

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC., and
AMAZON WEB SERVICES, INC.,

Petitioner,

v.

DIVX, LLC,

Patent Owner.

Case: IPR2025-01062

U.S. Patent No. 10,715,806

DECLARATION OF HARI KALVA

TABLE OF CONTENTS

	Page
I. INTRODUCTION	12
II. BACKGROUND AND QUALIFICATIONS	12
III. COMPENSATION AND RELATIONSHIP WITH PARTIES.....	17
IV. MATERIALS CONSIDERED	17
V. BASIS OF OPINION	24
A. Legal Standard for Anticipation.....	24
B. Legal Standard for Obviousness	24
C. Claim Construction.....	26
VI. RELATED LITIGATION AND RELATED PATENTS	27
VII. OVERVIEW OF THE 806 PATENT.....	27
A. Technical Background.....	27
1. Introduction to Video Coding	27
2. Introduction to Video Transcoding.....	39
3. Well-Known Transcoding Implementations.....	44
B. Patent Description	48
C. Priority Date	65
D. Level of Ordinary Skill in the Art.....	65
VIII. PRIOR ART TO THE CHALLENGED PATENT	66
A. Summary of the Prior Art.....	66
1. Sambe.....	66
2. Vetro.....	72

TABLE OF CONTENTS

(Continued)

	Page
3. Gu.....	80
B. General Motivations to Combine Prior Art References.....	85
1. Sambe in view of Vetro	85
2. Sambe in view of Vetro and Gu.....	96
IX. INVALIDITY	98
A. Ground 1: Claims 1-4, 7-14, and 17-21 of the 806 Patent are Rendered Obvious by Sambe in view of Vetro.....	98
1. Claim 1	98
[1.pre]: “A method for transcoding a source video file into a set of multiple alternate video streams, the method comprising:”	98
[1.a]: “generating, at a computer system configured as a media metadata generation device, media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene complexity information;”	100
[1.b]: “providing information based on the media metadata from the computer system to a plurality of transcoding devices; and”	107
[1.c]: “performing the following at each of the plurality of transcoding devices in parallel:”	114
[1.c(i)]: “receiving the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;”	114

TABLE OF CONTENTS (Continued)

	Page
[1.c(ii)]: decoding the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;	117
[1.c(iii)]: “receiving the information based on the media metadata from the computer system; and”	119
[1.c(iv)]: “encoding the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.”	122
2. Claim 2: “The method of claim 1, wherein a group of the plurality of transcoding devices perform processes in parallel for the same alternate video stream from the set of multiple alternate video streams.”	124
3. Claim 3: “The method of claim 1, wherein a group of the plurality of transcoding devices perform processes in parallel for different alternate video streams in the set of multiple alternate video streams.”	126
4. Claim 4: “The method of claim 1, wherein the media metadata further comprises scene change information indicating the start and end of a scene.”	127
5. Claim 7	129
[7.pre] “The method of claim 4, by further performing following at each of the plurality of transcoding devices in parallel:”	129

TABLE OF CONTENTS (Continued)

	Page
[7.a] “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.”	129
6. Claim 8: “The method of claim 1, wherein the source format and the target format have different resolutions.”	130
7. Claim 9: “The method of claim 1, wherein the source format and the target format correspond to different video encoding standards.”	132
8. Claim 10	134
[10.pre] “The method of claim 1, by further performing the following at each of the plurality of transcoding devices in parallel:”	134
[10.a] “dividing an image in the plurality of decoded images into a plurality of coding units based on the target format.”	135
9. Claim 11	137
[11.pre]: A system for transcoding video data, the system comprising:	137
[11.a]: a computer system configured as a media metadata generation device, wherein the computer system is configured to:	138
[11.a(i)]: generate media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene complexity information; and	138

TABLE OF CONTENTS (Continued)

	Page
[11.a(ii)] provide information based on the media metadata to a plurality of transcoding devices; and.....	138
[11.b]: the plurality of transcoding devices, configured to perform the following at each of the plurality of transcoding devices in parallel:	139
[11.b(i)]: receive the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;.....	139
[11.b(ii)]: decode the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;	139
[11.b(iii)]: receive the information based on the media metadata from the computer system; and.....	139
[11.b(iv)]: encode the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.....	140
10. Claim 12: “The system of claim 11, wherein a group of the plurality of transcoding devices perform processes in parallel for the same alternate video stream from the set of multiple alternate video streams.”	140
11. Claim 13: “The system of claim 11, wherein a group of the plurality of transcoding devices perform processes in parallel for different alternate video streams in the set of multiple alternate video streams.”	140

TABLE OF CONTENTS (Continued)

	Page
12. Claim 14: The system of claim 11, wherein the media metadata further comprises scene change information indicating the start and end of a scene.	141
13. Claim 17	141
[17.pre]: “The system of claim 14, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”	141
[17.a]: “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.”	142
14. Claim 18	142
[18.pre]: “The system of claim 11, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”	142
[18.a]: “dividing an image in the plurality of decoded images into a plurality of coding units based on the target format.....	142
15. Claim 19: “The system of claim 11, wherein the source format and the target format have different resolutions.”	143
16. Claim 20: “The system of claim 11, wherein the source format and the target format correspond to different video encoding standards.”	143
17. Claim 21	144

TABLE OF CONTENTS

(Continued)

	Page
[21.pre]: “A method for transcoding a source video file into a set of multiple alternate video streams, the method comprising:”	144
[21.a]: “generating, at a computer system configured as a media metadata generation device, media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene change information indicating the start and end of a scene, and scene complexity information;”	144
[21.b]: “providing information based on the media metadata from the computer system to a plurality of transcoding devices; and”	145
[21.c]: “performing the following at each of the plurality of transcoding devices in parallel:”	145
[21.c(i)]: “receiving the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;”	146
[21.c(ii)]: “decoding the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;”	146
[21.c(iii)]: “receiving the information based on the media metadata from the computer system;”	146
[21.c(iv)]: “dividing an image in the plurality of decoded images into a plurality of coding units based on a target format, wherein the source format and the target format have different resolutions;”	147

TABLE OF CONTENTS

(Continued)

	Page
[21.c(v)]: “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata; and”	148
[21.c(vi)]: “encoding the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on the target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.”	148
B. Ground 2: Claims 5-6 and 15-16 of the 806 Patent are Rendered Obvious by Sambe in view of Vetro and Gu	149
1. Claim 5	149
[5.pre]: “The method of claim 1, wherein the encoding of the plurality of decoded images of the decoded portion of video into an alternate video stream includes performing at least the following operations for images in the plurality of decoded images:”	149
[5.a]: “generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images using the scene change information and the scene complexity information within the information based on the media metadata according to the target format;”	150
[5.b]: “performing transforms on residual images of the plurality of coding units to generate sets of transform coefficients based on the target format; and”	154

TABLE OF CONTENTS (Continued)

	Page
[5.c]: “performing entropy encoding on the sets of transform coefficients to generate images for the second plurality of encoded images.”	157
2. Claim 6	160
[6.pre]: “The method of claim 1, by further performing the following at each of the plurality of transcoding devices in parallel:”	161
[6.a]: “performing quantization on the sets of transform coefficients for an image in the plurality of decoded images based at least in part on the scene complexity information within the information based on the media metadata; and”	161
[6.b]: “quantizing the generated set of transform coefficients according to the target format.”	163
3. Claim 15	165
[15.pre]: “The system of claim 11, wherein encoding the plurality of decoded images of the decoded portion of video into an alternate video stream is performed using at least the following operations for images in the plurality of decoded images:”	165
[15.a]: “generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images using the scene change information and the scene complexity information within the information based on the media metadata according to the target format;”	166

TABLE OF CONTENTS (Continued)

	Page
[15.b]: “performing transforms on residual images of the plurality of coding units to generate sets of transform coefficients based on the target format; and”	166
[15.c]: “performing entropy encoding on the sets of transform coefficients to generate images for the second plurality of encoded images.”	166
4. Claim 16	167
[16.pre]: “The system of claim 11, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”	167
[16.a]: “performing quantization on the sets of transform coefficients for an image in the plurality of decoded images based at least in part on the scene complexity information within the information based on the media metadata; and”	167
[16.b]: “quantizing the generated set of transform coefficients according to the target format.”	167
X. CONCLUSION.....	168

I, Hari Kalva, Ph.D., declare as follows:

I. INTRODUCTION

1. I have been asked by the parties requesting this review, Amazon.com, Inc. and Amazon Web Services, Inc. (“Petitioner”), to provide my expert opinions in support of the petition for Inter Partes review of U.S. Patent No. 10,715,806 (the “806 patent” or “challenged patent”) challenging the patentability of claims 1-21 (the “challenged claims”).

2. In summary, it is my opinion that the challenged claims are disclosed and rendered obvious by the prior art. I set forth below my detailed opinions.

II. BACKGROUND AND QUALIFICATIONS

3. I am Professor and Department Chair in the Department of Electrical Engineering and Computer Science at Florida Atlantic University. I regularly teach courses on Engineering Design, Video Communication, and Cloud Computing. I am also Director of Florida Atlantic University’s Multimedia Lab, in which capacity I conduct research in the broad area of visual computing that include video compression, communication, and analysis. Part of the research relates to development and optimization of video compression standards.

4. I received my Doctorate in Electrical Engineering from Columbia University in 2000. Prior to that, I received a Master of Philosophy in Electrical Engineering from Columbia University in 1999, a Master of Science in Computer

Engineering from Florida Atlantic University in 1994, and a Bachelor of Technology in Electronics and Communications Engineering from the N.B.K.R. Institute of Science and Technology at Sri Venkateswara University in Tirupati, Andhra Pradesh, India.

5. I have taught in the Department of Electrical Engineering and Computer Science at Florida Atlantic University since 2003. Before that, I was a Consultant in the Video Processing Group for Mitsubishi Electric Research Labs from 2001 to 2003; Co-Founder and Vice President of Engineering at Flavor Software, a New York based startup developing MPEG-4 standard based multimedia solutions, from 2000 to 2001; a Research Staff Associate in the Image and Advanced Television Laboratory at Columbia University from 1995 to 1996; and a Teaching Assistant in the Department of Electrical Engineering and Computer Science at Florida Atlantic University from 1993 to 1994.

6. I have extensive experience in multimedia data compression technologies. In addition to my education and nearly three decades of professional experience in data compression technologies, at both academic and commercial levels, I have been granted numerous patents for inventions relating to such technologies; I have authored over 100 peer refereed publications regarding such technologies; and I have contributed to several international video compression standards, including contributions of standard essential patents.

7. I am a 2025 Inductee to the Florida Inventors Hall of Fame and recognized for “Significant contributions to international standards for video compression, communications, and analytics that have enabled video streaming technologies, fundamentally transforming education, entertainment, and the workplace.” I am a Fellow of the National Academy of Inventors (NAI), an honor bestowed upon individuals whom the NAI’s Selection Committee deems to “have demonstrated a highly prolific spirit of innovation in creating or facilitating outstanding inventions that have made a tangible impact on the quality of life, economic development and the welfare of society.” I am also a senior member of the Institute of Electrical and Electronics Engineers (IEEE); a member of the IEEE’s Consumer Electronics, Circuits and Systems, Communications, and Signal Processing Societies, including the Multimedia Technical Committee of the IEEE’s Communications Society; Chair of the IEEE’s Palm Beach Section; and a member of the Association for Computer Machinery.

8. I have been granted 70 U.S. patents, covering inventions including systems and methods for encoding multimedia content; systems and methods for processing object-based audiovisual architecture; systems and methods for video and multimedia rendering, composition, and user-interactivity; methods for encrypting and compressing video, object-based audiovisual terminal and bitstream structures. I have also been granted over a dozen foreign patents covering similar

types of inventions. In addition, I have sold patent applications to Mitsubishi Electric Company covering multi-level bi-directional prediction and methods for encoding and decoding video.

9. In particular, I am an inventor of patented technology essential to the implementation of three video compression standards: H.264, also known as MPEG-AVC (for “Advanced Video Coding”) or MPEG-4 Part 10; H.265, also known as HEVC (for “High Efficiency Video Coding”); and H.266, also known as VVC (for “Versatile Video Coding”) or MPEG-I Part 3.

10. I was a key member of the team that initiated the MPEG-4 file format standards activity and wrote the software for the proposed file format. Based on the ideas and experiences with implementing the initial MPEG-4 file format, MPEG-4 video, and MPEG-4 Systems technologies, I developed requirements for the MPEG-4 file format that is used today. I have experience writing software for video compression and streaming systems including MPEG-1, MPEG-2, H.263, MPEG-4 part 2, and MPEG-4 part 10.

11. I have made over three dozen contributions to the development of international video compression standards from the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). Among other video compression standards, I made a variety of contributions to the MPEG-4 group of international standards for compressing digital audio and visual

data, multimedia systems, and file storage formats. A list of my contributions to international standards is provided in my curriculum vitae.

12. I served as U.S. delegate to the ISO standards committees during the development of the MPEG-4, MPEG-7, VVC, and VCM international video compression standards. Over the past three decades, I have also served on the technical program committees of a variety of conferences of the International Conference on Consumer Electronics, the International Conference on Multimedia and Expo, the International Workshop on Multimedia Signal Processing, the IEEE's International Symposium on Multimedia, and other organizations.

13. I have published numerous peer refereed articles, chapters, and books pertaining to video compression technology over the past three decades. Many of these publications have focused on various international video compression standards, including MPEG-2, H.264, and H.265, also known as HEVC or MPEG-H Part 2. I have also authored many publications on video compression technologies such as intra-frame coding, a technique that enables smaller files sizes and lower bit rates with little to no loss in quality. A list of my peer referred publications, including all publications authored in the previous ten years, is provided in my curriculum vitae (EX1030).

14. I have previously been retained as an expert witness in cases involving patents before the Patent Trial and Appeal Board of the U.S. Patent and Trademark

Office to assess the relevance of prior art to patents related to HEVC video decoding. A list of cases in which I have been retained as an expert witness, including all cases at which I testified as an expert at trial or by deposition during the previous four years, is provided in my curriculum vitae (EX1030).

III. COMPENSATION AND RELATIONSHIP WITH PARTIES

15. I am being compensated for my time at a rate of \$650 per hour. This compensation is not contingent upon my performance, the conclusions I reach in my analysis, the outcome of this matter, or any issues involved in or related to this matter.

16. I have no financial interest in Petitioner or any related parties.

IV. MATERIALS CONSIDERED

17. I have reviewed and considered, in preparing this declaration, the 806 patent (EX1001), its prosecution history (EX1002), and the following prior art references:

- Sambe, Y., Watanabe, S., Dong, Y., Nakamura, T., Wakamiya, N., *High-speed Distributed Video Transcoding for Multiple Rates and Formats*, E-88D(8) *IEICE Transactions on Information and Systems* 1923, 2005 (“Sambe”) (EX1004);

- U.S. Patent No. 6,490,320 to Vetro et al., filed on April 11, 2000 and titled “Adaptable Bitstream Video Delivery System” (“Vetro”) (EX1005);
- U.S. Pub. No. 2010/0189183 to Gu et al., filed on January 29, 2009 and titled “Multiple Bit Rate Video Encoding Using Variable Bit Rate and Dynamic Resolution for Adaptive Video Streaming (“Gu”) (EX1006).

18. I am also aware of, and considered in forming my opinions presented in this declaration, other information available to a person of ordinary skill in the art at the time of the effective filing date of the challenged patent (“POSITA”), such as other patents, reference materials, articles, and standards, and knowledge of a POSITA.¹ These include:

- Xin, J., Lin, C., and Sun, M., *Digital Video Transcoding*, 93(1) Proceedings of the IEEE 84, 2005 (available at: https://web.archive.org/web/20170810034657id_/http://www.ee.nthu.edu.tw/cwlin/pub/xcodingprocieee04.pdf) (EX1007);

¹ I understand that the challenged patent has an effective filing date of March 15, 2013, as I discuss in more detail below.

- Ahmad, I., Wei, X., and Zhang, Y., *Video Transcoding: An Overview of Various Techniques and Research Issues*, 7(5) IEEE Transactions on Multimedia 793, 2005 (available at: <https://ranger.uta.edu/~iahmad/journal-papers/j51.pdf>) (EX1008);
- Vetro, A., Sun, H., and Wang, Y., *Object-Based Transcoding for Adaptable Video Content Delivery*, 11(3) IEEE Transactions on Circuits and Systems for Video Technology 387, 2001 (available at: https://www.researchgate.net/profile/Anthony-Vetro/publication/3308265_Object-based_transcoding_for_adaptable_video_content_delivery/links/09e4150d079b1078ae000000/Object-based-transcoding-for-adaptable-video-content-delivery.pdf) (EX1009);
- Kuhn, P., Suzuki, T., and Vetro, A., *MPEG-7 Transcoding Hints for Reduced Complexity and Improved Quality*, Proceedings of the International Packet Video Workshop, 2001 (available at: https://www.researchgate.net/publication/2389933_MPEG-7_Transcoding_Hints_for_Reduced_Complexity_and_Improved_Quality) (EX1010);

- Lao, F., Xinggong, Z., and Zongming, G., *Parallelizing Video Transcoding Using Map-Reduce-Based Cloud Computing*, 2012 IEEE International Symposium on Circuits and Systems, 2012 (available at: <https://www.wict.pku.edu.cn/NetVideo/docs/20201104115437387542.pdf>) (EX1011);
- Dogan, S., Sadka, A.H., and Kondo, A.M., *MPEG-4 Video Transcoder for Mobile Multimedia Traffic Planning*, Second International Conference on 3G Mobile Communication Technologies, 2001 (available at: https://web.archive.org/web/20151220204359id_/http://personal.ee.surrey.ac.uk/Personal/S.Dogan/pubs/3G_2001_ppr_clr.pdf) (EX1012);
- Richardson, I., *The H.264 Advanced Video Compression Standard*, 2nd ed., John Wiley & Sons, 2010 (EX1013);
- Sun, H., Chen, X., and Chiang, T., *Digital Video Transcoding for Transmission and Storage*, CRC Press, 2005 (EX1014);
- Vetro, A., Charilaos, C., and Sun, H., *Video Transcoding Architectures and Techniques: An Overview*, IEEE Signal Processing Magazine, March 2003 (EX1015);

- Salomon, D. and Motta, G., *Handbook of Data Compression*, 5th ed., Springer, 2010 (EX1016);
- Van Beek, P., Smith, R., Touradj, E., Suzuki, T., and Askelof, J., *Metadata-Driven Multimedia Access*, IEEE Signal Processing Magazine, March 2003 (EX1017);
- Ho, C., Kuo, J., and Wu, J., *Building MPEG-7 Transcoding Hints from Intrinsic Characteristics of MPEG Videos*, 50(1) IEEE Transactions on Consumer Electronics 306, 2004 (EX1018);
- Ely, S. R., *MPEG Video Coding A Simple Introduction*, EBU Technical Review, 1995 (available at: https://tech.ebu.ch/docs/techreview/trev_266-ely.pdf) (EX1019);
- Yu, L. and Wang, J., *Review of the current and future technologies for video compression*, 11(1) Journal of Zhejiang University-Science C (Computers & Electronics) 1, 2010 (available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4262159904101e294d888389fe6248003c358f81>) (EX1020);

- Shi, C. and Bharat, B., *A Fast MPEG Video Encryption Algorithm*, ACM Multimedia '98, Bristol UK, 1998 (available at: <https://dl.acm.org/doi/10.1145/290747.290758>) (EX1021);
- Xie, B. and Zeng, W., *A Sequence-Based Rate Control Framework for Consistent Quality Real-Time Video*, 16(1) IEEE Transactions on Circuits and Systems for Video Technology 56, 2006 (available at: https://web.archive.org/web/20170808160855id_/http://vc.cs.ntu.edu.tw/home/paper/codfiles/kcyang/200612191526/A_sequence-based_rate_control_framework_for_consistent_quality_real-time_video.pdf) (EX1022);
- Bozóki, S., Westen, S.J.P., Legendijk, R.L., and Biemond, J., *Parallel Algorithms for MPEG Video Compression with PVM*, Proc. 1996 International EUROSIM Conference, Delft, The Netherlands, 1996 (available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9d38fbb62b965728f3f130eb2d91c1342388533b>) (EX1023);
- He, Y., Ahmad, I., and Liou, M.L., *A Software-Based MPEG-4 Video Encoder Using Parallel Processing*, 8(7) IEEE

Transactions on Circuits and Systems for Video Technology 909, 1998 (available at: [https://ranger.uta.edu/~iahmad/journal-papers/\[J19\]A%20software-based%20MPEG-4%20video%20encoder%20using%20parallel%20processing.pdf](https://ranger.uta.edu/~iahmad/journal-papers/[J19]A%20software-based%20MPEG-4%20video%20encoder%20using%20parallel%20processing.pdf)) (EX1024).

- G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191. PDF available here: <https://ieeexplore.ieee.org/abstract/document/6316136> (EX1029).

19. I understand that Petitioner and Patent Owner are involved in a parallel litigation involving seven patents, including the challenged patent.

20. Should new information be disclosed or discovered during this proceeding, I may supplement and further elaborate on the opinions I express in this declaration as necessary. Nothing in this declaration should be interpreted as an opinion that no grounds of invalidity exist other than the ones in the petition. I do not express such an opinion and have not reached such an opinion.

V. BASIS OF OPINION

A. Legal Standard for Anticipation

21. I understand that a claimed invention is unpatentable as “anticipated” by a single prior art reference if it discloses each and every element of the claim. I also understand that, even if a claim is not explicitly disclosed in the prior art reference, the prior art reference may nevertheless anticipate the claim if the subject matter of the claim is inherently present from the perspective of a POSITA. Moreover, I understand that even if the prior art reference does not disclose the limitations of a claim in the same arrangement or combination of the limitations of the claim, it may nevertheless anticipate if a POSITA would envisage the claimed arrangement or combination.

B. Legal Standard for Obviousness

22. I understand that a claimed invention is unpatentable as “obvious” if the differences between the invention and the prior art are such that the subject matter of the claim as a whole would have been obvious to a POSITA.

23. It is my understanding that “obviousness” is a question of law based on underlying factual issues including (1) the scope and content of the prior art, (2) the differences between the prior art and the asserted claims, (3) the level of ordinary skill in the pertinent art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, and failure of others.

24. I understand that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a POSITA. Alternatively, the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to a POSITA. In determining whether a combination based on either a single reference or multiple references would have been obvious, I understand that it is appropriate to consider the following factors:

- Whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- Whether a POSITA could implement a predictable variation, and would see the benefit of doing so;
- Whether a POSITA would have recognized a reason to combine known elements in the manner described in the claim;
- Whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- Whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

25. In an obviousness evaluation based on a combination of multiple prior art references, I understand that the prior art references themselves may provide a suggestion, motivation, or reason to combine, but other times the nexus linking two or more prior art references is simple common sense. I further understand that obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may be supplied by the direction of the marketplace.

C. Claim Construction

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

27. I applied plain-and-ordinary meaning to all terms of the challenged claims for my analysis in this proceeding.

28. I am not an attorney, and my opinions are limited to what I believe a POSITA would have understood the meaning of certain claim terms to be, based on the patent specification and prosecution history in light of the POSITA’s knowledge.

VI. RELATED LITIGATION AND RELATED PATENTS

29. I understand that the 806 patent is the subject of pending litigation involving Petitioner and Patent Owner: *DivX, LLC v. Amazon.com, Inc., and Amazon.com Web Services, Inc.*, No. 1:24-cv-02061 (Eastern District of Virginia, November 15, 2024) (the “parallel litigation”). I understand that Patent Owner sued Petitioner for alleged infringement of the 806 patent.

VII. OVERVIEW OF THE 806 PATENT

A. Technical Background

1. Introduction to Video Coding

30. Video coding is the process of compressing (encoding) and decompressing (decoding) digital video. EX1013, 7. To provide some context, “digital video data tends to take up a large amount of storage or transmission capacity and so video encoding and decoding, or video coding, is essential for any application in which storage capacity or transmission bandwidth is constrained.” EX1013, 2. Digital video is composed of a series of successive images, known as frames—commonly 24 per second for film, and more for television and other applications. *See* EX1014, 207-208. Each frame includes a matrix of hundreds of thousands of pixels that combine to form an image. *See* EX1013, 7-16; *see also* EX1014, 12-16. This large amount of information contained within digital videos makes them unwieldy to transmit and store in their original uncompressed form. *See* EX1013, 26; *see also* EX1014, 33-36.

31. Accordingly, “[t]he principal goal in the design of a video coding system is to reduce the number of bits, which is used to represent the video source, subject to some picture quality constraints,” resulting in a reduced video size and allowing for efficient transmission and storage. EX1014, 36; *see also* EX1016, 869-870; EX1019, 13. This reduction in bits, i.e., compression, is accomplished, at least in part, by removing spatial and temporal redundancies from the video source using tools such as the discrete cosine transform (DCT), quantization, and motion-compensated interframe prediction. EX1019, 13; *see also* EX1013, 25-26; EX1014, 34-53; EX1016, 869-873. Combining DCT with motion-compensated prediction is commonly referred to as hybrid coding, and it is used by major video coding standards such as MPEG-1, MPEG-2, MPEG-4, H.263, and H.264. EX1007, 85; EX1014, 35; EX1019, 17.

32. The coding process first breaks each video frame into processing units known as macroblocks or coding units, as described in more detail below:

The basic building block of an MPEG picture is the macroblock (Figure 9.17a). It consists of a 16×16 block of luminance (grayscale) samples (divided into four 8×8 blocks) and two 8×8 blocks of the matching chrominance samples. The MPEG compression of a macroblock consists mainly in passing each of the six blocks through a discrete cosine transform, which creates decorrelated values, then quantizing and encoding the results.

EX1016, 883; *see also* EX1019, 17; EX1014, 37-45.

33. Macroblocks or coding units are each individually processed—the coding steps described below are applied to them individually. EX1019, 14, 19; *see also* EX1014, 37-38, 44-45; EX1013, 31-35, 42-48. Specifically, operating at the macroblock or coding unit level, the coding process eliminates pixel redundancies within a given frame and between different frames in the video. EX1013, 25-26; *see also* EX1014, 36-47; EX1019, 13. This is possible because “pixel values are not independent but are correlated with their [sic] neighbours, both within the same frame and across frames.” EX1019, 13; *see also* EX1014, 36-37; EX1019, 13. Redundancy within the same frame is known as spatial redundancy. EX1013, 25-26. For example, many black pixels representing a night sky in a single frame. Redundancy across different frames is known as temporal redundancy. *Id.*; *see also* EX1014, 44. For example, pixels representing the same portion of the night sky in successive frames. This excerpt from Richardson describes the concept of spatial and temporal redundancies in more detail below:

In the temporal domain, there is usually a high correlation or similarity between frames of video that were captured at around the same time. Temporally adjacent frames, i.e. successive frames in time order, are often highly correlated, especially if the temporal sampling rate or frame rate is high. In the spatial domain, there is usually a high correlation between pixels (samples) that are close to each other, i.e. the values of neighbouring samples are often very similar (Figure 3.2).

EX1013, 25-26.

34. This means that “the value of a pixel is predictable, given the values of the neighbouring [sic] pixels” (i.e., the values of neighboring pixels within the same frame (spatial) or between different frames (temporal)). EX1019, 13; *see also* EX1016, 869-873; EX1013, 25-26; *see also* EX1014, 36-47.

35. These spatial and temporal redundancies are removed by constructing a prediction of current macroblock based on neighboring frame and/or pixel values. EX1013, 27; EX1014, 44-45; *see also* EX1019, 16-19. The output, called a “prediction residual,” represents the prediction macroblocks subtracted from the original uncompressed macroblocks. EX1013, 27; *see also* EX1019, 19. This excerpt below describes this process in more detail for the example H.264/AVC standard:

The prediction model attempts to reduce redundancy by exploiting the similarities between neighbouring [sic] video frames and/or neighbouring [sic] image samples, typically by constructing a prediction of the current video frame or block of video data. In H.264/AVC, the prediction is formed from data in the current frame or in one or more previous and/or future frames. It is created by spatial extrapolation from neighbouring [sic] image samples, intra prediction, or by compensating for differences between the frames, inter or motion compensated prediction. The output of the prediction model is a residual frame, created by subtracting the prediction from the actual current frame, and a set of model parameters indicating the intra prediction type or describing how the motion was compensated.

EX1013, 27; *see also* EX1014, 44-45; EX1019, 16-19.

36. Interframe prediction, also referred to as temporal prediction, is performed by comparing macroblocks of the frame currently being encoded to those

of a reference frame, and encoding the difference between those macroblocks as a residual macroblock, as discussed in more detail below:

Motion estimation: Motion estimation of a macroblock involves finding a 16×16 -sample region in a reference frame that closely matches the current macroblock. The reference frame is a previously encoded frame from the sequence and may be before or after the current frame in display order. A search area in the reference frame centred [sic] on the current macroblock position is searched and the 16×16 region within the search area that minimizes a matching criterion is chosen as the ‘best match’ (Figure 3.9).

Motion compensation: The luma and chroma samples of the selected ‘best’ matching region in the reference frame is subtracted from the current macroblock to produce a residual macroblock that is encoded and transmitted together with a motion vector describing the position of the best matching region relative to the current macroblock position.

EX1013, 31-32; *see also id.*, 42-48; EX1019, 14, 19; EX1014, 37-38, 44-45, Fig. 2.2, 2.8.

37. Similarly, Intra frame prediction, also called spatial prediction, operates by extrapolating blocks in a frame based on neighboring blocks, as described in more detail below:

The prediction for the current block of image samples is created from previously-coded samples in the same frame. Figure 3.23 shows a block that is to be predicted, centre, in the current frame. Assuming that the blocks of image samples are coded in raster-scan order, which is not always the case, the upper/left shaded blocks are available for intra prediction. These blocks have already been coded and placed in the output bitstream. When the decoder processes the current block, the shaded upper/left blocks are already decoded and can be used to re-create the prediction.

Many different approaches to intra prediction have been proposed. H.264/AVC uses spatial extrapolation to create an intra prediction for a block or macroblock. Figure 3.24 shows the general concept. One or more prediction(s) are formed by extrapolating samples from the top and/or left sides of the current block. In general, the nearest samples are most likely to be highly correlated with the samples in the current block (Figure 3.25) and so only the pixels along the top and/or left edges are used to create the prediction block. Once the prediction has been generated, it is subtracted from the current block to form a residual in a similar way to inter prediction. The residual is transformed and encoded, together with an indication of how the prediction was generated.

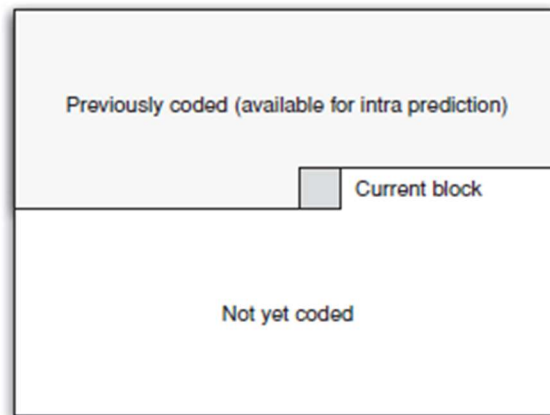


Figure 3.23 Intra prediction: available samples

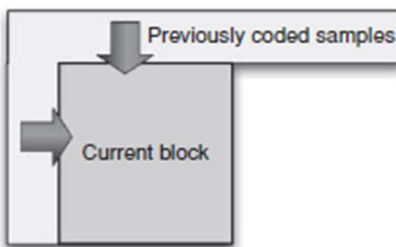


Figure 3.24 Intra prediction: spatial extrapolation

EX1013, 38-40, Figs. 3.23, 3.24.

38. A further mathematical transformation then compresses the prediction residuals, resulting in a set of values, known as quantized transform coefficients. EX1013, 27; *see also* EX1014, 37-41; EX1019, 14-15. Specifically, spatial redundancy is further reduced using a spatial model which converts the predicted residual macroblocks into relatively few transform coefficients.

The residual frame forms the input to the spatial model which makes use of similarities between local samples in the residual frame to reduce spatial redundancy. In H.264/AVC this is carried out by applying a transform to the residual samples and quantizing the results. The transform converts the samples into another domain in which they are represented by transform coefficients. The coefficients are quantized to remove insignificant values, leaving a small number of significant coefficients that provide a more compact representation of the residual frame. The output of the spatial model is a set of quantized transform coefficients.

EX1013, 27; *see also* EX1014, 37-41; EX1019, 14-15.

39. One common transform is the discrete cosine transform (“DCT”), which is described in more detail below:

[T]he DCT transform is a reversible process which maps between the normal 2-D presentation of the image and one which represents the same information in what may be thought of as the frequency domain. Each coefficient in the 8 x 8 DCT domain block indicates the contribution and quantization are used together to remove spatial redundancy. Each coefficient in the 8 x 8 DCT domain block indicates the contribution of a different DCT “basis” function to the original image block. The lowest frequency basis function (top-left in Fig. 3) is called the DC coefficient and may be thought of as representing the average brightness of the block.

The DCT does not directly reduce the number of bits required to represent the block. In fact, for an 8 x 8 image block of 8-bit pixels, the

DCT produces an 8 x 8 block of at least 11-bit DCT coefficients, to allow for reversibility! The reduction in the number of bits follows from the fact that, for typical blocks of natural images, the distribution of coefficients is non-uniform – the transform tends to concentrate the energy into the low-frequency coefficients, and many of the other coefficients are near zero. The bit-rate reduction is achieved by not transmitting the near-zero coefficients, and by quantizing and coding the remaining coefficients as described below. The distribution of the nonuniform coefficients is a result of the spatial redundancy present in the original image block.

Many different forms of transformation have been investigated for bit-rate reduction. The best transforms are those which tend to concentrate the energy of a picture block into a few coefficients. The DCT is one of the best transforms in this respect and has the advantage that the DCT and its inverse are easy to implement in digital processing. The choice of an 8 x 8 block-size is a trade-off between the need to use a large picture area for the transform, so the energy compaction described above is most efficient, and the fact that the content and movement of the picture varies spatially, which would tend to support a smaller block-size. A large block-size would also emphasize variations from block-to-block in the decoded picture; it would also emphasize the effects of “windowing” by the block structure.

EX1019, 14; *see also* EX1014, 37-41.

40. Then, as noted above, bit-rate reduction is in part accomplished through quantizing the set of coefficients remaining after the DCT is applied, resulting in the quantized transform coefficients. EX1019, 14. Quantization results in some information loss, but results in better compression. EX1014, 40; *see also* EX1019, 14-15. The quantization step is described in more detail below:

After a block has been transformed, the transform coefficients are quantized. Different quantization is applied to each coefficient depending on the spatial frequency within the block that it represents. The objective is to minimize the number of bits which must be

transmitted to the decoder, so that it can perform the inverse transform and reconstruct the image: reduced quantization accuracy reduces the number of bits which need to be transmitted to represent a given DCT coefficient, but increases the possible quantization error for that coefficient. Note that the quantization noise introduced by the coder is not reversible in the decoder, so the coding and decoding process is “lossy”.

More quantization error can be tolerated in the high-frequency coefficients, because HF noise is less visible than LF quantization noise. Also, quantization noise is less visible in the chrominance components than in the luminance component. MPEG uses weighting matrices to define the relative accuracy of the quantization of the different coefficients. Different weighting matrices can be used for different frames, depending on the prediction mode used. The weighted coefficients are then passed through a fixed quantization law, which is usually a linear law. However, for some prediction modes there is an increased threshold level (i.e. a dead-zone) around zero. The effect of this threshold is to maximize the number of coefficients which are quantized to zero: in practice, it is found that small deviations around zero are usually caused by noise in the signal, so suppressing these values actually “improves” the subjective picture quality.

Quantization noise is more visible in some blocks than in others; for example, in blocks which contain a high-contrast edge between two plain areas. In such blocks, the quantization parameters can be modified to limit the maximum quantization error, particularly in the high-frequency coefficients.

EX1019, 14-15; *see also* EX1014, 40-41.

41. The quantized transform coefficients are then passed to an entropy encoder for further processing. EX1013, 27; EX1019, 15, 19, Fig. 4. The following excerpts describe this process in detail:

The parameters of the prediction model, i.e. intra prediction mode(s) or inter prediction mode(s) and motion vectors, and the spatial model, i.e. coefficients, are compressed by the entropy encoder. This removes

statistical redundancy in the data, for example representing commonly occurring vectors and coefficients by short binary codes. The entropy encoder produces a compressed bit stream or file that may be transmitted and/or stored. A compressed sequence consists of coded prediction parameters, coded residual coefficients and header information.

EX1013, 27; *see also* EX1019, 15, 19, Fig. 4; EX1020, 9.

42. More specifically, the quantized transform coefficients are typically entropy encoded using a Huffman coding scheme, as described below:

Each of I, P and B frames is divided into macroblocks. A macroblock is a 16 x 16 pixel array which is further subsampled into four 8 x 8 *luminance* Y blocks, one 8 x 8 *chrominance* Cr block and one 8 x 8 *chrominance* Cb block, as shown in Figure 1. Each of 8 x 8 Y, Cb and Cr blocks is fed to a pipeline of DCT, quantization and Huffman entropy coding, as shown in Figure 2. DCT concentrates most of the energy in the lower spatial frequencies, i.e., the upper-left corner of the 8 x 8 block. After quantization, many DCT coefficients become zeros. The quantization output is linearized according to the zig-zag order to a vector $\langle \text{DC}, \text{AC}_1, \text{AC}_2, \dots, \text{AC}_{63} \rangle$, as shown in Figure 3.

DC coefficient denotes the average brightness in the spatial block, and AC coefficients contain detail image information. Zig-zag sequencing of AC coefficients places high-energy, low-frequency coefficients before the low-energy, high-frequency coefficients.

The run length encoding turns vector $\langle \text{DC}, \text{AC}_1, \text{AC}_2, \dots, \text{AC}_{63} \rangle$ into a sequence of (skip, value) pairs, then the Huffman entropy coding is used to change the (skip, value) pair sequence into a compressed bitstream. MPEG standard provides a Huffman codeword table, where more frequently occurring (skip, value) pairs are assigned fewer bits in order to achieve high compression rate. Every Huffman codeword reserves a sign bit, 0 mean positive, 1 mean negative. These sign bits are the exact positions to encrypt the video in VEA.

EX1021, 82; *see also* EX1020, 9.

43. Said more succinctly, the entropy encoder removes “statistical redundancy in the data, [like] for example representing commonly occurring vectors and coefficients by short binary codes.” EX1013, 27; EX1019, 15, 19, Fig. 4. Basically, entropy encoding assigns fewer bits to frequently recurring transform coefficients, thereby minimizing the number of bits needed to encode each frame or block. EX1020, 9; EX1021, 82.

44. In MPEG coding standards such as MPEG-2 and MPEG-4, interframe prediction (which reduces temporal redundancy) are combined with the DCT (which reduces spatial redundancy) and entropy coding tools (which reduces statistical redundancy from the data), as described in more detail below:

The popular international standards for video compression such as MPEG-2, MPEG-4 and H.264/AVC make use of the well-known ‘hybrid’ video encoder model in which a prediction, formed from previously-transmitted frames, is subtracted from the current frame to form a residual which is then transformed using a block transform such as the DCT, quantized and coded for transmission.

EX1013, 250; *see also id.*, 26 (explaining that popular compression formats including MPEG-2 Video and MPEG-4 Visual are based upon a model “that uses prediction and/or block-based motion compensation, transform, quantization and entropy coding”); EX1019, 17; EX1013, 27.

45. This coding process results in a compressed bitstream (i.e., the encoded video) consisting of “coded prediction parameters, coded residual coefficients, and header information” that can be efficiently transmitted or stored. EX1013, 27. Due

to the compression, the encoded video is smaller than the original video file. EX1013, 27. To view the encoded video requires using a decoder to recreate the original video by reversing the encoding process. *Id.*, 27-28. When decoding the compressed bitstream, the decoder reconstructs the video using the same steps as the encoding process but in reverse, as described below:

The video decoder reconstructs a video frame from the compressed bit stream. The coefficients and prediction parameters are decoded by an entropy decoder after which the spatial model is decoded to reconstruct a version of the residual frame. The decoder uses the prediction parameters, together with previously decoded image pixels, to create a prediction of the current frame and the frame itself is reconstructed by adding the residual frame to this prediction.

Id., 27-28.

46. The decoded video will not have identical visual quality to the video prior to the encoding process because common encoding techniques introduce some data loss as a trade-off for greater compression. *Id.* As noted above, this information loss occurs at the quantization step. EX1014, 40. However, because video coding techniques aim to remove *subjective* redundancy, the encoding steps outlined above remove redundant elements “without significantly affecting the viewer’s perception of video quality.” EX1013, 25; EX1016, 933 (“[t]he chief aim of a lossy video encoder is to lose only data that is perceptually irrelevant; data whose absence would not be noticed by a viewer.”). For example, the quantization process exploits limitations in human perception by “quantiz[ing] high[er] frequencies with coarser

quantization steps that will suppress high frequencies with no subjective degradation, thus taking advantage of human visual perception characteristics[,]” while “[t]he bits saved for coding high frequencies are used for lower frequencies to obtain better subjectively coded images.” EX1014, 40.

2. Introduction to Video Transcoding

47. The 806 patent is directed to a form of video coding known as transcoding. *See generally* EX1001; EX1003, ¶XX. Generally, “[v]ideo transcoding is the operation of converting a video from one format to another format” where “[a] format is defined by such characteristics as the bit rate, frame rate, spatial resolution, coding syntax, and content.” EX1007, 84; *see also* EX1008, 793; EX1015, 18. Transcoding emerged in the early 2000s to deal with rapid advancements in multimedia systems. EX1015, 18. These advancements presented the “problem of interactivity and integration of video data with computer, cellular, and television systems.” *Id.* Specifically, the problem was that “[a]s the number of networks, types of devices, and content representation formats increase[d], interoperability between different systems and different networks [became] more important.” *Id.*; *see also* EX1004, 1923; EX1009, 387. “Thus, devices such as gateways, multipoint control units, and servers [had to] be developed to provide a seamless interaction between content creation and consumption.” EX1015, 18. But there was no one-size-fits-all approach to video coding for use across all devices,

platforms, and networks. *See* EX1004, 1923; *see also* EX1009, 387. Transcoding addressed this problem by converting videos from one encoded format to another. *See id.*

48. Transcoding could change a video file from one encoding standard, such as MPEG-2, which was the common coding standard for DVDs, to another coding standard like MPEG-4, which was more commonly used for internet video transmission. EX1004, 1923; *see also* EX1009, 387. To address different device or network requirements, transcoding could also change coding parameters for a video, such as bit rate, frame rate, or resolution. EX1007, 84-85, Fig. 1. For example, devices may access the Internet through various types of networks, and each of these networks may have “different channel characteristics such as bandwidth, bit errors, and packet loss rates.” EX1007, 84; EX1008, 793; *see also* EX1015, 18-20, Fig. 1. A source video encoded at 5000 kilobits per second (“Kbps”) may incur long delays when transmitted over a network that supports only 600 Kbps. Transcoding addresses this by adapting the bit rate of a compressed video stream to a bit rate appropriate for the bandwidth of the channel over which it is to be transmitted. EX1007, 84 (“One of the earliest applications of transcoding is to adapt the bit rate of a precompressed video stream to a channel bandwidth.”), 91 (bit-rate control “is responsible for maintaining consistent video quality while satisfying bandwidth, delay, and memory constraints”); *see also* EX1009, 387; EX1008, 793, 797;

EX1015, 18-20, Fig. 1. Also, for example, a video could be changed from a constant bitrate format to a variable bitrate format, allowing the transcoder to allocate bit rates based on video content complexity (assigning higher bit rate to complex scenes) which ensures a more consistent level of video quality regardless of scene complexity. EX1014, 176; *see also* EX1007, 84 (describing the use of transcoding to “flexibly deliver multimedia data to users” over “access networks [with] different channel characteristics such as bandwidth, bit error rates, and packet loss rates”); EX1010, 1, 3; EX1015, 18 (describing conversion between constant bit-rate to variable bit-rate to “facilitate more efficient transport of video”).

49. As noted above, transcoding also promotes accessibility to multimedia content among various types of end-user devices, which may have disparate processing, display, and decoding capabilities. *See, e.g.*, EX1007, 84-85, Fig. 1; *see also, e.g.*, EX1008, 793. Xin provides more detail on this point:

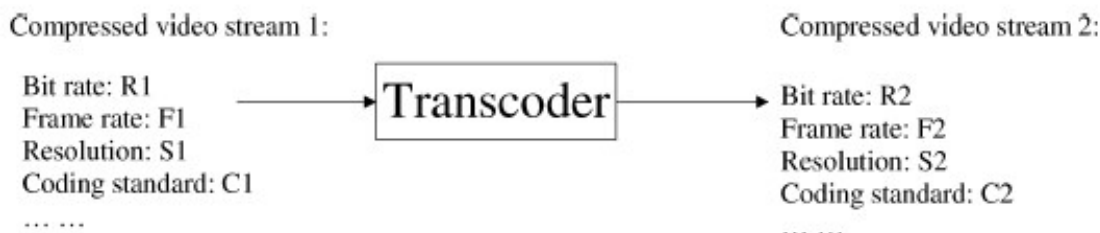
At the users’ end, network appliances including handheld computers, personal digital assistants (PDAs), set-top boxes, and smart cellular phones are slated to replace personal computers as the dominant terminals for accessing the Internet. These network terminals vary significantly in resources such as computing power and display capability. To flexibly deliver multimedia data to users with different available resources, access networks, and interests, the multimedia content may need to be adapted dynamically according to the usage environment [2]. Transcoding is one of the key technologies to fulfill this challenging task.

See, e.g., EX1007, 84.

50. Because of these differences, multimedia content originally authored and compressed with a certain format may need bit rate adjustment and format conversion to allow access by receiving devices with diverse capabilities. EX1008, 793; *see also* EX1007, 84-85, Fig. 1; EX1015, 18, 21, 24. For example, a source video compressed using the MPEG-2 compression format cannot be played on a mobile phone that only supports video compressed in the MPEG-4 format. *Id.* As noted above, video transcoding addresses these needs by converting the format of the source video into a target format. *See, e.g.,* EX1007, 84; EX1008, 793. Furthermore, transcoding may reduce spatial and/or temporal resolution of the multimedia content. EX1008, 793. Reduction of spatial resolution allows mobile devices “with limited display and processing power” to “access video originally captured in a higher resolution.” EX1015, 18, 21. Similarly, reduction of temporal resolution may “satisfy processing limitations” of mobile devices that “may only be capable of decoding and displaying” a certain number of frames per second, or it may allow such devices to prioritize battery life “at the cost of receiving fewer frames.” EX1015, 24.

51. Figure 1, below, illustrates the transcoding process at a high level. EX1007, Fig. 1. As can be seen, a transcoder receives a compressed video stream with certain coding parameters—e.g., bit rate, frame rate, resolution, and/or the

coding standard—and converts it into a second compressed bitstream with different coding parameters. *Id.*



Id., 84-85, Fig. 1; *see also* EX1008, 793, Fig. 1.

52. Transcoders are composed of both a decoder and an encoder. *See, e.g.*, EX 1005, 2:14-31, Fig. 1; *see also, e.g.*, EX1007, 85, Fig. 4; EX1014, 105. This is shown in the below image. A transcoder receives an encoded video with given bitrate (R_{in}). EX1005, 2:14-31, Fig. 1. The transcoder decodes the encoded video (step 110) and re-encodes it (step 120) at a new bitrate (R_{out}) based on the specific requirements 102 of a user's device. *Id.*

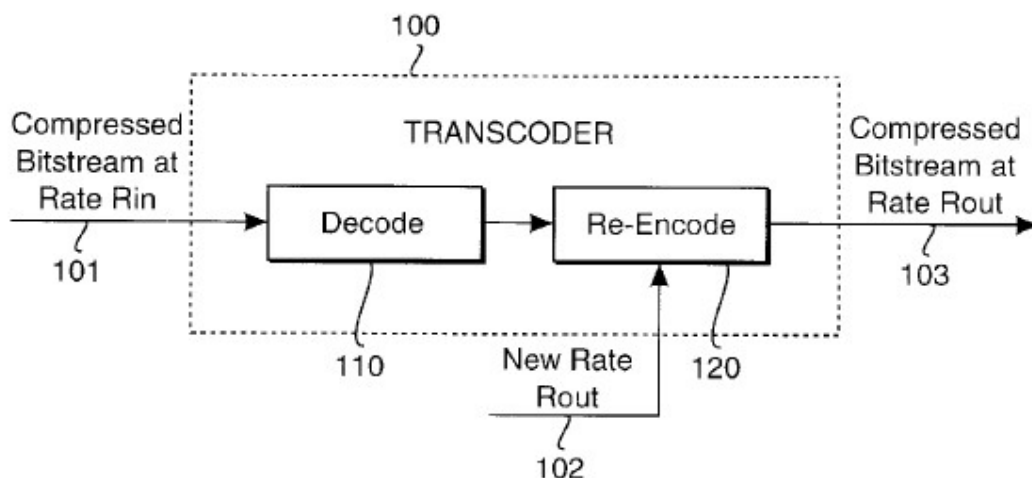


FIG. 1
PRIOR ART

EX1005, Fig. 1.

3. Well-Known Transcoding Implementations

53. An early area of focus for POSITAs was transcoding speed. Using a single transcoder to decode and then reencode an entire video was limited by the computing resources and processing capacity available to that single transcoder. EX1003, ¶ XX; *see also cf.* EX1011, 2905 (describing parallel transcoding as a solution for fast transcoding); EX1012, 3-4 (describing parallel transcoding as “more efficient and less delay prone” than sequential transcoding); EX1004, 1923. POSITAS recognized that by distributing the transcoding process among multiple transcoders working in parallel, the transcoding process could be made “more efficient and less delay prone.” EX1012, 3-4; *see also* EX1011, 2905; EX1004, 1923; EX1023, 1, 12; EX1024, 910. So, POSITAs began using multiple transcoders to transcode different segments of the same video in parallel. *See* EX1011, 2905; *see also* EX1004, 1923; EX1003, ¶XX. Those segments would then be recombined to form the reencoded video. *See* EX1011, 2905; *see also* EX1004, 1924.

54. But using multiple transcoders to transcode different segments of a video had its own set of challenges—primarily, ensuring a consistent video quality regardless of which transcoder handles which segment. EX1004, 1923; *see also* EX1011, 2906. Ensuring consistent video quality requires proper budgeting and allocation of bitrates to each video segment. EX1007, 84-85, 91-92; *see also* EX1009, 391-92; EX1010, 1. This is because video segments that contain a high

level of motion (for example, segments depicting a football game) require complex calculations of motion estimation, and therefore require more bitrate to compress than segments with less motion activity (such as segments depicting a news anchor reading the news from their desk). EX1017, 48; *see also* EX1005, 21:7-56. EX1017, 48. In other words, more complex video segments require more bits to compress and maintain a consistent quality than less complex video segments. *Id.* To properly budget and allocate bitrates for each segment, a POSITA would have to understand the complexity of each segment individually. EX1017, 48; *see also* EX1005, 21:7-56.

55. Since at least the introduction of MPEG-7 in 2001, well before the 2013 priority date claimed by the 806 patent, one well-known way to understand the complexity of a video segment was by extracting metadata from the video. EX1017, 47-48; *see also* EX1009, 391-92; EX1010, 1; EX1005, 4:14-5:11. Metadata is “any suitable information about the media content” of an encoded video. EX1001, 4:25-26. This can include general information about a video, such as the encoding standard with which a given video is encoded, or specific information about the video, such as complexity information about a given scene within the video. EX1017, 47-48; *see also* EX1009, 388, 391-92; EX1010, 1; EX1005, 21:8-22:61. Accordingly, to understand the complexity of each video segment, a POSITA would have looked to extracted metadata, like for example, the “difficulty hint” defined by

the MPEG-7 standard. EX1017, 47-48; *see also* EX1005, 4:14-5:11, 18:24-20:14, 21:7-56; EX1009, 391-92; EX1010, 1. “The difficulty hint describes the encoding complexity of segments within a video sequence (or regions within an image).” EX1017, 48; *see also* EX1018, 306; EX1010, 3-4. “A low difficulty value depicts a video segment which can be easily encoded using a low bitrate,” while “[a]t a high difficulty value[,] more bits have to be spent for this video segment to achieve a satisfactory subjective quality.” EX1010, 3. Thus, difficulty hints are defined for each video segment in a sequence. *Id.*

56. Uses of the difficulty hint in the transcoding process are described in more detail below.

This hint can be used for improved bit rate control and bit rate conversion from constant bit rate (CBR) to variable bit rate (VBR). The bit allocation and the selection of quantizer parameters depend on the rate-distortion characteristic [33], [34]. Based on these models, the difficulty hint is used to calculate the bit budget. The hint is a weight, which indicates the relative encoding complexity of each segment. It is normalized within the content. The encoder can assign more bits to a difficult scene and remove bits from a relatively easy scene. Since mainly the pictures with low quality are improved, the overall subjective quality is significantly improved.

EX1017, 48.

57. Using the difficulty hint, finite bitrate resources would be budgeted between video segments to ensure that each segment, regardless of its complexity, is transcoded with a consistent quality. EX1017, 48; *see also* EX1010, 3-4; EX1005, 21:7-56; EX1009, 391; EX1010, 1. However, “the basic assumption” of using

transcoding hints to reduce complexity and improve video quality in transcoding is that “transcoding hints are generated in advance, and are stored together with the host video itself (or stored separately but with a proper linkage to the video).” EX1018, 306; EX1010, 1; EX1009, 388. This allows for the meta-data to be efficiently extracted before the content is transmitted or needs to be converted. EX1009, 388. Then, “when a request for the content is presented, any external conditions, e.g., network conditions, can be assessed and the transcoding can be appropriately executed.” *Id.*

58. Consistent with that, it was known that for the difficulty hint to efficiently assign bits based on the complexity of each segment in a video sequence, the transcoders need to know the encoding complexity of each segment prior to any transcoding. EX1017, 48; EX1018, 306; EX1010, 3-4. As a result, the difficulty values for each segment are generated in advance and normalized to reflect their relative complexity compared to other segments, rather than using the absolute value of each segment’s coding complexity. EX1010, 3-4.

59. A natural application of the MPEG-7 difficulty hint would be for bitrate allocation between different transcoders in a parallel transcoding scheme as described above. This would allow for each of the video segments to be budgeted with an appropriate number of bits to maintain a consistent video quality throughout all of the segments, regardless of their complexity. EX1017, 48; *see also* EX1010,

3-4; EX1005, 21:7-56; EX1009, 391; EX1010, 1. And as described above, the encoding difficulty for each video segment would need to be generated in advance of any transcoding. EX1017, 48; EX1018, 306; EX1010, 3-4. Otherwise, in the absence of another technique for determining the complexity of each video segment, the bitrate allocated to each segment would be imprecise, resulting in quality discontinuities and degradation due to a lack of information regarding the coding complexity of neighboring frames. *Cf.* EX1004, 1923 (describing other techniques); *see also* EX1017, 48 (stating that to “encode efficiently in VBR mode, the encoder in the transcoder *has to know* the encoding difficulty of the whole sequence in advance” (emphasis added)). This would lead to improper allocation of bitrate and inconsistent video quality for the first video segments to be transcoded, and would have a cascading negative effect on the remaining segments in the video. EX1004, 1923-1927 (describing issues relating to poor quality information).

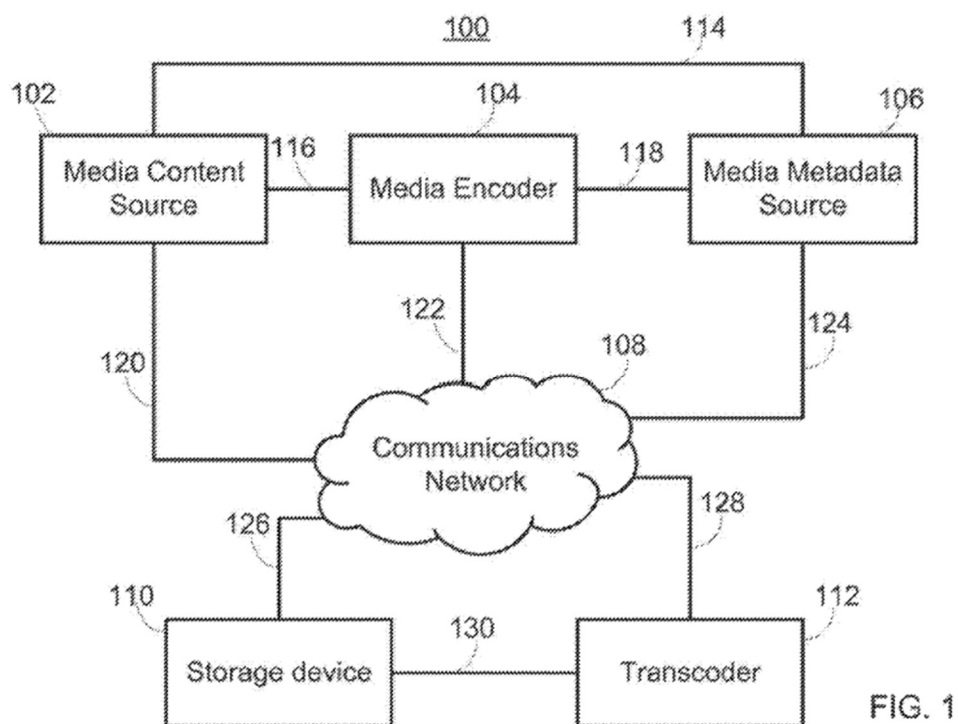
B. Patent Description

60. U.S. Patent No. 10,715,806 B2, titled “Systems, Methods, and Media for Transcoding Video Data,” to Ivan Vladimirovich Naletov and Sergey Zurpal, was filed on March 11, 2019 and issued on July 14, 2020. EX1001, Cover.

61. The 806 patent describes “new mechanisms for transcoding video data.” EX1001, 1:23-47. Specifically, the 806 patent describes transcoding input media content having a first format or resolution into media content having a second

format or resolution using metadata to facilitate the process. *See generally id.*, 8:40-10:15. The 806 patent suggests that the disclosed invention offers greater efficiency than prior art transcoding methods in which transcoders estimated encoding parameters rather than using metadata to determine transcoding parameters. *Id.*, 1:33-47. But as noted above and discussed in more detail below, the use of metadata in transcoding was well-known years before the priority date of the 806 patent. *See supra* § VII.A.3.

62. In Figure 1, shown below, the 806 patent depicts a hardware architecture in accordance with the alleged invention.



EX1001, 3:1-3, Fig. 1.

63. As depicted in Figure 1, the hardware architecture consists of the following conventional components: media content source 102, the media encoder 104, the media metadata source 106, storage device 110, and the transcoder 112. *Id.*, 3:3-8, Fig. 1. The media content source 102 “can include any suitable device that can provide media content,” such as a camera, television distribution equipment, or Internet provider equipment. EX1001, 3:9-22, Fig. 1. The media encoder 104 “can include any suitable circuitry that is capable of encoding media content,” such as “one or more video encoders that can encode video data including a set of images in accordance with a suitable coding standard[.]” EX1001, 4:8-17, Fig. 1. The media metadata source 106 “can include any suitable circuitry that is capable of providing metadata for media content,” which can include “any suitable information about the media content.” EX1001, 4:23-26, Fig. 1. And transcoder 112 “can include *any suitable circuitry* that is capable of converting input media content having a first format into media content having a second format.” EX1001, 6:9-11 (emphasis added). For example, it “can include a suitable video transcoder that can convert a first set of images that are encoded in accordance with a first coding scheme into a second set of images that are encoded in accordance with a second coding scheme.” EX1001, 6:11-17.

64. In Figure 2, the 806 patent depicts another hardware architecture in accordance with the alleged invention, including an expanded version of transcoder 112 from Figure 1.

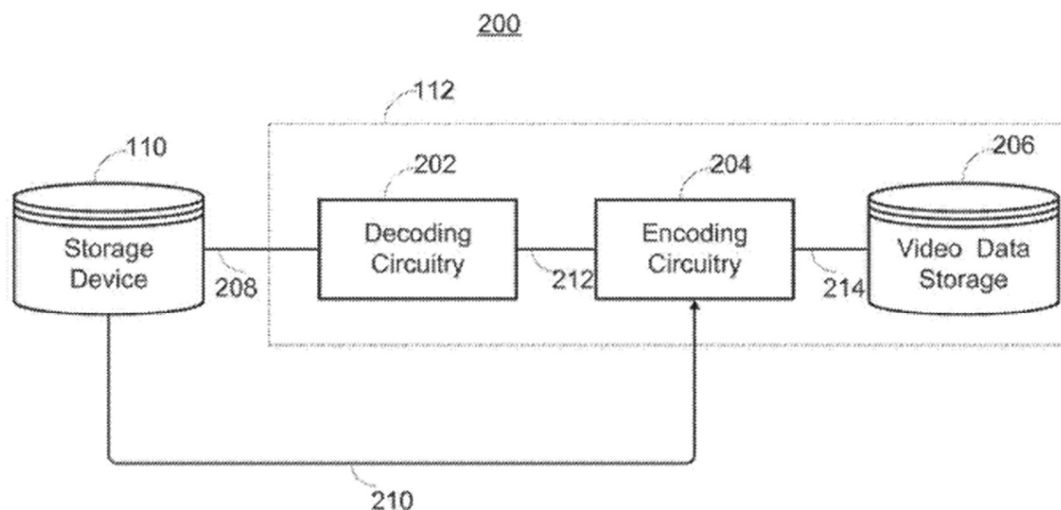


FIG. 2

EX1001, 7:9-14, Fig. 2.

65. The transcoder 112 includes cascaded decoding circuitry 202 and encoding circuitry 204, which as noted above, was well-known in the art, and a video-data storage component. *Id.*, Fig. 2.; *supra* § VII.A.2. The decoding circuitry 202 “can include any suitable circuitry that is capable of performing video decoding,” such as “one or more decoders that can decode a set of encoded images based on a suitable coding standard[.]” EX1001, 7:15-19. And the encoding circuitry 204 “can include any suitable circuitry that is capable of performing video encoding,” such as “one or more suitable encoders that can encode a set of images

based on a suitable encoding standard[.]” EX1001, 7:21-26. Thus, the transcoder disclosed in the 806 patent includes an encoder and decoder, which as noted above, was a known implementation in the prior art. *Supra* § VII.A.2.

66. In Figure 3, the 806 patent generally describes the transcoding process in accordance with the alleged invention.

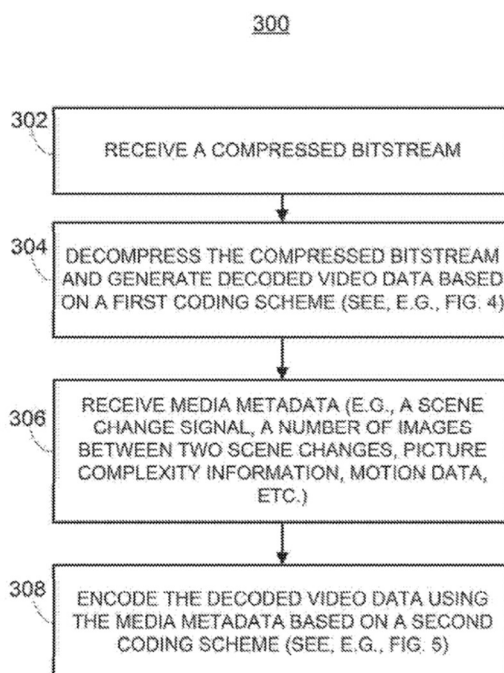


FIG. 3

EX1001, 8:35-37, Fig. 3.

67. The transcoding process shown in Figure 3 begins with the transcoder receiving a compressed bitstream from storage 110, media encoder 104, or media content source 102 (step 302). EX1001, 8:51-55, Fig. 3. The compressed bitstream “can include video data generated based on any suitable coding standard, such as

Motion JPEG, JPEG, MPEG-2, MPEG-4, H.263, H.264, HEVC, etc.” EX1001, 8:35-46. “[T]he video data can [also] include encoded images, decoding parameters, header information, etc.” EX1001, 8:46-48. The encoded images may also include quantized transform coefficients. EX1001, 8:48-50.

68. After receiving the compressed bitstream, the transcoder decompresses the compressed bitstream and decodes the decompressed bitstream “in any suitable manner” to produce decoded video data (step 304). EX1001, 8:56-63, Fig. 3. The decoded video data may include “multiple decoded images based on a suitable coding standard, such as Motion JPEG, JPEG 2000, MPEG-2, MPEG-4, H.263, H.264, HEVC, etc.” EX1001, 8:56-63.

69. After receiving the compressed bitstream, the transcoder also receives media metadata from storage 110 or media metadata source 106 (step 306). EX1001, 10:16-21, Fig. 3. The media metadata “can include any suitable data and can be received in any suitable manner.” EX1001, 10:1-6. The 806 patent goes on to list several types of metadata that may be used, such as information about scene changes, picture complexity, motion data about the media content such as motion vector maps, and intra-prediction data, such as a set of candidate intra-prediction modes and the coding costs associated with each candidate mode. EX1001, 10:6-15. As noted above and discussed in more detail below, these kinds of metadata were well-known in the prior art. *Supra* § VII.A.3.

70. After receiving the media metadata, the transcoder 112 can encode the decoded video data “in any suitable manner” (step 308). EX1001, 10:22-25. For example, the decoded images may be encoded “into a set of encoded images based on any suitable coding standard, such as MPEG-2, MPEG-4, H.263, H.264, HEVC, etc.,” or “into a compressed bitstream including a set of encoded images that has a given bitrate,” or “a given resolution” (step 308). EX1001, 10:22-34.

71. The steps of the transcoding process described by Figure 3 were well-known in the prior art. As noted above, numerous transcoding methods were known years before the priority date of the 806 patent that converted the bitrate, resolution, and coding format of an input video. *Supra* § VII.A.2. Moreover, the 806 patent itself acknowledges that transcoding video data from one coding standard, bitrate, or resolution to another was known in the prior art. EX1001, 1:23-32.

72. Also, as was noted above and will be discussed more in detail below, using metadata in the transcoding process was well-known in the prior art. *Supra* § VII.A.3. For example, as noted above, the MPEG-7 Framework describes generating and using transcoding hints, or “meta-data that can be used to support transcoding applications.” *See, e.g.*, EX1010, 1; *see also, e.g.*, EX1009, 387.

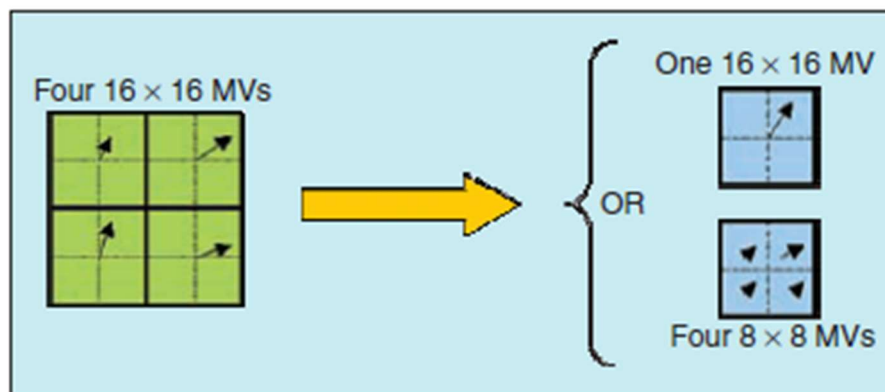
73. Additionally, I was informed that, during prosecution, the Applicant acknowledged that using metadata in transcoding was well-known. In response to a rejection of claim 1 on Anticipation grounds, the Applicant amended the claim to

recite “generating ... media metadata related to the source video file *prior to* decoding at least a portion of the source video file...” and argued that any prior art that discloses media metadata generation *after* decoding does not teach claimed limitation. EX1002, Response to OA dated July 9, 2019, at 9-10 (emphasis added). The Applicant did not argue that the anticipation reference did not disclose generating media metadata for use in transcoding. *Id.* Rather, the Applicant distinguished the claim from the anticipation reference based on when then generation of media metadata took place. *Id.* It is therefore clear that the Applicant did not view using metadata in transcoding as novel. Moreover, as was noted above and will be shown in more detail below, generating metadata prior to decoding was also well known. *Supra* § VII.A.3.

74. Also, as was noted above and will be discussed more in detail below, the types of metadata disclosed in the 806 patent and their use in the transcoding process were also well-known in the prior art. *Supra* § VII.A.3. For example, the 806 patent identifies scene change information such as information regarding the start and end of a scene as metadata. EX1001, 10:6-15. But it was well-known that metadata could include “storyboard information” or motion uncompensability data which could facilitate the transcoding process by delineating the start and end of a scene. EX1009, 391 (metadata may include “high-level features” such as “storyboard information”); EX1010, 3, Fig. 3 (describing the use of motion

uncompensability metadata to identify scene changes and thereby utilize metadata over defined video segments).

75. Also, for example, the 806 patent identifies motion data such as motion vector maps as metadata. EX1001, 10:6-15. But it was well-known that metadata could include motion vector data which could be used to inform the transcoding process. EX1009, 391 (metadata includes features of the video that describe motion); EX1007, 91 (explaining that motion vectors “can be easily obtained from the input video bit-stream” to facilitate re-encoding the bitstream); EX1004, 1923 (explaining that prior studies attempted to decrease computation complexity of transcoding by using motion vectors extracted from the original bitstream). For example, it was known that down-sampling macroblocks to reduce spatial resolution of a video may require “motion vectors to be mapped,” as shown in Figure 5 below:



▲ 5. Illustration of 4:1 and 1:1 motion vector mapping.

EX1015, 22.

76. Motion vector mapping is described in detail below:

When down-sampling four MBs to one MB, the associated motion vectors have to be mapped, where the number of motion vectors that may be associated with an MB depends on the standard being used and the coding tools available in a given profile or extension. Figure 5 illustrates the general problem of motion vector mapping. Several methods to perform the particular mapping illustrated in this figure have been described in past works [6], [7], [17], [19], [24]. To map from four motion vectors, i.e., one for each MB in a group, to one motion vector for the newly formed MB, a weighted average or median filters can be applied. This is referred to as a 4:1 mapping. However, with certain compression standards, such as MPEG-4 Visual [25] and H.263 [26], there is support in the syntax for advanced prediction modes that allow one motion vector per 8×8 luminance block. (It should be noted that the use of one motion vector per 8×8 block in the H.263 standard is supported in the extensions defined by the standard, i.e., this tool is not supported in the baseline specification.) In this case, each motion vector is mapped from a 16×16 MB in the original resolution to an 8×8 block in the reduced resolution MB with appropriate scaling by two. This is referred as a 1:1 mapping. While 1:1 mapping provides a more accurate representation of the motion, it is sometimes inefficient to use since more bits must be used to code four motion vectors. An optimal mapping would adaptively select the best mapping based on a rate-distortion criterion. A good evaluation of the quality that can be achieved using the different motion vector mapping algorithms can be found in [6], [7], and [19].

Id.

77. The 806 patent also explains that picture complexity can be used as metadata. EX1001, 10:6-15. But the use of picture complexity information to inform the transcoding process was well-known. For example, it was known that picture complexity could be used to control the bitrate of the output video stream, as described generally below:

Target bit allocation is the first step of rate control, which is the picture-level bit allocation. Before coding a picture, we need to estimate the number of bits available for coding the picture. The estimate is based on a linear model between the bit budget and the reciprocal of quantization step. This model is very intuitive since the number of bits generated by encoder is inversely proportional to the quantization step. Based on this model, the estimation is determined by several factors: the picture type, buffer fullness, and picture complexity. The picture complexity is measured by the product of the number of bits and the quantization parameter used for coding the same type of previous picture in the GOP[.]

EX1014, 87; *see also* EX1004, 1925 (“In our system, each transcoding PC determines the complexities of the first frames of the video segment from the source MPEG-2 coded data”); EX1005, 19:6-14 (describing the classification of video content based on motion complexity or spatial activity in a scene).

78. The 806 patent also describes metadata that may include “intra-prediction data” such as a set of candidate intra-prediction modes. EX1001, 10:6-15. But it was well-known that encoding can involve the encoder “select[ing] the best prediction” from among a set of intra-prediction modes. For example, it was known that the H.264 specifies prediction of the brightness (luma) and color (chroma) components of intra coded macroblocks as described generally below:

Intra prediction. In this mode, the luma component of a macroblock either remains a single 16×16 block or is split into small 4×4 partitions. In the former case, the standard specifies four methods (summarized later) for predicting the macroblock. In the latter case, the 16 elements of a 4×4 partition are predicted in one of nine modes from 12 elements of three previously-encoded partitions to its left and above it. These 12 elements are shown in the top-left part of Figure 9.44 where they are labeled A through H, I through L, and M. Either the x or the y

coordinate of these elements is -1 . Notice that these elements have already been encoded and reconstructed, which implies that they are also available to the decoder, not just to the encoder. The 16 elements of the partition to be predicted are labeled a–p, and their x and y coordinates are in the interval $[0, 3]$. The remaining nine parts of Figure 9.44 show the nine prediction modes available to the encoder. A sophisticated encoder should try all nine modes, predict the 4×4 partition nine times, and select the best prediction. The mode number for each partition should be written on the compressed stream for the use of the decoder.

EX1016, 914-915; *see also id.*, 916 (disclosing that in H.264, “[t]here are also four prediction modes for the 4×4 blocks of chroma components”); *id.*, 935 (explaining that “luma” describes brightness and “chroma” specifies color); EX1010, 2 (describing the use of motion uncompensability metadata to “determine[e] the mode selection”).

79. The partitioning of macroblocks and prediction modes from which the encoder selects “the best prediction” are shown in Figures 9.43 and 9.44 below:

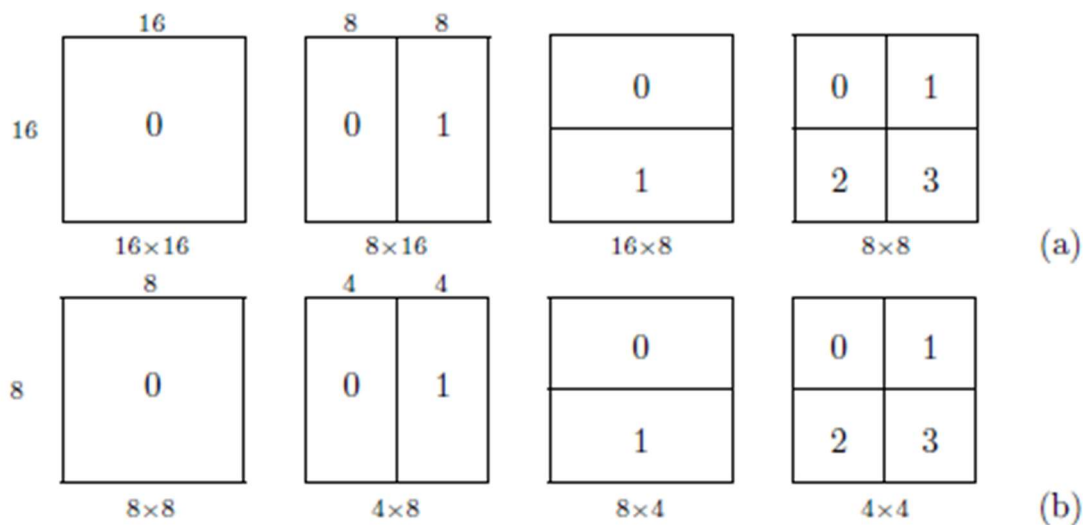


Figure 9.43: H.264 Macroblock Partitions.

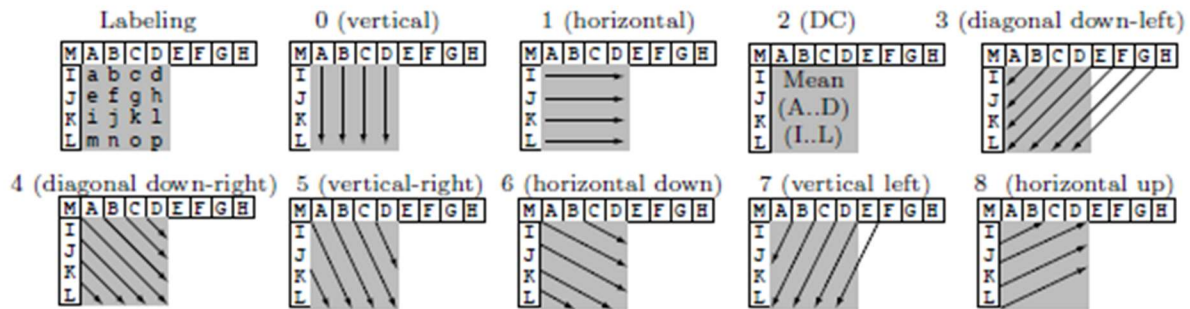


Figure 9.44: Nine Luma Prediction Modes.

EX1016, 915.

80. Figure 5 of the 806 patent, shown below, provides additional detail regarding the encoding step of the transcoding process described in Figure 3, describing a typical hybrid video encoding process. As explained above, all of the disclosed features and functionalities were known and conventional in the prior art.

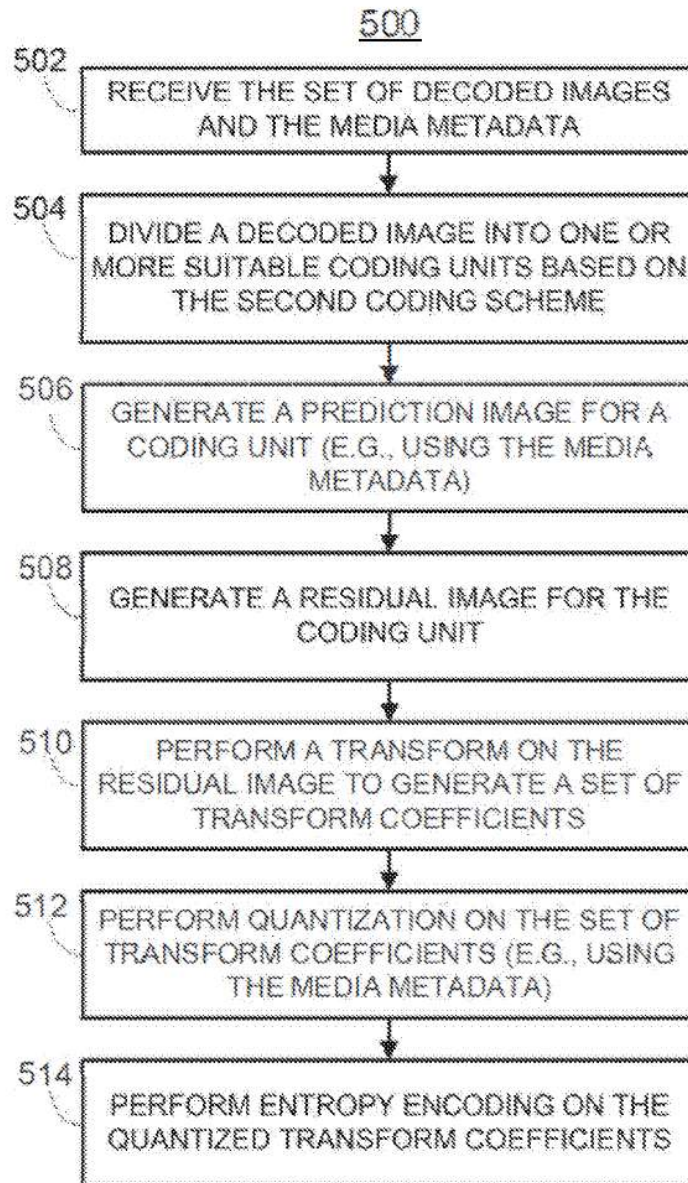


FIG. 5

EX1001, 10:35-39, Fig. 5; *Supra* § VII.A.1.

81. For example, Figure 5 discloses that the encoding circuitry can generate a prediction image for a coding unit of a decoded image based on media metadata (step 506). EX1001, 10:61-67, Fig. 5. But, as noted above, this was a well-known

technique in video encoding. *Supra* § VII.A.1; *see also, e.g.*, EX1010, 2 (“In case the motion uncompensability is near 0.0, the next frame can be predicted by a P or B frame.”); EX1016, 869-73, 911-913.

82. Also, for example, Figure 5 discloses that after generating a prediction image, the encoding circuitry can generate a residual image for the coding unit (step 508). EX1001, 11:48-49, Fig. 5. “The residual image can be generated in any suitable manner.” EX1001, 11:49-50. “For example, the residual image can be generated at 506 by subtracting the prediction image generated at from the original image of the coding unit.” EX1001, 11:50-53. But, as noted above, subtracting a predicted macroblock from a reference macroblock to produce a residual macroblock was well-known in the art. *Supra* § VII.A.1; *see also, e.g.*, EX1016, 869-73, 911-13 (“The prediction macroblock P is then subtracted from the current macroblock to produce a residual or difference macroblock Dn.”)

83. Also, for example, Figure 5 discloses that after generating a prediction image residual, the “encoding circuitry 204 can perform a transform on the residual image and generate a set of transform coefficients” (step 510). EX1001, 11:54-59, Fig. 5. The specification explains that “[t]he set of transform coefficients can be generated in any suitable manner.” *Id.*, 11:56-57. “For example, encoding circuitry 204 can perform a Discrete Cosine Transform (DCT) on the residual image and generate a set of DCT coefficients.” *Id.*, 11:57-59. But, as noted above, performing

transforms on residual macroblocks to produce a set of transform coefficients was a well-known video compression technique. *Supra* Section VII.A.1; *see also, e.g.*, EX1016, 869-73, 883-84, 910-13 (“The prediction macroblock P is then subtracted from the current macroblock to produce a residual or difference macroblock Dn. This macroblock is transformed and quantized to produce a set X of quantized transform coefficients.”); EX1006, [0034].

84. Also, for example, Figure 5 discloses that the encoding circuitry quantizes the transform coefficients using metadata (step 512). EX1001, 11:60-12:13, Fig. 5. The specification explains that “[t]he quantization can be performed in any suitable manner.” *Id.*, 11:61-62. “For example, encoding circuitry **204** can determine a suitable quantization parameter (QP) for a coding unit based on a target bitrate of the second coding scheme.” *Id.*, 11:62-65. But, as noted above, quantization of transform coefficients using metadata was well-known. *Supra* Section VII.A.1; *see also, e.g.*, EX1016, 869-73, 883-84; 910-13 (“The prediction macroblock P is then subtracted from the current macroblock to produce a residual or difference macroblock Dn. This macroblock is transformed and quantized to produce a set X of quantized transform coefficients.”); EX1009, 392; EX1010, 3-4 (disclosing the use of a transcoding hint to select a quantization parameter).

85. Also, for example, Figure 5 discloses that the encoding circuitry can perform entropy encoding on the quantized transform coefficients (step 514).

EX1001, 13:7-12, Fig. 5. “The entropy encoding can be performed in any suitable manner.” *Id.*, 13:8-9. “For example, encoding circuitry 204 can perform the entropy encoding using a suitable variable length encoding method.” *Id.*, 13:10-12. But as discussed above, entropy encoding was a well-known technique to compress video data using the least amount of bits possible. *See supra* Section VII.A.1; EX1016, 912 (explaining that quantized transform coefficients are “entropy encoded into a short bit string”); EX1006, [0037] ([t]he entropy coder 284 compresses the output of the quantizer 282”).

86. The claims of the 806 patent recite using metadata extracted prior to decoding to transcode a video file using multiple transcoders in parallel. EX1001, 13:49-59. As noted above, the use of metadata in transcoding was well-known, and it was also well-known that transcoding could be implemented by multiple transcoders working in parallel to enhance the efficiency of the transcoding process. *Supra* Section VII.A.3; *see also, e.g.*, EX1012, 3-4 (describing the implementation of multiple transcoders “combined with each other in parallel” because it was expected to be “more efficient and less delay-prone implementation” than a single transcoder); EX1011, 2905-06 (discussing the use of parallel transcoding “to optimize the transcoding speed”); EX1004, 1923, 1930-1931 (discussing the decrease in transcoding time achieved by using multiple transcoding PCs). Thus,

the use of parallel processing in transcoding applications would have been well-known to a POSITA prior to the earliest possible priority date of the 806 patent.

C. Priority Date

87. The 806 patent was filed as Application No. 16/298,345 (the “345 application”) on March 11, 2019. The 345 application is a continuation of application No. 15/905,695 (the “695 application”) filed on February 26, 2018, now patent No. 10,264,255 (the “255 patent”), which is a continuation of application No. 13/841,943 (the “943 application”), filed on March 15, 2013, now patent No. 9,906,785 (the “785 patent”). EX1001, cover. I understand that the 806 patent is therefore entitled to a priority date no earlier than March 15, 2013.

D. Level of Ordinary Skill in the Art

88. I understand that the 806 patent should be interpreted from the perspective of a person having “ordinary skill in the art” as of the effective filing date of the patent application (a POSITA). For the purposes of this declaration, I assume that the effective filing date of the patent application leading to the 806 patent is no earlier than March 15, 2013, and I understand that Patent Owner has not claimed an earlier priority date in the parallel litigation. I am familiar with the technological field at issue and the state of the art as of that date.

89. I understand that the education level of those in the technological field, sophistication of the technology, types of problems encountered, prior art solutions

to those problems, and the speed at which innovations are made in the field may factor into establishing what the appropriate level of ordinary skill should be.

90. In my opinion, a POSITA would have a bachelor's or master's degree in electrical engineering, computer engineering, or a related field, with specialized training or coursework in digital video coding and processing and related technologies. The individual would have at least four years of practical experience in digital video coding and processing, including software and/or hardware architectures used to perform digital video coding and processing. A person could also have qualified, as a POSITA, with more formal education and less technical experience, or vice versa.

91. I had at least the requisite level of skill with respect to the subject matter of the challenged patent at the time of the effective filing date as demonstrated by my qualifications listed above and my CV (EX1030).

VIII. PRIOR ART TO THE CHALLENGED PATENT

A. Summary of the Prior Art

1. Sambe

92. "High-Speed Distributed Video Transcoding for Multiple Rates and Formats" by Yasuo Sambe, Shintaro Watanabe, Dong Yu, Taichi Nakamura, and Naoki Wakamiya ("Sambe," EX1004), was published in IEICE Transactions on Information and Systems in August of 2005. EX1028.

93. I have been informed that even if the 806 patent were accorded the benefit of its earliest-claimed priority date, March 15, 2013, Sambe is prior art to the 806 patent under 35 U.S.C. §§ 102(a) and (b). I have been further informed that Sambe was not disclosed to the Patent Office during prosecution of the 806 patent. Sambe describes a transcoding system aimed at addressing the “growing demand to convert a pre-encoded MPEG-2 digital video in archives to other compressed formats such as MPEG- 1, MPEG-4, [and] H.263[.]” EX1004, 1923. The excerpt below provides more detail:

With the launch of digital broadcasting in many countries and the proliferation of digital video disks (DVD), both of which use the MPEG-2 video coding standard [1], it is expected that MPEG-2 will become the de facto video compression format in video archives. On the contrary, as the number of different video compression algorithms in use increases in the networked video application over the Internet, there is a growing demand to convert a pre-encoded MPEG-2 digital video in archives to other compressed formats such as MPEG- 1, MPEG-4, H.263 and so on.

Besides converting formats, video content will be altered in terms of bit-rate and resolution to meet the network bandwidth and terminal capability. For instance, the bandwidth of end user access networks can vary from several tens of kilo-bits per second to 20 to 30 megabits per second. Moreover, modern terminals use displays with various sizes and resolutions. Therefore, it is often necessary for service providers delivering video over the Internet to transcode the same content to yield different video formats, spatial resolution, and bit-rates simultaneously.

EX1004, 1923.

94. In view of the problems outlined above, Sambe’s goal “is to provide a video transcoding system that can convert MPEG-2 video files into other kinds of

formats and bit-rates at high-speed.” *Id.* To this end, Sambe proposes a distributed video transcoding system, as shown below in Figure 1, that divides video files into segments and transcodes them in parallel. The “distributed video transcoder” of Sambe includes a “Source PC,” a plurality of “Transcoding PCs,” and a “Merging PC.” *Id.*, 1924, Fig. 1. Upon request, the “Source PC” “divides the MPEG-2 video file into video segments—consisting of one or more consecutive Groups of Pictures (GOP)—of appropriate length” and transmits these segments to the [T]ranscoding PC[s]. *Id.*, 1923-24. It also uses the MPEG-2 data to determine re-encoding parameters—“transcoding parameters that specify the operation of the filter and the encoder of [the] transcoding PC[s]” and transmits them along with the video segments to the Transcoding PCs. *Id.* Sambe explains that “[t]hese parameters include [the] filter function specifications, spatial resolution, temporal resolution, and re-encoding formats desired.” *Id.* “Transcoded segments are [then] sent to the merging PC and concatenated to form the desired video format files at the merging PC.” *Id.*, 1924.

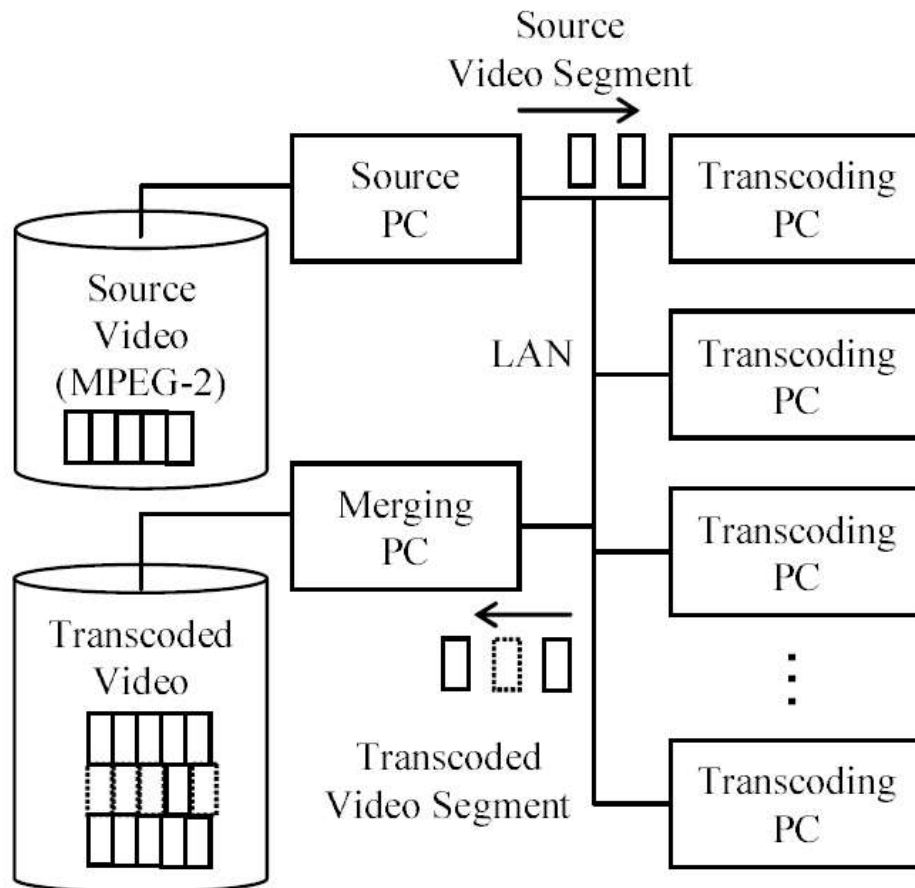


Fig. 1 Distributed video transcoder

Id., 1924, Fig. 1.

95. Upon receipt of the video segments and transcoding parameters, “[e]ach transcoding PC [as depicted below in Figure 2] decodes and re-encodes the video segments into the different formats specified.” *Id.*, Fig. 2.

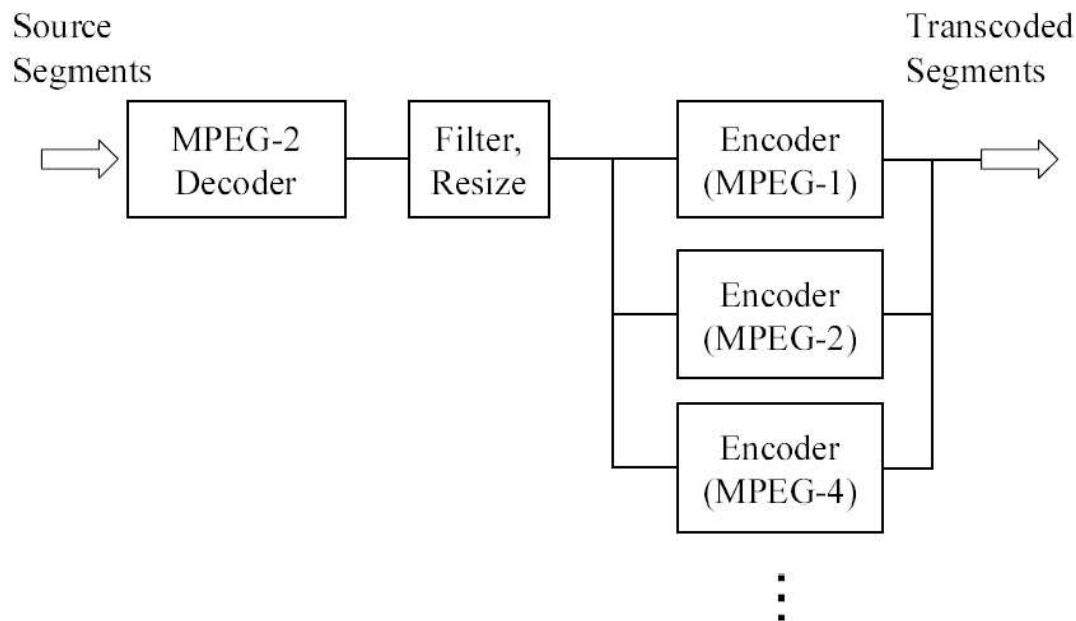


Fig. 2 Block diagram of a transcoding PC

96. Sambe explains that the “[d]ecoded frames are filtered, resized, and passed to the encoder *frame by frame*” providing the transcoding architecture with greater flexibility in transcoding. *Id.*, 1924 (emphasis added). “For example, new encoding modules or new filter operations like digital water marking can be easily added.” *Id.*

97. Further, all the modules shown in Figures 1 and 2, including the dividing, transmission, decoding, filtering, and encoding modules, are implemented by multi-thread programming. *Id.* Therefore, all these processes can run in parallel. *Id.* For example, as demonstrated by Figure 17 below, the Source PC can divide a video stream into at least segments 1 (red segment) and 2 (green segment), which are then transmitted to Transcoding PCs (1) and (2) respectively, which then

transcode their respective segments “in parallel.” *Id.*, 1923-24, 1929, Figs. 2, 17. After transcoding, encoded segments 1 (red) and 2 (green) are sent to the Merging PC where they are concatenated (stitched together) with the remaining transcoded segments to form the desired video file. *Id.*, 1923-24, 1929, Figs. 2, 17. Using a plurality of Transcoding PCs to process different segments of a video in parallel increased transcoding speeds by a factor of seven. *Id.*, 1923, 1930-31.

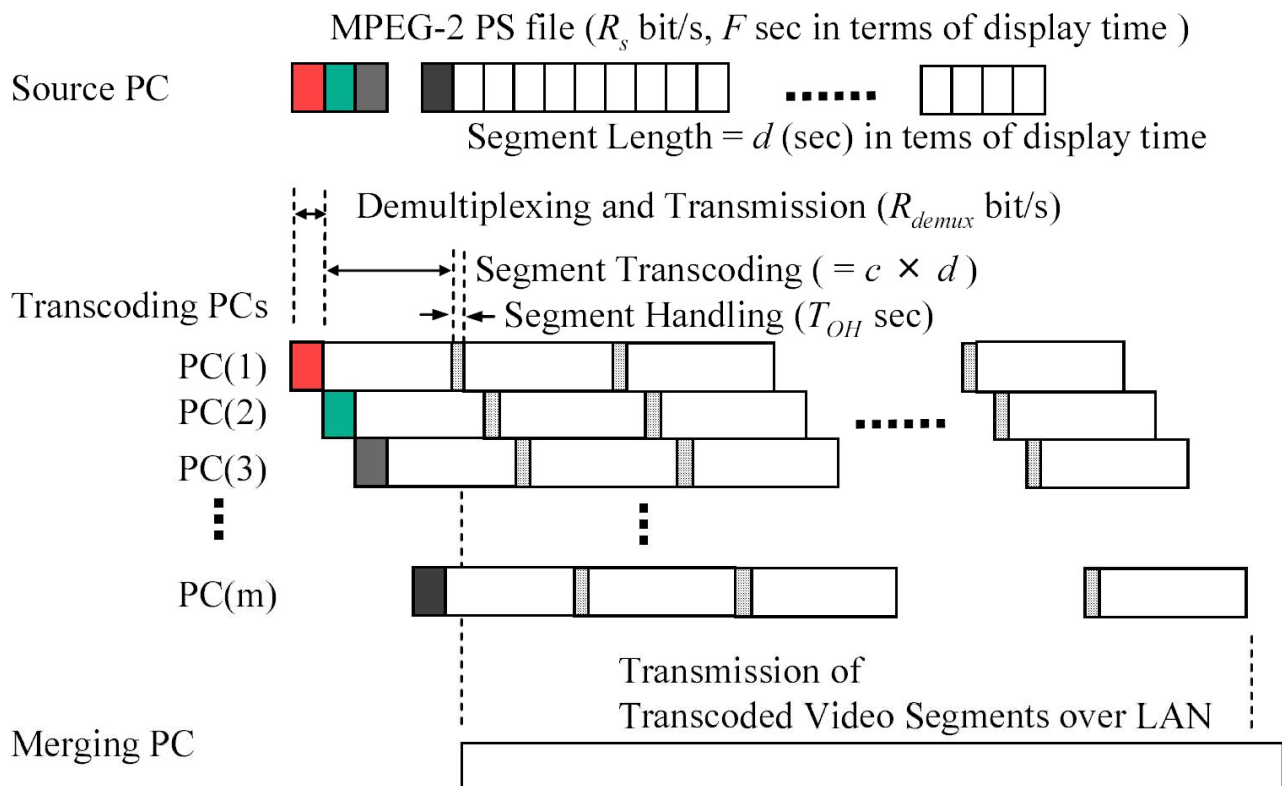


Fig. 17 Process flow of the distributed transcoding

Id., Fig. 17 (annotated).

98. Also, in addition to providing format, resolution, and bit-rate conversion, experimental results derived from testing of a Sambe distributed transcoding system with ten Transcoding PCs showed that transcoding times decreased by a factor of 7, a substantial speed up. *Id.*, 1923, 1930.

99. Sambe acknowledges that “[p]arallel transcoding along time axis usually suffers from quality discontinuity and degradation around the segmented cut points in the reencoded video, because of a lack of information such as the coding complexity of the previous video segment.” EX1004, 1923. “To get the information of the previous segment from another processor requires inter-processor communication, which leads to additional overhead and a significant performance degradation.” *Id.* “To achieve high performance without significant” quality degradation due to parallel transcoding,” Sambe proposes “[e]fficient video segment handling methods [] that minimize the inter-processor communication overhead and eliminate temporal discontinuities from the re-encoded video” and “investigate[s] how segment transcoding should be distributed to obtain the shortest total transcoding time.” *Id.*, 1923-30.

2. Vetro

100. U.S. Patent No. 6,490,320 B1 to Anthony Vetro *et al.*, titled “Adaptable Bitstream Video Delivery System,” filed for on April 11, 2000, and issued on December 3, 2002 (“Vetro,” EX1005), is a continuation-in-part of U.S. Patent

Application No. 09/504,323 which is a continuation-in-part of U.S. Patent Application No. 09/496,706.

101. I have been informed that even if the 806 patent were accorded the benefit of its earliest-claimed priority date, March 15, 2013, Vetro is prior art to the 806 patent under 35 U.S.C. §§ 102(a), (b), and (e). I have been further informed that Vetro was not disclosed to the Patent Office during prosecution of the 806 patent.

102. Vetro describes a system for transcoding compressed video. EX1005, Abstract. Vetro's system can use information contained in a bitstream to generate metadata, enabling the system to effectively transmit information contained in a bitstream while maintaining video quality despite reductions in the available bit rate. EX1005, 5:61-6:23. Specifically, Vetro describes an "an adaptable bitstream delivery system 300 [which] includes four major components: a content classifier 310, a model predictor 320, a content-network-device manager 330 and a switchable transcoder 340." *Id.*, 7:40-43, Figure 3, below, depicts the adaptable bitstream delivery system. *Id.*, Fig. 3.

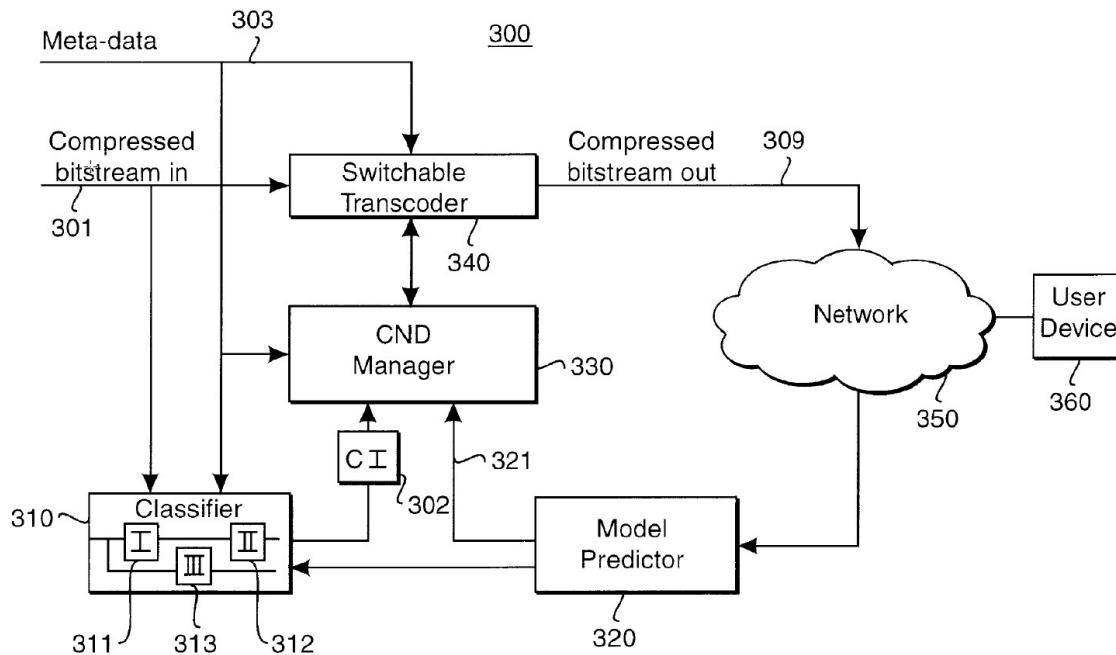


FIG. 3

103. The Content Classifier of Vetro generates media metadata from the look-head information provided by the input bitstream. Specifically, “[t]he content classifier 310 operates in three stages (I, II, and III 311-313).” *Id.*, 8:55-56. “First, [the content classifier] classif[ies] the bitstream content so that higher-level semantics can be inferred, and second, [the content classifier] adapt[s] the classified content to network and user device characteristics.” *Id.*, 8:56-59. Vetro then describes the function of the content classifier 310 as follows:

The main function of the content classifier is to map semantic features of content characteristics, such as motion activity, video change information and texture, into a set of parameters that are used to make rate-quality trade-offs in the content-network manager. To assist with this mapping function, the content classifier can also accept meta-data information 303. The meta-data can be low-level and high-level.

Examples of meta-data include descriptors and description schemes that are specified by the emerging MPEG-7 standard.

Id., 7:63-8:5.

104. Vetro then describes the operation of the content classifier 310 in more detail:

In the first stage (I) 311, we extract a number of low-level features from the compressed bitstream using conventional techniques, for example, motion activity, texture, or DCT coefficients. We can also access the meta-data 303, such as MPEG-7 descriptors and description schemes. If the meta-data are available, then less work needs to be performed on the compressed bitstream. As a final outcome of this first stage, a pre-determined set of content features are mapped to a finite set of semantic classes or high-level meta-data. Furthermore, within each semantic class, we differentiate based on the coding complexity, i.e., the complexity is conditional on the semantic class and network characteristics, and possibly device characteristics.

Id., 8:60-9:5.

105. Vetro further discloses that the content classifier can receive “metadata information 303” to assist with the content classifier’s function of mapping content characteristics into a set of parameters, i.e., metadata, used by the content-network device (“CND”) manager to make rate-quality trade-offs. *Id.*, 8:1-5, Fig. 3. This metadata is also sent directly to the CND manager and the switchable transcoder, and “include descriptors and description schemes that are specified by the [] MPEG-7 standard.” *Id.* The descriptors and description schemes specified in the MPEG-7 standard describe various features of video content, including the relative complexity of discrete segments of the video, referred to as “shots.” EX1005, 17:31-

19:22. Vetro’s “shots” are equivalent to scenes, as indicated by the 806 patent, which describes a “shot change” as an example of a “scene change[.]” EX1001, 4:36-40 (“the metadata can also include a set of parameters that can indicate the type of each of the scene changes, such as a shot change, a fading change, a dissolving change, etc.”). Likewise, Vetro describes “shots” as “a group of Frames (GOFs)” which “represent[] smaller segments of video that begin when a camera is turned on and last until the camera is turned off.” EX1005, 17:32-45 (describing a news program being partitioned into “a sequence of shots”); *see also* EX1010, 3, Fig. 3 (describing the use of motion uncompensability metadata to determine a “camera shot change *or* scene change”). This shot complexity information is mapped directly into metadata, which the CND manager uses to allocate bitrate among shots to achieve an optimal rate-quality objective. EX1005, 7:62-8:1, 21:8-49.

106. Vetro’s Content-Network Device (“CND”) Manager 330 utilizes the metadata received from the Content Classifier 310, referred to as “a set of parameters[,]” to provide transcoding strategy to the transcoder. Specifically, the content classifier 310 transmits the metadata to the CND manager 330. *Id.*, Fig. 3; *see also, id.*, 8:21-23, 9:6-19.

107. Figure 4, shown below, depicts an expanded version of the CND Manager and Vetro’s switchable transcoder.

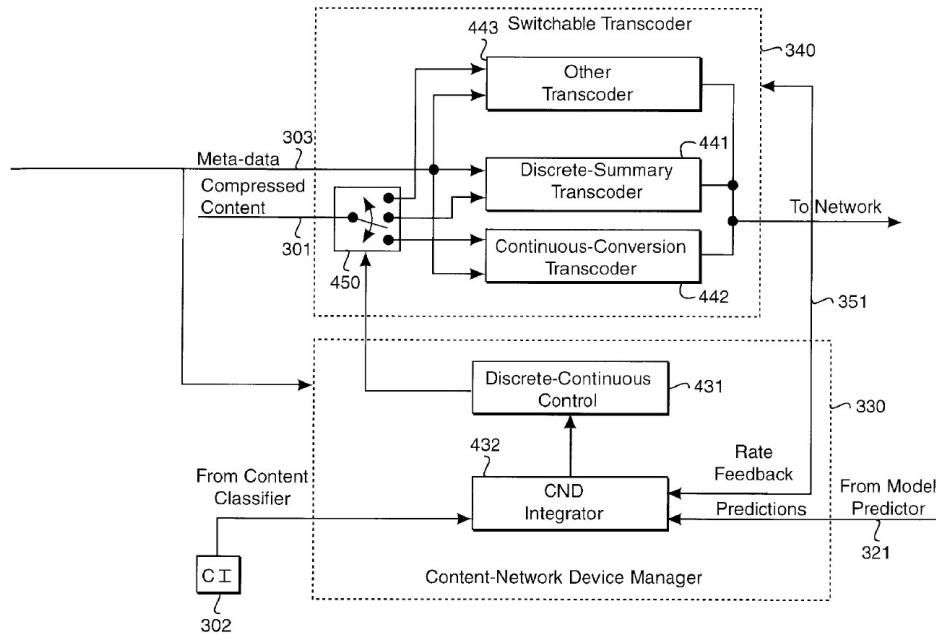


FIG. 4

108. “The CND manager includes a discrete-continuous control 431, and a content-network-device (CND) integrator 432,” and “transcoder 340 includes a plurality of transcoder[s] 441-443.” *Id.*, 9:26-30, Fig. 4. As depicted, the CND manager’s CND integrator 432 receives meta-data (such as MPEG-7 descriptors and description schemes), the generated metadata from the content classifier, referred to as “a set of parameters[,]” and input from model predictor 320, and combines them together to determine an optimal transcoding strategy for the switchable transcoder. *Id.*, 8:23-26, 8:63-64, 10:7-10, Fig. 4. The CND Integrator 432 “generates the model expressed as the rate-quality functions shown in [Figure] 5, or other like optimization functions.” *Id.*, 10:10-13. To form the optimal operating model 321, the CND integrator examines the generated metadata from the content classifier,

referred to as “mappings CI,” and the bit rate feedback output from the transcoder 340. *Id.*, 10:13-16. Vetro then explains how the CND integrator uses the metadata to form an optimal transcoding strategy:

Using this information, the integrator chooses the optimal modeling function 505 that has certain model parameters. The rate feedback 351 is used to dynamically refine the parameters. If the integrator finds that the chosen model is not optimal, then the integrator can decide to dynamically switch rate quality functions. Also, the integrator can track several functions for different objects or different bitstreams and consider the functions either separately or jointly.

Id., 10:16-24.

109. “To better understand how the control makes optimal selection decisions, [Figure 5, shown below,] graphs a plurality of rate-quality functions with respect to rate 501 and quality 502 scales.” *Id.*, 9:40-42, Fig. 5. Specifically, it depicts rate-quality functions for the discrete summary transcoder 441 (the linear function 504) and the continuous-conversion transcoder 442 (the convex function 503). *Id.*, 9:40-46, Fig. 5. The bold line shows “the optimal quality that can be achieved for a given bit rate and user device.” *Id.*, 9:56-57. The crossover point at $T=506$ shows that for target bit rates greater than T , the continuous-conversion transcoder 441 is preferable, while at rates less than T , it is better to use the discrete-summary transcoder. *Id.*, 9:58-63.

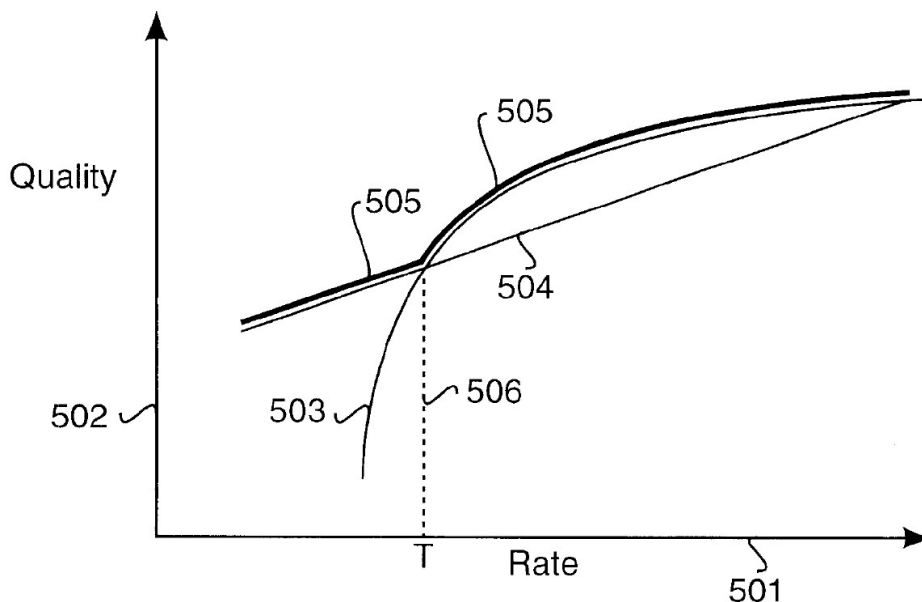


FIG. 5

Id., 9:40-46, Fig. 5.

110. But Vetro explains that “other transcoders can have different functions.” *Id.*, 9:46-47, Fig. 5. And “[i]t should be noted that these curves are only drawn for illustrative purposes” because “[t]he true forms of the functions for a particular transcoder can vary depending on the content, how the content has been classified and possibly the current state of the network and device constraining characteristics.” *Id.*, 9:48-53.

111. Vetro further discloses that the modeling functions are then passed to the CND Discrete-Continuous Control component 431, enabling it to determine how to transcode the input video. *Id.*, 9:31-47, Fig. 4.

The [CND Discrete-Continuous Control] 431, using a Switch 450, is responsible for deciding how the input compressed bitstream 301 should be transcoded, e.g., with the discrete summary transcoder 441, the continuous conversion transcoder, 442, or some other transcoder 443. The network-content manager also dynamically adapts to a target rate for the transcoder and considers resource constraining characteristics of the network and user device. These two very important items are decided by the control 431.

Id., 9:31-39.

112. Vetro's adaptive transcoding system also has an object "to generate variation and/or summary bitstreams 1308 from an original compressed bitstream (Video In) 1301." *Id.*, 23:8-30. Vetro explains that a "database will store a large number of [these] variations of the input bitstream 1301, so that in the future, a bit stream for some real-time operating condition will be readily available to the downstream transcoders." *Id.*, 23:50-54.

3. Gu

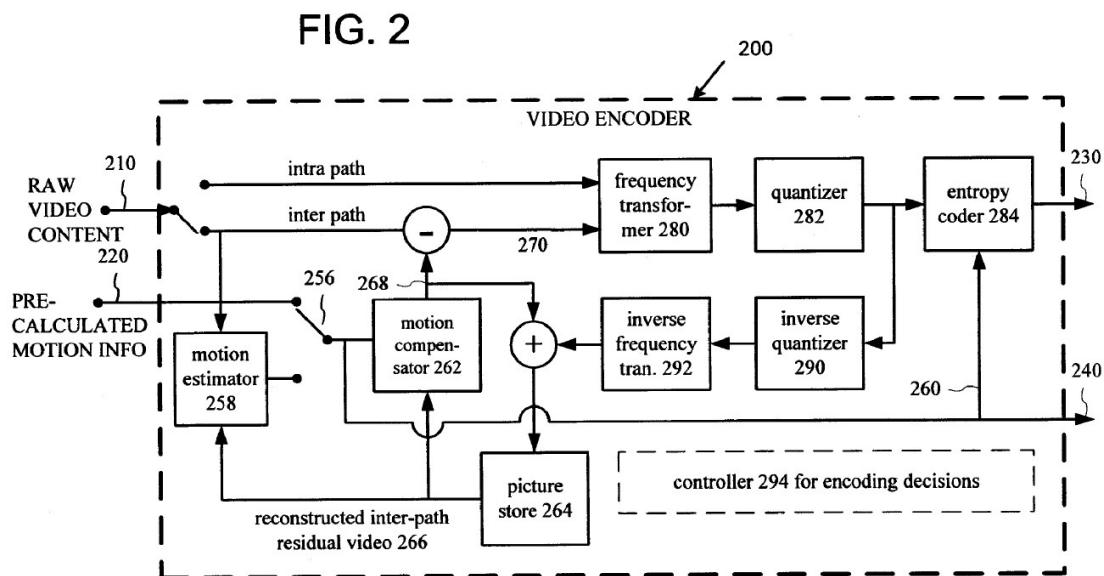
113. U.S. Patent Application Publication No. 2010/0189183 A1 to Chang Gu *et al.*, titled "Multiple Bit Rate Video Encoding Using Variable Bit Rate and Dynamic Resolution for Adaptive Video Streaming," filed for on January 29, 2009, and published on July 29, 2010 ("Gu," EX1006).

114. I have been informed that even if the 806 patent were accorded the benefit of its earliest-claimed priority date, March 15, 2013, Gu is prior art to the 806 patent under 35 U.S.C. §§ 102(a), (b), and (e). I have been informed that Gu was disclosed to the PTO during prosecution of the 806 patent but was not

substantively analyzed individually nor in combination with Sambe or Vetro. EX1006, Cover.

115. Gu describes systems and methods “for multiple bit rate video encoding, which are intended to make better use of the available bits with each bit rate so as to achieve generally higher quality video.” *See, e.g., id.*, [0007]. Specifically, the video encoding systems and methods of Gu use variable bit rate and dynamic resolution techniques to produce video streams at multiple bit rates for streaming. *Id.*, [0017], [0040]-[0063]. The goal of the Gu’s disclosed methods and systems is to provide better and more consistent visual quality. *Id.*, [0006]-[0009].

116. Figure 2 of Gu, shown below, depicts an implementation of a video encoder suitable “for producing multiple bit rate video streams using variable bit rate and variable resolution encoding” as described by Gu.



Id., [0027], Fig. 2.

117. The encoder of Figure 2 consists of standard components and processes: a motion estimator 258 and motion compensator 262 for generating prediction frames and/or macroblocks and prediction residuals for each frame and/or macroblock, a frequency transformer 280 for generating transform coefficients, a quantizer 282 for quantizing the transform coefficients, and an entropy coder 284 for encoding. *Id.*, [0028]– [0039]; *supra* § VII.A.1.

118. Gu explains that the input video content on the inter-path of the Figure 2 encoder “is encoded as a predicted picture based on motion information.” *Id.*, [0031]. During encoding, Gu explains that the video encoder can use pre-calculated motion information (if available), “which can be in the form of a set or sequence of motion vector for macroblocks or other sets of samples of the inter-path video picture with respect to one or more reference pictures.” *Id.*, [0031]. Or it can “choose to perform new motion estimation for the inter-path video content [] with motion estimator 258.” *Id.*, [0032]. “In general, the choice to use the pre-calculated motion information can be based on” “the availability of pre-calculated motion information” and “which and whether encoding parameters were changed from the previous calculation of the motion information and the parameters used for the current encoding of the video content.” *Id.*, [0031].

119. If the video encoder chooses to perform new motion estimation for the inter-path video content, Gu explains that “[t]he motion estimator 258 estimates

motion of macroblocks or other sets of samples of the video picture with respect to one or more reference pictures, which represent reconstructions of previously encoded video content frames.” *Id.*, [0032]. As a result, “[t]he motion estimator 258 outputs motion information 260 such as motion vector information.” *Id.* Gu further explains that “[t]he motion compensator 262 applies [the] motion vectors to certain reconstructed video content 266 (stored as reference picture(s)) when forming a motion-compensated current picture 268.” *Id.*, [0033].

120. Also, Gu describes calculating the prediction residual from a motion-compensated picture produced by motion compensator 262:

The difference (if any) between a block of the motion-compensated picture 268 and corresponding block of the original inter-path video picture is the prediction residual 270 for the block. During later reconstruction of the inter-path video frame (e.g., at a video decoder), reconstructed prediction residuals are added to the motion compensated residual video 268 to obtain reconstructed content closer to the original inter-path video 256.

Id.

121. Gu then describes generating transform coefficients from the prediction residuals:

A frequency transformer 280 converts spatial domain video information into frequency domain (i.e., spectral, transform) data. For block-based video content, the frequency transformer 280 applies a DCT, variant of DCT, or other forward block transform to blocks of the samples or prediction residual data, producing blocks of frequency

transform coefficients. The frequency transformer 280 may apply an 8×8, 8×4, 4×8, 4×4 or other size frequency transform.

Id., [0034].

122. Next, Gu discloses that the video encoder's quantizer 282 "quantizes the blocks of transform coefficients" on a frame-by-frame or macroblock-by-macroblock basis. *Id.*, [0035]-[0036], Fig. 2. Specifically, it "applies non-uniform, scalar quantization to the spectral data with a step size that varies spatially on a picture-by-picture basis, macroblock-by macroblock basis or other basis." *Id.*, [0035]. And Gu explains that "[t]he initial sample quantization step size" for the quantizer 282 of the Gu video encoder "can be chosen depending on the particular codec standard used by the encoder," as described below:

For example, in the case of the SMPTE 421-M video codec, the initial sample quantization step size may be chosen to be 4. In the case of that the video encoder uses the H.264 standard, an appropriate initial sample quantization step size may be 28. However, other initial sample quantization step sizes and resolutions can instead be selected.

Id., [0053].

123. Also, Gu describes the Gu encoder's entropy coder 284, which performs entropy encoding on "the output of the quantizer 282" resulting in a compressed bitstream as output. *Id.*, [0037]; *see also id.*, [0038]. Gu explains that the video encoding "can be performed according to a known video encoding standard, such as Windows Media Video format, SMPTE 421-M format, MPEG-x

format (e.g., MPEG-1, MPEG-2, or MPEG-4), H.26x format (e.g., H.261, H.262, H.263, or H.264), or other format.” *Id.*, [0026].

B. General Motivations to Combine Prior Art References

1. Sambe in view of Vetro

124. A POSITA would have been motivated to combine Sambe and Vetro. Specifically, a POSITA would have used Sambe’s distributed transcoding architecture, including a Source PC and multiple Transcoding PCs working in parallel, to achieve the high transcoding speeds disclosed in Sambe. *See supra* § VIII.A.1; EX1004, 1923-24, 1930-31, Fig. 1. This would fulfill a well-known and important design goal in the field of video transcoding, namely reducing transcoding delay. *See supra* § VII.A.3; *see also, e.g.*, EX1011, 2905 (describing the need for fast transcoding to meet the demand for transcoding into many different coding formats); EX1010, 1 (“[t]o avoid storing the content in different compressed representations for different network bandwidths and different AV-terminals, low-delay and real-time transcoding from one format to another is required”). A POSITA would have been motivated to modify Sambe’s distributed transcoding architecture with the teachings of Vetro’s transcoding system, to provide for the generation and use of metadata describing the coding complexity of video segments to be transcoded. EX1005, 7:40-43; 18:24-19:21; 21:8-21. This would provide for better bit allocation, which in turn would lead to consistent video quality, another key

design goal of transcoding systems. *See supra* § VII.A.3; *see also, e.g.*, EX1017, 47-48 (explaining that metadata describing the coding complexity of segments can be used for improved bit rate allocation among segments, resulting in significant improvements in subjective quality); EX1008, 794 (stating that a “basic requirement” of transcoding is that “the resulting video quality of the new bitstream should be as high as possible”); EX1010, 1 (stating that “it is critical that subjective quality be preserved” in transcoding applications). Thus, this combination would exploit both the high transcoding speed of Sambe’s distributed transcoding architecture and the improved understanding of coding complexity provided by Vetro’s transcoding techniques, resulting in better video quality.

125. The distributed transcoding system of Sambe consists of a “Source PC” which transmits video data to multiple “Transcoding PC[s]” that can perform transcoding steps in parallel, resulting in a significant speed improvement over sequential transcoding. EX1004, 1923-24, 1930-31. Indeed, Sambe’s “[e]xperimental results show the distributed transcoding system with 10 PCs decrease the total transcoding time by a factor of 7.” *Id.*, 1931. Additionally, Sambe proposes a segment handling method that allows for “high performance without significant quality degradation due to parallel transcoding” and further investigates scheduling algorithms and the segment length of the distributed transcoder to reduce overall transcoding time. *Id.*, 1923.

126. Sambe explains the motivating purpose of its segment handling method

below:

If each video segment is encoded independently, segment encoding quality may differ which leads to discontinuity around the segmentation points and irregular video quality. This is because re-encoding parameters of each segment are determined without regard to the coding complexity of the previous segment. The more complexity the video frame has, the more bits must be allocated to make coded video quality constant.

Id., 1925; *see also id.*, 1923.

127. To ensure consistent video quality at video segment boundaries, Sambe discloses that each Transcoding PC estimates the complexity of the first frames of their respective segments, and controls the rate allocated to each re-encoded frame based on the estimated frame complexity. *Id.* Specifically, Sambe employs the MPEG-2 Test Model 5 (“TM5”), and describes the process of estimating frame complexity below:

In the widely used MPEG-2 Test Model 5 (TM5) rate control [10], the target size of the frame is made proportional to complexity. In TM5, the frame complexity X_n is defined as the product of the coded frame size S_n (in bits) and the average quantization scale Q_n of the frame, where n denotes the coding picture type (I, P, B). The complexity of the frame to be coded is estimated as the same as that of the same type of previous coded frame. For example in Fig.3, in encoding B14, the complexity of the frame is estimated as the same as that of B12. By doing this, TM5 assures that the video quality keeps consistent. TM5 calculate the target frame size using the complexity. After deciding the target frame size, quantization scale Q of each macro-block is determined so that actual coded size will be equal to the target size using virtual buffer memory d_n . The virtual buffer d_n is the accumulation of difference between the actual size and the target size

of coded frames. The quantization scale Q is calculated as $Q = rd_n/31$, where r is reaction parameter which is defined as $r = 2 \text{ bit rate/frame rate}$.

Id., 1925.

128. As described above, the first B frame of a segment (B14) is estimated based on the last B frame of the previous segment (B12), as shown in Sambe's Figure 3 below:

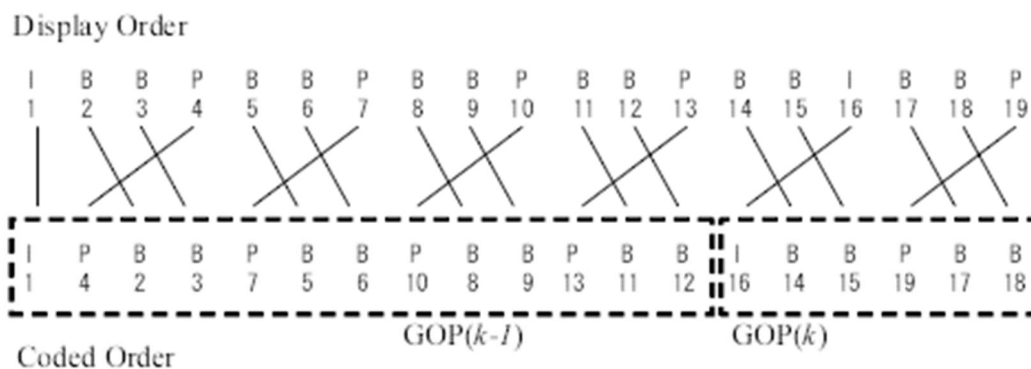


Fig. 3 Segmentation at Open-GOP

129. Sambe then describes how the TM5 framework is applied in its proposed segment handling method:

Therefore, if the X_n and d_n of the first frames in a segment can be estimated properly, the re-encoder can calculate the target bit budget and the initial quantization scale so that re-encoded video quality is made to be constant around segment cut point. In our system, each transcoding PC determines the complexities of the first frames of the video segment from the source MPEG-2 coded data. In fact, the complexities of the transcoded frames would be different from the input video due to the frame resolution and bit-rate change. However, since it has been shown that there are strong correlations between the input and output video [6], we employ the complexities of the input coded data, multiplying each of them by the ratio of output resolution to that of input. The virtual buffer memory d_n can't be estimated without actual

frame size of all re-encoded frames including those of frames on other transcoding PCs. However, d_n is expected to be stabilized within a few GOP transcoding. To shorten the stabilization period, d_n of the first frames is calculated from source MPEG-2 data as $d_n = 31Q_n/r$. The re-encoder of transcoding PC begins to transcode one GOP before segmentation point with the initial complexities and virtual buffer memory.

Id.

130. It was well-known that estimation of frame complexity based on the previous frame of the same type, as specified in the TM5 model and as shown in Sambe's Figure 3 above, is based on a "stationary assumption" that "[a] video sequence is stationary and all GOPs have similar characteristics, which is usually questionable." EX1007, 92; EX1022, 56. Furthermore, "it [was] well known that this estimation is poor when the stationary assumption fails." EX1007, 92. Thus, a POSITA would have looked for a way to avoid the need to *estimate* the complexity of the first frames in each segment to allocate bitrate, and would have turned to Vetro's teachings of extracting and using metadata describing the *actual* complexity of video segments referred to in Vetro as "shots." EX1005, 18:24-19:21.

131. More specifically, Vetro discloses "a system where the transcoding is according to features extracted from various levels of a video," i.e., metadata. *Id.*, 17:32-35. The highest level is the entire video program, and the next highest level is shot level. *Id.*, 17:35-41. As noted above, Vetro discloses that the input bitstream is partitioned into shots, which "can be a group of frames (GOFs)[,]" and that the

shots “represent[] smaller segments of video that begin when a camera is turned [on] and last until the camera is turned off.” *Id.*, 17:41-45; *supra* § VIII.A.2. Vetro explains that from each level, features are extracted that “represent ‘hints’ or ‘cues’ 905-908 that can be applied to the transcoding system,” and can “represent either high-level or low-level metadata.” *Id.*, 17:51-63. Vetro further describes that “[i]n general, meta-data for the higher level data, such as shot-level, are used for classification, bit allocation, and rate-quality considerations for that particular shot and among other shots.” *Id.*, 17:64-18:1. Vetro then explains that the content classifier 310 classifies the shots in a video program “into categories of similar motion activity and spatial distribution[,]” i.e., according to the complexity of each shot. *Id.*, 18:38-48. This classification process is described in more detail below:

To illustrate these rate-quality classes further, consider another set of three distinct classes as shown in FIG.11b. A first class 1121 indicates that the shot is very simple to compress, i.e., large compression ratios can easily be achieved for a given distortion. A third class 1123 represents the complete opposite, i.e., the content of the shot is very difficult to compress, either due to large/complex motion or a spatially active scene. A second class 1122 is somewhere in between the first and third classes. This set of classes will be referred to as SET-2 1120. As with the other set of classes 1110, these classes 1120 also illustrate the effects that content classification can have on the rate-quality decisions made by the CND manager 330 and how the switchable transcoder 340 can operate. As above, the compression difficulty can be classified on a numeric scale. It should be understood that other sets of classes can be defined for other types of video programs.

Id., 19:6-22.

132. Vetro further explains that this classification of shots according to their complexity allows for the CND manager to make informed rate-quality decisions, as described below:

In any case, whether the content classifier is given access to meta-data that described the content, or the classifier derives the data directly from the content itself, the utility of this information can directly be understood in view of the rate-quality trade-offs that the CND manager must ultimately make.

Id., 19:53-58.

133. More specifically, the CND manager uses these shot classifications to determine an optimal transcoding strategy for the switchable transcoder, as described in more detail below:

To the CND manager 330, these classifications suggest ways in which the content can be manipulated. In fact, classification can significantly reduce the number of scenarios to consider. For instance, if the CND manager has to consider the rate-quality trade-offs for multiple bit streams (frames or objects) at a given time instant, then the CND manager can consider the best way to distribute transcoding responsibility between continuous-conversion and discrete summary transcoding. Rather than choosing one way for all segments under consideration, it is also possible to consider a hybrid scheme. Priorities of the program, or compression difficulties according to its low-level features, are examples of useful parameters that can be used to make such decision.

Id., 20:27-39.

134. Based on the “strategy and instruction from the [CND] manager, the transcoder is responsible to manipulate the content in the best way possible.” *Id.*, 20:63-67. Through the combined functions of Vetro’s CND manager and switchable

transcoder, and using the shot complexity metadata extracted by the content classifier, an optimal allocation of bits for consistent video quality can be achieved, as described below:

There are two ways that meta-data can affect transcoding. Both are directly related to the bit allocation problem described above. The first way is in the CND manager 330 where the bit allocation is used to derive a strategy and ultimately a decision on how to use the functions provided by the Discrete-Summary and Continuous-Conversion Transcoders 441-442. In this way, the rate-quality functions of FIG. 5 are used for decision making. The second way is in the transcoder 340 itself. Again, the meta-data are used for estimation, but rather than making decisions on strategy, the meta-data are used to make real-time decisions on the coding parameters that can be used to meet the bit-rate objectives. In this way, the coding parameters are chosen so that the transcoders achieve the optimal rate-quality functions of FIG. 5.

Id., 21:8-21.

135. A POSITA would have been motivated to combine Sambe's teachings of a high-speed distributed transcoding system with Vetro's teachings of extracting and using metadata describing the complexity of "shots," or video segments. As noted above, Sambe acknowledges that when video segments are re-encoded independently, as they are by each of Sambe's transcoding PCs, it may result in irregular video quality due to a lack of accounting for the coding complexity of the previous segment. EX1004, 1925-26. To solve this, Sambe proposes an imperfect solution which estimates the complexity of the first frames in a segment based on frames in a previous segment. *Id.* While Sambe's parallel transcoding system is effective in speeding up the transcoding process, a POSITA would have known that

Sambe's frame complexity estimations may be inaccurate, resulting in improper bit allocation. EX1007, 91-92. A POSITA would have recognized that incorporating Vetro's extraction and use of shot complexity metadata would remove the need for *estimating* frame complexity, and instead provide the *actual* complexity of video segments. As disclosed in Vetro, the use of shot complexity metadata optimizes bitrate allocation for each shot and ensures consistent video quality. EX1005, 21:8-21, 6:20-22 (explaining that the Vetro's transcoding techniques "are able to achieve the target rate while maintaining the quality of the content of the bitstream"). Thus, a POSITA would have recognized that incorporating Vetro's use of metadata to determine the actual complexity of shots would improve bit rate allocation and video quality in Sambe's distributed transcoding system.

136. A POSITA would have had a reasonable expectation of success in combining these teachings of Sambe and Vetro because it would merely require incorporating Vetro's content classifier and CND manager into Sambe's Source PC and implementing Vetro's teachings regarding a switchable transcoder to improve Sambe's Transcoding PCs. The addition of Vetro's content classifier would provide Sambe's Source PC with the functionality to generate metadata describing a variety of information about the source video content including the complexity of "shots," i.e., the video segments that the Source PC divides for transcoding by each of Sambe's Transcoding PCs. EX1005, 5:61-6:23, 8:60-9:9, 9:25-10:24, 18:24-19:58.

Incorporating Vetro's CND manager would provide Sambe's Source PC with the ability to use the shot complexity metadata generated by the content classifier to determine an optimal transcoding strategy to ensure that each segment is allocated an appropriate number of bits based on its complexity to ensure consistent video quality upon playback. *Id.* Incorporating the teachings of Vetro's switchable transcoder into Sambe's Transcoding PCs would allow for improved rate-quality decisions to be achieved by allowing the CND manager to use the shot complexity metadata to choose between a discrete-summary or continuous control transcoding strategy, or a hybrid scheme. *See supra* § VIII.A.2; EX1005, 20:27-39.

137. A POSITA would have had a reasonable expectation of success in combining Vetro's content classifier and CND manager to Sambe's Source PC based on their knowledge of well-known and transcoding techniques such as those specified by the MPEG-7 standard, which defines a difficulty hint through which the complexity of a video segment is extracted from compressed video content and used for bit allocation. EX1017, 47-48. As noted above, this provides the same functionality performed by Vetro's content classifier and CND manager, respectively. *See supra* § VII.A.2; *see also, e.g.*, EX1017, 47-48 (stating that the MPEG-7 difficulty hint "describes the encoding complexity of segments within a video sequence" and "is used to calculate the bit budget" by "assign[ing] more bits to a difficult scene and remov[ing] bits from a relatively easy scene"). Furthermore,

a POSITA would have recognized that the content classifier should be incorporated into Sambe's Source PC, as this would ensure that all pre-processing of the source video, such as segmentation, determining transcoding parameters, generating metadata, and determining transcoding strategy for Sambe's Transcoding PCs, would occur only once at Sambe's Source PC rather than having to repeat some of these functions at each of the Transcoding PCs, which would be less efficient. *See also* EX1005, 5:31-32 (referring to the CND manager as a "transcoder manager").

138. A POSITA also would have had a reasonable expectation of success in modifying Sambe's Transcoding PCs in accordance with Vetro's teachings of a switchable transcoder. Sambe discloses that its Transcoding PCs were designed to be flexible and modular, explaining that its "transcoding architecture gives greater flexibility in transcoding" and that "new encoding modules or new filer operations like digital watermarking can be easily added." EX1004, 1924. Thus, Sambe expected that its Transcoding PCs could be customized with features such as those provided by Vetro's switchable transcoder. Therefore, a POSITA would understand that the combination of Vetro's content classifier and CND manager with Sambe's Source PCs, and the incorporation of Vetro's teaching regarding a switchable transcoder into Sambe's Transcoding PCs, could readily be implemented into Sambe's distributed transcoding architecture to improve its transcoding functions

without significantly impacting its design. There is no suggestion that this combination would not work in either Vetro or Sambe, or anywhere else in the art.

2. Sambe in view of Vetro and Gu

139. As discussed above, a POSITA would have been motivated to combine Sambe's distributed transcoding architecture with Vetro's transcoding techniques to exploit the high transcoding speeds provided by Sambe and the improved understanding of coding complexity provided by Vetro, resulting in better video quality. *Supra* § VIII.B.1. A POSITA would have further been motivated to incorporate the encoding techniques taught by Gu into the combination of Sambe and Vetro described above.

140. Sambe, Vetro, and Gu each disclose encoding processes including standard video coding techniques as described above, such as generating prediction images, performing transforms on residual images to generate sets of transform coefficients, quantizing the sets of transformation coefficients, and performing entropy encoding on these quantized transformation coefficients. *See, e.g.*, EX1004, 1925-27; EX1005, 12:4-16:67, 22:6-9; EX1006, [0031]-[0038]; *supra* § VII.A.1. Gu describes these steps explicitly, but a POSITA would have understood that Sambe and Vetro implicitly disclose these encoding steps, because as was noted above, these fundamental encoding steps were well-known and frequently used in the prior art to compress video for efficient storage and transmission by reducing

spatial and temporal redundancy. *Supra* §§ VII.A.1. Indeed, a POSITA would understand that most, if not all, conventional video encoding standards all contain these functions, as described below:

The popular international standards for video compression such as MPEG-2, MPEG-4 and H.264/AVC make use of the well-known ‘hybrid’ video encoder model in which a prediction, formed from previously-transmitted frames, is subtracted from the current frame to form a residual which is then transformed using a block transform such as the DCT, quantized and coded for transmission.

EX1013, 250; *see also id.*, 26 (explaining that popular compression formats including MPEG-2 Video, MPEG-4 Visual, H.263, H.264, and VC-1 are all based upon a model “that uses prediction and/or block-based motion compensation, transform, quantization and entropy coding”).

141. These techniques have been in existence since at least as early as 1970. EX1013, 250. Thus, for example, a POSITA would understand that Sambe’s disclosure of transcoding video to formats including MPEG-2 and MPEG-4 ASP would include the encoding steps explicitly disclosed in Gu because those steps are each part of the process of encoding video to those formats. *See* EX1004, 1926 (Table 1) (describing experimental conditions in which videos are transcoded into the MPEG-2 and MPEG-4 ASP formats); EX1013, 250, 26. But to the extent that Sambe in combination with Vetro do not explicitly disclose the conventional encoding steps such as using predictive techniques, transforms, quantization, and entropy encoding, a POSITA would have looked to Gu for clarification of these

processes. As noted above, transcoding has been described as a form of encoding, as explained in Xin:

What makes transcoding different from video encoding is that the transcoding has access to many coding parameters and statistics that can be easily obtained from the input compressed video stream. They may be used not only to simplify the computation, but also to improve the video quality. The transcoding can be considered as a special two-pass encoding: the “first-pass” encoding produces the input compressed video stream, and the “second-pass” encoding in the transcoder can use the information obtained from the first-pass to do a better encoding.

EX1007, 85.

142. Thus, a POSITA would have understood that Gu’s teachings regarding standard encoding steps are applicable to the combined transcoding system of Sambe and Vetro, and would have implemented those steps in the combined system to the extent that they are not explicitly disclosed in Sambe and Vetro. A POSITA would have had a reasonable expectation of success in implementing the conventional steps because they were each well-known and conventional steps in video coding.

IX. INVALIDITY

A. Ground 1: Claims 1-4, 7-14, and 17-21 of the 806 Patent are Rendered Obvious by Sambe in view of Vetro

1. Claim 1

[1.pre]: “A method for transcoding a source video file into a set of multiple alternate video streams, the method comprising:”

143. To the extent the preamble is considered limiting, both Sambe and Vetro individually render it obvious, as described below.

144. A POSITA would have understood that Sambe discloses a method for transcoding a source video file into a set of multiple alternative video streams of the source video file. Sambe “describes a distributed video transcoding system that can simultaneously transcode an MPEG-2 video file into various video coding formats with different rates.” EX1004, 1923.

145. Sambe also describes the growing demand for such a distributed transcoding system:

With the launch of digital broadcasting in many countries and the proliferation of digital video disks (DVD), both of which use the MPEG-2 video coding standard [1], it is expected that MPEG-2 will become the de facto video compression format in video archives. On the contrary, as the number of different video compression algorithms in use increases in the networked video application over the Internet, there is a growing demand to convert a pre-encoded MPEG-2 digital video in archives to other compressed formats such as MPEG- 1, MPEG-4, H.263 and so on.

Besides converting formats, video content will be altered in terms of bit-rate and resolution to meet the network bandwidth and terminal capability. For instance, the bandwidth of end user access networks can vary from several tens of kilo-bits per second to 20 to 30 megabits per second. Moreover, modern terminals use displays with various sizes and resolutions. Therefore, it is often necessary for service providers delivering video over the Internet to transcode the same content to yield different video formats, spatial resolution, and bit-rates simultaneously.

Id.

146. Sambe’s system includes a Source PC that breaks the source video down into segments and transmits the segments and their associated transcoding parameters to a plurality of Transcoding PCs to be transcoded. Sambe further

explains that each Transcoding PC receives a segment and transcodes it before sending it to a Merging PC that concatenates the segments from each Transcoding PC, i.e., stitches the segments together, resulting in multiple transcoded video files with “various video coding formats with different rates.” See EX1004, 1923-24, Figs. 2, 17. A POSITA would have understood that the transcoded video files with “various video coding formats with different rates” are a set of multiple alternative video streams of the source video file as recited in the preamble of claim 1.

147. Similarly, Vetro’s adaptive transcoding system has an object “to generate variation and/or summary bitstreams 1308 from an original compressed bitstream (Video In) 1301.” EX1005, 23:8-30. Vetro explains that a “database will store a large number of [these] variations of the input bitstream 1301, so that in the future, a bit stream for some real-time operating condition will be readily available to the downstream transcoders.” *Id.*, 23:50-54. A POSITA would have understood the generated bitstreams from an original compressed bitstream to be multiple alternate video streams.

[1.a]: “generating, at a computer system configured as a media metadata generation device, media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene complexity information;”

148. The combination of Sambe in view of Vetro, as described below, renders limitation 1[a] obvious.

149. Sambe’s “distributed video transcoder” includes a “Source PC” that “divides in-coming MPEG-2 data with minimum duplication and the data are used to determine re-encoding parameters.” EX1004, 1923-24. The Source PC then transmits video segments along with their associated transcoding parameters (the re-encoding parameters) to a plurality of Transcoding PCs that each transcode a segment. *Id.*, 1924. The transcoding parameters sent by the Source PC to the Transcoding PCs “specify the operation of the filter and the encoder of [the] Transcoding PC” and “include filter function specifications, spatial resolution, temporal resolution, and re-encoding formats desired.” *Id.*

150. To the extent that Sambe does not explicitly disclose that its Source PC generates “media metadata [that] comprises scene complexity information” during transcoding but prior to decoding, Vetro does disclose this limitation. Specifically, Vetro discloses a computer system that includes a “Content Classifier 310” that generates media metadata that classifies “shots” in the source video file according to their complexity, during transcoding but before decoding takes place. EX1005, 7:40-43, 7:62-8:26, 17:32-19:58, Fig. 3. More specifically, the content classifier maps characteristics of the input video, including “motion activity, video change information, and texture,” to a set of parameters used to optimize trade-offs between bitrate and quality that guide the transcoding process, as described below:

The content classifier 310 extracts content information (CI) 302 from the input bitstream for the manager. The main function of the content

classifier is to map semantic features of content characteristics, such as motion activity, video change information and texture, into a set of parameters that are used to make rate-quality trade-offs in the content-network manager. To assist with this mapping function, the content classifier can also accept meta-data information 303. The meta-data can be low-level and high-level. Examples of meta-data include descriptors and description schemes that are specified by the emerging MPEG-7 standard.

Id., 7:62-8:5.

151. Vetro then explains that as part of generating transcoding parameters from content information and metadata, the content classifier categorizes the input bitstream according to its complexity, as described below:

In the first stage (I) 311, we extract a number of low-level features from the compressed bitstream using conventional techniques, for example, motion activity, texture, or DCT coefficients. We can also access the meta-data 303, such as MPEG-7 descriptors and description schemes. If the meta-data are available, then less work needs to be performed on the compressed bitstream. As a final outcome of this first stage, a pre-determined set of content features are mapped to a finite set of semantic classes or high-level meta-data. Furthermore, within each semantic class, we differentiate based on the coding complexity, i.e., the complexity is conditional on the semantic class and network characteristics, and possibly device characteristics.

Id., 8:60-9:5.

152. Vetro then provides detailed disclosure of how “the spatial distribution parameters of the motion activity descriptor in MPEG-7 enable classification of the video segments of a program into categories of similar motion activity and spatial distribution.” *Id.*, 18:45-48; *see also id.*, 18:24-19:58. More specifically, Vetro

explains how “shots” (scenes in a video) are classified according to their coding complexity:

To illustrate these rate-quality classes further, consider another set of three distinct classes as shown in FIG. 11b. A first class 1121 indicates that the shot is very simple to compress, i.e., large compression ratios can easily be achieved for a given distortion. A third class 1123 represents the complete opposite, i.e., the content of the shot is very difficult to compress, either due to large/complex motion or a spatially active scene. A second class 1122 is somewhere in between the first and third classes. This set of classes will be referred to as SET-2 1120. As with the other set of classes 1110, these classes 1120 also illustrate the effects that content classification can have on the rate-quality decisions made by the CND manager 330 and how the switchable transcoder 340 can operate. As above, the compression difficulty can be classified on a numeric scale.

Id., 19:6-22.

153. As noted above, this classification of shots based on coding complexity allows for Vetro’s CND manager to make informed decisions on rate-quality trade-offs in transcoding. *See also id.*, 19:53-58 (“whether the content classifier is given access to meta-data that described the content, or the classifier derives the data directly from the content itself, the utility of this information can directly be understood in view of the rate-quality trade-offs that the CND manager must ultimately make”).

154. A POSITA would have understood the parameters generated by the content classifier to be “media metadata” because they describe the “characteristics” and “features” of the input bitstream. EX1005, 7:62-8:5, 8:60-9:5; EX1001, 4:25-

26 (stating that “metadata for media content can include any suitable information about the media content”). Furthermore, the parameters generated by Vetro’s content classifier “comprise[] scene complexity information” because they classify shots “into categories of similar motion activity and spatial distribution,” i.e., according to the complexity of each shot. EX1005, 8:60-9:5, 18:24-19:58. Vetro’s shots constitute scenes as recited in this limitation because they comprise partitioned “segments” of a video program “that begin when a camera is turned [on] and last until the camera is turned off.” *Id.*, 17:32-46. Vetro provides an example in which a news program includes “three shots 1201-1203, an anchor person shot, a reporter on scene shot, and a police chase shot.” *Id.*, 18:52-58; *see also supra* § VIII.A.2. It would be readily apparent to a POSITA that these three shots amount to scenes in the news program. Thus, Vetro’s content classifier generates media metadata comprising scene complexity information.

155. Vetro’s content classifier generates the transcoding parameters, i.e., the media metadata comprising scene complexity information, “prior to decoding, during a transcoding of, at least a portion of the source video file,” as recited in this limitation. The content classifier generates the media metadata as part of the transcoding process, as discussed above. *Supra* § VIII.A.2. More specifically, metadata is passed to the CND manager and “in part, characterizes the potential performance” of Vetro’s switchable transcoder. EX1005, 9:6-9. Thus, the content

classifier's generation of metadata is specifically tied to the transcoding process, and a POSITA would understand that the generation of media metadata is part of the transcoding process. The CND manager then uses the media metadata to "pre-select the best strategy for the transcoder," which includes determining whether to transcode the source video file using the discrete-summary transcoder or the continuous-conversion transcoder within Vetro's switchable transcoder. *Supra* § VIII.A.2; EX1005, 9:25-10:24, 19:53-58, 20:63-64, 21:8-15. As discussed above, the CND manager makes this decision based on whether the discrete-summary transcoder or the continuous-conversion transcoder will provide better video quality for a target bitrate, as demonstrated in Vetro's Figure 5. *See supra* § VIII.A.2, EX1005, 9:40-63. Because the CND manager uses the media metadata to determine which transcoder to select and the source video file is not decoded until it reaches the selected transcoder, it would be clear to a POSITA that Vetro's content classifier necessarily generates the media metadata prior to decoding. *See, e.g.*, EX1005, Fig. 1 (depicting the decoder as part of the transcoder).

156. A POSITA would have been motivated to incorporate Vetro's content classifier into Sambe's distributed transcoding architecture. More specifically, a POSITA would have recognized the benefit of incorporating Vetro's content classifier that generates parameters, i.e., media metadata describing scene complexity, into Sambe's Source PC. This combination would enhance bit

allocation among video segments based on their coding complexity and ensure consistent video quality, a stated goal of Sambe and especially important consideration in systems like Sambe's that are designed to increase transcoding speed. *Supra* § VIII.B.1; EX1005, 21:8-56 (describing how scene complexity metadata can be used for bit allocation and improved rate-quality decisions); EX1004, 1923 (explaining that quality discontinuity is a known problem in parallel transcoding systems); EX1015, 19 (explaining that in the "quest for efficiency" that has driven most research activities in the field of transcoding, "[o]f course, any gains in efficiency should have a minimal impact on the quality of the transcoded video"). A POSITA would have also recognized that Vetro's content classifier should specifically be incorporated into Sambe's Source PC, because that way, all pre-processing of the source video, such as segmenting, determining transcoding parameters, generating media metadata, and determining transcoding strategy, would occur only once at Sambe's Source PC, rather than having to duplicate some of these functions at each of Sambe's Transcoding PCs, which would result in greater cost and inefficiency.

157. As noted above, the combination of Vetro's content classifier with Sambe's Source PC would merely require a simple combination of known elements according to known methods to achieve predictable results, and use of known techniques to improve a similar method in the same way. *See supra* § VIII.B.1. As

discussed above, Sambe's Source PC is already responsible for determining re-encoding parameters (i.e., transcoding parameters) from the source video file, which "specify the operation of the filter and the encoder of [the] transcoding PC[s]." EX1004, 1924. The addition of Vetro's content classifier that generates scene complexity metadata to Sambe's Source PC could be accomplished using well-known methods such as those specified by the MPEG-7 standard, which defines a difficulty hint that "describes the encoding complexity of segments within a video sequence" and a "motion activity descriptor" that "enables classification of the video segments of a program into categories of similar motion activity and spatial distribution." EX1017, 48 (describing the use of a difficulty hint for bit allocation and improved video quality); EX1005, 18:45-19:22 (describing the use of the motion activity descriptor to classify segments into according to how difficult they are to compress); *see also supra* §§ VII.A.3, VIII.B.1. Therefore, this would be a reasonable modification that provides the expected results—that Vetro's content classifier incorporated into Sambe's Source PC would generate parameters (i.e., media metadata) to be used to determine how the source video file will be transcoded. EX1004, 1923-24.

[1.b]: "providing information based on the media metadata from the computer system to a plurality of transcoding devices; and"

158. The combination of Sambe in view of Vetro renders limitation 1[b] obvious, as described below.

159. As discussed above with respect to limitation [1.a], Sambe's Source PC determines re-encoding parameters (transcoding parameters) from the source video file that "specify the operation of the filter and the encoder of [the] transcoding PC[s]" and transmits them to a plurality of transcoding PCs. The Source PC then sends the transcoding parameters along with video segments to each of the plurality of Sambe's Transcoding PCs to be transcoded. *Supra* § IX.A.1 (limitation [1.a]); EX1004, 1924.

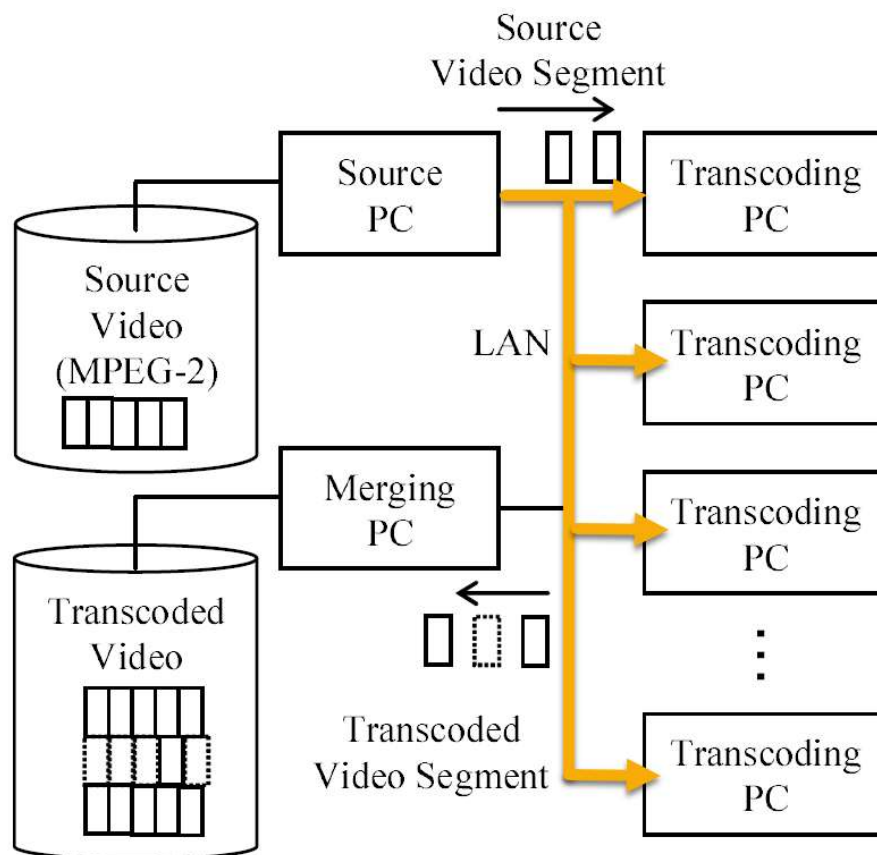


Fig. 1 Distributed video transcoder

EX1004, Fig. 1 (annotated).

160. To the extent that Sambe does not explicitly disclose providing information based on media metadata to a plurality of transcoding devices, Vetro does. As discussed above, Vetro discloses a content classifier that generates media metadata (referred to as a set of parameters) that are used as transcoding strategy for Vetro's switchable transcoder. *See supra* § IX.A.1 (limitation [1.a]). As shown below in Figure 4, Vetro's content classifier transmits the media metadata (as depicted by the green arrow below) to the Content-Network Device manager 330 (the yellow box below).

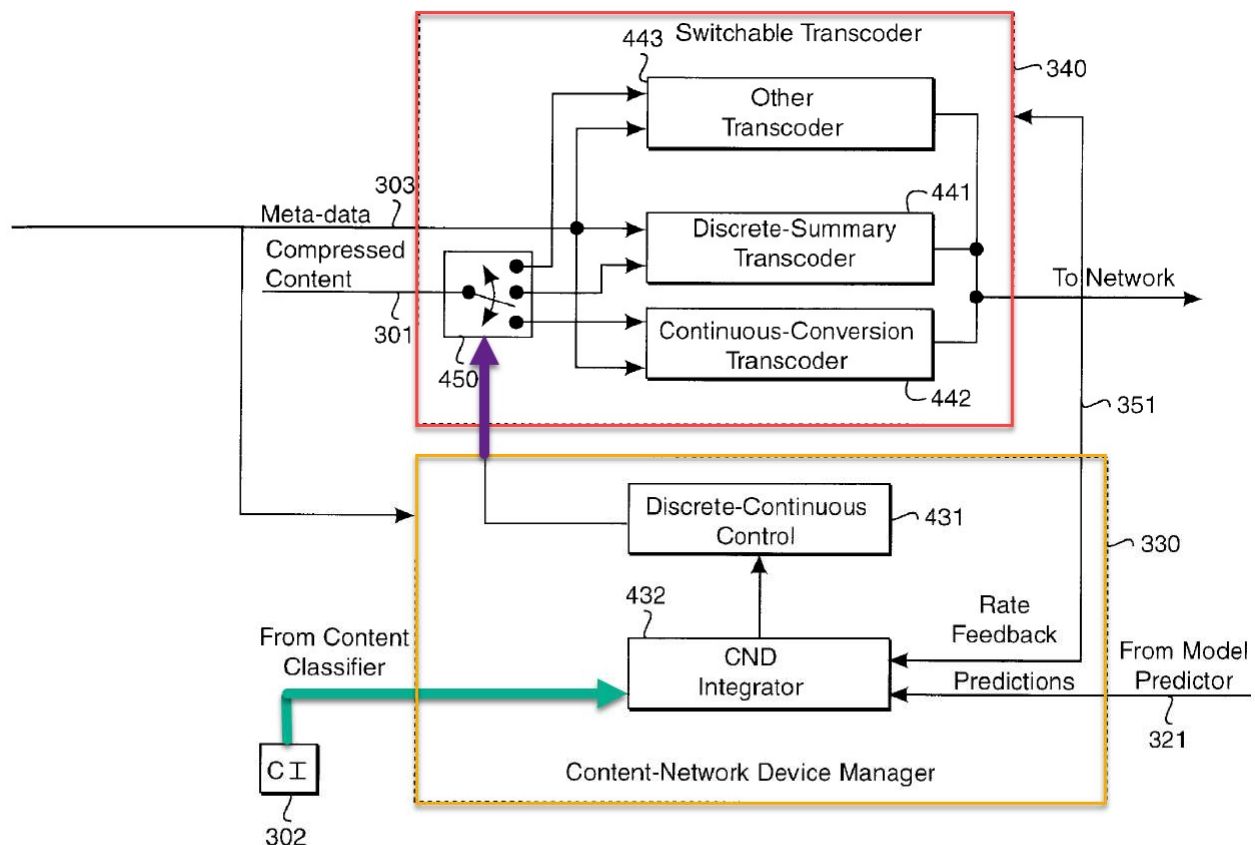


FIG. 4

161. The CND manager's CND integrator 432 receives meta-data (such as MPEG-7 descriptors and description schemes), the generated metadata from the content classifier, referred to as "a set of parameters[,]" and input from model predictor 320, and combines them together to determine an optimal transcoding strategy for the switchable transcoder. EX1005., 8:23-26, 8:63-64, 10:7-13, Fig. 4. The CND Integrator 432 then "generates the model expressed as the rate-quality functions shown in [Figure] 5, or other like optimization functions." *Id.*, 10:10-13. These functions are then passed to the CND Discrete-Continuous Control 431 component, enabling it to determine how to transcode the input video—the "transcoding strategy." *Id.*, 9:31-47, Fig. 4 (annotated). Specifically, the CND Discrete-Continuous Control 431 component, "using a switch 450, is responsible for deciding how the input compressed bitstream 301 should be transcoded, e.g., with the discrete summary transcoder 441, the continuous conversion transcoder, 442, or some other transcoder 443." *Id.*, 9:31-35, Fig. 4. In addition to determining how the input compressed bitstream 301 should be transcoded, "[t]he network-content manager also dynamically adapts to a target rate for the transcoder and considers resource constraining characteristics of the network and user device." *Id.*, 9:35-39, Fig. 4. Both of "[t]hese two very important items are decided by the control 431." *Id.*, 9:38-39, Fig. 4 (annotated). The processed strategy information (as depicted by the purple arrow above) is then passed to the switchable transcoder (depicted in the

red box above), which the transcoder uses as transcoding strategy for transcoding the “shots” (scenes in a video) disclosed in Vetro. EX1005, 8:21-8:26, Figs. 3, 4; *supra* § IX.A.1 (limitation [1.a]) (explaining that a POSITA would understand Vetro’s “shots” to be video scenes).

162. As an example, the content classifier may determine that a given shot, such as one depicting a news anchor sitting behind their desk, is “very easy to compress” and classifies it as such, then passes the classification to the CND manager. EX1005, 19:40-20:52. The CND manager uses this scene complexity metadata to “pre-select the best strategy for the transcoder 340.” *Id.*, 20:63-67. In the example of a news anchor, the CND manager may determine that the best option is to use the Discrete Summary Transcoder 441 to reduce the entire shot to a single frame, based on “resource constraining characteristics of the network and user device.” *Id.*, 9:31-39, 18:49-19:58. As discussed above, the CND manager determines the number of bits to allocate for each shot using a rate quality-function that derives the optimal trade-off between bitrate and video quality for the shot, and determines whether to use the discrete summary transcoder or the continuous conversion transcoder. *Supra* § VIII.A.2; EX1005, 9:40-63, Fig. 5. The transcoding strategy is then passed to the transcoder’s switch 450, which in turn passes this “strategy and instruction” to the transcoder, which “is responsible to manipulate the content in the best way possible[,]” i.e., in accordance with the transcoding strategy

provided by the CND manager. *Id.*, 18:49-20:67; *see also id.*, 9:31-35. In this way, the media metadata generated by Vetro's content classifier is used by Vetro's CND manager to determine how to allocate bits among shots based on their relative complexity.

163. A POSITA would have been motivated to incorporate Vetro's transcoding system into Sambe's distributed transcoding architecture, including by incorporating Vetro's CND manager into Sambe's Source PC. In this combined configuration, Vetro's content classifier incorporated into Sambe's Source PC would generate media metadata comprising scene complexity information, and Vetro's CND manager incorporated into Sambe's Source PC would send information based on the media metadata, i.e., the transcoding strategy determined by the CND manager, to Sambe's plurality of Transcoding PCs along with a video segment that they each respectively transcode. This combination would permit the Transcoding PCs to determine how to optimally allocate bits to each video segment to ensure consistent video quality, as explained in Vetro. *Supra* §§ VIII.A.2, VIII.B.1; EX1005, 9:40-63, Figs. 3-5. In addition, this configuration would ensure that all processing of the source video, such as breaking it down into segments, determining transcoding parameters, generating media metadata, and determining a transcoding strategy, would take place only once at Sambe's Source PC, rather than

having to duplicate some of these functions at each of the plurality of Transcoding PCs, resulting in less cost and greater efficiency.

164. Incorporating Vetro's content classifier and CND manager into Sambe's distributed transcoding architecture would merely require a simple combination of known elements using known methods to achieve predictable results, and use of a known technique to improve a similar method in the same way. *See supra* § VIII.B.1. As noted above, it was well-known that metadata describing the complexity of video scenes could be used in transcoding to improve bit allocation among scenes based on their complexity, resulting in improved video quality. *See, e.g.,* EX1017, 47-48 (describing the use of a difficulty hint specified by the MPEG-7 standard which "describes the encoding complexity of segments within a video sequence" and can be used to "assign more bits to a difficult scene and remove bits from a relatively easy scene," with the result being that "the overall subjective quality is significantly improved"). It was also well-known that media metadata could be used to "assist the transcoder in (re-)encoding parameter decisions to preserve visual quality through the transcoding process." EX1018, 310. As discussed above, Sambe's Source PC is already responsible for determining re-encoding parameters (i.e, transcoding parameters) from the source video file that "specify the operation of the filter and the encoder of [the] transcoding PC[s]" and transmitting them to the transcoding PCs. EX1004, 1924; *see also supra* § IX.A.1.

(limitation [1.a]). Thus, a POSITA would have a reasonable expectation of success in applying known techniques of using information derived from scene complexity metadata to inform the transcoder on how to transcode video data, to achieve expected results—that Sambe’s Transcoding PCs would take the strategy information derived from Vetro’s content classifier and CND manager into account when transcoding the input video.

[1.c]: “performing the following at each of the plurality of transcoding devices in parallel:”

165. The combination of Sambe in view of Vetro as described below renders limitation 1[c] obvious. As summarized above, Sambe discloses transcoding using multiple Transcoding PCs in parallel. *Supra* § VIII.A.1; EX1004, 1924. Based on transcoding parameters sent from the source PC, each transcoding PC may perform a variety of different steps in parallel. *Id.* The following subsections explain why the combination of Sambe and Vetro further discloses each of the claimed processes.

[1.c(i)]: “receiving the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;”

166. Sambe, as described below, discloses limitation 1[c(i)]. Sambe’s parallel Transcoding PCs receive the video segments, each of which is a portion of the source video file. EX1004, 1923-24, Figs. 1, 2. As Sambe explains, a “video segment consists of one or more consecutive Groups of Pictures (GOP).” *Id.*, 1924. Sambe also discloses a “*frame by frame* based transcoding architecture” that

decodes and subsequently encodes each frame. *Id.*, 1924 (emphasis added). “This frame by frame based transcoding architecture gives greater flexibility in transcoding.” *Id.* “For example, new encoding modules or new filter operations like digital water marking can be easily added.” *Id.* A POSITA would also have understood that the frames, which make up the GOPs in the received video segments, are a plurality of encoded images. *See, e.g.*, EX1013, 7 (explaining that digital video is composed of a series of frames produced by repeatedly sampling a visual scene at certain time intervals); EX1004, 1923 (describing the need to convert a “pre-encoded MPEG-2 digital video” to “other compressed formats such as MPEG-1, MPEG-4, H.263 and so on”); EX1014, 207-08 (explaining that “digital television video streams are generally encoded and transmitted by using MPEG-2 video at approximately 29.97 frames per second (FPS)”), 311 (explaining that that in a typical transcoding application converting a video stream from MPEG-2 format to MPEG-4 format, the input is encoded video content).

167. A POSITA would have also understood that the encoded images in the video segments received by Sambe’s Transcoding PCs are encoded according to a source format. This is because Sambe’s distributed transcoding system provides high speed transcoding *from* a source video format *to* different target video formats. EX1004, 1923-24. For example, Figure 2 of Sambe is a block diagram of a transcoding PC.

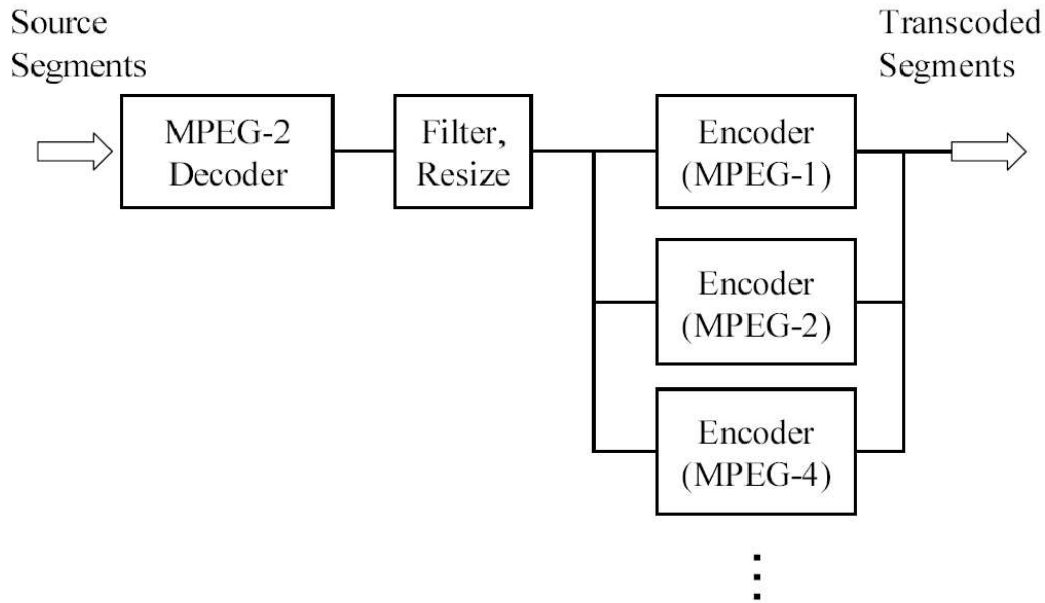


Fig. 2 Block diagram of a transcoding PC

Id., Fig. 2.

168. And Figure 2 depicts possible format conversions of MPEG-2 to MPEG-1, MPEG-2 to MPEG-2, and MPEG-2 to MPEG-4. *Id.* But note that a POSITA would have understood that the transcoding PC of Sambe was not limited to only the illustrated format conversions. *Id.* Sambe also describes converting “pre-encoded MPEG-2 digital video in archives to other compressed formats such as MPEG-1, MPEG-4, [and] H.263,” and altering video content “in terms of bit-rate and resolution to meet the network bandwidth and terminal capability.” *Id.*, 1923. Also, for example, Sambe’s authors disclose experiments they conducted including transcoding from a source video format of “MPEG-2 MP@ML” with a 720 x 480 spatial resolution and bitrate of 8 Megabits per second to different coding standards,

resolutions, and bitrates. *Id.*, 1925-26, Table 1. A POSITA would have understood that both the coding standard (such as MPEG-2) and the coding parameters (such as the resolution and bit-rate) constitute the source coding format of the video segments received by Sambe's Transcoding PCs. *See, e.g.*, EX1007, 84 (explaining that "[a] format is defined by such characteristics as the bit rate, frame rate, spatial resolution, coding syntax, and content"); *see also* EX1001, 1:26-32 (defining format both in terms of coding standards and coding parameters such as bitrate and resolution).

[1.c(ii)]: decoding the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;

169. Sambe, as described below, discloses limitation [1.c(ii)].

170. Sambe's parallel transcoders each decode a portion of the video file to generate a decoded portion of video including a plurality of decoded images. Sambe discloses that "[e]ach transcoding PC decodes and re-encodes the video segments into the different video formats specified. **Decoded frames** are filtered, resized, and passed to the encoder frame by frame." EX1004, 1924 (emphasis added). A POSITA would therefore understand that Sambe's parallel transcoders decode the source video file into decoded frames and that each decoded frame is a decoded image that the encoders then process. *See, e.g.*, EX1013, 83, Fig. 4.2 (showing that an H.264-compliant video decoder decodes a compressed H.264 bitstream to produce a series of "decoded frames"); EX1013, 7 (explaining that "[a] scene is

typically sampled at a point in time to produce a frame, which represents the complete visual scene at that point in time,” i.e., frames are sampled images of the scene). Thus, a POSITA would understand that the decoded frames of the source video segments disclosed by Sambe comprise a plurality of decoded images.

171. Sambe also discloses that the parallel Transcoding PCs decode the source video file according to a source format. As noted above for limitation [1.c(i)], the “video segments” of Sambe are each a portion of the source video file, and each have a source format. *See supra* § IX.A.1 (limitation [1.c(i)]; EX1004, 1923-26, Figs. 1, 2, Table 1. Because decoders must understand and implement the specific rules and algorithms of the corresponding video segment encoding method to accurately decode the video segment, a POSITA would have understood that decoding of the video segments would be based on their source format. *See, e.g.*, EX1014, 35 (explaining that a basic principle of video coding standards is that “[t]he standard should only define the syntax and semantics of the compressed bit stream,” which means that “a standard defines only the decoder process”); EX1013, 82 (stating that the H.264 standard “defines a format or syntax for compressed video and a method for decoding this syntax to produce a displayable video sequence”) (emphasis added); EX1016, 907 (“In H.261, both encoder and decoder operate in real time, so both have to be fast. *Still, the H.261 standard defines only the data stream and the decoder.* The encoder can use any method as long as it creates a valid

compressed stream”) (emphasis added). For example, Figure 2 of Sambe, shown below, is a block diagram of a transcoding PC, and Figure 2 includes only an “MPEG-2 Decoder.” EX1004, Fig. 2. This is because the distributed video transcoder of Sambe transcodes only from the “MPEG-2” coding format to other coding formats, including but not limited to MPEG-1, MPEG-2, and MPEG-4. *Id.*, 1923-24, Fig.2.

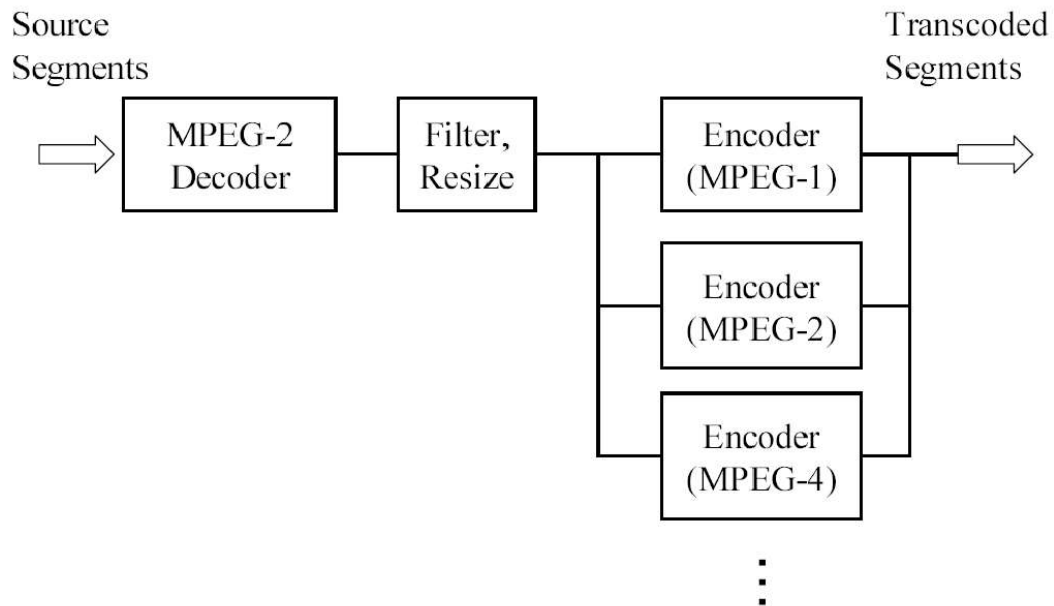


Fig. 2 Block diagram of a transcoding PC

[1.c(iii)]: “receiving the information based on the media metadata from the computer system; and”

172. The combination of Sambe in view of Vetro, as described below, renders limitation [1.c(iii)] obvious.

173. As noted above for claim limitation [1.a.], Sambe’s “distributed video transcoder” includes a “Source PC” that transmits video segments and their associated transcoding parameters to the plurality of “Transcoding PC[s]” to be transcoded in parallel. *Supra* § IX.A.1 (limitation [1.a]); EX1004, 1924. Further, as noted above for claim limitations [1.a.] and [1.b], in the combined system, Sambe’s “Source PC” is modified to include Vetro’s content classifier and CND manager. *See supra* § IX.A.1 (limitations [1.a] and [1.b]). Within the “Source PC” of the combined system, the content classifier generates parameters comprising scene complexity information, i.e., the claimed media metadata, and from the media metadata, the CND manager subsequently determines the transcoding strategy, i.e., the claimed information based on the media metadata. *Id.* Then, as demonstrated below in Figure 4, after the CND manager determines the transcoding strategy, the Source PC transmits it (purple arrow) to the Switchable Transcoder (red box) where it controls how the video segments (portions of the source video file) are transcoded—e.g., whether to transcode the video segment using the Discrete-Summary Transcoder or Continuous-Conversion Transcoder. EX1005, 21:8-14, Fig. 4 (annotated).

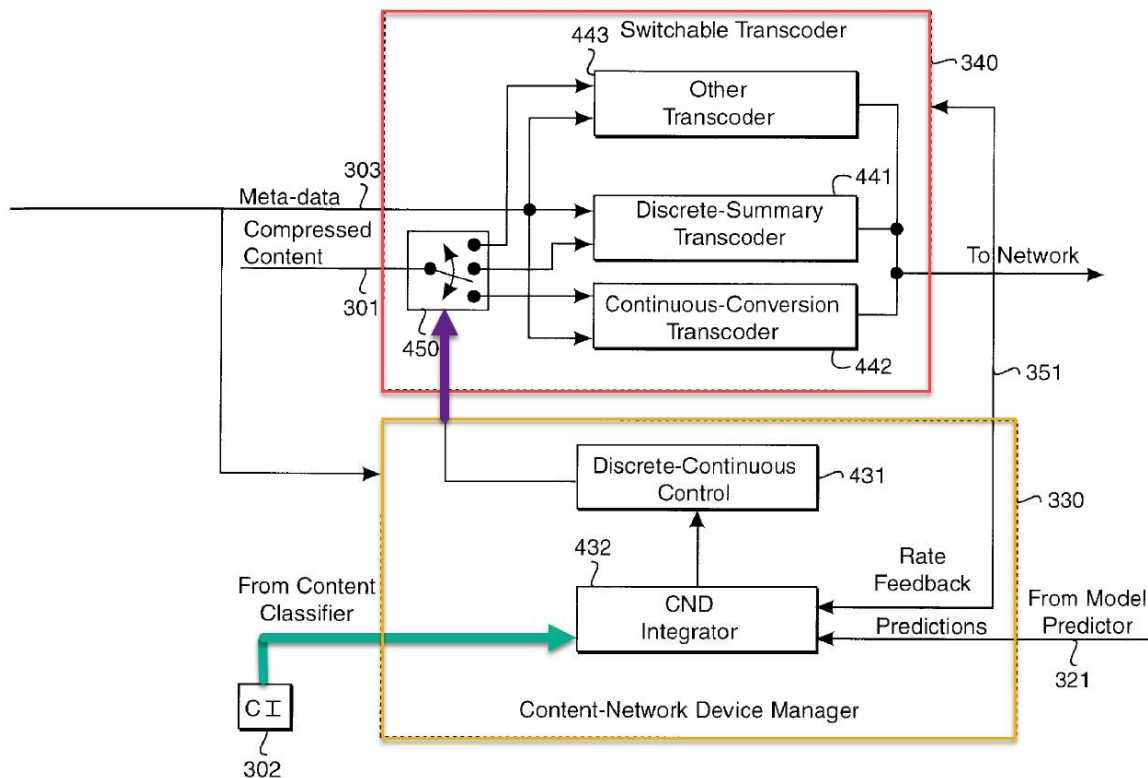


FIG. 4

EX1005, Fig. 4 (annotated).

174. In the combined system, as noted above the CND manager is in the source PC, and, thus, a POSITA would have understood that the source PC transmits the transcoding strategies (the claimed information based on the media metadata) to the plurality of transcoding PCs. Therefore, a POSITA would have understood that each of Sambe's parallel Transcoding PC[s] *receive* the transcoding strategy, i.e., information based on the media metadata, as discussed above for limitation [1.b], from the Source PC of the combined system. *Id.*, 1924, Figs. 1, 2. This would allow for the Transcoding PCs to implement the transcoding strategy determined by the CND manager of the combined system. As discussed above, a POSITA would make

this modification to provide for more efficient allocation of bits based on the complexity of each segment, ensuring consistent video quality. *Supra* § IX.A.1 (limitation [1.b]). A POSITA would have a reasonable expectation of success in making this modification for the same reasons discussed regarding limitation [1.b].
Id.

[1.c(iv)]: “encoding the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.”

175. The combination of Sambe in view of Vetro, as described below, renders limitation [1.c(iv)] obvious.

176. Sambe discloses that each Transcoding PC encodes the decoded images into of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format. As noted above for claim limitation [1.c(ii)], Sambe discloses decoding the source video file based on the source format into a plurality of decoded images, referred to as frames, to prepare for encoding. *Supra* § IX.A.1 (limitation [1.c(ii)]; EX1004, 1923-24, Figs. 1, 2. Sambe then discloses that after decoding their respective segments, the Transcoding PCs filter, resize, and pass the decoded frames to their encoders frame by frame. EX1004, 1924. The plurality of Transcoding PCs then “re-encode[] the video segments [frame-by-frame] ***into the different video formats specified.***” This

encoded portion of the video is then sent, along with the other portions of the source video file, to the Merging PC, where they are stitched together to form a video file in the desired format, i.e., the recited alternate video stream. *Id.* (emphasis added). Therefore, a POSITA would have understood that the combined transcoding system encodes “the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format.” Further, as noted above for the preamble of claim 1, Sambe discloses preparing a set of alternate video streams of the source video file. *Supra* § IX.A.1 (limitation [1.pre]); EX1004, 1923. Therefore, a POSITA would have understood that that the alternate video stream in the desired format is one of a set of multiple alternate video streams.

177. Additionally, Sambe’s Transcoding PCs encode the alternate video streams based on the information based on the media metadata as recited in this limitation. As noted above for limitation [1.c(iii)], in the combined system, each of Sambe’s Transcoding PCs receive from Sambe’s “Source PC” the transcoding strategy determined by Vetro’s CND manager, i.e., information based on the media metadata, based on media metadata describing scene complexity generated by Vetro’s content classifier. This transcoding strategy is used to inform the transcoding of the video segments received by Sambe’s Transcoding PCs. *See e.g.*, EX1004, 1924 (“When the source PC transmits segments to the transcoding PC, it

also sends the transcoding parameters that specify the operation of the filter and the encoder of [the] transcoding PC.”); EX1005, 8:21-26 (“In addition to receiving the meta-data 303, the manager 330 also receives input from both the content classifier 310 and the model predictor 320. The CND combines output data from these two sources of information together so that an optimal transcoding strategy is determined for the switchable transcoder 340.”); *id.*, 7:31-38, 7:40-43, 8:21-26, 9:26-39, Fig. 4 (340). For the reasons discussed regarding limitation [1.b], a POSITA would have been motivated to combine Vetro’s CND manager with Sambe’s Source PC to provide more efficient bit allocation for each segment based on the transcoding strategy provided by Vetro’s CND manager, resulting in consistent video quality. *Supra* § IX.A.1 (limitation [1.b]). And for the same reasons discussed regarding limitation [1.b], a POSITA would have had a reasonable expectation of success in making this combination. *Id.*

2. Claim 2: “The method of claim 1, wherein a group of the plurality of transcoding devices perform processes in parallel for the same alternate video stream from the set of multiple alternate video streams.”

178. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 2 depends. *See supra* § IX.A.1.

179. As noted above, parallel transcoding was well-known in the art prior to invention to speed up transcoding. *See supra* § VII.A.3; *see also, e.g.*, EX1011, 2905 (“Due to the complexity of video coding, fast transcoding is still a challenge.

Various parallel coding methods have been proposed.”). Regardless, Sambe also discloses the limitations of claim 2. For example, Sambe described a test in which its Source PC “implement[ed] a simple round-robin scheduling method,” as shown in Figure 17 below, in which the “Source PC” divides input video into segments (e.g., segments 1, 2, and so on). EX1004, 1929; *see also id.*, Fig. 17.

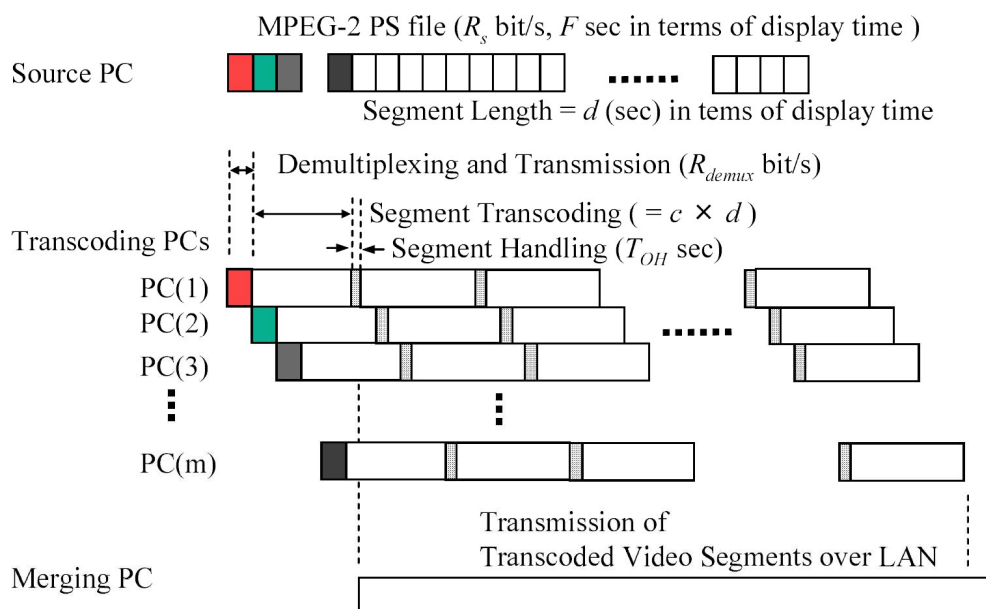


Fig. 17 Process flow of the distributed transcoding

180. As shown in Figure 17, Sambe discloses that segments 1 (red) and 2 (green) are sent to Transcoding PC1 and Transcoding PC2 respectively, and Transcoding PC1 and Transcoding PC2 encode segments 1 and 2 in parallel. EX1004, 1923-24, 1929; *see also id.*, Fig. 17. As the segments are encoded, they are sent to the merging PC and concatenated to form the desired video format files at the merging PC. *See id.*, 1923-24, Fig. 2. “After the second segment for each

desired file is received, the concatenation process begins and runs in parallel to the following segment transcoding.” *Id.*, 1924. A POSITA would have understood that the Merging PC stitches the segments together into the same alternative stream of the source video file (e.g., at a specific resolution and/or bit rate), referred to as the desired file, and this alternative stream of the source video file is one of the plurality of alternate video streams.

3. Claim 3: “The method of claim 1, wherein a group of the plurality of transcoding devices perform processes in parallel for different alternate video streams in the set of multiple alternate video streams.”

181. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 3 depends. *See supra* § IX.A.1.

182. As noted above, parallel transcoding was well-known in the art prior to invention to speed up transcoding. *See supra* § VII.A.3; *see also, e.g.*, EX1011, 2905 (“Due to the complexity of video coding, fast transcoding is still a challenge. Various parallel coding methods have been proposed.”). Regardless, Sambe also discloses the limitations of claim 3. For example, Sambe discloses that “[a]ll of the modules shown in [Figure 2], including transmission modules, are implemented by multi-thread programming.” EX1004, 1924, Fig. 2. This allows both transmission from the Source PC to the Transcoding PCs, and subsequent processing by the Transcoding PCs, to run in parallel. *Id.*, 1924. Also, as noted above for the preamble of claim 1, Sambe “describes a distributed video transcoding system that can

simultaneously transcode an MPEG-2 video file into various video coding formats with different rates.” *Id.*, 1923 (emphasis added). A POSITA would have understood that for Sambe to *simultaneously* transcode a video file into various coding formats with different rates, it must be capable of performing processes in parallel for different alternative video streams of the source video file in the set of multiple alternate video streams.

4. Claim 4: “The method of claim 1, wherein the media metadata further comprises scene change information indicating the start and end of a scene.”

183. As discussed above, Sambe in combination with Vetro teaches every limitation of claim 1, from which claim 4 depends. *See supra* § IX.A.1.

184. Vetro renders obvious media metadata that comprises scene change information indicating the start and end of a scene. Vetro teaches that a video, such as a news program, is partitioned into a sequence of “shots,” which are “smaller segments of video that begin when a camera is turned [on] and last until the camera is turned off.” EX1005, 17:32-46. As discussed above, a POSITA would have understood that these “shots” are scenes. *Supra* §§ VIII.A.2; VIII.B.1. Vetro discloses that shots are classified according to their coding complexity, and Vetro’s CND manager uses these classifications to inform how each shot will be transcoded. EX1005, 19:6-22, 20:27-29. A POSITA would have further understood that to use a “shot’s” complexity information during transcoding, Vetro’s system must maintain

metadata describing where the “shot” starts and ends. Otherwise, it would not have any way to identify the shot boundaries in order to use the complexity metadata to allocate an optimal bit rate to each shot, undermining Vetro’s purpose for generating and using shot complexity metadata. Without metadata describing the start and end of a scene, Vetro’s CND manager might allocate bitrate appropriate for a low complexity shot, but if that bitrate carries over to a high complexity shot without the CND manager’s knowledge, it may be insufficient to maintain a consistent video quality. *See, e.g.,* EX1006, [0005] (explaining that when bits are not properly allocated according to the complexity of segments, it leads to video quality that is “undesirable and inconsistent”). Vetro teaches the extraction and use of scene change information by disclosing that its content classifier “map[s] semantic features of content characteristics” including “video change information” into “a set of parameters,” i.e., media metadata, “that are used to make rate-quality trade-offs” in Vetro’s CND manager. EX1005, 7:63-8:1; *see also supra* § VIII.A.2. Thus, Vetro renders obvious that its disclosed transcoding system generates scene change information indicating the start and end of a scene, which provides Vetro’s CND manager with the necessary structure upon which to properly allocate bitrate based on scene complexity.

5. Claim 7

[7.pre] “The method of claim 4, by further performing following at each of the plurality of transcoding devices in parallel:”

185. As explained above, Sambe in combination with Vetro discloses every limitation of claim 4, from which claim 7 depends. *See supra* § IX.A.4. To the extent that the preamble of claim 7 is limiting, the combination of Sambe and Vetro disclose it. As discussed above, Sambe discloses a “distributed transcoding system” that includes multiple Transcoding PCs that perform transcoding in parallel. *See supra* § VIII.A.1.

[7.a] “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.”

186. As discussed above, Sambe in combination with Vetro teaches every limitation of claim 4, from which claim 7 depends. *See supra* § IX.A.4. Sambe in combination with Vetro also discloses limitation [7.a].

187. Vetro discloses this limitation. As discussed above, a POSITA would have understood from Vetro’s disclosure of generating “shot” level complexity information that Vetro’s disclosed system also generates media metadata comprising scene change information indicating the start and end of a scene. *Supra* § IX.A.4. Vetro further discloses that “shots” can be composed of a “group of frames.” EX1005, 17:47-49. And as noted above, a POSITA would have also understood that each frame in the shot constitutes an “image” or “picture.” *Supra* § IX.A.1

(limitation [1.c(ii)]). A POSITA would thus have understood that the group of frames making up Vetro's "shots" are the same as a "group of pictures" as recited in this limitation. *See, e.g.*, EX1016, 931 (referring to a "group of frames" as a "GOP"). Vetro further discloses determining a number of bits to encode each shot (i.e., each group of pictures (GOP)) based at least in part on the number of frames between the start and end of the shot as indicated by the information based on the media metadata. Generally, Vetro discloses that shot-level metadata data "are used for classification, bit allocation and rate-quality considerations for that particular shot." EX1005, 17:65-18:1. Specifically, Vetro explains that when users request multiple shots simultaneously, the CND manager must decide how much bit rate is allocated to each shot, and for the discrete-summary transcoder, "this [bit] rate can correspond to the *number of frames* that are sent." *Id.*, 21:35-36 (emphasis added). A POSITA would have, thus, understood that Vetro teaches determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.

6. Claim 8: "The method of claim 1, wherein the source format and the target format have different resolutions."

188. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 8 depends. *See supra* § IX.A.1.

189. As was noted above, resolution conversion was well-known in the art prior to the alleged invention. *See supra* § VII.A.2; *see also, e.g.*, EX1008, 793 (“As shown in Fig. 1, adjustment of coding parameters of compressed video, spatial and temporal resolution conversions, insertion of new information such as digital watermarks or company logos, and enhanced error resilience can also be done through transcoding.”) Regardless, Sambe also discloses the limitations of claim 8. For example, Sambe explains that due to modern terminals using displays with various sizes and resolutions, “it is often necessary for service providers delivering video content over the Internet to transcode the same content to yield different video formats, *spatial resolution*, and bit-rates simultaneously.” EX1004, 1923 (emphasis added). As shown below in Table 1, Sambe also provides example transcoding conditions in which the source video resolution (720 x 480) differs from the output video resolution (360 x 240). EX1004, 1926 (Table 1).

Table 1 Experimental transcoding conditions

Source Video Format	MPEG-2 MP@ML
Source Video	<i>football, flower garden mobile & calendar, sailboat</i>
Source Video Rate	8 Mbit/s
Source Frame Size	Horizontal 720 pixels, Vertical 480 lines
Output Video Coding	MPEG-2, MPEG-4 ASP
Output Video Rate	MPEG-2: 2 Mbps, 1 Mbps MPEG-4: 1 Mbps, 750 Kbps
Output Frame Size	Horizontal 360 pixels, Vertical 240 lines
GOP/GOV Structure	M=3, N=15

Id.

190. Thus, a POSITA would have understood that Sambe discloses a transcoding method where the source format and the target format have different resolutions.

7. Claim 9: “The method of claim 1, wherein the source format and the target format correspond to different video encoding standards.”

191. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 9 depends. *See supra* § IX.A.1.

192. Sambe also discloses the limitations of claim 9. For example, Sambe discusses the growing demand “to convert [] pre-encoded MPEG-2 digital video in archives to other compressed formats such as MPEG-1, MPEG-4, H.263 and so on.”

EX1004, 1923. Also for example, in the distributed transcoding system of Sambe,

“[e]ach transcoding PC decodes and re-encodes the video segments *into the different video formats specified.*” *Id.*, 1924 (emphasis added). As shown below, Figure 2 depicts a such a Transcoding PC consisting of encoder modules (MPEG-1 Encoder, MPEG-2 Encoder, and the MPEG-4 Encoder) that enable the Transcoding PC to encode an input video to several different video formats. EX1004, 1924.

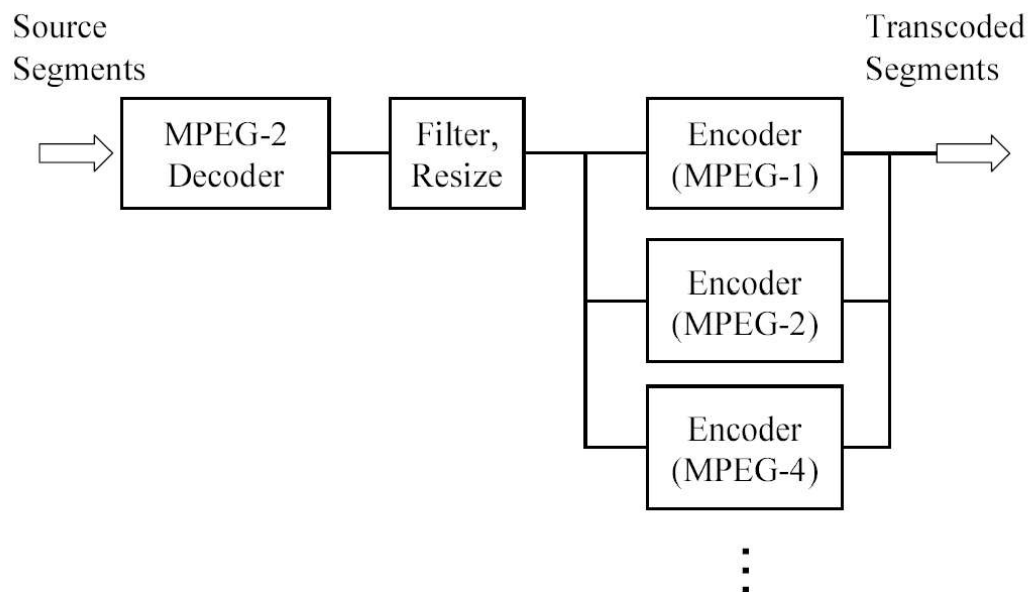


Fig. 2 Block diagram of a transcoding PC

Id.

193. Sambe also provides example transcoding conditions, in which the source video format (MPEG-2 MP@ML) differs from at least one of the output video formats (e.g., MPEG-4). EX1004, 1926 (Table 1) (annotated). A POSITA would have understood that the source and output video “format” and “coding” refer to different formats corresponding to different video standards.

Table 1 Experimental transcoding conditions

Source Video Format	MPEG-2 MP@ML
Source Video	<i>football, flower garden mobile & calendar, sailboat</i>
Source Video Rate	8 Mbit/s
Source Frame Size	Horizontal 720 pixels, Vertical 480 lines
Output Video Coding	MPEG-2, MPEG-4 ASP
Output Video Rate	MPEG-2: 2 Mbps, 1 Mbps MPEG-4: 1 Mbps, 750 Kbps
Output Frame Size	Horizontal 360 pixels, Vertical 240 lines
GOP/GOV Structure	M=3, N=15

Id.

8. Claim 10

[10.pre] “The method of claim 1, by further performing the following at each of the plurality of transcoding devices in parallel:”

194. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 10 depends. *See supra* § IX.A.1.

195. To the extent the preamble of claim 10 is considered limiting, in addition to teaching every limitation of claim 1, the combination of Sambe and Vetro further teaches the preamble of claim 10. As explained above, Sambe describes a system including transcoding devices, or PCs, that perform transcoding processes in parallel. *See supra* §§ VIII.A.1, IX.A.1.

[10.a] “dividing an image in the plurality of decoded images into a plurality of coding units based on the target format.”

196. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 1, from which claim 10 depends. *See supra* § IX.A.1.

197. Sambe also teaches limitation [10.a]. As noted above for limitation [1.c(ii)], the Transcoding PCs of Sambe decode the source video segments *frame by frame*, and each decoded frame is a decoded image. EX1004, 1924; *supra* § IX.A.1 (limitation [1.c(ii)]). All of the coding standards disclosed in Sambe, including MPEG-2, MPEG-2 MP@ML, MPEG-4, and MPEG-4 ASP, require dividing frames into macroblocks, as described below:

The H.264 Advanced Video Coding standard shares a number of common features with other popular compression formats such as MPEG-2 Video, MPEG-4 Visual, H.263, VC-1, etc. Each of these formats is based upon a CODEC ‘model’ that uses prediction and/or block-based motion compensation, transform, quantization and entropy coding.

EX1013, 26; *see also id.*, 31 (“[t]he macroblock, corresponding to a 16 x 16-pixel region of a frame, is the basic unit for motion compensated prediction in a number of important visual coding standards including MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263, and H.264”); 306 (“all of the current [video coding] formats have certain aspects in common, such as block-based operation, inter-frame prediction, and various forms of transform and entropy coding”); *supra* § VII.A.1.

198. Thus, a POSITA would have understood that each of the decoded frames disclosed in Sambe are subsequently divided into blocks and/or macroblocks for processing purposes. EX1004, 1925, 1928 (explaining that “the encoders we use employ simple block matching motion estimation”).

199. A POSITA would have understood that the blocks and/or macro blocks of each frame are a plurality of coding units. As noted in § VII.A.3 above, in video encoding, a coding unit is a block of pixels used as the basic building block for processing and compression, as described below:

The basic building block of an MPEG picture is the macroblock (Figure 9.17a). It consists of a 16×16 block of luminance (grayscale) samples (divided into four 8×8 blocks) and two 8×8 blocks of the matching chrominance samples. The MPEG compression of a macroblock consists mainly in passing each of the six blocks through a discrete cosine transform, which creates decorrelated values, then quantizing and encoding the results.

EX1016, 883; *see also* EX1019, 19 (“Whereas the basic coding unit for spatial redundancy reduction in MPEG is based on an 8 x 8 block, motion-compensation is usually based on a 16 pixel by 16 line macroblock.”); EX1014, 37-38 (explaining that of the “basic tools” of video coding is decomposing frames into blocks and performing transformation on each block); EX1013, 31 (“[t]he macroblock, corresponding to a 16 × 16-pixel region of a frame, is the basic unit for motion compensated prediction in a number of important visual coding standards including MPEG-1, MPEG-2, MPEG-4 Visual, H.261, H.263 and H.264”).

200. The blocks and/or macro blocks of Sambe serve as the basic building blocks for processing and compression—e.g., each macro block is individually quantized—and thus constitute “coding units.” EX1004, 1925. Further, a POSITA would have understood that each block and/or macro block coding unit is based on the target format, because the target formats determine the required coding units for the format (e.g., MPEG-2 requires macro blocks of size 16x16, while HEVC can have coding units of size 64x64). *See, e.g.*, EX1016, 883; EX1029, 1651; *see also*, *e.g.*, EX1019, 19; EX1014, 37-38; EX1013, 3.

9. Claim 11

[11.pre]: A system for transcoding video data, the system comprising:

201. To the extent the preamble is considered limiting, both Sambe and Vetro individually disclose it. I incorporate by reference my analysis of limitation [1.pre], as my analysis of “a method for transcoding a source video file into a set of multiple alternate video streams” also applies to the preamble of claim 11. The claimed “system for transcoding video data” is met both by Sambe’s disclosed “distributed video transcoder” consisting of a Source PC and a plurality of Transcoding PCs, and Vetro’s adaptive transcoding system for the same reasons discussed regarding limitation [1.pre]. *See supra* § IX.A.1.

[11.a]: a computer system configured as a media metadata generation device, wherein the computer system is configured to:

202. I incorporate by reference my analysis with respect to limitation [1.a], wherein I provide my analysis of “a computer system configured as a media metadata generation device.” The combination of Sambe in view of Vetro renders limitation [11.a] obvious for the same reasons as discussed regarding limitation [1.a]. *Supra* § IX.A.1.

[11.a(i)]: generate media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene complexity information; and

203. I incorporate by reference my analysis with respect to limitation [1.a], wherein I provide my analysis of a substantively identical limitation. The combination of Sambe in view of Vetro renders limitation [11.a] obvious for the same reasons as discussed regarding limitation [1.a]. *Supra* § IX.A.1.

[11.a(ii)] provide information based on the media metadata to a plurality of transcoding devices; and

204. I incorporate by reference my analysis with respect to limitation [1.b], wherein I provide my analysis of a substantively identical limitation. The combination of Sambe in view of Vetro renders limitation [11.a(ii)] obvious for the same reasons as discussed regarding limitation [1.b]. *Supra* § IX.A.1.

[11.b]: the plurality of transcoding devices, configured to perform the following at each of the plurality of transcoding devices in parallel:

205. I incorporate by reference my analysis with respect to limitation [1.c], wherein I provide my analysis of a substantively identical limitation. The combination of Sambe in view of Vetro renders limitation [11.b] obvious for the same reasons as discussed regarding limitation [1.c]. *Supra* § IX.A.1.

[11.b(i)]: receive the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;

206. I incorporate by reference my analysis with respect to limitation [1.c(i)], wherein I provide my analysis of a substantively identical limitation. Sambe discloses limitation [11.b(i)] for the same reasons as discussed regarding limitation [1.c(i)]. *Supra* § IX.A.1.

[11.b(ii)]: decode the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;

207. I incorporate by reference my analysis with respect to limitation [1.c(ii)], wherein I provide my analysis of a substantively identical limitation. Sambe discloses limitation [11.b(ii)] for the same reasons as discussed regarding limitation [1.c(ii)]. *Supra* § IX.A.1.

[11.b(iii)]: receive the information based on the media metadata from the computer system; and

208. I incorporate by reference my analysis with respect to limitation [1.c(iii)], wherein I provide my analysis of a substantively identical limitation. The

combination of Sambe in view of Vetro renders limitation [11.b(iii)] obvious for the same reasons as discussed regarding limitation [1.c(iii)]. *Supra* § IX.A.1.

[11.b(iv)]: encode the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on a target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.

209. I incorporate by reference my analysis with respect to limitation [1.c(iv)], wherein I provide my analysis of a substantively identical limitation. The combination of Sambe in view of Vetro renders limitation [11.b(iv)] obvious for the same reasons as discussed regarding limitation [1.c(iv)]. *Supra* § IX.A.1.

10. Claim 12: “The system of claim 11, wherein a group of the plurality of transcoding devices perform processes in parallel for the same alternate video stream from the set of multiple alternate video streams.”

210. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 12 depends. *See supra* § IX.A.9.

211. I incorporate by reference my analysis with respect to claim 2, wherein I provide my analysis of a substantively identical claim. Sambe discloses claim 12 for the same reasons as discussed regarding claim 2. *Supra* § IX.A.2.

11. Claim 13: “The system of claim 11, wherein a group of the plurality of transcoding devices perform processes in parallel for different alternate video streams in the set of multiple alternate video streams.”

212. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 13 depends. *Supra* § IX.A.9.

213. I incorporate by reference my analysis with respect to claim 3, wherein I provide my analysis of a substantively identical claim. Sambe discloses claim 13 for the same reasons as discussed regarding claim 3. *Supra* § IX.A.3.

12. Claim 14: The system of claim 11, wherein the media metadata further comprises scene change information indicating the start and end of a scene.

214. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 14 depends. *Supra* § IX.A.9.

215. I incorporate by reference my analysis with respect to claim 4, wherein I provide my analysis of a substantively identical claim. The combination of Sambe in view of Vetro renders claim 14 obvious for the same reasons as discussed regarding claim 4. *Supra* § IX.A.4.

13. Claim 17

[17.pre]: “The system of claim 14, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”

216. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 14, from which claim 17 depends. *Supra* § IX.A.9.

217. I incorporate by reference my analysis with respect to limitation [7.pre], wherein I provide my analysis of a substantively identical limitation. To the extent that the preamble of claim 17 is considered limiting, the combination of Sambe in view of Vetro disclose the preamble for the same reasons as discussed regarding limitation [7.pre]. *Supra* § IX.A.5.

[17.a]: “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.”

218. I incorporate by reference my analysis with respect to limitation [7.a], wherein I provide my analysis of a substantively identical limitation. The combination of Sambe in view of Vetro renders limitation [17.a] obvious for the same reasons as discussed regarding limitation [7.a]. *Supra* § IX.A.5.

14. Claim 18

[18.pre]: “The system of claim 11, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”

219. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 18 depends. *Supra* § IX.A.9.

220. I incorporate by reference my analysis with respect to limitation [10.pre], wherein I provide my analysis of a substantively identical limitation. To the extent that the preamble of claim 18 is considered limiting, in addition to teaching every limitation of claim 11, the combination of Sambe and Vetro further teaches the preamble of claim 18 for the same reasons as discussed regarding limitation [10.pre]. *Supra* § IX.A.8.

[18.a]: “dividing an image in the plurality of decoded images into a plurality of coding units based on the target format.”

221. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 18 depends. *Supra* § IX.A.9.

222. I incorporate by reference my analysis with respect to limitation [10.a], wherein I provide my analysis of a substantively identical limitation. Sambe discloses limitation [18.a] for the same reasons as discussed regarding limitation [10.a]. *Supra* § IX.A.8.

15. Claim 19: “The system of claim 11, wherein the source format and the target format have different resolutions.”

223. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 19 depends. *Supra* § IX.A.9.

224. I incorporate by reference my analysis with respect to claim 8, wherein I provide my analysis of a substantively identical claim. Sambe discloses claim 19 for the same reasons as discussed regarding claim 8. *Supra* § IX.A.6.

16. Claim 20: “The system of claim 11, wherein the source format and the target format correspond to different video encoding standards.”

225. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 18 depends. *Supra* § IX.A.9.

226. I incorporate by reference my analysis with respect to claim 9, wherein I provide my analysis of a substantively identical claim. Sambe discloses claim 20 for the same reasons as discussed regarding claim 9. *Supra* § IX.A.7.

17. Claim 21

[21.pre]: “A method for transcoding a source video file into a set of multiple alternate video streams, the method comprising:”

227. To the extent the preamble is considered limiting, both Sambe and Vetro individually render it obvious for the same reasons discussed regarding limitation [1.pre]. I incorporate by reference my analysis with respect to limitation [1.pre], wherein I provide my analysis of a substantively identical limitation. *Supra* § IX.A.1.

[21.a]: “generating, at a computer system configured as a media metadata generation device, media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises scene change information indicating the start and end of a scene, and scene complexity information;”

228. The combination of Sambe in view of Vetro renders obvious “generating, at a computer system configured as a media metadata generation device, media metadata related to the source video file prior to decoding, during a transcoding of, at least a portion of the source video file, where the media metadata comprises...scene complexity information” for the same reasons as discussed regarding limitation [1.a], and regarding that element, I incorporate by reference my analysis with respect to limitation [1.a], wherein I provide my analysis of a substantively identical element. *Supra* § IX.A.1. The combination of Sambe in view of Vetro render obvious “where the media metadata comprises scene change information indicating the start and end of a scene” for the same reasons as discussed

regarding and claim 4, and regarding that element, I incorporate by reference my analysis with respect to claim 4, wherein I provide my analysis of “wherein the media metadata further comprises scene change information indicating the start and end of a scene.” *Supra* § IX.A.4.

[21.b]: “providing information based on the media metadata from the computer system to a plurality of transcoding devices; and”

229. For the reasons discussed above for limitation 1[b] and claim 4, the combination of Sambe in view of Vetro renders limitation [21.b] obvious. *Supra* §§ IX.A.1, IX.A.4. I incorporate by reference my analysis with respect to limitation [1.b], wherein I provide my analysis of a substantively identical element to “providing information based on the media metadata from the computer system to a plurality of transcoding devices.” I also incorporate by reference my analysis with respect to claim 4, wherein I provide my analysis of “wherein the media metadata further comprises scene change information indicating the start and end of a scene.”

[21.c]: “performing the following at each of the plurality of transcoding devices in parallel:”

230. For the reasons discussed above for limitation 1[c] and claim 4, the combination of Sambe in view of Vetro renders limitation [21.c] obvious. *Supra* §§ IX.A.1, IX.A.4. I incorporate by reference my analysis with respect to limitation [1.c], wherein I provide my analysis of an element that is substantively identical element “performing the following at each of the plurality of transcoding devices in

parallel.” I also incorporate by reference my analysis with respect to claim 4, wherein I provide my analysis of the limitation “wherein the media metadata further comprises scene change information indicating the start and end of a scene.”

[21.c(i)]: “receiving the at least a portion of the source video file, including a first plurality of encoded images encoded according to a source format, from a media content source;”

231. The combination of Sambe in view of Vetro render limitation [21.c(i)] obvious for the same reasons as discussed regarding limitation [1.c(i)]. *Supra* § IX.A.1. I incorporate by reference my analysis with respect to limitation [1.c(i)], wherein I provide my analysis of a substantively identical limitation.

[21.c(ii)]: “decoding the at least a portion of the source video file based on the source format to generate a decoded portion of video including a plurality of decoded images;”

232. The combination of Sambe in view of Vetro renders limitation [21.c(ii)] obvious for the same reasons as discussed regarding limitation [1.c(ii)]. *Supra* § IX.A.1. I incorporate by reference my analysis with respect to limitation [1.c(ii)], wherein I provide my analysis of a substantively identical limitation.

[21.c(iii)]: “receiving the information based on the media metadata from the computer system;”

233. For the reasons discussed above for limitation [1.c(iii)] and claim 4, the combination of Sambe in view of Vetro renders limitation [21.c(iii)] obvious. *Supra* §§ IX.A.1, IX.A.4. I incorporate by reference my analysis with respect to limitation [1.c(i)], wherein I provide my analysis of an element substantively identical to

“receiving the information based on the media metadata from the computer system.”

Supra § IX.A.1. I also incorporate by reference my analysis with respect to claim 4, where I provide my analysis of the limitation “wherein the media metadata further comprises scene change information indicating the start and end of a scene.” *Supra* § IX.A.4.

[21.c(iv)]: “dividing an image in the plurality of decoded images into a plurality of coding units based on a target format, wherein the source format and the target format have different resolutions;”

234. The combination of Sambe in view of Vetro renders “dividing an image in the plurality of decoded images into a plurality of coding units based on a target format” obvious for the same reasons as discussed regarding limitation [10.a], and I incorporate by reference my analysis with respect to limitation [10.a], wherein I provide my analysis of a substantively identical limitation. *Supra* § IX.A.8. The combination of Sambe in view of Vetro renders “wherein the source format and the target format have different resolutions” obvious for the same reasons as discussed regarding claim 8, and I incorporate by reference my analysis with respect to claim 8, wherein I provide my analysis of a substantively identical limitation. *Supra* § IX.A.6. Thus, the combination of Sambe in view of Vetro renders limitation [21.c(iv)] obvious for the same reasons as discussed regarding limitations [10.a] and claim 8.

[21.c(v)]: “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata; and”

235. For the reasons discussed for limitation [7.a] and claim 4, the combination of Sambe in view of Vetro renders limitation [21.c(v)] obvious. I incorporate by reference my analysis with respect to limitation [7.a], wherein I provide my analysis of a limitation substantively identical to “determining a number of bits to encode a group of pictures (GOP) based at least in part on a number of frames between the start and end of a scene as indicated by the information based on the media metadata.” *Supra* § IX.A.5. I also incorporate by reference my analysis with respect to claim 4, where I provide my analysis of the limitation “wherein the media metadata further comprises scene change information indicating the start and end of a scene.” *Supra* § IX.A.4.

[21.c(vi)]: “encoding the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images based on the target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.”

236. For the reasons discussed for limitation [1.c(iv)] and claim 4, the combination of Sambe in view of Vetro renders limitation [21.c(vi)] obvious. *Supra* §§ IX.A.1, IX.A.4. I incorporate by reference my analysis with respect to limitation [1.c(iv)], wherein I provide my analysis of a limitation substantively identical to “encoding the plurality of decoded images of the decoded portion of video into an

alternate video stream including a second plurality of encoded images based on the target format and the information based on the media metadata, the alternate video stream being one of the set of multiple alternate video streams.” *Supra* § IX.A.1. I also incorporate by reference my analysis with respect to claim 4, where I provide my analysis of the limitation “wherein the media metadata further comprises scene change information indicating the start and end of a scene.” *Supra* § IX.A.4.

B. Ground 2: Claims 5-6 and 15-16 of the 806 Patent are Rendered Obvious by Sambe in view of Vetro and Gu

1. Claim 5

237. As discussed above, Sambe in view of Vetro teaches every limitation of claim 1. *See supra* § IX.A.1. And, as noted above, a POSITA would have looked to combine Gu with the combination of Sambe and Vetro. *See supra* § VIII.B.2. The combination of Sambe in view of Vetro and Gu, as described below, renders Claim 5 obvious.

[5.pre]: “The method of claim 1, wherein the encoding of the plurality of decoded images of the decoded portion of video into an alternate video stream includes performing at least the following operations for images in the plurality of decoded images:”

238. To the extent the preamble of claim 5 is limiting, the combination of Sambe in view of Vetro and Gu renders the preamble obvious. As noted above for claim limitation [1.c(iv)], Sambe in view of Vetro discloses encoding the plurality of decoded images of the decoded portion of video into an alternate video stream. *See supra* § IX.A.1.

[5.a]: “generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images using the scene change information and the scene complexity information within the information based on the media metadata according to the target format;”

239. The combination of Sambe in view of Vetro and Gu, as described below, renders claim limitation [5.a] obvious.

240. As discussed above regarding limitation [1.c(iv)] and claim 4, Sambe in combination with Vetro discloses encoding the plurality of decoded images “based on a target format and the information based on the media metadata,” where the media metadata comprises scene complexity information and scene change information. *Supra* §§ IX.A.2 (limitation [1.c(iv)]), IX.A.4. As noted above for claim 10, the blocks and/or macroblocks of Sambe serve as the Sambe Transcoding PCs’ basic building block of coding, and therefore, each block and/or macroblock is a “coding unit.” *See supra* § IX.A.8; *see also* EX1004, 1925.

241. At the time of the invention, generating a prediction image for each of a plurality of coding units, i.e., blocks or macroblocks, was a well-known and widely used technique for video encoding, as described in more detail below:

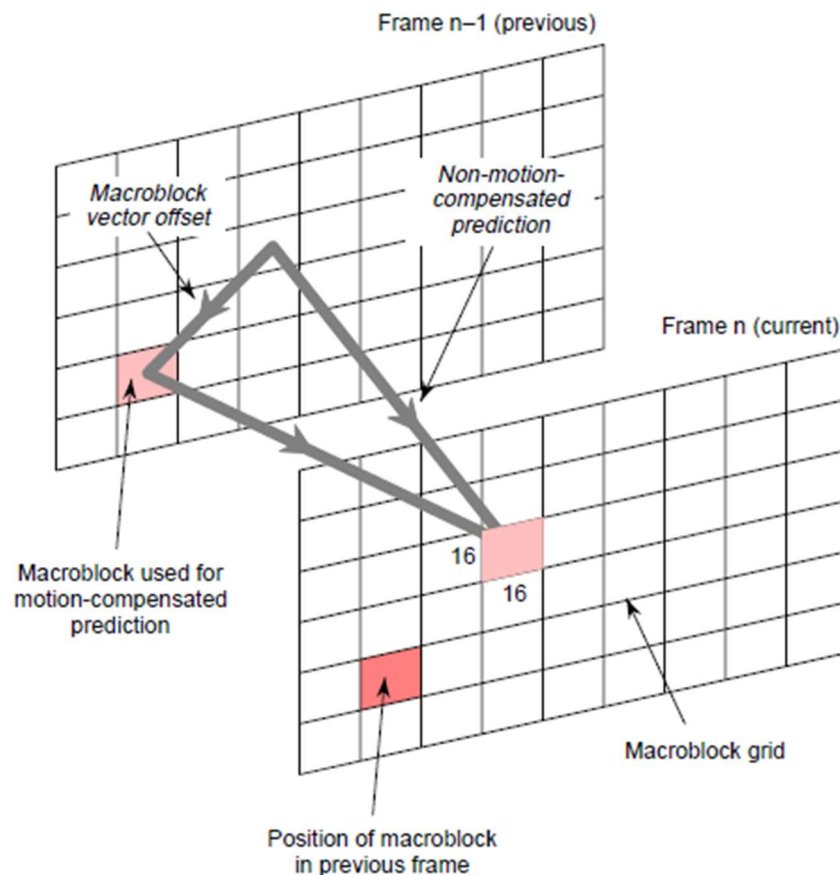
In MPEG coding, we combine interframe prediction (which reduces temporal redundancy) with the DCT and variable-length coding tools that were described in Section 3.3. (which reduce spatial redundancy), as shown in Fig. 7. The coder subtracts the prediction from the input to form a prediction-error picture. The prediction error is transformed with the DCT, the coefficients are quantized, and these quantized values are coded using a VLC.

The simplest interframe prediction is to predict a block of samples from the co-sited (i.e. the same spatial position) block in the reference frame.

In this case the “predictor” would comprise simply a delay of exactly one frame, as shown in Fig. 7. This makes a good prediction for stationary regions of the image but is poor in moving areas.

3.6. Motion-compensated interframe prediction

A more sophisticated prediction method, known as motion-compensated interframe prediction, offsets any translational motion which has occurred between the block being coded and the reference frame, and uses a shifted block from the reference frame as the prediction (see Fig. 8).



EX1019, 17, Fig. 8; *see also id.*, 14; *supra* § VII.A.1.

242. Thus, despite not explicitly saying so, a POSITA would understand that both Sambe and Vetro implicitly perform this limitation as part of their respective transcoding processes. For example, Sambe discloses converting video into MPEG-

2 and MPEG-4 ASP coding standards. EX1004, 1926 (Table 1). Each of those standards require generating prediction images as part of the encoding process, as described below:

The popular international standards for video compression such as MPEG-2, MPEG-4 and H.264/AVC make use of the well-known ‘hybrid’ video encoder model in which a prediction, formed from previously-transmitted frames, is subtracted from the current frame to form a residual which is then transformed using a block transform such as the DCT, quantized and coded for transmission.

EX1013, 250; *see also id.*, 26 (explaining that popular compression formats including MPEG-2 Video and MPEG-4 Visual are based upon a model “that uses prediction and/or block-based motion compensation, transform, quantization and entropy encoding”); EX1004, 1928 (“[i]n this paper, the encoders we use employ simple block matching motion estimation with fixed search range in the MC”). Thus, a POSITA would have understood that Sambe in combination with Vetro discloses generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images using the scene change information and the scene complexity information within the information based on the media metadata according to the target format.

243. Even if Sambe and/or Vetro do not explicitly disclose generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images, this limitation is expressly disclosed in Gu. Further, a POSITA would have looked to Gu for more explicit detail about the fundamental coding

functions recited in this limitation. *Supra* § VIII.B.2. Gu describes generating prediction images for each of a plurality of coding units, referred to as blocks and/or macroblocks, using the scene complexity information, including motion information associated with the blocks and/or macroblocks. Specifically, Gu explains that the video encoder can choose to either use pre-calculated motion information from input 120 (if available) or it can “choose ... to perform new motion estimation for the inter-path video content 110 with motion estimator 258.” EX1006, [0031]-[0032]. Gu explains that “[t]he motion estimator 258 estimates motion of macroblocks or other sets of samples of the video picture with respect to one or more reference pictures, which represent reconstructions of previously encoded video content frames.” *Id.*, [0032]. As a result, “[t]he motion estimator 258 outputs motion information 260 such as motion vector information.” *Id.* Gu further explains that “[t]he motion compensator 262 applies [the] motion vectors to certain reconstructed video content 266 (stored as reference picture(s)) when forming a motion-compensated current picture 268.” *Id.*, [0033]. Further, given each picture is composed of blocks and/or macroblocks, or coding units, *see, e.g., Id.*, [0031], [0035], it would have been obvious to a POSITA that Gu generates a prediction image for each block and/or macroblock of the “motion-compensated current picture.” As noted above, generating prediction pictures for individual blocks and/or macroblocks, or coding units, was the norm in encoding at the time of invention.

See supra § VII.A.1; *see also, e.g.,* EX1019, 17, Fig. 8 (describing and illustrating motion compensated interframe prediction on a macroblock-by-macroblock basis), 19 (“Whereas the basic coding unit for spatial redundancy reduction in MPEG is based on an 8 x 8 block, motion-compensation is usually based on a 16 pixel by 16 line macroblock.”).

[5.b]: “performing transforms on residual images of the plurality of coding units to generate sets of transform coefficients based on the target format; and”

244. The combination of Sambe in view of Vetro and Gu, as described below, renders claim limitation [5.b] obvious.

245. As discussed above with regard to limitation [1.c(iv)], Sambe in combination with Vetro discloses encoding the plurality of decoded images “based on a target format.” *Supra* § IX.A.2 (limitation [1.c(iv)]). As further discussed above, at the time of invention, “performing transforms on residual images of the plurality of coding units to generate sets of transform coefficients” was a widely understood and used step in the encoding process, as described below:

In MPEG coding, we combine interframe prediction (which reduces temporal redundancy) with the DCT and variable-length coding tools that were described in Section 3.3. (which reduce spatial redundancy), as shown in Fig. 7. *The coder subtracts the prediction from the input to form a prediction-error picture. The prediction error is transformed with the DCT, the coefficients are quantized, and these quantized values are coded using a VLC.*

EX1019, 17 (emphasis added); *see also id.*, 14; EX1014, 40 (“the goal of block transformation is to decorrelate the block data so that the resulting transform coefficients can be coded more efficiently”); *supra* § VII.A.1.

246. A POSITA would have understood a prediction-error picture to be the residual image as claimed. Indeed, despite not explicitly saying so, both Sambe and Vetro implicitly perform this step as part of their respective encoding processes. For example, Sambe discloses converting a video file according to MPEG-2 and MPEG-4 ASP standards. As part of their respective encoding processes, both of these standards require performing transforms on residual images of blocks, i.e, the recited plurality of coding units, which results in a set of transform coefficients, as described below:

The popular international standards for video compression such as MPEG-2, MPEG-4 and H.264/AVC make use of the well-known ‘hybrid’ video encoder model in which a prediction, formed from previously-transmitted frames, is subtracted from the current frame to form a residual which is then transformed using a block transform such as the DCT, quantized and coded for transmission.

EX1013, 250; *see also* EX1014, 18 (“[i]n the MPEG-1/2/4 standards, the residual signal after motion compensation is compressed with 8 x 8 floating-point precision DCT transform”), 37 (explaining that one of the “basic tools” for video coding standards is that “the video frame is first decomposed into [] blocks,” and then “[t]he blocks undergo transformation” after which “the original data are then converted to the transform domain for each block”).

247. But to the extent that Sambe or Vetro does not explicitly disclose performing transforms on residual images of the plurality of coding units, Gu does, and a POSITA would have looked to Gu for more explicit description of this fundamental encoding technique. *See supra* § VIII.B.2. Gu’s video encoder generates residual images of the plurality of coding units. As noted above for Claim limitation [5.a], the video encoder of Gu generates a prediction image, referred to as a “motion-compensated current picture.” EX1006, [0033]. Gu’s encoder then determines “[t]he difference (if any) between a block of the motion-compensated picture 268 and corresponding block of the original inter-path video picture” and explains that this difference “is the prediction residual 270 for the block.” *Id.*

248. The frequency transformer disclosed in Gu, a component of the video encoder, then processes “residual images of the plurality of coding units,” referred to as the “blocks of ... prediction residual data,” to generate sets of transform coefficients. *Id.*, [0034]. Specifically, Gu discloses as follows:

A frequency transformer 280 converts spatial domain video information into frequency domain (i.e., spectral, transform) data. For block-based video content, the frequency transformer 280 *applies a DCT, variant of DCT, or other forward block transform to blocks of the samples or prediction residual data*, producing blocks of frequency transform coefficients. The frequency transformer 280 may apply an 8×8, 8×4, 4×8, 4×4 or other size frequency transform.

Id. (emphasis added).

249. A POSITA would have understood that these blocks are the blocks and/or macroblocks of Gu, and as noted above for claim limitation [5.a], constitute coding units. *Id.*, [0032]. Therefore, a POSITA would have understood that the “blocks of ... prediction residual data” constitute “residual images of the plurality of coding units.” A POSITA would have further understood that by “applying a DCT,” Gu is referring to performing a discrete cosine transform (or a variant thereof) on the coding units, which was well-known to be “one of the best transforms” for bitrate reduction. EX1019, 14; EX1014, 4. For example, Gu discloses that its disclosed encoding process “can be performed according to a known video encoding standard, such as Windows Media Video format, SMPTE 421-M format, MPEG-x format (e.g., MPEG-1, MPEG-2, or MPEG-4), H.26x format (e.g., H.261, H.262, H.263, or H.264), or other format.” EX1006, [0026]. A POSITA would have known that in encoding to the H.264 standard, as one example, “a block of residual samples is transformed using a 4×4 or 8×8 integer transform, an approximate form of the Discrete Cosine Transform (DCT).” EX1013, 87.

[5.c]: “performing entropy encoding on the sets of transform coefficients to generate images for the second plurality of encoded images.”

250. The combination of Sambe in view of Vetro and Gu, as described below, renders claim limitation [5.c] obvious.

251. As discussed above regarding limitation [1.c(iv)], Sambe in combination with Vetro discloses “encoding the plurality of decoded images.”

Supra § IX.A.2 (limitation [1.c(iv)]). As noted above, at the time of invention, performing entropy encoding on sets of transform coefficients to generate images for a plurality of encoded images was a well-known, conventional step in the encoding process. *See supra* § VII.A.1; *see also* EX1019, 15-16; EX1014, 48; EX1013, 79 (explaining that “motion compensated prediction, transform coding, quantization and entropy coding[] form the basis of the reliable and effective coding model that has dominated the field of video compression for over 10 years”). This is because entropy encoding minimizes the amount of data that needs to be transmitted over a network, which, along with being a defining goal of video compression generally, is particularly important for applications like streaming video, where efficient bandwidth usage is crucial. *See supra* § VII.A.1; *see also* EX1019, 15-16; EX1014, 48; EX1013, 81. Indeed, despite not explicitly saying so, both Sambe and Vetro implicitly perform this step as part of their respective encoding processes. For example, Sambe discloses converting a source video to MPEG-2 and MPEG-4 ASP standards, each of which require the use of entropy encoding sets of transform coefficients, as described below:

The H.264 Advanced Video Coding standard shares a number of common features with other popular compression formats such as MPEG-2 Video, MPEG-4 Visual, H.263, VC-1, etc. Each of these formats is based upon a CODEC ‘model’ that uses prediction and/or block-based motion compensation, transform, quantization and entropy coding.

EX1013, 26; *see also id.*, 68 (“The major video coding standards released since the early 1990s have been based on the same generic design or model of a video CODEC that incorporates a motion estimation and compensation first stage, sometimes described as DPCM, a transform stage and an entropy encoder.”); *supra* § VII.A.1.

252. But even if Vetro and Sambe do not explicitly disclose performing entropy encoding on the sets of transform coefficients to generate images, Gu does, and a POSITA would have looked to Gu to provide more explicit description about this conventional encoding technique. *See supra* §§ VIII.B.2, VII.A.1. As noted above for claim limitation [5.b], the frequency transformer of Gu, a component of its disclosed video encoder, processes “residual images of the plurality of coding units,” referred to as the “blocks of ... prediction residual data” to generate sets of transform coefficients. EX1006, [0034]. The Gu video encoder’s “quantizer 282 then quantizes the blocks of transform coefficients.” *Id.*, [0035], Fig. 2. Specifically, “[t]he quantizer 282 applies non-uniform, scalar quantization to the spectral data with a step size that varies spatially on a picture-by-picture basis, macroblock-by macroblock basis or other basis.” *Id.* As noted above, quantization of transform coefficients was well-known in the art and minimizes the number of bits which must be transmitted, i.e., compresses the transform coefficients. *See supra* § VII.A.1; *see also, e.g.*, EX1019, 14. A POSITA would have understood that quantization compresses values into a discrete set of values that compresses them

without changing their overall nature and relationship. *See, e.g.*, EX1014, 40 (explaining that quantization is simply dividing the block of transform coefficients by a weighted quantization matrix to reduce the number of bits that must be sent). Next, the entropy coder 284 of Gu, another component of the Gu’s video encoder, performs entropy encoding on “the output of the quantizer 282,” which, as noted above, constitutes the sets of transform coefficients. EX1006, [0037]; *see also id.*, [0038].

253. As noted above for claim limitation [1.c(ii)], the combined system of Sambe and Vetro “encod[es] the plurality of decoded images of the decoded portion of video into an alternate video stream including a second plurality of encoded images.” *Supra* § IX.A.1 (limitation 1[c(ii)]; EX1004, 1923; EX1005, 23:8-30. Like the combined system of Sambe and Vetro, the video encoder of Gu “receives a sequence of video pictures (frames) as its raw video content input 210 and produces a compressed bit stream 230 as output.” EX1006, [0027]. A POSITA would have understood the output bit stream disclosed in Gu consists of a plurality of encoded images. *See, e.g.*, EX1016, 869 (explaining that a compressed video consists of “a sequence of compressed frames of the three types I, P, and B”).

2. Claim 6

254. As discussed above, Sambe in view of Vetro teaches every limitation of claim 1. *See supra* § IX.A.1. And, as noted above, a POSITA would have looked

to combine Gu with the combination of Sambe and Vetro. *See supra* § VIII.B.2.

The combination of Sambe in view of Vetro and Gu renders Claim 6 obvious.

[6.pre]: “The method of claim 1, by further performing the following at each of the plurality of transcoding devices in parallel:”

255. To the extent the preamble of claim 6 is limiting, the combination of Sambe in view of Vetro and Gu discloses it. As explained above, Sambe describes a system including transcoding devices, or PCs, that perform transcoding processes in parallel. *See supra* § IX.A.1.

[6.a]: “performing quantization on the sets of transform coefficients for an image in the plurality of decoded images based at least in part on the scene complexity information within the information based on the media metadata; and”

256. The combination of Sambe in view of Vetro and Gu, as described below, renders claim limitation [6.a] obvious. As noted above for claim limitation [5.b], the combined system generates sets of transform coefficients for an image in the plurality of decoded images. *Supra* § IX.B.1.

257. As discussed above with respect to claim 1, Sambe in combination with Vetro disclose encoding the plurality of decoded images “based on... the information based on the media metadata,” where the media metadata comprises scene complexity information. *Supra* § IX.A.1 (limitations 1[a], 1[c(iv)]). As further discussed above, at the time of invention, “performing quantization on the sets of

transform coefficients for an image in the plurality of decoded images” was a well-known and widely used step in the encoding process, as described below:

In MPEG coding, we combine interframe prediction (which reduces temporal redundancy) with the DCT and variable-length coding tools that were described in Section 3.3. (which reduce spatial redundancy), as shown in Fig. 7. The coder subtracts the prediction from the input to form a prediction-error picture. *The prediction error is transformed with the DCT, the coefficients are quantized, and these quantized values are coded using a VLC.*

EX1019, 17 (emphasis added); *see also* EX1014, 40; EX1016, 883; *supra* § VII.A.1.

258. Indeed, despite not explicitly saying so, both Sambe and Vetro implicitly perform this step as part of their respective encoding processes. For example, Sambe discloses converting a source video to MPEG-2 and MPEG-4 ASP standards, both of which require performing quantization on sets of transform coefficients of an image, as described below:

The popular international standards for video compression such as MPEG-2, MPEG-4 and H.264/AVC make use of the well-known ‘hybrid’ video encoder model in which a prediction, formed from previously-transmitted frames, is subtracted from the current frame to form a residual which is then transformed using a block transform such as the DCT, quantized and coded for transmission.

EX1013, 250.

259. Therefore, a POSITA would have understood that Sambe in combination with Vetro discloses “performing quantization on the sets of transform coefficients for an image in the plurality of decoded images based at least in part on

the scene complexity information within the information based on the media metadata.”

260. But even if Sambe and Vetro do not explicitly disclose performing quantization on the sets of transform coefficients for an image in the plurality of decoded images, Gu does, and a POSITA would have been motivated to look to Gu to provide more explicit description of this conventional encoding function. *Supra* § VIII.B.2. As discussed above regarding limitation [5.c], the quantizer component of Gu’s encoder “quantizes the blocks of transform coefficients.” *See supra* § IX.B.1; EX1006, [0035], Fig. 2.

[6.b]: “quantizing the generated set of transform coefficients according to the target format.”

261. The combination of Sambe in view of Vetro and Gu, as described below, discloses this limitation. As discussed above regarding limitation [6.a], Sambe in view of Vetro and Gu teaches quantizing the generated set of transform coefficients. *Supra* § IX.B.2.

262. Sambe discloses this limitation. Sambe’s reencoding step of its disclosed transcoding process involves first “deciding the target frame size,” and then determining the “quantization scale Q of each macro-block ... *so that actual coded size will be equal to the target size using virtual buffer memory d_n .*” EX1004, 1926. Because the quantization scale Q of each macroblock is chosen based on the target size of the encoding, a POSITA would have understood that Sambe discloses

quantizing the generated set of transform coefficients according to the target format. The target frame size refers to the output resolution of the output video. *See supra* §§ IX.A.6, IX.A.8; EX1004, 1926 (Table 1). A POSITA would understand that the target resolution (as well as the target coding standard) constitutes the target format of the output video. *See, e.g.*, EX1007, 84 (explaining that “[a] format is defined by such characteristics as the bit rate, frame rate, spatial resolution, coding syntax, and content”); *see also* EX1001, 1:26-32 (defining format both in terms of coding standards and coding parameters such as bitrate and resolution).

263. Gu also renders this limitation obvious. As discussed above regarding limitation [6.a], the quantizer component of Gu’s encoder “quantizes the blocks of transform coefficients.” *See supra* § IX.B.2; EX1006, [0035], Fig. 2. Gu also discloses that “encoding can be performed according to a known video encoding standard, such as Windows Media Video format, SMPTE 421-M format, MPEG-x format (e.g., MPEG-1, MPEG-2, or MPEG-4), H.26x format (e.g., H.261, H.262, H.263, or H.264), or other format.” EX1006, [0026].

264. A POSITA would have understood the video encoder of Gu quantizes the transform coefficients according to the target format. It was known that quantization tables used for quantization are typically specified by the encoding format. *See, e.g.*, EX1016, 884 (“MPEG specifies default quantization tables, but custom tables can also be used.”). Also, Gu explains that “[t]he initial sample

quantization step size” for the encoding process “can be chosen depending on the particular codec standard used by the video encoder.” EX1006, [0053]. “For example, in the case of the SMPTE 421-M video codec, the initial sample quantization step size may be chosen to be 4.” *Id.*

265. As noted above, at the time of invention, “quantizing the generated set of transform coefficients” was a widely understood and used step in the encoding process. EX1019, 17; EX1014, 40; EX1016, 883; *see also supra* § VII.A.1. Indeed, despite not explicitly saying so, Vetro implicitly performs this step as part of its encoding process. EX1005, 2:38-60 (describing quantization of transform coefficients in the Background section). Thus, a POSITA would have been motivated to look to Gu to provide more explicit description of the functions recited in this limitation. *See supra* § VIII.B.2.

3. Claim 15

[15.pre]: “The system of claim 11, wherein encoding the plurality of decoded images of the decoded portion of video into an alternate video stream is performed using at least the following operations for images in the plurality of decoded images:”

266. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 15 depends. *Supra* § IX.A.9.

267. To the extent that the preamble of claim 15 is considered limiting, the combination of Sambe in view of Vetro and Gu discloses the preamble for the same reasons as discussed regarding limitation [5.pre]. *Supra* § IX.B.1. I incorporate by

reference my analysis with respect to limitation [5.pre], wherein I provide my analysis of a substantively identical limitation.

[15.a]: “generating a prediction image for each of a plurality of coding units of an image in the plurality of decoded images using the scene change information and the scene complexity information within the information based on the media metadata according to the target format;”

268. The combination of Sambe in view of Vetro and Gu renders claim limitation [15.a] obvious for the same reasons as discussed regarding limitation [5.a]. *Supra* § IX.B.1. I incorporate by reference my analysis with respect to limitation [5.a], wherein I provide my analysis of a substantively identical limitation.

[15.b]: “performing transforms on residual images of the plurality of coding units to generate sets of transform coefficients based on the target format; and”

269. The combination of Sambe in view of Vetro and Gu renders claim limitation [15.b] obvious for the same reasons as discussed regarding limitation [5.b]. *Supra* § IX.B.1. I incorporate by reference my analysis with respect to limitation [5.b], wherein I provide my analysis of a substantively identical limitation.

[15.c]: “performing entropy encoding on the sets of transform coefficients to generate images for the second plurality of encoded images.”

270. The combination of Sambe in view of Vetro and Gu renders claim limitation [15.c] obvious for the same reasons as discussed regarding limitation [5.c]. *Supra* § IX.B.1. I incorporate by reference my analysis with respect to limitation [5.c], wherein I provide my analysis of a substantively identical limitation.

4. Claim 16

[16.pre]: “The system of claim 11, wherein the plurality of transcoding devices are configured to further perform the following at each of the plurality of transcoding devices in parallel:”

271. As discussed above, Sambe in combination with Vetro discloses every limitation of claim 11, from which claim 16 depends. *Supra* § IX.A.9.

272. To the extent that the preamble of claim 16 is considered limiting, the combination of Sambe in view of Vetro and Gu discloses the preamble for the same reasons as discussed regarding limitation [6.pre]. *Supra* § IX.B.2. I incorporate by reference my analysis with respect to limitation [6.pre], wherein I provide my analysis of a substantively identical limitation.

[16.a]: “performing quantization on the sets of transform coefficients for an image in the plurality of decoded images based at least in part on the scene complexity information within the information based on the media metadata; and”

273. The combination of Sambe in view of Vetro and Gu renders limitation [16.a] obvious for the same reasons as discussed regarding limitation [6.a]. *Supra* § IX.B.2. I incorporate by reference my analysis with respect to limitation [6.a], wherein I provide my analysis of a substantively identical limitation.

[16.b]: “quantizing the generated set of transform coefficients according to the target format.”

274. The combination of Sambe in view of Vetro and Gu renders limitation [16.b] obvious for the same reasons as discussed regarding limitation [6.b]. *Supra*

§ IX.B.2. I incorporate by reference my analysis with respect to limitation [6.b], wherein I provide my analysis of a substantively identical limitation.

X. CONCLUSION

275. For the reasons I have presented above, it is my opinion that the references serving as the bases for the grounds of rejection in the Petition render claims 1-21 of the challenged patent unpatentable.

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

Dated: May 27, 2025

Hari Kalva, Ph.D.

Hari Kalva, Ph.D.