UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

Petitioner,

v.

NOKIA TECHNOLOGIES OY,

Patent Owner.

Case IPR2024-01175
U.S. Patent No. 6,856,701

EXHIBIT 2001

DECLARATION OF DOUGLAS A. CHRISSAN, Ph.D.

TABLE OF CONTENTS

I. INTRODUCTION 1
II. BACKGROUND AND QUALIFICATIONS 2
III. BASES OF OPINIONS 4
IV. APPLICABLE LEGAL STANDARDS 5
A. Ordinary Skill in the Art 5
B. Claim Construction 6
C. Obviousness 7
V. OVERVIEW OF VIDEO COMPRESSION AND DECOMPRESSION TECHNOLOGY 10
A. Representation of Digital Video 12
B. Intra-prediction and Inter-prediction 14
C. The Discrete Cosine Transform 15
D. Quantization of DCT Coefficients 21
E. Run-level Coding 23
F. Entropy Coding 25
G. Variable Length Coding (VLC) 27
H. Arithmetic Coding 30
VI. OVERVIEW OF THE ’701 PATENT’S INVENTION 39
I. Challenged claims 1–40 42
VII. CLAIM CONSTRUCTION OF ’701 PATENT CLAIM TERMS 50
VIII. THE ASSERTED PRIOR ART 51
A. Tsai (Ex. 1005) 51
B. VCEG-L28 (Ex. 1006) 52
C. VCEG-L13 (Ex. 1007) 53
D. Savatier (Ex. 1008) 54

Nokia Exhibit 2001, p.2
Amazon.com v. Nokia
IPR2024-01175

E. Tsai in Combination with VCEG-L28 (Ground 1A) Does Not Teach or Disclose the Challenged Claims 55

F. A POSITA would not be motivated to combine and/or modify Tsai and VCEG-L28.....58

G. Applicant Admitted Prior Art (AAPA) in Combination with Savatier (Ground 2) Does Not Teach or Disclose Challenged Claims 1-9, 18-26, 35, 37, 39.....60

H. AAPA in Combination with VCEG-L13 (Ground 3) Does Not Teach or Disclose Challenged Claims 10-17, 27-34, 36, 38, 40 62

IX. OPINIONS.....64

Nokia Exhibit 2001, p.3
Amazon.com v. Nokia
IPR2024-01175

I, Douglas A. Chrissan, declare as follows:

I. INTRODUCTION

1. I have been retained as a technical expert on behalf of Patent Owner Nokia Technologies Oy ("Nokia") for the above-captioned inter-partes review ("IPR"). I have been asked to provide my opinions on the technology disclosed in U.S. Patent No. 6,856,701 ("the '701 Patent") (Ex. 1001) and on the unpatentability arguments presented in Amazon's ("Petitioner") Petition ("Pet.") alleging that claims 1–40 of the '701 Patent (the "Challenged Claims") are invalid under 35 U.S.C. § 103.

2. I make this declaration based on my personal knowledge, consideration of the materials I discuss herein, and my expert opinion.

3. I am being paid for my work on this matter on an hourly basis at $300/hour. My compensation is not contingent on reaching any particular findings or conclusions, or on any particular outcome in this matter. I have no financial interest in the outcome of this matter.

4. In preparing this Declaration, I have reviewed the '701 Patent, its prosecution history, and considered each of the documents cited herein. I have also read and considered the IPR Petition concerning the '701 Patent and the exhibits and evidence cited therein as grounds for invalidity, including but not limited to Exhibits 1001–23. Additional documents reviewed are discussed in Section III below.

Nokia Exhibit 2001, p.4
Amazon.com v. Nokia
IPR2024-01175

5. In formulating my opinions, I have also relied upon my experience, education, knowledge and expertise in the area of video processing, including video compression and decompression—also referred to as video encoding and decoding. In reviewing the material relevant to these proceedings, I have considered the viewpoint of a person of ordinary skill in the art in the September 2001 timeframe.

II. BACKGROUND AND QUALIFICATIONS

6. The following sections recite the most relevant aspects of my educational, professional, and technical background. A copy of my curriculum vitae, which describes my qualifications in more detail, is attached to this report as Exhibit 2002.

7. I am presently a technical consultant in the areas of communications systems, multimedia systems, computer systems, and digital signal processing. My past work experience includes at least four years of designing, programming and directly working with semiconductors and systems for video encoding and decoding.

8. I earned a B.S. and M.S. in Electrical Engineering from the University of Southern California in 1988 and 1990, respectively, and a Ph.D. in Electrical Engineering from Stanford University in 1998.

9. From 1993 to 1997, I was a full-time doctoral student at Stanford University. My Ph.D. dissertation and related publications were in the fields of statistical signal processing and communication systems, and more specifically in

Nokia Exhibit 2001, p.5
Amazon.com v. Nokia
IPR2024-01175

the area of impulsive noise modeling for low-frequency communication systems. In 1995, I was the instructor for the graduate Statistical Signal Processing class (EE278) in the Electrical Engineering department at Stanford University. Prior to teaching this class, I was a teaching assistant for ten different classes in signal processing and radio frequency electronics at Stanford.

10. I was a Masters Fellow and Member of the Technical Staff at Hughes Aircraft Company in El Segundo, California, from 1988–1993. While at Hughes Aircraft, I designed and developed wireless communication systems for commercial and military spacecraft, including for the MILSTAR satellite program.

11. Between 1992 and 1993, while at Hughes Aircraft Company, I designed and built a state-of-the-art, 800 megabit-per-second (Mbps) telecommunications modem for the NASA Lewis Research Center.

12. From 1997–2003, I worked at 8x8, Inc., starting as a DSP software engineer in 1997, becoming a manager in 1998, a director in 1999, and Vice President of Engineering in 2000 (managing a team of approx. 60 engineers in the company's microelectronics group). I played a key role in developing several semiconductor products used worldwide in multimedia and communications devices, mainly for video conferencing systems and Internet Protocol ("IP") telephones. Some of these semiconductor products were in production more than ten years.

Nokia Exhibit 2001, p.6
Amazon.com v. Nokia
IPR2024-01175

13. From 2003–2007, I was a Systems Architect and Engineering Program Manager at Texas Instruments in the Digital Subscriber Line ("DSL") product business unit. At Texas Instruments, I was directly involved in the architecture, design, development and production of multicarrier DSL modem products. My work specifically included architecting a multicarrier DSL semiconductor and software product and managing all aspects of its development from inception to production. I also gained further experience with IEEE 802.11 (WLAN/WiFi) and WiMAX systems.

14. From 2009–2011, I was an Engineering Director at Maxim Integrated Products. At Maxim, I was directly involved in the architecture, design, development and production of semiconductors and systems for video encoding and decoding.

15. I have developed, and managed the development of, several successful semiconductor, software and systems products in the communications and multimedia fields. These products are listed in the attached curriculum vitae.

III. BASES OF OPINIONS

16. In the course of performing my analysis and forming my opinions, I have reviewed the Petition for this IPR2024-01141 and its Exhibits 1001–23. Exhibits 1001–23 include:

- U.S. Patent No. 6,856,701 ("the '701 Patent") (Ex. 1001) and its file history (Ex. 1002)

Nokia Exhibit 2001, p.7
Amazon.com v. Nokia
IPR2024-01175

- Declaration of Dr. Eli Saber (Ex. 1003)
- U.S. Patent No. 5,818,877 ("Tsai") (Ex. 1005)
- ITU-T[1], "VLC concatenation for Intra coding" (2001), ("VCEG-L28") (Ex. 1006)
- ITU-T, "Adaptive Codes for H.26L" (2001), ("VCEG-L13") (Ex. 1007)
- U.S. Patent No. 5,400,075 ("Savatier") (Ex. 1008)

IV. APPLICABLE LEGAL STANDARDS

A. Ordinary Skill in the Art

17. My opinions in this declaration are based on the understandings of a person of ordinary skill in the art—sometimes referred to as an "ordinary artisan" or by the acronyms "POSA" or "POSITA"—as of the time of the invention, which I understand to be Sept. 14, 2001, the filing date of provisional application No. 60/322,112 leading to the '701 Patent. I understand that the person of ordinary skill in the art is a hypothetical person who is presumed to have known the art relevant to the Challenged Claims of the '701 Patent at the time of the invention.

18. I understand that, in assessing the level of skill of a person of ordinary

[1] ITU-T stands for "International Telecommunication Union-Telecommunication Sector."

Nokia Exhibit 2001, p.8
Amazon.com v. Nokia
IPR2024-01175

skill in the art, one should consider the type of problems encountered in the art, the prior solutions to those problems found in the prior art references, the rapidity with which innovations are made, the sophistication of the technology, the level of education of active workers in the field, and one's own experience working with those of skill in the art at the time of the invention.

19. Petitioner submits that a POSITA at the time of the '701 Patent would have had (1) a Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video and/or image processing or coding. More education could substitute for experience, and vice versa. I agree that this definition of a POSITA is sufficient for this IPR proceeding.

20. I was at the time of invention, and am, one of more than ordinary skill in the art through my education, research experience and work experience. As of the date of the invention, I was familiar with the types of problems encountered in video processing and coding and the rapidity at which innovations were made. I was and am familiar with people having this level of skill in the area of video processing and coding.

B. Claim Construction

21. I understand that the Challenged Claims of the '701 Patent in this IPR are generally interpreted according to their ordinary and customary meaning taking

Nokia Exhibit 2001, p.9
Amazon.com v. Nokia
IPR2024-01175

into consideration the so-called "intrinsic evidence" of the patent consisting of (1) the claim language; (2) the specification; and (3) the prosecution history.

22. I understand that claim terms may be explicitly defined in the patent specification or they may be implicitly defined through consistent usage in the specification. I also understand that the scope of claim terms may be limited by statements in the specification or prosecution history where the applicant clearly disavows or disclaims subject matter.

23. I understand that claim construction is a matter of law. As of this time, my understanding is that no claim terms have been proposed for construction.

C. Obviousness

24. Counsel has informed me, and I understand, that an issued patent claim is invalid as obvious if it can be shown that the differences between the patented subject matter and the prior art are such that the subject matter as a whole would have been obvious, at the time the invention was made, to a person having ordinary skill in the art. Relevant considerations include the level of ordinary skill in the art; the scope and content of the prior art; differences between the prior art and the claims at issue; and the objective indicia, or secondary considerations, of nonobviousness.

25. In this regard, I am informed and I understand that the secondary considerations of nonobviousness could include factors such as commercial success, long felt but unresolved needs, the failure of others, skepticism by experts, praise by

Nokia Exhibit 2001, p.10
Amazon.com v. Nokia
IPR2024-01175

others, award and accolades, and copying.

26. Counsel has informed me, and I understand, that in order to evaluate the obviousness of any claim of the ’701 Patent over a given prior art combination, I should analyze whether the prior art references, included collectively in the combination, disclose each and every element of the allegedly invalid claim as those references are read by the person of ordinary skill in the art at the time of the invention.

27. Counsel has informed me, and I understand, that even where all of the claim limitations are expressly disclosed in the prior art references, there must be some showing that a person of ordinary skill in the art would have been motivated to combine such prior art references, and explanation of how the combination would be achieved, and that there would have been a reasonable expectation of successfully achieving the claimed invention from such combination.

28. Counsel has informed me, and I understand, that absent some additional reasoning, mere routine optimization for a person of ordinary skill in the art to arrive at the claimed invention through combination of prior art references is insufficient to support a conclusion of obviousness.

29. Counsel has informed me, and I understand, that to rely on a reference it must be analogous prior art. Counsel has informed me and I understand that a reference is considered analogous prior art: (1) if the reference is from the same field

Nokia Exhibit 2001, p.11
Amazon.com v. Nokia
IPR2024-01175

of endeavor as the claimed subject matter, or (2) if the reference "is reasonably pertinent to the particular problem with which the inventor is involved," even though the reference is not within the field of the inventor's endeavor.

30. Counsel has informed me, and I understand, in considering the obviousness of a claimed invention, one should not view the invention and the prior art with the benefit of hindsight. In an obviousness analysis, one must be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. Counsel has instructed me and I understand that when considering obviousness, I should not consider what is known today or what was learned from the asserted patents. Instead, obviousness is assessed by the person of ordinary skill in the art at the time the invention was made.

31. In this regard, I am informed and I understand that the '701 Patent may not be used as a guide to select and understand the prior art, because such an approach, at a minimum, discounts the value of combining various existing features or principles in a new way so as to achieve a new result.

32. Counsel has informed me, and I understand, that obviousness cannot be predicated on what was unknown at the time of the invention, even if the inherency of a certain feature is later established. Counsel has also informed me, and I understand, that unknown properties of the prior art may not be relied upon to provide the rationale for modifying or combining the prior art to reach the claimed

Nokia Exhibit 2001, p.12
Amazon.com v. Nokia
IPR2024-01175

subject matter.

33. Counsel has informed me, and I understand, that a reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant of the '701 Patent.

V. OVERVIEW OF VIDEO COMPRESSION AND DECOMPRESSION TECHNOLOGY

34. Video compression (a.k.a. "video encoding" or "video coding") involves a series of data processing steps performed on an input digital video sequence to produce an output "encoded bitstream." The encoded bitstream can then be stored (e.g., on a DVD or Blu-Ray disk, USB storage device, SD card, or video streaming server disk drive) and later transmitted to another device, a "video decoder," which performs the video decompression (a.k.a. "video decoding") operation and displays a representation of the original video to an end user.

35. For video conferencing or digital video broadcast of live events, the system is typically designed to perform encoding, transmission and decoding in "real time," such that these operations all happen quickly enough that the end user perceives no or little delay (or "latency").

36. The encoded bitstream contains fewer bits than the uncompressed ("raw") video that is input to the encoder, allowing for fewer resources for storage

Nokia Exhibit 2001, p.13
Amazon.com v. Nokia
IPR2024-01175

and transmission (i.e., a smaller disk space or a lower bit rate communication link) versus storage or transmission of the raw video. In fact, a *significant* reduction in the amount of information used to represent video data must be achieved in order to enable transmission of digital video sequences over low bandwidth communication networks. Ex. 1001, 2:26–33.

37. For this reason, video compression techniques have been developed which reduce the amount of information transmitted. *Id*. Compression ratios of 250:1 are typical for the commonly-used H.264 video codec. However, video compression is a "lossy" compression scheme, meaning the decoded video is a distorted version of the original, raw video, rather than a 100% accurate version. The goal of a video encoding/decoding scheme, and of a video encoder operating within that scheme, is to encode the original video in a way such that the final, decoded video has acceptable quality to a viewer (and acceptable latency too, if real-time), while using the fewest number of bits in the encoded bitstream.

38. The '701 Patent provides a "Background of the Invention" section that covers the aforementioned data processing steps of video encoding/decoding in significant detail. *See* Ex. 1001, 1:15–15:61. I reproduce or paraphrase some of the '701 Patent's Background section for my overview description herein, and provide additional explanation based on my knowledge and experience.

Nokia Exhibit 2001, p.14
Amazon.com v. Nokia
IPR2024-01175

A. Representation of Digital Video

39. An uncompressed digital image comprises an N-by-M array of image picture elements, or "pixels," N being the integer number of columns in each horizontal row of pixels, and M being the number of rows in each vertical column of pixels. *See id*., 1:17–22. In turn, each pixel is represented by a certain number of bits, which carry information about the luminance and/or color content of the region of the image corresponding to the pixel. Commonly, a so-called YUV color model is used to represent the luminance and chrominance content of the image. The luminance, or Y, component represents the intensity (brightness) of the image, while the color content of the image is represented by two chrominance components, labeled U and V. *See id*., 1:22–44.

40. Note that it is possible to represent an image using color models other than YUV, such as the RGB (Red, Green, Blue) color model. The RGB color model represents each pixel using three numbers, but instead of the three numbers representing luminance and two color components as in the YUV model, the three numbers represent the amount of red, green and blue (the primary colors of light) in the pixel, respectively. *Id*., 1:30–33. It is possible to convert a YUV representation to an RGB representation, and vice versa, using linear transform operations.

41. However, color models based on a luminance/chrominance representation provide advantages compared with color models that are based on

Nokia Exhibit 2001, p.15
Amazon.com v. Nokia
IPR2024-01175

primary colors such as RGB. These advantages stem from the nature of the human visual system, which is more sensitive to intensity variations than it is to color variations. YUV color models typically exploit this property by using a lower spatial resolution for the chrominance components (U, V) than for the luminance component (Y). In this way the amount of information needed to represent the color information in an image can be reduced without a noticeable reduction in perceived image quality. *Id*., 1:33–44.

42. The lower spatial resolution of the chrominance components is usually attained by sub-sampling, e.g. of an input video that originally has a chrominance resolution that is the same as the luminance resolution. Typically, a block of 16×16 image pixels is represented by one block of 16×16 pixels comprising luminance information and the corresponding chrominance components are each represented by one block of 8×8 pixels representing an area of the image equivalent to that of the 16×16 pixels of the luminance component. The chrominance components are thus spatially sub-sampled by a factor of 2 in the x (horizontal) and y (vertical) directions. The resulting assembly of one 16×16 pixel luminance block and two 8×8 pixel chrominance blocks is commonly referred to as a YUV macroblock, or macroblock, for short. *Id*., 1:45–57.

43. Digital video *sequences* comprise a sequence of still images ("frames"), the illusion of motion being created by displaying the images one after the other at

Nokia Exhibit 2001, p.16
Amazon.com v. Nokia
IPR2024-01175

a given frame rate. *Id.*, 3:19–23. Consecutive frames in a video sequence tend to be similar and to contain a considerable amount of redundant information. For example, a typical scene may comprise some stationary elements, such as background scenery, and some moving areas, which may take many different forms. Alternatively, the camera recording the scene may itself be moving, in which case all elements of the image have the same kind of motion. In many cases, this means that the overall change between one video frame and the next is often small. *See id*., 3:29–44. Video codec algorithms use this redundancy to form predictions of most macroblocks based on previously-processed macroblocks, allowing for reductions in the number of bits in the encoded bitstream.

B. Intra-prediction and Inter-prediction

44. Video compression methods are based on reducing the redundant and perceptually irrelevant parts of video sequences. The term "spatial redundancy" is used to describe the correlation between neighboring pixels within a frame. The term "temporal redundancy" expresses the fact that the objects appearing in one frame of a sequence are likely to appear in subsequent frames, possibly shifted due to motion from frame to frame.

45. Video coding systems at the time of the '701 Patent made use of "motion compensated prediction." This is a form of temporal redundancy reduction in which the content of some (often many) frames in a video sequence is "predicted"

Nokia Exhibit 2001, p.17
Amazon.com v. Nokia
IPR2024-01175

from other frames in the sequence by tracing the motion of objects or regions of an image between frames. *See id*., 3:46–62. Temporally predicted images are called INTER-coded or P-frames. In the case of INTER frames, the predicted (motion-compensated) image is rarely precise enough to represent the image content with sufficient quality, and therefore a prediction error (PE) frame—representing the difference—is also associated with each INTER frame. *See id*., 3:62–4:14.

46. Frames of a video sequence which are not compressed using motion-compensated prediction are referred to as INTRA-coded or I-frames. Generally, INTRA-coded frames are encoded by applying block-based transform coding directly to the pixel values of the frame to be coded. Additionally, where possible, blocks of INTRA-coded frames are predicted from previously coded blocks within the same frame. This technique, known as INTRA-prediction, has the effect of further reducing the amount of data required to represent an INTRA-coded frame. *See id*., 4:15–4:24.

C. The Discrete Cosine Transform

47. The basic approach to image compression used in almost all still image and video encoders existing today involves using block-based transform coding of the Prediction Error (which can also refer to the original, unpredicted values if no prediction is used for a given image block). Transform coding uses a two-dimensional frequency transform—typically a Discrete Cosine Transform (DCT) or

Nokia Exhibit 2001, p.18
Amazon.com v. Nokia
IPR2024-01175

equivalent—to transform the spatial information into the frequency domain.

48. The transform process represents all values for an N x M block of pixels (typically an 8 x 8 block) through the use of the resulting N x M transform coefficients. For example, an 8 x 8 DCT operation takes an 8 x 8 array of prediction error (or "residual") values as its input, performs well-known mathematical operations, and outputs an 8 x 8 array of frequency coefficients corresponding to those residual values.

49. The equation to compute a two-dimensional N x N array of DCT coefficients {F (*u*, *v*), *u* = 0 to N-1, *v* = 0 to N-1} from a two-dimensional input array {f (*i*, *j*), *i* = 0 to N-1, *j* = 0 to N-1} is as follows:

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos\left(\frac{(2i + 1)u\pi}{2N}\right) \cos\left(\frac{(2j + 1)v\pi}{2N}\right)$$

where

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & , x = 0 \\ 1 & \text{otherwise} \end{cases}$$

Ex. 1022, p. 23. This is a reversible process; the N x N array of frequency coefficients F (*u*, *v*) may then be input to an "inverse DCT" (IDCT) operation that restores the original N x N input array f (*i*, *j*). The IDCT equation is as follows:

Nokia Exhibit 2001, p.19
Amazon.com v. Nokia
IPR2024-01175

$$f(i,j) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\left(\frac{(2i+1)u\pi}{2N}\right)\cos\left(\frac{(2j+1)v\pi}{2N}\right)$$

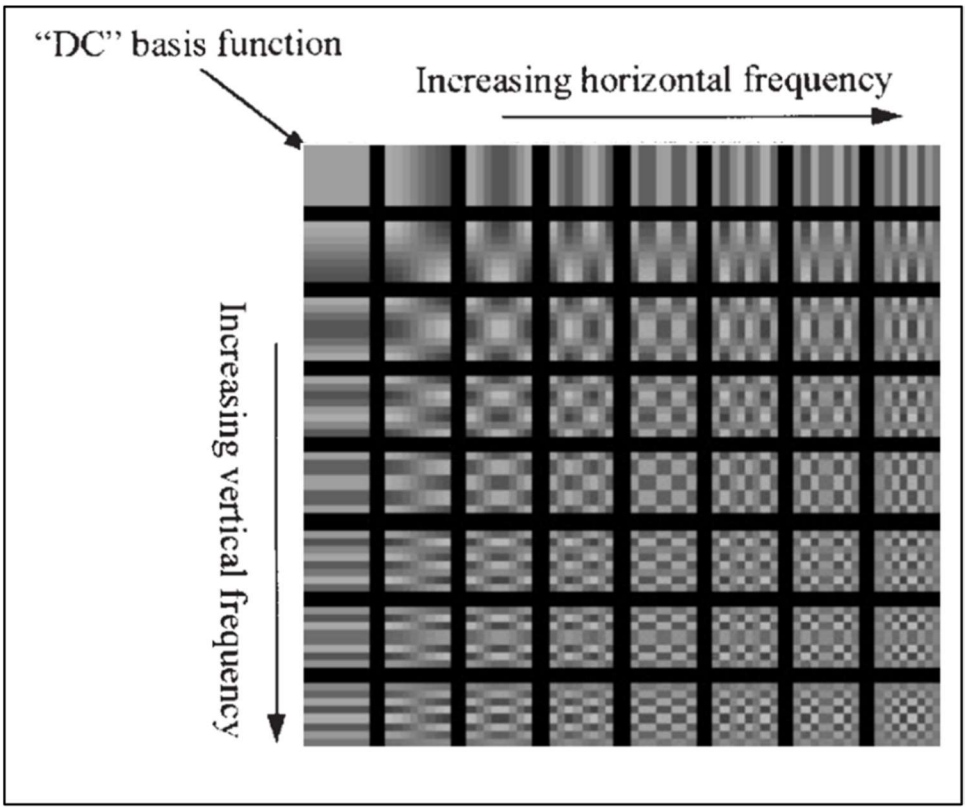
with $C(x)$ as above.

Id.

50. The DCT is a form of frequency transform, in that it "decomposes" an input image into is "frequency components." These frequency components are represented by a set of N^2 "basis functions" that are found in both the equations above: $\cos\left(\frac{(2i+1)u\pi}{2N}\right)\cos\left(\frac{(2i+1)v\pi}{2N}\right)$, u = 0 to N-1, v = 0 to N-1. For N = 8, the 64 basis functions can be visualized as follows:[2]

[2] This diagram is from Riley and Richardson (Ex. 1022), but modified using basis function images that can be seen more clearly. *See* Ex. 1022, p. 22, Figure 3.3 (modified with a similar but clearer image from https://www.mathworks.com/help/images/discrete-cosine-transform.html).

Nokia Exhibit 2001, p.20
Amazon.com v. Nokia
IPR2024-01175



The "Increasing horizontal frequency" arrow represents the *u* values starting at 0 on the left and ending at 7 on the right; the "Increasing vertical frequency" arrow represents the *v* values starting at 0 on the top and ending at 7 on the bottom.

51. As seen from the DCT and IDCT equations above, a frequency transform "decomposes" an image into a weighted sum of the basis functions, whereby the weight for each basis function $\cos\left(\frac{(2i+1)u\pi}{2N}\right)\cos\left(\frac{(2i+1)v\pi}{2N}\right)$ is the corresponding frequency coefficient F (*u*, *v*). The basis functions are designed such that any given input can be represented by a specific, unique weighted combination of them. As such, each coefficient value F (*u*, *v*) can be viewed as representing the "amount" that its respective basis function "contributes" to the image.

Nokia Exhibit 2001, p.21
Amazon.com v. Nokia
IPR2024-01175

52. For example, the DCT of image pixels with zero variation across the DCT input block (e.g. from an image of a blank wall) will only have one nonzero frequency component, F (0, 0).[3] In contrast, the DCT of an image with high variation and fine-grained detail across the DCT input block will typically have large values of F (u, v) for the higher indexes of u and v.[4]

53. As discussed above, for video encoding of a given block, an encoder performs a DCT of either: (a) residual values remaining after prediction values are applied, or (b) original image values if there is no prediction used for that block. A representation of the resulting frequency coefficients from the DCT is inserted into the encoded bitstream, and the decoder uses that representation to then perform an IDCT and re-generate the residual values or original image values.

54. Transform coefficients are typically arranged and depicted so that the lower indexed (i.e., lower spatial frequency) coefficients are included in the top left corner of the block. These coefficients represent low frequency detail which is more noticeable to the human visual system than higher frequency detail. The coefficients

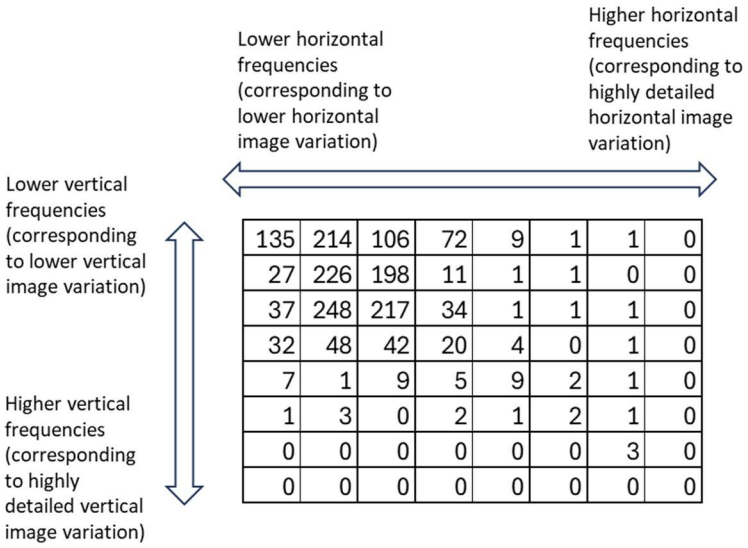
---

[3] F (0, 0) is also called the "DC" coefficient. See the "DC basis function" arrow in the diagram above, pointing to the top left image, which is uniform with no variation. The DC coefficient is the weight for the DC basis function.

[4] All the F (u, v) values other than F (0, 0) are called "AC" coefficients. The 63 images in the diagram above other than the DC basis function are "AC basis functions"; each has a corresponding AC coefficient {F (u, v), $u \neq 0$ and/or $v \neq 0$}.

Nokia Exhibit 2001, p.22
Amazon.com v. Nokia
IPR2024-01175

in the lower right of the block represent higher frequency content that add fine details to the IDCT reconstruction, and these fine details are often not noticeable to the human eye. A table of exemplary transformed coefficient values (i.e., a possible output of a DCT) is provided in the diagram below:



135	214	106	72	9	1	1	0
27	226	198	11	1	1	0	0
37	248	217	34	1	1	1	0
32	48	42	20	4	0	1	0
7	1	9	5	9	2	1	0
1	3	0	2	1	2	1	0
0	0	0	0	0	0	3	0
0	0	0	0	0	0	0	0

55. An encoder may generate a table of DCT-transformed values such as the above for each set of 8 x 8 pixels, typically resulting in many thousands of 8 x 8 DCT operations per second of video.

56. The DCT and IDCT are used in video coding because frequency transformed coefficients corresponding to actual image residual values tend to have large values concentrated in fewer coefficients, with many other coefficients being zero or close to zero. Having fewer larger-valued ("significant") coefficients to encode (versus all 64 coefficients) enables more effective use of the next steps, quantization and run-level coding, as discussed below.

Nokia Exhibit 2001, p.23
Amazon.com v. Nokia
IPR2024-01175

D. Quantization of DCT Coefficients

57. The frequency coefficients are not included in the encoded bitstream using their full, original precision. Instead, they are "quantized," which reduces their precision, sometimes significantly. The quantization operation reduces bits in the encoded bitstream but lowers image quality, because the 8 x 8 residual values reconstructed by the IDCT in the decoder are not the original 8 x 8 residual values that were input to the DCT in the encoder.

58. At the simplest level, quantization by a quantization level N means rounding to the nearest multiple of N, or rounding down to the nearest multiple of N.[5] For example, if a variable has a possible range of 0 to 99, then quantizing it using a quantization level of 10 would effectively approximate any value 0–9 as zero, any value 10–19 as 10, any value 20–29 as 20, and so on, up to any value 90–99 being approximated as 90. It takes fewer bits to encode one of ten possible values instead of one of 100 possible values, so this operation trades number of encoded bits for precision. If, for example, the quantization level were 50 instead, then any value 0–49 would be approximated as 0 and any value 50–99 would be approximated as 50. In this extreme example, the value could be represented as a single bit (a zero

[5] The examples herein use rounding down for simplicity.

Nokia Exhibit 2001, p.24
Amazon.com v. Nokia
IPR2024-01175

means 0 and a one means 50), but the loss of precision is huge.

59. After quantization, many coefficients at higher transform frequencies may be zero. Using the 8x8 table of transformed coefficients above, if those coefficient values are quantized to multiples of 4, and then divided by 4 for compactness, the resulting quantized coefficients would be in the table on the right below:

Transform Coefficients

135	214	106	72	9	1	1	0
27	226	198	11	1	1	0	0
37	248	217	34	1	1	1	0
32	48	42	20	4	0	1	0
7	1	9	5	9	2	1	0
1	3	0	2	1	2	1	0
0	0	0	0	0	0	3	0
0	0	0	0	0	0	0	0

Quantized Coefficients

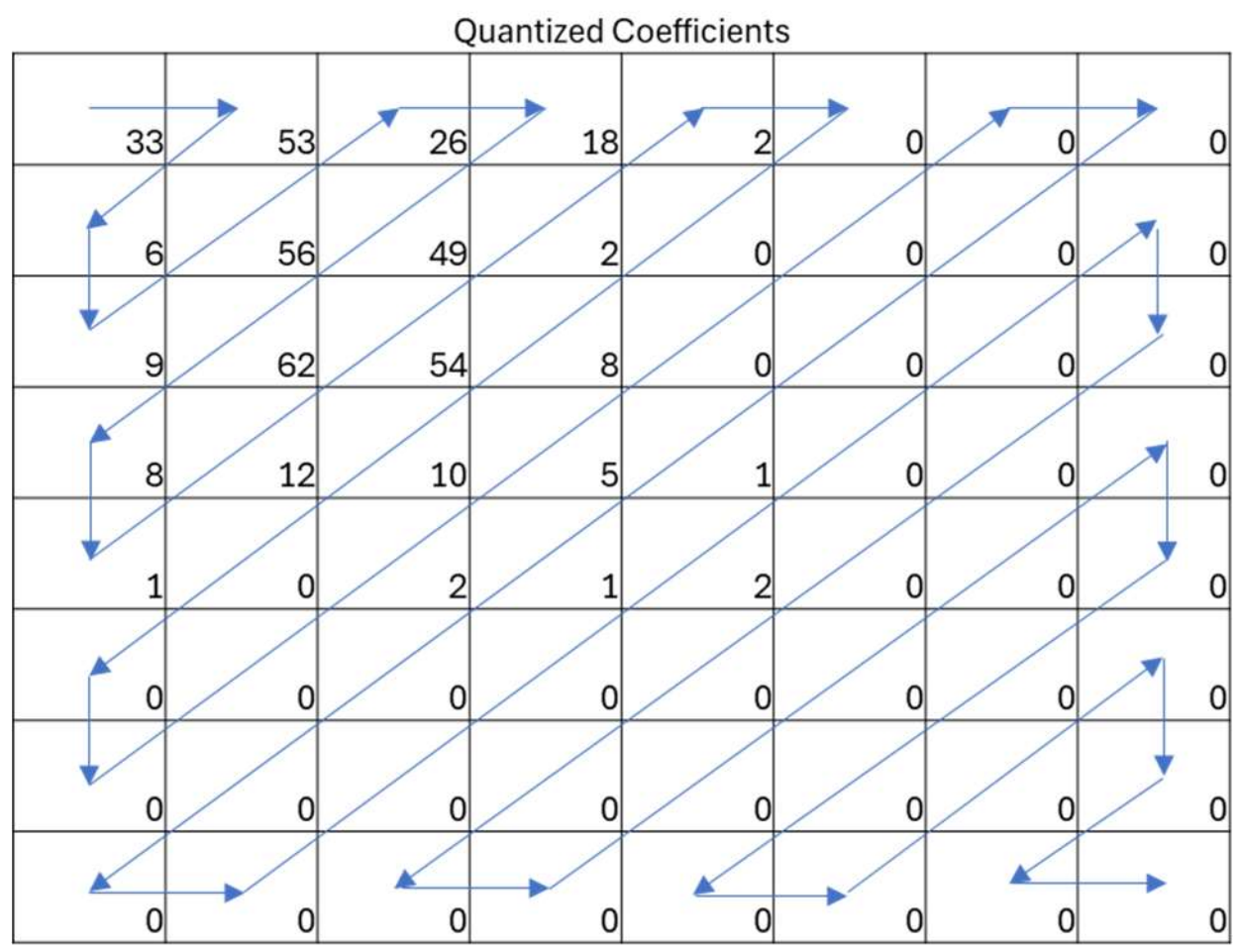
33	53	26	18	2	0	0	0
6	56	49	2	0	0	0	0
9	62	54	8	0	0	0	0
8	12	10	5	1	0	0	0
1	0	2	1	2	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

For example, for the upper left entry, 135/4 = 33 ¾, and dropping the remainder gives a quantized coefficient of 33. The decoder decodes the value 33 and also knows the quantization parameter 4 that the encoder used (having decoded it from earlier in the bitstream), but it calculates 33 x 4 = 132, which is close to but not exactly the original 135 value, so some distortion in the decoded video has been introduced by the process. The other 63 coefficients are also quantized in the encoder—and the quantized versions communicated to the decoder in the encoded bitstream—in a similar manner. Note that, in this example, all values in the left table that are less than four are quantized to zero in the table on the right above.

Nokia Exhibit 2001, p.25
Amazon.com v. Nokia
IPR2024-01175

E. Run-level Coding

60. The encoder and decoder must agree on the order in which the quantized coefficients are included in the encoded bitstream. Video codecs use a "zig-zag" scan as shown below:



A zig-zag scan of the table above results in the number sequence {33, 53, 6, 9, 56, 26, 18, 49, 62, 8, 1, 12, 54, 2, 2, 0, 0, 8, 10, 0, 0, 0, 0, 2, 5, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}. The '701 patent depicts a similar concept in its Fig. 6a, reproduced below:

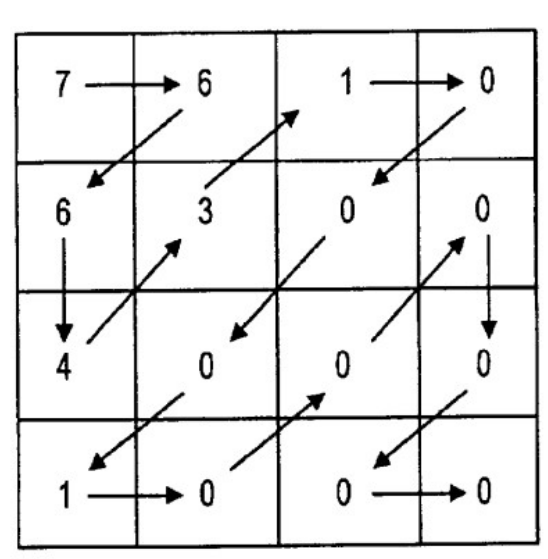


FIG. 6a

Id., Fig. 6a, 14:46–65.

61. With zig-zag ordering in video coding, there are often long runs of zeros between non-zero coefficients, especially towards the end. Therefore, rather than transmitting all 64 quantized coefficients of an 8 x 8 transform, video codecs use "run-level" encoding.[6] Run-level encoding represents each non-zero quantized coefficient in the one dimensional array by two values, referred to as level and run. Level is the value of the quantized coefficient and run is the number of consecutive zero-valued coefficients preceding the coefficient in question. The run and level values for a given coefficient are ordered such that the level value precedes the associated run value. A level value equal to zero is used to indicate that there are no more non-zero coefficient values in the block. This 0-level value is referred to as an EOB (end-of-block) symbol. *Id*., 6:7–17.

62. Using the example sequence above in the zig-zag order depicted, i.e. {33, 53, 6, 9, 56, 26, 18, 49, 62, 8, 1, 12, 54, 2, 2, 0, 0, 8, 10, 0, 0, 0, 0, 2, 5, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, the {level, run} pairs would be {33, 0}, {53, 0}, {6, 0}, {9, 0}, {56, 0}, {26,

[6] Run-level encoding as described in the '701 patent is a form of "run length" encoding, and the terms may be used interchangeably in this context. *See, e.g.*, '701 Patent, claim 27 (which recites "run-length," while other claims recite "run-level").

0}, {18, 0}, {49, 0}, {62, 0}, {8, 0}, {1, 0}, {12, 0}, {54, 0}, {2, 0}, {2, 0}, {8, 2}, {10, 0}, {2, 4}, {5, 0}, {1, 6}, {1, 0}, {2, 6}, {0}. In this example, 64 original values can be represented instead by 45 values, and the process is lossless (i.e., the inverse operation reproduces the original values exactly).

63. The '701 patent depicts a similar concept in its Fig. 6b, reproduced below:

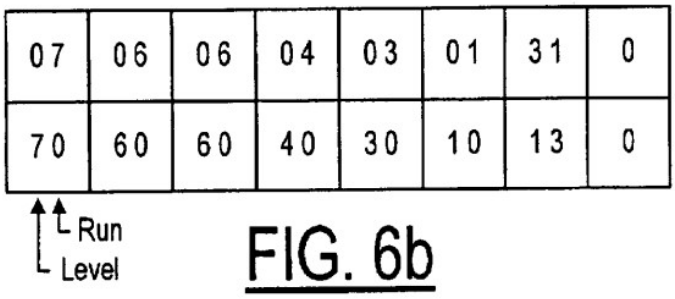


FIG. 6b

Id., Fig. 6b, 14:46–65 (note that the top row alone shows the run-level pairs; the bottom row is simply the top row with the values in each pair swapped to indicate level first then run).

F. Entropy Coding

64. As discussed above, for each block, the encoder performs intra prediction, inter prediction or no prediction. Intra prediction requires information about the reference blocks to be included in the encoded bitstream, so that the decoder knows what reference blocks to use too. Likewise, inter-prediction requires information about the reference blocks and the motion vectors to be included in the

Nokia Exhibit 2001, p.28
Amazon.com v. Nokia
IPR2024-01175

encoded bitstream. For each of intra prediction, inter prediction or no prediction, the encoder also typically generates quantized DCT coefficients for the block. These quantized DCT coefficients are then represented as a set of run-level pairs, and the run-level pair information must be included in the encoded bitstream as well. Information about other options the encoder chooses, such as quantization parameter and coding modes, are included in the encoded bitstream too, as well as markers indicating how to parse the bitstream itself. These different pieces of information may originate as bit fields, integers, fixed-point fractional values, integer value pairs, etc., and they all must somehow be included in the encoded bitstream.

65. Rather than concatenating bit field representations of all these values as-is into an encoded bitstream (e.g., by using their native binary representation in the computing device that generates them), the values are yet again encoded further using a lossless procedure known as entropy coding. The entropy-coded representations are what is inserted into the encoded bitstream.

66. Entropy coding exploits the fact that "symbols" within a data set to be coded generally have different probabilities of occurrence. A symbol can be viewed as a possible value that is to be entropy encoded, be it a single bit, multiple bits, an integer, a pair or vector of values, etc.

67. Instead of using a fixed number of bits to represent each symbol, a

Nokia Exhibit 2001, p.29
Amazon.com v. Nokia
IPR2024-01175

variable number of bits is assigned such that symbols which are more likely to occur are represented by "codewords" having fewer bits. For example, if the symbols are (run, level) pairs, certain values of levels and runs are more likely to occur than other values. Entropy coding techniques can be used effectively to reduce the number of bits required to represent the run and level values on average. Ex. 1001, 6:19–32.

G. Variable Length Coding (VLC)

68. A number of different methods can be used to implement entropy coding. For example, entropy coding of the run and level parameters may be implemented by means of look-up tables which define the mapping between each possible symbol in the data set to be coded and its corresponding variable length code (VLC). Such look-up tables are often defined by prior statistical analysis of training material comprising symbols identical to those to be coded and having similar statistical properties. Ex. 1001, 6:33–40.

69. An example of VLC is provided by VCEG-L28.[7] *See* Ex. 1006. VCEG-L28 proposes two possible look-up tables for performing VLC coding. *Id*., Table 1 (reproduced below):

[7] Note for this example that the symbols are not run-level pairs, but instead are possible values of a single integer.

Nokia Exhibit 2001, p.30
Amazon.com v. Nokia
IPR2024-01175

Table 1. The first nine codewords of UVLC and UVLC2

Index	UVLC	UVLC2
0	1	11
1	001	001
2	011	011
3	00001	1001
4	00011	1011
5	01001	00001
6	01011	00011
7	0000001	01001
8	0000011	01011

70. Given that the table header states "first nine codewords," the table presumably continues on with further rows. However, for this example, we will assume that the table as shown represents all possible input values ("symbols") of a given integer "X" that is to be encoded, i.e., X has an input range of zero through eight as shown in the first column, "Index."

71. Without VLC coding, the value of X could be included in the encoded bitstream as raw bits: a zero is 0000, a one is 0001, a two is 0010, a three is 0011, a four is 0100, a five is 0101, a six is 0110, a seven is 0111 and an eight is 1000. This would require four bits. However, consider instead that the 'UVLC" column of Table 1 above is used to determine the bit sequence that represents each of possible values zero through eight. If the probabilities of the 9 possible values are equal, then the average number of bits required is 1/9 (1 + 3 + 3 + 5 + 5 + 5 + 5 + 7 + 7) = 41/9 = 4.56 bits, which is not a savings over four bits at all. However, consider the case that the probability of a zero value is 60%, and the probability of a one to eight value is 5% each (totaling 100%). In this case, the average number of bits required is (0.6 x

Nokia Exhibit 2001, p.31
Amazon.com v. Nokia
IPR2024-01175

1) + (0.05 x (3 + 3 + 5 + 5 + 5 + 5 + 7 + 7)) = 2.6 bits, which is a significant savings over four bits.

72. Using the UVLC2 column of Table 1 instead, the equal probability case results in 1/9 (2 + 3 + 3 + 4 + 4 + 5 + 5 + 5 + 5) = 36/9 = 4 bits, the same as no coding. For the case where the probability of a zero value is 60%, and the probability of a one to eight value is 5% each, the average number of bits required is (0.6 x 2) + (0.05 x (3 + 3 + 4 + 4 + 5 + 5 + 5 + 5)) = 2.9 bits, which is still a savings over 4 bits.

73. Note that in both columns of Table 1 above, no codeword begins with another codeword. For example, in the UVLC column, a zero is encoded as a "1" bit, and none of the other encodings for values one through eight start with a "1." This allows the decoder to always know when it has reached the end of a codeword, assuming no bit errors in storage or transmission. Any following bits in the encoded bitstream then must start a new codeword for a new value to be decoded.

74. As seen from the cases above, the benefit of using VLC depends heavily on 1) the probabilities of the possible values of a given input value that is to be coded, and 2) developing a table mapping that uses unequal input probabilities to good advantage. Even for a coding scheme as conceptually simple as VLC, significant research and experimentation is performed on representative video sets to analyze these probabilities and choose the best table mappings. For example,

Nokia Exhibit 2001, p.32
Amazon.com v. Nokia
IPR2024-01175

VCEG-L28 itself was submitted to the H.26L standards effort as a proposed improvement for a VLC coding scheme in a new H.26L video codec, and VCEG-L28 was submitted with a Microsoft Excel Worksheet showing the results and benefit of the submitters' decisions and experimentations.[8]

75. As described above, Variable Length Coding is simply a defined mapping of possible input data bit fields to other, unique coded bit fields that are then included in the coded video bitstream and provided to the decoder. The decoder then performs the known, reverse mapping for each VLC-coded field.

H. Arithmetic Coding

76. An alternative coding technique, known as Arithmetic Coding, can be used instead of VLC coding to losslessly encode video parameters such as run and level values. Arithmetic coding uses fractional values between zero and one as codewords, and is more complicated than VLC. Exhibit 1020, "Arithmetic Coding for Data Compression" provides an explanation and example of Arithmetic Coding. I provide further explanation and different examples below.

77. Consider the decimal value 0.345. This is the same as the fraction 345/1000. It is also the same as $3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3}$. Now consider the

[8] *See* Ex. 1006. The Excel file itself was not submitted with the Petition, but the icon at the end of VCEG-L28 and related text describe it.

Nokia Exhibit 2001, p.33
Amazon.com v. Nokia
IPR2024-01175

decimal value 0.345… whereby the "…" represents anywhere between zero and infinite additional decimal digits after the 5. No matter what additional digits may follow the "5," the resulting value is still in the range [.345, .346), where "[" means inclusive and ")" means not inclusive. For example, both 0.345000000000000000001 and 0.3459999999999999999 are still within the range [.345, .346), a range which has a span of 1/1000. Similarly, consider the decimal value 0.2…, whereby the "…" represents anywhere between zero and infinite additional digits after the 2. No matter what additional digits may follow the "2," the resulting value is still in the range [.2, .3), a span of 1/10. Arithmetic coding effectively maps possible input symbol sequences to contiguous, mutually exclusive ranges between [0 and 1). When the encoder has to determine a codeword for a particular input symbol sequence, it chooses the *shortest* fractional value that is wholly within that input symbol sequence's assigned range, and uses that as the codeword.

78. The decoder has knowledge of the same mapping of possible input symbol sequences to contiguous, mutually exclusive ranges between [0 and 1) that is used by the encoder, so upon receiving the codeword, it can determine the specific range within which the received codeword falls, and map that back to the original input symbol sequence.

79. The example in Ex. 1020 and the example above use decimal fractions,

Nokia Exhibit 2001, p.34
Amazon.com v. Nokia
IPR2024-01175

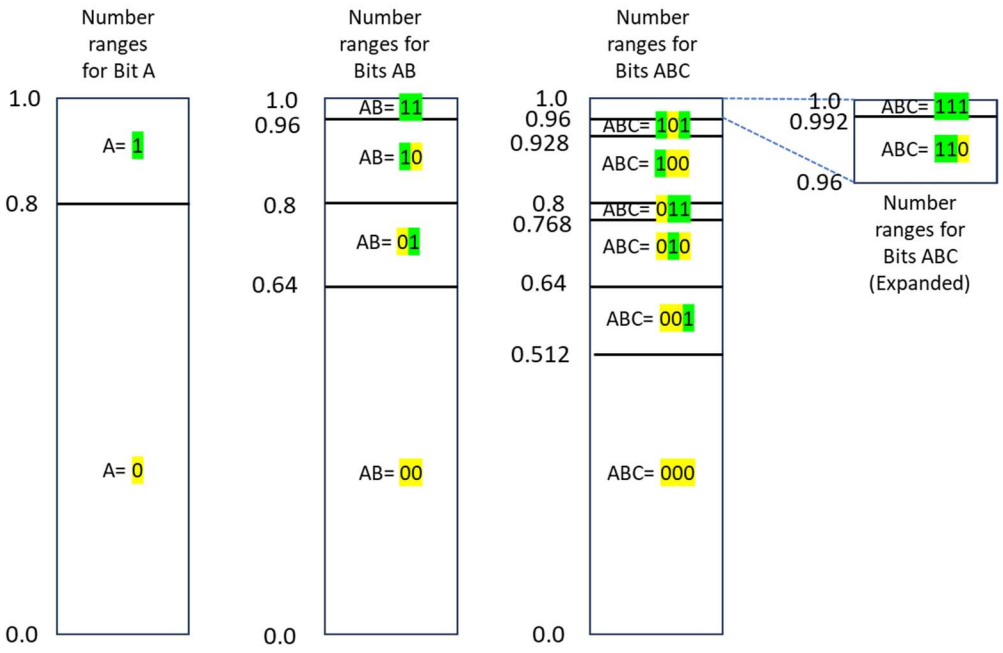
but typical systems use binary fractions, in which case base-2 is used instead of base-10, and the only allowed symbols are 0 and 1. An example of a binary fraction is 0.10010011, which is equivalent to the base-10 representation $1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6} + 1 \times 2^{-7} + 1 \times 2^{-8}$, or 147/256. No matter what binary digits may follow the 0.10010011, e.g., 0.10010011000000000000000001 or 0.10010011101101010101010010011 or 0.10010011111111111111111111111, or any of the other infinite possibilities, the resulting value is still in the base-10 range [147/256, 148/256). Therefore, the codeword "10010011" could be used to represent a particular input symbol sequence if that symbol sequence happens to have an assigned (i.e., mapped) range that wholly includes [147/256, 148/256). Similarly, the single digit codeword "0" could be used to represent a particular input symbol sequence if that symbol sequence has an assigned mapping that wholly includes [0, .5), since any binary fraction 0.0… (whereby the "…" represents anywhere between zero and infinite additional zero or one bits after the first fractional zero) is in the range [0, .5). This last case in fact occurs in an example below.

80. Based on the two examples of binary fractions above, it is seen that input symbol sequences mapped to wider ranges within [0, 1) can generally be represented with shorter codewords, because wider ranges can be represented by fewer required fractional bits. Arithmetic coding does, in fact, map more probable

Nokia Exhibit 2001, p.35
Amazon.com v. Nokia
IPR2024-01175

input symbol sequences to wider number ranges within the range [0, 1), allowing them to be represented by shorter codewords.

81. An example of the above-described binary arithmetic coding is presented here. Suppose three binary digits (bits) ABC are to be arithmetic encoded, and then later decoded back to their original values. Furthermore, assume it is known beforehand that for each bit, the probability of a zero is 80% and the probability of a one is 20%. The diagram below depicts how the eight possible values of ABC (i.e., 000, 001, 010, 011, 100, 101, 110, 111) would be mapped to consecutive number ranges within [0, 1.0).



82. First, consider bit A. A zero value for A is mapped to the range [0, 0.8) and a one value for A is mapped to the range [0.8, 1), as shown in the left "column"

Nokia Exhibit 2001, p.36
Amazon.com v. Nokia
IPR2024-01175

of the diagram above. This reflects the 80% probability that A is a zero. If the probability were different than 80%, the boundary between the zero and one ranges would be that different probability instead. Then, the A = 0 range is further subdivided for the cases AB = 00 and AB = 01, as shown in the middle column of the diagram above. Likewise, the A = 1 range is further subdivided for the cases AB = 10 and AB = 11. The two further subdivisions are still each in the ratio of 80%/20%, but this 80%/20% ratio would be different if the probability of B = 0 were not also 80%. Note, for example, that the range for the AB = 11 case is from 0.96 to 1, which is a span of 0.04, which is in fact the probability that AB = 11 (i.e., 20% x 20%, or 0.2 x 0.2) The spans of the other three 2-bit possibilities (00, 01 and 10) are equal to the probabilities of those possibilities, respectively, as well.

83. The third column of the diagram above, including its "blow-up" illustration to the right for the small ranges of the highest values, shows the final number ranges for the eight possibilities of bits ABC. This is performed by further subdividing each of the four ranges for bits AB, each subdivision adding two further ranges, one for the case that C = 0 and one for the case that C = 1. Since the probability that C = 0 is also 80%, the four further subdivisions are also performed using an 80%/20% ratio. If the probability of C = 0 were other than 80%, each of the four further subdivided ranges would be adjusted according to that different probability.

Nokia Exhibit 2001, p.37
Amazon.com v. Nokia
IPR2024-01175

84. The number ranges shown in the diagram above for the eight possible values of ABC can be summarized as follows:

Three bit input:	Range beginning:	Range end:
000	0	0.512
001	0.512	0.64
010	0.64	0.768
011	0.768	0.8
100	0.8	0.928
101	0.928	0.96
110	0.96	0.992
111	0.992	1

Both the encode and the decoder have knowledge of these ranges, i.e. of the table above, because the ranges are either 1) pre-agreed by design of the algorithm itself, 2) communicated in the encoded bitstream ahead of corresponding codewords, and/or 3) adjusted using the actual count of previously transmitted/received ones and zeros to change the probabilities and the corresponding ranges empirically. If the ranges are allowed to vary from one encoding/decoding operation to the next based on statistics of the data being encoded, as in 3) above and possibly 2) above, the coding is denoted "Context-based Adaptive Binary Arithmetic Coding," or "CABAC." *See also* Ex. 1001, 12:35–49.

85. Given the table of ranges for possible ABC values above, the encoder determines that the three bits ABC have arithmetic codewords as follows:

Nokia Exhibit 2001, p.38
Amazon.com v. Nokia
IPR2024-01175

ABC bits	Codeword	Codeword length
000	0	1
001	1001	4
010	1011	4
011	110010	6
100	1101	4
101	111100	6
110	111110	6
111	1111111	7

The one bit codeword "0" can be assigned to ABC = 000, because any binary fraction that starts with 0.0… is within the range [0, ½), which is wholly inside [0, 0.512), the range allocated to ABC = 000. However, the single bit "1" cannot be used as a codeword, since 0.1… by itself, representing fractions from [1/2, 1), does not uniquely isolate any one of the defined ranges. However, the codeword 1001 represents binary fraction 0.1001 = 9/16, and thus the range [9/16, 10/16) = [0.5625, 0.625), which is wholly within the range of ABC = 001, i.e., [0.512, 0.64). Therefore, the codeword 1001 is used to represent ABC = 001, and the decoder, recognizing that codeword 1001 is within the range of ABC = 001, correctly decodes ABC = 001 when that codeword is sent. The same concept is applied for the other possible ABC values. Lastly, ABC = 111 is very rare, with a 0.008 (i.e., 0.8%) probability of occurrence, so its range is only [0.992, 1), a span of 0.008. The shortest binary fraction that spans this small range is 0.1111111, corresponding to 127/128 = 0.9921875 in base-10. It thus represents a span of [0.9921875, 1), which indeed is entirely within the range [0.992, 1) for ABC =

Nokia Exhibit 2001, p.39
Amazon.com v. Nokia
IPR2024-01175

111. Note that even one bit less for the codeword, i.e., "111111," would not work, because that codeword's range is [63/64, 1) = [0.984374, 1), which could not distinguish between ABC = 110 and ABC = 111 (because it spills into a portion of ABC = 110's range).

86. The average number of codeword bits in the scheme described above is the probability of occurrence of each of the eight possible ABC values times its respective codeword's length, or 0.512 * 1 + 0.128 * 4 + 0.128 * 4 + 0.032 * 6 + 0.128 * 4 + 0.032 * 6 + 0.032 * 6 + 0.008 * 7 = 2.68 bits. This is a savings of 10.7% over sending the original three bits uncoded.

87. In the example just above, three bits are communicated. These three bits could represent a single integer having a range of possible values 0 to 7. In arithmetic coding, each bit ABC would be considered a "bin" and each different probability model a "context." Since in this example all three bins have the same probability of a zero, all three bins can be grouped with the same context. If the probability for one of the bins were to change or be different from the start, that would represent a distinct context. Each bin contains a "symbol," which in this case is one of two possible values: zero or one.

88. More generally, a bit field or integer can be mapped to bins and symbols in any number of ways prior to arithmetic coding. As usual, the encoder and decoder

Nokia Exhibit 2001, p.40
Amazon.com v. Nokia
IPR2024-01175

must both use the same "symbol mapping" and "symbol de-mapping" scheme, respectively, in order for the decoder to decode the correct, originally encoded values. The '701 Patent describes:

> CABAC is a binary arithmetic coding method and thus data symbols to be coded which have non-binary values are first converted to binary values ('binarized') in binary mapping block **710**. The binarization process involves mapping a symbol to be coded to a sequence of bins, each of which has a corresponding bin number and can be assigned a value of either 0 or 1. An example of such a mapping is given below in Table 1. In principle other binarization schemes can be envisaged.

TABLE 1

value	Bin Sequence							
0	1							
1	0	1						
2	0	0	1					
3	0	0	0	1				
4	0	0	0	0	1			
5	0	0	0	0	0	1		
6	0	0	0	0	0	0	1	
. . .	.	.	.	.	.	.	.	.
bin_nr.	1	2	3	4	5	6	7	.

> In the CABAC method each of the bins is assigned to a so-called 'context' (hence the name context-based arithmetic coding). A context can be thought of as grouping together bins, which have similar statistical characteristics. In other words, each bin assigned to a particular context is assumed to have a similar probability of containing the value 1 or 0 as the other bins belonging to that context. In this way, the probability estimates used to generate code-words in the arithmetic coder are defined for each context rather than for each possible bin to be encoded. Each context is defined according to a 'context model', established in advance and based on information about the

Nokia Exhibit 2001, p.41
Amazon.com v. Nokia
IPR2024-01175

statistical characteristics of the data symbols (and thus the bins) to be encoded.

Ex. 1001, 12:42–13:10.

89. In Table 1 above from the ’701 Patent, a non-negative integer (call it “X”) is mapped to a varying-length bin sequence before the bins are even CABAC encoded. For example, if X = 2, the symbol-mapped bin sequence into the CABAC encoder is 001; if X = 6, it is 0000001; and by extension, if X= 10, it is 00000000001. This symbol mapping is a simple table mapping; as such it can be viewed somewhat as a form of VLC coding that occurs before the CABAC encoding even takes place.

90. Based on the above discussions starting with Section V.F, CABAC is shown to be much different and more complex than Variable Length Coding. VLC defines a straightforward translation from inputs to codewords and back, whereas CABAC involves complex arithmetic and a complex and adaptive arrangement of bins, symbols and contexts. Concepts from VLC cannot simply be carried over to CABAC with the expectation of predicable results.

VI. OVERVIEW OF THE ’701 PATENT’S INVENTION

91. The ’701 Patent, entitled “Method and System for Context-Based Adaptive Binary Arithmetic Coding” and issued to Marta Karczewicz and Ragip Kurceren, claims novel ways to perform context-based adaptive binary arithmetic for image and video coding. These novel ways to perform CABAC are

Nokia Exhibit 2001, p.42
Amazon.com v. Nokia
IPR2024-01175

improvements over the prior-art CABAC at the time of the ’701 Patent.

92. As discussed above, CABAC for video coding typically uses binarization of data elements and subsequent assignment of the binarized data into bins to perform arithmetic coding. The assignment of the bins into different contexts is based on the statistical characteristics of the bins. The statistical characteristics are provided for the contexts based on “context models,” which are “established in advance and based on information about the statistical characteristics of the data symbols (and thus the bins).” Ex. 1001 at 13:5–10.

93. The CABAC method proposed for the H.26L standardization process at the time (H.26L TML8)[9] utilized four contexts for level values and three contexts for run values. With regards to level values, the first context is assigned to bin 1, the second to bin 2, the third to all remaining bits, and the fourth context is used to indicate the sign of the level. *Id*. at 14:14–24. The majority of level values were coded in context 3. *Id*. at 14:66–15:15. For run values, the first context is assigned to bin 1, the second to bin 2, and the third to all remaining bins. *Id*. at 14:20–24.

94. The ’701 Patent also discloses “probability estimates” assigned to contexts. Probability estimates are representations of the statistical characteristics

[9] TML stands for “Test Model Long”; TML8 was the source code implementation of the currently proposed H.26L reference model at that time.

Nokia Exhibit 2001, p.43
Amazon.com v. Nokia
IPR2024-01175

(the likelihood of being a 1 or 0) of the bins assigned to the context in question. The '701 Patent explains that the contexts are constant throughout coding but the probability estimates may be updated. *Id.* at 13:20–43.

95. Importantly, in the prior art (such as H.264 TML8), "the contexts for run and level values depend on the type of block being encoded and the bin number of the binarized level or run value. Different block types are defined according to the Scanning mode (single/double) used to order the coefficient values, the component type (luminance/chrominance, AC/DC), or the coding mode (INTER/INTRA). However, for a given block type, the context depends only on the bin number." *Id*., 14:8–15 (emphasis added).

96. The inventors of the '701 Patent conceived of an improved method of arithmetic coding that is capable of exploiting statistical relationships between the run and level values associated with DCT transform coefficients. As noted in the previous paragraph, prior art video coding algorithms did not exploit these statistical relationships for arithmetic coding, instead relying only on block type and bin number to assign a context. The '701's invention improves CABAC coding efficiency over the prior art by assigning contexts at least partly in dependence on run and level values other than those being presently coded. The '701's invention also improves CABAC coding efficiency over the prior art by assigning contexts for run bins at least partly in dependence on levels, and vice versa. This improved coding

Nokia Exhibit 2001, p.44
Amazon.com v. Nokia
IPR2024-01175

efficiency generally results in fewer coded bits for a given input, thus improving the compression ratio of the video codec overall.

I. Challenged claims 1–40

97. Challenged claims 1–40 recite as follows, using the indexing of the Petition:

> Claim 1[pre]: 1. A method of image coding, wherein an image is divided in an encoder into a plurality of blocks having a plurality of pixels, each pixel having a pixel value, and a transform coding operation is performed on a block of pixels to produce a corresponding block of transform coefficient values, the block of transform coefficient values being scanned in a given scanning order to produce a scanned array of coefficient values arranged according to the scanning order, the method comprising the steps of:
>
> 1[a]: representing the coefficient values in the scanned array by a plurality of number pairs, each of said number pairs having a first number and a second number; and
>
> 1[b]: assigning the first numbers to one of a plurality of contexts representative of the first numbers such that the first number of a first number pair is assigned to a context at least partly in dependence on a first number of a second number pair.
>
> Claim 2: A method according to claim 1, wherein the first number of the first number pair is assigned to a context at least partly in dependence on the first number of a preceding number pair in the scanning order.
>
> Claim 3: A method according to claim 1, wherein said plurality of contexts are contexts of a context-based arithmetic coder.

Nokia Exhibit 2001, p.45
Amazon.com v. Nokia
IPR2024-01175

Claim 4: A method according to claim 1, wherein said plurality of contexts are contexts of a context-based binary arithmetic coder.

Claim 5: A method according to claim 4, further comprising the step of: mapping the first numbers of said number pairs to a set of bins, each of which can be assigned one of either a first or a second value.

Claim 6: A method according to claim 5, wherein mapping of a number to a given one of said set of bins is indicated by assigning the bin to the first value.

Claim 7: A method according to claim 5, wherein the first value is 1 and the second value is 0.

Claim 8: A method according to claim 5, wherein the first number of a number pairs is assigned to a context at least partly in dependence on the bin to which the first number of said number pair is mapped.

Claim 9: A method according to claim 1, wherein said first number is indicative of a non-zero coefficient value and said second number is indicative of a number of consecutive zero coefficient values preceding said non-zero coefficient value.

Claim 10[pre]: A method of image coding, wherein an image is divided in an encoder into a plurality of blocks having a plurality of pixels, each pixel having a pixel value, and a transform coding operation is performed on a block of pixels to produce a corresponding block of transform coefficient values, the block of transform coefficient values is scanned in a given scanning order to produce a scanned array of coefficient values arranged according to the scanning order, the method comprising the steps of:

Nokia Exhibit 2001, p.46
Amazon.com v. Nokia
IPR2024-01175

10[a]: representing the coefficient values in the scanned array by a plurality of number pairs, each of said number pairs having a first number and a second number; and

10[b]: assigning the second numbers to one of a plurality of contexts representative of the second numbers such that the second number of a number pair is assigned to a context at least partly in dependence on the first number of the number pair.

Claim 11: A method according to claim 10, wherein said plurality of contexts are contexts of a context-based arithmetic coder.

Claim 12: A method according to claim 10, wherein said plurality of contexts are contexts of a context-based binary arithmetic coder.

Claim 13: A method according to claim 12, further comprising the step of: mapping the second numbers of said number pairs to a set of bins, each of which can be assigned one of either a first value or a second value.

Claim 14: A method according to claim 13, wherein mapping of a number to a given one of said set of bins is indicated by assigning the bin to the first value.

Claim 15: A method according to claim 13, wherein the first value is 1 and the second value is 0.

Claim 16: A method according to claim 13, wherein the second number of a number pair is assigned to a context at least partly in dependence on the bin to which the second number of the number pair is mapped.

Claim 17: A method according to claim 10, wherein said first number is indicative of a non-zero coefficient value and said second number is indicative of a number of consecutive zero coefficient values preceding said non-zero coefficient value.

Nokia Exhibit 2001, p.47
Amazon.com v. Nokia
IPR2024-01175

Claim 18[pre]: A system for image coding comprising:

18[a]: an input for receiving an image as a plurality of blocks having a plurality of pixels, each pixel having a pixel value;

18[b]: a transform coder for performing a transform coding operation on a block of pixels to produce a corresponding block of transform coefficient values;

18[c]: a scanner for scanning the block of transform coefficient values in a given scanning order to produce a scanned array of coefficient values, arranged according to the scanning order;

18[d]: a run-level coder for representing the coefficient values in the scanned array by a plurality of number pairs, said number pairs having a first number and a second number; and

18[e]: a context-based coder for assigning the first numbers to one of a plurality of contexts representative of the first numbers and operative to assign the first number of a first number pair to a context at least partly in dependence on a first number of a second number pair.

Claim 19: A system according to claim 18, wherein said context-based coder is a context-based arithmetic coder.

Claim 20: A system according to claim 18, wherein said context-based coder is a context-based binary arithmetic coder.

Claim 21: A system according to claim 20, wherein the context-based coder is operative to map the first numbers of said number pairs to a set of bins, each of which can be assigned to one of either a first or a second value.

Nokia Exhibit 2001, p.48
Amazon.com v. Nokia
IPR2024-01175

Claim 22: A system according to claim 21, wherein the context-based coder is operative to map a number to a given one of said set of bins by assigning the bin to the first value.

Claim 23: A system according to claim 21, wherein the first value is 1 and the second value is 0.

Claim 24: A system according to claim 21, wherein the context-based coder is operative to assign the first number of a number pair to a context at least partly in dependence on the bin to which the first number of said number pair is mapped.

Claim 25: A system according to claim 18, wherein the context-based coder is operative to assign the first number of the first number pair to a context at least partly in dependence on the first number of a preceding number pair in the scanning order.

Claim 26: A system according to claim 18, wherein said first number is indicative of a non-zero coefficient value and said second number is indicative of a number of consecutive zero coefficient values preceding said non-zero coefficient value.

Claim 27[pre]: A system for image coding comprising:

27[a]: an input for receiving an image as a plurality of blocks having a plurality of pixels, each pixel having a pixel value;

27[b]: a transform coder for performing a transform coding operation on a block of pixels to produce a corresponding block of transform coefficient values;

27[c]: a scanner for scanning the block of transform coefficient values in a given scanning order to produce a scanned array of coefficient values arranged according to the scanning order;

Nokia Exhibit 2001, p.49
Amazon.com v. Nokia
IPR2024-01175

27[d]: a run-length coder for representing the coefficient values in the scanned array by a plurality of number pairs, said number pairs having a first number and a second number; and

27[e]: a context-based coder for assigning the second numbers to one of a plurality of contexts representative of the second numbers, and operative to assign the second number of a number pair to a context at least partly in dependence on the first number of the number pair.

Claim 28: A system according to claim 27, wherein said context-based coder is a context-based arithmetic coder.

Claim 29: A system according to claim 27, wherein said context-based coder is a context-based binary arithmetic coder.

Claim 30: A system according to claim 29, wherein the context-based coder is operative to map the second numbers of said number pairs to a set of bins, each of which can be assigned to one of either a first or a second value.

Claim 31: A system according to claim 30, wherein the context-based coder is operative to map a number to a given one of said set of bins by assigning the bin to the first value.

Claim 32: A system according to claim 30, wherein the first value is 1 and the second value is 0.

Claim 33: A system according to claim 30, wherein the context-based coder is operative to assign the second number of a number pair to a context at least partly in dependence on the bin to which the second number of the number pair is mapped.

Claim 34: A system according to claim 27, wherein said first number is indicative of a non-zero coefficient value and said second number is

Nokia Exhibit 2001, p.50
Amazon.com v. Nokia
IPR2024-01175

indicative of a number of consecutive zero coefficient values preceding said non-zero coefficient value.

Claim 35[pre]: A computer program embodied in a computer readable medium comprising:

35[a]: machine-readable program code for dividing an image into a plurality of blocks having a plurality of pixels, each pixel having a pixel value;

35[b]: machine-readable program code for performing a transform coding operation on a block of pixel values to produce a corresponding block of transform coefficient values;

35[c]: machine-readable program code for scanning the block of transform coefficient values in a given scanning order to produce a scanned array of coefficient values arranged according to the scanning order;

35[d]: machine-readable program code for representing the coefficient values in the scanned array by a plurality of number pairs, said number pairs having a first number and a second number; and

35[e]: machine-readable program code for assigning the first numbers to one of a plurality of contexts representative of the first numbers such that the first number of a first number pair is assigned to a context at least partly in dependence on a first number of second number pair.

Claim 36[pre]: A computer program embodied in a computer readable medium comprising:

36[a]: machine-readable program code for dividing an image into a plurality of blocks having a plurality of pixels, each pixel having a pixel value;

Nokia Exhibit 2001, p.51
Amazon.com v. Nokia
IPR2024-01175

36[b]: machine-readable program code for performing a transform coding operation on a block of pixel values to produce a corresponding block of transform coefficient values;

36[c]: machine-readable program code for scanning the block of transform coefficient values in a given scanning order to produce a scanned array of coefficient values arranged according to the scanning order;

36[d]: machine-readable program code for representing the coefficient values in the scanned array by a plurality of number pairs, said number pairs having a first number and a second number; and

36[e]: machine-readable code for assigning the second numbers to one of a plurality of contexts representative of the second numbers such that the second number of a number pair is assigned to a context at least partly in dependence on the first number of the number pair.

Claim 37[pre]: A method of context-based arithmetic encoding in which an array of data symbols is represented with a code-word, the method comprising:

37[a]: representing the data symbols in said array as number pairs comprising a first number and a second number; and

37[b]: assigning the first number of a first number pair to a context selected from a plurality of contexts representative of the first numbers at least partly in dependence on a first number of a second number pair.

Claim 38[pre]: A method of context-based arithmetic encoding in which an array of data symbols is represented with a code-word, the method comprising:

38[a]: representing the data symbols in said array as number pairs comprising a first number and a second number;

Nokia Exhibit 2001, p.52
Amazon.com v. Nokia
IPR2024-01175

38[b]: assigning the first number of a first number pair to a context selected from a plurality of contexts representative of the second numbers at least partly in dependence on the first number of the number pair.

Claim 39[pre]: A context-based arithmetic encoder operative to represent an array of data symbols with a code-word and comprising:

39[a]: a run-level coder for representing the data symbols in said array as number pairs comprising a first number and a second number;

39[b]: a context assigner for assigning the first number of a first number pair to a context selected from a plurality of contexts representative of the first numbers at least partly in dependence on a first number of a second number pair.

Claim 40[pre]: A context-based arithmetic encoder operative to represent an array of data symbols with a code-word and comprising:

40[a]: a run-level coder for representing the data symbols in said array as number pairs comprising a first number and a second number;

40[b]: a context assigner for assigning the second number of a number pair to a context selected from a plurality of contexts representative of the second numbers at least partly in dependence on the first number of the number pair.

VII. CLAIM CONSTRUCTION OF ’701 PATENT CLAIM TERMS

98. My understanding is that no claim terms of the ’701 Patent have been proposed for construction by Petitioner. It is also my understanding that no claim terms of the ’701 Patent have been construed in any prior legal proceeding.

Nokia Exhibit 2001, p.53
Amazon.com v. Nokia
IPR2024-01175

VIII. THE ASSERTED PRIOR ART

A. Tsai (Ex. 1005)

99. U.S. Patent No. 5,818,877 to Min-Jen Tsai et. al. ("Tsai") relates to "data compression, and in particular, to compression of run/level coded data or other sequential, digital data." Ex. 1005 at 1:13–15. Tsai recognizes that many data compression algorithms use a sequence of steps similar to those described in the video codec overview above: transformation, quantization, run-length coding and entropy coding. *Id*., 1:50–52. Tsai's invention encompasses specific ways to perform lossless encoding of run-level pairs, independent of the original data source or of the data compression procedure that leads to the generation of those run-level pairs. *Id*., 2:40–55.

100. Tsai's run-level encoding scheme is quite unique, for at least the following reasons: a) The runs and levels are not coded separately; instead, they are combined, and the combination is partitioned such that the coding of runs is not distinct from the coding of levels. Tsai itself states "context analyzer 112 partitions the symbol stream into two different types of symbol groups...by breaking up the symbol stream into groups of symbols each of which is slightly different than a run word or a level word." Pet. at p. 13 (citing Tsai, 15:7–12). b) The symbols Tsai uses are not binary, but instead they are quaternary: each communicated symbol is chosen from a set of four possible values: {0, 1, –, +}. Tsai's use of these quaternary symbols

Nokia Exhibit 2001, p.54
Amazon.com v. Nokia
IPR2024-01175

to map runs and levels together reinforces, as discussed for point a) above, that these symbols are not designed to be associated only with a run or only with a level. c) Although Tsai discusses two contexts, the usage of these two contexts is fixed, and depends only on whether the symbol is encoding a run or a level; there is never a *choice* of which context to use for any given symbol, let alone based on any additional information such as a previous level value. Petitioner recognizes this, stating "Tsai discloses a specific mapping of (run, level) pairs to symbols and the use of two contexts: one for run and one for level." Pet. at p. 10 (citing Tsai at 4:6–5:9). As such, Tsai does not disclose assigning a context for a level given a choice from a plurality of available contexts for the level.

B. VCEG-L28 (Ex. 1006)

101. VCEG-L28 is a Video Coding Experts Group ("VCEG") proposal submitted to the ITU-T in January of 2001. As discussed previously, VCEG-L28 was submitted to the H.26L standards effort as a proposed improvement for a VLC coding scheme in a new H.26L video codec.

102. VCEG-L28 proposes a way to employ two different VLC tables in a way that it says somehow maintains the "universal-VLC approach of H.26L." Ex. 1006 at 1. This purported improvement is achieved through "switching between two VLC tables." *Id*. This results in changing symbol mappings for input values based on previous symbols in some cases. However, VCEG-L28 does not utilize, teach or

Nokia Exhibit 2001, p.55
Amazon.com v. Nokia
IPR2024-01175

disclose arithmetic coding or the assignment of input values or symbols to distinct contexts.

C. VCEG-L13 (Ex. 1007)

103. VCEG-L13 is a VCEG proposal submitted to the ITU-T in January of 2001. VCEG-L13 describes a method to improve entropy coding in H.26L through the implementation of CABAC instead of the "universal variable length code (UVLC) used in the … current H.26L test model." Ex. 1007 at 1. VCEG-L13 provides "a short overview of the main coding elements or [the] proposed entropy coding scheme." *Id*.

104. VCEG-L13 discloses the same scheme that the '701 patent discusses as prior art in the patent's "CABAC Method as Used in H.26L" section. *See* Ex. 1001, 14:5–15:61. Coding of (RUN, LEVEL) pairs is conditioned on the scanning mode, the DC/AC block type, the luminance/chrominance, and the intra/inter macroblock decision. Ex. 1007, § 3.3, p. 4. The arithmetic coding for level values uses four different models, where one model is used for coding of the sign information and the remaining 3 models are used for the first, the second and all remaining bins of the binarization of LEVEL. *Id*., § 3.3.1, p. 5. There is no dependence of context on previous level values.

Nokia Exhibit 2001, p.56
Amazon.com v. Nokia
IPR2024-01175

D. Savatier (Ex. 1008)

105. Savatier discloses "[a] variable length encoding/decoding system for use with a layered signal format, and for adaptively providing VLC changes without additional signal overhead, includ[ing] a plurality of VLC encoding tables." Ex. 1008 at Abstract. A plurality of VLC encoding tables are provided with selected VLC codes associated with anticipated signal statistics. *Id*. at 2:6–9.

106. Savatier further discloses, "The layered signal includes blocks of data subject to common parameters which parameters may be associated with signal statistics. The blocks of data include information indicative of such parameters. Each packet is examined for the common parameter and responsive thereto a default VLC coding/decoding table is assigned." *Id*. at 2:9–15. These parameters inform the coder to determine which VLC coding/decoding table is assigned to a given block.

107. Savatier also discloses an additional embodiment, wherein "the VLC table used for each successive codeword is made dependent upon the table utilized for the lastmost codeword and the relative value of the lastmost codeword and the relative value of the lastmost codeword." *Id*. at 2:15–19.

108. Savatier discloses a similar concept to that of VCEG-L28: choosing a VLC table based on previous symbol values or relative values. Similar to VCEG-L28, Savatier does not utilize, teach or disclose arithmetic coding or the assignment

Nokia Exhibit 2001, p.57
Amazon.com v. Nokia
IPR2024-01175

of input values or symbols to distinct contexts.

E. Tsai in Combination with VCEG-L28 (Ground 1A) Does Not Teach or Disclose the Challenged Claims

109. Neither Tsai nor VCEG-L28 nor the combination thereof teaches or discloses the challenged claims.

110. Claim element 1[b] recites "assigning the first numbers to one of a plurality of contexts representative of the first numbers such that the first number of a first number pair is assigned to a context at least partly in dependence on a first number of a second pair." Therefore, claim element 1[b] requires a "plurality of contexts representative of the first numbers," and as the '701 Patent explains, a "context can be thought of a grouping together bins, which have similar statistical characteristics." Ex. 1001 at 12:67–13:2. It is clear from this definition that a context applies to arithmetic coding and does not apply to VLC.

111. Furthermore, the '701 Patent teaches that, regarding a run/level pair, the "first number" is the level value. *Id*. at 6:12–14. Therefore, Claim 1[b] requires that there be a plurality of contexts representative of the level value for a run/level pair. However, for any given bin in Tsai, the choice of context is fixed. As discussed in Section VIII.A above, although Tsai discusses two contexts, the usage of these two contexts is fixed, and depends only on whether the symbol is encoding a run or a level; there is never a choice of which context to use for any given symbol, let

Nokia Exhibit 2001, p.58
Amazon.com v. Nokia
IPR2024-01175

alone based on any additional information such as a previous level value. Petitioner recognizes this, stating "Tsai discloses a specific mapping of (run, level) pairs to symbols and the use of two contexts: one for run and one for level." Pet. at p. 10 (citing Tsai at 4:6–5:9). As such, Tsai does not disclose assigning a context for a level given a choice from a plurality of available contexts for the level. Petitioner points to Tsai's context analyzer and arithmetic coder to satisfy the assignment of contexts in the first half of limitation 1[b], but as discussed above, Tsai does not teach or disclose the claimed assignment of contexts.

112. Petitioner further points to a combination of Tsai's "symbol mapper" and VCEG-L28's *previous_level* criteria to disclose limitation 1[b]. Pet. at 29 (stating, "A POSITA implementing Tsai's symbol mapper would have been motivated to incorporate VCEG-L28's teaching of switching based on the previous level. In the combination, Tsai's context assignment (i.e., selection between Symbol Map 1 or Symbol Map 2) is based on the ***previous_level*** criteria taught in VCEG-L28."). However, such a combination does not make sense. Symbol Map 1 and Symbol Map 2 have well-defined meanings and Tsai and there is no opportunity to switch between them in a way other than that defined by Tsai without breaking Tsai's invention, let alone by using some other level value. Put another way, Tsai's symbol mapper does not select between two symbol maps based on context or on any related statistical information, let alone on VLC mapping tables such as in

Nokia Exhibit 2001, p.59
Amazon.com v. Nokia
IPR2024-01175

VCEG-L28. Third, in Tsai's exemplary coder, the symbol mapper operates prior to and separate from the context analyzer and context assignment.

113. For the reasons described above, (a) Petitioner fails to provide a feasible combination of Tsai and VCEG-L28, and (b) any hypothesized combination of Tsai and VCEG-L28 inventions or concepts still fails to show that the entirety of claim element 1[b] is rendered obvious.

114. As discussed above with regards to Claim 1[b], Tsai does not disclose "a plurality of contexts." Claim 10[b] provides a similar limitation to claim 1[b]: "assigning the second numbers to one of a plurality of contexts representative of the second numbers such that the second number of a number pair is assigned to a context at least partly in dependence on the first number of the number pair." Ex. 1001 at Claim 10.

115. The main distinction between Claims 1 and 10 is that Claim 10[b] is directed to the assignment of the second number (e.g., a run value) based on the assignment of the first number (e.g., a level value) as opposed to the assignment of the first number based on a first number of a second pair. As such, the arguments for claim element 1[b] apply to 10[b], and Ground 1A does not teach or disclose claim 10 for the same reasons.

116. Claims 18, 35, 37, and 39 all contain the same limitation of "assigning

Nokia Exhibit 2001, p.60
Amazon.com v. Nokia
IPR2024-01175

the first numbers to one of a plurality of contexts representative of the first numbers such that the first number of a first number pair is assigned to a context at least partly in dependence on a first number of a second number pair" found in Claim 1. Petitioner's additional arguments regarding these limitations do not remedy the shortcomings of the arguments made regarding Claim 1. Indeed, Petitioner repeatedly incorporates its Claim 1 arguments into the additional claim sections. Because Petitioner failed to show that this limitation was met for claim 1, Petitioner's arguments regarding the additional claims fail as well.

117. Claims 27, 36, 38, and 40 all contain the same limitation of assigning the second numbers to one of a plurality of contexts representative of the second numbers, and operative to assign the second number of a number pair to a context at least partly in dependence on the first number of the number pair, as found in Claim 10. Petitioner's additional arguments regarding these limitations do not remedy the shortcomings of the arguments made regarding Claim 10. Indeed, Petitioner repeatedly incorporates its Claim 10 arguments into the additional claim sections. Because Petitioner failed to show that this limitation was met for claim 10, its arguments regarding the additional claims fail as well.

F. A POSITA would not be motivated to combine and/or modify Tsai and VCEG-L28

118. Tsai discloses a system related to arithmetic coding, while VCEG-L28

Nokia Exhibit 2001, p.61
Amazon.com v. Nokia
IPR2024-01175

discloses a system related to VLC. These are two very different ways of doing lossless compression, and they are distinct coding procedures in video codecs: only one or the other is chosen to code any given entity.[10] Furthermore, the arithmetic coding of TSAI's invention is unusually complicated relative to that typically used in video coding for two reasons: (a) Tsai's binarization scheme of mapping runs and levels to bins is over-complicated; it appears to be designed mainly to support the second part of Tsai's invention, the word-to-bit-plane reordering of symbols. *See* Ex. 1005, 16:34–17:61, Figs. 13, 14A, 14B; (b) Tsai's use of quaternary symbols is unusual and over-complicated for video coding… it does not even quality as binary arithmetic coding, or as CABAC.

119. Even if a POSITA were to try to combine Tsai's arithmetic coding and VCEG-L28, there is no way to do such a thing without over-complicating Tsai even further and arriving at a less efficient system, since there is no part of Tsai's arithmetic coding that is based on using one VLC mapping table versus another. If a POSITA were to *attempt* to combine Tsai with VCEG-L28's *previous_level* criteria, a POSITA would need to somehow combine Tsai's bin mapping and

[10] To the extent a binarization performed as an initial step of arithmetic coding may appear similar to a VLC coding operation, as described earlier, this is not considered to be the VLC coding used in video compression algorithms.

Nokia Exhibit 2001, p.62
Amazon.com v. Nokia
IPR2024-01175

contexts with an additional choice of VLC tables. As discussed earlier in this declaration, VLC and arithmetic coding are distinct and performed very differently, so a POSITA would not be motivated to make such a modification.

120. For all the reasons above, a POSITA would not have been motivated to consider Tsai at all, or to combine Tsai and VCEG-L28.

G. Applicant Admitted Prior Art (AAPA) in Combination with Savatier (Ground 2) Does Not Teach or Disclose Challenged Claims 1-9, 18-26, 35, 37, 39

121. In my opinion, Petitioner has failed to show that Claims 1, 18, 35, 37 and 39 are obvious in light of AAPA combined with Savatier. It also follows that Petitioner has failed to show that dependent claims 2–9 and 19–26 are obvious.

122. Petitioner alleges that Savatier overcomes the deficiency of AAPA to teach element 1[b], but Savatier does not. Savatier discloses a system for VLC coding, while the purported AAPA relates to CABAC. As already discussed, VLC coding and Arithmetic coding are distinct coding methods, and the processing steps used within each of them respectively are not interchangeable or combinable.

123. The VLC method of symbol mapping described in Savatier comprises two different tables for mapping inputs to codewords, and the choosing of one VLC table versus the other. However, these VLC table mappings do not relate to the

Nokia Exhibit 2001, p.63
Amazon.com v. Nokia
IPR2024-01175

probability of a symbol occurring in a context as in CABAC.

124. Petitioner states "Savatier teaches selecting a VLC coding/decoding table of codes having similar anticipated probabilities (i.e., a 'context') based upon 'the table utilized for the lastmost codeword and the relative value of the lastmost codeword.'" and "Savatier's table selection is context assignment." Pet. at 69–70. These are incorrect statements, because they attempt to broaden "context" beyond its definition in the '701 patent.

125. As discussed in Section V.H above, the '701 Patent describes "In the CABAC method each of the bins is assigned to a so-called 'context' (hence the name context-based arithmetic coding). A context can be thought of as grouping together bins, which have similar statistical characteristics. In other words, each bin assigned to a particular context is assumed to have a similar probability of containing the value 1 or 0 as the other bins belonging to that context." Ex. 1001, 12:42–13:10. As such, a context is related to statistical characteristics of bins used in the coding process. VLC uses table mappings, not groups of bins or statistical characteristics thereof.

126. As also discussed in Section V.H above, CABAC is shown to be much different and more complex than Variable Length Coding. VLC defines a straightforward translation from inputs to codewords and back, whereas CABAC

Nokia Exhibit 2001, p.64
Amazon.com v. Nokia
IPR2024-01175

involves complex arithmetic and a complex and adaptive arrangement of bins, symbols and contexts. Concepts from VLC cannot simply be carried over to CABAC with the expectation of predicable results.

127. Therefore, even if a POSITA would have been motivated to combine CABAC and Savatier as Petitioner suggests, the resulting system would not assign one of a plurality of contexts to a first number based on the first number of a second pair. At best, it might guide the bin mapping that would be used to represent an input value prior to the actual context assignment and arithmetic coding of binary values. This is because the VLC tables do not assign a context, but rather provide a mapping for a symbol to be coded.

128. Independent claims 18, 35, 37 and 39 rely on similar Petitioner arguments, so are also not taught by Ground 2 for the reasons discussed above. It also follows that Petitioner has failed to show that dependent claims 2–9 and 19–26 are obvious.

H. AAPA in Combination with VCEG-L13 (Ground 3) Does Not Teach or Disclose Challenged Claims 10-17, 27-34, 36, 38, 40

129. In my opinion, Petitioner has failed to show that Claims 10, 27, 36, 38 and 40 are obvious in light of AAPA and Savatier. It also follows that Petitioner has failed to show that dependent claims 11–17 and 28–34 are obvious.

Nokia Exhibit 2001, p.65
Amazon.com v. Nokia
IPR2024-01175

130. Petitioner's argument regarding a combination of AAPA with VCEG-L13 is flawed, because Petitioner's interpretation is distorted beyond the scope of the claim. Based on the inherent design of run-level coding, level values of zero are not coded… they simply add to the run count until the next non-zero level value does need to be encoded (along with its corresponding run length of preceding zero level values). However, some indication needs to be included in the encoded bitstream of which run/level pair is the last in the coefficient scan. Video coding algorithms may insert a level = 0 value to indicate an End of Block. Note that in this case, it is not a real level value that is being encoded, but rather just a syntax marker in the encoded bitstream.

131. Petitioner's argument depends on VCEG-L13's end of block symbol, which is just the level = 0 syntax marker described above. In VCEG-L13, the end of block is represented with a level value of 0. EX. 1007 at 5. When the end of block symbol is received, there is no need to code a run value; this is why VCEG-L13 only codes the run value if LEVEL ≠ 0. *Id.* at 5. However, coding a run value and assigning a context to a run value are different concepts. A method cannot assign a context to a field that doesn't exist and has no coding at all, yet that is what Petitioner is alleging by stating that a level = 0 value somehow affects the assignment of a context to a following run field that doesn't even exist.

132. Coding run/level pairs based on the already assigned contexts is distinct

Nokia Exhibit 2001, p.66
Amazon.com v. Nokia
IPR2024-01175

from the process of assigning the contexts. Under Petitioner's interpretation of the Challenged Claims, the lack of a value altogether is equal to making a determination for context assignment. This is improper, and for this reason, Ground 3 does not teach or disclose claim element 10[b].

133. Independent claims 27, 36, 38 and 40 rely on similar Petitioner arguments, so are also not taught by Ground 3 for the reasons discussed above. It also follows that Petitioner has failed to show that dependent claims 11–17 and 28–34 are obvious.

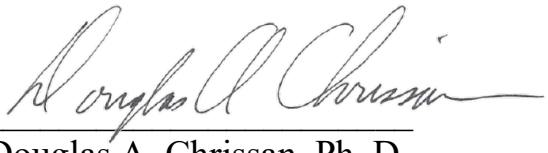
IX. OPINIONS

134. For all the reasons described above, it is my opinion that Petitioner's Grounds 1A, 2 and 3 do not render obvious the Challenged Claims 1–40 of the '701 Patent.

135. The reasons described above do not necessarily represent the entirety of my opinions regarding Petitioner's Grounds. I reserve the right to provide further analysis in the event this IPR is instituted. I declare under penalty of perjury that the foregoing is true and correct. I acknowledge that willful false statements and the like are punishable by fine and/or imprisonment, that all statements made of my own knowledge are true, and that all statements made on information and belief are believed to be true.

Nokia Exhibit 2001, p.67
Amazon.com v. Nokia
IPR2024-01175

Executed January 21, 2025, in Los Gatos, CA.

By: 
Douglas A. Chrissan, Ph. D.

Nokia Exhibit 2001, p.68
Amazon.com v. Nokia
IPR2024-01175