



(12) **DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION**

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2016/09/15
(87) Date publication PCT/PCT Publication Date: 2017/03/23
(85) Entrée phase nationale/National Entry: 2018/02/20
(86) N° demande PCT/PCT Application No.: US 2016/051987
(87) N° publication PCT/PCT Publication No.: 2017/048991
(30) Priorités/Priorities: 2015/09/18 (US62/220,699);
2015/10/21 (US62/244,690); 2016/09/14 (US15/265,743)

(51) Cl.Int./Int.Cl. *H04N 19/119* (2014.01),
H04N 19/11 (2014.01), *H04N 19/154* (2014.01),
H04N 19/176 (2014.01), *H04N 19/467* (2014.01),
H04N 19/593 (2014.01)
(71) Demandeur/Applicant:
QUALCOMM INCORPORATED, US
(72) Inventeurs/Inventors:
JACOBSON, NATAN HAIM, US;
THIRUMALAI, VIJAYARAGHAVAN, US;
JOSHI, RAJAN LAXMAN, US
(74) Agent: SMART & BIGGAR

(54) Titre : TAILLE DE PARTITION VARIABLE POUR UN MODE DE PREDICTION DE BLOC POUR UNE
COMPRESSION DE FLUX D'AFFICHAGE (DSC)
(54) Title: VARIABLE PARTITION SIZE FOR BLOCK PREDICTION MODE FOR DISPLAY STREAM COMPRESSION
(DSC)

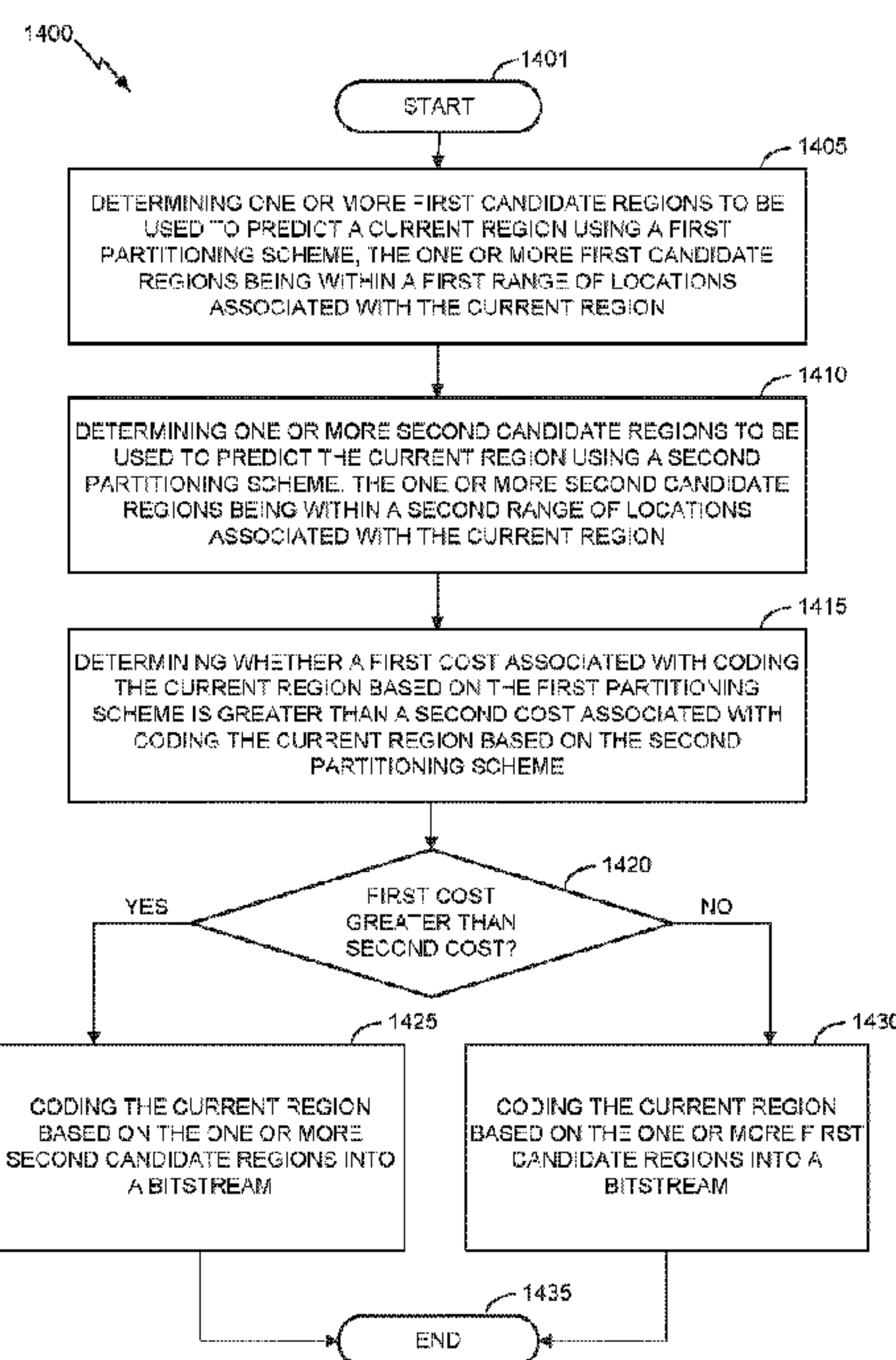


FIG. 14

(57) Abrégé/Abstract:

A method for coding a block of video data in block prediction mode of a constant bitrate video coding scheme for transmission over display links is disclosed. In one aspect, the method includes determining one or more first candidate regions to be used to predict a current region within the block of video data using a first partitioning scheme, determining one or more second candidate regions to be used to predict the current region using a second partitioning scheme, determining that a first cost associated with coding the current region using the first partitioning scheme is greater than a second cost associated with coding the current region using the second partitioning scheme, and coding the current region using the second partitioning scheme.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau



WIPO | PCT



(10) International Publication Number
WO 2017/048991 A1

(43) International Publication Date
23 March 2017 (23.03.2017)

(51) International Patent Classification:

H04N 19/119 (2014.01) *H04N 19/176* (2014.01)
H04N 19/11 (2014.01) *H04N 19/467* (2014.01)
H04N 19/154 (2014.01) *H04N 19/593* (2014.01)

(21) International Application Number:

PCT/US2016/051987

(22) International Filing Date:

15 September 2016 (15.09.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/220,699 18 September 2015 (18.09.2015) US
62/244,690 21 October 2015 (21.10.2015) US
15/265,743 14 September 2016 (14.09.2016) US

(71) Applicant: QUALCOMM INCORPORATED [US/US];

Attn: International IP Administration, 5775 Morehouse
Drive, San Diego, California 92121-1714 (US).

(72) Inventors: JACOBSON, Natan Haim;

5775 Morehouse
Drive, San Diego, California 92121-1714 (US). **THIRU-
MALAI, Vijayaraghavan;** 5775 Morehouse Drive, San
Diego, California 92121-1714 (US). **JOSHI, Rajan Lax-**

man; 5775 Morehouse Drive, San Diego, California
92121-1714 (US).

(74) **Agent: ABUMERI, Mark M.;** Knobbe Martens Olson &
Bear LLP, 2040 Main Street, Fourteenth Floor, Irvine,
California 92614 (US).

(81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM,
ZW.

(84) **Designated States** (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,

[Continued on next page]

(54) **Title:** VARIABLE PARTITION SIZE FOR BLOCK PREDICTION MODE FOR DISPLAY STREAM COMPRESSION (DSC)

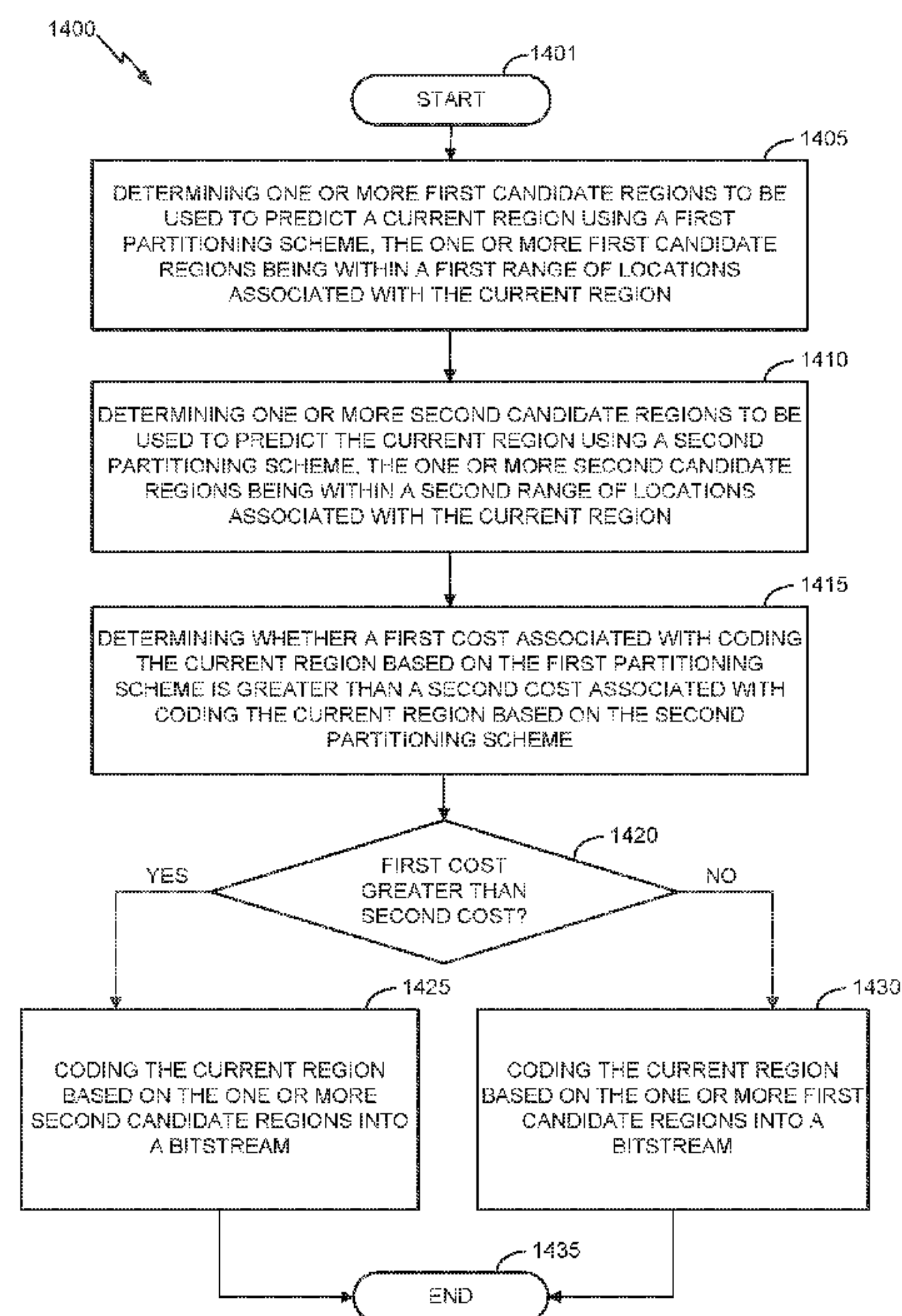


FIG. 14

(57) **Abstract:** A method for coding a block of video data in block prediction mode of a constant bitrate video coding scheme for transmission over display links is disclosed. In one aspect, the method includes determining one or more first candidate regions to be used to predict a current region within the block of video data using a first partitioning scheme, determining one or more second candidate regions to be used to predict the current region using a second partitioning scheme, determining that a first cost associated with coding the current region using the first partitioning scheme is greater than a second cost associated with coding the current region using the second partitioning scheme, and coding the current region using the second partitioning scheme.

WO 2017/048991 A1

WO 2017/048991 A1



LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, **Published:**
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, — *with international search report (Art. 21(3))*
GW, KM, ML, MR, NE, SN, TD, TG).

VARIABLE PARTITION SIZE FOR BLOCK PREDICTION MODE FOR DISPLAY STREAM COMPRESSION (DSC)

TECHNICAL FIELD

[0001] This disclosure relates to the field of video coding and compression, and particularly to video compression for transmission over display links, such as display link video compression.

BACKGROUND

[0002] Digital video capabilities can be incorporated into a wide range of displays, including digital televisions, personal digital assistants (PDAs), laptop computers, desktop monitors, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, video conferencing devices, and the like. Display links are used to connect displays to appropriate source devices. The bandwidth requirements of display links are proportional to the resolution of the displays, and thus, high-resolution displays require large bandwidth display links. Some display links do not have the bandwidth to support high resolution displays. Video compression can be used to reduce the bandwidth requirements such that lower bandwidth display links can be used to provide digital video to high resolution displays.

[0003] There are coding schemes that involve image compression on the pixel data. However, such schemes are sometimes not visually lossless or can be difficult and expensive to implement in conventional display devices.

[0004] The Video Electronics Standards Association (VESA) has developed Display Stream Compression (DSC) as a standard for display link video compression. The display link video compression technique, such as DSC, should provide, among other things, picture quality that is visually lossless (i.e., pictures having a level of quality such that users cannot tell the compression is active). The display link video compression technique should also provide a scheme that is easy and inexpensive to implement in real-time with conventional hardware.

SUMMARY

[0005] The Display Stream Compression (DSC) standard includes a number of coding modes in which each block of video data may be encoded by an encoder and, similarly, decoded by a decoder. In some implementations, the encoder and/or the decoder may predict the current block to be coded based on a previously coded block.

[0006] However, the existing coding modes (e.g., transform coding, differential pulse-code modulation, etc.) do not provide a satisfactory way of compressing highly complex regions in video data. Often, for this type of data (i.e., highly compressed video data), the current block to be coded (or the current block's constituent sub-blocks) is similar in content to previous blocks that have been encountered by the coder (e.g., encoder or decoder). However, the existing intra prediction may be too limited to provide a satisfactory prediction of such a current block (e.g., prediction of the current block that is sufficiently similar to the current block and would thus yield a sufficiently small residual). Thus, an improved method of coding blocks of video data is desired.

[0007] The systems, methods and devices of this disclosure each have several innovative aspects, no single one of which is solely responsible for the desirable attributes disclosed herein.

[0008] In one aspect, a method for coding a block of video data in block prediction mode of a constant bitrate video coding scheme may include: determining one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region, wherein the one or more first candidate regions are stored in a memory of a video encoding device; determining one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region, wherein the one or more second candidate regions are stored in the memory of the video encoding device; determining whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and in response to determining that the first cost is greater than the second cost, coding the current region in a bitstream based on the one or more second candidate regions.

[0009] In another aspect, an apparatus configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme may include: a memory configured to store video data associated with one or more candidate regions, and one or more processors in communication with the memory. The one or more processors may be configured to: determine one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region; determine one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region; determine whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and in response to determining that the first cost is greater than the second cost, code the current region in a bitstream based on the one or more second candidate regions.

[0010] In another aspect, non-transitory physical computer storage may comprise code configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme. The code, when executed, may cause an apparatus to: determine one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region; determine one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region; determine whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and in response to determining that the first cost is greater than the second cost, code the current region in a bitstream based on the one or more second candidate regions.

[0011] In another aspect, a video coding device may be configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme. The video coding device may comprise: means for determining one or more first

candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region; means for determining one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region; means for determining whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; means for coding, in response to determining that the first cost is greater than the second cost, the current region in a bitstream based on the one or more second candidate regions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] **FIG. 1A** is a block diagram illustrating an example video encoding and decoding system that may utilize techniques in accordance with aspects described in this disclosure.

[0013] **FIG. 1B** is a block diagram illustrating another example video encoding and decoding system that may perform techniques in accordance with aspects described in this disclosure.

[0014] **FIG. 2A** is a block diagram illustrating an example of a video encoder that may implement techniques.

[0015] **FIG. 2B** is a block diagram illustrating an example of a video decoder that may implement techniques.

[0016] **FIG. 3** is a block diagram illustrating the search space for a non-first line for a 1-D block.

[0017] **FIG. 4** is a block diagram illustrating the search space for a non-first line for a 2-D block.

[0018] **FIG. 5** is a block diagram illustrating the search space for a first line for a 1-D block.

[0019] **FIG. 6** is a block diagram illustrating the search space for a first line for a 2-D block.

[0020] **FIG. 7** is a flowchart illustrating a method for predicting a block of video data in block prediction mode.

[0021] FIG. 8 is a block diagram illustrating a block having partitions.

[0022] FIG. 9 is a block diagram illustrating a data flow for block prediction mode with adaptive partition size.

[0023] FIG. 10 is a block diagram illustrating the two different partition options for a 2x2 region within a block.

[0024] FIG. 11 is a block diagram illustrating the entropy coding groups for block prediction mode.

[0025] FIG. 12 is a block diagram illustrating the search space for a 2x8 block.

[0026] FIG. 13 is a block diagram illustrating different partition sizes being used for different regions of a block.

[0027] FIG. 14 is a flowchart illustrating a method for predicting a block of video data in block prediction mode using variable partition sizes.

[0028] FIG. 15 is a block diagram illustrating an example block prediction search for 2x2 partitions for 4:2:0 chroma sub-sampling.

[0029] FIG. 16 is a block diagram illustrating an example block prediction search for 1x2 partitions for 4:2:0 chroma sub-sampling.

[0030] FIG. 17 is a block diagram illustrating an example block prediction search for 2x2 partitions for 4:2:2 chroma sub-sampling.

[0031] FIG. 18 is a block diagram illustrating an example block prediction search for 1x2 partitions for 4:2:2 chroma sub-sampling.

DETAILED DESCRIPTION

[0032] In general, this disclosure relates to methods of improving video compression techniques, such as those utilized in display link video compression, for example. More specifically, the present disclosure relates to systems and methods for coding a block of video data in block prediction mode using variable partition sizes.

[0033] While certain embodiments are described herein in the context of the DSC standard, which is an example of a display link video compression technique, one having ordinary skill in the art would appreciate that systems and methods disclosed herein may be applicable to any suitable video coding standard. For example, embodiments disclosed herein may be applicable to one or more of the following standards: International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) H.261, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Moving Picture

Experts Group-1 (MPEG-1) Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC), High Efficiency Video Coding (HEVC), and any extensions to such standards. Also, the techniques described in this disclosure may become part of standards developed in the future. In other words, the techniques described in this disclosure may be applicable to previously developed video coding standards, video coding standards currently under development, and forthcoming video coding standards.

[0034] The DSC standard includes a number of coding modes in which each block of video data may be encoded by an encoder and, similarly, decoded by a decoder. In some implementations, the encoder and/or the decoder may predict the current block to be coded based on a previously coded block.

[0035] However, the existing coding modes (e.g., transform coding, differential pulse-code modulation, etc.) do not provide a satisfactory way of compressing highly complex regions in video data. Often, for this type of data (i.e., highly compressed video data), the current block to be coded (or the current block's constituent sub-blocks) is similar in content to previous blocks that have been encountered by the coder (e.g., encoder or decoder). However, the existing intra prediction may be too limited to provide a satisfactory prediction of such a current block (e.g., prediction of the current block that is sufficiently similar to the current block and would thus yield a sufficiently small residual). Thus, an improved method of coding blocks of video data is desired.

[0036] In the present disclosure, an improved method of coding a block in block prediction mode is described. For example, when searching for a candidate block (or a candidate region) to be used to predict the current block (or a current region within the current block), a search range may be defined such that the encoder has access to potential candidates that may be a good match while minimizing the search cost. In another example, the method may include explicitly signaling a prediction for each block (or each partition). In another example, the encoder may determine whether to code the current block using a single partition or multiple partitions based on a rate distortion (RD) analysis. By performing more operations (e.g., searching for a candidate block to be used for predicting the current block, calculating a vector identifying the location of the candidate block with respect to the current block, calculating the RD cost for different partition sizes and determining which partition size yields the best coding efficiency, etc., which may consume computing resources and processing power) on the encoder side, the method may reduce decoder complexity. Further, by allowing the encoder to adaptively

select the partition size for each block, the performance of the block prediction scheme may further be improved.

Video Coding Standards

[0037] A digital image, such as a video image, a TV image, a still image or an image generated by a video recorder or a computer, may include pixels or samples arranged in horizontal and vertical lines. The number of pixels in a single image is typically in the tens of thousands. Each pixel typically contains luminance and chrominance information. Without compression, the sheer quantity of information to be conveyed from an image encoder to an image decoder would render real-time image transmission impractical. To reduce the amount of information to be transmitted, a number of different compression methods, such as JPEG, MPEG and H.263 standards, have been developed.

[0038] Video coding standards include ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC), and HEVC including extensions of such standards.

[0039] In addition, a video coding standard, namely DSC, has been developed by VESA. The DSC standard is a video compression standard which can compress video for transmission over display links. As the resolution of displays increases, the bandwidth of the video data required to drive the displays increases correspondingly. Some display links may not have the bandwidth to transmit all of the video data to the display for such resolutions. Accordingly, the DSC standard specifies a compression standard for interoperable, visually lossless compression over display links.

[0040] The DSC standard is different from other video coding standards, such as H.264 and HEVC. DSC includes intra-frame compression, but does not include inter-frame compression, meaning that temporal information may not be used by the DSC standard in coding the video data. In contrast, other video coding standards may employ inter-frame compression in their video coding techniques.

Video Coding System

[0041] Various aspects of the novel systems, apparatuses, and methods are described more fully hereinafter with reference to the accompanying drawings. This disclosure may, however, be embodied in many different forms and should not be

construed as limited to any specific structure or function presented throughout this disclosure. Rather, these aspects are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the disclosure to those skilled in the art. Based on the teachings herein one skilled in the art should appreciate that the scope of the disclosure is intended to cover any aspect of the novel systems, apparatuses, and methods disclosed herein, whether implemented independently of, or combined with, any other aspect of the present disclosure. For example, an apparatus may be implemented or a method may be practiced using any number of the aspects set forth herein. In addition, the scope of the present disclosure is intended to cover such an apparatus or method which is practiced using other structure, functionality, or structure and functionality in addition to or other than the various aspects of the present disclosure set forth herein. It should be understood that any aspect disclosed herein may be embodied by one or more elements of a claim.

[0042] Although particular aspects are described herein, many variations and permutations of these aspects fall within the scope of the disclosure. Although some benefits and advantages of the preferred aspects are mentioned, the scope of the disclosure is not intended to be limited to particular benefits, uses, or objectives. Rather, aspects of the disclosure are intended to be broadly applicable to different wireless technologies, system configurations, networks, and transmission protocols, some of which are illustrated by way of example in the figures and in the following description of the preferred aspects. The detailed description and drawings are merely illustrative of the disclosure rather than limiting, the scope of the disclosure being defined by the appended claims and equivalents thereof.

[0043] The attached drawings illustrate examples. Elements indicated by reference numbers in the attached drawings correspond to elements indicated by like reference numbers in the following description. In this disclosure, elements having names that start with ordinal words (e.g., “first,” “second,” “third,” and so on) do not necessarily imply that the elements have a particular order. Rather, such ordinal words are merely used to refer to different elements of a same or similar type.

[0044] **FIG. 1A** is a block diagram that illustrates an example video coding system 10 that may utilize techniques in accordance with aspects described in this disclosure. As used described herein, the term “video coder” or “coder” refers generically to both video encoders and video decoders. In this disclosure, the terms “video coding” or “coding” may refer generically to video encoding and video decoding.

In addition to video encoders and video decoders, the aspects described in the present application may be extended to other related devices such as transcoders (e.g., devices that can decode a bitstream and re-encode another bitstream) and middleboxes (e.g., devices that can modify, transform, and/or otherwise manipulate a bitstream).

[0045] As shown in **FIG. 1A**, video coding system 10 includes a source device 12 (i.e., “video coding device 12” or “coding device 12”) that generates encoded video data to be decoded at a later time by a destination device 14 (i.e., “video coding device 14” or “coding device 14”). In the example of **FIG. 1A**, the source device 12 and destination device 14 constitute separate devices. It is noted, however, that the source device 12 and destination device 14 may be on or part of the same device, as shown in the example of **FIG. 1B**.

[0046] With reference once again, to **FIG. 1A**, the source device 12 and the destination device 14 may respectively comprise any of a wide range of devices (also referred to as video coding devices) including desktop computers, notebook (e.g., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called “smart” phones, so-called “smart” pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In various embodiments, the source device 12 and the destination device 14 may be equipped for (i.e., configured to communicate via) wireless communication.

[0047] The video coding devices 12, 14 of the video coding system 10 may be configured to communicate via wireless networks and radio technologies, such as wireless wide area network (WWAN) (e.g., cellular) and/or wireless local area network (WLAN) carriers. The terms “network” and “system” are often used interchangeably. Each of the video coding devices 12, 14 may be a user equipment (UE), a wireless device, a terminal, a mobile station, a subscriber unit, etc.

[0048] The WWAN carriers may include, for example, wireless communication networks such as Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Orthogonal FDMA (OFDMA), Single-Carrier FDMA (SC-FDMA) and other networks. A CDMA network may implement a radio technology such as Universal Terrestrial Radio Access (UTRA), CDMA2000, etc. UTRA includes Wideband CDMA (WCDMA) and other variants of CDMA. CDMA2000 covers IS-2000, IS-95 and IS-856 standards. A TDMA network may implement a radio technology such as Global System for Mobile Communications (GSM). An OFDMA network may implement a radio technology such

as Evolved UTRA (E-UTRA), Ultra Mobile Broadband (UMB), IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20, Flash-OFDMA, etc. UTRA and E-UTRA are part of Universal Mobile Telecommunication System (UMTS). 3GPP Long Term Evolution (LTE) and LTE-Advanced (LTE-A) are new releases of UMTS that use E-UTRA. UTRA, E-UTRA, UMTS, LTE, LTE-A and GSM are described in documents from an organization named “3rd Generation Partnership Project” (3GPP). CDMA2000 and UMB are described in documents from an organization named “3rd Generation Partnership Project 2” (3GPP2).

[0049] The video coding devices 12, 14 of the video coding system 10 may also communicate with each other via a WLAN base station according to one or more standards, such as the IEEE 802.11 standard, including, for example these amendments: 802.11a-1999 (commonly called “802.11a”), 802.11b-1999 (commonly called “802.11b”), 802.11g-2003 (commonly called “802.11g”), and so on.

[0050] The destination device 14 may receive, via link 16, the encoded video data to be decoded. The link 16 may comprise any type of medium or device capable of moving the encoded video data from the source device 12 to the destination device 14. In the example of **FIG. 1A**, the link 16 may comprise a communication medium to enable the source device 12 to transmit encoded video data to the destination device 14 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source device 12 to the destination device 14.

[0051] In the example of **FIG. 1A**, the source device 12 includes a video source 18, a video encoder 20 (also referred to as simply encoder 20) and an output interface 22. In some cases, the output interface 22 may include a modulator/demodulator (modem) and/or a transmitter. In the source device 12, the video source 18 may include a source such as a video capture device, e.g., a video camera, a video archive containing previously captured video, a video feed interface to receive video from a video content provider, and/or a computer graphics system for generating

computer graphics data as the source video, or a combination of such sources. As one example, if the video source 18 is a video camera, the source device 12 and the destination device 14 may form so-called “camera phones” or “video phones”, as illustrated in the example of **FIG. 1B**. However, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

[0052] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video data may be transmitted to the destination device 14 via the output interface 22 of the source device 12. The encoded video data may also (or alternatively) be stored onto the storage device 31 for later access by the destination device 14 or other devices, for decoding and/or playback. The video encoder 20 illustrated in **FIG. 1A** and **1B** may comprise the video encoder 20 illustrated **FIG. 2A** or any other video encoder described herein.

[0053] In the example of **FIG. 1A**, the destination device 14 includes the input interface 28, a video decoder 30 (also referred to as simply decoder 30), and a display device 32. In some cases, the input interface 28 may include a receiver and/or a modem. The input interface 28 of the destination device 14 may receive the encoded video data over the link 16 and/or from the storage device 31. The encoded video data communicated over the link 16, or provided on the storage device 31, may include a variety of syntax elements generated by the video encoder 20 for use by a video decoder, such as the video decoder 30, in decoding the video data. Such syntax elements may be included with the encoded video data transmitted on a communication medium, stored on a storage medium, or stored a file server. The video decoder 30 illustrated in **FIG. 1A** and **1B** may comprise the video decoder 30 illustrated in **FIG. 2B** or any other video decoder described herein.

[0054] The display device 32 may be integrated with, or external to, the destination device 14. In some examples, the destination device 14 may include an integrated display device and also be configured to interface with an external display device. In other examples, the destination device 14 may be a display device. In general, the display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0055] In related aspects, **FIG. 1B** shows an example video coding system 10' wherein the source device 12 and the destination device 14 are on or part of a device 11.

The device 11 may be a telephone handset, such as a “smart” phone or the like. The device 11 may include a processor/controller device 13 (optionally present) in operative communication with the source device 12 and the destination device 14. The video coding system 10' of **FIG. 1B**, and components thereof, are otherwise similar to the video coding system 10 of **FIG. 1A**, and components thereof.

[0056] The video encoder 20 and the video decoder 30 may operate according to a video compression standard, such as DSC. Alternatively, the video encoder 20 and the video decoder 30 may operate according to other proprietary or industry standards, such as the ITU-T H.264 standard, alternatively referred to as MPEG-4, Part 10, AVC, HEVC or extensions of such standards. The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples of video compression standards include MPEG-2 and ITU-T H.263.

[0057] Although not shown in the examples of **FIGS. 1A** and **1B**, the video encoder 20 and the video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, in some examples, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0058] The video encoder 20 and the video decoder 30 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of the video encoder 20 and the video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder in a respective device.

Video Coding Process

[0059] As mentioned briefly above, the video encoder 20 encodes video data. The video data may comprise one or more pictures. Each of the pictures is a still image forming part of a video. In some instances, a picture may be referred to as a video

“frame.” When the video encoder 20 encodes the video data (e.g., video coding layer (VCL) data and/or non-VCL data), the video encoder 20 may generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. A coded picture is a coded representation of a picture. VCL data may include coded picture data (i.e., information associated with samples of a coded picture(s)) and non-VCL data may include control information (e.g., parameter sets and/or supplemental enhancement information) associated with the one or more coded pictures.

[0060] To generate the bitstream, the video encoder 20 may perform encoding operations on each picture in the video data. When the video encoder 20 performs encoding operations on the pictures, the video encoder 20 may generate a series of coded pictures and associated data. The associated data may include a set of coding parameters such as a quantization parameter (QP). To generate a coded picture, the video encoder 20 may partition a picture into equally-sized video blocks. A video block may be a two-dimensional array of samples. The coding parameters may define a coding option (e.g., a coding mode) for every block of the video data. The coding option may be selected in order to achieve a desired RD performance.

[0061] In some examples, the video encoder 20 may partition a picture into a plurality of slices. Each of the slices may include a spatially distinct region in an image (e.g., a frame) that can be decoded independently without information from the rest of the regions in the image or frame. Each image or video frame may be encoded in a single slice or each image or video frame may be encoded in several slices. In DSC, the number of bits allocated to encode each slice may be substantially constant. As part of performing an encoding operation on a picture, the video encoder 20 may perform encoding operations on each slice of the picture. When the video encoder 20 performs an encoding operation on a slice, the video encoder 20 may generate encoded data associated with the slice. The encoded data associated with the slice may be referred to as a “coded slice.”

DSC Video Encoder

[0062] FIG. 2A is a block diagram illustrating an example of the video encoder 20 that may implement techniques in accordance with aspects described in this disclosure. The video encoder 20 may be configured to perform some or all of the techniques of this disclosure. In some examples, the techniques described in this

disclosure may be shared among the various components of the video encoder 20. In some examples, additionally or alternatively, a processor (not shown) may be configured to perform some or all of the techniques described in this disclosure.

[0063] For purposes of explanation, this disclosure describes the video encoder 20 in the context of DSC coding. However, the techniques of this disclosure may be applicable to other coding standards or methods.

[0064] In the example of **FIG. 2A**, the video encoder 20 includes a plurality of functional components. The functional components of the video encoder 20 include a color-space converter 105, a buffer, 110, a flatness detector 115, a rate controller 120, a predictor, quantizer, and reconstructor component 125, a line buffer 130, an indexed color history 135, an entropy encoder 140, a substream multiplexor 145, and a rate buffer 150. In other examples, the video encoder 20 may include more, fewer, or different functional components.

[0065] The color-space 105 converter may convert an input color-space to the color-space used in the coding implementation. For example, in one exemplary embodiment, the color-space of the input video data is in the red, green, and blue (RGB) color-space and the coding is implemented in the luminance Y, chrominance green Cg, and chrominance orange Co (YCgCo) color-space. The color-space conversion may be performed by method(s) including shifts and additions to the video data. It is noted that input video data in other color-spaces may be processed and conversions to other color-spaces may also be performed.

[0066] In related aspects, the video encoder 20 may include the buffer 110, the line buffer 130, and/or the rate buffer 150. For example, the buffer 110 may hold (e.g., store) the color-space converted video data prior to its use by other portions of the video encoder 20. In another example, the video data may be stored in the RGB color-space and color-space conversion may be performed as needed, since the color-space converted data may require more bits.

[0067] The rate buffer 150 may function as part of the rate control mechanism in the video encoder 20, which will be described in greater detail below in connection with rate controller 120. The number of bits spent on encoding each block can vary highly substantially based on the nature of the block. The rate buffer 150 can smooth the rate variations in the compressed video. In some embodiments, a constant bit rate (CBR) buffer model is employed in which bits stored in the rate buffer (e.g., the rate buffer 150) are removed from the rate buffer at a constant bit rate. In the CBR buffer model, if the

video encoder 20 adds too many bits to the bitstream, the rate buffer 150 may overflow. On the other hand, the video encoder 20 may need to add enough bits in order to prevent underflow of the rate buffer 150.

[0068] On the video decoder side, the bits may be added to rate buffer 155 of the video decoder 30 (see **FIG. 2B** which is described in further detail below) at a constant bit rate, and the video decoder 30 may remove variable numbers of bits for each block. To ensure proper decoding, the rate buffer 155 of the video decoder 30 should not “underflow” or “overflow” during the decoding of the compressed bit stream.

[0069] In some embodiments, the buffer fullness (BF) can be defined based on the values BufferCurrentSize representing the number of bits currently in the buffer and BufferMaxSize representing the size of the rate buffer 150, i.e., the maximum number of bits that can be stored in the rate buffer 150 at any point in time. The BF may be calculated as:

$$BF = ((\text{BufferCurrentSize} * 100) / \text{BufferMaxSize})$$

[0070] The flatness detector 115 can detect changes from complex (i.e., non-flat) areas in the video data to flat (i.e., simple or uniform) areas in the video data. The terms “complex” and “flat” will be used herein to generally refer to the difficulty for the video encoder 20 to encode the respective regions of the video data. Thus, the term complex as used herein generally describes a region of the video data as being complex for the video encoder 20 to encode and may, for example, include textured video data, high spatial frequency, and/or other features which are complex to encode. The term flat as used herein generally describes a region of the video data as being simple for the video encoder 20 to encode and may, for example, include a smooth gradient in the video data, low spatial frequency, and/or other features which are simple to encode. The transitions between complex and flat regions may be used by the video encoder 20 to reduce quantization artifacts in the encoded video data. Specifically, the rate controller 120 and the predictor, quantizer, and reconstructor component 125 can reduce such quantization artifacts when the transitions from complex to flat regions are identified.

[0071] The rate controller 120 determines a set of coding parameters, e.g., a QP. The QP may be adjusted by the rate controller 120 based on the buffer fullness of the rate buffer 150 and image activity of the video data in order to maximize picture quality for a target bitrate which ensures that the rate buffer 150 does not overflow or underflow.

The rate controller 120 also selects a particular coding option (e.g., a particular mode) for each block of the video data in order to achieve the optimal RD performance. The rate controller 120 minimizes the distortion of the reconstructed images such that the rate controller 120 satisfies the bit-rate constraint, i.e., the overall actual coding rate fits within the target bit rate.

[0072] The predictor, quantizer, and reconstructor component 125 may perform at least three encoding operations of the video encoder 20. The predictor, quantizer, and reconstructor component 125 may perform prediction in a number of different modes. One example predication mode is a modified version of median-adaptive prediction. Median-adaptive prediction may be implemented by the lossless JPEG standard (JPEG-LS). The modified version of median-adaptive prediction which may be performed by the predictor, quantizer, and reconstructor component 125 may allow for parallel prediction of three consecutive sample values. Another example prediction mode is block prediction. In block prediction, samples are predicted from previously reconstructed pixels in the line above or to the left in the same line. In some embodiments, the video encoder 20 and the video decoder 30 may both perform an identical search on reconstructed pixels to determine the block prediction usages, and thus, no bits need to be sent in the block prediction mode. In other embodiments, the video encoder 20 may perform the search and signal block prediction vectors in the bitstream, such that the video decoder 30 need not perform a separate search. A midpoint prediction mode may also be implemented in which samples are predicted using the midpoint of the component range. The midpoint prediction mode may enable bounding of the number of bits required for the compressed video in even the worst-case sample. As further discussed below with reference to **FIGS. 3-18**, the predictor, quantizer, and reconstructor component 125 may be configured to code (e.g., encode or decode) the block of video data (or any other unit of prediction) based on one or more techniques described herein. For example, the predictor, quantizer, and reconstructor component 125 may be configured to perform the methods illustrated in **FIGS. 7 and 14**. In other embodiments, the predictor, quantizer, and reconstructor component 125 may be configured to perform one or more methods or techniques described herein with one or more other components of the video encoder 20.

[0073] The predictor, quantizer, and reconstructor component 125 also performs quantization. For example, quantization may be performed via a power-of-2 quantizer which may be implemented using a shifter. It is noted that other quantization

techniques may be implemented in lieu of the power-of-2 quantizer. The quantization performed by the predictor, quantizer, and reconstructor component 125 may be based on the QP determined by the rate controller 120. Finally, the predictor, quantizer, and reconstructor component 125 also performs reconstruction which includes adding the inverse quantized residual to the predicted value and ensuring that the result does not fall outside of the valid range of sample values.

[0074] It is noted that the above-described example approaches to prediction, quantization, and reconstruction performed by the predictor, quantizer, and reconstructor component 125 are merely illustrative and that other approaches may be implemented. It is also noted that the predictor, quantizer, and reconstructor component 125 may include subcomponent(s) for performing the prediction, the quantization, and/or the reconstruction. It is further noted that the prediction, the quantization, and/or the reconstruction may be performed by several separate encoder components in lieu of the predictor, quantizer, and reconstructor component 125.

[0075] The line buffer 130 holds (e.g., stores) the output from the predictor, quantizer, and reconstructor component 125 so that the predictor, quantizer, and reconstructor component 125 and the indexed color history 135 can use the buffered video data. The indexed color history 135 stores recently used pixel values. These recently used pixel values can be referenced directly by the video encoder 20 via a dedicated syntax.

[0076] The entropy encoder 140 encodes the prediction residuals and any other data (e.g., indices identified by the predictor, quantizer, and reconstructor component 125) received from the predictor, quantizer, and reconstructor component 125 based on the indexed color history 135 and the flatness transitions identified by the flatness detector 115. In some examples, the entropy encoder 140 may encode three samples per clock per substream encoder. The substream multiplexor 145 may multiplex the bitstream based on a headerless packet multiplexing scheme. This allows the video decoder 30 to run three entropy decoders in parallel, facilitating the decoding of three pixels per clock. The substream multiplexor 145 may optimize the packet order so that the packets can be efficiently decoded by the video decoder 30. It is noted that different approaches to entropy coding may be implemented, which may facilitate the decoding of power-of-2 pixels per clock (e.g., 2 pixels/clock or 4pixels/clock).

DSC Video Decoder

[0077] **FIG. 2B** is a block diagram illustrating an example of the video decoder 30 that may implement techniques in accordance with aspects described in this disclosure. The video decoder 30 may be configured to perform some or all of the techniques of this disclosure. In some examples, the techniques described in this disclosure may be shared among the various components of the video decoder 30. In some examples, additionally or alternatively, a processor (not shown) may be configured to perform some or all of the techniques described in this disclosure.

[0078] For purposes of explanation, this disclosure describes the video decoder 30 in the context of DSC coding. However, the techniques of this disclosure may be applicable to other coding standards or methods.

[0079] In the example of **FIG. 2B**, the video decoder 30 includes a plurality of functional components. The functional components of the video decoder 30 include a rate buffer 155, a substream demultiplexor 160, an entropy decoder 165, a rate controller 170, a predictor, quantizer, and reconstructor component 175, an indexed color history 180, a line buffer 185, and a color-space converter 190. The illustrated components of the video decoder 30 are analogous to the corresponding components described above in connection with the video encoder 20 in **FIG. 2A**. As such, each of the components of the video decoder 30 may operate in a similar fashion to the corresponding components of the video encoder 20 as described above. In some embodiments, one or more components of the video encoder 20 and/or video decoder 30 may be implemented by one or more hardware processors configured to execute software code configured to perform the tasks of such components. In other embodiments, one or more components of the video encoder 20 and/or the video decoder 30 may be implemented by hardware circuitry configured to perform the tasks of such components.

Slices in DSC

[0080] As noted above, a slice generally refers to a spatially distinct region in an image or a frame that can be decoded independently without using the information from the rest of the regions in the image or frame. Each image or video frame may be encoded in a single slice or it may be encoded in several slices. In DSC, the target bits allocated to encode each slice may be substantially constant.

Block Prediction Mode

[0081] A single block of video data may contain a number of pixels, and each block of video data has a number of potential coding modes in which the block can be coded. One of such coding modes is block prediction mode. In block prediction mode, the coder attempts to find a candidate block in the previous reconstructed line (e.g., if the current block is not in the first line of the current slice) or previous reconstructed blocks in the same line (e.g., if the current block is in the first line of the current slice) that is close (e.g., in pixel values) to the current block to be coded. In some embodiments, closeness between pixel values is determined by the Sum of Absolute Differences (SAD) metric. The coder may attempt to find the candidate block in any portion of the previously reconstructed blocks defined by a search range (e.g., which may be a predetermined value known to both the encoder and the decoder). The search range is defined such that the encoder has potential candidates within the search range to find a good match while minimizing the search cost. The coding efficiency of block prediction mode comes from the fact that, if a good candidate (i.e., a candidate within the search range that is determined to be close in pixel values to the current block to be coded) is discovered, the difference (known as the residual) between the candidate block and the current block will be small. The small residual will take a fewer number of bits to signal compared to the number of bits needed to signal the actual pixel values of the current block, thereby resulting in a lower RD cost and increasing the likelihood of being selected by the RD mechanism. The performance boost from enabling block prediction mode is extremely significant for certain types of graphics content.

Parameters in Block Prediction Mode

[0082] The block prediction mode is designed to produce a candidate block, given a specified search range, that provides the minimum distortion from the current block to be encoded. In some embodiments, minimum distortion is defined using SAD. In some implementations of the present disclosure, the block prediction method is defined by three parameters: search range (SR), skew (α), and partition size (β). These three parameters affect the performance of the block prediction mode, and may be tuned (i.e., modified or reconfigured) during implementation. These parameters may be known to both the encoder and the decoder.

Search Space in Block Prediction Mode

[0083] In some embodiments of the present disclosure, the search space (e.g., spatial locations of pixels that the encoder may search in order to find a candidate block) may differ based on the characteristics of the current block. The search space may encompass all previously reconstructed blocks/pixels, but the encoder and/or the decoder may limit the search for a candidate block to a specified portion (e.g., a “search range” defined by one or more parameters that are either predefined or signaled in the bitstream) within the search space, for example, to reduce computational complexity. Examples of the block prediction search space are illustrated in **FIGS. 3-6**. **FIGS. 3** and **4** illustrate cases involving a current block (e.g., current blocks 308 and 408) that is not in the first line of the current slice. **FIGS. 5** and **6** illustrate cases involving a current block (e.g., current blocks 506 and 606) that is in the first line of the current slice. These two cases are handled separately because the first line in a slice has no vertical neighbors. Therefore, the reconstructed pixels from the current line can be leveraged as a search range (e.g., search ranges 508 and 608). In the present disclosure, the first line in the current slice may be referred to as an FLS and any other line in the current slice may be referred to as an NFLS.

[0084] Further, the block prediction techniques described herein may be implemented in either a codec using a single line buffer (i.e., 1-D block size) or a codec using multiple line buffers (i.e., 2-D block size). The codec may be a fixed-bit codec, in which Examples of the search space for the 1-D case are shown in **FIGS. 3** and **5**, and examples of the search space for the 2-D case are shown in **FIGS. 4** and **6**. In the 2-D case, the search range may include pixels from the previous reconstructed line (e.g., previous line 402) or reconstructed blocks from the same lines as those in the 2-D block (e.g., previous 604 in the current line 602, which is immediately to the left of the current block 606). The 2-D block may be partitioned either horizontally or vertically or both. In the case involving block partitions, a block prediction vector may be specified for each block partition.

Example Implementations of Block Prediction Mode

[0085] In some embodiments of the present disclosure, a distortion metric other than SAD may be used, e.g. sum of squared differences (SSD). Alternately or additionally, the distortion may be modified by weighting. For example, if the YCoCg color space is being used, then the cost may be calculated as:

$$SAD(Y) + \frac{SAD(Co) + SAD(Cg)}{2}$$

[0086] The block prediction techniques described herein may be performed either in the RGB or YCoCg color space. In addition, an alternative implementation may use both color spaces and signal a 1-bit flag to the decoder indicating which of the two color spaces is selected (e.g., whichever color space that has the lowest cost in terms of rate and distortion).

[0087] In some embodiments of the present disclosure concerning FLS, the direct previous reconstructed block or blocks may be excluded from the search range due to pipelining and timing constraints. For example, depending on the hardware implementation, the coder may not have completed the processing of the direct previous reconstructed block by the time the current block is processed by the coder (e.g., the reconstructed pixels for the previous block may not be known when the coder begins processing the current block), resulting in delays or failures. In such an implementation, by restricting the use of previous reconstructed blocks to those blocks for which reconstructed pixel values are known (e.g., by excluding the direct previous reconstructed block or blocks), the pipelining concerns illustrated above may be resolved. In some embodiments of the present disclosure concerning NFLS, the search range to the left of the current block may be from the same line rather than the previous reconstructed line. In some of such embodiments, one or more previous reconstructed blocks may be excluded from the search range due to pipelining and timing constraints.

Example Implementation of NFLS

[0088] As shown in **FIG. 3**, the block prediction method may search through the search range 310 (SR) in the search space to find a candidate for the current block 308 (and similarly in the search space 400 of **FIG. 4**). If the x-coordinate position of the first pixel of the current block 308 to be encoded is j , then the set of starting positions k of all candidate blocks within the search space may be given as:

$$k \in [j - SR + (\alpha + 1), j + \alpha]$$

[0089] In this example, the parameter α skews the x-coordinate position of the search range 310 relative to the current block to be encoded. A higher value of α shifts the search range 310 to the right, while a lower value of α shifts the search range 310 to the left. For example, (i) SR of 32 and α of 15 may place the search range 310 in the center of the previous line 302, (ii) SR of 32 and α of 0 may place the search range 310 on

the left side of the previous line 302, and (iii) SR of 32 and α of 31 may place the search range 310 on the right side of the previous line 302.

[0090] In some implementations of the present disclosure, a pixel that is within the search range but outside of the slice boundary may be set to half the dynamic range for that pixel. For example, if the content is RGB888, then the default value of 128 may be used for R, G, and B. If the content is in the YCoCg space, then a default value of 128 may be used for Y, and a default value of 0 may be used for Co and Cg (e.g., Co and Cg are 9-bit values that are centered around 0).

Example Implementation of FLS

[0091] As shown in FIG. 5, the search range may be different for the FLS case. This is because vertical neighbors are not available because such vertical neighbors are outside of the current frame, or because such vertical neighbors are contained within a different slice. In some embodiments of the present disclosure concerning the FLS case, pixels in the current line may be used for block prediction. In one embodiment, any pixel in the current line to the left of the current block may be considered as part of the search range. In another embodiment, one or more previously coded blocks (e.g., the previous block 504 that is immediately to the left of the current block) may be excluded from the search range due to pipelining and timing constraints.

[0092] In some implementations of FLS, the available range for the first few blocks in the first line of the slice may be less than the search range that is typically expected for other blocks. This is because the valid position for candidate blocks starts at the beginning of the line and ends before the current block. For the first few blocks in the FLS, this valid range may be smaller than the desired range (e.g., 32 or 64 positions). Thus, for these blocks, the search range may need to be adjusted such that each block partition of the candidate block is fully contained within the search range. For NFLS, the search range may be shifted left or right such that the total number of search positions is equal to the defined search range (e.g., 32 or 64 pixel positions). Since j is the first pixel in the current block, the last pixel in the current block will be $j + blkWidth - 1$. For this reason, the search range may need to be shifted $(blkWidth - 1)$ pixels to the left.

[0093] In some implementations of FLS, if the x-coordinate location of the first pixel of the current block to be encoded is referred to as j , then the set of starting positions of all candidate blocks within the search range is given as:

[0094] (i) if most recent previous reconstructed block is part of the search range, e.g., $\alpha = -1$:

$$k \in [j - SR - (blkWidth - 1), j - 1 - (blkWidth - 1)]$$

[0095] (ii) if n most recent previous reconstructed blocks are to be excluded from the search range:

$$k \in [j - (n \cdot blk_x + SR) - (blkWidth - 1), j - (n \cdot blk_x + 1) - (blkWidth - 1)]$$

[0096] where blk_x is the block width. Any pixel outside of the slice boundary may be set to a default value as described above in connection with the NFLS case. It should also be noted that no skew parameter need be associated with the FLS case.

Example Flowchart for Coding in Block Prediction Mode

[0097] With reference to **FIG. 7**, an example procedure for coding a block of video data in block prediction mode will be described. The steps illustrated in **FIG. 7** may be performed by a video encoder (e.g., the video encoder 20 in **FIG. 2A**), a video decoder (e.g., the video decoder 30 in **FIG. 2B**), or component(s) thereof. For convenience, method 700 is described as performed by a video coder (also simply referred to as coder), which may be the video encoder 20, the video decoder 30, or another component.

[0098] The method 700 begins at block 701. At block 705, the coder determines a candidate block to be used for predicting a current block in a current slice. The candidate block may be within a range of locations defined by one or more block prediction parameters. For example, the block prediction parameters may include (i) a search range parameter defining the size of the range of locations, (ii) a skew parameter defining the relative location of the range of locations with respect to the current block, and (iii) a partition size parameter defining the size of each partition in the current block. In some embodiments of the present disclosure, each of the search range parameter, the skew parameter, and the partition size parameter spatially, rather than temporally, define the locations of the candidate block.

[0099] At block 710, the coder determines a prediction vector based on the candidate block and the current block. The prediction vector may identify the location of the candidate block with respect to the current block. The prediction vector may include one or more coordinate values (e.g., a coordinate value indicating the offset in the 1-D space). At block 715, the coder codes the current block in block prediction mode at least in part via signaling the prediction vector. In some embodiments, the coder may also

signal the residual between the candidate block and the current block. Bit savings may be achieved by signaling the prediction vector identifying the location of the candidate block and the residual representing the difference between the current block and the candidate block, instead of having to signal the actual pixel values of the current block. The method 700 ends at block 720.

[0100] In the method 700, one or more of the blocks shown in **FIG. 7** may be removed (e.g., not performed) and/or the order in which the method is performed may be switched. In some embodiments, additional blocks may be added to the method 700. The embodiments of the present disclosure are not limited to or by the example shown in **FIG. 7**, and other variations may be implemented without departing from the spirit of this disclosure.

After Finding Candidate Block

[0101] After the best candidate block has been determined, the pixel values of the candidate block are subtracted from the pixel values of the current block, resulting in the residual. The residual may be quantized based on a pre-selected QP associated with the block prediction mode. The quantized residual may be encoded using a codebook (which can be either fixed-length or variable-length) and signaled using a fixed-length code or a variable-length code. The selected codebook may be based on the coding efficiency and hardware complexity requirements. For example, the selected codebook may be an Exp-Golomb codebook. In some embodiments of the present disclosure, an entropy coding scheme that is similar to the delta size unit variable length coding (DSU-VLC) of existing DSC implementations may be used. In some embodiments, the residual may be transformed (e.g., using a direct cosine transform, a Hadamard transform, or other known transforms) before the quantization described above.

[0102] In some embodiments of the present disclosure, the samples in the residual of the current block may be partitioned into multiple groups (e.g., 4 samples per group for a block that contains 16 samples). If all the coefficients in the block are zero, then the residual of the block is coded using skip mode, i.e., 1-bit flag per block (per component) is signaled to indicate if the current component in the block is coded using skip mode or not. If at least one non-zero value is contained within the block, each group may be coded using DSU-VLC only if the group has one non-zero value. If the group (e.g., 4 samples of the 16 samples in the residual) does not contain any non-zero values, the group is coded using skip mode, i.e., 1-bit flag per group is signaled to indicate if the

group is coded using skip mode or not. More specifically, for each group, a search may be performed to determine whether all the values in the group are zero. If all the values in the group are zero, a value of '1' may be signaled to the decoder; otherwise (if at least one value is non-zero), a value of '0' may be signaled to the decoder, followed by the coding of the DSU-VLC coding. In an alternative example, a value of '0' may be signaled if all the values in the group are zero and a value of '1' may be signaled if the group contains at least one non-zero value.

[0103] In some embodiments of the present disclosure, the best candidate block is signaled explicitly to the decoder by transmitting a fixed-length code containing the best offset. The offset may be referred to as a "vector". The advantage of signaling the vector explicitly to the decoder is that the decoder will not have to perform the block search itself. Rather, the decoder will receive the vector explicitly and add the candidate block to the decoded, de-quantized residual values to determine the pixel values of the current block.

Block Partitioning

[0104] In some embodiments of the present disclosure, the current block to be coded may be partitioned, resulting in multiple candidate blocks and multiple vectors per block. In some of such embodiments, the vector(s) may be explicitly signaled using a fixed-length code. For example, the length of this fixed-length code may be $\log_2(SR)$. In another embodiment, the vector(s) may be explicitly signaled using a variable-length code, such as a code from the Exponential-Golomb or Golomb-Rice code families. This codebook could be selected based on the statistical distribution associated with vector(s). In yet another embodiment, the vector(s) may be predicted based on the previously-coded vector(s), and the residual of the vector(s) may be coded using some fixed-length or variable-length code. In yet another embodiment, the vector(s) may be predicted based on the previously-coded vector(s), and a 1-bit flag may be used to signal whether the two vectors are identical. This flag may be referred to as *SameFlag*. If *SameFlag* = 1, then the vector value itself need not be signaled to the decoder. If *SameFlag* = 0, then the vector will be signaled explicitly (e.g., using either a fixed-length or variable-length code). An example block partitioning scheme is illustrated in **FIG. 8**.

[0105] As shown in **FIG. 8**, a current block 802 contains a single partition. The information signaled for the current block 802 comprises a mode header, a vector *SameFlag*, a vector A, and a payload. A current block 804 contains two partitions,

partition A and partition B. The information signaled for the current block 804 comprises a mode header, a vector *SameFlag*, a vector A, a vector *SameFlag*, a vector B, and a payload. As described above, one or more items listed above may not be signaled. For example, if the vector *SameFlag* is equal to 1, the following vector need not be signaled.

[0106] The partition size β may determine the partitioning of the current block into separate sub-blocks. In such a case, a separate block prediction may be performed for each sub-block. For example, if the block size is $N = 16$ and partition size $\beta = 8$, then the search will be performed for each of the $16 / 8 = 2$ partitions. In another example, if $\beta = N$, block partitioning is disabled. If $\beta < N$, then each vector may be signaled explicitly to the decoder. If vector prediction (e.g., using previously signaled vectors to define the current vectors) is not employed, then each vector will be signaled using a fixed-length or variable-length code. If vector prediction is employed, the first vector may be predicted from the previous coded vector (e.g., stored in memory) and for $n > 0$, vector n is predicted from vector $n - 1$.

Variable Partition Size in Block Prediction Mode

[0107] The examples above illustrate how blocks having a size of 1×8 (e.g., having a height of 1 pixel and a width of 8 pixels) or 2×8 (e.g., having a height of 2 pixels and a width of 8 pixels) may be coded in block prediction mode. As shown in **FIG. 8**, a block may be partitioned into multiple regions and each region can be coded using different partitioning schemes (e.g., using 1×2 partitions, using 2×2 partitions, etc.), and a block prediction vector may be specified for each partition (e.g., signaled in the bitstream along with the residual associated with each partition). For example, each block may be partitioned into multiple 1×2 partitions containing two pixels (or partitions of other fixed sizes).

[0108] In other embodiments, the encoder may determine the block partition size that is most efficient for each block (for each sub-region within the block). The efficiency may be measured based on the rate and distortion associated with coding the block (or a sub-region therein) using the given block partition size. For example, when coding a block containing four 2×2 regions, the encoder may determine that the greatest coding efficiency can be achieved by coding the first three 2×2 regions using single partitions (e.g., a single 2×2 partition for each 2×2 region) and coding the fourth 2×2 region using two partitions (e.g., two 1×2 partitions). By allowing the encoder to adaptively select the partition size for each block, the performance of the block prediction

scheme can be further improved. This is because large partitions can be used for smooth regions (e.g., regions exhibiting no change or less than a threshold amount of change in pixel values across the region), thereby requiring fewer bits to signal block prediction vectors (e.g., relative to the size of the region), while using smaller partitions can be used for complex regions (where the decrease in distortion and/or entropy coding rate outweighs the additional signaling cost). For example, the encoder may determine whether a given region or block satisfies a smoothness threshold condition, and in response to determining that the given region or block satisfies the smoothness threshold condition, encode the given region or block in block prediction mode using a larger partition size (and otherwise, encode the given region or block in block prediction mode using a smaller partition size). As another example, the encoder may determine whether a given region or block satisfies a complexity threshold condition, and in response to determining that the given region or block satisfies the complexity threshold condition, encode the given region or block in block prediction mode using a smaller partition size (and otherwise, encode the given region or block in block prediction mode using a larger partition size). The ability to adaptively select different partition sizes may allow the block prediction mode to be used in a larger range of content types (e.g., graphics content, natural images, test patterns, fine text rendering, etc.).

Example Data Flow of Coding in Block Prediction Mode

[0109] **FIG. 9** illustrates an example data flow 900 for coding a block in block prediction mode using adaptive partition size. As illustrated in **FIG. 9**, a current block 902 to be predicted in block prediction mode includes a block partition 904. In one example, the block partition has a size of 1x2 or 2x2. A block prediction (BP) search 906 is conducted to identify a block or partition that has already been coded and available for predicting the current block 902 (or the block partition 904) in block prediction mode. As shown in **FIG. 9**, the BP search 906 may search within a search range, for example, including one or more previous reconstructed blocks 907A in a previous line (e.g., the line coded prior to coding the current line including the current block, such as the immediately preceding line or another preceding line) and/or previous reconstructed blocks 907B from the current line (e.g., the line including the current block).

[0110] The encoder determines a block predictor 908 based on a candidate block or partition identified in the search range. The block predictor 908 is subtracted from the current block 902 (or the current block partition 904 within the candidate block

902) at block 910, and the residual determined based on the subtraction is quantized at block 912. The quantized residual is entropy coded by the entropy coder 920. In addition, inverse quantization 914 is performed on the quantized residual and the result is added to the block predictor 908 at block 916 to produce a reconstructed block 918. A BP partition size selection 922 is performed based on the distortion performance (D) of the reconstructed block 918 and the rate performance (R) of the entropy encoded residual. A bitstream 924 is generated based on the selected BP partition size.

[0111] For example, the BP partition size selection 922 may take as input the rate (e.g., R) and distortion (e.g., D) of each partition region (e.g., 2x2) within the current block 902 and determine whether the partition region should be coded using a single block prediction vector (BPV) (e.g., 1 BPV total for a single 2x2 partition) or be partitioned and coded using multiple BPVs (e.g., 2 BPVs total, 1 BPV each for two 1x2 partitions) for prediction based on the RD tradeoff between the two options. Although some examples discussed herein involve a partition region size of 2x2 (thereby having partition sizes of 1x2, 2x1, and 2x2 as selectable options), the partition sizes selectable by the encoder are not limited to those used in such examples, (e.g., 1x2 and 2x2), and may include other sizes (e.g., 2x1) based on the block size and/or region size.

[0112] In some embodiments, the partition sizes are fixed (e.g., 1x2, 2x2, or any other sub-combination of pixels in the current partition region or block. For example, a block may have a block size of 2x8, and the block may be divided into sub-blocks or regions having a size of 2x2. The 2x2 sub-blocks or regions within the 2x8 block may further be partitioned into partitions having a size of 1x2. In such an example, each 1x2 partition may be predicted using a single BPV, independently from other partitions. In other embodiments, the partition sizes are variable, and how each block, sub-block, and/or region is coded in block prediction using which partition sizes may be determined by the encoder based on the rate and distortion performance of each partitioning scheme. For example, for a 2x2 region (e.g., current region) within the current block, if predicting the current region by dividing the current region into two 1x2 partitions and predicting the two 1x2 partitions separately using two BPVs (e.g., each pointing to a previously coded 1x2 partition within the defined search range) yields better rate and/or distortion performance (e.g., compared to other partitioning schemes such as 2x2), the current region may be predicted using the 1x2 partitioning scheme. On the other hand, if predicting the current region as a single 2x2 partition using one BPV (e.g., pointing to a previously coded 2x2 partition within the defined search range) yields better rate and/or

distortion performance (e.g., compared to other partitioning schemes such as 1x2), the current region may be predicted using the 2x2 partitioning scheme. The process of determining the partitioning scheme to be used for coding a block in prediction mode is described in greater detail below with reference to **FIG. 14**.

Block Sizes and Sub-Block Sizes

[0113] For a block size of $M \times N$, some embodiments are described with reference to sub-blocks (also referred to herein as regions) of size $M_{\text{sub}} \times N_{\text{sub}}$ where $M_{\text{sub}} \leq M$ and $N_{\text{sub}} \leq N$. In some implementations, for ease of computation, both M_{sub} and N_{sub} are aligned with the entropy coding groups within the $M \times N$ block. Each sub-block $M_{\text{sub}} \times N_{\text{sub}}$ within the block may either be (i) predicted using a single BPV without being further partitioned or (ii) partitioned into multiple partitions (e.g., into two 1x2 partitions), with a BPV used for each partition. The effective trade-off between using a single BPV for the entire sub-block or partitioning the sub-block into partitions that each have a BPV of its own is that signaling more BPVs will incur extra rate in the bitstream, however by using more BPVs, the distortion and entropy coding rates may decrease. In other words, by using more bits to signal additional BPVs, the number of bits used for signaling the residual (difference between the candidate block/region and the current block/region) may be reduced, which may further cause the number of bits used for entropy coding to be reduced as well. The encoder may compare each option (e.g., no partition vs. multiple partitions) in terms of RD cost and select whether or not to partition each sub-block or region based on the cost comparison or select a partitioning scheme from a plurality of partitioning schemes that provides the best RD performance.

Example Partitioning Scheme

[0114] **FIG. 10** illustrates a diagram 1000 illustrating an example partitioning scheme. In the **FIG. 10**, two partitioning options for a 2x2 sub-block or region is illustrated. In this example, a block 1002 (e.g., including pixels X_0 - X_{15}) has a size of 2x8, and a sub-block or region 1004 (e.g., including pixels X_0 , X_1 , X_8 , and X_9) within the block has a size of 2x2. Partitioning option 1006 illustrates an example in which the sub-block or region 1004 is predicted using a single BPV, and partitioning option 1008 illustrates an example in which the sub-block or region 1004 is predicted using two BPVs for each 1x2 partition within the sub-block or region 1004. Sub-blocks or regions having a size of 2x2 are used in some implementations such as the Advanced DSC (Adv-DSC) to

align the sub-blocks or regions with the entropy coding group structures 1100 for block prediction mode, shown in **FIG. 11**. In the example of **FIG. 11**, entropy coding groups 0, 1, 2, and 3 are illustrated, each corresponding to one of the four 2x2 sub-blocks or regions within the block. However, the techniques described herein are not limited to such an embodiment and may be extended to any block size $M \times N$ and any sub-block size $M_{\text{sub}} \times N_{\text{sub}}$. However, in the examples illustrated below, parameters $M = 2$, $N = 8$, $M_{\text{sub}} = 2$, $N_{\text{sub}} = 2$ are used. In some embodiments, the sub-blocks and/or partitioning schemes may be determined based on the entropy coding groups. For example, the sub-blocks and/or partitioning schemes may be determined such that each sub-block and/or partitioning scheme is contained within a single entropy coding group.

Determining the Partition Size

[0115] The encoder may determine whether to (i) code each 2x2 region as a single 2x2 partition or (ii) divide the region into two 1x2 partitions and code each 1x2 partition separately, based on the minimum RD cost. The RD cost may be computed as shown below:

$$\begin{aligned} \text{cost}(2x2) &= D_{2x2} + \lambda \cdot R_{2x2} \\ \text{cost}(1x2) &= D_{1x2} + \lambda \cdot R_{1x2} \\ R_{2x2} &= 1 + BPV_{\text{bits}} + EC_{\text{bits}2x2} \\ R_{1x2} &= 1 + (2 \cdot BPV_{\text{bits}}) + EC_{\text{bits}1x2} \end{aligned}$$

[0116] In some implementations, the BPV is signaled with a fixed number of bits (BPV_{bits}), equal to $\log_2(\text{SR})$, where SR is the search space (or search range) associated with the block prediction mode. For example, if the search space consists of 64 positions, then $\log_2(64) = 6$ bits are used to signal each BPV.

[0117] The search space for block prediction with variable partition size may be slightly different than the search range discussed with reference to **FIGS. 3-6**. In particular, a $M_{\text{sub}} \times N_{\text{sub}}$ sub-block may utilize a search space with height M_{sub} . In such cases, additional line buffers may be needed to implement block prediction with variable partition size relative to block prediction without variable partition size. An example of such search space is demonstrated in **FIG. 12** for a sub-block size of 2x2. **FIG. 12** illustrates a diagram 1200 illustrating an example search range. As shown in **FIG. 12**, a current line 1202 includes (i) a current block 1204 having a current sub-block 1206 and (ii) a previous block 1208. In the example of **FIG. 12**, a previous line 1210 includes a search range 1212 from which the encoder may select a candidate sub-block 1214 for

predicting the current sub-block 1206. The search range or space for 1-D partitions (e.g. 1x2) may be similar to the search range previously described with reference to **FIG. 3**, relying on a single previous reconstructed line.

[0118] In some embodiments, distortions $D_{2 \times 2}$ and $D_{1 \times 2}$ may be computed using a modified sum of absolute differences (SAD) in the YCoCg color space. For example, the SAD distortion between pixel A (e.g., in the current sub-block or partition) and pixel B (e.g., in the candidate sub-block or region) in the YCoCg color space may be calculated as follows:

$$SAD(A, B) = |A_Y - B_Y| + 0.5 \cdot |A_{Co} - B_{Co}| + 0.5 \cdot |A_{Cg} - B_{Cg}|$$

[0119] If the current sub-block or partition has more than one pixel, the distortion for the entire current sub-block or partition may be calculated by summing the individual SADs calculated for each pixel in the current sub-block or partition. The pixel values of the current sub-block or partition may be the actual pixel value or a reconstructed pixel value (e.g., calculated based on a candidate predictor and a residual). In some implementations, the lambda parameter may be fixed at a value of 2. In other implementations, this parameter may be tuned depending on the block size, bitrate, or other coding parameters.

[0120] The entropy coding cost EC_{bits} may be computed for each 2x2 region. The four samples in each entropy coding group may either come from the 2x2 quantized residual predicted from a single BPV (e.g., a 2x2 partition), or the 2x2 quantized residual utilizing two vectors (e.g., two 1x2 partitions). For example, the entropy coding cost may represent the number of bits needed to signal each entropy coding group in the bitstream (e.g., including the vector(s) and the residual). Based on the computed entropy coding costs, the encoder may select the partitioning scheme having the lowest cost for each 2x2 region. Although some embodiments are discussed with reference to 2x8 blocks having 2x2 sub-block sizes, 2x2 entropy coding groups, and two partitioning schemes (1x2 and 2x2), the techniques described herein may be extended to other block sizes, sub-block sizes, entropy coding groups, and/or partitioning schemes.

Signaling Coding Information in the Bitstream

[0121] In the 2x8 block 1002 shown in **FIG. 10**, each of the four 2x2 regions may be partitioned based on the RD cost analysis discussed above. For example, each 2x2 region may be partitioned either into a single 2x2 partition or two 1x2 partitions. Four examples of such partitioning are illustrated by a diagram 1300 of **FIG. 13**. As

shown in FIG. 13, block 1302 has four sub-blocks predicted based on the 2x2 partitioning scheme, block 1304 has three sub-blocks predicted based on the 2x2 partitioning scheme and one sub-block predicted based on the 1x2 partitioning scheme, block 1306 has four sub-blocks predicted based on the 1x2 partitioning scheme, and block 1308 has one sub-block predicted based on the 2x2 partitioning scheme and three sub-blocks predicted based on the 1x2 partitioning scheme. In addition to signaling the BPVs to the decoder, the encoder may also send one bit for each 2x2 region so that the decoder can properly infer the partitioning. In some implementations such as the Adv-DSC implementation, a group of four bits indicative of the partitioning scheme selected for each region within the block (e.g., each 2x2 region in the 2x8 block) is signaled in the bitstream. In such implementations, the four bits “1011” may indicate that the first, third, and fourth region (e.g., 2x2 sub-block) in the block are to be predicted or coded based on a first partitioning scheme (e.g., based on 1x2 partitions), while the second region (e.g., 2x2 sub-block) is to be predicted or coded based on a second partitioning scheme (e.g., based on 2x2 partitions). In some embodiments, following these four bits in the bitstream, the BPVs may be signaled using fixed bits per BPV. In the previous example (e.g., bit sequence of “1011”), 7 BPVs may be signaled.

Example Flowchart for Coding in Block Prediction Mode

[0122] With reference to **FIG. 14**, an example procedure for coding a block of video data in block prediction mode will be described. The steps illustrated in **FIG. 14** may be performed by a video encoder (e.g., the video encoder 20 in **FIG. 2A**) or component(s) thereof. For convenience, method 1400 is described as performed by a coder, which may be the video encoder 20 or another component.

[0123] The method 1400 begins at block 1401. At block 1405, the coder determines one or more first candidate regions to be used to predict a current region (e.g., within a block of video data that is coded in block prediction mode) based on a first partitioning scheme. For example, the current region may be one of the 2x2 regions in a 2x8 block. The first partitioning scheme may be a partitioning scheme in which the current region is partitioned into multiple partitions (e.g., two 1x2 partitions, or other combinations of partitions having partition sizes determined based on the size of the current region). Alternatively, the first partitioning scheme may be a partitioning scheme in which the current region is used as a whole (e.g., as a 2x2 partition) and not partitioned into multiple partitions. In some embodiments, the one or more first candidate regions

are within a first range (e.g., the search range associated with the first partitioning scheme) of locations associated with the first partitioning scheme. The one or more first candidate regions may be stored in a memory of a video encoding device.

[0124] At block 1410, the coder determines one or more second candidate regions to be used to predict the current region based on a second partitioning scheme. For example, the second partitioning scheme may be a partitioning scheme in which the current region is not partitioned into multiple partitions (e.g., the current region is coded as a single 2x2 partition). In another example, the second partitioning scheme may be a partitioning scheme in which the current region is partitioned into a different number of partitions than the number of partitions used for the first partitioning scheme. In yet another example, the second partitioning scheme may be a partitioning scheme in which the current region is partitioned into multiple partitions (e.g., two 1x2 partitions, or other combinations of partitions having partition sizes determined based on the size of the current region). In some embodiments, the one or more second candidate regions are within a second range (e.g., the search range associated with the second partitioning scheme) of locations associated with the second partitioning scheme. In some embodiments, the second range is the same as the first range used for identifying the one or more first candidate regions. In some cases, the one or more second candidate regions may be identical to the one or more first candidate regions. In other cases, the one or more second candidate regions include the one or more first candidate regions. Alternatively, the one or more first candidate regions may include the one or more second candidate regions. In some cases, the one or more second candidate regions do not overlap with the one or more first candidate regions. The size of the one or more second candidate regions may be different from the size of the one or more first candidate regions. In other embodiments, the second range is different from the first range used for identifying the one or more first candidate regions. The one or more second candidate regions may be stored in the memory of the video encoding device.

[0125] At block 1415, the coder determines whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme. For example, the code may calculate the first cost based on the rate and distortion associated with coding the current region based on the first partitioning scheme and the second cost based on the rate and distortion associated with coding the current region based on the second partitioning scheme, and compare the calculated first and

second costs. In one example, the first cost may be determined as (a first distortion value + (a lambda parameter * a first rate value)), where the first distortion value may be calculated based on the modified SAD of the individual pixels in the current region (or a partition thereof) in the YCoCg color space with respect to the one or more first candidate regions, and the second cost may be determined as (a second distortion value + (a lambda parameter * a second rate value)), where the second distortion value may be calculated based on the modified SAD of the individual pixels in the current region (or a partition thereof) in the YCoCg color space with respect to the one or more second candidate regions. In some embodiments, the coder may determine the first cost based at least in part on (i) a sum of absolute differences between the current region and the one or more first candidate regions and (ii) a number of bits needed to signal the one or more prediction vectors and corresponding residuals in the bitstream, and determine the second cost based at least in part on (i) a sum of absolute differences between the current region and the one or more second candidate regions and (ii) a number of bits needed to signal the one or more prediction vectors and corresponding residuals in the bitstream.

[0126] At block 1420, if the coder has determined that the first cost associated with coding the current region based on the first partitioning scheme is greater than the second cost associated with coding the current region based on the second partitioning scheme, the method 1400 proceeds to block 1425. Otherwise, the method 1400 proceeds to block 1430.

[0127] At block 1425, the coder codes the current region based on the one or more second candidate regions into a bitstream. The coder may signal, in a bitstream, one or more prediction vectors indicative of a location of the one or more second candidate regions with respect to the current region and a quantized residual indicative of a difference between the one or more second candidate regions and the current region (e.g., difference between corresponding pixel values). For example, the coder may signal a single vector indicative of the location of the first or initial pixel of the one or more second candidate regions, where the value of the single vector is based on the distance between such first or initial pixel and the first or initial pixel of the current region. If the one or more prediction vectors comprise multiple vectors, the coder may signal multiple vectors each indicative of the location of the respective candidate region to be used to predict one of the partitions of the current region.

[0128] At block 1430, the coder codes the current region based on the one or more first candidate regions into a bitstream. The coder may signal, in a bitstream, one or

more prediction vectors indicative of a location of the one or more first candidate regions with respect to the current region and a quantized residual indicative of a difference between the one or more first candidate regions and the current region. For example, the coder may signal a single vector indicative of the location of the first or initial pixel of the one or more first candidate regions, where the value of the single vector is based on the distance between such first or initial pixel and the first or initial pixel of the current region. If the one or more prediction vectors comprise multiple vectors, the coder may signal multiple vectors each indicative of the location of the respective candidate region to be used to predict one of the partitions of the current region. The coder may further signal a partition indicator in the bitstream, the partition indicator indicative of a partitioning scheme associated with each region within the block, the block comprising at least one region other than the current region. For example, the partition indicator may indicate that current region is associated with the second partitioning scheme. The partition indicator may further indicate that the at least one region other than the current region in the block is associated with the first partitioning scheme different from the second partitioning scheme. The method 1400 ends at block 1435.

[0129] In the method 1400, one or more of the blocks shown in **FIG. 14** may be removed (e.g., not performed) and/or the order in which the method is performed may be switched. In some embodiments, additional blocks may be added to the method 1400. The embodiments of the present disclosure are not limited to or by the example shown in **FIG. 14**, and other variations may be implemented without departing from the spirit of this disclosure.

Extension to 4:2:0 and 4:2:2 Chroma Subsampling Formats

[0130] In some implementations, the block prediction techniques described in the present disclosure (e.g., using variable partition sizes in block prediction mode) may be utilized for 4:4:4 chroma sampling format only. This format is commonly used for graphics content. For example, the 4:4:4 chroma sampling format utilizes image or video data containing color components (e.g., luma components and chroma components) that have the same sampling rate (e.g., not using chroma sub-sampling). However, the 4:4:4 chroma sampling format may be less commonly used for other video applications. Due to the significant compression that chroma sub-sampling may provide, both 4:2:0 and 4:2:2 chroma sub-sampling formats are commonly used for video applications. For example, some versions of DSC (e.g., DSCv1.x) may support 4:2:0 and 4:2:2. Support for such

chroma sub-sampling formats may be utilized or required by future DSC implementations. Thus, in some embodiments, the block prediction techniques described in the present disclosure (e.g., using variable partition sizes in block prediction mode) are extended to the 4:2:0 and/or 4:2:2 formats. Although 4:2:0 and 4:2:2 chroma sub-sampling formats are used herein, the various techniques described in the present application may be applied to other known sampling formats.

[0131] In some embodiments, the algorithm for block prediction with variable partition size works much in the same way independent of the chroma sampling format. In such embodiments, regardless of the format (e.g., 4:4:4, 4:2:2, 4:2:0, etc.), the determination of whether to use a single partition (e.g., 2x2) or to use multiple partitions (e.g., two separate 1x2 partitions) or the determination of the number of partitions to be used to code the current sub-block or region (e.g., 1, 2, 3, 4, etc.) may be made for each sub-block or region (e.g., 2x2 block) of luma samples. However, the number of chroma samples in each partition or in each block may differ depending on the sub-sampling format. In addition, the encoder decision may need to be modified in 4:2:2 and/or 4:2:0 chroma sub-sampling formats since alignment with entropy coding groups may no longer be possible for chroma components. Therefore, the rate (e.g., rate value associated with the partitions, such as the single 2x2 partition or the two separate 1x2 partitions) for each partition for the encoder decision (e.g., when the encoder decides whether to divide each 2x2 region into a single 2x2 partition or two 1x2 partitions based on the minimum RD cost) may rely solely on the luma samples for 4:2:2 and 4:2:0. For example, when calculating the SAD distortion, any terms related to the chroma component(s) may be set to zero.

BP Search for 4:2:0 Chroma Subsampling Format

[0132] For 2x2 partitions in 4:2:0 mode (4:2:0 chroma sub-sampling format), each partition may contain a single chroma sample for each of the chroma components (e.g., Co and Cg, or Cb and Cr). In some embodiments, the chroma sample to be used (e.g., for calculating the RD cost and/or for predicting the samples in the current region or block) is the one that intersects with the partition. In other embodiments, the chroma sample to be used may be derived from an adjacent partition. An example 2x2 search 1500 for the 4:2:0 mode is shown in **FIG. 15**. In **FIG. 15**, the chroma sites (e.g., sample/pixel locations having chroma samples) are indicated using “X”. For example, the top left sample of partition A, the top right sample of partition B, and the top left

sample of the current partition comprise chroma sites that intersect the respective partitions. Such chroma sites may be used for all calculations performed for the respective partitions (e.g., to calculate the difference value using the chroma sample values).

[0133] For 1x2 partitions in 4:2:0 mode, a distinction may need to be made between 1x2 partitions in the first line of the current block and 1x2 partitions in the second line of the current block, because there may be no chroma sites in the second line of the current block. For example, for partitions in the first line of the current block, the calculation of the distortion values may involve two luma samples and one chroma sample for each chroma component. For partitions in the second line of the current block, the calculation of the distortion values may involve only the luma samples (e.g., two luma samples). In the example 1600 of **FIG. 16**, the current 1x2 partition A is in the first line and includes a chroma site. Thus, the candidate partition selected for predicting the current 1x2 partition A is the candidate 1x2 partition A, which also includes a chroma site. Similarly, the current 1x2 partition B is in the second line and does not include a chroma site. Thus, the candidate partition selected for predicting the current 1x2 partition B is the candidate 1x2 partition B, which also does not include a chroma site.

BP Search for 4:2:2 Chroma Subsampling Format

[0134] For 2x2 partitions in 4:2:2 mode (4:2:2 chroma sub-sampling format), each partition may contain 4 luma samples, and 2 chroma samples for each of the chroma components (e.g., Co and Cg, or Cb and Cr). An example 2x2 search 1700 for the 4:2:2 mode is shown in **FIG. 17**. In **FIG. 17**, the chroma sites (e.g., pixel locations having chroma samples) are indicated using “X”. For example, the two left samples of partition A, the two right samples of partition B, and the two left samples of the current partition comprise chroma sites that intersect the respective partitions. Such chroma sites may be used for all calculations performed for the respective partitions (e.g., to calculate the difference value using the chroma sample values).

[0135] For 1x2 partitions in 4:2:2 mode, each partition contains 2 luma samples and 1 chroma sample for each of the chroma components (e.g., Co and Cg, or Cb and Cr). Unlike in the 4:2:0 mode, there may be no distinction between partitions in the first line of the current block and partitions in the second line of the current block in the 4:2:2 mode. An example block prediction search 1800 for 1x2 partitions for 4:2:2 chroma sub-sampling is illustrated in **FIG. 18**. In the example of **FIG. 18**, the current

1x2 partition A is in the first line and the current 1x2 partition B is in the second line, and each of current partitions A and B includes a chroma site. Current partition A is predicted based candidate 1x2 partition A, which includes a chroma site in the first sample, and current partition B is predicted based on candidate 1x2 partition B, which includes a chroma site in the second sample. Thus, regardless of where the chroma site is located within the candidate partition, the chroma sample may be used to predict the chroma sample in the current partition.

Encoder Decision

[0136] In the 4:2:2 and 4:2:0 formats, there may be fewer than 4 entropy coding groups per block for each chroma component. For example, four entropy coding groups may be used for the luma component, and two (or one) entropy coding groups may be used for the orange chroma component, and two (or one) entropy coding groups may be used for the green chroma component. The number of entropy coding groups used for coding a given block may be determined based on the number of luma or chroma samples in the given block. In some embodiments, the entropy coding groups are determined by the encoder based on the coding mode in which a given block is coded. In other embodiments, the entropy coding groups are set by the applicable coding standard (e.g., based on the coding mode in which the given block is coded).

[0137] In some embodiments, the quantity EC_{bits} is not determined exactly by the encoder for chroma. In some of such embodiments, the encoder may determine whether to use 1x2 or 2x2 partitions, based on the entropy coding rate calculated using only the luma samples for 4:2:2 and 4:2:0 formats. In other embodiments, the quantity EC_{bits} is determined by the encoder for chroma, and the encoder may determine whether to use 1x2 or 2x2 partitions, based on the entropy coding rate calculated using both luma and chroma samples for 4:2:2 and 4:2:0 formats.

Signaling

[0138] In some embodiments, the number of entropy coding groups to be transmitted from the encoder to the decoder for each block or for each color component may be changed depending on the chroma sub-sampling format. In some implementations, the number of entropy coding groups is changed to ensure that the codec throughput is sufficiently high. For example, in the 4:4:4 mode, a 2x8 block may include four entropy coding groups, as illustrated in **FIG. 11**. In such an example, four

entropy coding groups may be used (e.g., signaled by the encoder) for each color component (e.g., Y, Co, and Cg). **Table 1** describes example changes to the number of entropy coding groups used for the 4:2:2 and 4:2:0 modes. The remainder of the signaling described above (e.g., signaling of the BPVs, signaling of the indication of the partitioning scheme, etc.) may be unchanged (from the signaling described with respect to the 4:4:4 mode) for the 4:2:2 and 4:2:0 modes. For example, in **Table 1**, component 0 may correspond to luma (Y), component 1 may correspond to orange chroma (Co), and component 2 may correspond to green chroma (Cg).

Chroma format	Component 0	Component 1	Component 2
4:4:4	4	4	4
4:2:2	4	2	2
4:2:0	4	1	1

Table 1: number of entropy coding groups per component for different chroma sub-sampling formats (assuming a block size of 2x8)

Advantages

[0139] One or more block prediction mode techniques described in the present disclosure may be implemented using an asymmetrical design. The asymmetric design allows more expensive procedures to be performed on the encoder side, decreasing complexity of the decoder. For example, because the vector(s) are explicitly signaled to the decoder, the encoder does the majority of the work compared with the decoder. This is desirable as the encoder is often part of a System on a Chip (SoC) design, running at a high frequency on a cutting-edge process node (e.g., 20nm and below). Meanwhile, the decoder is likely to be implemented on a Display Driver Integrated Circuit (DDIC) chip-on-glass (COG) solution with a limited clock speed and a much larger process size (e.g., 65nm and above).

[0140] Additionally, the adaptive selection of block partition sizes allows the block prediction mode to be used for a broader range of content types. Since signaling the BPVs explicitly can be expensive, the variable partition size allows for reduced signaling cost for image regions which can be well-predicted using a 2x2 partition. For highly complex regions, the 1x2 partition size can be selected if either the entropy coding rate can be sufficiently reduced to make up for the higher signaling cost, or if distortion

can be sufficiently reduced such that the RD tradeoff is still in favor of 1x2. For example, the adaptive selection of block partition sizes may increase performance across all content types, including natural images, test patterns, fine text rendering, etc. In some embodiments, the adaptive partitioning techniques discussed herein may be extended by considering block partition sizes larger than 2x2 and/or block sizes larger than 2x8.

[0141] One or more techniques described herein may be implemented in a fixed-bit codec employing a constant bit rate buffer model. Such a model, bits stored in the rate buffer are removed from the rate buffer at a constant bit rate. Thus, if the video encoder adds too many bits to the bitstream, the rate buffer may overflow. On the other hand, the video encoder may need to add enough bits in order to prevent underflow of the rate buffer. Further, on the video decoder side, the bits may be added to rate buffer at a constant bit rate, and the video decoder may remove variable numbers of bits for each block. To ensure proper decoding, the rate buffer of the video decoder should not “underflow” or “overflow” during the decoding of the compressed bitstream. The one or more techniques described herein may ensure that such underflow or overflow is prevented during encoding and/or decoding. In some embodiments, the encoder may operate under a bit-budget constraint, in which the encoder has a fixed number of bits to code a given region, slice, or frame. In such embodiments, being able to know exactly (and not having to estimate) how many bits each one of a plurality of coding modes would need to be able to code a given region, slice, or frame is critical to the encoder, so that the encoder can ensure that the bit-budget or other bit/bandwidth related constraints can be satisfied. For example, the encoder may code the given region, slice, or frame in a given coding mode without having to implement any precautionary measures in case the coding of the given region, slice, or frame requires more bits than estimated.

[0142] Further, one or more techniques described herein overcome specific technical problems associated with the video compression technology in transmission over display links. By allowing a region to be coded based on multiple candidate regions (e.g., each partition in the region predicted based on the corresponding one of the multiple candidate regions), video encoders and decoders can provide a customized prediction based on the nature of the region (e.g., smooth, complex, etc.), thereby improving the video encoder and decoder (e.g., hardware and software codecs) performance.

Other Considerations

[0143] Information and signals disclosed herein may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0144] The various illustrative logical blocks, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0145] The techniques described herein may be implemented in hardware, software, firmware, or any combination thereof. Such techniques may be implemented in any of a variety of devices such as general purposes computers, wireless communication device handsets, or integrated circuit devices having multiple uses including application in wireless communication device handsets and other devices. Any features described as devices or components may be implemented together in an integrated logic device or separately as discrete but interoperable logic devices. If implemented in software, the techniques may be realized at least in part by a computer-readable data storage medium comprising program code including instructions that, when executed, performs one or more of the methods described above. The computer-readable data storage medium may form part of a computer program product, which may include packaging materials. The computer-readable medium may comprise memory or data storage media, such as random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like. The techniques additionally, or alternatively, may be realized at least in part by a computer-readable communication medium that

carries or communicates program code in the form of instructions or data structures and that can be accessed, read, and/or executed by a computer, such as propagated signals or waves.

[0146] The program code may be executed by a processor, which may include one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Such a processor may be configured to perform any of the techniques described in this disclosure. A general purpose processor may be a microprocessor; but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. Accordingly, the term “processor,” as used herein may refer to any of the foregoing structure, any combination of the foregoing structure, or any other structure or apparatus suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated software or hardware configured for encoding and decoding, or incorporated in a combined video encoder-decoder (CODEC). Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0147] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of inter-operative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0148] Although the foregoing has been described in connection with various different embodiments, features or elements from one embodiment may be combined with other embodiments without departing from the teachings of this disclosure. However, the combinations of features between the respective embodiments are not necessarily limited thereto. Various embodiments of the disclosure have been described. These and other embodiments are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method for coding a block of video data in block prediction mode of a constant bitrate video coding scheme, the method comprising:

determining one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region, wherein the one or more first candidate regions are stored in a memory of a video encoding device;

determining one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region, wherein the one or more second candidate regions are stored in the memory of the video encoding device;

determining whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and

in response to determining that the first cost is greater than the second cost, coding the current region based on the one or more second candidate regions into a bitstream.

2. The method of claim 1, further comprising signaling, in the bitstream, one or more prediction vectors indicative of a location of the one or more second candidate regions with respect to the current region and a residual indicative of a difference between the one or more second candidate regions and the current region.

3. The method of claim 2, wherein the residual comprises a luma residual and a chroma residual, the method further comprising (i) performing entropy coding on the luma residual using a first number of entropy coding groups, and (ii) performing entropy coding on the chroma residual using a second number of entropy coding groups that is different from the first number.

4. The method of claim 2, further comprising signaling, in the bitstream, one or more prediction vectors indicative of a location of one or more first candidate regions

with respect to at least one other region within the block, wherein a number of prediction vectors signaled for the at least one other region is different from a number of prediction vectors signaled for the current region.

5. The method of claim 1, wherein the one or more first candidate regions and the one or more second candidate regions comprise the same candidate region or regions.

6. The method of claim 1, further comprising determining the second cost based at least in part on (i) a sum of absolute differences between the current region and the one or more second candidate regions and (ii) a number of bits needed to signal the one or more prediction vectors and corresponding residuals in the bitstream.

7. The method of claim 1, further comprising signaling a partition indicator in the bitstream, the partition indicator indicative of a partitioning scheme associated with each region within the block, the block comprising at least one region other than the current region.

8. The method of claim 7, wherein the partition indicator indicates that current region is associated with the second partitioning scheme and that the at least one region other than the current region in the block is associated with the first partitioning scheme different from the second partitioning scheme.

9. The method of claim 1, wherein the one or more first candidate regions and the one or more second candidate regions have different sizes.

10. The method of claim 1, wherein the one or more first candidate regions and the one or more second candidate regions have the same sizes.

11. The method of claim 1, wherein the block of video data is associated with one of a 4:2:0 chroma sub-sampling format or a 4:2:2 chroma sub-sampling format.

12. The method of claim 1, wherein the one or more second candidate regions include a first region having a first number of chroma samples and a second region having a second number of chroma samples that is different from the first number, wherein the first and second regions do not overlap.

13. The method of claim 1, wherein the one or more second candidate regions include at least one region that is not associated with any chroma samples.

14. The method of claim 1, wherein the current region includes a first current region associated with a first number of chroma samples and a second current region associated with a second number of chroma samples that is different from the first number, wherein the one or more second candidate regions include a first one of the one or more second candidate regions including the first number of chroma samples and a second one of the one or more second candidate regions including the second number of chroma samples.

15. An apparatus configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme, the apparatus comprising:

a memory configured to store video data associated with one or more candidate regions;

one or more processors in communication with the memory, the one or more processors configured to:

determine one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region;

determine one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region;

determine whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and

in response to determining that the first cost is greater than the second cost, code the current region based on the one or more second candidate regions into a bitstream.

16. The apparatus of claim 15, wherein the one or more processors are further configured to signal, in the bitstream, one or more prediction vectors indicative of a location of the one or more second candidate regions with respect to the current region and a residual indicative of a difference between the one or more second candidate regions and the current region.

17. The apparatus of claim 16, the residual comprises a luma residual and a chroma residual, the method further comprising (i) performing entropy coding on the luma residual using a first number of entropy coding groups, and (ii) performing entropy coding on the chroma residual using a second number of entropy coding groups that is different from the first number.

18. The apparatus of claim 16, wherein the one or more processors are further configured to signal, in the bitstream, one or more prediction vectors indicative of a location of one or more first candidate regions with respect to at least one other region within the block, wherein a number of prediction vectors signaled for the at least one other region is different from a number of prediction vectors signaled for the current region.

19. The apparatus of claim 15, wherein the one or more first candidate regions and the one or more second candidate regions comprise the same candidate region or regions.

20. The apparatus of claim 15, wherein the one or more processors are further configured to determine the second cost based at least in part on (i) a sum of absolute differences between the current region and the one or more second candidate regions and (ii) a number of bits needed to signal the one or more prediction vectors and corresponding residuals in the bitstream.

21. The apparatus of claim 15, wherein the one or more processors are further configured to signal a partition indicator in the bitstream, the partition indicator indicative of a partitioning scheme associated with each region within the block, the block comprising at least one region other than the current region, wherein the partition indicator indicates that current region is associated with the second partitioning scheme and that the at least one region other than the current region in the block is associated with the first partitioning scheme different from the second partitioning scheme.

22. The apparatus of claim 15, wherein the one or more second candidate regions include a first region having a first number of chroma samples and a second region having a second number of chroma samples that is different from the first number, wherein the first and second regions do not overlap.

23. The apparatus of claim 15, wherein the one or more second candidate regions include at least one region that is not associated with any chroma samples.

24. The apparatus of claim 15, wherein the current region includes a first current region associated with a first number of chroma samples and a second current region associated with a second number of chroma samples that is different from the first number, wherein the one or more second candidate regions include a first one of the one or more second candidate regions including the first number of chroma samples and a second one of the one or more second candidate regions including the second number of chroma samples.

25. Non-transitory physical computer storage comprising code configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme, wherein the code, when executed, causes an apparatus to:

determine one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region;

determine one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region;

determine whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and

in response to determining that the first cost is greater than the second cost, code the current region based on the one or more second candidate regions into a bitstream.

26. The non-transitory physical computer storage of claim 25, wherein the code further causes the apparatus to signal, in the bitstream, one or more prediction vectors indicative of a location of the one or more second candidate regions with respect to the current region and a residual indicative of a difference between the one or more second candidate regions and the current region.

27. The non-transitory physical computer storage of claim 25, wherein the code further causes the apparatus to signal a partition indicator in the bitstream, the partition indicator indicative of a partitioning scheme associated with each region within the block, the block comprising at least one region other than the current region, wherein the partition indicator indicates that current region is associated with the second partitioning scheme and that the at least one region other than the current region in the block is associated with the first partitioning scheme different from the second partitioning scheme.

28. A video coding device configured to code a block of video data in block prediction mode of a constant bitrate video coding scheme, the video coding device comprising:

means for determining one or more first candidate regions to be used to predict a current region within the block of video data based on a first partitioning scheme associated with block prediction mode, the one or more first candidate regions being within a first range of locations associated with the current region;

means for determining one or more second candidate regions to be used to predict the current region based on a second partitioning scheme associated with block prediction mode, the one or more second candidate regions being within a second range of locations associated with the current region;

means for determining whether a first cost associated with coding the current region based on the first partitioning scheme is greater than a second cost associated with coding the current region based on the second partitioning scheme; and

means for coding, in response to determining that the first cost is greater than the second cost, the current region based on the one or more second candidate regions into a bitstream.

29. The video coding device of claim 28, further comprising means for signaling, in the bitstream, one or more prediction vectors indicative of a location of the one or more second candidate regions with respect to the current region and a residual indicative of a difference between the one or more second candidate regions and the current region.

30. The video coding device of claim 28, further comprising means for signaling a partition indicator in the bitstream, the partition indicator indicative of a partitioning scheme associated with each region within the block, the block comprising at least one region other than the current region, wherein the partition indicator indicates that current region is associated with the second partitioning scheme and that the at least one region other than the current region in the block is associated with the first partitioning scheme different from the second partitioning scheme.

10

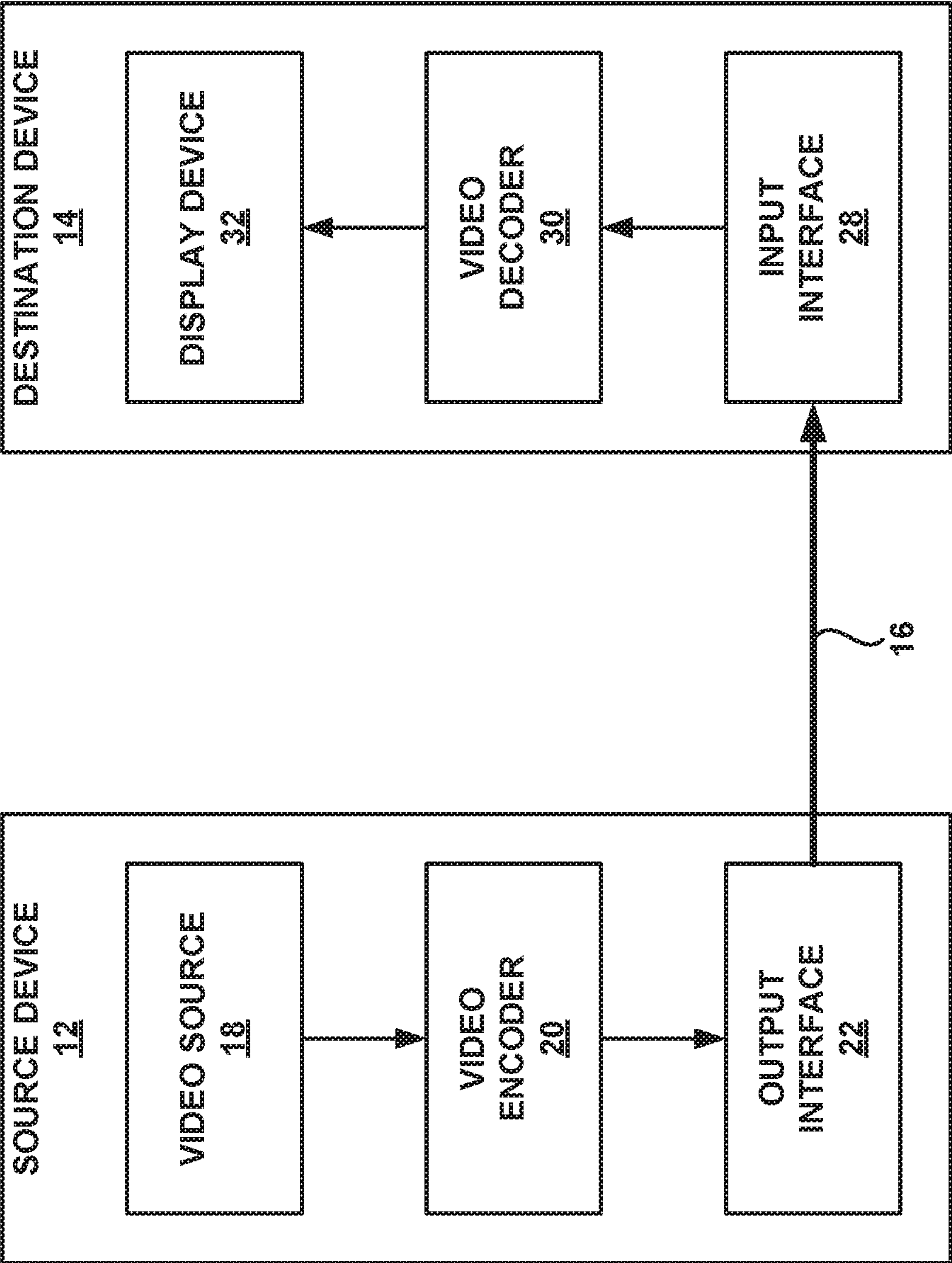


FIG. 1A

10'

2/20

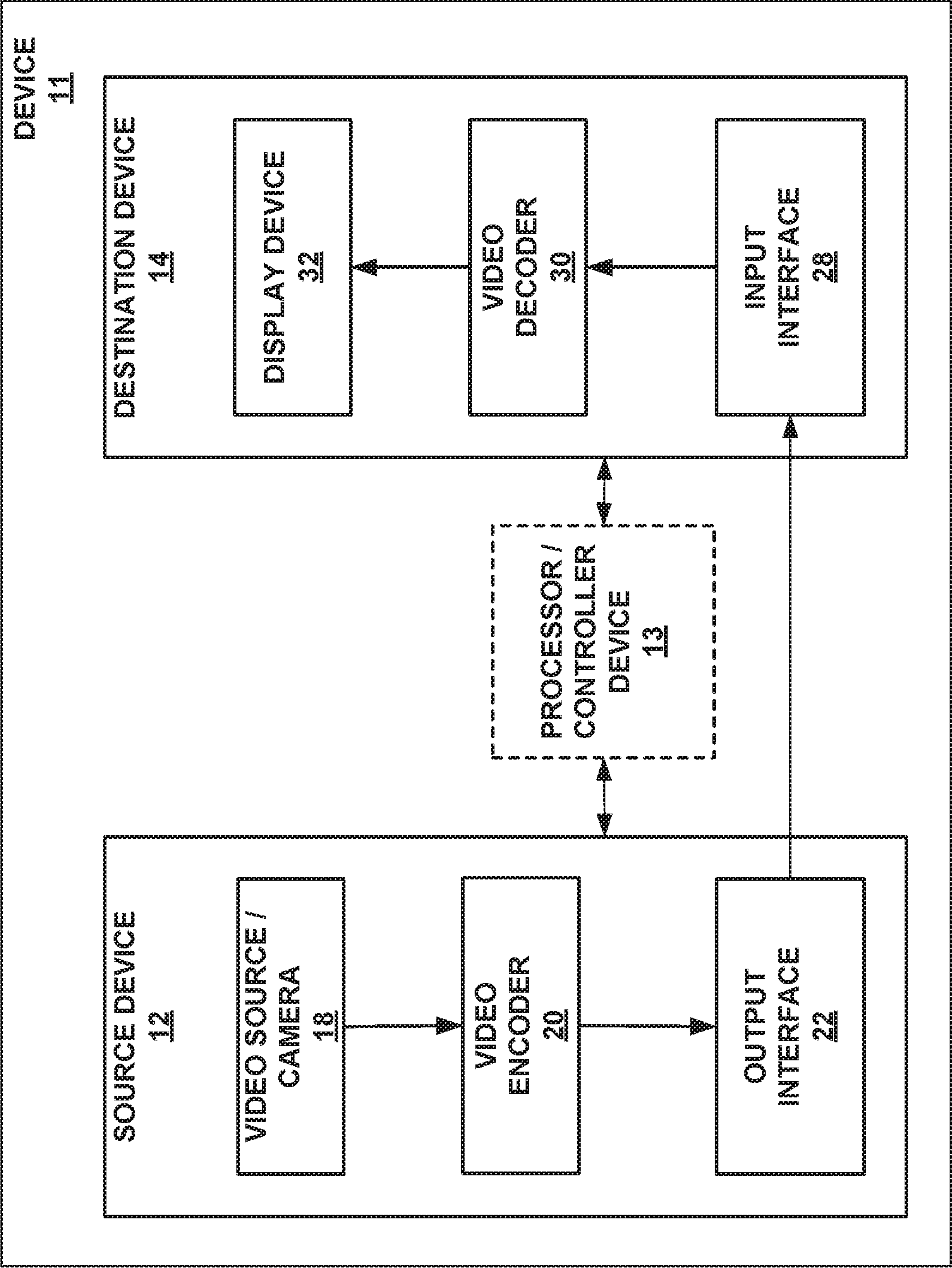


FIG. 1B

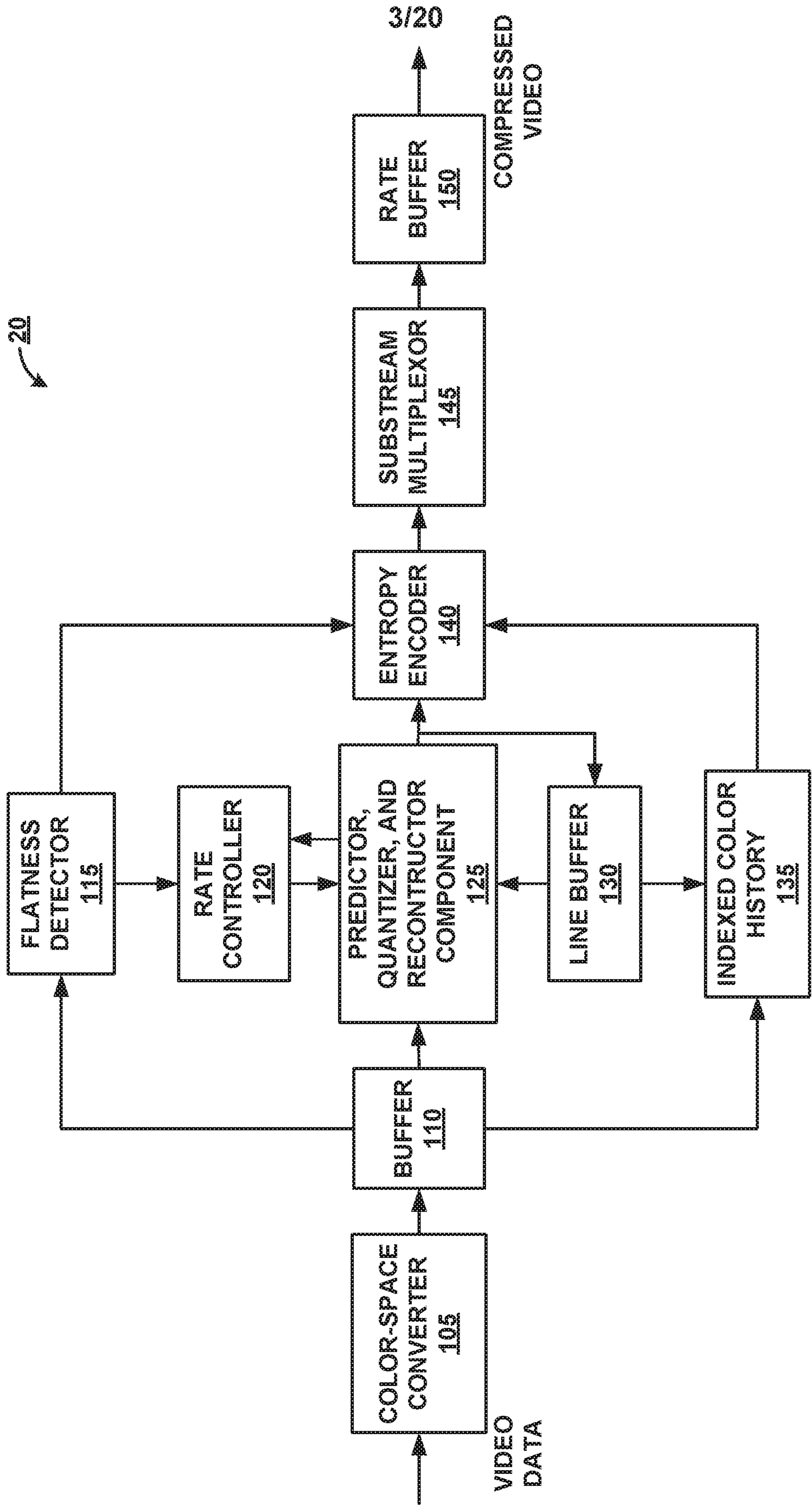


FIG. 2A

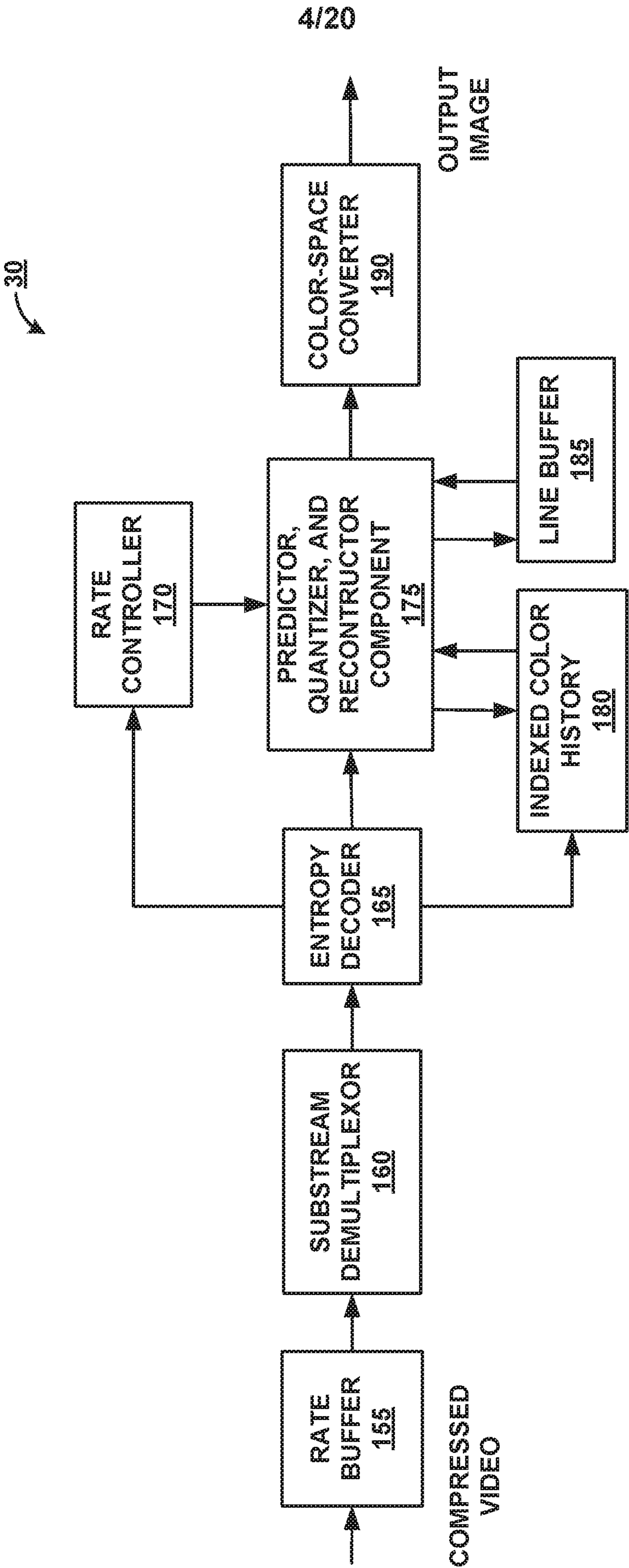


FIG. 2B

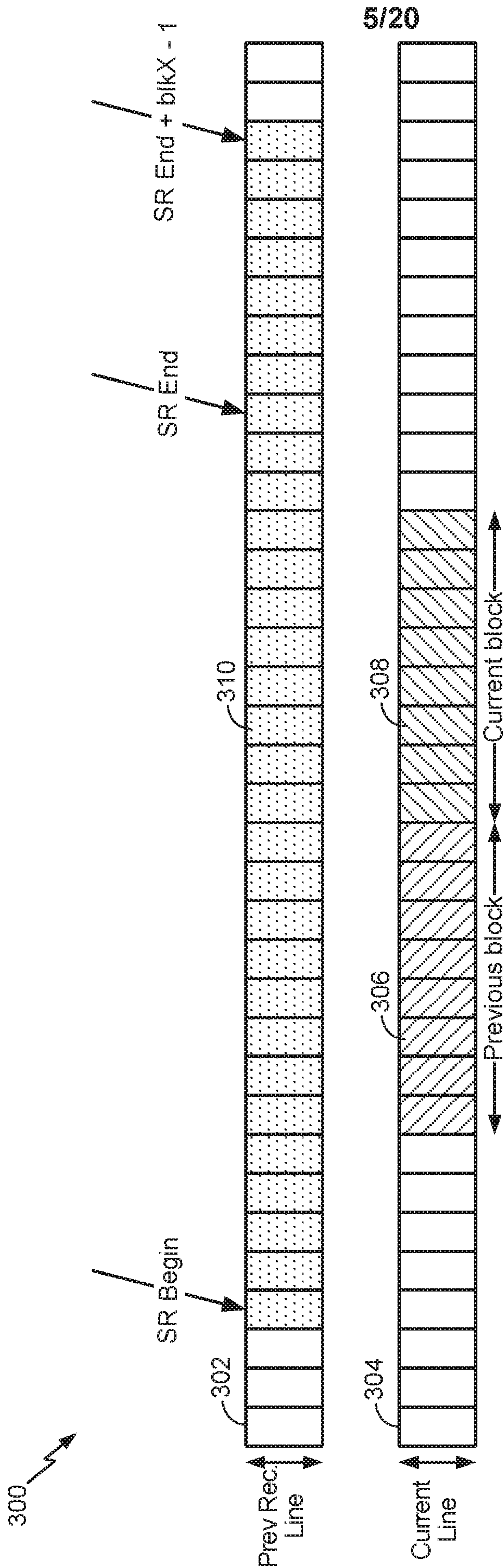


FIG. 3

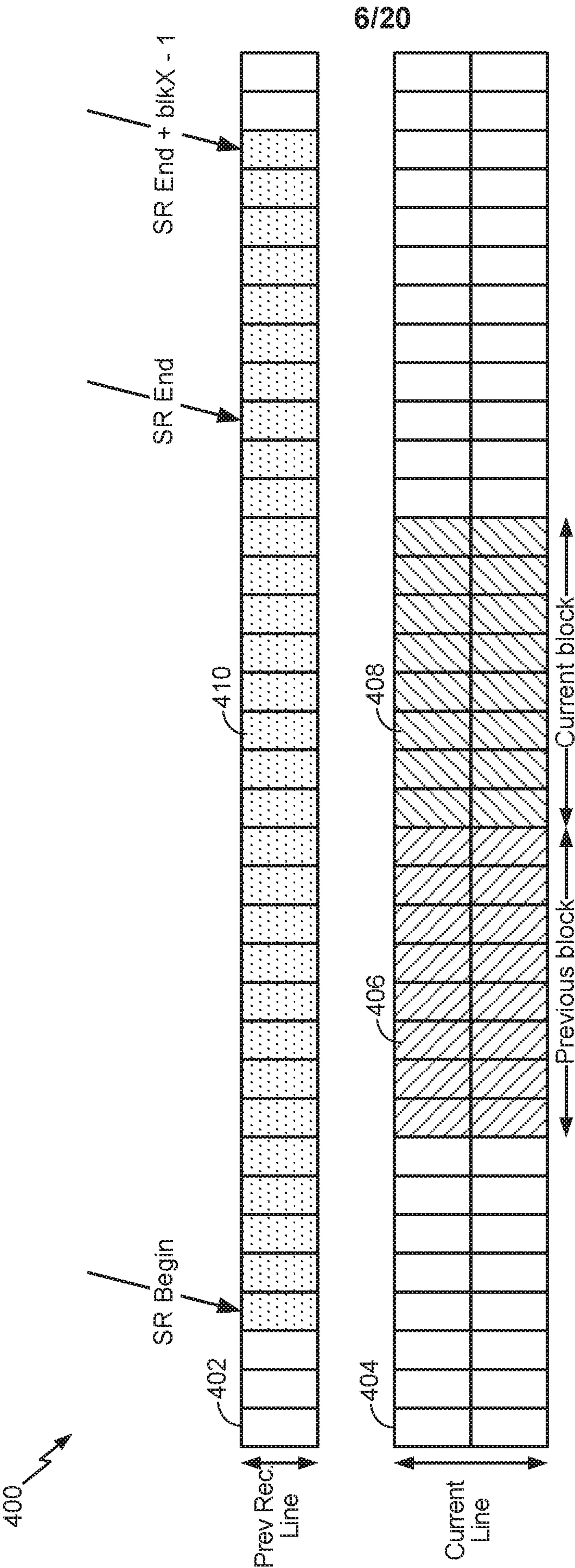


FIG. 4

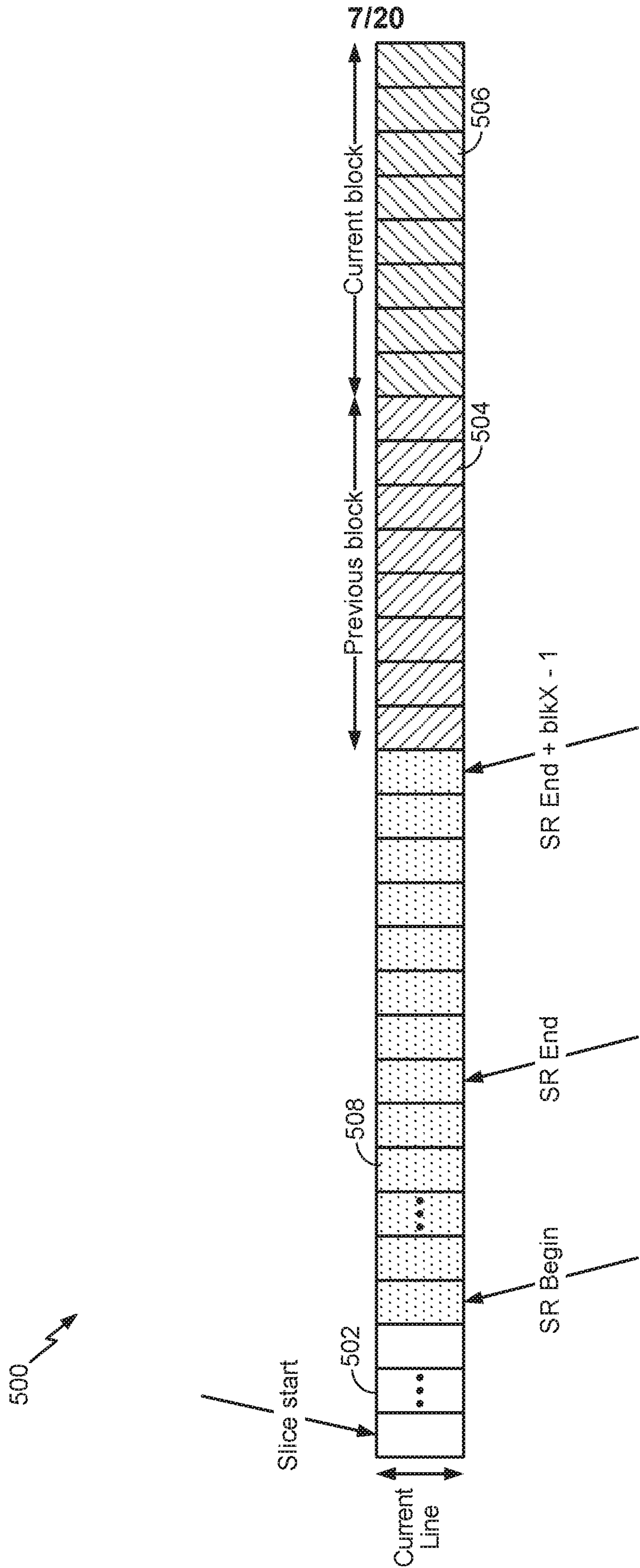
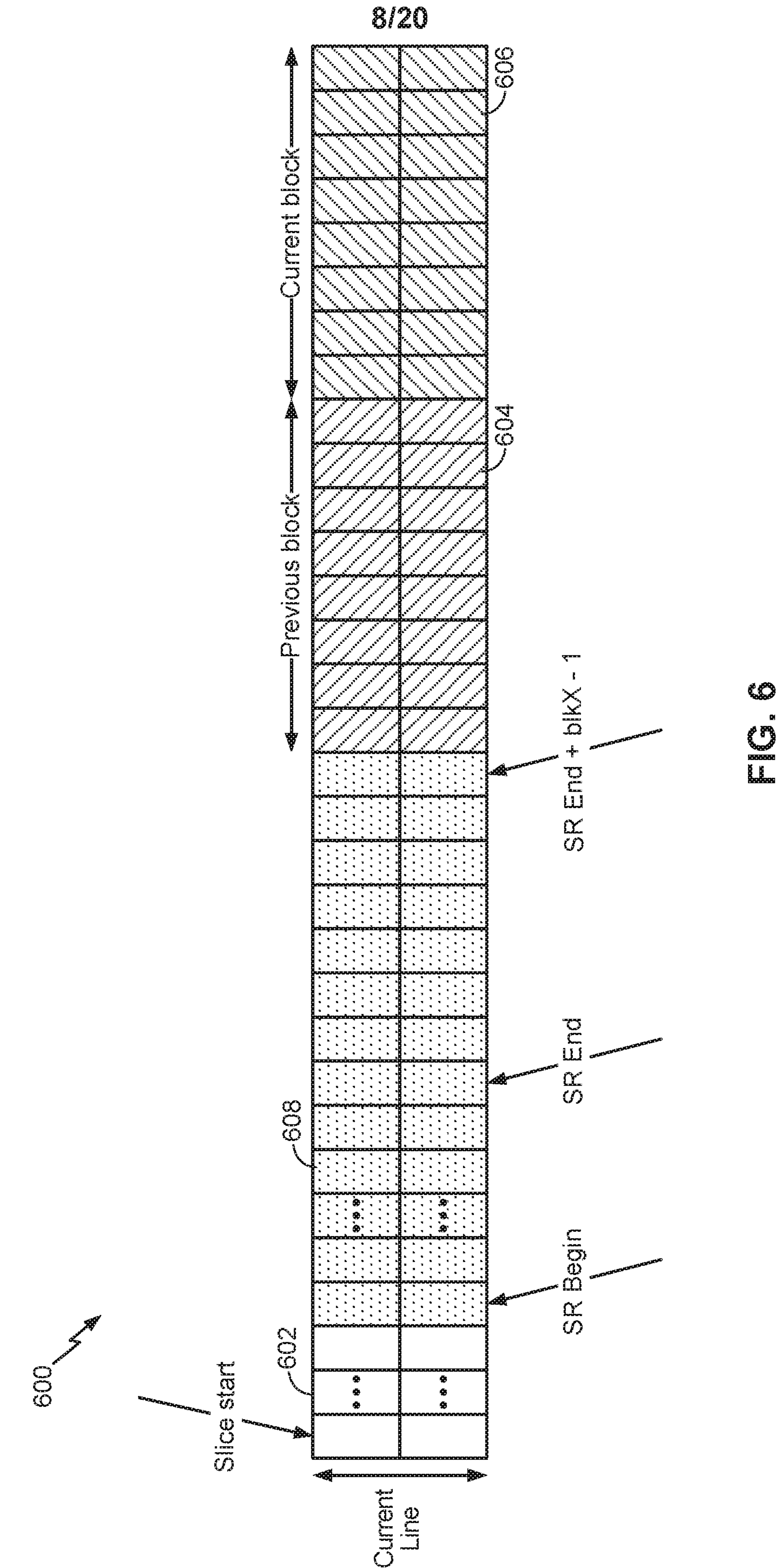


FIG. 5



9/20

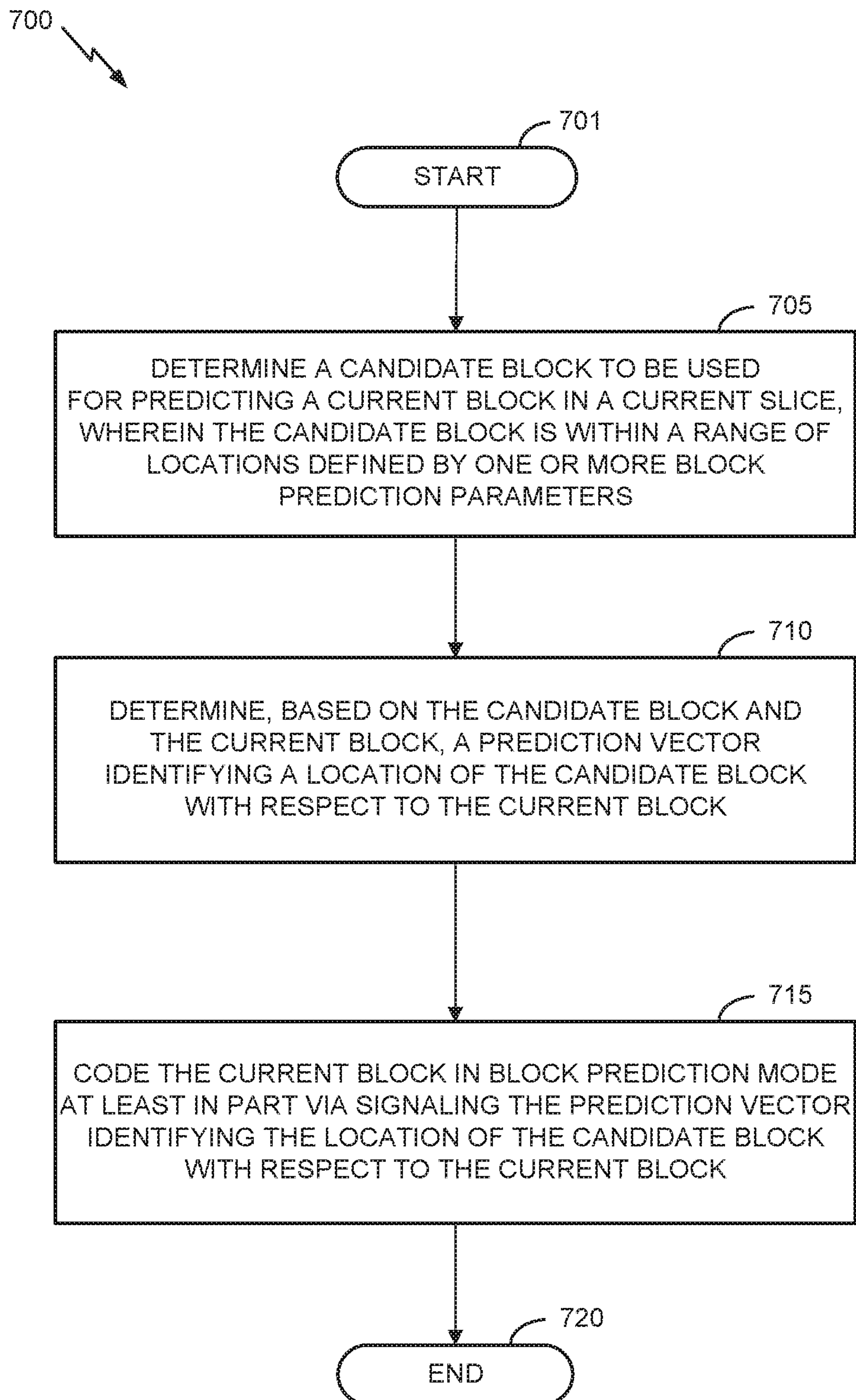


FIG. 7

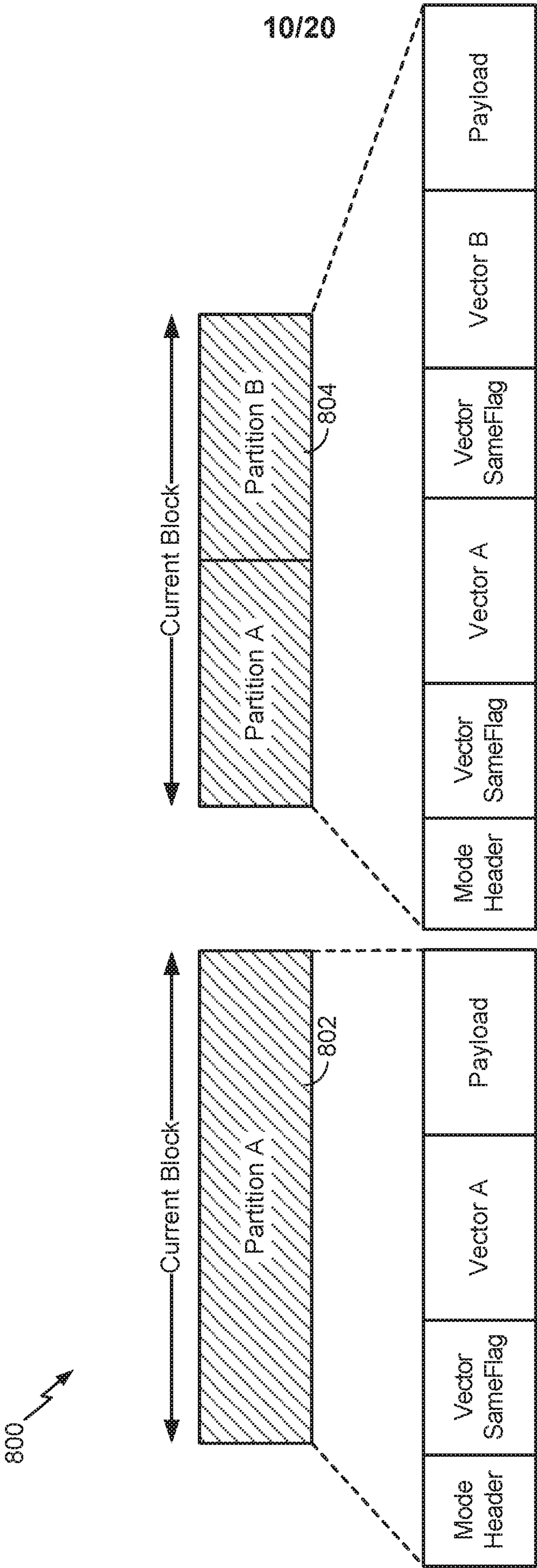
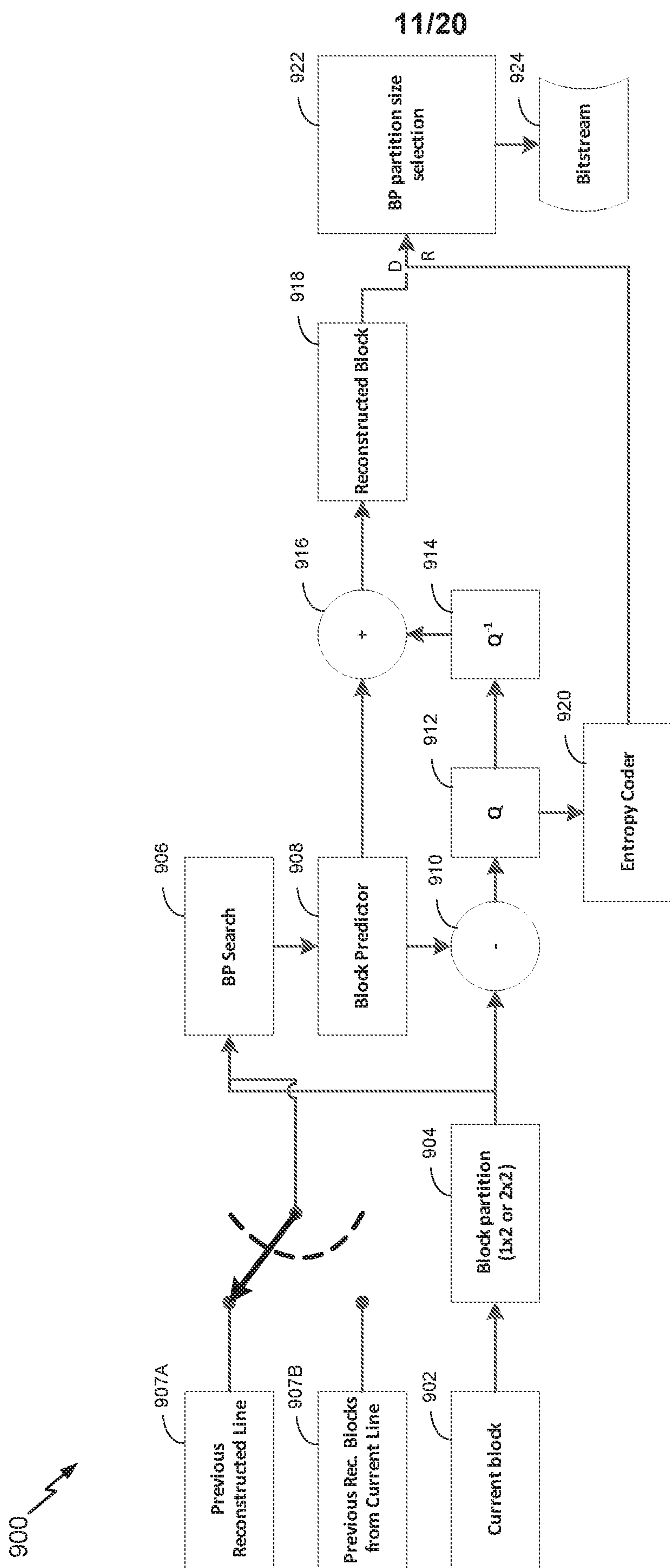


FIG. 8



၈၆၂

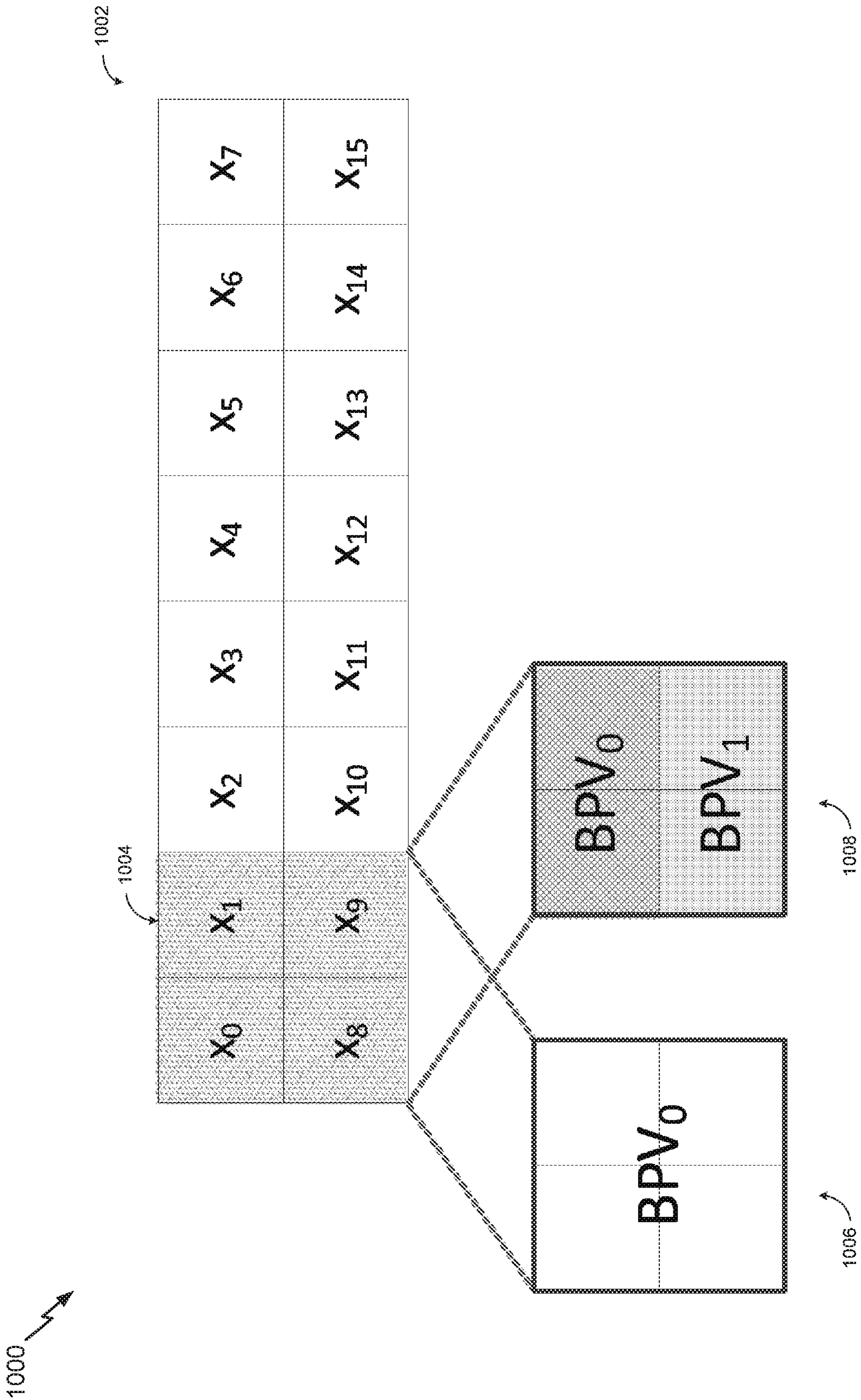


FIG. 10

1100 ↗

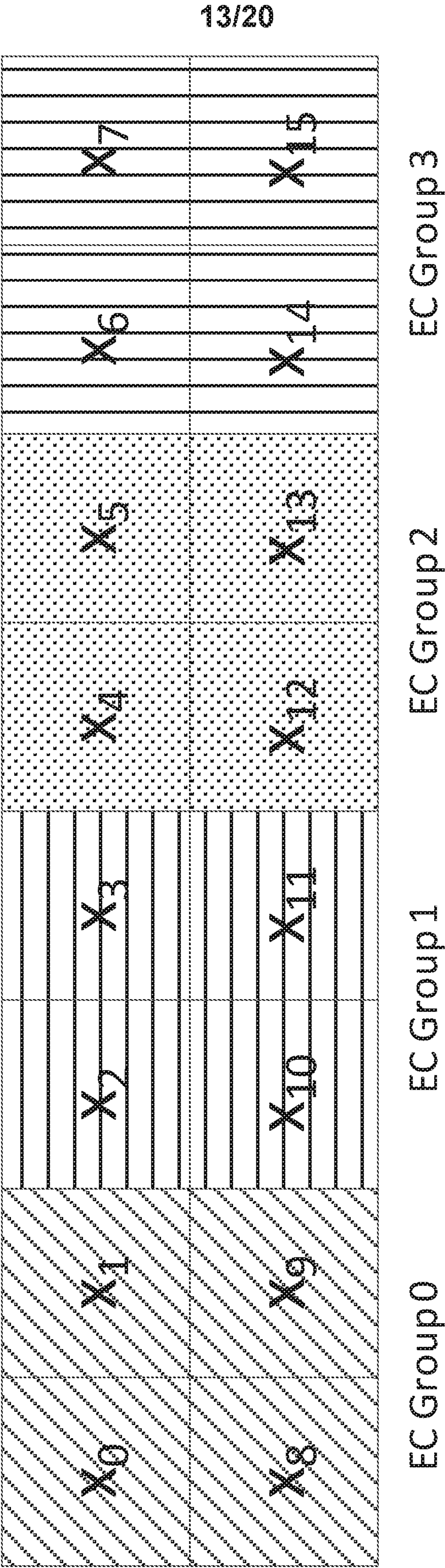


FIG. 11

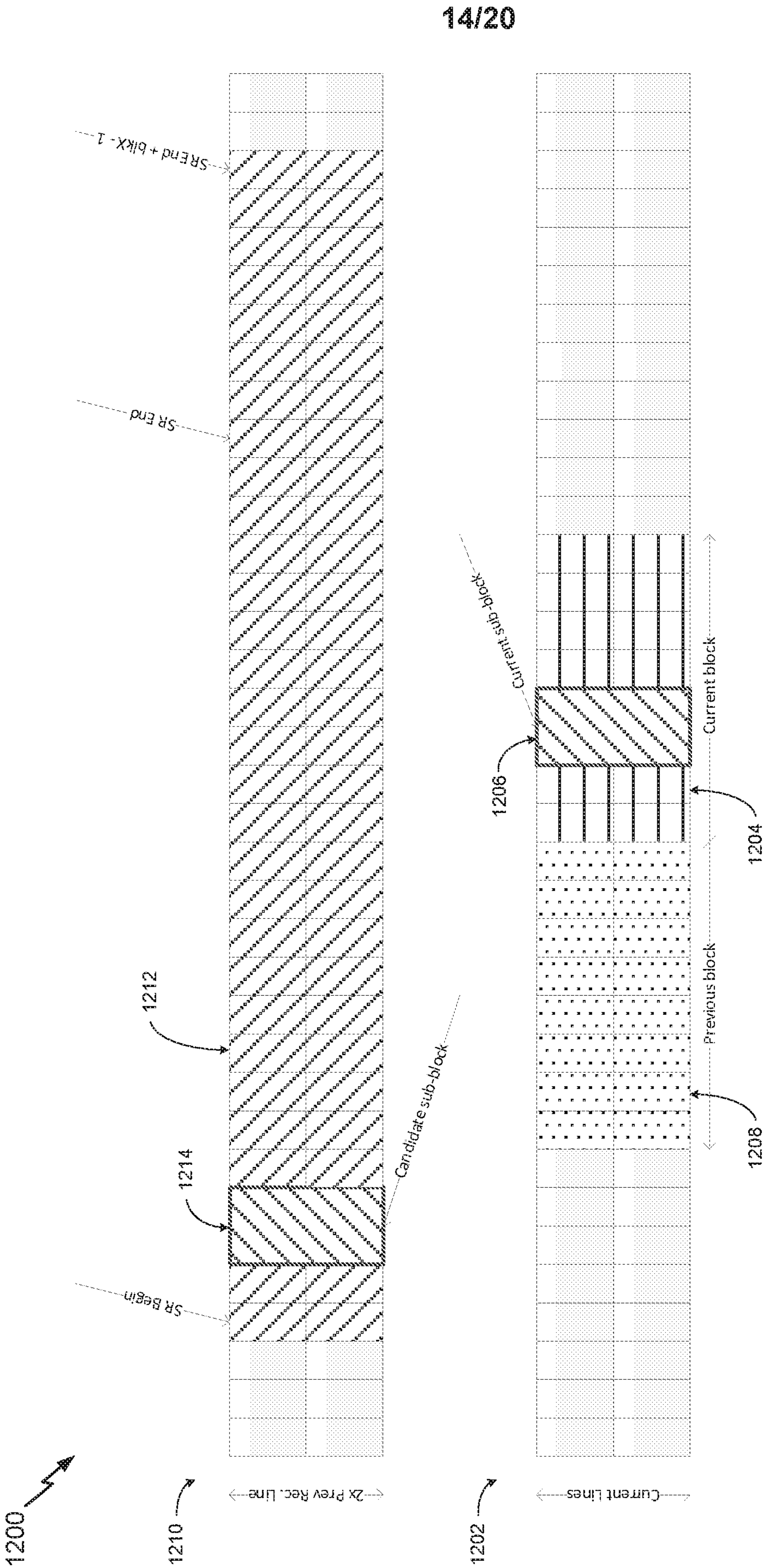


FIG. 12

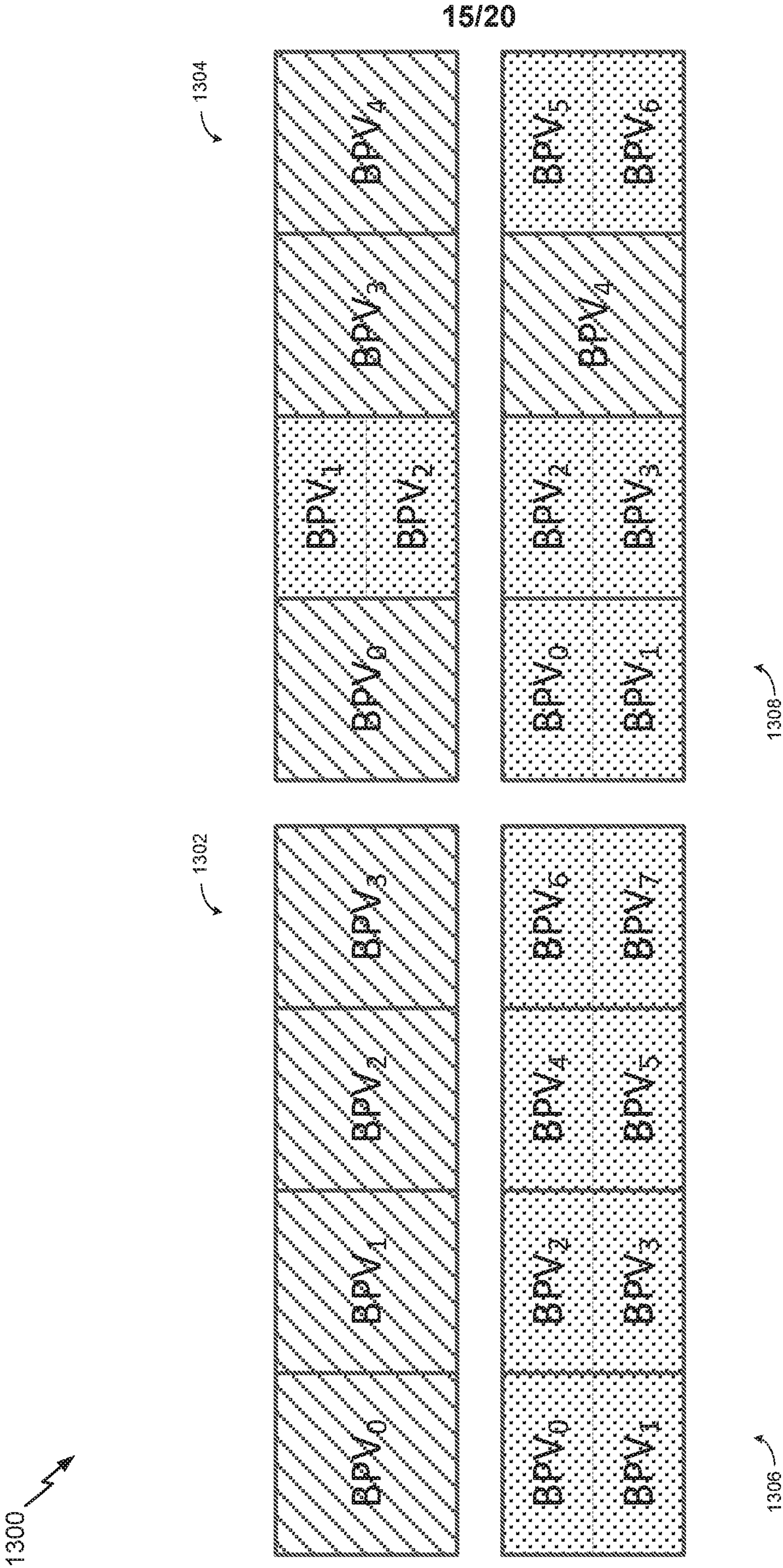


FIG. 13

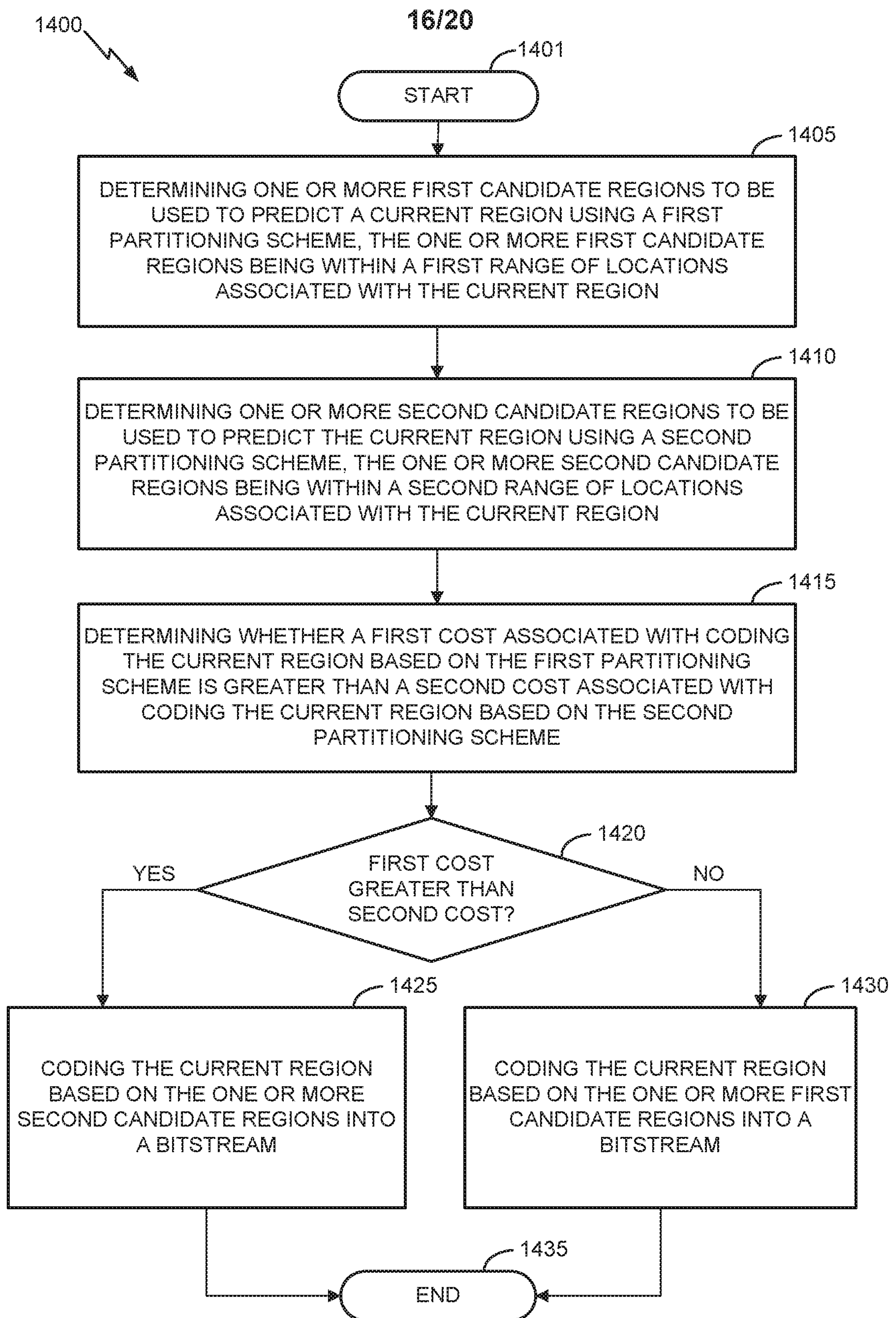


FIG. 14

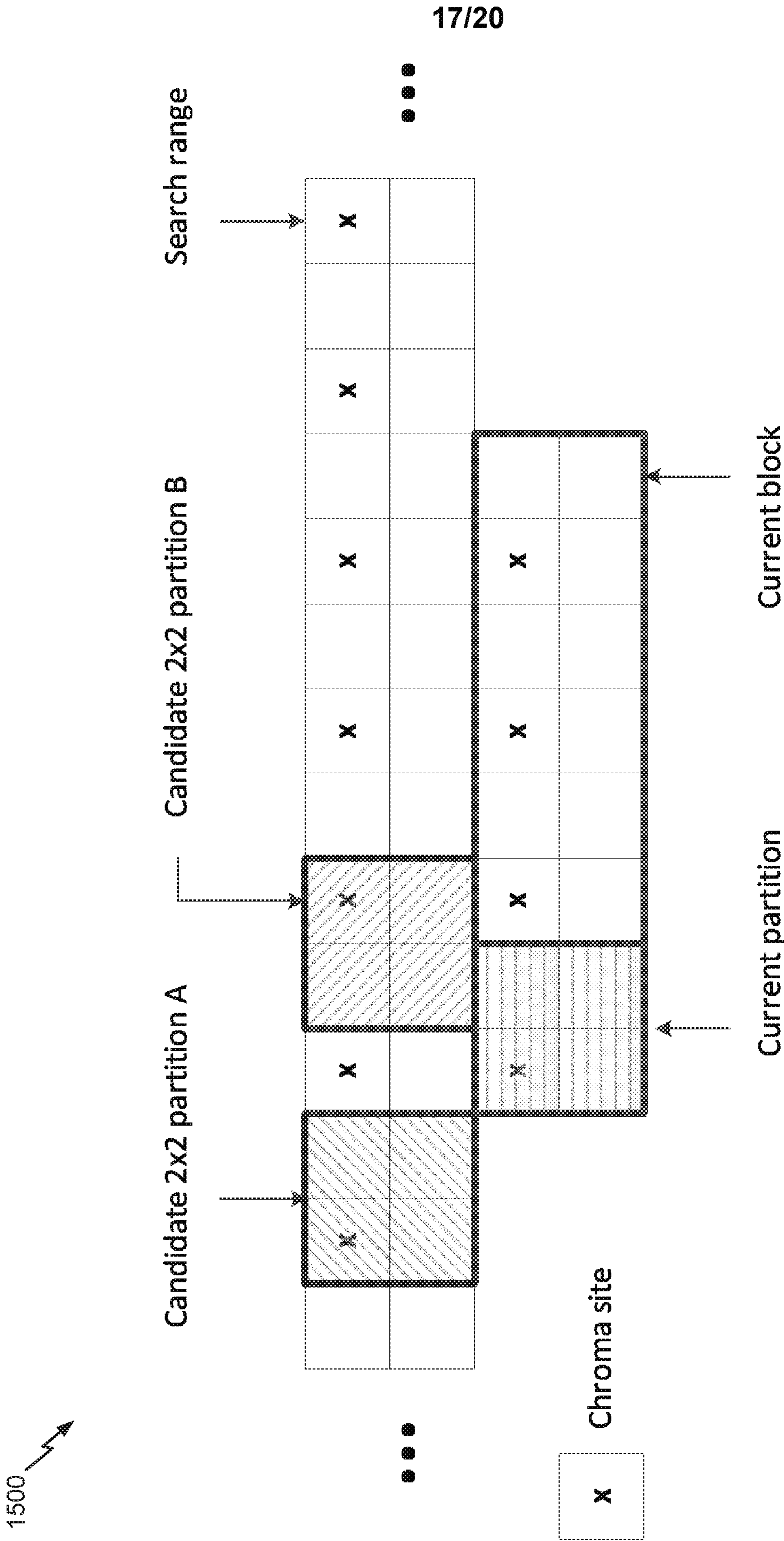
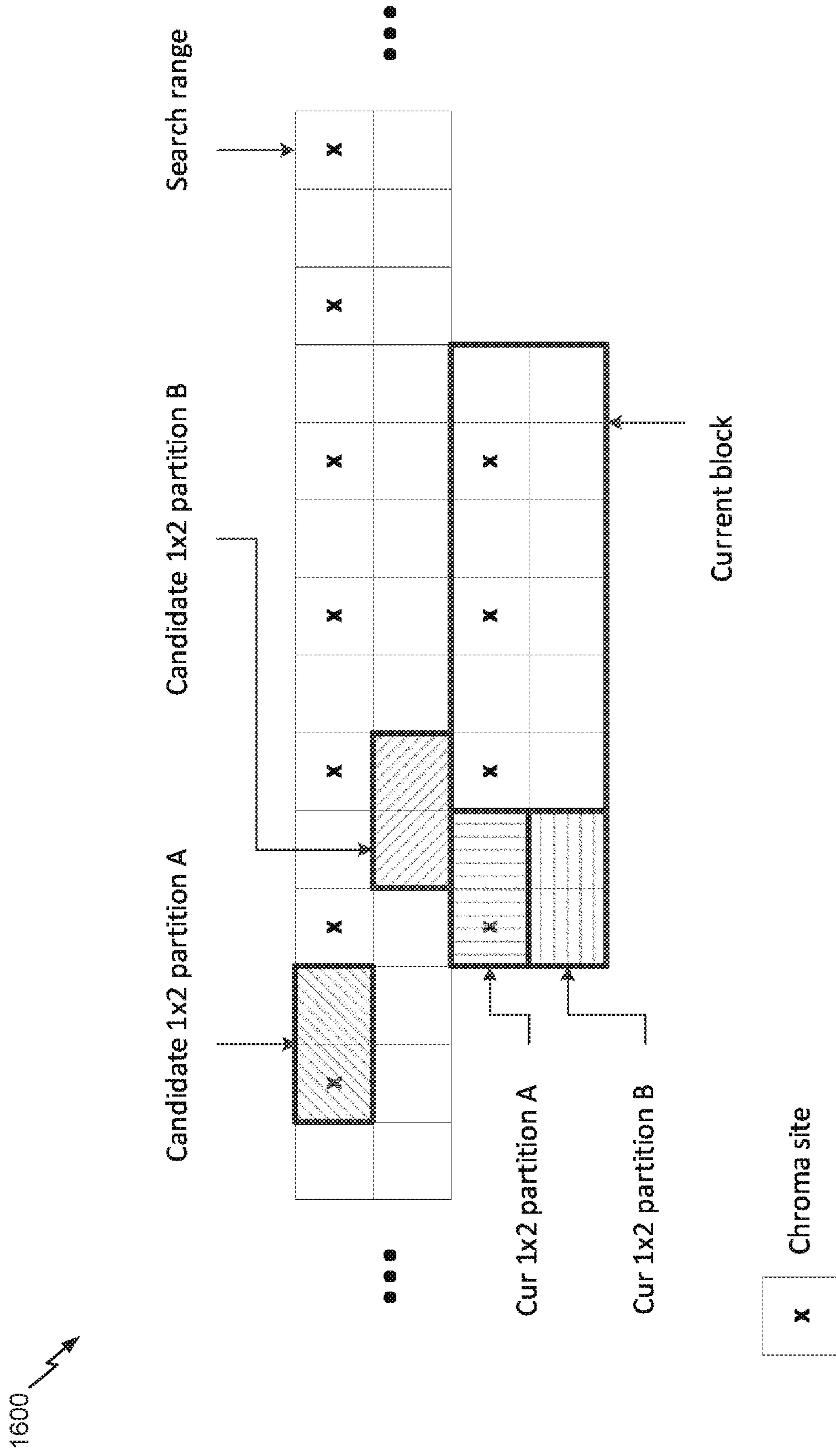
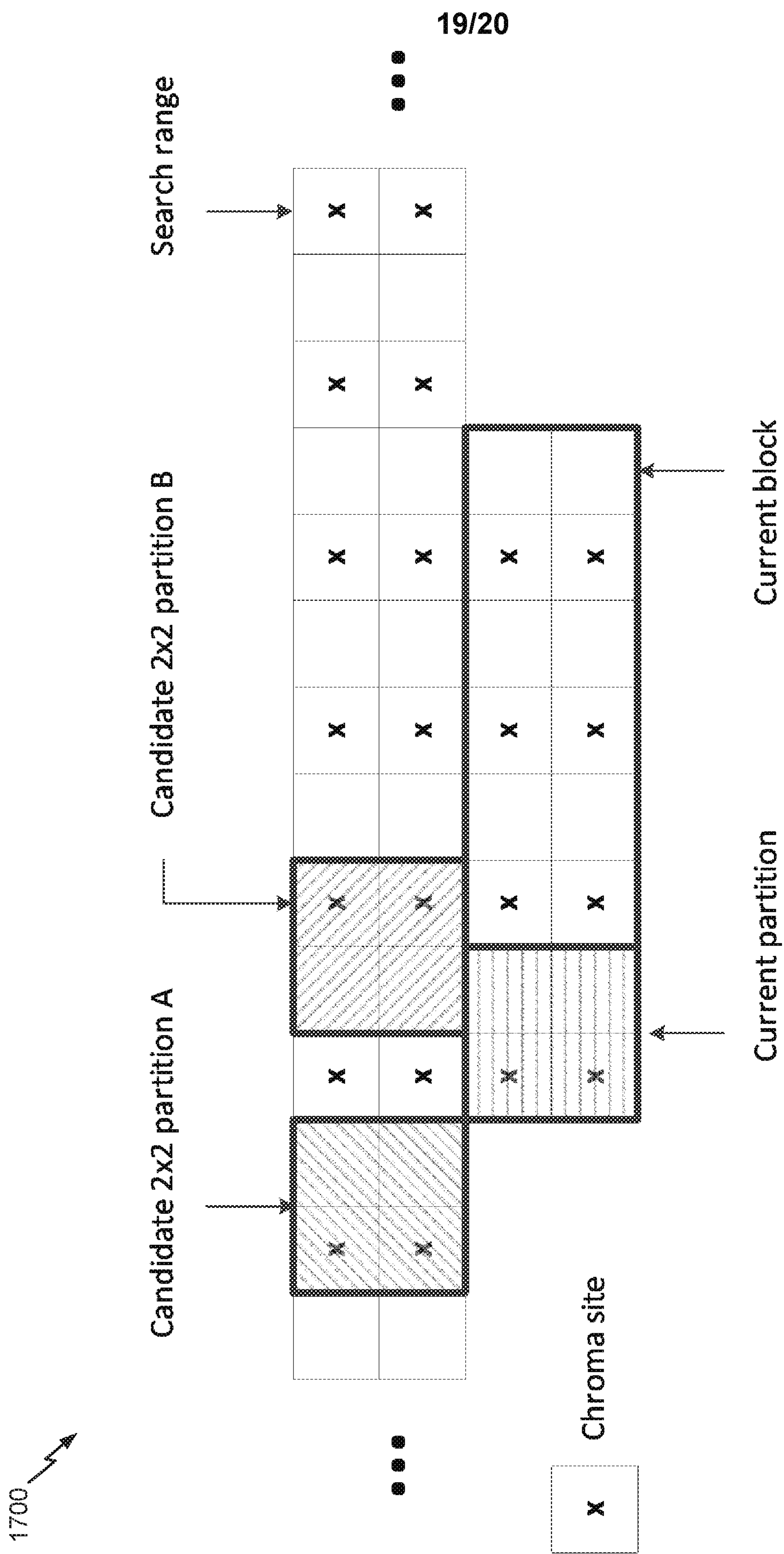
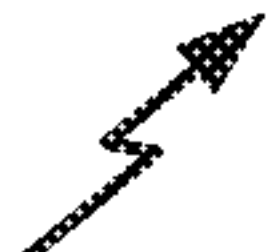


FIG. 15





715

1800 

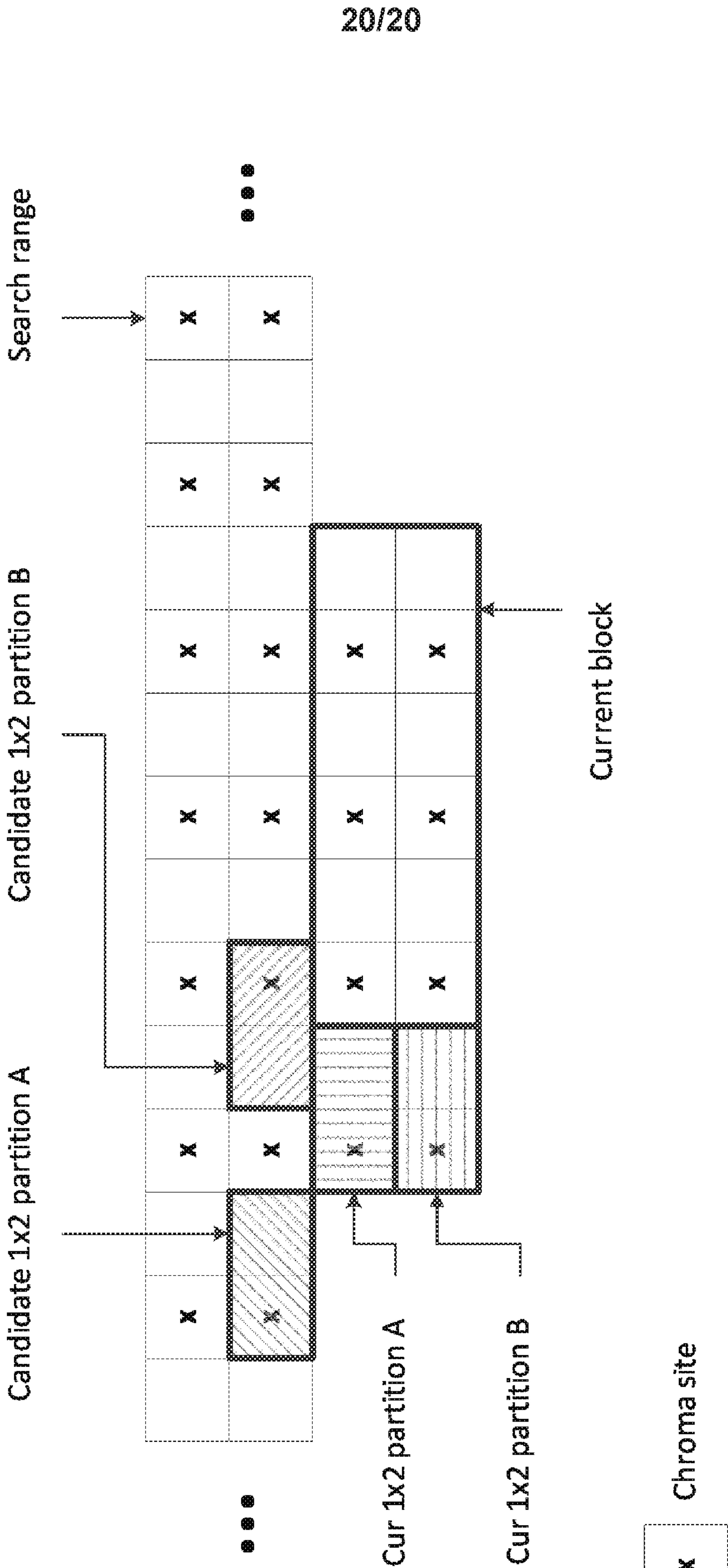


FIG. 18

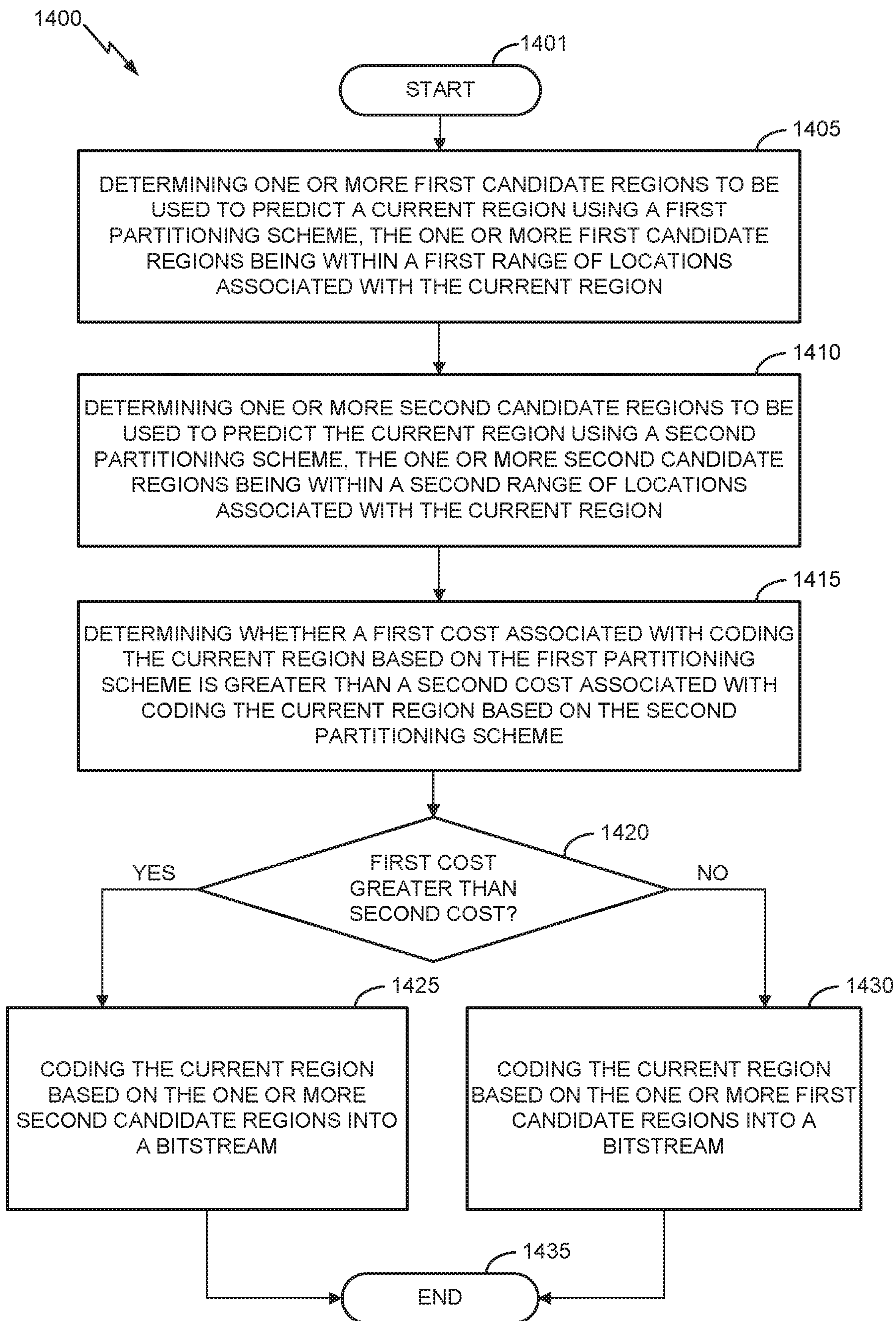


FIG. 14