



BUNDESPATENTGERICHT

IM NAMEN DES VOLKES

URTEIL

Verkündet am
17. Februar 2023

2 Ni 3/21 (EP)
(Aktenzeichen)

...

In der Patentnichtigkeitssache

...

betreffend das europäische Patent 2 842 318
(DE 60 2013 015 861)

hat der 2. Senat (Nichtigkeitssenat) des Bundespatentgerichts aufgrund der mündlichen Verhandlung vom 17. Februar 2023 durch die Vorsitzende Richterin Hartlieb sowie die Richter Dipl.-Phys. Univ. Dr. Forkel, Dipl.-Ing. Univ. Hoffmann, Dr. Himmelmann und Dipl.-Phys. Univ. Dr. Städele für Recht erkannt:

- I. Das europäische Patent EP 2 842 318 wird mit Wirkung für das Hoheitsgebiet der Bundesrepublik Deutschland im Umfang seiner Ansprüche 1 und 15 für nichtig erklärt.
- II. Die Kosten des Rechtsstreits trägt die Beklagte.
- III. Das Urteil ist gegen Sicherheitsleistung in Höhe von 120 % des zu vollstreckenden Betrages vorläufig vollstreckbar.

Tatbestand

Die Beklagte ist Inhaberin des auch mit Wirkung für die Bundesrepublik Deutschland erteilten europäischen Patents EP 2 842 318 (Streitpatent), das am 28. Dezember 2016 in englischer Sprache veröffentlicht wurde. Das Streitpatent geht zurück auf eine PCT-Anmeldung mit der Veröffentlichungsnummer WO 2013/153226 A1 (in englischer Sprache), die unter dem EP-Aktenzeichen 13719279.5 in die europäische regionale Phase eingetreten ist. Sie nimmt die Prioritäten der US-amerikanischen Patentanmeldungen US 2012 61/624,098 vom 13. April 2012 und US 2012 61/666,185 vom 29. Juni 2012 in Anspruch. Das Streitpatent hat den Anmeldetag 15. April 2013 und die Bezeichnung „LOW DELAY PICTURE CODING = BILDCODIERUNG MIT GERINGER VERZÖGERUNG“. Es ist mit der EP 2 842 318 B1 veröffentlicht worden und wird beim Deutschen Patent- und Markenamt unter dem Aktenzeichen 60 2013 015 861.4 geführt.

Das Streitpatent betrifft das Codieren und Decodieren digitaler Bilder. Es umfasst 19 Patentansprüche, darunter den auf einen „Decodierer zum Rekonstruieren eines Bildes aus einem Datenstrom“ gerichteten Patentanspruch 1 und den nebengeordneten, auf „Ein Verfahren [zum] Rekonstruieren eines Bildes aus einem Datenstrom“ gerichteten Patentanspruch 15. Die nebengeordneten Patentansprüche 11 bzw. 16 sind auf einen „Codierer zur Codierung eines Bildes in einen Datenstrom“ bzw. auf „Ein Verfahren zur Codierung eines Bildes in einen Datenstrom“ gerichtet. Die nebengeordneten Patentansprüche 17 und 19 betreffen jeweils einen „Datenstrom, der unter Verwendung des Verfahrens gemäß Anspruch 16 codiert wird“ sowie „Ein Computerprogramm mit einem Programmcode zum Durchführen des Verfahrens gemäß einem der Ansprüche 15 oder 16“. In den Patentansprüchen 2 bis 10, 12 bis 14 und 18 enthält das Streitpatent Unteransprüche.

Die Klägerin begehrt die Nichtigkeitsklärung des deutschen Teils des Streitpatents im Umfang der Patentansprüche 1 und 15. Die Beklagte verteidigt das

Streitpatent in vollem Umfang und hilfsweise beschränkt mit 5 Hilfsanträgen.

Die Klägerin stützt ihre Klage auf den Nichtigkeitsgrund der mangelnden Patentfähigkeit mit Blick auf fehlende Neuheit und fehlende erfinderische Tätigkeit.

Die hiesige Klägerin und die Klägerin eines wegen Klagerücknahme wieder abgetrennten Nichtigkeitsverfahrens gegen das Streitpatent haben zum Teil eine gleichlautende Nummerierung ihrer unterschiedlichen Anlagen verwendet. Zur Unterscheidung bezeichnet der Senat die Anlagen der Klägerin mit dem Präfix 1 und die aus dem abgetrennten Verfahren 2 Ni 17/21 (EP) mit dem Präfix 2.

Zur Stützung ihres Vorbringens hat die Klägerin die folgenden Dokumente genannt:

- 1NK1** Streitpatentschrift EP 2 842 318 B1, Anmeldetag 15. April 2013, Tag der Veröffentlichung 28. Dezember 2016;
- 1NK2** Klageschrift vom 17. Juli 2020 im Verletzungsverfahren vor dem Landgericht Düsseldorf, Az. 4c O 42/20;
- 1NK3** Registerauszug des Deutschen Patent- und Markenamtes vom 18. Januar 2021 zum Streitpatent;
- 1NK4** Merkmalsgliederung der Klägerin des Patentanspruchs 1 des Streitpatents;
- 1NK5** Merkmalsgliederung der Klägerin des Patentanspruchs 15 des Streitpatents;
- 1NK6** US-amerikanische Patentanmeldung US 2012 61/624,098 P, Anmeldetag 13. April 2012, vorgelegt als beglaubigte Kopie eines Prioritätsdokuments zur internationalen Patentanmeldung PCT/EP2013/057798, Anmeldetag 15. April 2013;
- 1NK7** US-amerikanische Patentanmeldung US 2012 61/666,185 P, Anmeldetag 29. Juni 2012, vorgelegt als

beglaubigte Kopie eines Prioritätsdokuments zur internationalen Patentanmeldung PCT/EP2013/057798, Anmeldetag 15. April 2013;

1NK8

Auszug aus 2NK4;

1NK9

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Wavefront Parallel Processing for HEVC Encoding and Decoding“, JCTVC-F274, 6th Meeting, Torino, IT, 14-22 July, 2011, veröffentlicht am 16. Juli 2011;

1NK10

Auszug aus 2NK6;

1NK11

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Low latency CABAC initialization for dependent tiles“, JCTVC-G0197, 7th Meeting, Geneva, CH, 21-30 November, 2011, veröffentlicht am 21. November 2011;

1NK12

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Generalized slices“, JCTVC-D378, 4th Meeting, Daegu, KR, 20-28 January, 2011, veröffentlicht am 20. Januar 2011;

1NK13

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „High efficiency video coding (HEVC) text specification draft 6“, JCTVC-H1003, 8th Meeting, San José, CA, USA, 1–10 February, 2012, veröffentlicht am 2. April 2012 (Auszüge);

In dem Verfahren befindet sich noch die im abgetrennten Verfahren klägerseits eingereichte

2NK4

Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC-1/SC-29/WG-11, „High Efficiency Video Coding (HEVC) text specification draft 9“,

JCTVC-K1003_v13, 11th Meeting, Shanghai, CN, 10–19
October 2012, veröffentlicht am 18. Dezember 2012;

Die Klägerin behauptet, die Gegenstände der Patentansprüche der Hilfsanträge I bis V seien nicht patentfähig.

Die Klägerin stellt den Antrag,

das europäische Patent EP 2 842 318 mit Wirkung für das Hoheitsgebiet der Bundesrepublik Deutschland im Umfang der Ansprüche 1 und 15 für nichtig zu erklären.

Die Beklagte stellt den Antrag,

die Klage abzuweisen,
hilfsweise

das europäische Patent EP 2 842 318 unter Klageabweisung im Übrigen dadurch teilweise für nichtig zu erklären, dass seine Patentansprüche die Fassung eines der Hilfsanträge I bis III vom 1. August 2022 sowie IV und V vom 21. Oktober 2022 in dieser Reihenfolge, erhalten.

Die Beklagte tritt der Argumentation der Klägerin in allen wesentlichen Punkten entgegen. Das Streitpatent sei jedenfalls in der Fassung eines der Hilfsanträge patentfähig.

Zur Stützung ihres Vorbringens hat die Beklagte die folgenden Dokumente genannt:

- ES1a** Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Dependent Slices“, JCTVC-10229, 9th Meeting, Geneva, CH, 27 April-7 May 2012;
- ES1b** Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Dependent Slices support in HEVC main profile“, JCTVC- J0264, 10th

- Meeting, Stockholm, SE, 11-20 July 2012;
- ES1c** Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, „Meeting report of the 11th meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Shanghai, CN, 10–19 Oct 2012“, JCTVC-K1000, 11th Meeting, Shanghai, CN, 10–19 Oct. 2012;
- ES1d** Hilfsanträge I-III vom 1. August 2022 (Reinschrift sowie mit hervorgehobenen Änderungen);
- ES2** Auszug aus: Sze V. et al. (Editors), „High Efficiency Video Coding (HEVC) – Algorithms and Architectures“, Springer, 2014.

Die Beklagte hat in der mündlichen Verhandlung am 17. Februar 2023 erklärt, dass sie die Patentansprüche gemäß Hauptantrag und Hilfsanträgen als jeweils geschlossene Anspruchssätze ansehe, die jeweils insgesamt beansprucht würden.

Sie hat in der mündlichen Verhandlung am 17. Februar 2023 außerdem erklärt, dass die Hilfsanträge sich ausschließlich auf die angegriffenen Patentansprüche beziehen würden.

Die Merkmale des erteilten Patentanspruchs 1 des Streitpatents können in Anlehnung an die von der Klägerin vorgeschlagene Merkmalsgliederung (Anlage **1NK4**) wie folgt gegliedert werden:

Patentanspruch 1:

	Anspruch 1	Übersetzung
M1.1	Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is	Ein Decodierer zum Rekonstruieren eines Bildes (10) aus einem Datenstrom (12), in den das Bild in Einheiten von Stücken (14) codiert ist, in die das Bild (10)

	partitioned,	partitioniert ist,
M1.2	wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and	wobei der Decodierer dazu ausgebildet ist, die Stücke (14) gemäß einer Stückreihenfolge (16) aus dem Datenstrom (12) zu decodieren, und
M1.3	the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and	der Decodierer auf einen Syntaxelementabschnitt (18) innerhalb eines aktuellen Stückes der Stücke anspricht, um das aktuelle Stück gemäß einem von zumindest zwei Modi (20, 22) zu decodieren, und
M1.3.1a	in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and	das aktuelle Stück gemäß einem ersten (20) der zumindest zwei Modi unter Verwendung von kontextadaptiver Entropiedecodierung (24) einschließlich einer Ableitung von Kontexten über Stückgrenzen hinweg, einer fortlaufenden Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte und einer Initialisierung (38, 40) der Symbolwahrscheinlichkeiten abhängig von gespeicherten Zuständen von Symbolwahrscheinlichkeiten eines zuvor decodierten Stückes,
M1.3.1b	predictive decoding across the slice boundaries, and	sowie voraussagender Decodierung über die Stückgrenzen hinweg aus dem

		Datenstrom (12) zu decodieren, und
M1.3.2a	in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and	das aktuelle Stück gemäß einem zweiten (22) der zumindest zwei Modi unter Verwendung von kontextadaptiver Entropiedecodierung mit einem Beschränken der Ableitung der Kontexte, um die Stückgrenzen nicht zu überschreiten, einer fortlaufenden Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte und einer Initialisierung der Symbolwahrscheinlichkeiten unabhängig von jeglichem zuvor decodierten Stück,
M1.3.2b	predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,	sowie voraussagender Decodierung mit einem Beschränken der voraussagenden Decodierung, um die Stückgrenzen nicht zu überschreiten, aus dem Datenstrom (12) zu decodieren und
M1.4	wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and	wobei das Bild (10) in Codierblöcke (32) partitioniert ist, die in Zeilen und Spalten angeordnet sind und eine untereinander definierte Rasterabtastreihenfolge (36) aufweisen, und
M1.5	the decoder is configured to associate each slice (14) with a	der Decodierer dazu ausgebildet ist, jedes Stück (14) einem

	continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and	fortlaufenden Teilsatz der Codierblöcke (32) in der Rasterabtastreihenfolge (36) zuzuordnen, so dass die Teilsätze einander entlang der Rasterabtastreihenfolge (36) gemäß der Stückreihenfolge folgen, und
M1.6	wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and,	wobei der Decodierer dazu ausgebildet ist, Symbolwahrscheinlichkeiten, wie bei einer kontextadaptiven Entropie-decodierung des zuvor decodierten Stückes bis zu einem zweiten Codierblock (32) in einer Zeile gemäß der Rasterabtastreihenfolge (36) erhalten, zu speichern, und
M1.6.1	in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and,	bei einem Initialisieren der Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes gemäß dem ersten Modus zu überprüfen, ob ein erster Codierblock des fortlaufenden Teilsatzes von Codierblöcken (32), der dem aktuellen Stück zugeordnet ist, ein erster Codier-block (32) in einer Zeile gemäß der Rasterabtastreihenfolge ist, und,
M1.6.2	if so, initialize (40) the symbol probabilities for the context	falls dies zutrifft, die Symbolwahrscheinlichkeiten für

	<p>adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and,</p>	<p>die kontextadaptive Entropiedecodierung des aktuellen Stückes abhängig von den gespeicherten Symbolwahrscheinlichkeiten, wie bei der kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu einem zweiten Codierblock in einer Zeile gemäß der Rasterabtastreihenfolge (36) erhalten, zu initialisieren (40), und,</p>
<p>M1.6.3</p>	<p>if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice.</p>	<p>falls dies nicht zutrifft, die Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes abhängig von Symbolwahrscheinlichkeiten, wie bei der kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu dem Ende des zuvor decodierten Stückes erhalten, zu initialisieren (38).</p>

Patentanspruch 15 lautet in Anlehnung an die von der Klägerin vorgeschlagene Merkmalsgliederung (Anlage **1NK5**) (englisch: gemäß Streitpatentschrift, deutsch: teilweise verbesserte Übersetzung seitens des Senats):

Patentanspruch 15:

	Anspruch 15	Übersetzung
M15.1	Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned,	Ein Verfahren [zum] Rekonstruieren eines Bildes (10) aus einem Datenstrom (12), in den das Bild in Einheiten von Stücken (14) codiert ist, in die das Bild (10) partitioniert ist,
M15.2	wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and	wobei das Verfahren ein Decodieren der Stücke (14) aus dem Datenstrom (12) gemäß einer Stückreihenfolge (16) aufweist, und
M15.3	the method is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22),	das Verfahren auf einen Syntaxelementabschnitt (18) innerhalb eines aktuellen Stückes der Stücke anspricht, um das aktuelle Stück gemäß einem von zumindest zwei Modi (20, 22) zu decodieren,
M15.3.1a	wherein in accordance with a first (20) of the at least two modes), the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol	wobei das aktuelle Stück gemäß einem ersten (20) der zumindest zwei Modi unter Verwendung von kontextadaptiver Entropiedecodierung (24) einschließlich einer Ableitung von Kontexten über Stückgrenzen hinweg, einer fortlaufenden Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte und einer Initialisierung (38, 40) der

	probabilities of a previously decoded slice, and	Symbolwahrscheinlichkeiten abhängig von gespeicherten Zuständen von Symbolwahrscheinlichkeiten eines decodierten Stückes,
M15.3.1b	predictive decoding across the slice boundaries, and	sowie voraussagender Decodierung über die Stückgrenzen hinweg aus dem Datenstrom (12) decodiert wird, und
M15.3.2a	in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and	das aktuelle Stück gemäß einem zweiten (22) der zumindest zwei Modi unter Verwendung von kontextadaptiver Entropiedecodierung mit einem Beschränken der Ableitung der Kontexte, um die Stückgrenzen nicht zu überschreiten, einer fortlaufenden Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte und einer Initialisierung der Symbolwahrscheinlichkeiten unabhängig von jeglichem zuvor decodierten Stück,
M15.3.2b	predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,	sowie voraussagender Decodierung mit einem Beschränken der voraussagenden Decodierung, um die Stückgrenzen nicht zu überschreiten, aus dem Datenstrom (12) decodiert wird,
M15.4	wherein the picture (10) is partitioned in coding blocks (32)	wobei das Bild (10) in Codierblöcke (32) partitioniert ist,

	arranged in rows and columns and having a raster scan order (36) defined among each other, and	die in Zeilen und Spalten angeordnet sind und eine untereinander definierte Rasterabtastreihenfolge (36) aufweisen, und
M15.5	the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and	das Verfahren ein Zuordnen jedes Stückes (14) zu einem fortlaufenden Teilsatz der Codierblöcke (32) in der Rasterabtastreihenfolge (36) aufweist, so dass die Teilsätze einander entlang der Rasterabtastreihenfolge (36) gemäß der Stückreihenfolge folgen, und
M15.6	wherein the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and,	wobei das Verfahren ein Speichern von Symbolwahrscheinlichkeiten, wie bei einer kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu einem zweiten Codierblock (32) in einer Zeile gemäß der Rasterabtastreihenfolge (36) erhalten, und
M15.6.1	in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the	bei einem Initialisieren der Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes gemäß dem ersten Modus ein Überprüfen, ob ein erster Codierblock des fortlaufenden Teilsatzes von

	current slice is a first coding block (32) in a row in accordance with the raster scan order, and,	Codierblöcken (32), der dem aktuellen Stück zugeordnet ist, ein erster Codierblock (32) in einer Zeile gemäß der Rasterabtastreihenfolge ist, und,
M15.6.2	if so, initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and,	falls dies zutrifft, ein Initialisieren (40) der Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes abhängig von den gespeicherten Symbolwahrscheinlichkeiten, wie bei einer kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu einem zweiten Codierblock in einer Zeile gemäß der Rasterabtastreihenfolge (36) erhalten, und
M15.6.3	if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice.	falls dies nicht zutrifft, ein Initialisieren (38) der Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes abhängig von Symbolwahrscheinlichkeiten, wie bei einer kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu dem Ende des zuvor decodierten Stückes erhalten, aufweist.

Hilfsantrag I vom 1. August 2022 lautet:

Hilfsantrag I

Neue Ansprüche (Reinschrift)

Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and
in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the decoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and
wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy

decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice,

wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice decoded in the second mode.

2. Decoder according to claim 1, wherein the decoder is configured to be responsive to the syntax element portion (18) within the current slice of the slices (14), so as to decode the current slice in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the decoder is configured to
in accordance with the third mode (42), decode the current slice from the datastream using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding across the slice boundaries, wherein the one of the first and third modes is selected depending on a syntax element.
3. Decoder according to claim 1 or 2, wherein the decoder is configured to be responsive to a generic syntax element in the datastream so as to operate in one of at least two generic operating modes, with, according to a first generic operating mode, performing the responsiveness to the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.
4. Decoder according to any of claims 1 to 3, wherein the decoder is configured to according to the first and second modes, inevitably und uninterruptedly continue continuously updating the symbol probabilities from a beginning to an end of the current slice.
5. Decoder according to any of claims 1 to 4, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to an end of the previously decoded slice.

6. Decoder according to any of claims 1 to 5, wherein the decoder is configured to, in the first and second mode, restrict the predictive decoding within tiles into which the picture is sub-divided.
7. Decoder according to any of 1 to 6, where the decoder is configured to, in the first and second mode, read information from current slice revealing a subdivision of the current slice into parallel subsections, cease the context adaptive entropy decoding at the end of the first parallel subsection and resume the context adaptive entropy decoding anew at a beginning of any subsequent parallel subsection including, in the first mode, an initialization of the symbol probabilities depending on saved states of symbol probabilities of the preceding parallel subsection and, in the second mode, an initialization of the symbol probabilities independent from any previously decoded slice and any previously decoded parallel subsection.
8. Decoder according to any of the preceding claims, wherein the decoder is configured to reconstruct the picture (10) from the datastream (12) using WPP processing, wherein each slice (14) comprises a start syntax portion (400) indicating a position of a decoding begin of the respective slice within the picture (10) and wherein the decoder is configured to identify entry points of WPP substreams into which the slices are grouped, by identifying, using the slices' start syntax portions, slices starting at a left hand side of the picture, and parallel decoding the WPP substreams in a staggered manner with sequentially commencing the decoding of the WPP substreams in accordance with the slice order.
9. Decoder according to any of 1 to 8, wherein the decoder is configured to split chroma transform blocks differently than luma transform blocks in response to information in the data stream.
10. Encoder for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the encoder is configured to encode the slices (14) into the datastream (12) in accordance with a slice order (16) and the encoder is configured to determine a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol

probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and

if the current slice is to be coded in accordance with a second (22) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the encoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

wherein the encoder is configured to save symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice,

wherein the encoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice encoded in the second mode.

11. Encoder according to claim 10, wherein the encoder is configured to split chroma transform blocks differently than luma transform blocks.

12. Encoder according to claim 9, wherein the encoder is configured code the syntax element portion (18) into the current slice of the slices (14) so that the current slice is signaled to be coded thereinto in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the encoder is configured to
in accordance with the third mode (42), encode the current slice into the datastream using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive coding across the slice boundaries, wherein the encoder distinguishes between the one of the first and third modes using a syntax element.
13. Encoder according to claims 9 or 10, wherein the encoder is configured to determine a generic syntax element and write same into the datastream with operating in one of at least two generic operating modes depending on the generic syntax element, namely, with, according to a first generic operating mode, performing coding the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.
14. Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and the method is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), wherein
in accordance with a first (20) of the at least two modes), the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a con-

tinuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, and

wherein the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice decoded in the second mode.

15. Method for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises encoding the slices (14) into the datastream (12) in accordance with a slice order (16) and the method comprises
determining a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and

if the current slice is to be coded in accordance with a first (20) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and if the current slice is to be coded in accordance with a second (22) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

wherein the method further comprises saving symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice encoded in the second mode.

16. Datastream encoded using the method according to claim 15.

17. Datastream according to claim 16, wherein chroma transform blocks are split differently than luma transform blocks.
18. Computer program having a program code for performing, when running on a computer, a method according to claim 14 or 15.

Hilfsantrag II vom 1. August 2022 lautet:

Hilfsantrag II

Neue Ansprüche (Reinschrift)

1. Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and
in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the decoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and
wherein the decoder is configured to be responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster

scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the decoder is configured to initialize the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not.

2. Decoder according to claim 1, wherein the decoder is configured to be responsive to the syntax element portion (18) within the current slice of the slices (14), so as to decode the current slice in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the decoder is configured to
in accordance with the third mode (42), decode the current slice from the datastream using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding across the slice boundaries, wherein the one of the first and third modes is selected depending on a syntax element.
3. Decoder according to claim 1 or 2, wherein the decoder is configured to be responsive to a generic syntax element in the datastream so as to operate in one of at least two generic operating modes, with, according to a first generic operating mode, performing the responsiveness to the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.
4. Decoder according to any of claims 1 to 3, wherein the decoder is configured to according to the first and second modes, inevitably und uninterruptedly continue

continuously updating the symbol probabilities from a beginning to an end of the current slice.

5. Decoder according to any of claims 1 to 4, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to an end of the previously decoded slice.
6. Decoder according to any of claims 1 to 5, wherein the decoder is configured to, in the first and second mode, restrict the predictive decoding within tiles into which the picture is sub-divided.
7. Decoder according to any of 1 to 6, where the decoder is configured to, in the first and second mode, read information from current slice revealing a subdivision of the current slice into parallel subsections, cease the context adaptive entropy decoding at the end of the first parallel subsection and resume the context adaptive entropy decoding anew at a beginning of any subsequent parallel subsection including, in the first mode, an initialization of the symbol probabilities depending on saved states of symbol probabilities of the preceding parallel subsection and, in the second mode, an initialization of the symbol probabilities independent from any previously decoded slice and any previously decoded parallel subsection.
8. Decoder according to any of 1 to 7, where the decoder is configured to, in accordance with the first (20) of the at least two modes, copy for the current slice a portion of a slice header syntax from a preceding slice decoded in the second mode.
9. Decoder according to any of the preceding claims, wherein the decoder is configured to reconstruct the picture (10) from the datastream (12) using WPP processing, wherein each slice (14) comprises a start syntax portion (400) indicating a position of a decoding begin of the respective slice within the picture (10) and wherein the decoder is configured to identify entry points of WPP substreams into which the slices are grouped, by identifying, using the slices' start syntax portions, slices starting at a left hand side of the picture, and parallel decoding the WPP substreams in a staggered manner with sequentially commencing the decoding of the WPP substreams in accordance with the slice order.
10. Decoder according to any of 1 to 9, wherein the decoder is configured to split chroma transform blocks differently than luma transform blocks in response to information in the data stream.

11. Encoder for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the encoder is configured to encode the slices (14) into the datastream (12) in accordance with a slice order (16) and the encoder is configured to
determine a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and
if the current slice is to be coded in accordance with a second (22) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the encoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and
wherein the encoder is configured to determine a syntax element for, and code same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, the encoder is configured to save symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy

encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the encoder is configured to initialize the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not.

12. Encoder according to claim 11, wherein the encoder is configured to split chroma transform blocks differently than luma transform blocks.
13. Encoder according to claim 10, wherein the encoder is configured code the syntax element portion (18) into the current slice of the slices (14) so that the current slice is signaled to be coded thereinto in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the encoder is configured to
in accordance with the third mode (42), encode the current slice into the datastream using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive coding across the slice boundaries, wherein the encoder distinguishes between the one of the first and third modes using a syntax element.
14. Encoder according to claims 10 or 11, wherein the encoder is configured to determine a generic syntax element and write same into the datastream with operating in one of at least two generic operating modes depending on the generic syntax element, namely, with, according to a first generic operating mode, performing coding the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.

15. Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and the method is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), wherein
in accordance with a first (20) of the at least two modes), the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, and
wherein the method is responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block

in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the method comprises initializing the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not.

16. Method for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises encoding the slices (14) into the datastream (12) in accordance with a slice order (16) and the method comprises
determining a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and
if the current slice is to be coded in accordance with a first (20) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and
if the current slice is to be coded in accordance with a second (22) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

wherein the method further comprises determining a syntax element for, and coding same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, the method further comprises saving symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the method further comprises initializing the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not.

17. Datastream encoded using the method according to claim 16.
18. Datastream according to claim 17, wherein chroma transform blocks are split differently than luma transform blocks.
19. Computer program having a program code for performing, when running on a computer, a method according to claim 15 or 16.

Hilfsantrag III vom 1. August 2022 lautet:

Hilfsantrag III

Neue Ansprüche (Reinschrift)

1. Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and
in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the decoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and
wherein the decoder is configured to be responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster

scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the decoder is configured to initialize the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not, wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice decoded in the second mode.

2. Decoder according to claim 1, wherein the decoder is configured to be responsive to the syntax element portion (18) within the current slice of the slices (14), so as to decode the current slice in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the decoder is configured to
in accordance with the third mode (42), decode the current slice from the datastream using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding across the slice boundaries, wherein the one of the first and third modes is selected depending on a syntax element.
3. Decoder according to claim 1 or 2, wherein the decoder is configured to be responsive to a generic syntax element in the datastream so as to operate in one of at least two generic operating modes, with, according to a first generic operating mode, performing the responsiveness to the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.

4. Decoder according to any of claims 1 to 3, wherein the decoder is configured to according to the first and second modes, inevitably und uninterruptedly continue continuously updating the symbol probabilities from a beginning to an end of the current slice.
5. Decoder according to any of claims 1 to 4, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to an end of the previously decoded slice.
6. Decoder according to any of claims 1 to 5, wherein the decoder is configured to, in the first and second mode, restrict the predictive decoding within tiles into which the picture is sub-divided.
7. Decoder according to any of 1 to 6, where the decoder is configured to, in the first and second mode, read information from current slice revealing a subdivision of the current slice into parallel subsections, cease the context adaptive entropy decoding at the end of the first parallel subsection and resume the context adaptive entropy decoding anew at a beginning of any subsequent parallel subsection including, in the first mode, an initialization of the symbol probabilities depending on saved states of symbol probabilities of the preceding parallel subsection and, in the second mode, an initialization of the symbol probabilities independent from any previously decoded slice and any previously decoded parallel subsection.
8. Decoder according to any of the preceding claims, wherein the decoder is configured to reconstruct the picture (10) from the datastream (12) using WPP processing, wherein each slice (14) comprises a start syntax portion (400) indicating a position of a decoding begin of the respective slice within the picture (10) and wherein the decoder is configured to identify entry points of WPP substreams into which the slices are grouped, by identifying, using the slices' start syntax portions, slices starting at a left hand side of the picture, and parallel decoding the WPP substreams in a staggered manner with sequentially commencing the decoding of the WPP substreams in accordance with the slice order.
9. Decoder according to any of 1 to 8, wherein the decoder is configured to split chroma transform blocks differently than luma transform blocks in response to information in the data stream.

10. Encoder for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the encoder is configured to encode the slices (14) into the datastream (12) in accordance with a slice order (16) and the encoder is configured to
determine a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and
if the current slice is to be coded in accordance with a second (22) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,
wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the encoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and
wherein the encoder is configured to determine a syntax element for, and code same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, the encoder is configured to save symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy

encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the encoder is configured to initialize the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,

wherein the encoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice encoded in the second mode.

11. Encoder according to claim 10, wherein the encoder is configured to split chroma transform blocks differently than luma transform blocks.
12. Encoder according to claim 9, wherein the encoder is configured code the syntax element portion (18) into the current slice of the slices (14) so that the current slice is signaled to be coded thereinto in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the encoder is configured to
in accordance with the third mode (42), encode the current slice into the datastream using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive coding across the slice boundaries, wherein the encoder distinguishes between the one of the first and third modes using a syntax element.
13. Encoder according to claims 9 or 10, wherein the encoder is configured to determine a generic syntax element and write same into the datastream with operating in one of at least two generic operating modes depending on the generic syntax element, namely, with, according to a first generic operating mode, performing coding the syntax element portion for each slice, and, according to a second generic operating

mode, inevitably using a different one of the at least two modes other than the first mode.

14. Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and the method is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), wherein
in accordance with a first (20) of the at least two modes), the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and
in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, and
wherein the method is responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so,

initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the method comprises initializing the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not, wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice decoded in the second mode.

15. Method for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises encoding the slices (14) into the datastream (12) in accordance with a slice order (16) and the method comprises determining a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and if the current slice is to be coded in accordance with a second (22) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice,

and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

wherein the method further comprises determining a syntax element for, and coding same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, the method further comprises saving symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the method further comprises initializing the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice encoded in the second mode.

16. Datastream encoded using the method according to claim 15.

17. Datastream according to claim 16, wherein chroma transform blocks are split differently than luma transform blocks.
18. Computer program having a program code for performing, when running on a computer, a method according to claim 14 or 15.

Hilfsantrag IV vom 21. Oktober 2022 lautet:

Hilfsantrag IV

Neue Ansprüche (mit hervorgehobenen Änderungen)

1. Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the decoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and wherein the decoder is configured to be responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks

(32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the decoder is configured to initialize the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,
wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the immediately preceding slice decoded in the second mode.

2. Decoder according to claim 1, wherein the decoder is configured to be responsive to the syntax element portion (18) within the current slice of the slices (14), so as to decode the current slice in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the decoder is configured to
in accordance with the third mode (42), decode the current slice from the datastream using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding across the slice boundaries, wherein the one of the first and third modes is selected depending on a syntax element.
3. Decoder according to claim 1 or 2, wherein the decoder is configured to be responsive to a generic syntax element in the datastream so as to operate in one of at least two generic operating modes, with, according to a first generic operating mode, performing the responsiveness to the syntax element portion for each slice,

and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.

4. Decoder according to any of claims 1 to 3, wherein the decoder is configured to according to the first and second modes, inevitably und uninterruptedly continue continuously updating the symbol probabilities from a beginning to an end of the current slice.
5. Decoder according to any of claims 1 to 4, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to an end of the previously decoded slice.
6. Decoder according to any of claims 1 to 5, wherein the decoder is configured to, in the first and second mode, restrict the predictive decoding within tiles into which the picture is sub-divided.
7. Decoder according to any of 1 to 6, where the decoder is configured to, in the first and second mode, read information from current slice revealing a subdivision of the current slice into parallel subsections, cease the context adaptive entropy decoding at the end of the first parallel subsection and resume the context adaptive entropy decoding anew at a beginning of any subsequent parallel subsection including, in the first mode, an initialization of the symbol probabilities depending on saved states of symbol probabilities of the preceding parallel subsection and, in the second mode, an initialization of the symbol probabilities independent from any previously decoded slice and any previously decoded parallel subsection.
- ~~8. Decoder according to any of 1 to 7, where the decoder is configured to, in accordance with the first (20) of the at least two modes, copy for the current slice a portion of a slice header syntax from a preceding slice decoded in the second mode.~~
89. Decoder according to any of the preceding claims, wherein the decoder is configured to reconstruct the picture (10) from the datastream (12) using WPP processing, wherein each slice (14) comprises a start syntax portion (400) indicating a position of a decoding begin of the respective slice within the picture (10) and wherein the decoder is configured to identify entry points of WPP substreams into which the slices are grouped, by identifying, using the slices' start syntax portions, slices starting at a left hand side of the picture, and parallel decoding the WPP

substreams in a staggered manner with sequentially commencing the decoding of the WPP substreams in accordance with the slice order.

| 940. Decoder according to any of 1 to 89, wherein the decoder is configured to split chroma transform blocks differently than luma transform blocks in response to information in the data stream.

| 1044. Encoder for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the encoder is configured to encode the slices (14) into the datastream (12) in accordance with a slice order (16) and the encoder is configured to

determine a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and

if the current slice is to be coded in accordance with a second (22) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the encoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

| wherein the encoder is configured to determine a syntax element for, and code same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, wherein the encoder is configured to save symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in ini-

tializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the encoder is configured to initialize the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not.

wherein the encoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the immediately preceding slice encoded in the second mode.

1142. Encoder according to claim 1044, wherein the encoder is configured to split chroma transform blocks differently than luma transform blocks.

1243. Encoder according to claim 940, wherein the encoder is configured code the syntax element portion (18) into the current slice of the slices (14) so that the current slice is signaled to be coded therein in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the encoder is configured to in accordance with the third mode (42), encode the current slice into the datastream using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive coding across the

slice boundaries, wherein the encoder distinguishes between the one of the first and third modes using a syntax element.

| [1314](#). Encoder according to claims [910](#) or [1011](#), wherein the encoder is configured to determine a generic syntax element and write same into the datastream with operating in one of at least two generic operating modes depending on the generic syntax element, namely, with, according to a first generic operating mode, performing coding the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.

| [1415](#). Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and the method is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), wherein in accordance with a first (20) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, and

wherein the method is responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the method comprises initializing the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the immediately preceding slice decoded in the second mode.

1546. Method for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises encoding the slices (14) into the datastream (12) in accordance with a slice order (16) and the method comprises
- determining a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and
- if the current slice is to be coded in accordance with a first (20) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive

entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and

if the current slice is to be coded in accordance with a second (22) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and

wherein the method further comprises determining a syntax element for, and coding same into, a picture parameter set, so that the syntax element signals operation according to one of two operating types, wherein, according to the first operating type, wherein the method further comprises saving symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, wherein, according to the second operating type, the method further comprises initializing the symbol probabilities for the context adaptive entropy encoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy encoding the previ-

ously encoded slice up to the end of the previously encoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the **immediately preceding** slice encoded in the second mode.

1647. Datastream encoded using the method according to claim 1546.

1748. Datastream according to claim 1647, wherein chroma transform blocks are split differently than luma transform blocks.

1849. Computer program having a program code for performing, when running on a computer, a method according to claim 1445 or 1546.

Hilfsantrag V vom 21. Oktober 2022 lautet:

Hilfsantrag V

Neue Ansprüche (mit hervorgehobenen Änderungen)

1. Decoder for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the decoder is configured to decode the slices (14) from the datastream (12) in accordance with a slice order (16) and the decoder is responsive to a syntax element portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), and in accordance with a first (20) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and in accordance with a second (22) of the at least two modes, decode the current slice from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the decoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and wherein the decoder is configured to be responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, wherein~~the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks~~

(32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the decoder is configured to initialize the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,
wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the immediately preceding slice decoded in the second mode.

2. Decoder according to claim 1, wherein the decoder is configured to be responsive to the syntax element portion (18) within the current slice of the slices (14), so as to decode the current slice in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the decoder is configured to
in accordance with the third mode (42), decode the current slice from the datastream using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding across the slice boundaries, wherein the one of the first and third modes is selected depending on a syntax element.
3. Decoder according to claim 1 or 2, wherein the decoder is configured to be responsive to a generic syntax element in the datastream so as to operate in one of at least two generic operating modes, with, according to a first generic operating mode, performing the responsiveness to the syntax element portion for each slice,

and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.

4. Decoder according to any of claims 1 to 3, wherein the decoder is configured to according to the first and second modes, inevitably und uninterruptedly continue continuously updating the symbol probabilities from a beginning to an end of the current slice.
5. Decoder according to any of claims 1 to 4, wherein the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to an end of the previously decoded slice.
6. Decoder according to any of claims 1 to 5, wherein the decoder is configured to, in the first and second mode, restrict the predictive decoding within tiles into which the picture is sub-divided.
7. Decoder according to any of 1 to 6, where the decoder is configured to, in the first and second mode, read information from current slice revealing a subdivision of the current slice into parallel subsections, cease the context adaptive entropy decoding at the end of the first parallel subsection and resume the context adaptive entropy decoding anew at a beginning of any subsequent parallel subsection including, in the first mode, an initialization of the symbol probabilities depending on saved states of symbol probabilities of the preceding parallel subsection and, in the second mode, an initialization of the symbol probabilities independent from any previously decoded slice and any previously decoded parallel subsection.
- ~~8. Decoder according to any of 1 to 7, where the decoder is configured to, in accordance with the first (20) of the at least two modes, copy for the current slice a portion of a slice header syntax from a preceding slice decoded in the second mode.~~
89. Decoder according to any of the preceding claims, wherein the decoder is configured to reconstruct the picture (10) from the datastream (12) using WPP processing, wherein each slice (14) comprises a start syntax portion (400) indicating a position of a decoding begin of the respective slice within the picture (10) and wherein the decoder is configured to identify entry points of WPP substreams into which the slices are grouped, by identifying, using the slices' start syntax portions, slices starting at a left hand side of the picture, and parallel decoding the WPP

substreams in a staggered manner with sequentially commencing the decoding of the WPP substreams in accordance with the slice order.

910. Decoder according to any of 1 to 89, wherein the decoder is configured to split chroma transform blocks differently than luma transform blocks in response to information in the data stream.

~~11. Encoder for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the encoder is configured to encode the slices (14) into the datastream (12) in accordance with a slice order (16) and the encoder is configured to determine a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and if the current slice is to be coded in accordance with a first (20) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and if the current slice is to be coded in accordance with a second (22) of the at least two modes, encode the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries, wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the encoder is configured to associate each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and wherein the encoder is configured to save symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, check as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a~~

~~first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initialize (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice.~~

~~12.—Encoder according to claim 11, wherein the encoder is configured to split chroma transform blocks differently than luma transform blocks.~~

~~13.—Encoder according to claim 10, wherein the encoder is configured code the syntax element portion (18) into the current slice of the slices (14) so that the current slice is signaled to be coded therein in accordance with one of at least three modes, namely in the one of the first (20) and a third mode (42) or a second mode (22), wherein the encoder is configured to in accordance with the third mode (42), encode the current slice into the datastream using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive coding across the slice boundaries, wherein the encoder distinguishes between the one of the first and third modes using a syntax element.~~

~~14.—Encoder according to claims 10 or 11, wherein the encoder is configured to determine a generic syntax element and write same into the datastream with operating in one of at least two generic operating modes depending on the generic syntax element, namely, with, according to a first generic operating mode, performing coding the syntax element portion for each slice, and, according to a second generic operating mode, inevitably using a different one of the at least two modes other than the first mode.~~

1015. Method for reconstructing a picture (10) from a datastream (12) into which the picture is coded in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises decoding the slices (14) from the datastream (12) in accordance with a slice order (16) and the method is responsive to a syntax ele-

ment portion (18) within a current slice of the slices, so as to decode the current slice in accordance with one of at least two modes (20, 22), wherein in accordance with a first (20) of the at least two modes), the current slice is decoded from the datastream (12) using context adaptive entropy decoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously decoded slice, and predictive decoding across the slice boundaries, and in accordance with a second (22) of the at least two modes, the current slice is decoded from the datastream (12) using context adaptive entropy decoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously decoded slice, and predictive decoding with restricting the predictive decoding so as to not cross the slice boundaries,

wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the coding blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice, and

wherein the method is responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types, wherein, according to the first operating type, the method comprises saving symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy decoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initializing (40) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy decoding of the current slice depending on symbol probabilities as obtained in context

adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice, wherein, according to the second operating type, the method comprises initializing the symbol probabilities for the context adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not,

wherein the method comprises, in accordance with the first (20) of the at least two modes, inferring for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the immediately preceding slice decoded in the second mode.

16. ~~Method for encoding a picture (10) into a datastream (12) in units of slices (14) into which the picture (10) is partitioned, wherein the method comprises encoding the slices (14) into the datastream (12) in accordance with a slice order (16) and the method comprises~~
- ~~determining a syntax element portion (18) for, and code same into, a current slice of the slices so that the syntax element portion signals the current slice to be coded in accordance with one of at least two modes (20, 22), and~~
- ~~if the current slice is to be coded in accordance with a first (20) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding (24) including a derivation of contexts across slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization (38, 40) of the symbol probabilities depending on saved states of symbol probabilities of a previously encoded slice, and predictive encoding across the slice boundaries, and~~
- ~~if the current slice is to be coded in accordance with a second (22) of the at least two modes, encoding the current slice into the datastream (12) using context adaptive entropy encoding with restricting the derivation of the contexts so as to not cross the slice boundaries, a continuous update of symbol probabilities of the contexts and an initialization of the symbol probabilities independent on any previously encoded slice, and predictive encoding with restricting the predictive encoding so as to not cross the slice boundaries,~~
- ~~wherein the picture (10) is partitioned in coding blocks (32) arranged in rows and columns and having a raster scan order (36) defined among each other, and the method comprises associating each slice (14) with a continuous subset of the cod-~~

~~ing blocks (32) in the raster scan order (36) so that the subsets follow each other along the raster scan order (36) in accordance with the slice order, and wherein the method further comprises saving symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and, in initializing the symbol probabilities for the context adaptive entropy encoding of the current slice in accordance with the first mode, checking as to whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order, and, if so, initialize (40) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on the saved symbol probabilities as obtained in context adaptive entropy encoding the previously encoded slice up to a second coding block in a row in accordance with the raster scan order (36), and, if not, initializing (38) the symbol probabilities for the context adaptive entropy encoding of the current slice depending on symbol probabilities as obtained in context adaptive entropy encoding the previously decoded slice up to the end of the previously encoded slice.~~

~~17. — Datastream encoded using the method according to claim 16.~~

~~18. — Datastream according to claim 17, wherein chroma transform blocks are split differently than luma transform blocks.~~

1149. Computer program having a program code for performing, when running on a computer, a method according to claim ~~1045 or 16~~.

Wegen der weiteren Einzelheiten wird auf den Akteninhalt verwiesen.

Entscheidungsgründe

Die Klage, mit der der Nichtigkeitsgrund der fehlenden Patentfähigkeit (Art. II § 6 Abs. 1 Satz 1 Nr. 1 IntPatÜG, Art. 138 Abs. 1 lit. a) EPÜ i. V. m. Art. 52 Abs. 1, 54 und 56 EPÜ) mit Blick auf fehlende Neuheit und fehlende erfinderische Tätigkeit geltend gemacht wird, ist gemäß § 81 PatG zulässig.

Die Klage ist auch begründet. Denn die Patentansprüche 1 und 15 des Streitpatents haben weder hinsichtlich des Hauptantrags noch hinsichtlich der Hilfsanträge Bestand, weil den Patentansprüchen 1 und 15 des Streitpatents gemäß Hauptantrag und Hilfsanträgen der Nichtigkeitsgrund der fehlenden Patentfähigkeit entgegensteht.

I.

1. Das Streitpatent betrifft eine Bildcodierung mit geringer Verzögerung und beschäftigt sich mit der Verbesserung herkömmlicher Codierverfahren und Decodierverfahren zur Videokompression (Streitpatentschrift, Abs. [0001], [0011] bis [0014]).

Als bekannte Codierverfahren nennt das Streitpatent den im Jahr 2003 verabschiedeten H.264-Standard (auch AVC-Standard genannt) sowie den im Prioritätszeitraum im Januar 2013 verabschiedeten HEVC-Standard (auch H.265-Standard genannt) (Streitpatentschrift, Abs. [0003]). Das Streitpatent beschreibt, dass in früheren Video-Codierverfahren, wie z. B. dem H.264-Standard, eine parallele Verarbeitung von Bilddaten aufgrund begrenzter Möglichkeiten der Unterteilung eines Bildes in Bildpartitionen nur beschränkt möglich gewesen sei. Außerdem hätten sich diese dort vorgesehenen Möglichkeiten der Unterteilung in Bildpartitionen auf die Effizienz der Codierung nachteilig ausgewirkt. Insbesondere habe für den Fall einer parallelen Decodierung nach dem H.264-Standard lediglich eine begrenzte Reduktion von Bildlatenzzeiten bei gleichzeitig

hohem Speicherplatzverbrauch erzielt werden können (Streitpatentschrift, Abs. [0003]). Bislang seien alle Versuche einer parallelen Codierung/Decodierung daran gescheitert, bei der Codierung, Übertragung und Decodierung von Videodaten eine hohe Kompressionseffizienz bei einer gleichzeitig geringen Ende-zu-Ende- Verzögerung zu erreichen (Streitpatentschrift, Abs. [0011]).

Das Streitpatent lehrt Parallelverarbeitungskonzepte wie das Wellenfrontparallel-Verfahren, mit dem eine reduzierte Ende-zu-Ende-Verzögerung realisiert werden kann, wenn Bildbereiche bzw. „Stücke“ („slices“) hinsichtlich der Entropiecodierung und –decodierung nicht mehr zwingend unabhängig voneinander codiert/decodiert werden, sondern wenn „Stücke“ verschiedener Modi vorgesehen werden, nämlich solche, die als „abhängige Stücke“ bezeichnet werden und die auch für Entropiecodierung und –decodierung Interdependenzen über die Stückgrenzen hinweg zulassen, und solche, die dies nicht ermöglichen und als „normale Stücke“ bezeichnet werden (Streitpatentschrift, Abs. [0015]).

2. Erläuterungen zur „Wellenfrontparallelverarbeitung“ („wavefront parallel processing“ WPP)

Um bei der Videocodierung eine geringe Latenz, d. h. einen geringen Zeitversatz zwischen Beginn und Ende der Codierung eines Bildes zu erreichen, ist es hilfreich, wenn der Codierer parallel, d. h. an mehreren Teilen eines Bildes gleichzeitig arbeiten kann. In diesem Zusammenhang stellt die Wellenfrontparallelverarbeitung („wavefront parallel processing“ WPP) ein Konzept für eine leistungsstarke Parallelisierung dar, das in den H.265/HEVC-Standard mit übernommen wurde. WPP ermöglicht die gleichzeitige Verarbeitung aufeinanderfolgender Zeilen mit Codierungsbaumeinheiten CTUs innerhalb eines Frames. Gewöhnlich wird ein waagrechter Offset von wenigstens zwei CTUs implementiert, um die Abhängigkeiten zwischen den CTUs zu erhalten. Oder anders ausgedrückt: die Grundidee von WPP besteht darin, eine parallele Codierung einer Zeile zu beginnen, sobald mindestens zwei CTUs der darüber liegenden Zeile codiert sind. Dadurch werden mehrere CTUs gleichzeitig codiert.

Die gleichzeitig ablaufenden Unterprozesse werden „Threads“ genannt (siehe Abb. 1).

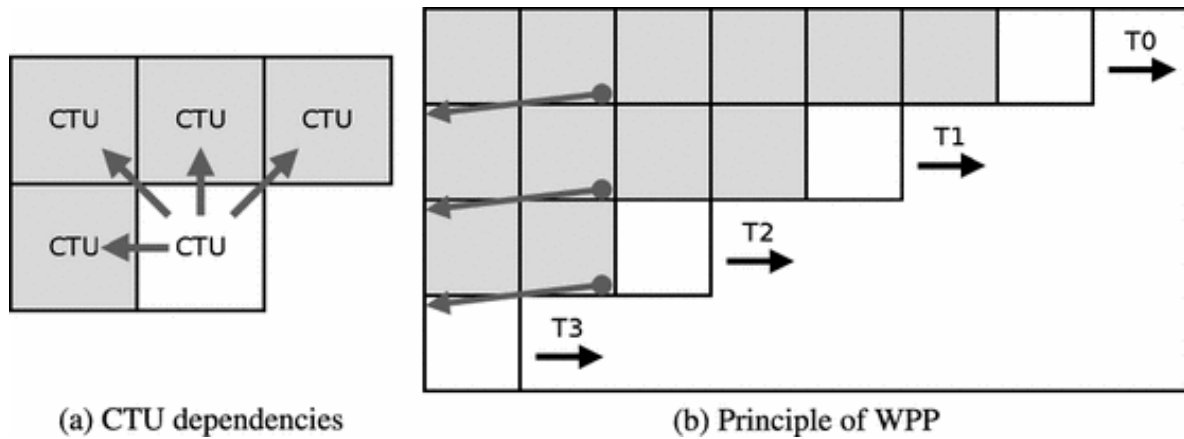


Abbildung 1: Prinzip der Wellenfrontparallelverarbeitung WPP mit Threads T0 bis T3 (aus: S. Radicke et al.: Many-Core HEVC Encoding Based on Wavefront Parallel Processing and GPU-accelerated Motion Estimation, Conference Paper, Part of the [Communications in Computer and Information Science](#) book series (CCIS, volume 554)).

Die für die Prädiktion verfügbaren CTUs sind dann die CTUs links und über einer gedachten Diagonalen durch das Bild, die sich wie eine Wellenfront durch das Bild fortpflanzt.

Was die Entropiecodierung mit WPP anbelangt, so verwendet jeder der parallelen Threads seine eigene Codiertabelle. Um die bereits codierten CTUs auszunutzen, wird bei Beginn des Threads die Codiertabelle von demjenigen Thread übernommen („synchronisiert“), der die Zeile unmittelbar darüber codiert. Da ein Thread dann beginnt, wenn der vorhergehende Thread seine zweite CTU codiert hat, übernimmt ein neuer Thread somit die Codiertabelle nach der zweiten CTU der darüberliegenden Zeile. Mit der so synchronisierten Codiertabelle codiert der Thread in seiner Zeile von links nach rechts CTU für CTU und aktualisiert seine Codiertabelle dabei laufend (siehe Abb. 2).

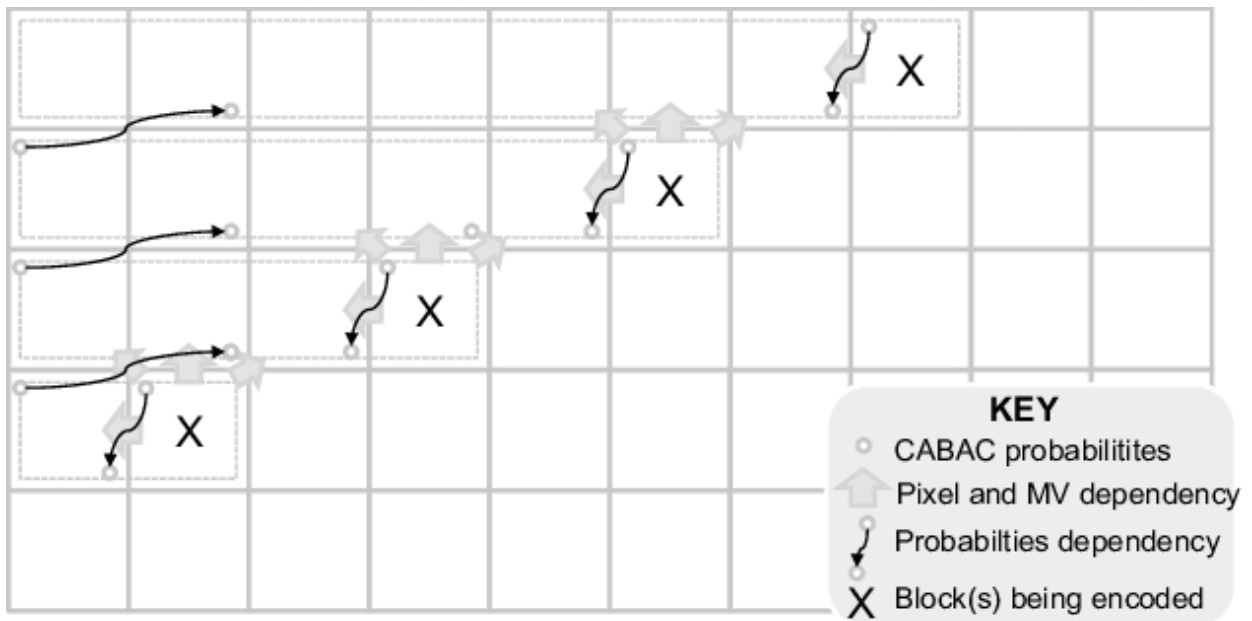


Abbildung 2: Wellenfrontparallelverarbeitung WPP mit Abhängigkeiten von Pixeln, Bewegungsvektoren MV und CABAC Wahrscheinlichkeiten bzw. Codiertabellen (aus: P. Bordes et al.: An overview of the emerging HEVC Standard, Conference Paper, International Symposium on Signal, Image, Video and Communications ISIVC 2012)).

3. Ausgehend von diesem Stand der Technik macht es sich das Streitpatent zur Aufgabe, ein Bildcodierungskonzept bereitzustellen, das eine parallele Decodierung z. B. nach der Wellenfrontparallelverarbeitung („wavefront parallel processing“) mit erhöhter Effizienz ermöglicht, wie zum Beispiel eine weitere Reduzierung der Ende-zu-Ende-Verzögerung („end-to-end delay“) sowie eine Verbesserung der Codiereffizienz durch Verringerung des Codieraufwandes (Streitpatentschrift, Abs. [0013]).

4. Die oben genannte Aufgabe soll erfindungsgemäß gelöst werden durch einen „Decodierer zum Rekonstruieren eines Bildes aus einem Datenstrom“ nach Patentanspruch 1 und „Ein Verfahren [zum] Rekonstruieren eines Bildes aus einem Datenstrom“ nach Patentanspruch 15.

5. Als zuständigen **Durchschnittsfachmann**, auf dessen Wissen und Können es insbesondere für die Auslegung der Merkmale des Streitpatents und für die Interpretation des Standes der Technik ankommt, sieht der Senat einen Hochschul-Absolventen aus dem Bereich der Informationstechnik, Informatik oder Elektrotechnik an, der über mehrjährige Berufserfahrung auf dem Gebiet der Entwicklung von digitaler Videocodierung und –decodierung verfügt, der mit den dabei zum Einsatz kommenden Techniken und den wichtigen Standards vertraut ist und dem insbesondere die im Rahmen der Standardisierungsprozesse diskutierten Beiträge zur Weiterentwicklung bestehender Standards bekannt sind.

6. Die streitpatentgemäße Lehre ist aus Sicht eines solchen Fachmanns wie folgt zu erläutern:

a) Zum Begriff „Slice“ bzw. „Stück“

Für den englischen Begriff „Slice“ im Streitpatent wird in der deutschen Übersetzung der Patentansprüche der Begriff „Stück“ verwendet. Beide Begriffe betreffen denselben Bedeutungsinhalt: ein „Slice“ stellt ein Strukturelement dar, das sowohl im H.264/AVC- als auch im H.265/HEVC-Standard definiert wird, um die Größe der für die Übertragung verwendeten Datenpakete an die jeweiligen Bedürfnisse anpassen zu können. Laut Streitpatent wird ein „Slice“ durch eine beliebige Anzahl von aufeinanderfolgenden Blöcken (z. B. Makroblöcke in AVC, „coding tree units“ CTUs in HEVC) gebildet, so dass jedes Bild in einen oder mehrere „Slices“ aufgeteilt werden kann (Streitpatentschrift, Fig. 1, Abs. [0001] bis [0004], Abs. [0044], siehe Seite 6, Zeilen 50 bis 52). Insoweit entspricht ein „Slice“ gleichzeitig einem „Stück“ des zu codierenden bzw. decodierenden Bildes. Ausgehend vom Streitpatent wird im Folgenden abweichend von der deutschen Übersetzung der technisch zutreffendere Begriff „Slice“ anstelle von „Stück“ verwendet.

Weiterhin unterscheidet das Streitpatent in einem ersten Ausführungsbeispiel zwischen normalen bzw. unabhängigen „Slices“ (siehe „regular or normal slices“,

„independent slice“) und abhängigen „Slices“ (Streitpatentschrift, Abs. [0090] bis [0153], siehe „dependent slices“). Während unabhängige „Slices“ sowohl Prädiktions- als auch Entropiecodierung „brechen“ (Streitpatentschrift, Abs. [0004]), „brechen“ abhängige „Slices“ weder Prädiktions- noch Entropiecodierung (Streitpatentschrift, Abs. [0015]). Damit ist gemeint, dass im ersten Fall bei der Codierung die Blöcke eines „Slices“ lediglich durch Bezugnahme auf andere Blöcke desselben „Slices“ prädiktionscodiert werden; Blöcke eines anderen „Slices“ dürfen aber dazu nicht herangezogen werden. In gleicher Weise dürfen zur Entropiecodierung der Blöcke eines „Slices“ nur Bilddaten aus anderen Blöcken desselben „Slices“ verwendet werden, um die Häufigkeitsverteilung der zu codierenden Werte abzuschätzen. Im Gegensatz hierzu kann im Fall eines abhängigen „Slices“ zur Prädiktions- und Entropiecodierung auf die Blöcke bzw. Häufigkeitsverteilungen in einem anderen „Slice“ zurückgegriffen werden. Dementsprechend verfügt jedes unabhängige bzw. jedes abhängige „Slice“ über Slice-Grenzen (Streitpatentschrift, Fig. 9, 20, 24 u. a.), an denen Prädiktions- und Entropiecodierung „gebrochen“ werden bzw. nicht gebrochen werden.

In einem zweiten Ausführungsbeispiel spricht das Streitpatent in einem geringfügig abgeänderten Wortlaut („slightly different wording“) von unabhängigen bzw. abhängigen „Slice-Segmenten“ (Streitpatentschrift, Abs. [0154] bis [0170]). Die genannten „Slice-Segmente“ können wiederum derart zu einem (übergeordneten) „Slice“ zusammengefasst werden, dass dieses ein unabhängiges „Slice-Segment“ und kein, ein oder mehrere nachfolgende abhängige „Slice-Segmente“ beinhaltet (Streitpatentschrift, Fig. 27; Abs. [0155]; [0156], siehe Seite 22, Zeilen 19 bis 21). Dieses zusammengesetzte „Slice“ hat als logische Einheit dieselben Eigenschaften bezüglich Prädiktions- und Entropiecodierung wie ein herkömmliches bzw. reguläres „Slice“, wie es etwa aus dem H.264/AVC-Standard bekannt ist (Streitpatentschrift, Abs. [0004]).

Vor dem Hintergrund der Beschreibung des Streitpatents und dem klägerseitigen Vorschlag folgend, erachtet der Senat eine Auslegung des Begriffs „Slice“ für sachgerecht, die zwei Alternativen vorsieht:

- (aa) „Slice“ iSd ersten Alternative des Streitpatents ist ein Strukturelement, das eine Anzahl von aufeinanderfolgenden Blöcken eines zu codierenden Bildes enthält, als logische Einheit mit Grenzen für die Paketierung von Videodaten aufgefasst und mittels Syntaxelementen so konfiguriert werden kann, dass seine Grenzen Prädiktions- und Entropiecodierung entweder „brechen“ oder nicht. In diesem Fall wird die Codierung nicht notwendigerweise durch die Grenzen eines übergeordneten Strukturelements eingeschränkt.
- (bb) „Slice“ iSd zweiten Alternative des Streitpatents ist ein Strukturelement, das eine Anzahl von aufeinanderfolgenden Blöcken innerhalb eines übergeordneten, herkömmlichen „Slices“ enthält und das als logische Einheit mit Grenzen für die Paketierung von Videodaten aufgefasst werden kann, wobei mittels Syntaxelementen angegeben werden kann, ob es den Anfang des übergeordneten, herkömmlichen „Slices“ bildet oder nicht. Außerdem kann das „Slice“ iSd zweiten Alternative so konfiguriert werden, dass an seinen Grenzen Prädiktions- und Entropiecodierung entweder „gebrochen“ werden oder nicht, wobei allerdings die Grenzen des herkömmlichen (regulären) „Slices“ die Codierung beschränken.
- b) Zum Begriff „Syntaxelementabschnitt“ („syntax element portion“)

Die einzelnen Elemente, die neben den codierten Bilddaten den Daten- bzw. Bitstrom bilden, werden Syntaxelemente genannt. Sie definieren die grammatikalische Struktur des Bitstroms und bestimmen, wie der Bildinhalt aus den einzelnen Blöcken wiederhergestellt werden kann. Syntaxelemente betreffen z. B. Daten zur Aufteilung der Bilder in „Slices“, Makroblöcke, Prädiktionsblöcke und Transformationsblöcke oder die bei der Codierung gewählten Prädiktionsmodi (Streitpatentschrift, Abs. [0131]). Syntaxelemente, die den jeweiligen Modus festlegen, gemäß dem ein anspruchsgemäßes „Slice“ decodiert werden soll, sind in einem Syntaxelementabschnitt („syntax element portion“) untergebracht. Der Syntaxelementabschnitt bildet dabei einen Teil des Slice-

Headers (Fig. 21; Seite 12, Zeilen 25 ff) und umfasst insbesondere die Syntaxelemente *dependent_slice_flag* und *no_cabac_reset_flag* (Streitpatentschrift, Abs. [0147]; Fig. 24).

c) Zum Begriff „Symbolwahrscheinlichkeit“ („symbol probability“)

Mit „Symbolwahrscheinlichkeit“ ist die Auftrittswahrscheinlichkeit eines zu codierenden Symbols, wie beispielsweise eines Zeichens oder einer Ziffer, gemeint, die sich aus der relativen Häufigkeit seines Auftretens ableiten lässt. Anstatt die aus der Transformationscodierung erhaltenen Werte mit konstanter Codewortlänge zu codieren, werden häufiger auftretende bzw. wahrscheinlichere Symbole mit kürzeren Codewörtern bzw. Bitsequenzen codiert als seltener Symbole.

d) Zum Begriff „kontextadaptive Entropiedecodierung“ („context adaptive entropy decoding“).

Mit „kontextadaptiver Entropiedecodierung“ ist gemeint, dass sich die Entropiedecodierung an den jeweiligen Kontext anpasst. Das Streitpatent lehrt hierzu, dass der anspruchsgemäße Decodierer gemäß einem ersten Modus ein „Slice“ unter Verwendung von „kontextadaptiver Entropiedecodierung“ aus dem Datenstrom decodiert, wobei der Kontext über die Slice-Grenzen hinweg hergeleitet wird, d. h. aus vorhergehenden, bereits decodierten „Slices“ stammende Informationen verwendet werden. Nach fachmännischem Verständnis handelt es sich bei einem solchen Kontext um ein spezifisches Wahrscheinlichkeitsmodell, das durch einen Satz von Kontextvariablen bestimmt ist (vgl. Streitpatentschrift, Abs. [0078] bis [0089] u. a.). Insbesondere ist der Decodierer in der Lage, sich während des Decodiervorgangs dynamisch an die statistischen Eigenschaften der zu decodierenden Daten anzupassen. Dies bedeutet insbesondere, dass die Symbolwahrscheinlichkeiten des Codecs (d. h. des Verbunds aus Codierer und Decodierer) ständig aktualisiert werden und zu Beginn des Decodiervorgangs eines „Slices“ eine Initialisierung der Symbolwahrscheinlichkeiten auf die gespeicherten Symbolwahrscheinlichkeiten

eines vorher decodierten „Slices“ stattfindet (Streitpatentschrift, Abs. [0125]).

e) Zum Begriff „Codierblock“ („coding block“)

Das Streitpatent geht aus von Codierungsbaumblöcken CTB's („coding treeblocks“), die ihrerseits für jede Luma/Chroma-Komponente in Blöcke unterteilt werden (Streitpatentschrift, Abs. [0044], siehe „treeblock: A NxN block of luma samples and two corresponding blocks of chroma samples of a picture that has three sample arrays ...“). Insoweit entspricht der im Streitpatent genannte Codierungsbaumblock CTB einer Codierungsbaumeinheit CTU („coding tree unit“), die die grundlegende Verarbeitungseinheit des Videocodierstandards H.265/HEVC bildet.

Weiterhin wird im Streitpatent der Begriff Codierblock unterschiedlich verwendet. Einerseits kann es sich bei den Codierblöcken z. B. um die jeweiligen Blätter bzw. Endknoten eines Quadrees handeln, in die die Codierungsbaumblöcke CTBs durch rekursive Quadtree-Partitionierung aufgeteilt worden sind (Streitpatentschrift, Abs. [0130], siehe „Coding blocks 30 are, for example, leaf blocks into which coding tree blocks or largest coding blocks 32 are partitioned by recursive multitree partitioning such as quadtree partitioning.“), und die die Wurzelknoten von Prädiktions- und Transformationsbäumen darstellen können (Streitpatentschrift, Abs. [0050], siehe „The coding block is the root node of two trees, the prediction tree and the transform tree.“). Weiterhin ist aus Absatz [0144] der Streitpatentschrift abzuleiten, dass das Streitpatent grundsätzlich jede Partition eines Codierungsbaumblocks als Codierblock versteht und nicht nur die Blätter bzw. Endknoten eines Quadrees mit Prädiktions- und Transformationsbäumen (Streitpatentschrift, Abs. [0144], siehe „Naturally, the partitioning of blocks 32 in a recursive manner into further coding blocks 30 ...“).

Jedenfalls kann es sich bei solchen Codierblöcken zum einen um Untereinheiten von Codierungsbaumblöcken („coding tree blocks“ CTB) handeln.

Andererseits können Codierblöcke aber auch direkt durch die Codierungsbaumblöcke CTBs selbst repräsentiert werden (Streitpatentschrift, Fig. 25, Abs. [0130], siehe „...merely one code treeblock 32 is shown to be further partitioned into coding blocks 30, while the other code treeblocks 32 are shown to be not further partitioned so as to directly form a coding block, instead.“; Abs. [0144], siehe „Naturally, the partitioning of blocks 32 [the coding tree blocks] in a recursive manner into further coding blocks 30 is optional and accordingly, in a more general sense, blocks 32 [the coding tree blocks 32] could be called „coding blocks“ as well.“).

Da obige Untereinheiten von Codierungsbaumblöcken zwar rechteckig, aber nicht immer gleich groß sein müssen, kann in Hinblick auf Merkmal **M1.4**, das eine rasterförmige Anordnung von Codierblöcken in Zeilen und Spalten vorsieht, eine widerspruchsfreie Auslegung allenfalls dann gelingen, wenn miteinander kombinierte Untereinheiten als Codierblöcke Zeilen und Spalten bilden. Werden Codierblöcke mit ganzen Codierbaumblöcken CTBs gleichgesetzt, so erfolgt eine regelmäßige Partitionierung des Bildes in Codierungsbaumblöcke CTBs, wie in der Patentbeschreibung explizit vorgesehen (Streitpatentschrift, Abs. [0130], siehe „The code treeblocks 32, in turn, may be regularly arranged in columns and rows ...“; Abs. [0143]; [0144]; [0149], Seite 20, Zeilen 24, 25; [0150], Seite 21, Zeilen 9, 10).

f) Zum Begriff „Rasterabtastreihenfolge“ („raster scan order“)

Das Streitpatent beschreibt eine Rasterabtastung („raster scan“) als die Abbildung einer rechteckigen, zweidimensionalen und gerasterten Struktur, z. B. eines partitionierten Bildes auf eine eindimensionale Struktur. Dabei wird die zweidimensionale Struktur in einer Reihenfolge von links oben nach rechts unten zeilenweise durchlaufen, und die jeweils abgetasteten Einträge, z. B. Codierblöcke werden in dieser Reihenfolge auf die eindimensionale Struktur abgebildet (Streitpatentschrift, Abs. [0044], siehe „raster scan“). Hierzu zeigt Fig. 24 des Streitpatents ein Bild 10, das in fünf Zeilen zu je sechs Codierblöcken partitioniert worden ist. Die Rasterabtastreihenfolge wird dort durch den Pfeil 16

wiedergegeben, so dass die Codierblöcke der ersten Zeile von links nach rechts durchlaufen werden, dann die Codierblöcke der zweiten Zeile, beginnend mit dem Block ganz links, dann die Codierblöcke der dritten Zeile usw.

g) Zum Begriff „Stückreihenfolge“ („slice order“)

Basierend auf der „Rasterabtastrihenfolge“ für die Codierblöcke werden mehrere aufeinanderfolgende Codierblöcke zu einem unabhängigen/abhängigen „Slice“ bzw. „Slice-Segment“ zusammengefasst (Streitpatentschrift, Fig. 24; Fig. 27; Abs. [0125], [0126]). Die Reihenfolge für die so definierten „Slices“ bzw. „Slice-Segmente“ ergibt sich dann aus der für die Codierblöcke festgelegten „Rasterabtastrihenfolge“ (vgl. Fig. 24, siehe Ziffern 1 bis 5 für die gestrichelten Rechtecke, die die „Slices“ darstellen sollen).

h) Zur Lehre des erteilten Patentanspruchs 1

Der Patentanspruch 1 lehrt eine Decodier-Vorrichtung, die einen Daten- bzw. Bitstrom empfängt und verarbeitet. Der Datenstrom repräsentiert – in codierter Form – ein Bild bzw. eine Sequenz von Bildern. Der Decodierer „versteht“ das codierte Format des Datenstroms und ist in der Lage, es zu decodieren, um daraus ein Bild zu rekonstruieren. Das Bild ist dabei in Einheiten von „Slices“ codiert, die zusammenhängende Bereiche des Bildes bezeichnen, in die das Bild zum Zwecke der Codierung bzw. Decodierung unterteilt wird (Merkmal **M1.1**).

Gemäß Merkmal **M1.2** ist der Decodierer dazu ausgebildet, die „Slices“ entsprechend einer „Slice“- bzw. „Stückreihenfolge“ aus dem Datenstrom zu decodieren. Fig. 24 des Streitpatents zeigt eine solche Reihenfolge, in der die „Slices“ aus dem Datenstrom extrahiert werden. Dabei müssen die „Slices“ nicht notwendigerweise seriell decodiert werden, sondern sie können auch teilweise parallel decodiert werden. So kann der Decodierer die „Slices“ mittels „Wellenfrontparallelverarbeitung“ parallel auf gestaffelte Weise decodieren (Streitpatentschrift, Abs. [0124], siehe „... a decoder which is able to decode slices 14 in parallel in a staggered manner ...“).

Merkmal **M1.3** sieht für den Decodierer zwei verschiedene Modi vor, um den jeweiligen aktuellen „Slice“ zu decodieren. Welcher Modus anzuwenden ist, ergibt sich aus einem „Syntaxelementabschnitt“ des aktuellen „Slices“, auf den der Decodierer „anspricht“ bzw. den der Decodierer auswertet. In einem Ausführungsbeispiel des Streitpatents enthält der „Syntaxelementabschnitt“ das „Syntaxelement“ *dependent_slice_flag*, das für einen unabhängigen „Slice“ den Wert 0 und für einen abhängigen „Slice“ den Wert 1 hat (Streitpatentschrift, Abs. [0147]; [0149]).

In den Merkmalen **M1.3.1a/b** bzw. **M1.3.2a/b** werden der erste Modus und der zweite Modus festgelegt. Demnach wird das aktuelle „Slice“ in beiden Modi unter Verwendung von kontextadaptiver Entropiedecodierung und Prädiktionsdecodierung aus dem Datenstrom decodiert. Unterschiede zwischen den Modi bestehen darin, wie die Grenzen des aktuellen „Slices“ zu den benachbarten „Slices“ beim Decodieren berücksichtigt werden sollen.

Die Merkmale **M1.3.1a/b** sehen vor, dass im ersten Modus die Ableitung der Kontexte, die fortlaufende Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte, die Initialisierung der Symbolwahrscheinlichkeiten und die Prädiktionsdecodierung über die „Slice“-Grenzen hinweg erfolgen sollen.

Demgegenüber sehen die Merkmale **M1.3.2a/b** vor, dass im zweiten Modus weder die Ableitung der Kontexte, die fortlaufende Aktualisierung von Symbolwahrscheinlichkeiten der Kontexte, die Initialisierung der Symbolwahrscheinlichkeiten noch die Prädiktionsdecodierung die „Slice“-Grenzen überschreiten sollen.

In den Merkmalen **M1.3.1a/b** bzw. **M1.3.2a/b** kommt somit zum Ausdruck, dass im ersten Modus weder Entropiedecodierung noch Prädiktionsdecodierung an den „Slice“-Grenzen „gebrochen“ werden (weil es sich beim aktuellen „Slice“ um ein abhängiges „Slice“ handelt) und dass im zweiten Modus sowohl Entropiedecodierung als auch Prädiktionsdecodierung „gebrochen“ werden (weil

das aktuelle „Slice“ ein unabhängiges „Slice“ darstellt).

Merkmal **M1.4** besagt, dass das zu rekonstruierende Bild in Codierblöcke partitioniert ist, die in einem Raster aus Zeilen und Spalten angeordnet sind. Den Codierblöcken ist dabei eine Rasterabtastrihenfolge zugeordnet, mit der die zweidimensionale Anordnung der Codierblöcke in Zeilen und Spalten auf eine sequentielle Anordnung abgebildet wird.

In Merkmal **M1.5** wird beansprucht, dass der Decodierer dazu ausgebildet ist, jedem „Slice“ einen in der Rasterabtastrihenfolge durchgehenden Teilsatz der Codierblöcke zuzuordnen, und dass die Teilsätze einander entlang der Rasterabtastrihenfolge gemäß der „Slice“-Reihenfolge folgen.

Merkmal **M1.6** sieht vor, dass der anspruchsgemäße Decodierer Symbolwahrscheinlichkeiten speichert, wie er sie bei der kontextadaptiven Entropiedecodierung des zuvor decodierten „Slices“ bis zu einem zweiten Codierblock in einer Zeile erhalten hat. Merkmal **M1.6** bezieht sich damit auf die Initialisierung der Entropiedecodierung im Rahmen der Wellenfrontparallelverarbeitung WPP („wavefront parallel processing“), bei der der erste Codierblock einer Zeile mit den Symbolwahrscheinlichkeiten initialisiert wird, die bis zur Decodierung des zweiten Codierblocks in der darüber liegenden Zeile ermittelt worden sind.

Die Merkmale **M1.6.1** bis **M1.6.3** beziehen sich auf den Fall, dass die Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen „Slices“ gemäß dem ersten Modus initialisiert werden. Hierbei soll in Merkmal **M1.6.1** zunächst überprüft werden, ob es sich bei dem ersten Codierblock des aktuellen „Slices“ um den ersten Codierblock in einer Zeile der Codierblöcke handelt.

Wenn dies der Fall ist, werden gemäß Merkmal **M1.6.2** die Symbolwahrscheinlichkeiten für die Decodierung des aktuellen „Slices“ mit den unter Merkmal **M1.6** gespeicherten Werten initialisiert, d. h. mit Werten für

Symbolwahrscheinlichkeiten, wie sie bei der Entropiedecodierung des zuvor decodierten „Slices“ bis zu einem zweiten Codierblock in der Zeile darüber erhalten worden sind.

Wenn es sich andernfalls nicht um den ersten Codierblock in einer Zeile handelt, werden gemäß Merkmal **M1.6.3** diejenigen Symbolwahrscheinlichkeiten für eine Initialisierung übernommen, die bis zum Ende der Decodierung des vorherigen „Slices“ ermittelt worden sind.

Die im Merkmalskomplex **M1.6.1** bis **M1.6.3** festgelegte Fallunterscheidung soll also für den Fall durchgeführt werden, dass die Entropiedecodierung nicht durch „Slice“-Grenzen „gebrochen“ wird, d. h. das zu decodierende „Slice“ ein abhängiges „Slice“ darstellt.

Wie weiter oben bereits ausgeführt, sieht Merkmal **M1.3.2a** für den zweiten Modus vor (d. h. das aktuelle „Slice“ ist ein unabhängiges „Slice“), dass die Symbolwahrscheinlichkeiten unabhängig von jeglichem zuvor decodierten „Slice“ initialisiert werden.

7. Dem Vorbringen der Beklagten zur Auslegung der Merkmale des erteilten Patentanspruchs 1 kann nicht gefolgt werden.

7.1 Der Auffassung der Beklagten, dass erstes Ausführungsbeispiel und zweites Ausführungsbeispiel in ihrem technischen Gehalt parallel liefen und das zweite Ausführungsbeispiel nichts anderes sei als eine Verschiebung der Terminologie („Slices“ können demnach auch „Slice-Segmente“ sein) und dem Erfindungsgedanken keinen neuen Aspekt hinzufüge, ist nicht zu folgen.

(a) So ist die Auslegung des Patentanspruchs nach der ständigen Rechtsprechung des Bundesgerichtshofs stets geboten und darf auch dann nicht unterbleiben, wenn der Wortlaut des Anspruchs eindeutig zu sein scheint (*s. nur BGH, Urteil vom 29. April 1986 - X ZR 28/85, BGHZ 98, 12, 18 - Formstein; Urteil vom 12. März 2002 - X ZR 168/00, BGHZ 150, 149, 153 - Schneidmesser I;*

Beschluss vom 17. April 2007 - X ZB 9/06, BGHZ 172, 108 - Informationsübermittlungsverfahren I; Urteil vom 17. Juli 2012 – X ZR 117/11, BGHZ 194, 107, Rn. 27 - Polymerschaum I). Denn die Beschreibung des Patents kann Begriffe eigenständig definieren und insoweit ein „patenteigenes Lexikon“ darstellen (*BGH, Urteil vom 2. März 1999 - X ZR 85/96, GRUR 1999, 909 - Spannschraube*).

(b) Unter Beachtung dieser Grundsätze ist festzustellen, dass bei Auslegung der Patentansprüche im Lichte des zweiten Ausführungsbeispiels der Patentbeschreibung das anspruchsgemäße „Slice“ als „Slice-Segment“ zu verstehen ist, das Bestandteil eines übergeordneten Strukturelements ist, wobei es sich bei diesem übergeordneten Strukturelement um ein herkömmliches (reguläres) „Slice“ handelt.

Laut Absatz [0156] der Streitpatentschrift beginnt ein solches übergeordnetes „Slice“ immer mit einem unabhängigen „Slice-Segment“, auf das kein, ein oder mehrere abhängige „Slice-Segmente“ folgen können, weswegen jedes unabhängige „Slice-Segment“ stets den Beginn eines neuen übergeordneten „Slices“ markiert. Weil es sich bei dem übergeordneten „Slice“ außerdem um ein reguläres „Slice“ handelt, dessen Grenzen Entropie- und Prädiktionscodierung „brechen“, schränkt das übergeordnete „Slice“ zudem die Verfügbarkeit der Daten benachbarter Blöcke bei der Codierung der „Slice-Segmente“ ein.

Diese beiden Eigenschaften eines übergeordneten „Slices“ führen dazu, dass die unabhängigen „Slice-Segmente“ des zweiten Ausführungsbeispiels im Unterschied zum unabhängigen „Slice“ des ersten Ausführungsbeispiels eine Anker-Funktion haben, die die Abfolge der „Slice-Segmente“ in unabhängig voneinander codierbare Teilfolgen zerlegt, unabhängig von deren geometrischer Anordnung.

Im Ergebnis ist festzustellen, dass sich die „Slices“ des ersten Ausführungsbeispiels und die „Slice-Segmente“ des zweiten Ausführungsbeispiels hinsichtlich ihrer Struktur- und Codiereigenschaften

voneinander unterscheiden, weswegen es für die Auslegung durchaus auf das zweite Ausführungsbeispiel ankommt.

7.2 Die Beklagte argumentiert außerdem, die Anspruchsauslegung müsse sich auf die Begriffe im Patentanspruch beziehen und nicht auf solche, die darin keinen Niederschlag gefunden hätten. Soweit beansprucht sei es für das Verständnis des Patentanspruchs 1 völlig einerlei, ob die „Slices“ Bestandteil eines übergeordneten Strukturelements bzw. regulären „Slices“ sein können oder nicht. Mit übergeordneten Strukturelementen befasse sich der Patentanspruch 1 gar nicht. Vielmehr gebe es im Patentanspruch 1 nur zweierlei „Slices“ (aber nichts Übergeordnetes): die unabhängigen „Slices“ der Merkmale **M1.3.2a/b** einerseits und die abhängigen „Slices“ der Merkmale **M1.3.1a/b**. Diese „Slices“ (jedes für sich) seien wiederum einem Teilsatz der Codierblöcke zugeordnet (Merkmal **M1.5**), aber dem gesamten Patentanspruch 1 sei nichts dafür zu entnehmen, dass es etwa um jeweils ein unabhängiges „Slice“ und ein oder mehrere abhängige „Slices“ herum noch ein weiteres Strukturelement geben müsse.

Der Einwand der Beklagten greift nicht durch. So stellt der Bundesgerichtshof in der Entscheidung „Spannschraube“ fest, dass bei der Auslegung eines europäischen Patents nicht am Wortlaut zu haften, sondern auf den technischen Gesamtzusammenhang abzustellen ist, den der Inhalt der Patentschrift dem Fachmann vermittelt. Weichen die dort gebrauchten Begriffe vom allgemeinen (technischen) Sprachgebrauch ab, ist letztlich nur der aus der Patentschrift sich ergebende Begriffsinhalt maßgebend (*vgl. BGH a.a.O. – Spannschraube, Leitsätze*). Weiterhin führt der Bundesgerichtshof aus: „Begriffe in den Patentansprüchen und in der Patentbeschreibung sind deshalb so zu deuten, wie sie der angesprochene Durchschnittsfachmann nach dem Gesamtinhalt der Patentschrift unter Berücksichtigung von Aufgabe und Lösung der Erfindung versteht“ (*vgl. BGH a.a.O. – Spannschraube, III. 3. a*). Bei Auslegung des anspruchsgemäßen „Slices“ als „Slice-Segment“ bestimmt sich demnach der Begriff „Slice“ nicht allein aus dem Offenbarungsgehalt der Patentansprüche, sondern ergänzend aus dem Offenbarungsgehalt der Patentschrift, d. h.

insbesondere aus dem zweiten Ausführungsbeispiel der Patentbeschreibung, das neben einem unabhängigen „Slice-Segment“ und einer Folge von abhängigen „Slice-Segmenten“ noch ein übergeordnetes, reguläres „Slice“ vorsieht, welches die Codiereigenschaften der „Slice-Segmente“ beschränkt.

7.3 Weiterhin ist die Beklagte unter Bezugnahme auf Figur 20 und Absatz [0099] der Streitpatentschrift der Auffassung, im ersten Ausführungsbeispiel des Streitpatents sei eine Gruppe aus „Slices“ zu erkennen, die hinsichtlich der Prädiktions- und Entropiecodierung dieselben Eigenschaften habe wie ein „Slice“ gemäß dem zweiten Ausführungsbeispiel. Demnach sei es möglich, dass ein „unabhängiges Slice“ von „abhängigen Slices“ gefolgt werde und das „unabhängige Slice“ die Funktion eines Ankers habe, um die Abhängigkeiten zu den vorhergehenden „Slices“ zu brechen.

(a) Der Beklagten ist zwar darin zuzustimmen, dass die in Figur 20 schraffierten Bildbereiche, die jeweils zwei abhängige „Slices“ (DS) unterhalb eines herkömmlichen, unabhängigen „Slices“ (RS) darstellen, unabhängig voneinander codier- und decodierbar sind. Die unabhängige Codier- und Decodierbarkeit der beiden schraffierten Bereiche aus Figur 20 folgt aber aus der geometrischen Anordnung der regulären „Slices“, die sich über die gesamte Bildbreite erstrecken, und nicht etwa aus einer Gruppenbildung aus einem unabhängigen und einigen nachfolgenden abhängigen „Slices“. So kann es für die Blöcke des abhängigen „Slices“ der fünften Zeile wegen des darüber liegenden regulären „Slices“ prinzipiell keinerlei Nachbarblöcke außerhalb des unteren schraffierten Bereichs geben, auf die für die Prädiktions- und Entropiecodierung zugegriffen werden könnte.

(b) Dass dem unabhängigen „Slice“ des ersten Ausführungsbeispiels nicht zwangsläufig eine Anker-Funktion zukommt, findet seine Stütze z. B. in den Absätzen [0100] bis [0103] der Streitpatentschrift, wo ein Verfahren zur Ermittlung der Verfügbarkeit eines (benachbarten) Codierblocks für die Codierung eines aktuellen Codierblocks beschrieben wird. Die Verfügbarkeit hängt im Wesentlichen nur davon ab, ob die beiden Blöcke im gleichen „Slice“ liegen und

wenn nicht, ob das aktuelle „Slice“ ein abhängiges oder unabhängiges „Slice“ ist.

(c) Entsprechendes folgt aus den Absätzen [0133] bis [0138] und Figur 25 der Streitpatentschrift, in denen der Begriff einer „neighborhood“ erläutert wird. Die „neighborhood“ betrifft die Blöcke aus der Nachbarschaftsumgebung eines aktuellen „Slices“, deren Daten über die Grenzen des aktuellen „Slices“ hinweg für Prädiktion und Entropiecodierung herangezogen werden können. Entscheidend hierbei ist lediglich, ob es sich beim aktuellen „Slice“ um ein abhängiges oder unabhängiges „Slice“ handelt. Liegt z. B. ein abhängiges „Slice“ vor, so erfolgt die Intra-Prädiktion über die das aktuelle „Slice“ umgebenden Grenzen hinweg (vgl. Streitpatentschrift, Abs. [0137], siehe „The predictor 28 behaves the same: for slices being of the first mode 20, predictor 28 uses spatial prediction across the slice boundary encircling the current slice.“). Liegt hingegen ein unabhängiges „Slice“ vor, hängen Entropiekontexte und Prädiktionscodierung alleinig von Blöcken innerhalb des aktuellen „Slices“ ab (vgl. Streitpatentschrift, Abs. [0138], siehe „For slices, however, having the second mode 22 ... entropy decoder 24 und predictor 28 restrict the derivation of entropy contexts and predictive decoding to depend on attributes relating to portions lying within the current slice only.“). Ob ein Block der „neighborhood“ für die Codierung verfügbar ist oder nicht, ist also unabhängig davon, ob sich zwischen dem aktuellen „Slice“ und dem „Slice“ aus der Nachbarschaftsumgebung noch ein unabhängiges „Slice“ befindet. Dementsprechend ist auch an keiner Stelle der Absätze [0133] bis [0138] die Rede von einer Gruppe bestehend aus einem unabhängigen „Slice“ und nachfolgenden abhängigen „Slices“, an deren Außengrenze aufgrund einer Anker-Funktion Prädiktions- und Entropiecodierung gebrochen würden.

7.4 Die Beklagte meint außerdem unter Bezugnahme auf Absatz [0099] des Streitpatents, dass bereits aus der Header-Struktur der „Slices“ des ersten Ausführungsbeispiels des Streitpatents ersichtlich sei, dass ein unabhängiges „Slice“ und die nachfolgenden abhängigen „Slices“ eine Gruppe bilden, die man sinnvollerweise als „Slice“ bezeichnen könne. Auch unter diesem Gesichtspunkt sei ein „übergeordnetes Strukturelement“ bereits im ersten Ausführungsbeispiel angelegt.

Der Beklagten ist zwar darin zuzustimmen, dass die abhängigen „Slices“ auf die Werte von Syntaxelementen eines vorhergehenden unabhängigen „Slices“ referenzieren. Jedoch bedeutet dies nicht zwangsläufig, dass eine Beschränkung der Codiereigenschaften der abhängigen „Slices“ über die Außengrenzen eines „übergeordneten Strukturelements“ hinweg vorgesehen ist, mithin also eine Anker-Eigenschaft vorliegt, die wiederum zu einer unabhängigen Decodierbarkeit des „übergeordneten Strukturelements“ führt.

7.5 Die Ansicht der Beklagten, die in den Absätzen [0100] bis [0103] der Streitpatentschrift angegebenen Kriterien für die Entscheidung, wann ein Block für die Codierung eines anderen Blocks zur Verfügung steht, bezöge sich nur auf eine Überprüfung innerhalb einzelner Gruppen, ist unzutreffend. Denn weder in der genannten noch irgendeiner anderen Textstelle der Streitpatentschrift ist die Rede davon, dass sich die genannte Überprüfung der Verfügbarkeit eines Blocks auf eine Überprüfung innerhalb einzelner Gruppen bezieht, die die Codiereigenschaften an ihren jeweiligen Grenzen beschränken.

7.6 Weiterhin vermag die Argumentation der Beklagten nicht zu überzeugen, wonach es in der streitpatentgemäßen Codierung ohnehin nicht zu der geometrischen Anordnung komme, bei der über die Gruppengrenzen hinweg Daten für die Prädiktion und Entropiecodierung verwendet werden können. So zeigt Figur 24 i. V. m. Absatz [0125] der Streitpatentschrift eine Anordnung von „Slices“, in der jede Zeile eines zu codierenden/decodierenden Bildes über zwei „Slices“ verfügt, wobei jedes „Slice“ nach einem von zwei Modi codiert/decodiert wird. Wird z. B. bei „Slice“ #3 von einem unabhängigen und bei „Slice“ #4 von einem abhängigen „Slice“ ausgegangen, so wird bei der Codierung/Decodierung des ersten Blocks des abhängigen „Slices“ #4 nicht nur auf CABAC Wahrscheinlichkeiten und Daten zur Intra-Prädiktion von „Slice“ #3 zugegriffen, sondern gleichermaßen auf entsprechende Daten aus dem darüber angeordneten „Slice“ #2, also über die Grenzen des abhängigen „Slices“ #3 hinweg.

8. Der jeweilige Gegenstand der erteilten Patentansprüche 1 und 15 kann – soweit er „Slice-Segmente“ betrifft – die Priorität der beiden Prioritätsdokumente nicht wirksam in Anspruch nehmen. Entsprechendes gilt für die Patentansprüche 1 und 14 gemäß Hilfsantrag I, III und IV, die Patentansprüche 1 und 15 gemäß Hilfsantrag II sowie die Patentansprüche 1 und 10 gemäß Hilfsantrag V.

8.1 Aus den Prioritätsunterlagen zum Streitpatent (Anlagen **1NK6** und **1NK7**) konnte der Fachmann nicht die Lehre entnehmen, dass herkömmliche bzw. reguläre „Slices“ entsprechend Absatz [0156] der Streitpatentschrift in unabhängige und abhängige „Slice-Segmente“ unterteilt werden, um diese Segmente als kleinste Einheiten für die Video-Paketierung zu verwenden. Vielmehr hätte der Fachmann ausgehend von den Prioritätsunterlagen aufgrund des dort fehlenden Ausführungsbeispiels betreffend die Einführung von „Slice-Segmenten“ keine andere Auslegung für den Begriff „Slice“ vorgenommen, als es bis zum jeweiligen Prioritätstag üblich war. Demnach wäre er beim Begriff „Slice“ zunächst von einem herkömmlichen bzw. unabhängigen „Slice“ ausgegangen, neben dem dann zusätzlich abhängige „Slices“ vorgesehen sind, d. s. Strukturelemente, die Prädiktions- und Entropiecodierung nicht „brechen“.

So findet sich in den beiden Prioritätsdokumenten, die in Form und Inhalt deutlich vom Streitpatent abweichen, keinerlei Entsprechung zu den Absätzen [0127] bis [0171] und den Figuren 24 bis 30 des Streitpatents. Insbesondere wird aber die in den Absätzen [0154] ff der Streitpatentschrift beschriebene Ausführungsform, in der auch der Begriff „Slice-Segment“ eingeführt wird, in den Prioritätsunterlagen nicht erwähnt. Letztlich führt dieser Umstand dazu, dass die durch die Absätze [0154] ff der Patentbeschreibung gestützte Auslegung des Begriffes „Slice“ über das in den Prioritätsdokumenten verwendete Verständnis dieses Begriffes hinausgeht. Der Gegenstand der erteilten Patentansprüche 1 und 15, wie er sich aus der Auslegung im Lichte der Absätze [0154] ff der Streitpatentschrift ergibt, geht daher über die Offenbarung der Prioritätsunterlagen hinaus. Insoweit ist der Gegenstand der mit den erteilten Patentansprüchen 1 und 15 beanspruchten Erfindung nicht mit der jeweiligen Erfindung in den Prioritätsunterlagen identisch. Folglich kann der jeweilige

Gegenstand der erteilten Patentansprüche 1 und 15 – soweit er „Slice-Segmente“ betrifft - die Priorität der beiden Prioritätsdokumente nicht wirksam beanspruchen und hat den Zeitrang der Anmeldung vom 15. April 2013. Entsprechendes gilt für die beiden angegriffenen Patentansprüche in der jeweiligen Fassung nach Hilfsantrag I bis V.

8.2 Die Ausführungen der Beklagten zum Offenbarungsgehalt der Prioritätsunterlagen halten einer näheren Überprüfung nicht stand.

8.2.1 Dem Einwand der Beklagten, die Offenbarung der Prioritätsdokumente entspreche nicht nur den unabhängigen/abhängigen „Slices“, sondern auch den in der Nomenklatur des Streitpatents umbenannten unabhängigen/abhängigen „Slice-Segmenten“, kann nicht gefolgt werden. Die von der Beklagten in diesem Zusammenhang genannten Textstellen der Prioritätsdokumente belegen allenfalls, dass sich der Wortlaut des erteilten Anspruchs 1 in den Prioritätsdokumenten wiederfinden lässt (vgl. **1NK6**, Seite 26, 27; **1NK7**, Seite 70, 71) und dass den herkömmlichen regulären bzw. unabhängigen „Slices“ neue abhängige „Slices“ zur Seite gestellt werden (vgl. **1NK6**, Seite 21, erster Absatz, siehe „Thus the upcoming HEVC standard should offer two general types of slices: independent (regular) or dependent. Therefore we introduce the new type of slice, a Dependent Slice.“), um die Beschränkung von Prädiktions- und Entropiecodierung zu überwinden.

8.2.2 Die Beklagte argumentiert, das Prioritätsdokument **1NK6** offenbare, dass ein unabhängiges „Slice“ von abhängigen „Slices“ gefolgt werden könne, und dass das unabhängige „Slice“ die Funktion eines Ankers habe, um die Abhängigkeiten zu den vorhergehenden „Slices“ zu brechen, was bedeute, dass ein unabhängiges „Slice“ und die nachfolgenden abhängigen „Slices“ zusammen eine Gruppe bildeten, die als Ganzes unabhängig von anderen „Slices“ decodierbar sei.

Die Argumentation der Beklagten ist nicht überzeugend. So offenbaren die Prioritätsdokumente **1NK6** und **1NK7** allenfalls die Einführung abhängiger

„Slices“ als Ergänzung zu den herkömmlichen, unabhängigen „Slices“ gemäß dem ersten Ausführungsbeispiel des Streitpatents. Insbesondere geht die von der Beklagten angesprochene Gruppe aus der Druckschrift **1NK6** nicht hervor. Die Druckschrift **1NK6** lehrt lediglich, dass in einer ganz bestimmten geometrischen Konstellation - nämlich derjenigen aus Figur 13 - ein unabhängiges „Slice“ als Anker zur Brechung der Abhängigkeit zu den früheren „Slices“ fungieren kann. Weder ist dort offenbart, dass eine Gruppe aus einem unabhängigen „Slice“ und darauffolgenden abhängigen „Slices“ für beliebige geometrische Anordnungen eine logische Einheit bildet, noch, dass diese Gruppe die Eigenschaften eines herkömmlichen „Slices“ hat, das Prädiktions- und Entropiecodierung an seinen Grenzen immer bricht. Die Argumentation der Beklagten stützt sich v. a. auf eine Offenbarungsstelle in der Druckschrift **1NK6**, die sich mit dem „WPP Low Delay Mode“ befasst (vgl. **1NK6**, Seite 24, Mitte bis Seite 26, oben). In dem gezeigten Beispiel stimmen die „Slices“ mit den WPP-Zeilen überein bzw. erstrecken sich über diese (vgl. **1NK6**, Fig. 13). Daher ergibt sich aus den WPP-Zeilen keine Gruppe, in der ein unabhängiges und wenigstens ein abhängiges „Slice“ gemeinsam eine WPP-Zeile bilden, die eine Prüfung gemäß Merkmalsgruppe **M1.6** ermöglicht. An dieser Feststellung vermögen auch die beiden grau unterlegten und gepunktet umrandeten Rechtecke der Figur 13 nichts zu ändern. Diese lassen allenfalls den Schluss zu, dass jeweils zwei abhängige „Slices“ (DS) mit einem unabhängigen „Slice“ (RS) in irgendeiner Beziehung stehen. Ein übergeordnetes Strukturelement in Gestalt eines herkömmlichen (regulären) „Slices“ lässt sich aus der Darstellung aber nicht unmittelbar und eindeutig entnehmen. Da es auch sonst keine weitere Offenbarungsstelle in der Druckschrift **1NK6** gibt, die eine beliebige Zusammensetzung von WPP-Zeilen zu entnehmen ist, zeigt die Druckschrift **1NK6** keine Gruppe, die dem herkömmlichen (regulären) „Slice“ mit „Slice-Segmenten“ aus dem zweiten Ausführungsbeispiel des Streitpatents entspricht.

Entsprechendes gilt für das Dokument **1NK7**.

8.2.3 Der Einwand der Beklagten, dass das Prioritätsdokument **1NK6** nicht nur auf „Slices“ als Einheit abstelle, sondern auch noch kleinere Einheiten in Form

von Teilen eines „Slices“ („fragment of a slice“) kenne, bei denen es sich um die unabhängigen bzw. abhängigen „Slice-Segmente“ des Streitpatents handle, vermag nicht zu überzeugen.

So geht aus der Zusammenfassung der Druckschrift **1NK6** allenfalls hervor, dass ein abhängiges „Slice“ ein Fragment eines „Slices“ enthalten kann („may contain“), nicht aber, dass ein abhängiges „Slice“ mit einem Fragment eines „Slices“ gleichzusetzen ist. Ein solches Fragment ist demnach zwar als Teil eines abhängigen „Slices“ anzusehen, jedoch stellt es kein Strukturelement dar, für das zwei verschiedene Verarbeitungsmodi i. S. d. Streitpatents vorgesehen sind. Vielmehr offenbart die Druckschrift **1NK6** verschiedene Verarbeitungsmodi ausschließlich für „Slices“ und nicht etwa für Fragmente.

Außerdem ist dem Dokument **1NK6** zu entnehmen (Seite 4, vorletzter Absatz bis Seite 5, oben; Fig. 4), dass ein Fragment denjenigen Teilen eines „Slices“ entspricht, die einem bestimmten vertikalen „Tile“ zugeordnet sind, wobei alle Fragmente zusammengenommen ein „Slice“ bilden. Auch in dieser Hinsicht unterscheidet sich das Fragment der Druckschrift **1NK6** also von einem streitpatentgemäßen „Slice-Segment“, welches bei der Codierung eines Bildes ganz ohne Verwendung von „Tiles“ auskommen kann (Streitpatentschrift, Fig. 27).

II.

Das Streitpatent ist in der erteilten Fassung nicht rechtsbeständig, weil die jeweiligen Gegenstände des Patentanspruchs 1 und des nebengeordneten Patentanspruchs 15 nicht patentfähig sind.

1. Der Lehre des erteilten Patentanspruchs 1 fehlt es mit Rücksicht auf den der Druckschrift **2NK4** entnehmbaren Stand der Technik an der für die Patentfähigkeit erforderlichen Neuheit, wenn die anspruchsgemäßen „Slices“ als

„Slice-Segmente“ ausgelegt werden.

1.1 Bei der Druckschrift **2NK4** handelt es sich um die neunte Version des Entwurfs für den Text des zum Veröffentlichungszeitpunkt in Entwicklung befindlichen H.265/HEVC-Standards. Sie hat u. a. die Einführung von „Slice-Segmenten“ zum Gegenstand. Wie oben ausgeführt, kann das Streitpatent für den Gegenstand des Patentanspruchs 1 keine der Prioritäten wirksam in Anspruch nehmen, soweit die Auslegung verwendet wird, wonach ein „Slice“ auch ein „Slice-Segment“ sein kann. Daher gehört der Standardisierungsbeitrag **2NK4** für das Streitpatent zum Stand der Technik gemäß Art. 54 Abs. 2 EPÜ.

1.2 Die Druckschrift **2NK4** zeigt alle Merkmale des erteilten Patentanspruchs 1.

Sie betrifft die Codierung und Decodierung von Videodaten, also auch die Rekonstruktion eines Bildes aus einem Datenstrom durch einen Codec (Seite 3, Abschnitt 1, siehe „This document specifies High efficiency video coding.“). Die Druckschrift **2NK4** offenbart in den Definitionen 3.125 bis 3.127 der Seite 10 „Slice-Segmente“, in die jedes Bild partitioniert werden kann. Die bekannten „Slice-Segmente“ sind demnach zu übergeordneten (regulären) „Slices“ zusammengesetzt. Im Datenstrom ist das Bild in Einheiten von „Slice-Segmenten“ codiert (siehe „3.127 slice segment: An integer number of *coding tree units* ordered consecutively in the *tile scan* and contained in a single *NAL unit*; the division of each *picture* into slice segments is a *partitioning*.” – Merkmal **M1.1**).

In Abschnitt 6.3 der Druckschrift **2NK4** wird beschrieben, dass Bilder in „Slices“, „Slice-Segmente“ und „Tiles“ (Kacheln) partitioniert werden. Ein „Slice“ besteht dabei aus einer Folge von „Slice-Segmenten“, die mit einem unabhängigen „Slice-Segment“ beginnt und nach diesem – falls vorhanden – abhängige „Slice-Segmente“ angeordnet sind. Ein „Slice-Segment“ setzt sich wiederum aus Codierungsbaumeinheiten CTUs bzw. Codierblöcken zusammen. In Figur 6-4 werden die Aufteilung eines Bildes in „Slice-Segmente“ sowie die Anordnung der

Codierblöcke in Zeilen und Spalten dargestellt. Definition 3.127 der Seite 10 in Verbindung mit Figur 6-4 zeigt darüber hinaus, dass die „Slice-Segmente“ aus fortlaufenden Teilsätzen von Codierblöcken gebildet werden, wobei die Teilsätze einander entlang einer Rasterabtastrihenfolge folgen (siehe „3.127 slice segment: An integer number of *coding tree units* ordered consecutively in the *tile scan* ...” – Merkmale **M1.4**, **M1.5**). Die Teilsätze legen damit eine Segment-Reihenfolge fest, mit der sie vom Decodierer aus dem Datenstrom decodiert werden (Merkmal **M1.2**).

In Abschnitt 7.4.8.1 wird auf das Syntaxelement „dependent_slice_segment_flag“ im Header eines „Slice-Segments“ hingewiesen, welches anzeigt, ob es sich beim aktuellen „Slice-Segment“ um ein abhängiges oder unabhängiges „Slice-Segment“ handelt. Der Wert dieses Syntaxelements (0 bzw. 1) wirkt sich darauf aus, in welchem Modus das aktuelle „Slice-Segment“ decodiert wird (Merkmal **M1.3**).

Entsprechend der Definition 3.61 auf Seite 6 der Druckschrift **2NK4** wird eine Intra-Prädiktion bzw. „voraussagende Decodierung“ als eine Prädiktion beschrieben, die ausschließlich von Datenelementen (z.B. Abtastwerten) desselben decodierten „Slices“ abgeleitet wird. Liegt zudem mit dem aktuellen „Slice-Segment“ ein abhängiges „Slice-Segment“ vor (erster Modus), so geht diesem laut Definition 3.125 auf Seite 10 der Druckschrift **2NK4** wenigstens ein weiteres „Slice-Segment“ in demselben „Slice“ voraus. Dementsprechend stehen dann auch Datenelemente aus demselben „Slice“ für eine Prädiktion zur Verfügung. Weiterhin ergibt sich u. a. durch die Umstellung des Wertes der Variablen SliceAddrRS auf den Wert von SliceAddrRS für das vorhergehende „Slice-Segment“ infolge des Syntaxelements „dependent_slice_segment_flag“ (Seite 79, siehe „dependent_slice_segment_flag“), dass auf Prädiktionsblöcke (vgl. „neighbouring blocks“) nicht nur innerhalb von abhängigen „Slice-Segmenten“, sondern auch über deren Grenzen hinweg zugegriffen werden kann (vgl. Seite 24, Abschnitt 6.4.2 i. V. m. Seite 23, Abschnitt 6.4.1 – Merkmal **M1.3.1b**).

Figur 9-3 der Seite 171 für die CABAC-Initialisierung zeigt, dass bei der Decodierung der ersten CTU eines abhängigen „Slice-Segments“ (erster Modus) Kontextvariablen mit den in den Tabellen „TableStatIdxDS“ und „TableMPSValDS“ hinterlegten Werten synchronisiert werden. Aus der Figur 9-4 der Seite 179 geht zudem hervor, dass die in den genannten Tabellen enthaltenen Kontextvariablen stets nach dem Decodieren des Endes des vorhergehenden „Slice-Segments“ gespeichert werden (siehe „end_of_slice_segment_flag“), d. h. Kontexte werden über die Grenzen des abhängigen „Slice-Segments“ hinweg abgeleitet. Entsprechendes gilt für die Variable „pStatIdx“ bzw. „valMPS“, die jeder Kontextvariablen zugeordnet bzw. in dieser enthalten ist (Seite 178, Abschnitt 9.2.1.2; Seite 179, Abschnitt 9.2.1.3) und die einem Index für die Wahrscheinlichkeit von Zuständen bzw. dem Wert des wahrscheinlichsten Symbols entspricht (Seite 171, Abschnitt 9.2.1.1, siehe „NOTE 1 – The variable pStatIdx corresponds to a probability state index and the variable valMPS corresponds to the value of the most probable symbol as further described in subclause 9.2.3.2.“).

Im Rahmen einer Erläuterung der arithmetischen Decodierung, die dem CABAC-Verfahren zugrunde liegt, wird zudem auf die Bedeutung des wahrscheinlichsten Symbols („MPS“) und des unwahrscheinlichsten Symbols („LPS“) hingewiesen (Seite 194, erster Absatz, siehe „NOTE“). Demnach wird ein Kontext durch die Symbolwahrscheinlichkeit p_{LPS} des unwahrscheinlichsten Symbols „LPS“ und des Werts „valMPS“ des wahrscheinlichsten Symbols „MPS“ beschrieben (siehe „Given this terminology, each context is specified by the probability p_{LPS} of the LPS and the value of MPS (valMPS), which is either 0 or 1.“). Dies bedeutet aber gleichzeitig, dass bei sich änderndem Kontext von CTU zu CTU die Symbolwahrscheinlichkeiten p_{LPS} ständig angepasst werden müssen. Dies gilt dann auch für die auf Seite 194 genannte Zustandsmaschine („finite-state machine“), die der Abschätzung von Symbolwahrscheinlichkeiten dient.

Der Fachmann wird erkennen, dass bei Decodierung der ersten CTU eines abhängigen „Slice-Segments“ auf den gespeicherten Kontext des jeweils vorangegangenen „Slice-Segments“ zurückgegriffen werden muss und die

Symbolwahrscheinlichkeiten dieses Kontextes bestimmt werden müssen, ohne die eine Entropiedecodierung der Syntaxelemente der ersten CTU des aktuellen „Slice-Segments“ kaum möglich wäre, da ja hierfür noch keine neuen Symbolwahrscheinlichkeiten zur Verfügung stünden. Dass die Symbolwahrscheinlichkeiten des zuvor decodierten „Slice-Segments“ zu diesem Zweck zumindest zwischengespeichert werden müssen, ist selbstverständlich (Merkmal **M1.3.1a**).

Weiterhin ergibt sich unter Berücksichtigung der Definitionen 3.61 und 3.125 aus Druckschrift **2NK4**, dass, wenn das aktuelle „Slice-Segment“ ein unabhängiges „Slice-Segment“ ist, diesem kein weiteres „Slice-Segment“ in demselben „Slice“ vorausgeht. Damit stehen auch nur solche Datenelemente aus demselben „Slice“ für die Prädiktion zur Verfügung, die zu dem unabhängigen „Slice-Segment“ selbst gehören und keine Datenelemente, die über die Grenzen des „Slice-Segments“ hinausgehen (Merkmal **M1.3.2b**).

Bei Decodierung der ersten CTU eines unabhängigen „Slice-Segments“ werden die Kontextvariablen initialisiert, und zwar unabhängig von „TableStatIdxDS“ und „TableMPSValDS“ und somit auch unabhängig von zuvor decodierten „Slice-Segmenten“ (Fig. 9-3). Für die weiteren CTUs werden die Symbolwahrscheinlichkeiten fortlaufend aktualisiert, allerdings fließen keine Datenelemente aus anderen „Slice-Segmenten“ in die Kontextvariablen ein (Merkmal **M1.3.2a**).

In Abschnitt 9.2.1.1 der Druckschrift **2NK4** wird beschrieben, wie die Kontextvariablen für die CABAC-Entropiedecodierung initialisiert werden. Den Kontextvariablen sind die Variablen „valMPS“ und „pStatIdx“ zugeordnet, die mit der Bestimmung von Symbolwahrscheinlichkeiten verknüpft sind (siehe oben). Figur 9-4 zeigt, wie und wann die Kontextvariablen gespeichert werden müssen. Dabei wird zunächst überprüft, ob die aktuelle CTU die zweite CTU einer Zeile darstellt und ob die Entropiesynchronisierung grundsätzlich aktiv ist. Ist dies der Fall, werden die Kontextvariablen mitsamt den Variablen „valMPS“ und

„pStateldx“ in den Tabellen „TableStateldxWPP“ und „TableMPSValWPP“ gespeichert. Dass neben den Kontextvariablen auch die bis nach der Decodierung der zweiten CTU ermittelten Symbolwahrscheinlichkeiten gespeichert werden müssen, ist selbstverständlich, da ansonsten zur Decodierung einer nachfolgenden CTU keine aktualisierten Symbolwahrscheinlichkeiten zur Verfügung stünden (siehe oben - Merkmal **M1.6**).

Figur 9-3 zeigt eine Fallunterscheidung, bei der überprüft wird, ob es sich bei der aktuellen CTU um die erste CTU in einer Zeile handelt („First CTU in row ...?“). Hierbei ist klar, dass die Fallunterscheidung u. a. dann angewendet wird, wenn die aktuelle CTU die erste CTU eines abhängigen „Slice-Segments“ ist (Merkmal **M1.6.1**). Laut der Figur 9-3 wurde ein unabhängiges „Slice-Segment“ davor ausgeschlossen.

Demnach werden für den Fall, dass die aktuelle CTU die erste CTU in einer Zeile und zugleich die erste CTU eines abhängigen „Slice-Segments“ ist, die Kontextvariablen mit den zuvor gespeicherten Werten synchronisiert („Synchronization of context variables with TableStateldxWPP and TableMPSValWPP“), d. h. mit den Kontextvariablen aus der zweiten CTU in der Zeile darüber. Zugleich müssen die zur Decodierung dieser ersten CTU notwendigen Symbolwahrscheinlichkeiten auf die Symbolwahrscheinlichkeiten initialisiert werden, die sich bis nach der Decodierung der zweiten CTU aus der darüber liegenden Zeile ergeben haben (wobei diese zweite CTU zum davor decodierten „Slice-Segment“ gehören kann) (Merkmal **M1.6.2**).

Für den Fall, dass es sich bei der aktuellen CTU nicht um die erste CTU einer Zeile handelt, diese aber die erste CTU eines abhängigen „Slice-Segments“ ist, werden die Kontextvariablen mit den in den Tabellen „TableStateldxDS“ und „TableMPSValDS“ gespeicherten Werten synchronisiert (Fig. 9-3). Diese Tabellen enthalten die Werte der Kontextvariablen, wie sie am Ende des letzten abhängigen „Slice-Segments“ gespeichert wurden (Fig. 9-4). Dementsprechend werden die Symbolwahrscheinlichkeiten zur Decodierung der aktuellen CTU auf

die Symbolwahrscheinlichkeiten initialisiert, wie sie bis zum Ende des zuvor decodierten „Slice-Segments“ ermittelt worden sind (Merkmal **M1.6.3**).

1.3 Damit sind sämtliche Merkmale des Gegenstandes des erteilten Patentanspruchs 1 aus der Druckschrift **2NK4** bekannt. Dem Gegenstand des Patentanspruchs 1 gemäß Hauptantrag fehlt es daher an der für die Patentfähigkeit erforderlichen Neuheit.

2. Der auf „Ein Verfahren [zum] Rekonstruieren eines Bildes aus einem Datenstrom“ gerichtete, nebengeordnete Patentanspruch 15 ist nicht günstiger als Patentanspruch 1 zu beurteilen, da er inhaltlich nicht über diesen hinausgeht und somit nichts enthält, was eine Patentfähigkeit rechtfertigen würde.

3. Weder der erteilte Vorrichtungsanspruch 1 noch der nebengeordnete Verfahrensanspruch 15 des Streitpatents hat daher Bestand. Das Streitpatent ist im Umfang der angegriffenen Patentansprüche 1 und 15 für nichtig zu erklären.

III.

Auch die Hilfsanträge bleiben ohne Erfolg.

Der Nichtigkeitsgrund der mangelnden Patentfähigkeit besteht in den Fassungen der Hilfsanträge I bis V unverändert fort. Im Hinblick darauf kann dahingestellt bleiben, ob die Fassungen jeweils zulässig sind.

1. Hilfsantrag I kann keinen Erfolg haben, da die darin vorgenommenen Änderungen eine Patentfähigkeit nicht begründen können.

1.1 Gemäß Hilfsantrag I ist der erteilte Patentanspruch 1 dadurch weiter beschränkt, dass nach Merkmal **M1.6.3** das folgende Merkmal eingefügt ist:

H1M1.8 „wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in a preceding slice decoded in the second mode.“

Auf Deutsch:

H1M1.8 „wobei der Decodierer dazu ausgebildet ist, gemäß dem ersten (20) der zumindest zwei Modi, für das aktuelle Stück den Wert von nicht vorhandenen Stück-Header-Syntaxelementen als gleich dem Wert von Stück-Header-Syntaxelementen in einem in dem zweiten Modus decodierten vorhergehenden Stück abzuleiten.“

wobei der abhängige Patentanspruch 8 gestrichen wurde. Entsprechende Änderungen wurden in den erteilten unabhängigen Patentansprüchen 11, 15 und 16 (jetzt unabhängige Patentansprüche 10, 14 und 15 gemäß Hilfsantrag I) gemacht.

1.2 Mit Merkmal **H1M1.8** nimmt Hilfsantrag I die „Header-Struktur“ der „Slices“ in die unabhängigen Patentansprüche auf. Damit möchte die Beklagte in den unabhängigen Patentansprüchen neben der unabhängigen Decodierbarkeit auch den zweiten Aspekt berücksichtigen, wonach die „Slices“ Bestandteile übergeordneter Strukturelemente sein können. Merkmal **H1M1.8** besagt nichts anderes, als dass der Decodierer im ersten Modus (d. h. wenn ein abhängiges „Slice“ decodiert wird) für das aktuelle „Slice“ den Wert eines (dort) nichtvorhandenen Header-Syntaxelements aus dem Wert eines Header-Syntaxelements eines vorhergehenden „Slices“ ableitet, welches im zweiten Modus (d. h. als unabhängiges „Slice“) decodiert worden ist. Ein aktuelles „Slice“, das ein abhängiges „Slice“ darstellt, referenziert also ein unabhängiges „Slice“,

um auf den Wert eines Syntaxelements zu schließen.

1.3 Mit Rücksicht auf den aus der Druckschrift **2NK4** bekannten Stand der Technik fehlt es dem Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag I an der für die Patentfähigkeit erforderlichen Neuheit.

1.3.1 Der Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag I kann – soweit er „Slice-Segmente“ betrifft – die Priorität der beiden Prioritätsdokumente nicht wirksam beanspruchen. Die Druckschrift **2NK4** gehört somit zum bei der Prüfung der Patentfähigkeit zu berücksichtigenden Stand der Technik.

1.3.2 Merkmal **H1M1.8** ist auf Seite 5, Punkt 3.43 der Druckschrift **2NK4** offenbart. Dort wird Folgendes definiert:

„3.43 dependent slice segment: *A slice segment for which the values of some syntax elements of the slice segment header are inferred from the values for the preceding independent slice segment in decoding order.*“

Werte von Syntaxelementen, die im Header eines abhängigen „Slice-Segments“ nicht vorhanden sind, werden der Druckschrift **2NK4** zufolge also aus den Werten des vorhergehenden unabhängigen „Slice-Segments“ abgeleitet.

1.4 Unter Berücksichtigung der Ausführungen zum Hauptantrag gehen damit sämtliche Merkmale des Patentanspruchs 1 gemäß Hilfsantrag I aus der Druckschrift **2NK4** hervor. Der Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag I ist daher nicht neu und aus diesem Grund nicht patentfähig. Mit dem Patentanspruch 1 fällt der gesamte Hilfsantrag I.

2. Hilfsantrag II ist nicht günstiger zu beurteilen, da der Gegenstand seines Patentanspruchs 1 nicht patentfähig ist.

2.1 Gemäß Hilfsantrag II ist der erteilte Patentanspruch 1 dadurch weiter beschränkt, dass anstelle von Merkmal **M1.6** die folgenden Merkmale eingefügt sind:

H2M1.5a „wherein the decoder is configured to be responsive to a syntax element in a picture parameter set so as to operate according to one of two operating types.”

und

H2M1.6 „wherein, according to the first operating type, the decoder is configured to save symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to a second coding block (32) in a row in accordance with the raster scan order (36), and,”

Auf Deutsch:

H2M1.5a „wobei der Decodierer dazu ausgebildet ist, auf ein Syntaxelement in einem Picture Parameter Set anzusprechen, um gemäß einem von zwei Betriebstypen zu arbeiten,“

und

H2M1.6 „wobei der Decodierer gemäß dem ersten Betriebstyp dazu ausgebildet ist, Symbolwahrscheinlichkeiten, wie bei einer kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu einem zweiten Codierblock (32) in einer Zeile gemäß der Rasterabtastreihenfolge (36) erhalten, zu speichern, und“.

Außerdem wird nach Merkmal **M1.6.3** das folgende Merkmal eingefügt:

H2M1.7 „wherein, according to the second operating type, the decoder is configured to initialize the symbol probabilities for the context

adaptive entropy decoding of the current slice according to the first mode depending on symbol probabilities as obtained in context adaptive entropy decoding the previously decoded slice up to the end of the previously decoded slice independent of whether a first coding block of the continuous subset of coding blocks (32) associated with the current slice is a first coding block (32) in a row in accordance with the raster scan order or not."

Auf Deutsch:

H2M1.7 „wobei, gemäß dem zweiten Betriebstyp, der Decodierer dazu ausgebildet ist, die Symbolwahrscheinlichkeiten für die kontextadaptive Entropiedecodierung des aktuellen Stückes gemäß dem ersten Modus abhängig von Symbolwahrscheinlichkeiten, wie bei der kontextadaptiven Entropiedecodierung des zuvor decodierten Stückes bis zu dem Ende des zuvor decodierten Stückes erhalten, zu initialisieren, unabhängig davon, ob ein erster Codierblock des fortlaufenden Teilsatzes von Codierblöcken (32), der dem aktuellen Stück zugeordnet ist, ein erster Codierblock (32) in einer Zeile gemäß der Rasterabtastrihenfolge ist oder nicht.“

Entsprechende Änderungen wurden in den erteilten unabhängigen Patentansprüchen 11, 15 und 16 gemacht.

2.2 Mit den Merkmalen **H2M1.5a**, **H2M1.6** und **H2M1.7** nach Hilfsantrag II kann die Wavefront Parallel Processing-Funktionalität mittels eines im Picture Parameter Set übertragenen Syntaxelements ein- und ausgeschaltet werden. Bei dieser Funktionalität erfolgt die Initialisierung der CABAC-Wahrscheinlichkeiten eines abhängigen „Slices“, dessen erste CTU zugleich die erste CTU einer CTU-Zeile ist, bekanntlich auf die Symbolwahrscheinlichkeiten, die sich bis nach der Decodierung der zweiten CTU aus der darüber liegenden CTU-Zeile ergeben haben. Ist die WPP-Funktionalität eingeschaltet, erfolgt die Initialisierung der CABAC-Wahrscheinlichkeiten der abhängigen „Slices“ wie in den Merkmalen

M1.6, **M1.6.1**, **M1.6.2** und **M1.6.3** vorgesehen. Ist sie hingegen ausgeschaltet, erfolgt die Initialisierung der CABAC-Wahrscheinlichkeiten der abhängigen „Slices“ auf die Symbolwahrscheinlichkeiten, wie sie bis zum Ende des zuvor decodierten „Slices“ ermittelt worden sind – unabhängig davon, ob eine erste CTU eines abhängigen „Slices“ die erste CTU einer CTU-Zeile ist oder nicht.

2.3 Soweit die Auslegung verwendet wird, wonach ein „Slice“ auch ein „Slice-Segment“ sein kann, ist auch der Gegenstand des Patentanspruchs 1 gemäß Hilfsantrag II nicht neu gegenüber Druckschrift **2NK4**.

So können die Merkmale **H2M1.6'** und **H2M1.7** aus Figur 9-3 auf Seite 171 der Druckschrift **2NK4** abgeleitet werden. Dabei werden zwei Betriebstypen bzw. -zustände durch die beiden Werte des Syntaxelements „entropy_coding_sync_enabled_flag“ festgelegt. Liegt in der Figur 9-3 die erste CTU einer Zeile vor und ist entropy_coding_sync_enabled_flag gleich 1, dann erfolgt eine Synchronisierung der Kontextvariablen mit den in den Tabellen TableStatIdxWPP und TableMPSValWPP enthaltenen Daten, die gemäß dem Speicherprozess der Figur 9-4 auf Seite 179 der Druckschrift **2NK4** in Abhängigkeit von entropy_coding_sync_enabled_flag aktualisiert worden sind oder nicht. Ist die aktuelle CTU die erste CTU in einem abhängigen „Slice-Segment“, aber nicht die erste CTU einer Zeile, so erfolgt bei entropy_coding_sync_enabled_flag gleich 1 eine Synchronisation der Kontextvariablen mit den Daten in den Tabellen TableStatIdxDS und TableMPSValDS, die wiederum entsprechend Figur 9-4 aktualisiert worden sind. Demnach bedeutet entropy_coding_sync_enabled_flag gleich 1 in jedem Fall, dass die Synchronisierung bzw. Initialisierung der CABAC-Wahrscheinlichkeiten der abhängigen „Slice-Segmente“ wie in den Merkmalen **M1.6'** bis **M1.6.3** erfolgt.

Liegt hingegen entropy_coding_sync_enabled_flag gleich 0 vor, so werden die Kontextvariablen der ersten CTU eines abhängigen „Slice-Segments“ stets mit den Daten in TableStatIdxDS und TableMPSValDS synchronisiert, unabhängig davon, ob die CTU die erste CTU einer Zeile ist oder nicht. In diesem Fall werden also die CABAC-Wahrscheinlichkeiten der abhängigen „Slice-Segmente“ immer

auf die Symbolwahrscheinlichkeiten initialisiert, wie sie bis zum Ende des zuvor decodierten „Slice-Segments“ ermittelt worden sind (vgl. Fig. 9-3 iVm Fig. 9-4, siehe „end_of_slice_segment_flag“).

Die Bedeutung des Syntaxelements „entropy_coding_sync_enabled_flag“ wird im Übrigen auch auf Seite 71, achter Absatz der Druckschrift **2NK4** erläutert.

Die Merkmale **H2M1.6'** und **H2M1.7** gehen damit aus der Druckschrift **2NK4** hervor.

Was Merkmal **H2M1.5a** anbelangt, so offenbart Abschnitt 7.3.2.3 auf Seite 34, Zeile 5 der Druckschrift **2NK4** zudem, dass das Syntaxelement „entropy_coding_sync_enabled_flag“ Teil der Datenstruktur „pic_parameter_set_rbsp()“ ist, d. h. im Bildparametersatz („Picture Parameter Set“) enthalten ist. Damit wird auch Merkmal **H2M1.5a** durch die **2NK4** vorweggenommen.

2.4 Unter Berücksichtigung der Ausführungen zum Hauptantrag ist somit auch der Gegenstand des Patentanspruchs 1 in der Fassung des Hilfsantrags II nicht patentfähig, da dieser Gegenstand nicht neu ist. Mit dem Patentanspruch 1 fällt der gesamte Hilfsantrag II.

3. Hilfsantrag III kann keinen Erfolg haben, weil der Gegenstand seines Patentanspruchs 1 durch die Druckschrift **2NK4** vorweggenommen ist.

3.1 Hilfsantrag III ist eine Kombination der Hilfsanträge I und II.

3.2 Es gelten die Ausführungen zu Hilfsantrag I und II. Demnach ist die Lehre des Patentanspruchs 1 gemäß Hilfsantrag III nicht neu und deshalb nicht patentfähig. Mit dem Patentanspruch 1 fällt der gesamte Hilfsantrag III.

4. Die **Hilfsanträge IV** und **V** bleiben ohne Erfolg, weil ihr jeweiliger Patentanspruch 1 nichts Zusätzliches enthält, was eine Patentfähigkeit trägt.

4.1 Patentanspruch 1 gemäß Hilfsantrag IV unterscheidet sich von Patentanspruch 1 gemäß Hilfsantrag III dadurch, dass Merkmal **H1M1.8** durch Merkmal **H4M1.8** ersetzt wird:

H4M1.8 „wherein the decoder is configured to, in accordance with the first (20) of the at least two modes, infer for the current slice the value of slice header syntax elements not present to be equal to the value of slice header syntax elements in the *immediately preceding* slice decoded in the second mode.“

Auf Deutsch:

H4M1.8 „wobei der Decodierer dazu ausgebildet ist, gemäß dem ersten (20) der zumindest zwei Modi, für das aktuelle Stück den Wert von nicht vorhandenen Stück-Header-Syntaxelementen als gleich dem Wert von Stück-Header-Syntaxelementen in dem *unmittelbar vorhergehenden* in dem zweiten Modus decodierten Stück abzuleiten.“

Entsprechende Änderungen wurden in den unabhängigen Ansprüchen 10, 14 und 15 gemäß Hilfsantrag III vorgenommen.

4.2 Merkmal **H4M1.8** betrifft genauso wie Merkmal **H1M1.8** die „Header-Struktur“ der „Slices“. Allerdings stellt die Beklagte in Hilfsantrag IV jetzt explizit klar, dass die abhängigen „Slices“ (d. h. die „Slices“ gemäß dem ersten Modus) die Header-Syntaxelemente jeweils von dem unmittelbar vorhergehenden unabhängigen „Slice“ (d. h. dem „Slice“ gemäß dem zweiten Modus) „erben“.

4.3 Hilfsantrag V beruht auf Hilfsantrag IV. Die Patentansprüche 1 und 10 gemäß Hilfsantrag V entsprechen den Patentansprüchen 1 und 14 gemäß

Hilfsantrag IV.

4.4 Mit Rücksicht auf den aus der Druckschrift **2NK4** bekannten Stand der Technik ist die Lehre des jeweiligen Patentanspruchs 1 gemäß Hilfsantrag IV und V nicht neu.

So ist Merkmal **H4M1.8** auf Seite 5, Punkt 3.43 der Druckschrift **2NK4** offenbart.

Unter Berücksichtigung der Ausführungen zum Hauptantrag und zu Hilfsantrag I gehen damit sämtliche Merkmale des jeweiligen Patentanspruchs 1 gemäß Hilfsantrag IV und V aus der Druckschrift **2NK4** hervor.

4.5 Mit seinem jeweiligen Patentanspruch 1 fällt auch der gesamte Hilfsantrag IV und V.

5. Aus diesen Gründen war das Streitpatent, das somit in keiner seiner durch die Beklagte verteidigten Fassungen bestandsfähig ist, im angegriffenen Umfang für nichtig zu erklären.

IV.

Die Kostenentscheidung beruht auf § 84 Abs. 2 Satz 1 und Satz 2 Halbsatz 1 PatG i. V. m. § 91 Abs. 1 ZPO.

Die Entscheidung über die vorläufige Vollstreckbarkeit beruht auf § 99 Abs. 1 PatG i. V. m. § 709 Satz 1 und 2 ZPO.

V.

Rechtsmittelbelehrung

Gegen dieses Urteil ist das Rechtsmittel der Berufung gemäß § 110 PatG statthaft.

Die Berufung ist innerhalb eines Monats nach Zustellung des in vollständiger Form abgefassten Urteils - spätestens nach Ablauf von fünf Monaten nach Verkündung - durch einen in der Bundesrepublik Deutschland zugelassenen Rechtsanwalt oder Patentanwalt schriftlich beim Bundesgerichtshof, Herrenstraße 45a, 76133 Karlsruhe, einzulegen.

Die Berufungsschrift muss

- die Bezeichnung des Urteils, gegen das die Berufung gerichtet ist, sowie
- die Erklärung, dass gegen dieses Urteil Berufung eingelegt werde,

enthalten. Mit der Berufungsschrift soll eine Ausfertigung oder beglaubigte Abschrift des angefochtenen Urteils vorgelegt werden.

Auf die Möglichkeit, die Berufung nach § 125a PatG in Verbindung mit § 2 der Verordnung über den elektronischen Rechtsverkehr beim Bundesgerichtshof und Bundespatentgericht (BGH/BPatGERVV) auf elektronischem Weg beim Bundesgerichtshof einzulegen, wird hingewiesen (www.bundesgerichtshof.de/erv.html).

Hartlieb Dr. Forkel Hoffmann Dr. Himmelmann Dr. Städele