.

287. Next, claims 4 and 5 each recite (in part) splitting a multimedia segment/object (which claim 5 recites is itself split from multiple MPEG4 composite layer streams) at an I-frame. As explained above in Ground 1 (and incorporated here), Carmel in view of the knowledge of one of ordinary skill in the art (and in view of Mattavelli and the knowledge of one of ordinary skill in the art) would have appreciated the benefits of and found it obvious to split video streams or video object streams at an I-frame rather than in an arbitrary or random location. Wang’s teachings regarding the use of a MPEG-4 AVC codec (*see, e.g.*, Wang at 51) further suggest and render obvious this claim element.

288. A person of ordinary skill in the art would have understood that the MPEG-4 AVC standard coded (and codes) pictures/objects as one or more “slices” that can be I (Intra)-slices that contain only intra-coded macroblocks or predicted slices (*e.g.*, P- or B-slices) which contain macroblocks predicted from other reference pictures. (*See* EX1028 at 164.) The MPEG-4 AVC standard’s Extended Profile also introduced an “SI” slice type, which contains a special type of intra-coded macroblocks that facilitates switching between streams. (EX1028 at 164, 216.) The Extended Profile is recognized as “particularly useful for applications such as video streaming” and included “additional features to support efficient streaming over networks such as the Internet.” (EX1028 at 216.)

289. A person of ordinary skill in the art would have appreciated and found obvious that the multi-level data streaming embodiment taught by Carmel (*see* Carmel at 10:64-11:22, Figs. 3D, 6B) would have been advantageously implemented using SI slices provided under the Extended Profile of the MPEG-4 AVC standard. For example, EX1028 (Richardson) provides:

SP and SI slices are specially-coded slices that enable (among other things) efficient switching between video streams and efficient random access for video decoders [10]. A common requirement in a streaming application is for a video decoder to switch between one of several encoded streams. For example, the same video

material is coded at multiple bitrates for transmission across the Internet and a decoder attempts to decode the highest-bitrate stream it can receive but may require switching automatically to a lower-bitrate stream if the data throughput drops.

(EX1028 at 216.)

290. Further, it would have been obvious to a person of ordinary skill in the art that when using the MPEG-4 AVC encoding (as taught by Wang) the first video picture in a multimedia object created according to Carmel's teachings would and should be an I-frame—and, more specifically, it would and should be an Instantaneous Decoder Refresh (IDR) coded picture (consisting of I- or SI- slices). EX1028 (Richardson) describes this feature of the MPEG-4 AVC standard:

Instantaneous Decoder Refresh Picture

An encoder sends an IDR (Instantaneous Decoder Refresh) coded picture (made up of I- or SI-slices) to clear the contents of the reference picture buffer. On receiving an IDR coded picture, the decoder marks all pictures in the reference buffer as 'unused for reference'. All subsequent transmitted slices can be decoded without reference to any frame decoded prior to the IDR picture. The first picture in a coded video sequence is always an IDR picture.

(EX1028 at 167.) In other words, a person of ordinary skill in the art would have appreciated and found it obvious to have each multimedia video object (as taught by

Carmel)—which is a coded video sequence—to start with an IDR picture (*i.e.*, an I-frame made of I- or SI-slices) when using MPEG-4 AVC encoding (as taught and/or suggested by Wang and Carmel).

291. Accordingly, Carmel in view of Wang also renders obvious splitting a multimedia stream such that each multimedia object starts with an I-frame, as recited in claims 4 and 5.

(1) Manner and rationales for combining the teachings of Carmel and Wang

292. A person of ordinary skill in the art would have found it obvious and would have been motivated to combine the teachings of Carmel and Wang, predictably resulting in the multimedia transmission and playback system taught by Carmel which uses Wang’s MPEG-4 Advanced Video Coding (AVC) decoder to play back high-quality MPEG-4 AVC encoded multimedia data. A person of ordinary skill in the art would have understood that using Carmel’s method of streaming video and audio data, which has been encoded and compressed “using any suitable compression method known in the art” (Carmel at 11:27-28), which includes, e.g., MPEG-4 AVC, to a mobile recipient client device that uses Wang’s optimized decoding software and hardware design to decode AVC-encoded data would have been successful and an improvement over alternative AVC decoders running on the same types of processors. (*See, e.g.*, Wang at 51 (“For comparison,

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

the proposed prototype decoder and MPEG-4 AVC reference decoder are evaluated on an ARM platform, which is one of most popular portable devices. Our experiments show that the throughput of the MPEG-4 reference decoder can be improved by 6 to 7 times.”.)

293. Carmel and Wang are both directed towards the processing of multimedia data ultimately for playback “for the appreciation of a user.” (*See* Carmel at 10:49-50.) Both references teach methods and techniques for attaining the shared goal of providing the highest practicable streaming quality audio and video in view of device and/or system constraints. (*See* Carmel at 10:48-54 (“In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user. Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.”), 11:9-18 (“Periodically, client 30 makes an assessment of the rate of data transfer over the link from the server and, if necessary, changes the quality level accordingly. For example, if the rate is low, such that time stamps 59 indicate that the slices need to be played as fast as or faster than they are being received, the client will preferably select a lower quality level if one is available. On the other hand, if the rate is substantially higher than what is needed to receive the successive slices on time, the

client may select a higher quality level to take advantage of the available bandwidth.”), 7:35-49, Figs. 3D, 6B; Wang at 51 (“Multimedia services over mobile network are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile. ... But the performance of the optimized AVC modules is seriously constrained by the low computation power and small memory for mobile devices to achieve the least service requirement of QCIF 7.5 fps. To address such constraints with limited resource, we adopt platform based design method to increase overall throughput.”), 52.)

294. Both references also refer to and acknowledge the encoding and compressing of multimedia data (which requires subsequent decoding) according to well-known coding standards/formats, including, in particular, use of the MPEG-4 AVC standard/format taught by Wang. (*See, e.g.*, Carmel at 7:25-26, 11:24-52; Wang at 51, 52-55.) Both references teach techniques that involve the encoding and/or decoding of multimedia data as part of a process for streaming and playing multimedia content to an end user. And, as discussed above, the MPEG-4 AVC format/standard would have been viewed as particularly well-suited for multimedia streaming implementations involving wireless transmission to a mobile client (as

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

taught and rendered obvious by both Carmel and Wang) due to its high compression efficiency. (See Wang at 51 (“Multimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices. For this kind of application, MPEG-4 Advanced Video Coding [7] provides better image quality due to its significant improvement on the coding efficiency as compared to that of MPEG-4 visual simple profile.”); EX1027 at 657 (“[C]ompression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC [MPEG-4 Part 10] coding an attractive candidate for all wireless applications including multimedia messaging services (MMS), packet-switched streaming services (PSS), and conversational applications.”); EX1028 at 164, 216 (“The Extended Profile [of H.264] may be particularly useful for applications such as video streaming” as it includes “additional features to support efficient streaming over networks such as the Internet.”).) A person of ordinary skill would have been motivated to use the MPEG-4 AVC format/standard taught by Wang (along with Wang’s optimized AVC decoder) in the systems and methods taught by Carmel, in particular for wireless/mobile clients, because of its high compression efficiency which permits higher image quality with less data transfer. (See *id.*) A POSA would have been motivated to utilize Wang’s optimized decoding processes for decoding AVC-

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

encoded multimedia to facilitate or enable such benefits from the use of AVC encoding on processing-power constrained mobile clients. (*See* Wang at 51.)

295. In addition to being directed to similar subject matter (and being within the same field of endeavor as the '849 patent), a person of ordinary skill in the art would have also appreciated that Carmel and Wang are complementary. Carmel and Wang are focused on mitigating different bottlenecks or constraints in the chain of transmission and processing of streamed multimedia data from the data source to the ultimate user/consumer/player of the video and audio data.

296. As discussed above, Carmel is primarily focused on constraints imposed by the available network bandwidth. It teaches methods for mitigating the effects of these bandwidth constraints on playback continuity and quality by providing slices of multimedia data (including slices of video and audio data that have been encoded at multiple quality levels) for download by a client multimedia player where the bitrate/data rate of the slices generally corresponds to the network bandwidth available for download. (*See, e.g.*, Carmel at Abstract, 9:6-9 (“Each of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36. A method for selecting and, as required, varying the level is described hereinbelow with reference to FIG. 6B), 9:13-17, 10:64-11:22, 11:40-12:17, Figs. 3D, 6B.) Carmel teaches that the slice download rate can thereby

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

“keep[] up” with the multimedia slice output/play rate for a given available network bandwidth. (Carmel at 7:36-41.)

297. Wang is focused on a different constraint in the playback of streamed, encoded multimedia data—namely, limitations on the processing resources available to the client/multimedia player (particularly on mobile devices) when decoding a compressed bitstream. Also as discussed above, Wang teaches that “the performance of the optimized AVC modules is seriously constrained by the low computation power and small memory for mobile devices.” (Wang at 51; *see also* EX1027 at 658 (“Mobile devices are hand-held and constrained in processing power and storage capacity.”).) Wang discloses an AVC-decoder that uses optimized decoding processes to allow such processing-resource constrained mobile devices to better/faster decode AVC-encoded multimedia data. (*See* Wang at 51 (“Our experiments show that the throughput of the MPEG-4 reference decoder can be improved by 6 to 7 times.”), 51 (“To address such constraints with limited resource[s], we adopt platform based design method to increase overall throughput.”), 52-55 (describing decoder hardware and software optimizations).)

298. A person of ordinary skill in the art would have been motivated to combine the complementary methods and techniques taught by Carmel and Wang to form a streaming system that is more robust and better able to handle multiple

potential bottlenecks or constraints in the progress of a multimedia bitstream from its source to the end user across a network in a way that provides the best possible playback experience. A person of ordinary skill in the art would have recognized that a multimedia streaming system would potentially be subject to constraints on (i) available network bandwidth, (ii) multimedia player processing resources, or (iii) both at the same time, and would have been motivated to use MPEG-4 AVC encoding for its compression efficiency (allowing one to transmit higher quality video using less bandwidth) and apply Wang's teachings regarding an optimized AVC decoder to the multimedia player/decoder on a (processing power-constrained) mobile client 30 taught and rendered obvious by Carmel and Wang to allow such a decoder to decode this AVC-encoded data.

299. Carmel contains express disclosure that would have motivated a person of ordinary skill in the art to combine its teachings with those of Wang. Namely, Carmel provides: "Time stamps in the data stream are used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer 34, preferably with only a minimal necessary transmission **and decoding delay**." (Carmel at 10:50-54; *see also id.* at 3:66-4:2, 5:22-25, 11:7-8.) Carmel thus expressly acknowledges that there are (at least) two potential sources of delay in playing back a multimedia sequence: (i) transmission delay and (ii) decoding delay. As discussed

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

above, Carmel primarily teaches techniques for minimizing transmission delays. A person of ordinary skill in the art would have been motivated to use Wang's optimized AVC decoding techniques for mobile clients with the multimedia player taught by Carmel in order to minimize any potential decoding delays or problems in wireless client implementations.

300. Additionally, a person of ordinary skill in the art would have considered a system that incorporated the bandwidth optimization techniques taught by Carmel with high compression efficiency data encoding (using MPEG-4 AVC) and the decoder optimization techniques taught by Wang to be an obvious combination of known methods to address known issues (or known set of issues, *e.g.*, network bandwidth and decoder processing constraints) to yield predictable results—*i.e.*, a system where a mobile client multimedia player/decoder advantageously optimizes both the available network bandwidth and the available decoding resources to facilitate use of the highly efficient AVC encoding standard when playing/outputting a multimedia data stream “for the appreciation of a user.” (Carmel at 10:48-49.) Similarly, a person of ordinary skill in the art would have found the combined teachings of Carmel and Wang to be an obvious application of a known technique (*e.g.*, the optimized AVC decoding techniques taught by Wang) to a known device, method, or product ready for improvement (*e.g.*, the streaming method and system

taught by Carmel) to yield predictable results (*e.g.*, a method and system that utilizes the highly efficient AVC encoding standard and optimizes both the available network bandwidth and the available decoding resources when playing/outputting an AVC-encoded multimedia data stream on a mobile client for the appreciation of a user).

D. Ground 3: Claims 1-9 Are Obvious Over Carmel in View of Wang and Wolters

301. Ground 3 incorporates Ground 1’s analysis of Carmel’s teachings and Ground 2’s analysis of Wang’s teachings. Ground 3 adds Wolters’s teachings for claim element 1[c.iv], reciting “optimized decoding processes.”

302. As discussed above in Ground 2, for mobile streaming applications utilizing the MPEG-4 AVC video standard for video compression (as taught and rendered obvious by Carmel and Wang), a person of ordinary skill in the art would have found it advantageous and obvious to use High-Efficiency MPEG Advanced Audio Coding (HE AAC), which Wolters teaches was/is an audio compression standard that “extend[ed] the reach of high-quality MPEG-4 Audio to much lower bitrates” enabling higher quality compressed audio with less data. (EX1041 at 1.) Wolters provides: “[s]pecifically in mobile devices, computational complexity and memory requirements are directly related to power consumption and implementation cost. HE AAC was specifically designed to minimize these resources,” particularly

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

in certain profiles. (EX1041 at 12; *see also id.* (Example of “Basic configuration for cell phones: Since the listening environment is not very demanding but computational resources might be extremely limited, a down-sampled, low-power HE AAC decoder supporting level 2 of the HE AAC profile might be suitable.”), 13 (“Typical scenarios for the Low power version are mobile devices where complexity and thus power consumption is critical while the listening environment is usually not as demanding as in other applications.”).)

303. Wolters also teaches that “optimized source code” was available for license and “[o]ptimized binary implementations” of audio decoders of the HE AAC standard—that is, decoders that utilized optimized audio decoding processes—were available and known in the art. (EX1041 at 12.) Thus, to the extent that “optimized decoding processes” is interpreted to require use of both optimized video and optimized audio decoding processes, this element is rendered obvious by the combined teachings of Carmel, Wang, and Wolters.

304. A person of ordinary skill in the would have been motivated to combine Wolter’s teachings regarding use of HE AAC (and optimized processes for decoding HE AAC compressed audio data) with those of Carmel and Wang. As discussed above, Carmel and Wang teach and render obvious multimedia streaming to wireless devices and further teach and render obvious the use of MPEG-4 AVC video coding

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

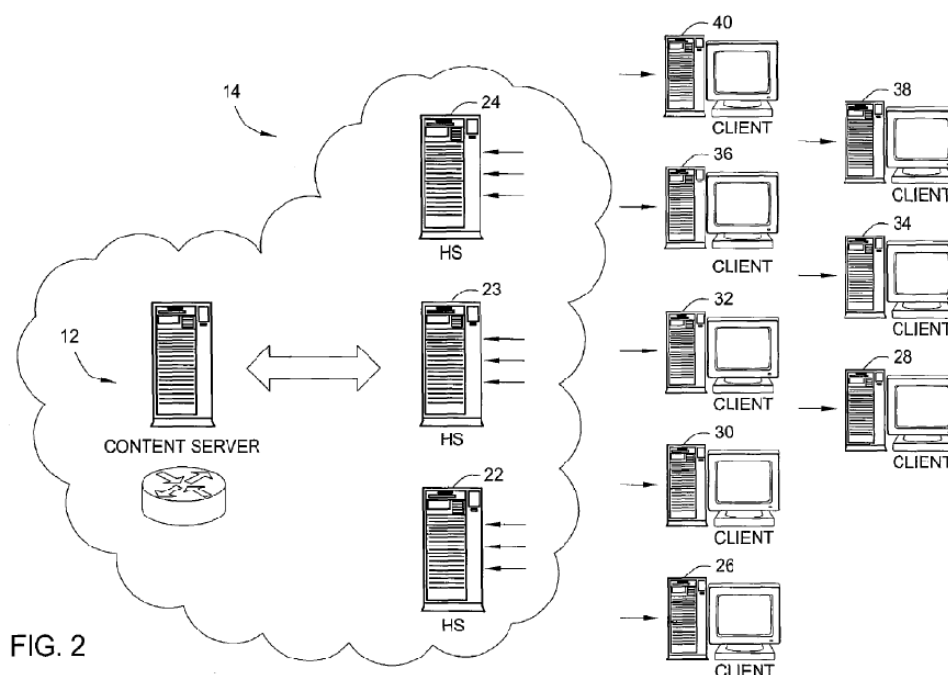
for such bandwidth constrained applications due to its high compression efficiency. A person of ordinary skill in the art would have understood and been motivated to also use HE AAC audio coding, as taught by Wolters, for the same reason—that is because it provides high compression efficiency allowing transmission of high-quality audio using less data, and by employing optimized decoding processes, a person of ordinary skill in the art would maximize the available processing resources to deliver the highest quality possible/practicable. Indeed, Wolters even provides an explicit teaching that suggests combining use of the MPEG-4 AVC standard (as taught by Wang) with use of the HE AAC standard (as taught by Wolters) in circumstances where quality at very low bitrates is needed or desired, like when transmission bandwidth and decoding resources are constrained in streaming over a wireless connection to a mobile device. (EX1041 at 13 (“By coupling HE AAC with MPEG-4 Video, more bits can be allocated to the video signal without degrading the quality of the audio signal.... Combined with the new MPEG/ITU Advanced Video Coding (AVC) standard (included in MPEG-4 as part 10), even more significant gains in quality are possible.”).) And, by employing optimized decoding processes, a person of ordinary skill in the art would maximize the available resources to enable use of this compression standard/method and deliver the highest quality video and audio playback experience possible/practicable.

E. Ground 4: Claims 1-9 are Obvious Over Carmel in View Wang, Wolters, Guo, Yu, and Forecast

305. Ground 4 incorporates the analysis of Grounds 1, 2, & 3 and adds Guo's [EX1051] teachings regarding the use of multiple servers to host and serve multimedia segments, Yu's [EX1061] teachings regarding transcoding (recited in claims 3 & 8) and Forecast's [EX1016] teachings regarding splitting a video stream at an I-frame (recited in claims 4 & 5).

306. Guo teaches streaming systems that use multiple "helper servers" (HSs) "to better support streaming media storage and distribution." (EX1051 at Abstract, Fig. 2.) Guo provides that "[t]he HSs [helper servers] serve to implement several methods specifically designed to support streaming multimedia, including segmentation of streaming multimedia objects (SM objects) into smaller units (i.e., chunks), cooperation of the HSs, and novel cache placement and replacement policies of the constituent units (chunks) which make up the SM objects." (*Id.* at 2:54-3:8.) Guo teaches that a multimedia stream is divided into smaller chunks/segments of data and that copies of these chunks are stored/cached on multiple helper servers that work together to transmit the full data stream to a client upon request. (*Id.* at 6:24-46 ("In accordance with the object segmentation approach of the present invention, the SM [streaming multimedia] object is divided into multiple chunks where each chunk is made up of a fixed number of segments. The

chunks which make up the SM object can then be cached and replaced independently, thereby significantly increasing the utilization of the cache storage.... **A SM object request will likely result in the client receiving some number of segments of the SM object from more than one HS 22-24.**”), 7:11-16 (“In either case, **it may be necessary for a client to get all the constituent segments of a SM [streaming multimedia] object from more than one HS 22-24.** To do so, some form of intelligent pre-fetching is required which pre-fetches subsequent segments while a current segment is being streamed to the client.”); *see also id.* at Abstract, 2:34-50, 3:65-4:6, 4:48-67, 6:2-46, 7:18-32, 11:17-40, 12:50-61, 13:12-19.)



Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

(EX1051 at Fig. 2.)

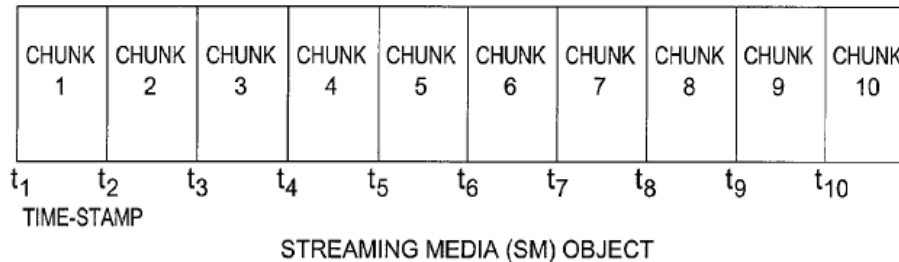


FIG. 5a

LOCAL SERVER CACHE 22	REMOTE CACHE 23	REMOTE CACHE 24	CONTENT SERVER 14
2,3,4,7,8,9	1,2,5,6	1,3,5,6,7,8,9,10	1,2,3,4,5,6,7,8,9,10

CACHE STORAGE TABLE

FIG. 5b

(EX1051 at Figs. 5a and 5b.) Thus, to the extent that claim element 1[b.ii] (and the corresponding elements in independent claims 6 & 7) requires that the multimedia objects are received by the multimedia player from a plurality of (that is, two or more) servers, Carmel teachings in view of Guo's render such an arrangement obvious.

307. A person of ordinary skill in the art would have been motivated to combine these teachings from Guo with the system taught by Carmel because such a person would have understood the resulting system employing multiple servers

would have been more robust, adaptable and better able to ensure that sufficient bandwidth/download rates are maintained and that the series of multimedia objects (akin to the “chunks” of multimedia data taught by Guo) are received by a client 30 (and the multimedia player thereon) in the proper sequence “with only a minimal necessary transmission ... delay.” (Carmel at 10:48-63; *see also id.* at 1:51-54 (“It is an object of some aspects of the present invention to provide substantially continuous, high-bandwidth data streaming over a network using common, existing server and network infrastructure.”), 3:36-39 (download rate by clients 30 should be generally equal to the data rate), 9:10-52, 10:64-11:22, 12:6-12.)

308. A person of ordinary skill would have recognized—as both Carmel and Guo teach—streaming multimedia data presented some distinctive challenges. For example, Guo provides:

SM objects can be generally differentiated from static web objects in that SM objects consist of multimedia data whose transmission has temporal characteristics such that the transmission rate is explicitly regulated or else the data becomes useless. In addition, the size of SM objects is typically at least an order of magnitude or two larger than that of a static web object, and therefore, do not lend themselves to be cached in their entirety. Given that caches have finite disk space, it is not feasible to statically store more than a few complete SM objects. If there are several simultaneous requests for different SM objects, it is

easy to show that the cache will be busy replacing one SM object with another resulting in significant performance degradation.

Based on the foregoing, there is a need for a system and method for enhancing current caching systems to support streaming multimedia over a public network system, such as the Internet.

(EX1051 at 2:34-50.)

309. A person of ordinary skill in the art would have been motivated to use the techniques taught by Guo, including the use of multiple helper servers to cooperatively serve chunks/slices of a multimedia data stream, to mitigate these challenges achieve the benefits and advantages from such as system as taught by Guo. (EX1051 at 3:1-8 (“The HSs reduce a content provider's memory and processing requirements by reducing the server load. Further, the invention reduces congestion problems by not being constrained by the unicast architecture of the prior art. And, the invention, reduces the long delays between the time video content is requested by a client and the time when the video content actually begins playing (*i.e.*, reduces high start-up latency).”), 5:11-16, 11:24-27 (reducing delay by locating content on servers geographically close to clients).)

310. Further, the application of Guo’s teaching regarding multiple servers to the streaming system taught by Carmel would be nothing more than an obvious combination of known elements (*e.g.*, implementing Carmel’s streaming system

using more than one host server 36) to achieve predictable results (*e.g.*, a system that reduces server load, congestion, and delays/latency in streaming multimedia content to one or more clients). Indeed, other systems that employed multiple servers to better serve multimedia content (and the advantages offered by such systems) were known in the art. (*E.g.*, EX1052 at Abstract, 6:29-33 (“In a further embodiment, the client module 114 is configured **to receive different portions of a content stream from a plurality of locations simultaneously. For example, the client module 114 may request and receive content from any of the plurality of web servers 116.**”), 7:66-8:13 (“In a further embodiment, the client module 114 may be configured to retrieve streamlets 212 from a plurality of web servers 310.”), 13:21-62, Fig. 1; EX1053 at [0007], [0014] (“In a further embodiment, the client module 114 may be configured to retrieve streamlets 212 from a plurality of web servers 310, each web server 116 may be located in various locations across the Internet 106.”), Figs. 1-4; EX1050 at Abstract, 3:20-26 (“According to the present invention, **digitized movie data may be accessed from different sources** (*e.g.*, **network servers**, local disks). These sources vary in speed and response time as a result of their capabilities (*e.g.*, limited I/O bandwidth) and transient events (*e.g.*, resource contention). The present invention adapts to these varying conditions in order to maintain a stable output.”), 3:48-56 (“Clips [consisting of a sequence of frames] may be stored in different

locations [including on different servers] within the digital network environment 102.”), 11:63-12:2, Fig. 1.)

311. Next, Yu teaches a transcoder. (*See, e.g.*, EX1061 at Abstract, 2:16-20, 2:44-49, Fig. 1.) Yu also teaches examples where source multimedia data has been compressed using an older, less sophisticated, or less desirable codec where such compressed source data is first decompressed before it is re-encoded in the desired/optimal format for streaming. (*See, e.g.*, EX1061 at 1:26-39 (“Another reason older codecs continue to be used is that many video streams have already been compressed with the older algorithms, and may easily be decompressed by the older codecs. However where high coding efficiency is desired, there arises the mixed use of both older and newer codecs. In some applications it is desirable to re-transmit video streams compressed with an older codec at a new bit-rate that is lower than the older codec can achieve for the same video quality. Therefore to obtain higher compression efficiency a transcoder having mixed codecs—an older decoder and a newer encoder—is used. A good example is a transcoder that converts MPEG2 compressed video streams to H.264 compressed video streams.”); 1:56-62 (“Accordingly the present invention provides rate control with a picture-based lookahead window for encoders/transcoders having mixed codecs in a dual-pass compressed video architecture. Statistics are extracted from an input video signal

according to a simple compression standard, the input video signal being a compressed video signal for transcoding”), 2:16-20 (“The extraction of the statistics is achieved in a transcoder by using a simple compression decoder to produce the statistics from the compressed video signal where the compressed video signal is the input video signal, ...”), 2:44-49 (“Referring now to FIG. 1 an encoder/transcoder architecture is shown having a simple compression decoder 12 for receiving and decoding a compressed video stream encoded according to a simple compression standard, such as MPEG2, to produce an uncompressed video signal and related statistics.”)

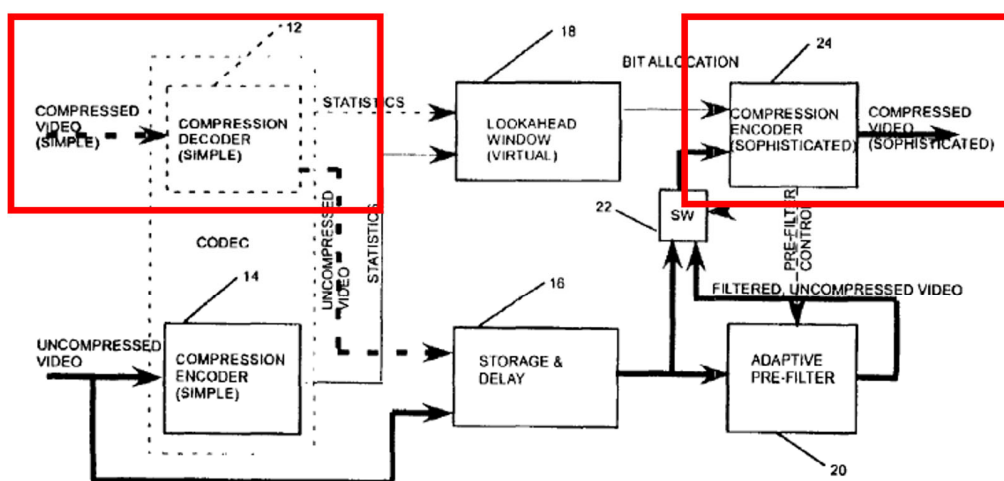


FIG. 1

(EX1061 at Fig. 1; *see also* EX1012 at 8:35-37 (“Typically, the input video will have been compressed using a conventional video compression process.”), 8:37-41, 8:48-59, 4:11-5:8, Fig. 2; *see also* EX1013 at [0025] (“If a digital video camera is the only

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

camera allowed, the base unit 12 includes only a digital video decompressor 18 that decompresses the lightly compressed AVI signals output by digital video cameras to output a digital format suitable for manipulation of the video image to mix in status data and suitable for [re-]compression.”.) Thus, to the extent the term “transcoded” in claims 3 & 8 is interpreted to require first decoding a compressed data source and then re-encoding the data in a desired format, this element is also taught and rendered obvious by Yu.

312. A person of ordinary skill in the art would have been motivated to combine these teachings of Yu with the system taught by Carmel because such a person would have understood that the source multimedia data (either from input devices 22 or a prerecorded sequence (Carmel at 6:32-35, 6:57-60)) often was and would have been compressed in some manner (as taught and suggested by Yu). (*See, e.g.*, EX1061 at 1:26-39, 2:44-49, Fig. 1.) A person of ordinary skill in the art would have been motivated to combine Yu’s transcoder teachings with Carmel because Yu teaches that its transcoder can beneficially extract statistics when decoding the source data that are used and helpful when re-encoding the data with a desired (typically more sophisticated and/or efficient) codec, making dual-pass encoding (which, Yu teaches, can provide higher coding efficiency than single pass-encoding) more efficient/less costly. (EX1061, 1:6-52, 1:56-2:23, 3:23-26, 9:7-19.) A person

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

of ordinary skill in the art would have further known that it was routine practice (particularly when dealing with lossy compression methods like MPEG) to first decode compressed data before re-encoding it in desired format at a desired rate, as taught and suggested by Yu. (*E.g.*, EX1061 at 1:26-39 (“Therefore to obtain higher compression efficiency a transcoder having mixed codecs—an older decoder and a newer encoder—is used. A good example is a transcoder that converts MPEG2 compressed video streams to H.264 compressed video streams.”).) Further, to the extent that such a transcoder/transcoding process was not already implemented within computer 34 taught by Carmel, adding such a process would be nothing more than an obvious combination (or substitution) of one known element/process (an encoder for encoding source data) with another known element/process (an encoder/transcoder that first decodes compressed source data before re-encoding it) to obtain predictable results (*e.g.*, data encoded/converted into the desired format with a desired size and encoding rate in an efficient and effective manner, less likely to have decoding errors, artifacts, and/or redundant and counterproductive encoding overhead). (*See, e.g.*, EX1061 at 1:26-39; EX1049 at 2:6-50, Figs. 1, 2 (transcoding techniques for converting an input bitstream to a different scaled output bitstream that involved fully or partially decoding the input stream before re-encoding the stream were known in the art); EX1058 at 2:1-4. 2:16-31 (compressed input streams

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

are first decompressed before being re-encoded for transmission to match the requirements of the destination terminal (including encoded parameters such as bit rate, frame rate, frame resolution, compression algorithm, etc.)); *also cf.* EX1059 at 2:1-6 (“It is generally considered ‘impossible’ to meaningfully compress already compressed data. More accurately, perhaps, it should be said that it is considered impractical to compress already compressed data, though it is true that most attempts at compression of already compressed files fails altogether and actually results in an increase in file size.”).)

313. Next, Forecast teaches and suggests that when splitting an MPEG encoded audio and video streams (*e.g.*, for purposes of splicing), the start of a new sequence (*e.g.*, a clip that is being spliced in) is scanned and split such that it begins with an I-frame. (EX1016 at 14:41-45; *see also id.* at 12:7-14, 22:53-65.) Forecast explains the rationale doing so, providing: “[t]he splicing of MPEG encoded audio/visual streams, however, is considerably more difficult than splicing of the uncompressed audio and video. The P and B frames cannot be decoded without a preceding I frame, so that cutting into a stream after an I frame renders the P and B frames meaningless.” (*Id.* at 2:11-16.) Forecast further teaches that such a procedure in which the start of a subsequent sequence of video frames/group of pictures (*i.e.*, an “In Point” of a second stream) is required to start with an I-frame was

incorporated into a standard promulgated by the Society of Motion Picture and Television Engineers (SMPTE). (*Id.* at 2:33-50 (“The In Point is just before a sequence header and I frame in a ‘closed’ GOP (**i.e., no prediction is allowed back before the In Point**).”). These teachings, combined with Carmel’s teachings regarding splitting a media stream into slices, suggest and render obvious splitting a multimedia stream such that each slice/segment/composite stream begins with an I-frame, as is recited by claims 4 & 5.

314. A person of ordinary skill would have appreciated and found it obvious to combine the teachings of Carmel and Forecast in such a manner because a POSA would have understood that the same issues and decoding errors that would potentially arise when a new spliced in segment started with one or more predicted frames (*i.e.*, P-frames or B-frames) as taught by Forecast (EX1016 at 2:11-16) would also potentially arise when decoding/playing back the segments/slices split according to Carmel’s teachings, particularly if/when slices were delayed or dropped during transmission. (*E.g.*, Carmel at 12:6-12.) A person of ordinary skill in the art would have been motivated to apply these teaching from Forecast to the systems and methods taught by Carmel and ensure that each slice was scanned and split to begin at an I-frame to avoid these potential decoding issues/errors.

**F. Ground 5: Claims 1-9 are Obvious Over Carmel in View
Mattavelli, Guo, Yu, and Forecast**

315. Ground 5 incorporates the analysis of Ground 1 and adds Guo's [EX1051] teachings regarding the use of multiple servers to host and serve multimedia segments, Yu's [EX1061] teachings regarding transcoding (recited in claims 3 & 8) and Forecast's [EX1016] teachings regarding splitting a video stream at an I-frame (recited in claims 4 & 5). Guo, Yu, and Forecast teach and render obvious these respective elements, as discussed in Ground 4 above and incorporated here. A person of ordinary skill in the art would have been motivated to combine the teachings of Carmel with Guo, Yu and Forecast, respectively, also for the reasons discussed in Ground 4 and incorporated here.

VII. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS

316. I understand from counsel that Patent Owner in the underlying district court litigation has not yet identified any evidence with respect to secondary considerations of non-obviousness. To the extent Patent Owner subsequently identifies any such evidence, I reserve the right to consider and revise any of my opinions should it be necessary to do so.

Declaration of Nathaniel Polish, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

VIII. CONCLUSION

317. I hereby declare that all statements made herein of my own knowledge are true and that all statements are 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 18 U.S.C. § 1001.

Dated: March 20, 2025

Respectfully submitted,

A handwritten signature in blue ink, appearing to read 'Nathaniel Polish', is written above a horizontal line.

Nathaniel Polish, Ph.D.