

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

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,771,849 B2**

Table of Authorities

	Page(s)
I. Introduction.....	1
II. Mandatory Notices under § 42.8(A)(1)	1
A. Real Parties-In-Interest under § 42.8.(b)(1)	1
B. Related Matters under § 42.8(b)(2)	1
C. Lead and Back-Up Counsel under § 42.8(b)(3)	1
D. Service Information	2
III. Fee Payment Under § 42.103	2
IV. Requirements under §§ 42.104 and 42.108	3
A. Grounds for Standing	3
B. Identification of Challenge and Statement of Precise Relief Requested	3
C. Discretionary Denial Is Not Appropriate Here	3
1. Discretionary Denial Under <i>General Plastic</i> Is Not Applicable	3
2. Discretionary Denial Under 35 U.S.C. § 314(a) Is Not Appropriate	4
3. Discretionary Denial Under 35 U.S.C. § 325(d) Is Not Appropriate	5
V. Overview of the '849 Patent.....	5
A. Level of Ordinary Skill in the Art	5
B. Priority Date	6
C. The Challenged Claims	6
D. Claim Listing	6
VI. Claim Construction.....	6
VII. The Challenged Claims are Obvious	7
A. Summary and Date Qualification of the Prior Art	7
1. Carmel [EX1003]	7
2. Mattavelli [EX1004]	10

Table of Authorities
(continued)

		Page(s)
3.	Wang [EX1007]	11
4.	Wolters [EX1041], Guo [EX1051], Yu [EX1061], and Forecast [EX1016]	12
B.	Ground 1: Claims 1-9 Are Obvious Over Carmel in View of Mattavelli.....	13
1.	Claim 1	13
a.	“A method comprising:” (claim 1 [preamble])	13
b.	“receiving audio and video segments encoded in a digital encoding format and with an encoding rate;” (claim 1[a.i])	13
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]).....	18
d.	“requesting by a multimedia player, transmission of said multimedia objects;” (claim 1[b.i]).....	26
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])	28
f.	“playing back received multimedia objects by the multimedia player,” (claim 1[c.i])	33
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])	35
h.	“autonomously adjusting said selection and playback according to the multimedia object parameters and supplied host path identification,” (claim 1[c.iii])	40

Table of Authorities
(continued)

	Page(s)
i. “and by utilizing optimized decoding processes to maintain quality playback.” (claim 1[c.iv]).....	45
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.”.....	55
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.”	56
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.”	60
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.”	64
6. Claim 6 (Independent)	65
7. Claim 7 (Independent)	66
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.”	67

Table of Authorities
(continued)

	Page(s)
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.”	68
C. Ground 2: Claims 1-9 Are Obvious Over Carmel in View of Wang.....	70
D. Ground 3: Claims 1-9 Are Obvious Over Carmel in View of Wang and Wolters	78
E. Ground 4: Claims 1-9 are Obvious Over Carmel in View of Wang, Wolters, Guo, Yu, and Forecast	80
F. Ground 5: Claims 1-9 are Obvious Over Carmel in View of Mattavelli, Guo, Yu, and Forecast	86
VIII. Secondary Considerations	86
IX. Conclusion	86
CERTIFICATE OF SERVICE	89

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

List of Exhibits

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) (“’849 patent”)
1002	Declaration of Dr. Nathaniel Polish (“Polish”)
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”)
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)

List of Exhibits

Exhibit No.	Description of Document
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”)
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, Digital Video Compression (2004) (“Symes”)
1021	Excerpts from Fernando Pereira, Touradj Ebrahimi, The MPEG-4 book (2002) (“Pereira et al.”)

List of Exhibits

Exhibit No.	Description of Document
1022	Excerpts from Craig Hunt and Robert Bruce Thompson, Windows NT TCP/IP Network Administration (1998) (“Hunt et al.”)
1023	Excerpts from Douglas E. Comer, Internetworking with TCP/IP Vol. 1: Principles, Protocols, and Architectures (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, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia (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”)
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”)

List of Exhibits

Exhibit No.	Description of Document
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 (April 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”)

List of Exhibits

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-Pute: 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”)

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

List of Exhibits

Exhibit No.	Description of Document
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”)
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”)

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

List of Exhibits

Exhibit No.	Description of Document
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

I. INTRODUCTION

Petitioner Google LLC respectfully submits this Petition for *Inter Partes* Review of claims 1-9 of U.S. Patent No. 10,771,849 B2 (“’849 patent”) (EX1001). Petitioner requests institution of *Inter Partes* review (“IPR”) and a finding that all challenged claims are unpatentable.

II. MANDATORY NOTICES UNDER § 42.8(A)(1)

A. Real Parties-In-Interest under § 42.8(b)(1)

Google LLC (“Petitioner”) and YouTube, LLC, are the real parties-in-interest to this IPR petition.¹

B. Related Matters under § 42.8(b)(2)

The ’849 patent is asserted in pending litigation involving Petitioner and Patent Owner (“PO”): *Withrow Networks, Inc. v. Google LLC*, No. 5:24-cv-03203-PCP, N.D Cal. The complaint in this related case was first served on June 4, 2024. (EX1055; EX1056.)

C. Lead and Back-Up Counsel under § 42.8(b)(3)

Petitioner provides the following designation of counsel.

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

LEAD COUNSEL	BACK-UP COUNSEL
Eamonn Gardner (Reg. No. 63,322) egardner@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (858) 550-6086 Fax: (720) 566-4099	Reuben Chen (<i>pro hac vice</i> to be filed) rchen@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5480 Fax: (650) 849-7400
	Joseph Van Tassel (Reg. No. 74,136) jvantassel@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (703) 456-8647 Fax: (703) 456-8100

D. Service Information

This Petition is being served by Federal Express to the attorneys of record for the '849 patent: Allan Colquhoun and Yasin Bismilla, 132285-ELAN IP INC., 2 Bloor Street East, Unit 3500, Toronto, ON M4W1A8, Canada. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

III. FEE PAYMENT UNDER § 42.103

Petitioner requests review of 9 claims, with a \$51,875 payment.

IV. REQUIREMENTS UNDER §§ 42.104 AND 42.108

A. Grounds for Standing

Petitioner certifies that the '849 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioner requests the Board institute IPR of claims 1-9 based on:

Ground	Claims	Basis for Challenge under §103(a)
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

Submitted with this Petition is a Declaration of Nathaniel Polish, Ph.D., (EX1002; "Polish"), a qualified expert. (Polish, ¶¶4-17.)

C. Discretionary Denial Is Not Appropriate Here

1. Discretionary Denial Under *General Plastic* Is Not Applicable

This is the first IPR petition filed against the '849 patent. The factors set forth in *General Plastic* are not applicable here and do not support discretionary denial.

2. Discretionary Denial Under 35 U.S.C. § 314(a) Is Not Appropriate

The *Fintiv* factors do not support discretionary denial.

The related litigation between PO and Petitioner is still in its initial stages—no trial date has been set (*Fintiv* factor 2), discovery is in its opening phases (PO has served two sets of written discovery and its preliminary infringement contentions, and the parties have exchanged initial disclosures, but, other than the document production accompanying Withrow’s infringement contentions, no documents have been produced nor have any depositions been noticed; Petitioner’s invalidity contentions are not due until April 21, 2025), no claim construction exchanges or rulings have occurred, and no *Markman* hearing date has been set (*Fintiv* factors 3 & 4). (See EX1065, 12-13.) The Petition was filed months before the one-year deadline under §315(b). Additionally, as demonstrated below, the merits of the petition are particularly strong (*Fintiv* factor 6), presenting compelling evidence of unpatentability.

Accordingly, *Fintiv* factors 2, 3, 4, and 6 weigh against discretionary denial. And, the remaining *Fintiv* factors 1 (stay of the district court litigation) and 5 (parties in the parallel proceeding) are presently neutral and do not alter this balance against discretionary denial.

3. Discretionary Denial Under 35 U.S.C. § 325(d) Is Not Appropriate

This Petition does not present a situation in which “the same or substantially the same prior art or arguments previously were presented to the Office.” 35 U.S.C. § 325(d). Although Carmel was included in an Information Disclosure Statement submitted by the applicant (EX1008, 141-143, 524-526), it was not discussed or otherwise acknowledged by the Examiner during prosecution. The Examiner also did not have the Mattavelli and Wang references that this petition relies on, which complement the teachings of Carmel and together with Carmel provide compelling grounds of obviousness.

Further, Petitioner respectfully submits that the Examiner materially erred by not rejecting the claims as obvious over Carmel in view of Mattavelli or Wang. As explained below, the combinations of these references render obvious claims 1-9, including the features/limitations that the Examiner mistakenly identified as not present in the prior art in the notice of allowance. (EX1008, 589; Polish, ¶73.)

V. OVERVIEW OF THE '849 PATENT

A. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSA”) as of the alleged priority date would have possessed at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or an equivalent field requiring learning

computation principles, and two years of work experience in networking or streaming media systems, particularly audio and video, over the Internet. Additional education could compensate for less practical experience and vice versa. (Polish, ¶¶18-23.)

B. Priority Date

The '849 patent claims priority to several earlier-filed applications, the earliest of which (application no. 11/107,952) was filed on April 18, 2005. Petitioner uses this date as the priority date of the challenged claims. PO has not alleged that the '849 patent is entitled to an earlier priority date.

C. The Challenged Claims

This Petition challenges claims 1-9 of the '849 patent.

D. Claim Listing

Exhibit 1057 lists the claims, enumerating each element. Bracketed notations (*e.g.*, [a.i]) are used to refer to individual limitations.

VI. CLAIM CONSTRUCTION

For purposes of this petition, Petitioner does not presently contend that any term requires express construction. Petitioner and PO have not yet exchanged any terms or proposed constructions in their related litigation involving the '849 patent.

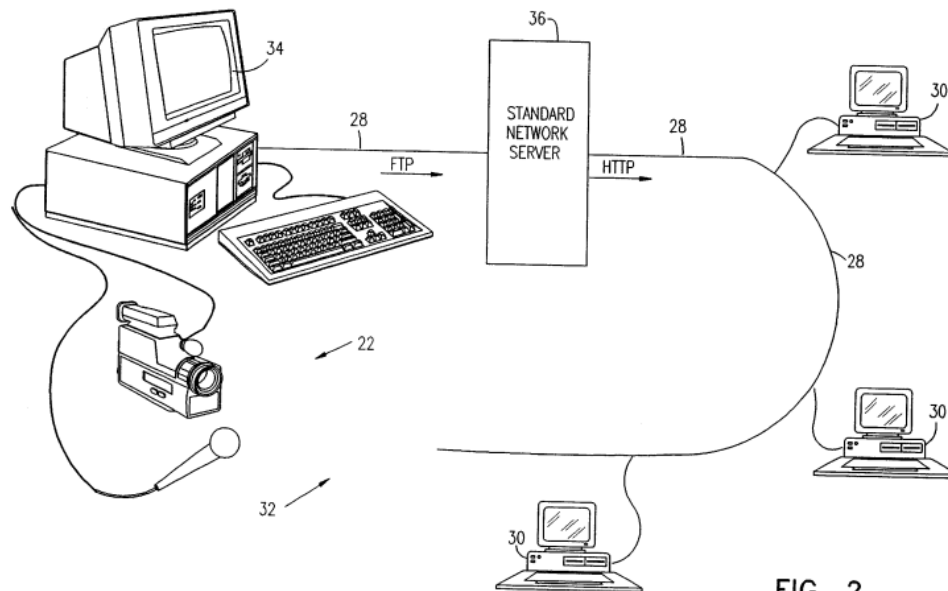
VII. THE CHALLENGED CLAIMS ARE OBVIOUS

As detailed below, claims 1-9 of the '849 patent are obvious in view of the identified prior art. The technology and concepts underlying the challenged claims were long known (alone and in combination) to those in the field. (*See Polish*, ¶¶40-60, 75-110.)

A. Summary and Date Qualification of the Prior Art

1. Carmel [EX1003]

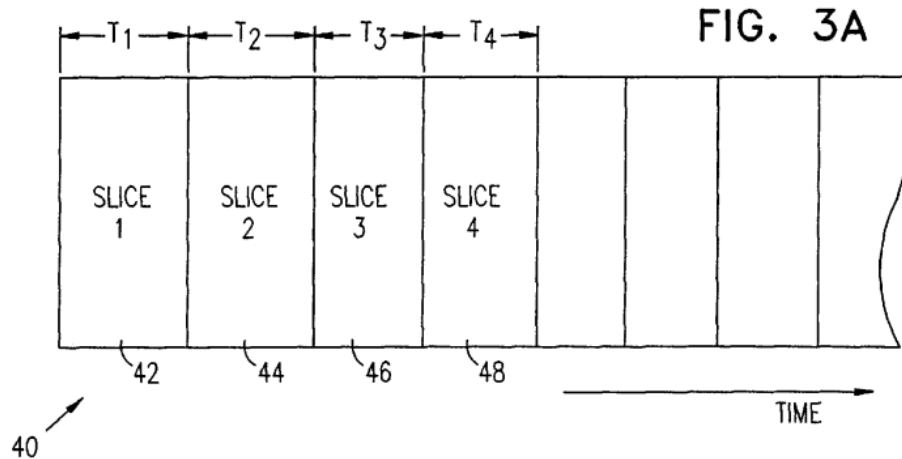
Carmel, U.S. Patent No. 6,389,473, is titled “Network Media Streaming” and issued on May 14, 2002. Carmel relates to “real-time multimedia broadcasting over a network.” (Carmel, 1:10-13, 6:24-35.)



(*Id.*, Fig. 2.)

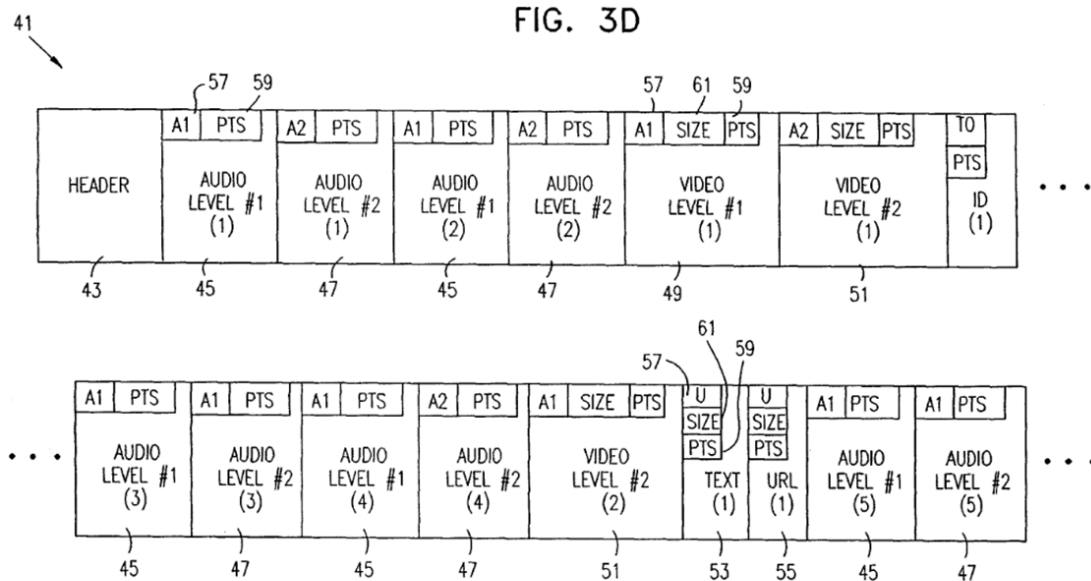
Carmel teaches that after receiving an input/source multimedia stream,

computer 34 produces a “stream of broadcast data 40” that “comprises a series of data slices.” (Carmel, 6:30-34, 7:18-28.)



(*Id.*, Fig. 3A.)

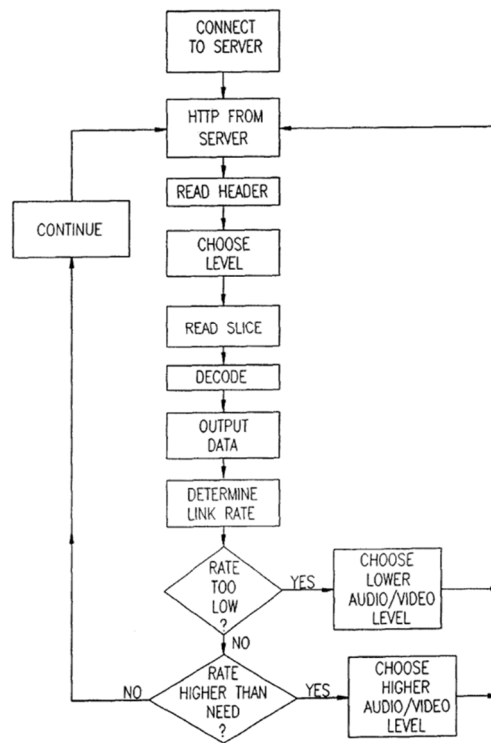
In a multi-level data stream embodiment, the stream is divided into different slices of audio and video wherein “[e]ach time slice in stream 41 includes multimedia data at multiple quality levels.” (Carmel, 8:42-57.) Each client 30 chooses “the quality level appropriate to the bandwidth of its link on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



(*Id.*, Fig. 3D.)

Clients 30 connect to server 36 and begin downloading and consuming segmented stream 40/41, “substantially in real time.” (Carmel, 7:4-17, 8:1-7, 10:24-54.) The client “monitors the rate of data coming in over ... its links with server 36.” (*Id.*, 10:55-56.) Carmel teaches multiple means of maintaining a sufficient transmission rate between server 36 and clients 30, that can be used separately or together, explaining the data rate “should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.” (*Id.*, 2:57-59, 11:1-22.)

FIG. 6B



(*Id.*, Fig. 6B.)

Prior to slicing, Carmel teaches that the multimedia data is preferably compressed by computer 34 “using any suitable compression method known in the art.” (Carmel, 11:23-52, Fig. 7.) Carmel also teaches that each client includes a media player that controls the selection and playback of the broadcast data stream. (*Id.*, 7:14-17, 10:28-34, 13:53-14:16; *see also* Polish, ¶¶78-88.)

2. Mattavelli [EX1004]

Mattavelli is a paper entitled “Computational Graceful Degradation for Video Decoding” by Marco Mattavelli and Sylvain Brünnettem. Mattavelli was published

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

as part of the materials delivered at the 9th European Signal Processing Conference (EUSIPCO 1998), held between September 8-11, 1998. (EX1004.) Mattavelli was additionally published at least by April 22, 1999, as further explained in the expert declaration of Dr. Ingrid Hsieh-Yee. (EX1063, ¶¶26-42.) Mattavelli teaches computationally graceful degradation techniques for video decoding that use simplified, optimized processes to ensure playback continuity in periods where decoding processing resources are overtaxed. (EX1004, 2309-12.) An additional brief summary of Mattavelli is provided in Dr. Polish's declaration. (Polish, ¶¶89-94.)

3. Wang [EX1007]

Wang is a paper entitled "A Platform-Based MPEG-4 Advanced Video Coding (AVC) Decoder with Block Level Pipelining" by Shih-Hao Wang et al. 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), held between December 15-18, 2003. (EX1007.) Wang was additionally published at least by May 4, 2004, as explained by Dr. Hsieh-Yee. (EX1063, ¶¶43-69.) Wang teaches decoding optimizations to allow processing-constrained mobile devices to decode the efficient MPEG-4 Advanced Video Coding (AVC) compression standard. (EX1007, 51.) An

additional brief summary of Wang is provided in Dr. Polish's declaration. (Polish, ¶¶95-98.)

4. Wolters [EX1041], Guo [EX1051], Yu [EX1061], and Forecast [EX1016]

Wolters is a paper entitled "A closer look into MPEG-4 High Efficiency AAC" that was published as part of the 115th AES Convention held between October 10-13, 2003, and was additionally published by December 12, 2003. (EX1041; EX1054; EX1063, ¶¶70-106.) Brief summaries of **Wolters**, which teaches optimized processes for decoding high efficiency compressed audio; **Guo** (U.S. Patent No. 7,398,312, filed March 29, 2000), which teaches use of multiple servers to stream multimedia content; **Yu** (U.S. Patent No. 7,099,389, filed December 10, 2002), which teaches transcoding compressed multimedia to a different standard; and **Forecast** (U.S. Patent No. 7,096,481, filed March 31, 2000), which teaches techniques for splitting and splicing compressed video, are provided in Dr. Polish's declaration. (Polish, ¶¶99-110.)

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

1. Claim 1

a. “A method comprising:” (claim 1 [preamble])

Carmel discloses “[a] **method**² for real-time broadcasting from a transmitting computer to one or more client computers over a network[.]” (Carmel, Abstract, 3:24-27.) Accordingly, if the preamble is limiting, it is taught by Carmel. (Polish, ¶¶113-115.)

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

First, Carmel teaches the creation, transmission, and receipt of “audio and video segments” from a multimedia data stream—consisting of video images and audio/sound—from, *e.g.*, input devices 22 or a prerecorded multimedia sequence. (Carmel, 1:20-28, 2:29-40, 6:24-35, 6:57-60, 9:64-66, Figs. 1 & 2; Polish, ¶¶117-118.)

² Unless indicated otherwise, all bold emphasis in quotes has been added.

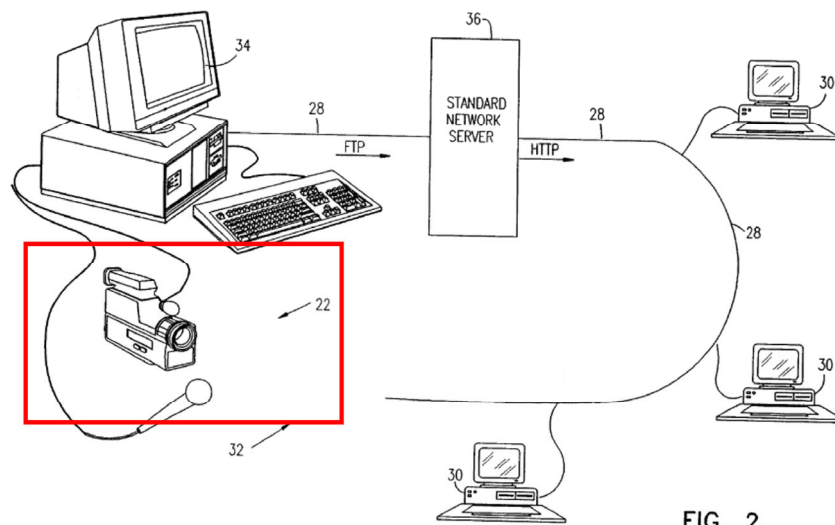
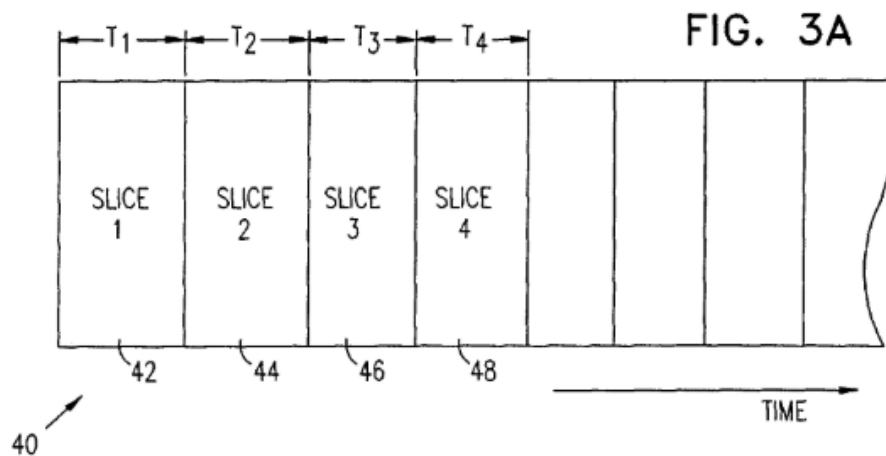


FIG. 2

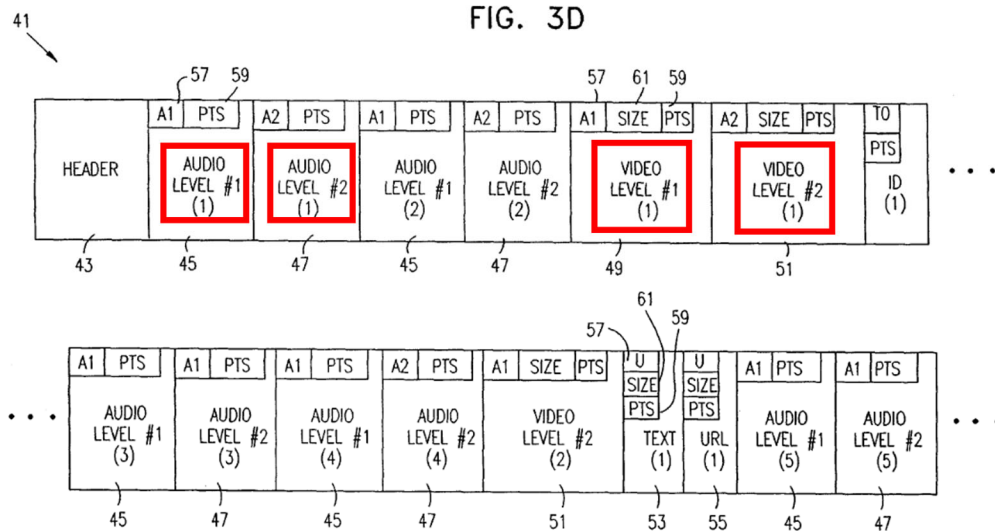
(Carmel, Fig. 2.)

Carmel teaches that the source multimedia data stream is divided by computer 34 into a sequence of “slices” or segments of video and audio data. (Carmel, 7:18-25 (“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[.]”), Abstract, Fig. 3A.)



Carmel further teaches that the audio and video data can be divided into

multiple slices/segments with multiple, different quality levels for each interval in the sequence. (*Id.*, 8:42-59, Fig. 3D; Polish, ¶¶119-120.)



Next, Carmel teaches that the audio and video segments/slices are “encoded in a digital encoding format with an encoding rate.” Carmel teaches that prior to slicing it into segments, the source multimedia data is first “encoded” or compressed “using any suitable compression method known in the art.” (Carmel, 11:23-28, *see also id.*, 2:1-7, 2:41-50, 6:50-7:3, 7:25-26, 9:62-10:1 (“[t]he data are compressed at step 80, and are then ‘sliced’ at step 82”), 10:19-22 (“encoding/quality level (step 80)” may be adjusted), 11:23-64, Figs. 3A, 3D, 5 & 7.)

FIG. 5

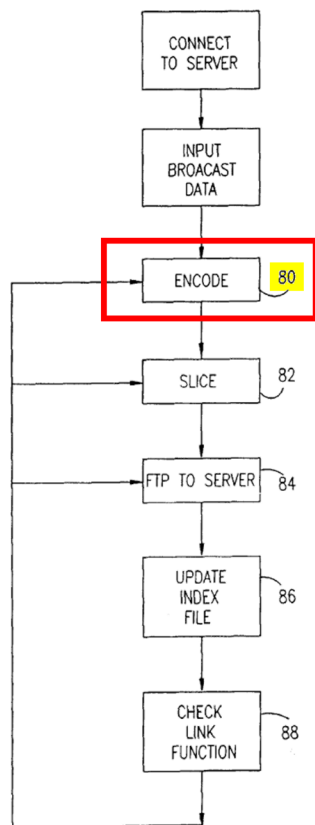
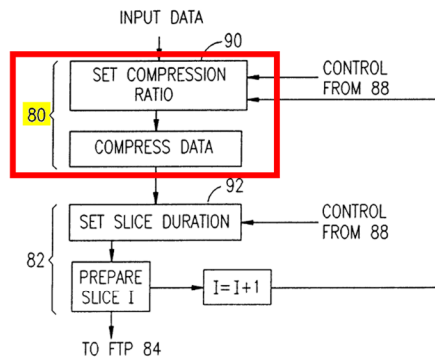


FIG. 7



(Carmel, Figs. 5 & 7.) Carmel also provides examples of “suitable compression method[s] known in the art” including GSM 6.10 (audio), H.263 (video), and MPEG standards (audio and video). (Carmel, 11:23-52; Polish, ¶¶121-123.)

Carmel also teaches and suggests that the audio and video data would be compressed with a particular bitrate or encoding rate, in accord with how a POSA would have understood such “suitable compression method[s] known in the art” to function. (Polish, ¶¶124-126.) For example, Carmel provides that the compressed data stream has a “given data rate,” and teaches that this data- or encoding-rate

should approximate the available network bandwidth for uploading/downloading the audio and video data to/from host server 36. (Carmel, Abstract, 2:56-59, 3:24-59, 9:31-33, 11:9-22, 14:18-32.) Carmel teaches multiple ways (that can be used separately or together) to achieve this encoding rate/available bandwidth balance, including, *e.g.*, by adjusting the compression ratio used and/or by providing each time slice of audio and video data “at multiple resolution or quality levels” where “[e]ach such level has a different degree of data compression, and thus corresponds to a different data bandwidth requirement.” (Carmel, 3:5-13, 4:39-47, 5:15-18, 8:42-9:9, 10:19-22, 10:64-11:22, 11:39-52; Polish, ¶¶124-126.)

Finally, Carmel also teaches the “receiving” of such encoded/compressed audio and video segments/slices—*e.g.*, by clients 30 from server 36. (*E.g.*, Carmel, 7:4-5, 7:12-17, 10:24-34, 10:64-11:22 (“The slices are received, decoded and output by the client.”), 14:61-64, Figs. 2, 3D, 6B; Polish, ¶127.)

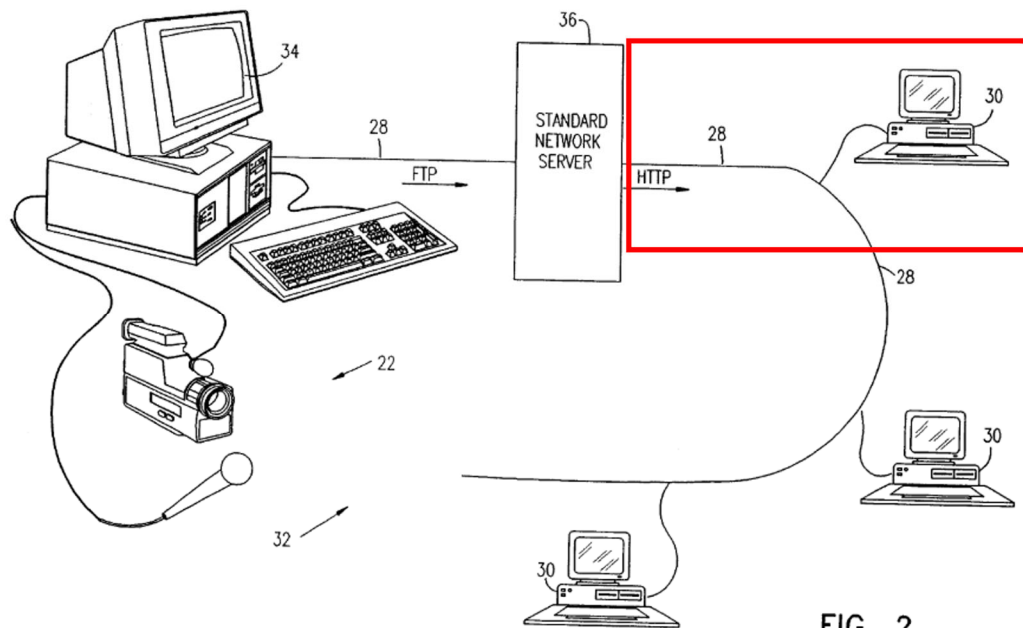


FIG. 2

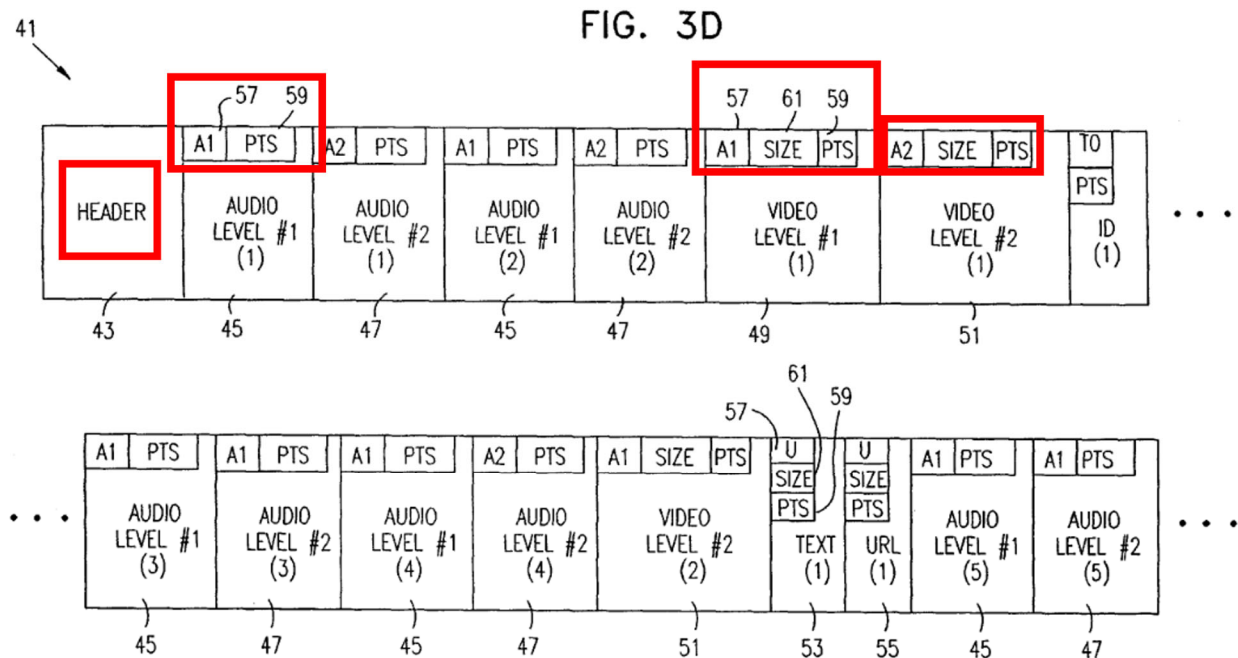
Carmel thus satisfies this claim element. (Polish, ¶¶116-128.)

- 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])

Carmel teaches that the audio and video segments/slices (*see* element 1[a.i])—which can be a sequence of discrete files (*e.g.*, Carmel, Abstract, 2:1-23, 3:24-39, 5:15-18; Polish, ¶131)—are associated with object parameters and supplied host path identification to form multimedia objects.

Carmel teaches that each slice, or file corresponding to each slice, includes metadata or object parameters pertaining to that slice, including, *e.g.*, a slice index identifier and time stamps/codes. (*See, e.g.*, Carmel, 7:23-31 (“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[.]”), 2:6-7, 7:36-49, 3:24-39, 14:18-32, 15:8-17, Fig. 3A.) For multi-level streams, Carmel teaches that each multimedia slice also includes “size identifier 61” (in addition to slice identifiers and time stamps), 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, 8:42-55, Fig. 3D; Polish, ¶132.)



Carmel teaches and renders obvious embodiments of a multi-level data stream where the slices of multimedia data at multiple quality levels (and the object parameters associated with each slice) are saved within a single file and embodiments where each slice is saved as a separate file. (See, e.g., Carmel, 8:42-

9:5, 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.”), 2:22-28, 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. (*Id.*, 9:49-52.) It further would have been obvious for each slice/file to have a header containing any necessary or helpful information or parameters (*e.g.*, slice index, time stamps, encoding/decoding metadata) for ensuring synchronized playback in the correct sequence. (*E.g., id.*, 2:15-21, 8:47-55, 10:45-54.) Either way—whether the slices are saved within a single file or as separate files—each audio and video slice has associated object parameters (and, as discussed below, host path identification) and thus would be understood to form a multimedia object. (Polish, ¶133.)

Carmel also teaches that the audio and video slices may be accompanied by an index file containing “**parameters relating generally to the data stream and/or instructions to [client] computers 30**” associated with the audio and video slices. (Carmel, 7:59-8:1, 8:12-17, Fig. 3B.) A POSA would have understood and found it obvious that such parameters and/or instructions could additionally or alternatively be included in the header of a multi-level multimedia object slice/file, as this would

be nothing more than a simple substitution of one known element for another known element to obtain predictable results. (Polish, ¶134.)

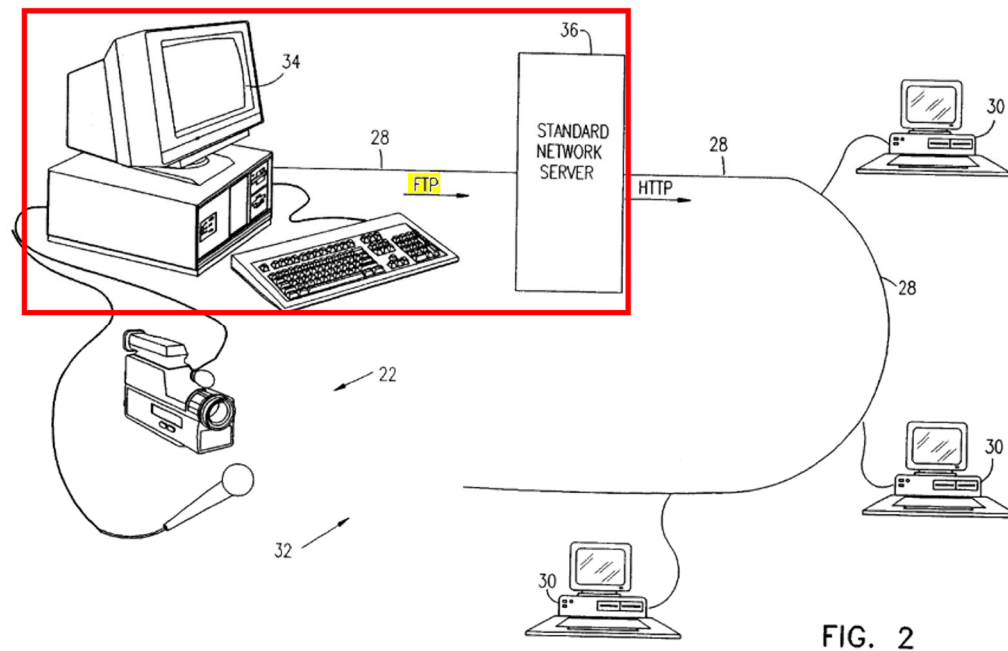
In addition to these object parameters disclosed by Carmel (Polish, ¶¶133-135), a POSA would have recognized that additional metadata/object parameters (*e.g.*, video frame size, resolution, audio sampling rate) would have been included and associated with audio and video data slices that have been compressed “using any suitable compression method[s] known in the art,” such as MPEG-4-based standards. (Polish, ¶136 (citing, *e.g.*, EX1005, 8-9, 13, 20, 27-28, 38, 49-51, 54 (“MPEG [allows] control code to be embedded in the sequence header to help the decoder.”), 273, 329-34; EX1021, 44-45 (MPEG-4 coded data includes “scene description data, control data in the form of object descriptors, or metadata describing the content”), 47, 65-66, 72-75, 76-81, 86-87; EX1028, 3, 19-20, 103.)

Mattavelli teaches, for example, that the MPEG-4 video standard can include decoding “complexity prediction information” along with the video information itself, which (as will be discussed below) is useful in implementing the computationally graceful decoding techniques taught by Mattavelli.³ (Mattavelli,

³ The reasons why a POSA would have been motivated to use MPEG-4 compression (and this decoding complexity prediction metadata) and combine the teachings of

2309, 2311; *see also* EX1024, 2:16-3:4 (“[p]arameters relating to the [decoding] computational complexity requirements” included in audiovisual packet headers); EX1021, 77-79.) This metadata added by the compression codec are additional object parameters associated with the audio and video slices. (Polish, ¶¶136-137.)

Carmel also teaches and renders obvious supplying the audio and video segments/slices with “host path identification.” Carmel teaches that the sequence of multimedia slices/files created by computer 34 are uploaded to the host server 36, preferably using FTP. (Carmel, 6:50-7:3, 9:10-10:23, Figs. 2, 4-5.)



Mattavelli and Carmel are described with claim element 1[c.iv], below. (Polish, ¶¶136, 138.)

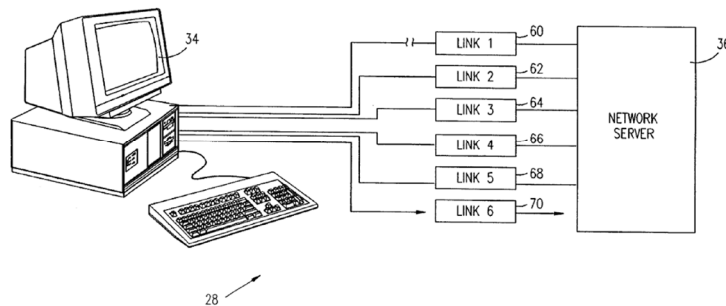
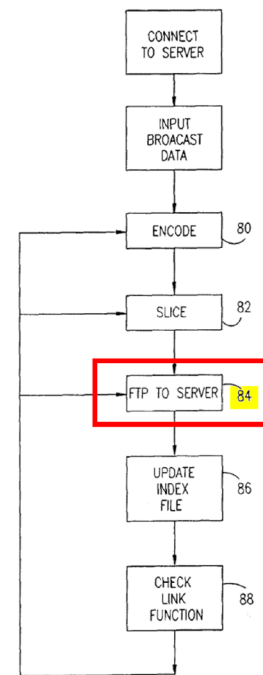


FIG. 5



Carmel states that when uploading the slices/files to server 36, “the computer preferably opens a plurality of links, each link having its own URL (Uniform Resource Locator)” and further teaches that a similar technique may be employed by clients 30 for downloading the files. (Carmel, 9:17-19, 9:46-52, 13:29-35.) Once uploaded, “[c]lients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. Clients 30 preferably download the sequence using Hypertext Transfer protocol (HTTP), although other Internet protocols may also be used[.]” (*Id.*, 7:4-17, 10:36-42, 10:67-11:8.) Client 30 then “reconstructs and outputs the multimedia data for the appreciation of a user.” (*Id.*, 10:48-50.)

Carmel’s teachings regarding the use of URLs and Internet protocols, such as

FTP and HTTP, for downloading/uploading the multimedia slices/files indicate that the slices are supplied with host path identification—*e.g.*, where on the host server 36 each respective slice/file in the sequence is stored so that it can be supplied to clients 30 that request it using, *e.g.*, HTTP. (Polish, ¶¶139-143.) As a POSA would have appreciated, “URLs point to pieces of electronic information, telling you where they are located and how to interact with them[,]” and include a “path” component that “specifies where on the server machine the resource lives.” (EX1006, 23, 28.) As shown below, the URL structure used in HTTP and FTP are similar, and each includes a path identifier component.

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>
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>

(EX1006, 38-39; EX1022, 413-14 (host path identifiers in URLs); Polish, ¶¶143-

144.)⁴

Further, even if Carmel’s disclosure regarding using URLs in conjunction with Internet protocols such as FTP and HTTP did not expressly or implicitly disclose supplying the multimedia slices/files with host path identification, at minimum, these teachings render such host path identification obvious. It would have been obvious and beneficial in providing a functional streaming system, like Carmel’s, to supply the audio and video slices/segments with host path identification that identifies where on server 36 the slices/files are located so that they can be uploaded and subsequently located and downloaded correctly by clients 30 for playback, using, *e.g.*, these routine, well-known Internet protocols and URL path identifiers. (Polish, ¶145.) This would be nothing more than the use of well-known techniques to yield predictable results—*e.g.*, identifying the location of resources on a server. (*Id.*)

Accordingly, as they are associated with object parameters and supplied host path identification, the audio and video segments/slices/files taught by Carmel “form multimedia objects,” and Carmel (and Carmel in view of Mattavelli) teach and

⁴ PO alleges that URLs constitute host path identifiers and satisfy this term in the related litigation. (*E.g.*, EX1055 at 20, 39-41, 75-77; Polish, ¶143, n.5.)

render obvious this claim element. (Polish, ¶¶129-130, 146-147.)

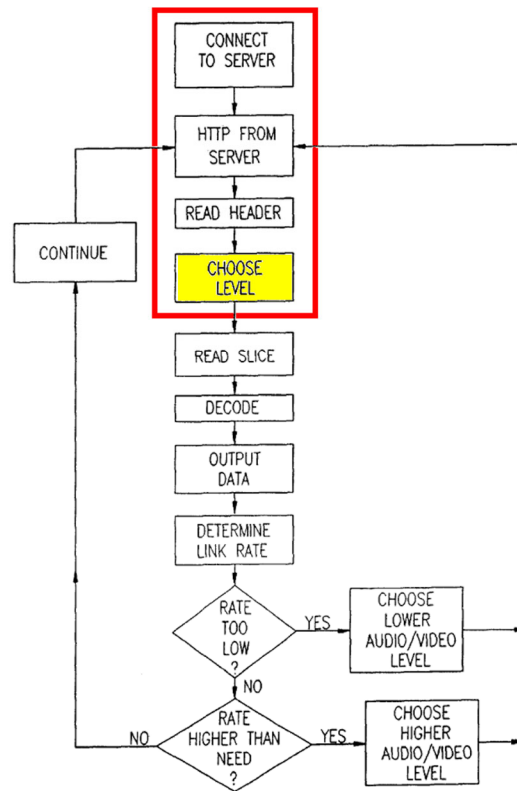
d. “requesting by a multimedia player, transmission of said multimedia objects;” (claim 1[b.i])

Carmel discloses and renders obvious a multimedia player, teaching that client computers 30 are configured to play back a sequence of multimedia slices received from server 36. (*E.g.*, Carmel, 2:15-17, 3:66-4:2, 4:3-7, 5:22-25, 10:24-11:22 (“[C]lient 30 reconstructs and outputs the multimedia data for the appreciation of a user.”).) Carmel teaches that a multimedia player controls the operation of the client (*e.g.*, the selection, receipt, decoding, and output/playback of the multimedia slices) and that it can be a “Java applet or plug-in” to the client’s web browser, providing code for a “sample applet” with its disclosure. (Carmel, 7:12-17, 10:27-35 (“**The operation of client is controlled by a Java applet[.]**”), 10:64-11:2, 13:53-14:16; Polish, ¶¶148-151.)

Next, Carmel teaches that the client 30 (controlled by the media player thereon) chooses and requests transmission of the desired multimedia objects (and quality level in the multi-level data stream embodiment). For example, Carmel provides: “each client 30 connects to the server, ... 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.**” (Carmel,

10:64-11:8, *see also id.*, 11:9-22, Fig. 6B.)

FIG. 6B

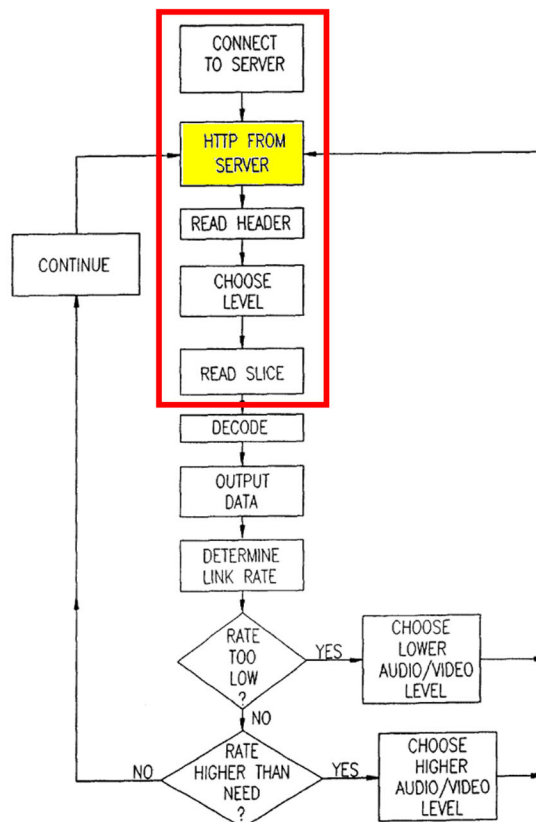


Indeed, Carmel repeatedly teaches and suggests that client 30 (controlled by the multimedia player) requests the transmission of the desired multimedia objects from server 36. (*E.g.*, Carmel, 2:11-15 (client preferably uses HTTP), 3:8-11, 7:4-9, 10:35-47, Fig. 6A; EX1006, 4-5, 8-9 (HTTP GET requests for transmission of resource from server).) Accordingly, Carmel teaches and renders obvious this claim element. (Polish, ¶¶152-155, *see also id.*, ¶¶48-51.)

- 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])

First, Carmel teaches locating and downloading the multimedia objects/slices using HTTP. As discussed above, Carmel provides that “clients download the data stream from the server ... preferably using the Hypertext Transfer Protocol[.]” (Carmel, 2:11-15, 10:64-11:22 (HTTP link used to locate and select the desired multimedia objects).)

FIG. 6B



(*Id.*, Fig. 6B; *see also id.*, 7:4-9, 10:35-47, Figs. 2-4, 6A; Polish, ¶¶156-158.)

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. (Polish, ¶159.) A POSA would have understood that to download a particular object/slice of multimedia data, the particular object/slice must first be identified and located. It would have been a routine, obvious use of HTTP commands and URLs to perform these actions, as is taught and suggested by Carmel (Carmel, 9:13-27, 9:46-47, 10:35-47, 10:64-11:8) and was known by those of ordinary skill in the art. (*E.g.*, EX1006, 4-9, 23-29, 38-39 (use of HTTP with URLs to locate/identify and request resources), 120-126; EX1022, 415; Polish, ¶¶48-51, 159.)

Next, Carmel also teaches and renders obvious having the multimedia player on client 30 receive the multimedia objects from server 36 using a wireless connection. As shown in Figure 2, Carmel teaches that clients 30 connect to host server 36 via a network 28.

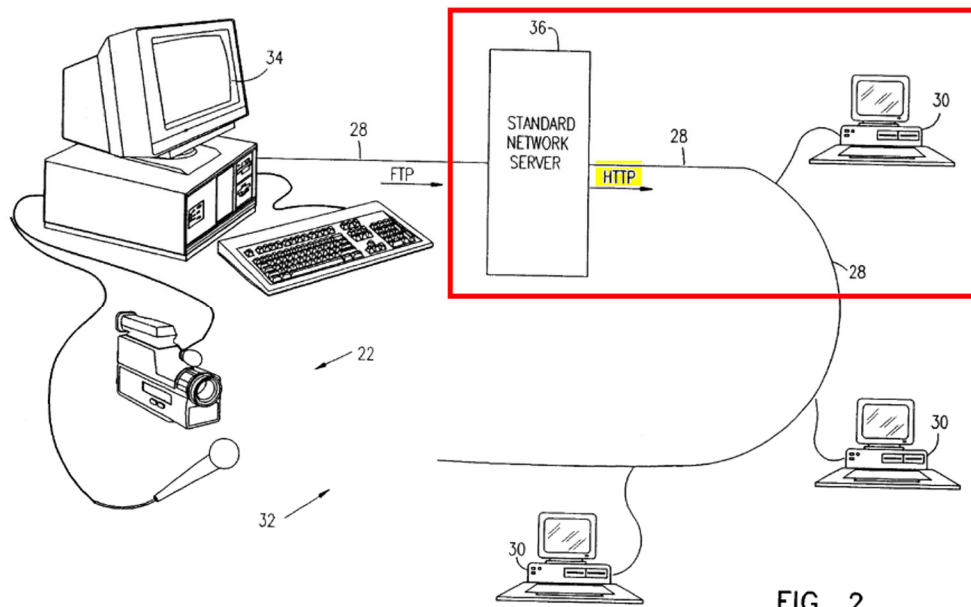


FIG. 2

(Carmel, Fig. 2.) Carmel provides that “[n]etwork 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, 6:36-38; Polish, ¶¶160-161.)

Carmel’s teaching of a generic computer network 28 suggests and renders obvious the use of a wireless connection by client 30, as wireless networks/connections and clients—including for use in multimedia streaming—were well-known in the art before the priority date. (*E.g.*, EX1007, 1 (“Multimedia services over mobile network[s] are becoming important with the pervasive presence of the Internet and wireless devices.”); EX1018, 36-37, Fig. 1; EX1012, 1:29-31; EX1025, [0015]-[0016], Fig. 1; EX1027, 657; EX1021, 2 (“Mobile connections will not be limited to voice; other types of data, including real-time media, are already emerging.”); Polish, ¶¶59, 162.)

Carmel also expressly teaches the use of a wireless connection by a client to a server via its incorporation of Carmel '432 (EX1009) and Carmel '230 (EX1010). (See Carmel, 6:65-7:3, 2:41-46; Polish, ¶¶86-87, 164.)⁵ Carmel '432 provides, for example: “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....** Such a network consists of multiple computers ... all connected through telephone lines, **or other communication links,**

⁵ Via their incorporation by Carmel, the disclosure of Carmel '432 and Carmel '230 is properly considered to be disclosure of Carmel itself. See *Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1329 (Fed. Cir. 2001). But even if Carmel was not considered to include all of the incorporated disclosure, a POSA would have found it obvious 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 across a computer network), and Carmel's express reference to and incorporation of the other two Carmel patents. Even if Carmel is not considered a single combined reference, Carmel in view of the teachings of Carmel '432 and/or Carmel '230 further renders obvious this wireless connection claim element.

both physical and wireless.” (EX1009, 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 similarly teaches examples of clients using wireless (*e.g.*, satellite) connections. (EX1010, 11:50-57.)

Further, if this claim element requires that multimedia objects are received by the multimedia player from a plurality of servers, Carmel, in view of the knowledge of a POSA, renders such an arrangement obvious. (Polish, ¶¶165-166.) Just as Carmel discloses multiple client computers 30, a POSA would have understood that the multimedia streaming system taught by Carmel could also 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 POSA would have been motivated to implement such an arrangement with multiple host servers 36 to provide a faster, more 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 multimedia player thereon) in the proper sequence “with only a minimal necessary transmission ... delay.” (Carmel, 10:48-63, 1:51-54, 3:36-39, 9:10-52, 10:64-11:22, 12:6-12; EX1051, 5:11-16, 11:24-27 (reducing delay by locating content on servers geographically close to

clients); *cf.* EX1060, 8:3-7.) Indeed, the receipt of multimedia data from multiple host servers (and benefits from doing so, including, *e.g.*, more stable and scalable streaming output, reducing latency) was known in the art. (*E.g.*, EX1051, 2:54-3:8, 4:48-67, 6:2-46, 7:11-16, 11:17-40, 12:50-61, 13:12-19, Figs. 2, 5a-5b; EX1052, 2:39-59, 6:29-33, 7:66-8:13, 13:21-62, Fig. 1; EX1053, [0007], [0014] (“In a further embodiment, **the client module 114 may be configured to retrieve streamlets 212 from a plurality of web servers 310[.]**”), Figs. 1-4; EX1050, Abstract, 3:20-26, 3:48-56, 11:63-12:2, Fig. 1; Polish, ¶166.)

Accordingly, Carmel discloses and renders obvious this claim element.
(Polish, ¶¶156-167.)

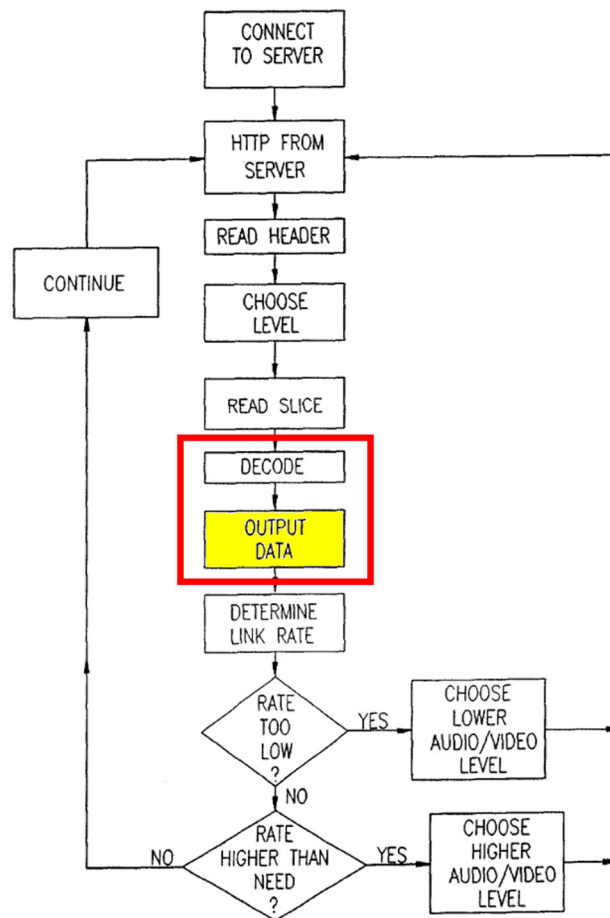
f. “playing back received multimedia objects by the multimedia player,” (claim 1[c.i])

Again, Carmel teaches that client computers 30 (and the multimedia players thereon) are configured to play back the sequence of multimedia objects that have been prepared and received according to the methods taught by Carmel. (*E.g.*, Carmel, 3:66-4:2, 7:12-17 (“[C]lients 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.”);⁶ *see also id.*, 2:15-17, 4:3-7, 5:22-25, 10:24-11:22; Polish, ¶¶169-171.) Carmel teaches that the slices of video and audio are “received, decoded and output by the client” “for the appreciation of a user.” (*Id.*, 11:7-8, 10:48-54. Fig. 6B.)

⁶ The ’849 patent indicates that exemplary multimedia players can be “java applets” or “embedded or downloaded non-java multimedia players[.]” (’849, 6:29-31.)

FIG. 6B



Carmel thus discloses and renders obvious this claim element. (Polish, ¶¶168-172.)

- 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])

Carmel further teaches configuring the multimedia player on client 30 to play the multimedia objects/slices in a sequence such that fluidity and video and audio quality are maintained by selecting a plurality of the multimedia objects in a manner

that reflects the available network bandwidth. (Polish, ¶¶173-174.)

Carmel repeatedly refers to the series of slices of the multimedia data stream as a “sequence” and teaches that the object parameters/metadata (including, *e.g.*, slice indices and timestamps) are used to ensure that the series of slices are synchronized and played in the correct sequential order. (*E.g.*, Carmel, 2:1-21 (“The clients use the slice indices of the frames to maintain proper synchronization of the playback.”), 3:24-5:35 (repeated use of “sequence” referring to the series of data slices), 7:4-8, 7:18-31 (“Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T₁, T₂, T₃, 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.”), 8:42-9:9, 12:6-12, Figs. 3A, 3D, 6B; Polish ¶¶175-176.)

Carmel also teaches configuring the multimedia player to play the multimedia objects in a sequence such that fluidity or playback continuity and video and audio quality are maintained. For example, Carmel teaches:

[C]lient 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**

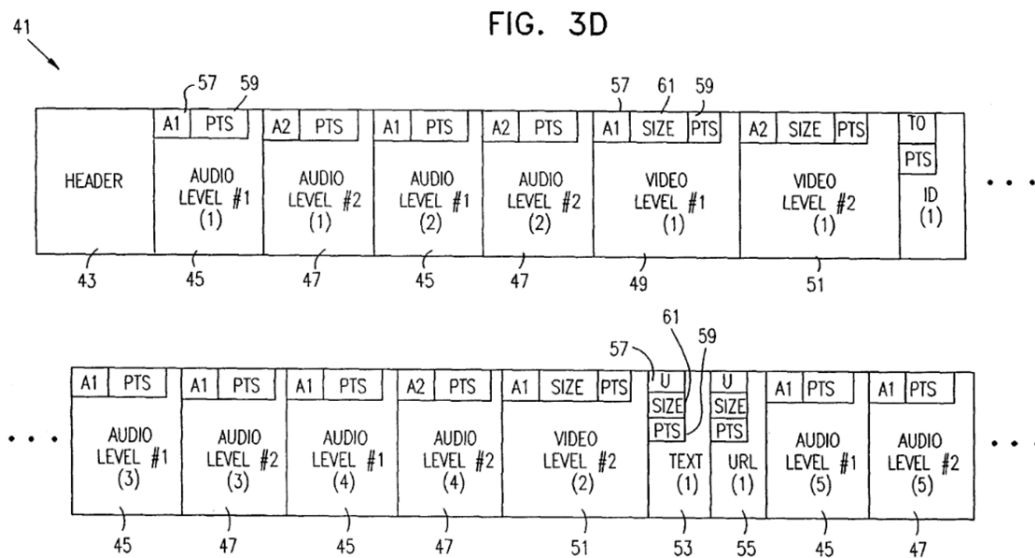
just as it was input at computer 34, preferably with only a minimal necessary transmission and decoding delay.

(Carmel, 10:48-54; *see also id.*, 12:6-12 (describing “proper sequence” of files and minimizing the impact on fluidity/continuity from “drop-out”), 13:29-35 (teachings relating to upload are applicable to download); Polish, ¶177.)

In addition to these teachings, the goal of maintaining playback fluidity/continuity and the highest possible or practicable audio and video playback quality (in view of resource constraints) when streaming multimedia content was well-known and something that those in the field generally sought to achieve. (Polish, ¶178 (citing, *e.g.*, Mattavelli, 2309 (providing highest playback quality practicable while “respecting all real-time constraints”), 2310-12; EX1012, 2:20-22, 4:11-5:8; EX1024, 1:8-24; EX1026, 1:8-19; EX1020, 199, 209-10, 213-15, 218-20; EX1021, 294-95 (scalable video coding)).) A POSA would have found it obvious and would have been motivated to achieve this fundamental streaming goal of maintaining fluidity and the highest possible/practicable quality multimedia playback. (*Id.*)

Carmel also teaches configuring the multimedia player to maintain playback fluidity and quality by selecting a plurality of multimedia objects in a manner that reflects available network bandwidth. For example, as discussed, in Carmel’s multi-level data stream embodiment, “[e]ach time slice in stream 41 includes multimedia

data at multiple quality levels.” (Carmel, 8:42-9:9, Fig. 3D.) Carmel provides: “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.” (*Id.*) Carmel teaches that “[e]ach of clients 30 chooses or is assigned the quality level appropriate to the bandwidth of its link on network 28 to server 36.” (*Id.*)

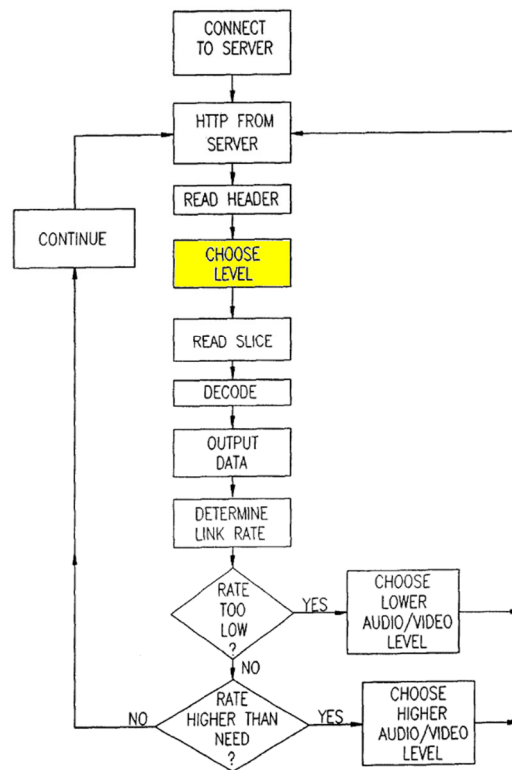


(Carmel, Fig. 3D.)

In connection with Figure 6B, Carmel again teaches that the client 30 selects an initial quality level for the audio and video slices based on the initial available network bandwidth. (Carmel, 10:64-11:9 (“**Responsive to the selection [by client**

30], server 36 begins to transmit data slices **at the chosen quality level**. The slices are received, decoded and output by the client.”), Fig. 6B; Polish, ¶¶179-180.)⁷

FIG. 6B



In its infringement allegations in the related litigation, PO alleges that this same type of functionality satisfies this claim element. (EX1055, 50-52; Polish, ¶181.)

And, Carmel generally teaches and repeatedly emphasizes seeking to match

⁷ Carmel’s teachings regarding subsequent adjustments to this selection based on changes to available bandwidth are discussed below for element 1[c.iii]. (Polish, ¶182.)

the bitrate/data rate of the multimedia slices with the available network bandwidth for download (and upload) so as to avoid lag and maintain fluidity of the multimedia playback at the best possible/practicable quality. (*E.g.*, Carmel, Abstract, 2:1-21, 2:56-59, 3:24-39, 3:55-59, 3:66-4:2, 4:20-24, 4:63-5:28, 7:35-41 (“[C]lients 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[.]”), 9:10-51, 10:19-23, 10:48-63, 11:39-52, 11:53-12:17, 13:23-35 (teachings regarding upload apply to download), Figs. 3A, 4-5, 7-9; Polish, ¶¶183-184.)

Carmel thereby discloses and renders obvious this claim element. (Polish, ¶¶173-185.)

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

Carmel teaches autonomously adjusting the selection of multimedia objects (*i.e.*, 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. As discussed above for element 1[b.ii] and incorporated here, the audio and video slices—whether stored within a single file or as separate files—are associated with metadata or object parameters and host path

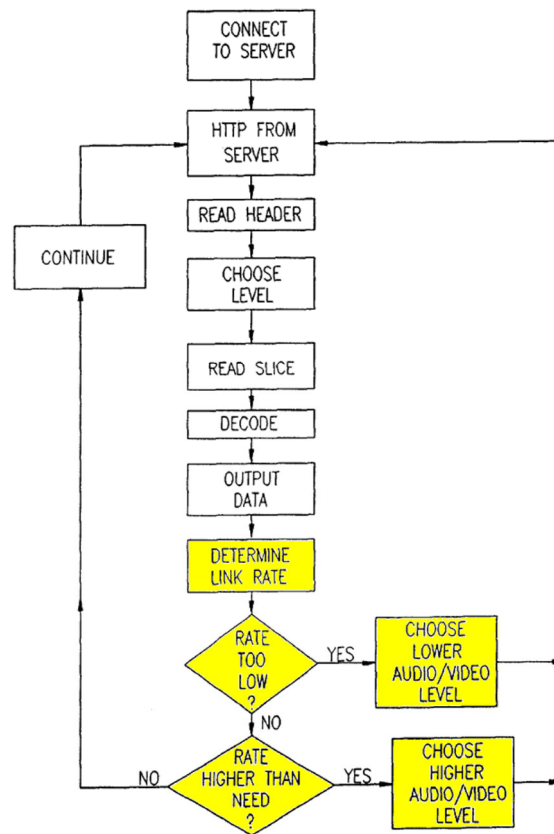
identification, including, *e.g.*, slice indices/identifiers, one or more timestamps, and a size/quality level parameter. (Carmel, 8:42-9:9, Fig. 3D; Polish, ¶¶186-188.)

In its multi-level streaming embodiment, Carmel teaches how the client 30/media player is configured, after an initial selection of the quality level, to periodically reassess the available network bandwidth and data transfer and adjust the quality level if needed (or beneficial), providing:

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.

(Carmel, 10:64-11:22, Fig. 6B; Polish, ¶189.)

FIG. 6B



Carmel teaches that this assessment of whether the network bandwidth or data transfer rate is insufficient for (or sufficiently in excess of) the currently selected bitrate/quality level of the multimedia slices is performed by the client via the use of object parameters, including, in particular, the “time stamps 59” and slice size/quality parameters. (Carmel, 11:11-18.) This disclosure further teaches and suggests that the client/multimedia player is configured 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

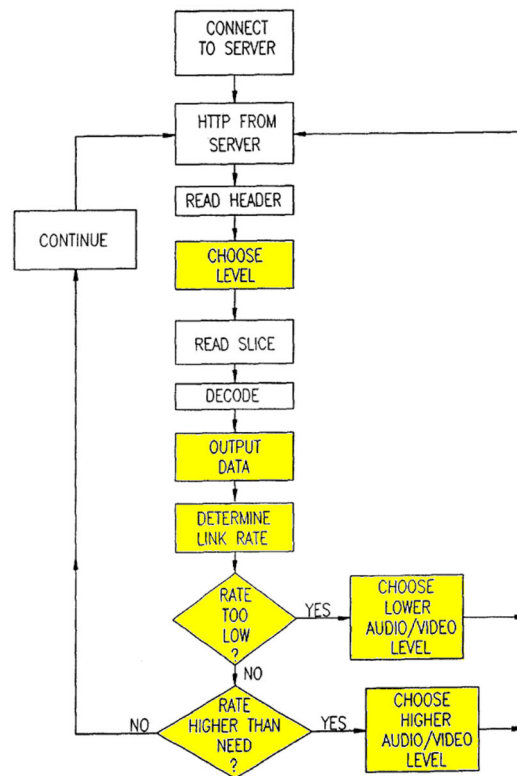
player will receive, decode and output/play. (Carmel, 10:64-11:22, Fig. 6B; Polish, ¶¶190-191.)

Additional Carmel teachings in accord with those concerning the multi-level streaming embodiment, also show how slice timestamps (*i.e.*, object parameters) are used to assess whether the available network bandwidth is sufficient to keep up with the multimedia playback and maintain fluidity and quality. (*E.g.*, Carmel, 7:28-49 (“[C]lients 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” and take steps to adjust the data reception rate and/or vary the compression levels “to adjust the data streaming rate to the available bandwidth”), 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.”), 10:64-11:22, 9:49-52; Polish, ¶¶192-193.)

Finally, Carmel also teaches that this adjustment to said selection and playback is performed autonomously by the client computer 30. Carmel teaches that the client computer 30 (controlled by the multimedia player) is the entity that chooses, selects, and adjusts (when needed or beneficial) the quality level of the multimedia slices that it downloads and receives for playback in a manner that

reflects the available network bandwidth. (*E.g.*, Carmel, 11:1-8, 11:9-15, 10:27-31, 10:64-67, Fig. 6B.)

FIG. 6B



Carmel thus teaches and suggests that the client/multimedia player “autonomously”—that is, without server-side direction or intervention—adjusts the selection and playback of the multimedia objects in view of the available network bandwidth. (Polish, ¶¶194-196.)

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 by the client/decoder/multimedia player according to multimedia object parameters. Mattavelli teaches techniques where the decoder/media player uses object parameters (including, *e.g.*, decoding complexity metadata that is available with the MPEG-4 standard) to determine when and by how much to adjust the decoding/playback of received video segments. (*See* Mattavelli at 2309, 2311-12; Polish, ¶197.)

Carmel, and Carmel in view of Mattavelli, thus teach and render obvious this claim element. (Polish, ¶¶186-198.) Indeed, Carmel’s teachings regarding such autonomous, client-directed feedback and adjustment mechanisms of the bitrate/quality level in view of the available network bandwidth are in accord with similar techniques that were known in the art. (*Id.*, ¶195 (citing EX1019, 228-29).)

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

Carmel in view of Mattavelli teaches and renders obvious configuring a Carmel’s multimedia player to utilize optimized decoding processes to play multimedia objects in a sequence such that fluidity and video and audio quality playback are maintained.⁸ (Polish, ¶¶199-200.)

⁸ While the precise bounds of this claim term (and similar claim terms such as claim 3’s recitation of “optimal” encoding formats and rates) are not defined, Mattavelli,

Mattavelli provides “motivations and ideas of the Computational Graceful Degradation (CGD) approach for the implementation of video decoding.” (Mattavelli, 2309.) Mattavelli teaches CGD methods where degraded, less computationally intensive algorithms and processes for decoding compressed multimedia data are used during periods where the decoding processing needs 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. (Mattavelli, 2309 (“The main rationale ... is to provide results of poorer quality when the results of the desirable quality cannot be obtained in real time.”), 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.”).) Similar types of gracefully degrading decoding techniques were known in the art. (*E.g.*, EX1029, 373; Polish, ¶201.)

Wang, Wolters, and Carmel teach and render obvious the use of optimized decoding processes to maintain quality playback and/or optimal encoding formats and rates, falling within the core scope of these terms, as explained below.

Mattavelli teaches that there are several decoding processes that can be simplified to save computational resources when needed. (Mattavelli, 2309-10.) 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).” (*Id.*, 2310.)

Mattavelli teaches two strategies for determining when (and to what extent) such CGD techniques should be employed: the first monitors the execution of the decoding processes and the second uses complexity information carried in the frames or VOP headers, as can be done in MPEG-4 systems, and Mattavelli provides examples using each approach. (Mattavelli, 2310-12, Figs. 2-3.)

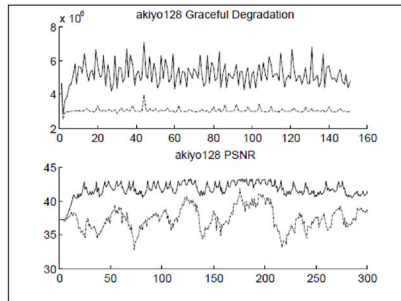


Figure 2: Example of CGD decoding of 300 frames of the sequence Akiyo at compressed at 128 kb/sec with H.263. The upper diagram reports the number of processor cycles for the original decoding (upper curve) and for the CGD decoding lower curve, versus the frame number. The lower diagram reports the two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

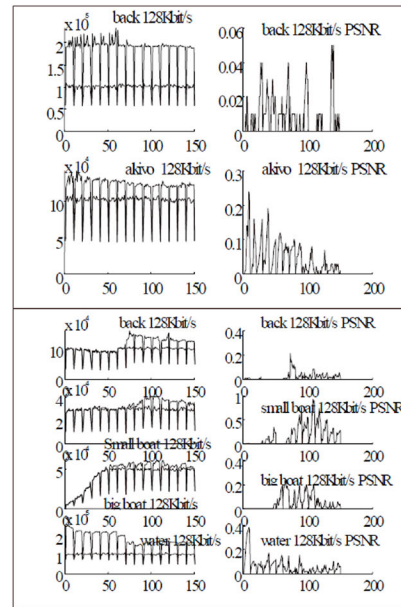


Figure 3: Left, decoding time results for the various VOP of sequences Akiyo and Coastguard, with and without imposing a bound. Right, the PSNR degradations versus the sequence VOP number.

Mattavelli thus teaches and renders obvious: (i) maintaining the quality and fluidity of playback (*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 using “means of many suitable **optimization method[s] or heuristics**” for determining when and by how much to apply these CGD techniques), including those that use object parameters (*e.g.*, decoding complexity prediction information that is included as part of the MPEG-4 standard). (Mattavelli, 2311-12, 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.”); Polish,

¶¶201-205.)

Carmel in view of Mattavelli thus render obvious this claim element. (Polish,
¶¶199-205, 208.)

Further, if “optimized decoding processes” is interpreted to mean that both optimized video *and audio* decoding processes must be employed, Mattavelli’s teachings further suggest and render obvious this limitation. This is so because “audio compression is built on many of the same principles as video compression” (EX1020, 322), and a POSA would have appreciated that the same CGD 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 overstretched. (*E.g.*, Mattavelli, 2311-12 (“[G]oal is to respect the real-time constraints with the lowest degradation in the video/**audio** content.”); EX1039, 1119-20, 1125 (teaching graceful degradation of audio decoding processes); EX1042, 379 (same); Polish, ¶206.) Further, optimized audio decoding processes were well-known, and a POSA would have been motivated to use such optimized processes, particularly in implementations involving wireless clients with more limited processing resources. (Polish, ¶207 (citing EX1039, 1119, 1122-25;

EX1040, 671 (teaching “optimization techniques to implement MPEG audio decoding algorithms” beneficial for a portable device), 672-75; EX1041, 1, 12-14; EX1042, 376-79; EX1043, 303-06; EX1047, 117-18, 120-22.)

Rationales for combining Carmel and Mattavelli’s teachings: A POSA would have found it obvious and would have been motivated to combine the teachings of Carmel and Mattavelli, predictably resulting in the multimedia streaming system taught by Carmel which uses Mattavelli’s CGD techniques to optimize the decoding of received slices of audio and video data for playback. A POSA 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, 11:27-28), to a recipient client device that uses Mattavelli’s CGD techniques for video decoding for media playback would have been successful. (*See, e.g.*, Mattavelli, 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.”); Polish, ¶209.)

Carmel and Mattavelli are both directed to processing multimedia data ultimately for playback “for the appreciation of a user” (*see* Carmel, 10:49-50), with both references teaching techniques for providing the highest perceived audio and

video output quality possible or practicable in view of system constraints, while ensuring playback continuity. (*E.g.*, Carmel, 10:48-54, 11:9-18, 7:35-49, Figs. 3D, 6B; Mattavelli, 2309, 2310.) Mattavelli teaches, for example, abrupt interruptions in media playback “are perceptually very annoying” and, accordingly, **“it is evident the importance of techniques”** that eliminate or reduce such undesirable, perceptually annoying outcomes. (Mattavelli, 2310; Polish, ¶210.)

Both references also discuss the compression of multimedia data according to well-known coding/compression standards for streaming.⁹ (*E.g.*, Carmel, 7:25-26, 11:24-52; Mattavelli, 2309, 2310-11.) Moreover, the incentives for and advantages of scalable multimedia encoding and decoding when streaming (as taught by Carmel

⁹ A POSA would have appreciated that using MPEG-4 compression standards (as taught and suggested by Carmel and Mattavelli (*see* Carmel, 11:23-51)) would have been particularly suitable in applications with a mobile client using a wireless connection (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 taught by Mattavelli at the decoder/mobile client. (*See, e.g.*, EX1021, 6-7, 10, 40-41, 77-79, 295; EX1044, 52; Polish, ¶211 & n.9.)

and Mattavelli) were well-known to those of skill in the art. (Polish, ¶211 (citing, *e.g.*, EX1028, 142-44; EX1020, 209-11, EX1044, 16; EX1062, 2:24-27, 2:40-3:15).)

A POSA would have further appreciated that Carmel and Mattavelli are complementary, focused on mitigating different (though related) bottlenecks or constraints when streaming multimedia data from an input/source to the ultimate user/consumer/player of the video/audio data. Carmel is primarily focused on constraints imposed by the available network bandwidth and teaches methods for addressing these constraints by matching the encoded bitrate/quality level to the bandwidth available for download by the client/player. (*E.g.*, Carmel, Abstract, 7:36-41, 9:6-9, 9:13-17, 10:64-11:22, 11:40-12:17, Figs. 3D, 6B.) Mattavelli is primarily focused on limitations on the processing resources available to the client/multimedia player when decoding a compressed bitstream and teaches optimized CGD decoding techniques that are progressively employed “when the required processing needs exceed the available processing resources.” (Mattavelli, 2309, 2310-11, 2312.)

A POSA 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 that addresses both bandwidth-based and client processing-based constraints together to provide an improved or best possible/practicable playback experience. (Polish, ¶¶212-215.)

Indeed, Mattavelli 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 a decoder, noting that certain bandwidth optimizations can sometimes make the decoding more computationally expensive. (Mattavelli, 2310.) This provides an express motivation to combine the complementary teachings of Carmel and Mattavelli to ensure that these different, related links in the streaming playback chain are optimized together. (Polish, ¶¶216-217.) Mattavelli teaches that its CGD techniques “enable more efficient usage of the hardware processing power” without needing “sustained over-sizing of processing resources,” thereby allowing, *e.g.*, a given client to handle decoding more complex, more efficient compression methods, which provide more efficient network bandwidth usage and ultimately improved/higher quality playback overall. (Mattavelli, 2309, 2312; Polish ¶217.) A POSA would have been motivated to combine Carmel and Mattavelli’s teachings to achieve these synergistic benefits. (*Id.*)

Carmel also contains express disclosure that would have motivated a POSA to combine its teachings with Mattavelli’s, teaching that playback of the multimedia sequence is achieved “preferably with only a minimal necessary transmission **and decoding delay.**” (Carmel, 10:50-54, 3:66-4:2, 5:22-25, 11:7-8.) Carmel teaches

techniques for minimizing transmission delays, and this teaching would have motivated a POSA to use Mattavelli's CGD decoding techniques with the multimedia player taught by Carmel to minimize any potential decoding delays, thereby maintaining quality and continuity. (Polish, ¶218.)

Further, a POSA would have appreciated that using Mattavelli's CGD decoding optimizations would have been particularly advantageous for the wireless client implementations taught and rendered obvious by Carmel. (Polish, ¶219.) This is so because wireless/mobile client devices were understood to generally have comparatively more processing constraints/limitations. (*See, e.g.*, EX1007, 51; EX1027, 658; EX1021, 295; Polish, ¶219.) But, more efficient compression methods that provide higher quality at a given amount of bandwidth (*e.g.*, MPEG-4, H.264/AVC) typically were/are more complex or computationally expensive to encode/decode. (Polish, ¶220; Carmel, 11:36-38, EX1061, 1:10-17; EX1007, 51.) A POSA thus would have been particularly motivated to use Mattavelli's CGD decoding optimizations to facilitate or enable the beneficial use of such higher-efficiency compression methods/codecs in implementations with processing power-constrained wireless clients. (Polish, ¶220; Mattavelli, 2309, 2312 (teaching its CGD techniques may enable the successful use of compression standards/levels/profiles that would otherwise have been "forbidden").)

Additionally, a POSA 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 a known issue (or a known set of issues, *i.e.*, constraints on both network and decoder processing bandwidth) to yield predictable results—*e.g.*, 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 playback of a multimedia data stream “for the appreciation of a user.” Similarly, a POSA 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) to a known device, method, or product ready for improvement (*e.g.*, the streaming method and system taught by Carmel) to yield these predictable results. (Polish, ¶¶221-222.)

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.”

As discussed above, Carmel teaches carrying out the transmission of multimedia objects using any one of multiple well-known Internet protocols, including HTTP and FTP. (*E.g.*, Carmel, 2:1-15, 2:22-28, 3:60-67, 5:19-28, 6:50-54, 7:4-9, 13:29-35, Fig. 2.) Carmel thereby discloses and renders obvious the use of protocols that can serve files in a directory, as HTTP and FTP are examples of

such protocols. (Polish, ¶¶224-227 (citing EX1006, 25 (directories on an FTP server), 39, 122-23 (directories on HTTP servers), 126-27, 200-01, 220-21, 229, 439-41; EX1023, 528-29; EX1022, 293-307, 317-18.)

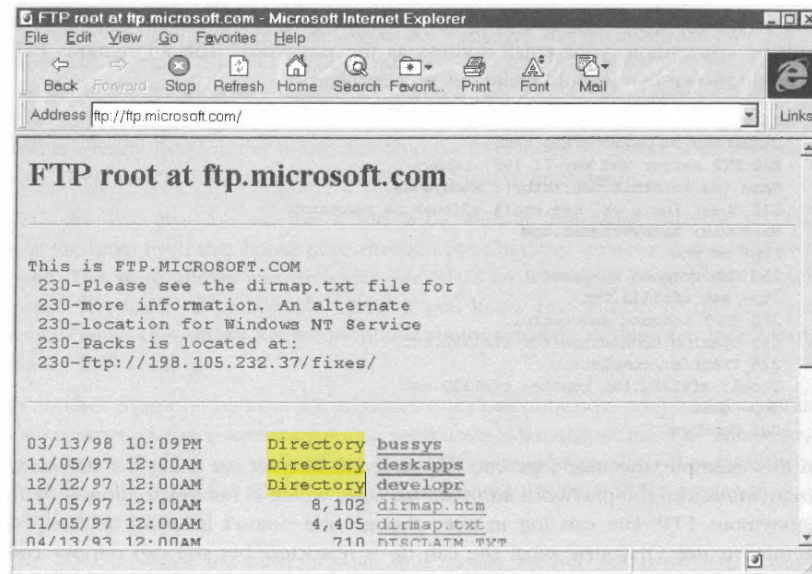


Figure 13-3. Using the Web for file transfer

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, 415; Polish, ¶¶223-227.)

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.”**

First, Carmel teaches the creation of multimedia objects from a “live multimedia stream,” providing, *e.g.*, that the source multimedia data “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, 6:57-60, 12:54-58 (referencing “live broadcast transmission”), 9:64-10:1 (broadcast of data from input devices 22), 6:24-35, 1:23-33, Fig. 2; Polish, ¶229.)

Next, Carmel teaches and suggests that the multimedia objects are transcoded into an optimal¹⁰ digital format at an optimal encoding rate reflecting available network bandwidth. Again, Carmel provides that the audio and video data is encoded “using any suitable compression method known in the art,” including, *e.g.*, MPEG-4-based standards, and teaches and renders obvious determining the (optimal) compression ratio/encoding rate based on available bandwidth. (Carmel, 2:42-50, 3:63-4:2, 4:39-47, 7:44-49 (“**[T]he compression level of the data is varied ... so as to adjust the data streaming rate to the available bandwidth ... between computer 34 and server 36, and/or between server 36 and client 30.**”), 11:23-52, 9:64-10:1, 10:19-23, 8:56-9:5 (multiple encoding rates/quality levels available for the client to select optimum level based on bandwidth), 9:46-52, 10:64-11:22, Figs. 3D, 5, 6B, 7; Polish, ¶¶230-231.)

¹⁰ See footnote 8, *supra*.

FIG. 5

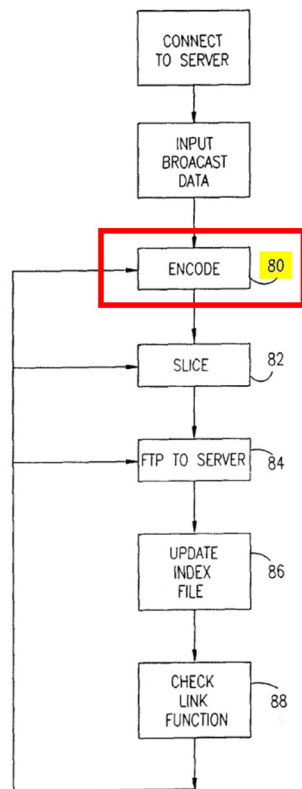
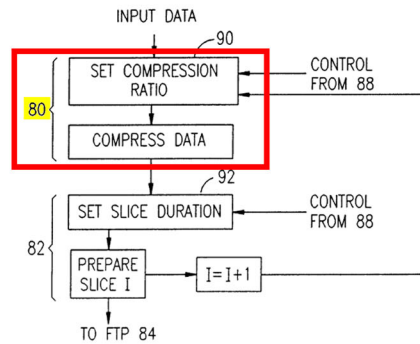


FIG. 7



And, a POSA would have appreciated that compression using MPEG-4 standards would have been an optimal digital format that was particularly well suited for mobile clients (as taught and suggested by Carmel), due to its compression efficiency, robustness to transmission errors, and inclusion of decoding complexity metadata. (Polish, ¶231 & n.10, ¶211.) MPEG-4-based compression is also presented as an optimal format by the '849 patent. (*Id.* (citing '849 patent at Abstract, 2:50-3:16 (source multimedia data is converted “into true MPEG4/AAC multimedia objects”), 7:57-8:7, Figs. 3a, 6).)

Carmel also teaches and suggests “transcoding” the source multimedia data via its disclosure regarding using suitable, well-known compression standards to encode the data stream. A POSA would have understood that the terms “encode” and “encoder” are often used broadly to refer to any process that “converts the information to a particular form or format.” (Polish, ¶232 (citing EX1011).) Thus, the terms “encoding” and “transcoding” were/are often used synonymously to mean any conversion of data from one form to another, as the ’849 appears to do. (*Id.* (citing EX1048, 1:16-20; ’849 patent, Abstract, 7:57-64).) Carmel’s references to encoding an input data stream (*e.g.*, from input devices 22) with a suitable compression method (*e.g.*, MPEG-4) would be understood to constitute “transcoding” this input stream by converting it from one data format to another. (*Id.*)

Even if a narrower understanding of “transcoding” is used, however, that requires a compressed input stream that is converted from one type of compression to another, Carmel renders such a process obvious. Carmel’s teachings regarding receiving data from input devices 22 or using a prerecorded multimedia sequence would reasonably suggest to a POSA implementations or embodiments where the input data was compressed. (Polish, ¶¶233-235 (citing EX1012, 8:35-37 (“Typically, the input video will have been compressed using a conventional video compression

process.”), 8:16-29; EX1013, [0025]; EX1021, 2; EX1005, 3-4, EX1020, 3-4).)

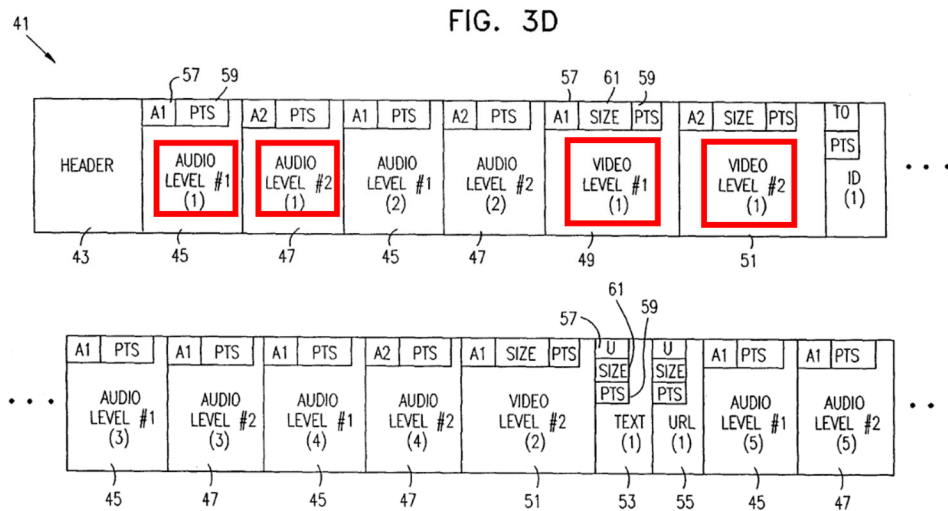
Further, a POSA would have understood that compressed input multimedia data should and would first be decompressed (at least to some extent) before recompressing it with a different optimal codec at an optimal encoding rate for streaming, as taught by Carmel. (*Id.* (citing, *e.g.*, EX1012, 8:37-41, 8:48-59, Fig. 2; EX1013, [0025]; EX1049, 2:6-50; EX1058, 2:1-31; EX1059, 2:1-6; EX1061, 1:31-39).) Carmel thus teaches, suggests, and renders obvious “transcoding” under either understanding of this term.

And, for the reasons previously discussed for claim elements 1[a.i], 1[a.ii], and 1[b.ii] (and incorporated here), Carmel teaches and renders obvious “converting the input multimedia stream into audio and video segments by splitting the encoded stream into intervals for access by wireless clients,” and Carmel renders this claim obvious. (Polish, ¶¶228, 236-238.)

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.”

As discussed, Carmel teaches that source multimedia data (which can be from input devices 22 or a prerecorded sequence “stored on disk or tape”) are compressed “using any suitable compression method known in the art,” including, *e.g.*, MPEG-4, and is split into a sequence of slices of audio and video data. (Carmel, 9:62-67,

7:18-34, 8:42-9:5, 9:62-10:1, 11:23-52, Figs. 3A, 3D, 5.)



Carmel, combined with the knowledge that many of the suitable compression methods known the art separately encode audio and video streams (*e.g.*, EX1020, 216), thus teaches and renders obvious splitting a multimedia stream into separate (temporally corresponding) video and audio segments that are recombined at the decoder/multimedia player. (Polish, ¶¶239-240.)

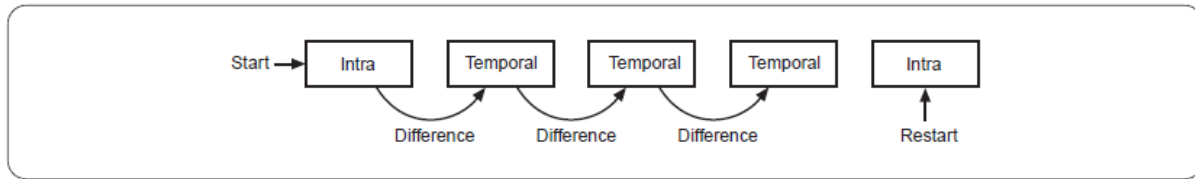
Further, Carmel (alone and in view of Mattavelli) also teaches and renders obvious scanning the video portion and splitting the file/stream such that each video slice starts with an I-frame. Carmel's exemplary suitable video coding formats (*e.g.*, H.263, MPEG-4) used/use both intra-coding (which exploits spatial redundancy in the image data within a particular frame) and differential or inter-coding (which exploits temporal redundancies in the image data between multiple successive frames) to compress video data. (Polish, ¶¶54-56, 241-242.) Sets of coded frames

were/are grouped as a Group of Pictures (GOP) where “each GOP begins with an independently encoded (intra) I[-]frame and may include one or more following [differentially-coded/predicted] P-frames and B-frames. Each I-frame can be decoded without information from any preceding and/or following frame.” (Polish, ¶242 (citing EX1016, 1:34-47; EX1005, 15-16, 248-57; EX1014, 15; EX1017, 1:34-47; EX1018, 237).)

When using such compression methods, a POSA would have been motivated and found it obvious to scan and start each video segment “slice” with an I-frame (*i.e.*, the starting point of a GOP) to ensure that the slice is able to be decoded properly from the very first frame, which would not be guaranteed if a slice began at any arbitrary point/frame. A POSA would have appreciated that if a slice were to start with a series of predicted frames (P- and B-frames), there would have been a significant risk that these frames would not be decoded properly because they lacked the reference I-frame on which their encoded difference data was based. (Polish, ¶243.)

This issue was well-known and discussed in the art. For example, Mattavelli teaches “as can be easily understood,” missing a reference frame “can have catastrophic consequences for prediction[-]based coding systems. ... 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.” (Mattavelli, 2309-10; *see also* Polish, ¶244 (citing, *e.g.*, EX1005, 248; EX1016, 2:11-19 (“cutting into a stream after an I frame renders the P and B frames meaningless.”); EX1045, 4:61-5:8; EX1014, 13, Fig. 2-10).)



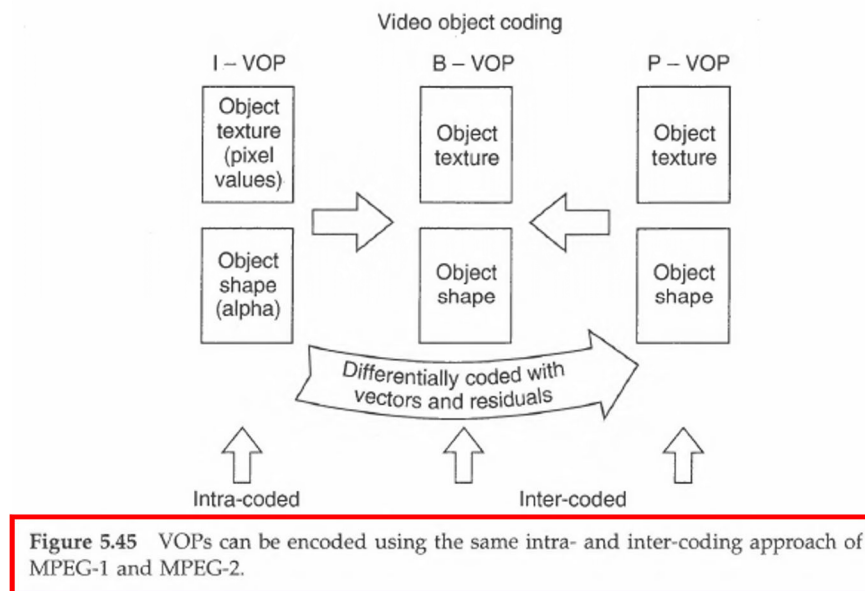
► Figure 2-10.

A POSA implementing Carmel’s video slices would have found it obvious to avoid this known issue (potentially arising, *e.g.*, in circumstances where the previous slice has been delayed or dropped in transmission (*see* Carmel, 4:34-38, 10:13-17, 12:10-12, 13:21-35)) by scanning the encoded video stream and splitting/starting each slice with an I-frame rather than slicing the video at an arbitrary location. (Polish, ¶245 (citing, *e.g.*, EX1005, 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, 255; EX1045, 5:58-64 (describing scan system that identifies where I-frames are located/stored in a video stream); EX1046, 1:29-35, 1:43-2:2, 4:54-5:18, Fig. 3).) Indeed, this technique was known in the art and used to avoid/mitigate this potential missing reference frame issue. (Polish, ¶¶246-247 (citing EX1016, 2:11-19, 2:33-50, 12:7-14, 14:40-45, 22:53-65; EX1019, 229;

EX1012, 16:18-21; EX1045, 13:44-50; *cf.* EX1021, 260-61).)

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.”

As discussed, Carmel teaches and suggests the use of MPEG-4 standards as a suitable compression method known in the art. (*E.g.*, Carmel, 11:23-52; Polish, ¶¶248-249, 231, 211.) And, a feature of MPEG-4 video encoding is the ability to separately encode multiple visual objects within a scene as separate video object planes (VOPs) that are later composited by the decoder. (Polish, ¶¶250-251 (citing EX1005, 276-92; EX1020, 197; EX1021, 297).) These VOPs were/are encoded using the same type of intra- and inter- (or predictive) coding methods used in other compression standards. (*E.g.*, EX1005, 277, 279, Fig. 5.45; EX1020, 198-210.)



Carmel further teaches and suggests splitting these different video object streams into separate slices (like any other part of a multimedia data stream). (Carmel, 8:42-55, Fig. 3D; Polish, ¶252.)

And for the reasons discussed for claim 4 (and incorporated here), a POSA would have recognized the risks of decoding errors arising from a missing reference frame/VOP and understood, motivated to use, and found obvious that one solution for avoiding this risk was splitting the video stream and starting each slice with an I-frame/I-VOP. (Polish, ¶¶253-255; *see also* EX1020, 201; EX1021, 297-98, 304 (display order of group of VOPs begins with I-VOP); EX1045, 5:58-64 (describing scan system that identifies where I-frames are stored in a video stream); EX1046, 1:29-35, 1:43-2:2, 4:54-5:18, Fig. 3.)

6. Claim 6 (Independent)

For the purposes of this petition, claim 6 is materially the same as claim 1 except that it is written as a claim to a device (*i.e.*, multimedia player) that includes computer executable instructions for performing a method instead of a method. (*See* Polish, ¶¶256-257 (providing side-by-side comparison); EX1057.) Claim 6 recites materially the same basic functions as claim 1. (*Id.*)

As discussed for claim 1, Carmel discloses and renders obvious a multimedia player—*e.g.*, “a Java applet or plug-in to the client’s web browser”—that decodes and plays back the sequence of multimedia slices “for the appreciation of a user.”

(Carmel, 2:15-17, 3:66-4:2, 4:3-7, 5:22-25, 7:12-17, 10:24-11:22.) Carmel further teaches and suggests that this multimedia player comprises “computer executable instructions stored on a non-transitory computer readable medium which when executed by a processor.” (*See id.*) Indeed, Carmel included as part of its disclosure a sample multimedia player applet consisting of computer executable instructions. (*E.g., id.*, 10:34-35, 13:54-14:16.)

For these reasons and those provided in claim 1, Carmel in view of Mattavelli renders obvious claim 6. (Polish, ¶¶256-259.)

7. Claim 7 (Independent)

For the purposes of this petition, claim 7 is materially the same as claim 1 except that it is written as a system for performing a method instead of a method. (*See* Polish, ¶¶260-261 (providing side-by-side comparison); EX1057.) Claim 7 recites materially the same basic functions as claim 1. (*Id.*)

Further, Carmel teaches and renders obvious a system for performing the methods discussed in claim 7. (*E.g.,* Carmel, 6:24-49 (Fig. 2 illustrates “**system 32** for remote broadcasting of a multimedia sequence”).) Carmel also teaches a system that includes “at least one centralized content server transmitting multimedia objects”—*e.g., network server 36*. (*Id.*, 6:24-49, 7:4-5, Fig. 2.)

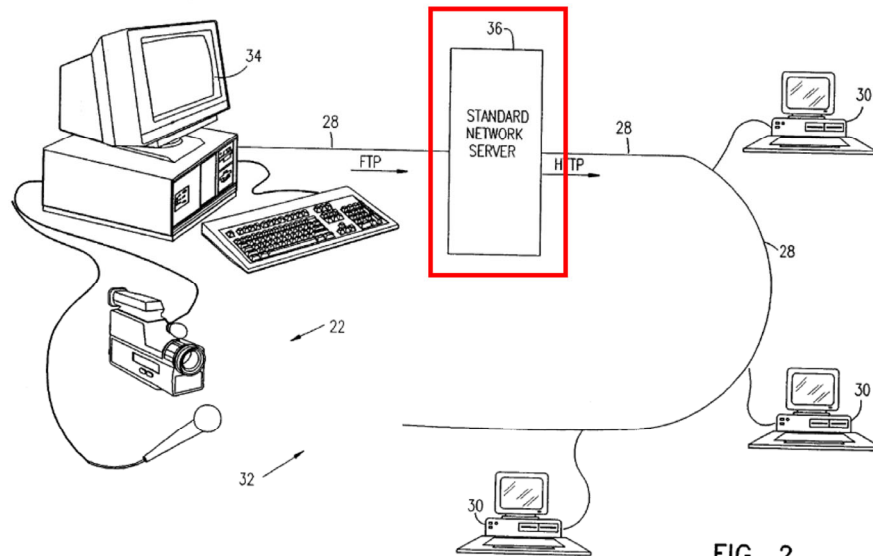


FIG. 2

For these reasons and those provided in claim 1, Carmel in view of Mattavelli renders obvious claim 7. (Polish, ¶¶260-264.)

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.”**

Carmel teaches the use of “existing multimedia files” as an input multimedia stream that is encoded/transcoded and sliced by computer 34 to create multimedia objects. Carmel provides, *e.g.*: “[t]he sequence is preferably generated and compressed in real time ... **although a prerecorded sequence may similarly be broadcast in this manner.**” (Carmel, 6:50-60.) Carmel also teaches: “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. (*Id.*, 9:62-10:1, Fig. 5.) Carmel thus satisfies this claim element. (Polish, ¶266.)

Outside of the “existing multimedia files” limitation, the rest of claim 8 is materially the same as claim 3, and it is rendered obvious by Carmel and Mattavelli for the same reasons provided above for claims 1 and 3. (Polish, ¶¶265-269.)

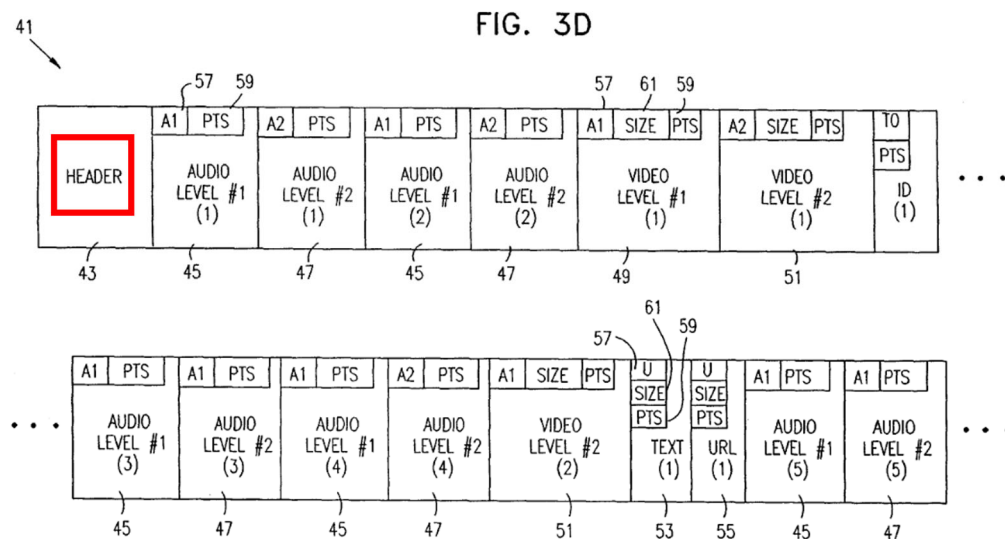
9. **Claim 9: “The method according to claim 1, wherein the object parameters and supplied identification of the first player received multimedia object define a sequence of similar multimedia objects and configure the player accordingly, using object parameters and video decoding optimization, such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

As discussed for claim 1 (and incorporated here), Carmel (alone and in view of Mattavelli) teaches and renders obvious multimedia objects having object parameters and supplied host path identification and utilizing object parameters and decoding optimizations to configure the multimedia player to maintain playback fluidity and the best possible/practicable video and audio quality in view of the network and client resources/constraints and multimedia stream characteristics. (Polish, ¶¶270-271.)

Carmel further teaches and renders obvious providing object parameters and metadata with the first object (*e.g.*, an index file) that the client computer/media player reads when it begins to download the data stream. (*See* Carmel, 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.”.) Carmel further teaches that this first file/message “contains parameters relating generally to the data stream and/or instructions to computers 30.” (Carmel, 8:12-17, Fig. 3B.)

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. (Carmel, 8:51-55, Fig. 3D.)



Carmel thus teaches and renders obvious having the first received multimedia object received by the multimedia player include (e.g., in the header of the first multimedia object) information about and that “define[s] a sequence of similar multimedia objects [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). (Polish, ¶¶272-274.) And, due to their similarities in their description and purpose, a POSA would have understood and found it obvious that the configuration parameters and instructions contained in an index file could additionally or alternatively be included in the header of the first multimedia object containing multimedia data. (Polish, ¶275; Carmel, 8:9-11, Fig. 3D; EX1024, 2:16-3:4; EX1025, [0017].)

It further would have been logical and obvious to include any necessary or desired configuration metadata/parameters or instructions to clients 30 globally relating to the data stream or sequence of multimedia objects with the *first* object received and read by the multimedia player on client 30. (Polish, ¶275; EX1025, [0018] (“As media decoding and playback are typically not possible without file-level metadata, 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

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 corresponding elements recited in independent claims 6 and 7) and dependent claims 3, 4, 5, 8 and 9. In light of the analysis below of these limitations and Ground 1’s analysis of the remaining elements and claims, claims 1-9 are obvious based on Carmel in view of Wang. (Polish, ¶276.)

Claim element 1[a.i] recites (in part) encoding audio and video segments “in a digital encoding format.” As discussed, Carmel discloses compressing/encoding the multimedia data stream 40 “using any suitable compression method known in the art”, including, *e.g.*, MPEG standards. (Carmel, 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) codec. (Wang [EX1007], 51.) The MPEG-4 AVC coding/compression format would have been viewed as particularly well-suited (and an example of “an optimal digital format,” discussed below) for multimedia streaming implementations involving wireless transmission to a mobile client (as taught and suggested by both Carmel and Wang) due to its high compression efficiency. (Wang, 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, [AVC] provides better image quality due to its significant improvement on the coding efficiency.”); EX1027, 657.) Accordingly, Carmel in view of Wang renders obvious encoding audio and video segments “in a digital encoding format”—including, in particular, the AVC format. (Polish, ¶277.)

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, Carmel teaches and renders obvious the use of a wireless connection by

a client 30 to receive the multimedia objects from server 36. (*See* Ground 1, element 1[b.ii].) Wang also teaches and suggests a multimedia player/decoder receiving compressed multimedia data via a wireless connection. (*E.g.*, Wang, 51, 52 (emulating proposed AVC decoder architecture on an ARM development board because “the ARM platform is one of the most typical portable devices”).) Accordingly, Carmel in view of Wang also renders obvious having a multimedia player receive multimedia objects from a server “using a wireless connection.” (Polish, ¶278.)

Next, claim element 1[c.iv] recites: “utilizing optimized decoding processes to maintain quality playback.” Wang teaches a “platform-based MPEG-4 Advanced Video Coding (AVC) decoder” 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, 51.) Wang provides that to address constraints from “the low computation power and small memory for mobile devices” it adopts a “platform[-] based design method to increase overall throughput.” (*Id.*) Wang repeatedly states that its proposed decoder has “optimized” the decoding processes that it uses to achieve this goal of increased throughput/decoding speed of the AVC codec. (*E.g.*, Wang 51 (“[W]e **optimize the decoder software** by enhancing the

frame buffer management, boundary padding, and content aware inverse transform.”), 52 (describing “**an optimized AVC decoder**”), 53-54 (“**Software Optimization**” section 4.3).) Accordingly, Carmel in view of Wang teaches and renders obvious utilizing optimized decoding processes to maintain quality playback. (Polish, ¶¶279-281.)

Based on these same teachings, 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. (Polish, ¶282.)

And, for similar reasons discussed for claim element 1[c.iv] in Ground 1 above, to the extent that “optimized decoding processes” is interpreted to mean that optimized processes for decoding both video *and audio* must be employed, Wang’s teachings further suggest and render obvious configuring a multimedia player, particularly one running on a computationally-constrained mobile device, to utilize optimized audio decoding processes as well. (Polish, ¶283; EX1020, 322.) Numerous optimized audio decoding methods were known in the art, including optimized processes/software for decoding high efficiency audio coding standards. (Polish, ¶¶283-285 (citing, *e.g.*, EX1039, 1119, 1122-25); EX1040, 671-75, EX1042, 376-79; EX1043, 303-06; EX1047, 117-18, 120-23; EX1041, 1, 11-12, 13); *see also* Ground 3, below.)

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.” Wang also teaches and suggests these elements, teaching that MPEG-4 AVC is a particularly suitable/optimal compression format for mobile streaming applications on mobile/wireless devices, providing higher quality with less network bandwidth. (Wang, 51, 52; *see also* EX1027, 657.) And, 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, 51, 52, 53-55.) Accordingly, Carmel in view of Wang also renders obvious these elements of claims 3 and 8. (Polish, ¶286.)

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. A POSA would have been motivated to split a video stream so that each slice of video data started with an I-frame, as discussed above for Ground 1. Wang’s teachings regarding the use of the AVC standard further suggest and render obvious this claim element. (Polish, ¶¶287.)

The AVC compression standard codes pictures as one or more “slices” that can be I (Intra)-slices that contain only intra-coded macroblocks or predicted slices which contain macroblocks predicted from other reference pictures. (EX1028, 164.)

The MPEG-4 AVC standard's Extended Profile—which is “particularly useful for applications such as video streaming”—also introduced an “SI” slice type, which contains a special type of intra-coded macroblocks that facilitates switching between streams. (EX1028, 164, 216.) A POSA would have found it obvious that the multi-level streaming embodiment taught by Carmel, would have been advantageously implemented using SI slices provided under the Extended Profile of the AVC standard. (Polish, ¶¶288-289 (citing EX1028, 216 (describing advantages of SI slices when switching between material coded at multiple bitrates)).)

Further, it would have been obvious to a POSA that when using AVC encoding (as taught by Wang), the first video picture in a multimedia object 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). (Polish, ¶290 (citing EX1028, 167 (IDR coded pictures are used to clear the contents of the reference picture buffer, where the “first picture in a coded video sequence is always and IDR picture”))).)

Accordingly, Carmel in view of Wang also renders obvious scanning and splitting a multimedia stream such that each multimedia object starts with an I-frame, as recited in claims 4 and 5. (Polish, ¶¶287-291.)

Rationale for combining Carmel and Wang's teachings: A POSA would

have found it obvious and would have been motivated to combine Carmel and Wang's teachings, predictably resulting in the multimedia transmission and playback system taught by Carmel using Wang's optimized AVC decoder to play back high-quality AVC-encoded multimedia data. A POSA would have understood that using Carmel's method of streaming video and audio data, compressed "using any suitable compression method known in the art" including, *e.g.*, 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. (*E.g.*, Wang, 51 ("Our experiments show that the throughput of the MPEG-4 reference decoder can be improved by 6 to 7 times."); Polish, ¶292.)

Carmel and Wang are both directed towards the processing of compressed/encoded multimedia data ultimately for playback "for the appreciation of a user" at the best possible/practicable quality in view of system constraints. (Carmel, 7:25-26, 11:24-52, 10:48-54, 11:9-18 7:35-49, Figs. 3D, 6B; Wang, 51, 52-55.) And, as discussed above, the AVC format/standard would have been viewed as particularly well-suited for multimedia streaming implementations involving wireless transmission to a mobile client due to its high compression efficiency. (*See* Wang, 51; EX1027, 657; EX1028, 164, 216.) A POSA would have been motivated

to use the AVC format/standard in applications with a wireless client—and Wang’s AVC decoding optimizations to facilitate/enable such usage—for this reason. (*See id.*; Polish, ¶¶292-294.)

A POSA would have further appreciated that Carmel and Wang are complementary, focusing on mitigating different (though related) bottlenecks in the sequence of transmission and processing of streamed multimedia data from the source to the ultimate user/consumer/player of the data. Carmel is primarily focused on mitigating constraints imposed by the available network bandwidth (*e.g.*, Carmel, 7:36-41, 9:6-9, 9:13-17, 10:64-11:22, 11:40-12:17, Fig. 6B), while Wang is focused on mitigating limited processing resources on a mobile client/AVC decoder. (*E.g.*, Wang, 51, 52-55; *see also* EX1027, 628.) A POSA would have been motivated to combine the complementary teachings of Carmel and Wang to form a streaming system that is more robust, capable of using AVC-encoding, and able to handle both of these potential bottlenecks when streaming AVC-encoded data to a mobile device. (Polish, ¶¶295-298.)

Also, as discussed in Ground 1, Carmel’s teachings acknowledge potential decoding delays, and these express teachings would motivate a POSA to look to the decoding optimizations taught by Wang to minimize any such delays/issues. (Carmel, 10:50-54; *see also id.*, 3:66-4:2, 5:22-25, 11:7-8; Polish, ¶299.)

Additionally, a POSA would have considered a system that incorporated the bandwidth optimization techniques taught by Carmel with the decoder optimization techniques taught by Wang to be an obvious combination of known methods to address a known set of related issues (*i.e.*, constraints on both network and decoding processing bandwidth) to yield predictable results—*i.e.*, a system where a mobile client multimedia player/decoder advantageously optimizes both the available network bandwidth (using the highly-efficient AVC standard) and the available decoding resources to facilitate/enable the decoding/playing/outputting AVC-encoded multimedia data on a mobile wireless device. Similarly, a POSA would have found the combined teachings of Carmel and Wang an obvious application of a known technique (*e.g.*, the optimized AVC decoding techniques taught by Wang) to a known device/method/system (*e.g.*, the streaming method and system taught by Carmel) to yield these predictable results. (Polish, ¶300.)

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

Ground 3 incorporates the analysis in Grounds 1 and 2 of Carmel and Wang’s teachings. Ground 3 adds Wolters’s [EX1041] teachings for claim element 1[c.iv], reciting “optimized decoding processes.”

Wolters teaches that High-Efficiency MPEG Advanced Audio Coding (HE AAC) 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, 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 with certain profiles. (EX1041, 12; *see also id.*, 11, 13.)

Wolters also teaches that “optimized source code” was available for license and “[o]ptimized binary implementations” of audio decoders of the HE AAC standard were known and available. (EX1041, 12.) Thus, to the extent that “optimized decoding processes” is interpreted to require use of both optimized video *and audio* decoding processes, this element is rendered obvious by the combined teachings of Carmel, Wang, and Wolters. (Polish, ¶¶301-303.)

A POSA would have been motivated to combine Wolter’s teachings regarding use of HE AAC—and optimized processes for decoding HE AAC data—with those of Carmel and Wang. A POSA would have been motivated to use HE AAC audio coding in Carmel’s streaming for the same reasons a POSA would be motivated to use AVC video coding—*e.g.*, because it provides high compression efficiency allowing transmission of high-quality audio using less data. Indeed, Wolters provides an explicit teaching suggesting such a combination. (EX1041, 13 (“By

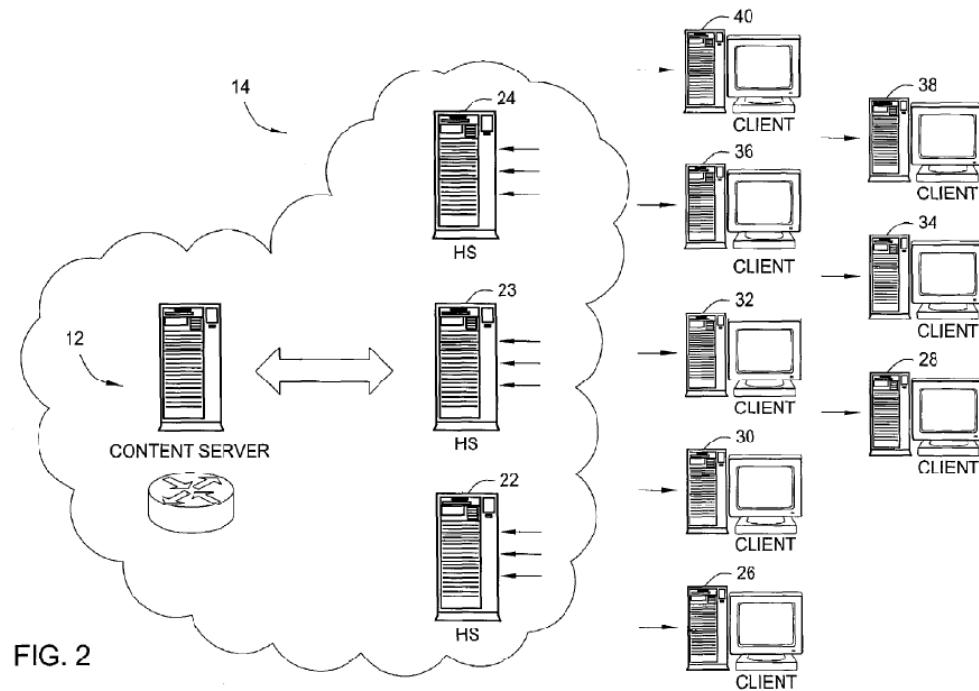
coupling HE AAC with MPEG-4 Video, more bits can be allocated to the video signal without degrading the quality of the audio signal. ... Combined with the new [AVC standard], even more significant gains in quality are possible.”); Polish, ¶304.) And, by employing optimized decoding processes, a POSA would maximize the available decoding processing resources to enable use of this standard and deliver the highest quality possible/practicable. (*Id.*)

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

Ground 4 incorporates the analysis of Carmel, Wang, and Wolters in Grounds 1-3 and adds Guo’s [EX1051] teachings on the use of multiple servers to serve multimedia segments ((possibly) recited in claim element 1[b.ii]), 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).

First, Guo teaches systems that use multiple “helper servers” (HSs) “to better support streaming media storage and distribution.” (EX1051, Abstract, 2:54-3:8, Fig. 2.) Guo teaches that a multimedia stream is divided into smaller chunks/segments 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.*, 6:24-46 (“A SM [streaming multimedia] 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; see also *id.* 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, Figs. 2, 5a-5b.)



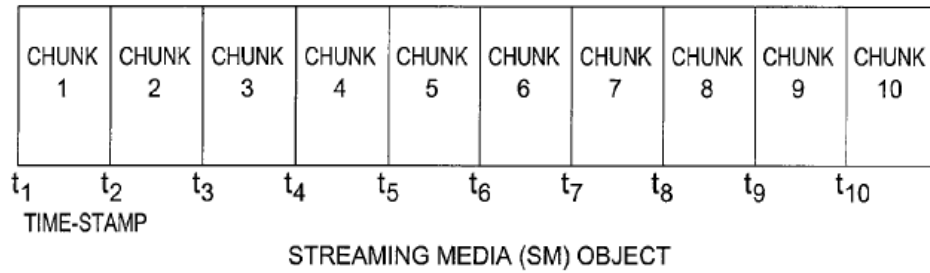


FIG. 5a

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

Thus, to the extent that claim element 1[b.ii] (and corresponding elements 6[b.ii] and 7[b.ii]) requires that multimedia objects are received by the multimedia player from a plurality of servers, Carmel teachings in view of Guo's render such an arrangement obvious. (Polish, ¶306.)

A POSA would have appreciated that Carmel's streaming system using multiple host servers (as taught by Guo) would have been faster, more robust, adaptable, and better able to ensure that sufficient bandwidth/download rates are maintained and that the series of multimedia objects are received by a client 30/media player in the proper sequence "with only a minimal necessary transmission

... delay.” (Carmel, 10:48-63; 1:51-54, 3:36-39, 9:10-52, 10:64-11:22, 12:6-12.)

Indeed, Guo provides express motivation to combine, teaching that its use of multiple servers helps mitigate several challenges associated with serving multimedia data, including reducing server load, congestion, and transmission and playback delays/latency. (EX1051, 3:1-8, 2:34-50, 5:11-16, 11:24-27 (reducing delay by locating content on servers geographically close to clients); Polish, ¶¶307-309.)

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 to achieve predictable results. Indeed, other systems that employed multiple servers to improve multimedia streaming across networks were known in the art. (Polish, ¶310.)

Next, Yu teaches a transcoder and examples where the source/input multimedia data (whether live or pre-recorded) is previously compressed using an older or different codec/compression standard and where such data is first decompressed before it is re-encoded in a desired/optimal format for streaming. (E.g., EX1061, Abstract, 1:26-39, 1:59-62, 2:16-20, 2:44-49, Fig. 1.) 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. (Polish, ¶311.)

A POSA would have been motivated to combine Yu's teachings 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, 6:32-35, 6:57-60)) often would be compressed in some manner (as taught and suggested by Yu). And, Yu teaches that its transcoder can beneficially extract statistics when decoding the source data that are helpful when re-encoding the data with a desired (typically more sophisticated or efficient) codec, making dual-pass encoding more efficient/less costly. (EX1061, 1:6-52, 1:56-2:23, 3:23-26, 9:7-19.) Further, it was (and still is) generally known, routine practice—and beneficial in avoiding potential(/likely) errors or issues—to first decode compressed data in some manner before re-encoding it in a desired/optimal format at a desired/optimal rate. Thus, 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 with another known element/process to obtain predictable results. (Polish, ¶312.)

Next, Forecast teaches and suggests that when splitting 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, 14:41-45, 12:7-14, 22:53-65.) Forecast explains “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 also teaches that a requirement for starting a new sequence with an I-frame was known and incorporated into a SMPTE standard. (*Id.*, 2:33-50.) These teachings, combined with Carmel’s teachings regarding splitting a media stream into slices, suggest and render obvious scanning and splitting a video stream such that each slice/segment/composite stream begins with an I-frame, as recited by claims 4 & 5. (Polish, ¶1313.)

A POSA would have found it obvious to combine the teachings of Carmel and Forecast in such a manner because a POSA would have understood that the same decoding problems and errors resulting from missing reference frames that would arise when a new spliced-in segment started with predicted frames (EX1016, 2:11-16) would also arise when decoding/playing back the segments/slices split according to Carmel’s teachings, particularly if/when earlier slices were delayed or dropped during transmission. (*E.g.*, Carmel, 12:6-12.) A POSA would have been motivated to apply these teachings 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

known potential decoding issues/errors arising from a missing initial reference frame. (Polish, ¶314.)

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

Ground 5 incorporates the analysis of Carmel and Mattavelli in Ground 1 and adds Guo's teachings regarding the use of multiple servers ((possibly) element 1[b.ii]), Yu's teachings regarding transcoding (recited in claims 3 & 8) and Forecast's 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, and a POSA would have been motivated to combine their teachings with Carmel's (and Mattavelli's) for the reasons provided above in Ground 4. (Polish, ¶315.)

VIII. SECONDARY CONSIDERATIONS

PO has not identified any evidence of secondary indicia of non-obviousness for the '849 patent. Accordingly, there is no objective evidence of non-obviousness that might warrant a finding that the petitioned claims are patentable. (Polish, ¶316.)

IX. CONCLUSION

Petitioner respectfully requests institution of review on claims 1-9, and a finding they are unpatentable.

Petition for *Inter Partes* Review of
U.S. Patent No. 10,771,849 B2

Dated: March 24, 2025

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (858) 550-6086
Fax: (720) 566-4099

By: /Eamonn Gardner/
Eamonn Gardner
Reg No. 63,322
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,878 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: March 24, 2025

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (858) 550-6086
Fax: (720) 566-4099

/Eamonn Gardner/
Eamonn Gardner
Reg No. 63,322
Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 10,771,849 B2**, including all exhibits (**Nos. 1001-1065**) and related documents, are being served via Federal Express on the March 24, 2025, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon the PO by serving the correspondence address of record with the USPTO as follows:

132285 – ELAN IP INC.
C/O ALLAN COLQUHOUN AND YASIN BISMILLA
2 BLOOR STREET EAST
UNIT 3500
TORONTO, ON M4W1A8
CANADA

And, via Federal Express upon lead counsel of record for the PO in the litigation pending before the U.S. District Court for the Northern District of California entitled Withrow Networks, Inc. v. Google, LLC, No. 5:24-cv-03203-PHK (N.D. Cal.) as follows:

Francisco A. Villegas (fvillegas@villegascefo.com)
Damir Cefo (dcefo@villegascefo.com)
Mark W. Halderman (mhalderman@villegascefo.com)
Evan M. Rosenbaum (erosenbaum@villegascefo.com)
VILLEGAS & CEFO, LLP
1350 Avenue of the Americas, Floor 2
New York, NY 10019

DATED: March 24, 2025

/Eamonn Gardner/
Eamonn Gardner
Reg. No. 63,322
egardner@cooley.com
COOLEY LLP
1299 Pennsylvania Avenue NW
Suite 700
Washington D.C. 20004
Tel: (858) 550-6086