



(51) International Patent Classification:
H04N 7/26 (2006.01) *G06T 3/40* (2006.01)
H04N 7/46 (2006.01)

(74) Agent: **PATEL, Ashish L.**; Attn: International Ip Administration, 5775 Morehouse Drive, San Diego, California 92121 (US).

(21) International Application Number:
PCT/US2010/043681

(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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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.

(22) International Filing Date:
29 July 2010 (29.07.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/511,349 29 July 2009 (29.07.2009) US

(71) Applicant (*for all designated States except US*): **QUALCOMM INCORPORATED** [US/US]; Attn: International Ip Administration, 5775 Morehouse Drive, San Diego, California 92121 (US).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **MARTINEZ BAUZA, Judit** [ES/US]; 5775 Morehouse Drive, San Diego, California 92121 (US). **VARADARAJAN, Srenivas** [IN/US]; 5775 Morehouse Drive, San Diego, California 92121 (US).

Declarations under Rule 4.17:

[Continued on next page]

(54) Title: SYSTEM AND METHOD OF COMPRESSING VIDEO CONTENT

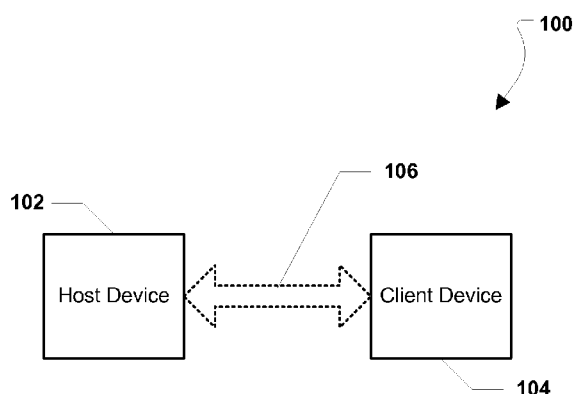


FIG. 1

(57) Abstract: A method of interpolating a pixel value is disclosed and may include locating a missing pixel. Further, the method may include determining a plurality of closest pixels, determining a value for each of the plurality of closest pixels, and determining a distance between the missing pixel and each of the plurality of closest pixels. The method may also include classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel and determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.



-
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
- Published:**
- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

SYSTEM AND METHOD OF COMPRESSING VIDEO CONTENT

RELATED APPLICATIONS

[0001] The present application incorporates by reference in its entirety co-pending U.S. Patent Application No. 12/511,329, entitled SYSTEM AND METHOD OF COMPRESSING VIDEO CONTENT, filed on even date with U.S. Patent Application No. 12/511,349 to which this application claims priority.

DESCRIPTION OF THE RELATED ART

[0002] When an external display or projector is connected to a mobile device using a wireless link, the data has to be transferred through a wireless channel. If the capacity of the channel is not large enough to handle the data transfer, the source, i.e., raw video, may be compressed. The capacity may not be large enough due to technical constraints of the physical layer of the radio link, e.g., interference that prevents the channel from being used to fully capacity. The capacity may also not be large enough if the channel is shared among other devices. Or, the capacity may not be large enough if a portion of the channel bandwidth is reserved for error mitigation or correction. Current compression methods have various limitations.

[0003] Therefore, what is needed is an improved system and method for compressing video content.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] In the figures, like reference numerals refer to like parts throughout the various views unless otherwise indicated.

[0005] FIG. 1 is a diagram of a wireless system;

[0006] FIG. 2 is a diagram of a telephone;

[0007] FIG. 3 is a flowchart illustrating a method of compressing graphic content and video content for transmission via a wireless link is shown;

[0008] FIG. 4 is a flowchart illustrating a method of detecting image edges;

- [0009] FIG. 5 is a flowchart illustrating a first aspect of a method of selectively refining image corners;
- [0010] FIG. 6 is a flowchart illustrating a second aspect of a method of selectively refining image corners;
- [0011] FIG. 7 is a flowchart illustrating a method of pattern matching;
- [0012] FIG. 8 is a diagram illustrating text block corners;
- [0013] FIG. 9 is a flowchart illustrating a method of run length encoding;
- [0014] FIG. 10 is a flowchart illustrating a method of obtaining image midpoints;
- [0015] FIG. 11 is a flowchart illustrating a method of refining midpoints;
- [0016] FIG. 12 is a diagram illustrating a plurality of pixels;
- [0017] FIG. 13 is a flowchart illustrating a method of recovering midpoints;
- [0018] FIG. 14 is a flowchart illustrating a method of inter-edge/inter-midpoint prediction;
- [0019] FIG. 15 is a diagram illustrating a plurality of pixels;
- [0020] FIG. 16 is a flowchart illustrating a method of encoding horizontal and vertical midpoints;
- [0021] FIG. 17 is a flowchart illustrating a method of intra-edge/intra-midpoint prediction for unsent pixels;
- [0022] FIG. 18 is a flowchart illustrating a first aspect of a method of interpolating a pixel value;
- [0023] FIG. 19 is a flowchart illustrating a second aspect of a method of interpolating a pixel value; and
- [0024] FIG. 20 is a representation of a plurality of pixels.

SUMMARY OF THE DISCLOSURE

[0025] A method of interpolating a pixel value is disclosed and may include locating a missing pixel, determining a plurality of closest pixels, determining a value for each of the plurality of closest pixels, determining a distance between the missing pixel and each of the plurality of closest pixels, classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel, and determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

[0026] In a particular aspect, the plurality of closest pixels may include a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis. The distance between the missing pixel and each of the plurality of closest pixels may include a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis. Further, the first pixel along the vertical axis may include a first value, the second pixel along the vertical axis may include a second value, the third pixel along the horizontal axis may include a third value, and the fourth pixel along the horizontal axis may include a fourth value. The first pixel along the vertical axis may include a first classification, the second pixel along the vertical axis may include a second classification, the third pixel along the horizontal axis may include a third classification, and the fourth pixel along the horizontal axis may include a fourth classification.

[0027] In this aspect, the method may further include setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel, setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel, setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel, and setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel. The method may also include determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (\text{LA} * \text{A} * \text{b} + \text{LB} * \text{B} * \text{a}) / (\text{LB} * \text{a} + \text{LA} * \text{b})$, wherein Vert_Pred is the vertical predictor value, LA is the first mask, A is the first value, b is the second distance, LB is the second mask, B is the second value, and a is the first distance. Moreover, the method may include determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (\text{LC} * \text{C} * \text{d} + \text{LD} * \text{D} * \text{c}) / (\text{LD} * \text{c} + \text{LC} * \text{d})$, wherein Horz_Pred is the horizontal predictor value, LC is the third mask, C is the third value, d is the fourth distance, LD is the fourth mask, D is the fourth value, and c is the third distance. The method may also include determining an interpolated value for the missing pixel from $P = (\text{Vert_Pred} + \text{Horz_Pred}) / 2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

[0028] Further, in this aspect, the non-edge pixel may be a midpoint pixel. The midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof. Moreover, the value of each of the plurality of

closest pixels may be selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

[0029] In another aspect, a decoder device is disclosed and may include means for locating a missing pixel, means for determining a plurality of closest pixels, means for determining a value for each of the plurality of closest pixels, means for determining a distance between the missing pixel and each of the plurality of closest pixels, means for classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel, and means for determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

[0030] In this aspect, the plurality of closest pixels may include a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis. The distance between the missing pixel and each of the plurality of closest pixels may include a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis. Further, the first pixel along the vertical axis may include a first value, the second pixel along the vertical axis may include a second value, the third pixel along the horizontal axis may include a third value, and the fourth pixel along the horizontal axis may include a fourth value. The first pixel along the vertical axis may include a first classification, the second pixel along the vertical axis may include a second classification, the third pixel along the horizontal axis may include a third classification, and the fourth pixel along the horizontal axis may include a fourth classification.

[0031] The decoder device may also include means for setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel, means for setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel, means for setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel, and means for setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel. Further, the decoder device may include means for determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (LA * A * b + LB * B * a) / (LB * a + LA * b)$, wherein Vert_Pred is the vertical predictor value, LA is the first mask, A is the first value, b is the second distance, LB is the second mask, B is the

second value, and a is the first distance. The decoder device may include means for determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (\text{LC} * \text{C} * d + \text{LD} * \text{D} * c) / (\text{LD} * c + \text{LC} * d)$, wherein Horz_Pred is the vertical predictor value, LC is the third mask, C is the third value, d is the fourth distance, LD is the fourth mask, D is the fourth value, and c is the third distance. The decoder device may also include means for determining an interpolated value for the missing pixel from $P = (\text{Vert_Pred} + \text{Horz_Pred})/2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

[0032] In this aspect, the non-edge pixel may be a midpoint pixel. The midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof. Moreover, the value of each of the plurality of closest pixels may be selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

[0033] In yet another aspect, a device is disclosed and may include a processor. The processor may be operable to locate a missing pixel, to determine a plurality of closest pixels, to determine a value for each of the plurality of closest pixels, to determine a distance between the missing pixel and each of the plurality of closest pixels, to classify each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel, and to determine a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

[0034] In this aspect, the plurality of closest pixels may include a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis. The distance between the missing pixel and each of the plurality of closest pixels may include a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis. Further, the first pixel along the vertical axis may include a first value, the second pixel along the vertical axis may include a second value, the third pixel along the horizontal axis may include a third value, and the fourth pixel along the horizontal axis may include a fourth value. The first pixel along the vertical axis may include a first classification, the second pixel along the vertical axis may include a second classification, the third pixel along the horizontal axis may include a third classification, and the fourth pixel along the horizontal axis may include a fourth classification.

[0035] The processor may be further operable to set a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel, to set a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel, to set a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel, and to set a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel. The processor may be operable to determine a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (\text{LA} * \text{A} * \text{b} + \text{LB} * \text{B} * \text{a}) / (\text{LB} * \text{a} + \text{LA} * \text{b})$, wherein Vert_Pred is the vertical predictor value, LA is the first mask, A is the first value, b is the second distance, LB is the second mask, B is the second value, and a is the first distance. The processor may be operable to determine a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (\text{LC} * \text{C} * \text{d} + \text{LD} * \text{D} * \text{c}) / (\text{LD} * \text{c} + \text{LC} * \text{d})$, wherein Horz_Pred is the horizontal predictor value, LC is the third mask, C is the third value, d is the fourth distance, LD is the fourth mask, D is the fourth value, and c is the third distance. Moreover, the processor may be operable to determine an interpolated value for the missing pixel from $\text{P} = (\text{Vert_Pred} + \text{Horz_Pred}) / 2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

[0036] In this aspect, the non-edge pixel may be a midpoint pixel. The midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof. Moreover, the value of each of the plurality of closest pixels may be selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

[0037] In another aspect, a computer program product is disclosed and may include a computer-readable medium. The computer-readable medium may include at least one instruction for locating a missing pixel, at least one instruction for determining a plurality of closest pixels, at least one instruction for determining a value for each of the plurality of closest pixels, at least one instruction for determining a distance between the missing pixel and each of the plurality of closest pixels, at least one instruction for classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel, and at least one instruction for determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

[0038] In this aspect, the plurality of closest pixels may include a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis. The distance between the missing pixel and each of the plurality of closest pixels may include a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis. Further, the first pixel along the vertical axis may include a first value, the second pixel along the vertical axis may include a second value, the third pixel along the horizontal axis may include a third value, and the fourth pixel along the horizontal axis may include a fourth value. The first pixel along the vertical axis may include a first classification, the second pixel along the vertical axis may include a second classification, the third pixel along the horizontal axis may include a third classification, and the fourth pixel along the horizontal axis may include a fourth classification.

[0039] The computer-readable medium may include at least one instruction for setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel, at least one instruction for setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel, at least one instruction for setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel, and at least one instruction for setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel. Moreover, the computer-readable medium may include at least one instruction for determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (\text{LA} * \text{A} * \text{b} + \text{LB} * \text{B} * \text{a}) / (\text{LB} * \text{a} + \text{LA} * \text{b})$, wherein Vert_Pred is the vertical predictor value, LA is the first mask, A is the first value, b is the second distance, LB is the second mask, B is the second value, and a is the first distance. Also, the computer-readable medium may include at least one instruction for determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (\text{LC} * \text{C} * \text{d} + \text{LD} * \text{D} * \text{c}) / (\text{LD} * \text{c} + \text{LC} * \text{d})$, wherein Horz_Pred is the vertical predictor value, LC is the third mask, C is the third value, d is the fourth distance, LD is the fourth mask, D is the fourth value, and c is the third distance. Further, the computer-readable medium may include at least one instruction for determining an interpolated value for the missing pixel from $P = (\text{Vert_Pred} + \text{Horz_Pred}) / 2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

[0040] In this aspect, the non-edge pixel may be a midpoint pixel. The midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof. Moreover, the value of each of the plurality of closest pixels may be selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

[0041] In still another aspect, a method of selectively refining image corners is disclosed and may include scanning a region of an image, classifying the region as either text-containing or non-text-containing, identifying any missing corners within a text-containing region, and adding the missing corners to an edge map. Further, the method may include obtaining an overlapping block of pixels within the image and plotting a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity. The method may also include determining a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels. The method may include determining a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram. Moreover, the method may include determining whether BPR is greater than a second threshold, Th2; determining whether NC is less than a third threshold, Th3, and classifying the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.

[0042] In this aspect, the method may include performing pattern matching on the text-containing region. Also, the method may include locating a non-edge pixel, considering the non-edge pixel in at least one corner position, locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position, and locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position. The method may include determining whether any of the plurality of horizontal pixels and the plurality of vertical pixels is an edge pixel and labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel. The at least one corner position may be selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

[0043] In another aspect, a device is disclosed and may include means for scanning a region of an image, means for classifying the region as either text-containing or non-

text-containing, means for identifying any missing corners within a text-containing region, and means for adding the missing corners to an edge map. Further, the device may include means for obtaining an overlapping block of pixels within the image and plotting a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity. The device may also include means for determining a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN})/\text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels. The device may include means for determining a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram. Moreover, the device may include means for determining whether BPR is greater than a second threshold, Th2; determining whether NC is less than a third threshold, Th3, and classifying the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.

[0044] In this aspect, the device may include means for performing pattern matching on the text-containing region. Also, the device may include means for locating a non-edge pixel, means for considering the non-edge pixel in at least one corner position, means for locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position, and means for locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position. The device may include means for determining whether any of the plurality of horizontal pixels and the plurality of vertical pixels is an edge pixel and means for labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel. The at least one corner position may be selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

[0045] In yet another aspect, a device is disclosed and may include a processor. The processor may be operable to scan a region of an image, to classify the region as either text-containing or non-text-containing, to identify any missing corners within a text-containing region, and to add the missing corners to an edge map. The processor may also be operable to obtain an overlapping block of pixels within the image and to plot a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity. Further, the processor may be operable to determine a background pixel ratio (BPR) using $BPR =$

$\text{Max}(\text{HistN})/\text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels. The processor may be operable to determine a number of components (NC) using $\text{NC} = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram. Also, the processor may be operable to determine whether BPR is greater than a second threshold, Th2, to determine whether NC is less than a third threshold, Th3, and to classify the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.

[0046] In this aspect, the processor may be operable to perform pattern matching on the text-containing region. The processor may be operable to locate a non-edge pixel, to consider the non-edge pixel in at least one corner position, to locate a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position, and to locate a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position. Also, the processor may be operable to determine whether any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel and to label the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel. The at least one corner position is selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

[0047] In yet another aspect, a computer program product is disclosed and may include a computer-readable medium. The computer-readable medium may include at least one instruction scanning a region of an image, at least one instruction classifying the region as either text-containing or non-text-containing, at least one instruction identifying any missing corners within a text-containing region, and at least one instruction adding the missing corners to an edge map. Further, the computer-readable medium may include at least one instruction obtaining an overlapping block of pixels within the image and plotting a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity. The computer-readable medium may also include at least one instruction determining a background pixel ratio (BPR) using $\text{BPR} = \text{Max}(\text{HistN})/\text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels. The computer-readable medium may include at least one instruction determining a number of components (NC) using $\text{NC} = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value,

Th1, based on the histogram. Moreover, the computer-readable medium may include at least one instruction determining whether BPR is greater than a second threshold, Th2; determining whether NC is less than a third threshold, Th3, and classifying the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.

[0048] In this aspect, the computer-readable medium may include at least one instruction performing pattern matching on the text-containing region. Also, the computer-readable medium may include at least one instruction locating a non-edge pixel, at least one instruction considering the non-edge pixel in at least one corner position, at least one instruction locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position, and at least one instruction locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position. The computer-readable medium may include at least one instruction determining whether any of the plurality of horizontal pixels and the plurality of vertical pixels is an edge pixel and at least one instruction labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel. The at least one corner position may be selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

[0049] In another aspect, a method of inter-edge/inter-midpoint prediction is disclosed and may include receiving previous frame data, initializing a search radius to be equal to one, scanning around a co-located point in the previous frame data using the search radius, and determining whether a reference point is found. The previous frame data and the reference point may include at least one midpoint. Also, the previous frame data and the reference point may include at least one edge.

[0050] In this aspect, the method may include setting an inter-predicted value equal to a value of the reference point, when the reference point is found. Further, the method may include increasing the search radius by one, when the reference point is not found and determining whether the search radius is equal to a maximum range. The method may also include scanning around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range, determining whether a reference point is found, and setting an inter-predicted value equal to a value of the reference point, when the reference point is found. The method may include setting an inter-predicted value equal to a predetermined default value,

when the search radius is equal to the maximum range. The predetermined default value may be a middle value in a range of values of frame data.

[0051] In another aspect, a device is disclosed and may include means for receiving previous frame data, means for initializing a search radius to be equal to one, means for scanning around a co-located point in the previous frame data using the search radius, and means for determining whether a reference point is found. The previous frame data and the reference point may include at least one midpoint. Also, the previous frame data and the reference point may include at least one edge.

[0052] In this aspect, the device may include means for setting an inter-predicted value equal to a value of the reference point, when the reference point is found. Further, the device may include means for increasing the search radius by one, when the reference point is not found and determining whether the search radius is equal to a maximum range. The device may also include means for scanning around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range, means for determining whether a reference point is found, and means for setting an inter-predicted value equal to a value of the reference point, when the reference point is found. The device may include means for setting an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range. The predetermined default value may be a middle value in a range of values of frame data.

[0053] In still another aspect, a device is disclosed and may include a processor. The processor may be operable to receive previous frame data, to initialize a search radius to be equal to one, to scan around a co-located point in the previous frame data using the search radius, and to determine whether a reference point is found. The previous frame data and the reference point may include at least one midpoint. Further, the previous frame data and the reference point may include at least one edge.

[0054] In this aspect, the processor may be operable to set an inter-predicted value equal to a value of the reference point, when the reference point is found. The processor may also be operable to increase the search radius by one, when the reference point is not found and to determine whether the search radius is equal to a maximum range. Further, the processor may be operable to scan around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range, to determine whether a reference point is found, and to set an inter-predicted value equal to a value of the reference point, when the reference point is found. The

processor may be operable to set an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range. The predetermined default value may be a middle value in a range of values of frame data.

[0055] In another aspect, a computer program product is disclosed and may include a computer-readable medium. The computer-readable medium may include at least one instruction for receiving previous frame data, at least one instruction for initializing a search radius to be equal to one, at least one instruction for scanning around a co-located point in the previous frame data using the search radius, and at least one instruction for determining whether a reference point is found. The previous frame data and the reference point may include at least one midpoint. Also, the previous frame data and the reference point may include at least one edge.

[0056] In this aspect, the computer-readable medium may include at least one instruction for setting an inter-predicted value equal to a value of the reference point, when the reference point is found. Further, the computer-readable medium may include at least one instruction for increasing the search radius by one, when the reference point is not found and determining whether the search radius is equal to a maximum range. The computer-readable medium may also include at least one instruction for scanning around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range, at least one instruction for determining whether a reference point is found, and at least one instruction for setting an inter-predicted value equal to a value of the reference point, when the reference point is found. The computer-readable medium may include at least one instruction for setting an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range. The predetermined default value may be a middle value in a range of values of frame data.

[0057] In yet another aspect, a method of intra-edge/intra-midpoint prediction is disclosed and may include determining a plurality of sent pixels, dividing an image into a plurality of pixel blocks having a predetermined size, and determining whether any pixels within each pixel block is within the plurality of sent pixels. The plurality of sent pixels may include a plurality of edge pixels and a plurality of midpoint pixels. The predetermined size may be set equal to a scale factor in a last iteration.

[0058] In this aspect, the method may include setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels. The method may also include setting a predictor value equal

to a mean of all sent pixel values within the block of pixels. Further, the method may include determining a residue of unsent pixels. The residue of unsent pixels may include an actual value minus the predictor values.

[0059] In another aspect, a device is disclosed and may include means for determining a plurality of sent pixels, means for dividing an image into a plurality of pixel blocks having a predetermined size, and means for determining whether any pixels within each pixel block is within the plurality of sent pixels. The plurality of sent pixels may include a plurality of edge pixels and a plurality of midpoint pixels. The predetermined size may be set equal to a scale factor in a last iteration.

[0060] In this aspect, the device may include means for setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels. The device may also include means for setting a predictor value equal to a mean of all sent pixel values within the block of pixels. Further, the device may include means for determining a residue of unsent pixels. The residue of unsent pixels may include an actual value minus the predictor values.

[0061] In yet another aspect, a device is disclosed and may include a processor. The processor may be operable to determine a plurality of sent pixels, to divide an image into a plurality of pixel blocks having a predetermined size, and to determine whether any pixels within each pixel block is within the plurality of sent pixels. The plurality of sent pixels may include a plurality of edge pixels and a plurality of midpoint pixels. The predetermined size may be set equal to a scale factor in a last iteration.

[0062] In this aspect, the processor may be operable to set a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels. The processor may also be operable to set a predictor value equal to a mean of all sent pixel values within the block of pixels. Further, the processor may be operable to determine a residue of unsent pixels. The residue of unsent pixels may include an actual value minus the predictor values.

[0063] In another aspect, a computer program product is disclosed and may include a computer-readable medium. The computer-readable medium may include at least one instruction for determining a plurality of sent pixels, at least one instruction for dividing an image into a plurality of pixel blocks having a predetermined size, and at least one instruction for determining whether any pixels within each pixel block is within the plurality of sent pixels. The plurality of sent pixels may include a plurality of edge

pixels and a plurality of midpoint pixels. The predetermined size may be set equal to a scale factor in a last iteration.

[0064] In this aspect, the computer-readable medium may include at least one instruction for setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels. The computer-readable medium may also include at least one instruction for setting a predictor value equal to a mean of all sent pixel values within the block of pixels. Further, the computer-readable medium may include at least one instruction for determining a residue of unsent pixels. The residue of unsent pixels may include an actual value minus the predictor values.

[0065] In yet another aspect, a method of refining midpoints is disclosed and may include receiving a plurality of midpoints, scanning a plurality of pixels in a predetermined number of directions using a predetermined step angle from each of the plurality of midpoint pixels, determining whether a current pixel location is greater than a maximum scan distance, and determining whether a current pixel is an edge pixel if the current pixel location is not greater than the maximum scan distance. The method may also include determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel and removing the current pixel from a midpoint map when the current pixel is a midpoint pixel. The predetermined number of directions may include eight directions. The predetermined step angle may be forty-five degrees (45°).

[0066] In yet another aspect, a device is disclosed and may include means for receiving a plurality of midpoints, means for scanning a plurality of pixels in a predetermined number of directions using a predetermined step angle from each of the plurality of midpoint pixels, means for determining whether a current pixel location is greater than a maximum scan distance, and means for determining whether a current pixel is an edge pixel if the current pixel location is not greater than the maximum scan distance. The device may also include means for determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel and removing the current pixel from a midpoint map when the current pixel is a midpoint pixel. The predetermined number of directions may include eight directions. The predetermined step angle may be forty-five degrees (45°).

[0067] In still another aspect, a device, the device is disclosed and may include a processor. The processor may be operable to receive a plurality of midpoints, to scan a plurality of pixels in a predetermined number of directions using a predetermined step

angle from each of the plurality of midpoint pixels, to determine whether a current pixel location is greater than a maximum scan distance, and to determine whether a current pixel is an edge pixel if the current pixel location is not greater than the maximum scan distance. Further, the processor may be operable to determine whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel and to remove the current pixel from a midpoint map when the current pixel is a midpoint pixel. The predetermined number of directions may include eight directions. The predetermined step angle may be forty-five degrees (45°).

[0068] In yet another aspect, a computer program product is disclosed and may include a computer-readable medium. The computer-readable medium may include at least one instruction for receiving a plurality of midpoints, at least one instruction for scanning a plurality of pixels in a predetermined number of directions using a predetermined step angle from each of the plurality of midpoint pixels, at least one instruction for determining whether a current pixel location is greater than a maximum scan distance, and at least one instruction for determining whether a current pixel is an edge pixel if the current pixel location is not greater than the maximum scan distance. The computer-readable medium may also include at least one instruction for determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel and removing the current pixel from a midpoint map when the current pixel is a midpoint pixel. The predetermined number of directions may include eight directions. The predetermined step angle may be forty-five degrees (45°).

DETAILED DESCRIPTION

[0069] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects.

[0070] In this description, the term “application” may also include files having executable content, such as: object code, scripts, byte code, markup language files, and patches. In addition, an “application” referred to herein, may also include files that are not executable in nature, such as documents that may need to be opened or other data files that need to be accessed.

[0071] The term “content” may also include files having executable content, such as: object code, scripts, byte code, markup language files, and patches. In addition, any “content” referred to herein, may also include files that are not executable in nature,

such as documents that may need to be opened or other data files that need to be accessed.

[0072] In this description, the terms “communication device,” “wireless device,” “wireless telephone,” “wireless communications device,” and “wireless handset” are used interchangeably. With the advent of third generation (3G) wireless technology, more bandwidth availability has enabled more electronic devices with wireless capabilities. Therefore, a wireless device could be a cellular telephone, a pager, a PDA, a smartphone, a navigation device, or a computer with a wireless connection.

[0073] Referring initially to FIG. 1, a wireless mobile digital display interface (WMDDI) system is shown and is generally designated 100. As shown, the WMDDI system 100 includes a host device 102 and a client device 104. The host device 102 and the client device 104 may be connected via a wireless link 106. In a particular aspect, the host device may be a mobile device such as a cellular phone, a mobile phone, a portable data assistant (PDA), or some other handheld wireless device. Further, the client device 104 may be a wireless display having embedded capabilities such as audio/video decoding, HID, or a combination thereof. For example, the client device 104 may be an external display, a projector, or some other similar device. The wireless link 106 may be a WMDDI.

[0074] In a particular aspect, the host device 102 and the client device 104 may be able to establish association and secure communication for the exchange of multimedia content and control messages using the WMDDI protocol. In one aspect, the communication may be asymmetric in the sense that a larger amount of data may be exchanged in the forward link, i.e., from the host device 102 to the client device 104.

[0075] In a particular aspect, the host device 102 may include several applications that may provide multimedia content. For example, the host device 102 may include a media player that may decode compressed video and audio bitstreams from a file. The host device 102 may also include one or more video games that may process graphic commands to render image content. Further, the host device 102 may include a camera, e.g., a charged coupled device (CCD) camera, a complementary metal oxide semiconductor (CMOS) camera, or some other camera, that may be used to stream captured red/green/blue (RGB) images. Each of these applications, and other applications that generate image content, may provide raw video, e.g., RGB or Cr/Cb/Y, to an internal display on the host device 102 with a resolution and a bit-depth that matches the resolution and bit-depth of the internal display. The internal display may be

a liquid crystal display (LCD) or some other display. The bit rates for raw video content, e.g., RGB, at common display resolutions and frame rates are shown below in Table 1.

[0076] Table 1. RGB Bit Rates for Common Display Resolutions and Frame Rates.

Video Sequence (24 bpp)	Bit Rate
QVGA (320x240), 30 fps	55.3 Mbps
VGA (640x480), 30 fps	221.2 Mbps
WVGA (864x480), 30 fps	298.6 Mbps
HDTV (1080x1920, 30 fps)	1.5 Gbps

When the client device 104 is connected to the host device 102 using the wireless link 106, the data transferred from the host device 102 to the client device 104 may be transferred through a wireless channel. If the capacity of the wireless channel is not large enough, the raw video data may be compressed. The capacity of the wireless channel may not be sufficient due to technical constraints of the physical layer of the wireless channel, due to the wireless channel being shared among other devices, due to some margin of the channel being retained for error mitigation techniques or correction techniques, or a combination thereof.

[0077] The wireless link 106 may not provide a fixed capacity. Accordingly, video transmission through the wireless link 106 may be adaptable to varying channel conditions.

[0078] Referring to FIG. 2, an exemplary, non-limiting aspect of a wireless telephone is shown and is generally designated 220. As shown, the wireless device 220 includes an on-chip system 222 that includes a digital signal processor 224 and an analog signal processor 226 that are coupled together. As illustrated in FIG. 2, a display controller 228 and a touchscreen controller 230 are coupled to the digital signal processor 224. In turn, a touchscreen display 232 external to the on-chip system 222 is coupled to the display controller 228 and the touchscreen controller 230.

[0079] FIG. 2 further indicates that a video encoder 234, e.g., a phase alternating line (PAL) encoder, a sequential couleur a memoire (SECAM) encoder, or a national television system(s) committee (NTSC) encoder, is coupled to the digital signal processor 224. Further, a video amplifier 236 is coupled to the video encoder 234 and the touchscreen display 232. Also, a video port 238 is coupled to the video amplifier 236. As depicted in FIG. 2, a universal serial bus (USB) controller 240 is coupled to the digital signal processor 224. Also, a USB port 242 is coupled to the USB controller

240. A memory 244 and a subscriber identity module (SIM) card 246 may also be coupled to the digital signal processor 224. Further, as shown in FIG. 2, a digital camera 248 may be coupled to the digital signal processor 224. In an exemplary aspect, the digital camera 248 is a charge-coupled device (CCD) camera or a complementary metal-oxide semiconductor (CMOS) camera.

[0080] As further illustrated in FIG. 2, a stereo audio CODEC 250 may be coupled to the analog signal processor 226. Moreover, an audio amplifier 252 may be coupled to the stereo audio CODEC 250. In an exemplary aspect, a first stereo speaker 254 and a second stereo speaker 256 are coupled to the audio amplifier 252. FIG. 2 shows that a microphone amplifier 258 may be also coupled to the stereo audio CODEC 250.

Additionally, a microphone 260 may be coupled to the microphone amplifier 258. In a particular aspect, a frequency modulation (FM) radio tuner 262 may be coupled to the stereo audio CODEC 250. Also, an FM antenna 264 is coupled to the FM radio tuner 262. Further, stereo headphones 266 may be coupled to the stereo audio CODEC 250.

[0081] FIG. 2 further indicates that a radio frequency (RF) transceiver 268 may be coupled to the analog signal processor 226. An RF switch 270 may be coupled to the RF transceiver 268 and an RF antenna 272. As shown in FIG. 2, a keypad 274 may be coupled to the analog signal processor 226. Also, a mono headset with a microphone 276 may be coupled to the analog signal processor 226. Further, a vibrator device 278 may be coupled to the analog signal processor 226. FIG. 2 also shows that a power supply 280 may be coupled to the on-chip system 222. In a particular aspect, the power supply 280 is a direct current (DC) power supply that provides power to the various components of the wireless device 220 that require power. Further, in a particular aspect, the power supply is a rechargeable DC battery or a DC power supply that is derived from an alternating current (AC) to DC transformer that is connected to an AC power source.

[0082] As depicted in FIG. 2, the touchscreen display 232, the video port 238, the USB port 242, the camera 248, the first stereo speaker 254, the second stereo speaker 256, the microphone 260, the FM antenna 264, the stereo headphones 266, the RF switch 270, the RF antenna 272, the keypad 274, the mono headset 276, the vibrator 278, and the power supply 280 are external to the on-chip system 222.

[0083] In a particular aspect, one or more of the methods described herein may be stored in the memory 244 as computer program instructions. These instructions may be executed by a processor 224, 226 in order to perform the methods described herein.

Further, the processors, 224, 226, the memory 244, the instructions stored therein, or a combination thereof may serve as a means for performing one or more of the methods described herein.

[0084] Referring now to FIG. 3, a method of compressing graphic content and video content for transmission via a wireless link is shown and is generally designated 300. Commencing at block 302, edge detection may be performed on a plurality of pixels 304 at least partially based on a Gauss Sigma value 306, a percentage of edges value 308, and a low threshold to high threshold ratio 310. At block 312, selective corner refinement may be performed on the pixels 304 and on an edge-map 314 created at block 302. Moving to block 317, run length encoding may be performed on a refined edge-map 316 created at block 308. Thereafter, at block 318 arithmetic encoding may be performed on the run length encoded data. In a particular aspect, the refined edge-map 316 may be stored in a current frame data (CFD) buffer 320 for use later as described herein.

[0085] Proceeding to block 322, the midpoints may be obtained from the refined edge-map 316 to create a midpoints-map 324. Thereafter, the midpoints within the midpoints-map 324 may be refined at block 326 at least partially based on the refined edge-map 316. At block 328, edge values 330 may be obtained from the refined edge-map 316. The edge values may be stored in the CFD buffer 320.

[0086] Moving to block 332, midpoints may be recovered based on the refined edge-map 316, the midpoints-map 324, the pixels 304, and a refined midpoints-map 334 created at block 326. At block 336, inter prediction may be performed on the pixels 304 and previous frame data (PFD) received from a PFD buffer 338. The result of the inter prediction method may include edges residue 340 and midpoints residue 342 and arithmetic encoding may be performed on the edges residue 340 and the midpoints residue 342 at block 318. The inter prediction may be performed for multiple frames.

[0087] At block 344, horizontal and vertical midpoints may be encoded based on the midpoints-map 324, the refined edge-map 316, the pixels 304, or a combination thereof. The horizontal and vertical midpoint encoding may be performed for multiple iterations. The result of this encoding step may include midpoints residue 346 and arithmetic encoding may be performed on the midpoints residue 346 at block 318. Further, a midpoints map 348 may be output to the CFD buffer 320.

[0088] Moving to block 350, intra prediction for unsent pixels may be performed at least partially based on the pixels 304, the current frame data within the CFD buffer

320, the midpoints-map 348, the edges map 314, or a combination thereof. After intra prediction is performed, the encoding may be considered lossless. The results of the intra prediction may include lossless pixel residue on which arithmetic encoding may be performed at block 318. As shown, the output of the method shown in FIG. 3, may be bitstream 354.

[0089] FIG. 4 illustrates a method of detecting image edges, generally designated 400. Beginning at block 402, an encoder may construct a Gaussian mask using Gauss sigma notation 404. At block 406, the encoder may smooth the image using a plurality of pixels 408. Moving to block 410, the encoder may compute horizontal gradients, vertical gradients, and a gradient magnitude (MG).

[0090] At block 412, the encoder may plot a histogram of the MG and obtain a high threshold ("high Th") at least partially based on a percentage of edges 414. Thereafter, at block 416, the encoder may determine a low threshold ("low Th"). The encoder may determine low Th based on a threshold ratio 418 that is equal to low Th/high Th. At block 420, the encoder may perform non-maximal suppression. At block 422, the encoder may perform hysteresis thresholding. Thereafter, at block 424, the encoder may output an edge-map.

[0091] FIG. 5 illustrates a first aspect of a method of selectively refining image corners. Beginning at block 502, a region may be scanned. At block 504, the region may be classified as either text-containing or non-text-containing. At decision 506, it may be determined whether the region is text-containing or non-text-containing. If the region is text-containing, the method 500 may continue to block 508 and the missing corners within the text-containing region may be identified. Then, at block 510, the missing corners may be added to an edge map.

[0092] Moving to decision 512, it may be determined if another region is available for scanning. If so, the method 500 may return to block 502 and continue as described herein. Otherwise, the method 500 may move to block 514 and a refined edge-map may be transmitted. The method 500 may then end. Returning to decision 506, if the region is non-text, the method 500 may proceed directly to decision 512 and continue as described herein.

[0093] Referring now to FIG. 6, a second aspect of a method of selectively refining image corners is shown and is generally designated 600. Commencing at block 602, the encoder may obtain an overlapping block of pixels 604. Each frame of an image may be divided into square blocks of pixels. The size of each square may have a dimension

that is equal to a basic size (Basic_Size) plus two times a pattern matching block size (Pattern_Matching_Block_Size). In other words, the dimension of each square may be determined using the following formula:

$$\text{Side} = \text{Basic_Size} + 2 * \text{Pattern_Matching_Block_Size}$$

Further, the bordering Pattern_Matching_Block_Size pixels may overlap with adjacent blocks of pixels.

[0094] At block 606, the encoder may obtain a histogram, HistN, of the pixels in the block and plot the histogram so that each bin in the histogram corresponds to one pixel intensity. At block 608, a background pixel ratio (BPR) may be determined using the following formula:

$$\text{BPR} = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$$

Where:

Max(HistN) = the maximum number of pixels in the histogram;

Tot_Blk_Pix = the total number of pixels in the block of pixels.

At block 608, a number of components (NC) may also be determined using the following formula:

$$\text{NC} = \text{Sum} (\text{HistN} > \text{Th1})$$

The number of components may represent a number of pixels whose occurrences exceed a first threshold value, Th1, based on the histogram.

[0095] Moving to decision step 610, the encoder may determine whether BPR is greater than a second threshold, Th2, and whether NC is less than a third threshold, Th3. If so, the block of pixels may be classified as a text block and the method 600 may proceed to block 612 and for every non-edge pixel in an edge-map 614, pattern matching may be performed. Also, at block 612, for every match found, new edges may be inserted into the edge-map to create a refined edge-map. At block 616, the encoder may output the refined edge-map. Thereafter, the method 600 may end. Returning to decision 610, if the conditions are not met, the block of pixels may be classified as a

non-text block and the method 600 may end. In a particular aspect, the second threshold, Th2, and the third threshold, Th3, may be preset values based on the histogram.

[0096] FIG. 7 is a flow chart illustrating a method of pattern matching that is generally designated 700. FIG. 8 is a diagram illustrating text block corners that may be referred to during the description of the method 700 shown in FIG. 7.

[0097] Commencing at block 702, for each non-edge pixel, P, in a text block, the following steps may be performed. At block 704, P, may be considered as if it is in a corner position, e.g., a top left corner, a top right corner, a bottom left corner, or a bottom right corner. At block 706, the nearest three (3) horizontal pixels may be located. At block 708, the nearest three (3) vertical pixels may be located. In a particular aspect, three (3) is the selected Pattern_Matching_Block_Size.

[0098] Moving to decision 710, the encoder may determine whether any of the nearest three (3) horizontal pixels and any of the nearest three (3) vertical pixels are edges. If so, the method 700 may move to block 712, and the pixel, P, may be labeled as an edge. Thereafter, at decision 714, the encoder may determine if another pixel is available for pattern matching. If so, the method 700 may return to block 704 and continue as described herein. Otherwise, the method 700 may end.

[0099] Returning to decision step 710, if none of the nearest six pixels is an edge, the method 700 may move to decision 716 and the encoder may determine whether the pixel, P, is tested in all four corners. If not, the method 700 may proceed to block 718 and the pixel, P, may be moved to a next corner position. If the pixel, P, is tested in all corner locations, the method 700 may continue to decision 714 and continue as described herein.

[00100] In a particular aspect, the methods 600, 700 shown in FIG. 6 and FIG. 7, may be used to refine an edge-map by including missing corners in regions that contain text. These methods 600, 700 may locally classify regions as either text-containing or non-text-containing. This may be determined based on the local histogram of luminance. The decisions in method 600 may be based on predefined threshold values in the luminance histogram, i.e., Th1, Th2, and Th3. Further, the methods 600, 700 may select an area in text-containing regions that may contain a missing corner. These areas may be selected using pattern matching over an initial edge-map. For the selected areas, the missing corners may be added to the edge-map.

[00101] Many well known edge detects may fail to detect the corners in text portions of an image wherein the image gradient is not strongly oriented in the horizontal or the vertical direction. Many edge-pixels may be undetected and cause the midpoints from one region being used for interpolation in another region and leads to a streaking artifact. The corner refinement method described in conjunction with FIG. 6 and FIG. 7 uses pattern matching to predict locations of undetected edges, e.g., corners, and introduce new edges in an edge-map. Since this may lead to a large number of new edges, and as such, introduce a higher bit-rate for non-text regions, preprocessing steps (steps 602-510) may be included in order to classify a frame into blocks of text or non-text regions. The corner refinement is then selectively applied for text regions of the image.

[00102] The methods 600, 700 shown in FIG. 6 and FIG. 7 may improve a pixel signal-to-noise ratio (PSNR) by approximately five to eight decibels (5-8 dB) for a similar bit-rate transmission. Further, these methods 600, 700 are relatively computationally inexpensive as they utilize simple pattern matching that may be optimized using bit-wise operators. A need for elaborate gradient computation methods, e.g., the Harris Corner Detector, may be eliminated.

[00103] FIG. 9 illustrates a method of run length encoding that is generally designated 900. In a particular aspect, the method 900 may be performed on a refined edge-map 902 that may be input to an encoder. Further, the method 900 may utilize one or more pixel positions 904 in performing the run length encoding.

[00104] Commencing at block 906, the encoder may raster-scan the refined edge-map 902 until a next edge is found. The pixel count to the edge may be recorded as "Run". At decision step 908, the encoder may determine whether the Run value is greater than a Max_Run value. If not, the Run value may be put in an input buffer for arithmetic encoding. The method 900 may then end. Returning to decision 908, if the Run value is greater than the Max_Run value, the Max_Run value may be put in the input buffer "N" times, wherein $N = \text{ceil}(\text{Run}/\text{Max_Run})$ and $\text{ceil}()$ takes the largest integer such that $N \leq \text{Run}/\text{Max_Run}$, followed by the value obtained as $\text{Run} = \text{Run} \% \text{Max_Run}$. The method 900 may then end.

[00105] Referring to FIG. 10, a method of obtaining image midpoints within a frame is shown and is generally designated 1000. The method 1000 of FIG. 10 may be performed by an encoder on a refined edge-map 1002 having a plurality of rows 1004

and a plurality of columns 1006. Further, the method 1000 may be performed on each row 1004 and each column 1006.

[00106] Beginning at block 1008, the encoder may scan from a current position, Curr_Pos, along a current row, Curr_Row until an edge, midpoint, or a border is found at a new position, New_Pos. At block 1010, a midpoint is defined as the current position plus the new position divided by two, i.e., $\text{Midpoint} = (\text{Curr_Pos} + \text{New_Pos})/2$. Further, at block 1010, the current position is set equal to the new position, i.e., $\text{Curr_Pos} = \text{New_Pos}$. At block 1012, a plurality of horizontal midpoints may be output.

[00107] At block 1014, the encoder may scan from a current position, Curr_Pos, along a current column, Curr_Col until an edge, midpoint, or a border is found at a new position, New_Pos. At block 1016, a midpoint is defined as the current position plus the new position divided by two, i.e., $\text{Midpoint} = (\text{Curr_Pos} + \text{New_Pos})/2$. Further, at block 1016, the current position is set equal to the new position, i.e., $\text{Curr_Pos} = \text{New_Pos}$. At block 1018, a plurality of vertical midpoints may be output. The method 1000 may then end.

[00108] FIG. 11 illustrates a method of refining midpoints, generally designated 1100. The method 1100 may be performed using plurality of midpoints 1102 and a plurality of midpoint locations 1104. Commencing at block 1106, the encoder may scan from every midpoint in eight (8) directions in steps of forty-five degrees (45°), and for a maximum of Max_Scan_Distance positions for every scan direction, as shown in FIG. 12.

[00109] Moving to decision 1108, the encoder may determine if the current pixel location is greater than a Max_Scan_Distance. If so, the method 1100 may proceed to decision 1110 and the encoder may determine whether all eight (8) directions are scanned. If not, the method 1100 may proceed to block 1112 and the encoder may proceed to other next scan direction. The method 1100 may then return to block 1106 and continue as described herein.

[00110] Returning to decision 1110, if all eight (8) directions are scanned, the method 1100 may proceed to decision 1114 and the encoder may determine whether the last midpoint has been scanned. If not, the method 1100 may proceed to block 1116 and the encoder may move to the next midpoint. Thereafter, the method 1100 may return to block 1106 and continue as described herein. Otherwise, at decision 1114, if encoder has scanned around the last midpoint, the method 1100 may proceed to block 1118 and

the encoder may output a plurality of refined midpoints. Then, the method 1100 may end.

[00111] Returning to decision 1108, if the current pixel location is not greater than a Max_Scan_Distance, the method may proceed to decision 1120 and the encoder may determine whether the current pixel is an edge. If so, the method 1100 may return to decision 1110 and continue as described herein. If the current midpoint is not an edge, the method 1100 may proceed to 1122 and the encoder may determine whether the current pixel is a midpoint. If not, the method 1100 may proceed to block 1124 and the encoder may move to the next pixel in the scan direction. At decision 1122, if the current pixel is a midpoint, the method 1100 may proceed to block 1126 and the current midpoint may be removed from a midpoint map. After all midpoints are processed as described in FIG. 11, a plurality of refined midpoints 1112 may be output and the method 1100 may end.

[00112] The method 1100 shown in FIG. 11 may be further explained with reference to FIG. 12, starting from a midpoint, scan-center, the neighboring pixels around the midpoint, in a block of five-by-five, may be scanned in steps of forty-five degrees (45°) along each of the eight lines (8) shown in FIG. 12. The neighboring midpoints may be eliminated until an edge pixel is encountered or the end of the block is reached. In the example shown in FIG. 12, pixel M0 may act as a scan center and midpoints M1, M2, and M4 may be eliminated as a result of the refinement. Pixels M5 and M6 may not be eliminated because they are across an edge and their values may differ quite significantly from the value of pixel M0.

[00113] In a particular aspect, the method 1100 shown in FIG. 11 may be used to reduce the amount of midpoints encoded in the initial iterations without significantly reducing the quality of reconstruction. The method 1100 may be considered part of a scalable encoding method in that further iterations may add further detail to an image frame which may result in better quality in the image frame. Further, the method 1100 may provide substantially better image quality than other methods for the same bitrate.

[00114] During the encoding of frame data, pixels at the centers between two edges in the horizontal direction or in the vertical direction are encoded as midpoints. Due to the nature of edges in an image, many uniform regions in an image may include a high density of midpoints. The additional midpoints, i.e., samples, may be unnecessary and may not lead to a significant improvement in image quality. The method 1100 shown in FIG. 11 may be used to eliminate unnecessary midpoints and prevent the encoding of

such midpoints. Since the linear prediction may assume the presence of a midpoint between every two edges in the vertical and horizontal directions encoded in the previous iterations, the eliminated midpoints may need to be recovered from the neighboring midpoints. These midpoints may be considered landmark points and these landmark points may need to be recovered.

[00115] FIG. 13 illustrates a method of recovering midpoints that is generally designated 1300. The method 1300 of FIG. 13 may be performed using a refined edge-map 1302, a plurality of pixels 1304, a midpoints-map 1306, a refined midpoints-map 1308, or a combination thereof.

[00116] Beginning at block 1310, the set of landmark points may be determined by those midpoints that are part of the set of Midpoints but not part of the set of Refined Midpoints, i.e., $\{\text{Landmark Pts}\} = \{\text{Midpoints}\} - \{\text{Refined Midpoints}\}$. At block 1312, the encoder may scan from every midpoint in eight (8) directions in steps of forty-five degrees (45°). At decision 1314, the encoder may determine whether a current pixel is a landmark point. If so, the method 1300 may proceed to 1316 and the encoder may fill a current pixel with a midpoint value. The method 1300 may then end and a recovered midpoint map may be output.

[00117] Returning to decision 1314, if the current pixel is not a landmark point, the method 1300 may continue to decision 1318. At decision 1318, the encoder may determine whether the current pixel is an edge. If not, the method 1300 may return to block 1312 and continue as described herein. Otherwise, the method 1300 may end and a recovered midpoint map may be output.

[00118] Using the method 1300 described in FIG. 13, the locations of the landmark points, i.e., the eliminated midpoints, may be computed first. Then, the values at those locations may be filled with the value of the midpoint that is at the scan center, e.g., pixel M0 in FIG. 12. Since the decoder does not have these pixels, it may be useful to ensure that there is a midpoint between every two edges in both horizontal and vertical directions. Since the landmark pixels, or landmark points, may act as predictors, they should be identical at the encoder and the decoder. Hence, the recovery method 1300 shown in FIG. 13 may be performed at the encoder and the decoder.

[00119] The methods 1100, 1300 shown in FIG. 11 and FIG. 13 may be used to reduce the data rate by eliminating unnecessary midpoints without suffering a significant loss in image quality. These methods 1100, 1300 may rely on the edge information to preserve midpoints across object boundaries. Further, the methods 1100, 1300 are

relatively simple and the elimination, at the encoder (method 1100), and the recovery at the decoder (1200) may follow the same scanning procedure.

[00120] Referring to FIG. 14, a flowchart illustrating a method of inter-edge/inter-midpoint prediction is shown and is generally designated 1400. The method 1400 may be performed on multiple frames one frame number at a time. Further, the method 1400 may utilize previous frame data (PFD) 1402 and a plurality of pixels 1404, e.g., edge pixels, midpoint pixels, or a combination thereof. For clarity, FIG. 15 may be referred to during the description of the method 1400 shown in FIG. 14.

[00121] Commencing at block 1406, a search radius, r , may be initialized as having a value of one (1). At block 1408, the encoder may scan around a co-located point in the previous frame around a radius, r . More specifically, the scan may be performed in a square region around the central pixel, i.e., pixel 1 in FIG. 15. During the first iteration, the scan is performed from pixel 2 through pixel 8.

[00122] At decision 1410, the encoder may determine whether a reference point, RefPt, is found during the first scan iteration. If so, the method 1400 may continue to block 1412 and the inter-predicted value is set equal to the reference point value. Then, the method 1400 may end. At decision 1410, if a reference point is not found, the method 1400 may continue to block 1414 and the value of the search radius, r , may be increased by one (1) unit, e.g., one (1) pixel.

[00123] Thereafter, at decision 1416, the encoder may determine whether the search radius, r , is greater than a maximum value, MaxRange. If r is not greater than MaxRange, the method 1400 may return to block 1408 and continue as described herein. Otherwise, if r is greater than MaxRange, the method 1400 may move to block 1418 and the encoder may set the inter-predicted value equal to a predetermined default value. In a particular aspect, the predetermined default value is equal to a middle value in a range of values. For example, the predetermined default value may be one hundred twenty-eight (128) when the range of values of the data frame is between zero (0) and two hundred fifty-five (255). Then, the method 1400 may end.

[00124] The method 1400 described in conjunction with FIG. 14 may be used for both edges and midpoints. In the case the method is used to predict edge values, the RefPt may correspond to locations given by an edge-map. In the case the method 1400 is used to predict midpoint values, the RefPt may correspond to locations given by a midpoints-map.

[00125] FIG. 16 shows a method of encoding horizontal and vertical midpoints that is designated 1600. The method 1600 may be performed on current frame data (CFD) 1602 and pixels 1604. The CFD 1602 may include a midpoints-map and an edges-map. Beginning at block 1606, the locations of horizontal midpoints may be obtained. Further, at block 1608, the locations of the vertical midpoints may be obtained. At block 1610, neighbor midpoint values may be obtained. Further, at block 1612, linear intra-edge/intra-midpoint prediction may be performed. Thereafter, midpoints residue may be output and the method 1600 may end.

[00126] Referring now to FIG. 17, a method of intra-edge/intra-midpoint prediction is shown and is generally designated 1700. The method 1700 shown in FIG. 17 may be performed on an edge-map 1702 and a midpoints-map 1704. Commencing at block 1706, the set of sent pixels, {SentPix}, may be determined by union of the edge pixels locations, {EdgeLocs}, and the midpoint pixels locations, {MidPtLocs}. In other words, $\{\text{SentPix}\} = \{\text{EdgeLocs}\} + \{\text{MidPtLocs}\}$.

[00127] At block 1708, a frame image may be divided into blocks having dimensions based on the scale factor in the last iteration (SFLI). For example, if the SFLI is equal to two (2), the frame image may be divided into blocks two-by-two (2 x 2) pixels. Moving to decision 1710, the encoder may determine whether at least one of the pixels in the block of image pixels is one of the SentPix. If not, the method 1700 may proceed to block 1712 and a predictor value may be set equal to a predefined default value. In a particular aspect, the predetermined default value is equal to one hundred twenty-eight (128). Otherwise, if one of the pixels in the block of image pixels is one of the SentPix, the method 1700 may move to block 1714 and the predictor may be set equal to the mean value of the pixels in the block.

[00128] At block 1716, the residue of unset pixels is determined to be the actual value minus the predictor value determined at block 1712 or block 1714. Thereafter, lossless pixel residue 1718 may be output and the method 1700 may end.

[00129] In a particular aspect, an edge-based video codec may encode the edges and midpoints in a frame. The edge-based video codec may achieve scalable coding by coding more and more midpoints (samples) progressively, as the number of iterations increases. After the last iteration, a subset of the original frame's pixels may remain un-encoded. These un-encoded pixels may be encoded in a "lossless encoding pass" in order to achieve lossless quality. The method 1700 described in FIG. 17 provides a predictive coding technique based on the local average of the encoded edges and

midpoints. The method 1700 may use a bit-map that indicates the location of the previously encoded pixels, e.g., edge pixels, midpoint pixels, or a combination thereof. The scale factor may be used to determine the minimum distance between every pair of edges in horizontal or vertical direction in order to determine a midpoint there between.

[00130] FIG. 18 illustrates a first aspect of a method of interpolating a pixel value, generally designated 1800. Commencing at block 1802, a missing pixel may be located by a decoder. At block 1804, the decoder may determine a plurality of closest pixels. Further, at block 1806, the decoder may determine a value, e.g., a luminance value, for each of the plurality of closest pixels. At block 1808, the decoder may determine a distance to each of the plurality of closest pixels. At block 1810, the decoder may classify each of the plurality of closest pixels as an edge-pixel or a non-edge pixel. Thereafter, at block 1812, the decoder may determine a value of the missing pixel based on the value of each of the plurality of closest pixels, the distance to each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels. The method 1800 may then end.

[00131] Referring to FIG. 19, a second aspect of a method of interpolating a pixel value is shown and is generally designated 1900. FIG. 20 is an illustration of pixels that may be referred to during the discussion of the method outlined in FIG. 19.

[00132] Beginning at block 1902, a missing pixel, P, may be located by a decoder. At block 1904, the decoder may locate the two closest vertical pixels: A and B. These two closest vertical pixels may include a first pixel along a vertical axis and a second pixel along a vertical axis. Further, at block 1906, the decoder may locate the two closest horizontal pixels: C and D. These two closest horizontal pixels may include a first pixel along a horizontal axis and a second pixel along a horizontal axis. At block 1908, the decoder may determine a value, e.g., a color density value, for each of the closest pixels. The values may be denoted A, B, C, and D. At block 1910, a distance from P to each closest pixel may be determined by the decoder. The distances may be denoted a, b, c, and d. As shown in FIG. 20, a is the distance to pixel A, b is the distance to pixel B, c is the distance to pixel C, and d is the distance to pixel D.

[00133] Moving to decision 1912, the decoder may determine whether a particular pixel, A, B, C, or D, is an edge or a midpoint. If a particular pixel is an edge, the method 1900 may proceed to block 1914 and the decoder may set a mask, L, for the particular pixel equal to zero (0). On the other hand, if a particular pixel is a midpoint, the method 1900 may continue to block 1916 and the decoder may set a mask, L, for the particular

pixel equal to one (1). From block 1914 or block 1916, the method 1900 may continue to block 1918.

[00134] At block 1918, the decoder may determine a vertical predictor value, Vert_Pred. The vertical predictor value may be determined using the following formula:

$$\text{Vert_Pred} = (L_A * A * b + L_B * B * a) / (L_B * a + L_A * b)$$

Where:

L_A = the mask of pixel A,
 A = the value of pixel A,
 b = the distance to pixel B,
 L_B = the mask of pixel B,
 B = the value of pixel B, and
 a = the distance to pixel A.

[00135] Moving to block 1920, the decoder may determine a horizontal predictor value, Horz_Pred. The horizontal predictor value may be determined using the following formula:

$$\text{Horz_Pred} = (L_C * C * d + L_D * D * c) / (L_D * c + L_C * d)$$

Where:

L_C = the mask of pixel C,
 C = the value of pixel C,
 d = the distance to pixel D,
 L_D = the mask of pixel D,
 D = the value of pixel D, and
 c = the distance to pixel D.

[00136] At block 1922, the decoder may determine an interpolated value for P from the vertical predictor value and the horizontal predictor value. The interpolated value for P may be determined from the following formula:

$$\text{Interpolated Value (P)} = (\text{Vert_Pred} + \text{Horz_Pred})/2$$

Where:

Vert_Pred = the vertical predictor determined at block 1918; and

Horz_Pred = the horizontal predictor determined at block 1920.

[00137] In a particular aspect, the method 1900 described in conjunction with FIG. 19 may predict, or interpolate, a value of a missing pixel using the value of the four closest pixels (two horizontal and two vertical), the location of the four closest pixels (distance), and the classification of the closest available pixels (edge or midpoint). If a closest pixel is an edge, that pixel may not be used for interpolation. If a closest pixel is a midpoint, that pixel may be used for interpolating the value of the missing pixel applying a distance-weighted factor.

[00138] The method 1900 described in conjunction with FIG. 19 is relatively simple and accordingly, the method 1900 may be computationally inexpensive since it only uses linear operators. However, the method 1900 may be extended to a higher order equation for better performance at the expense of additional computational costs. Further, since only midpoints, and not edges, are used for interpolation, the method 1900 may ensure that samples across an edge which may be significantly different from the true value of the interpolated pixel value are not used as predictors. Moreover, the weighted prediction may help in varying the contribution of the various reference points. The weighted prediction may be very effective in regions having a smooth transition from top to bottom or from left to right.

[00139] It is to be understood that the methods described herein do not necessarily have to be performed in the order as described. Further, words such as “thereafter”, “then”, “next”, etc. are not intended to limit the order of the steps. These words are simply used to guide the reader through the description of the methods.

[00140] With the configuration described herein, the system and method disclosed herein provides an interpolation algorithm that may be used to recover one or more missing pixels from available pixels using interpolation. Further, the system and method herein provides selective corner refinement. Regions that contain text may be identified and then, corners within the text may be detected and added to an edge map. Additionally, the system and methods disclosed herein provided an inter-frame prediction method and an intra-frame prediction method. The system and methods

herein also provide a method for reducing the density of midpoints and recovering landmark points. Pixels at the centers between two edges in the horizontal or vertical direction may be encoded as midpoints. However, due to the nature of the edges in an image, many uniform regions in an image may include a high density of midpoints. Any additional midpoints may be reduced from a bitstream transmitted since they do not lead to a significant improvement in image quality.

[00141] In one or more exemplary aspects, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage media may be any available media that may be accessed by a computer. By way of example, and not limitation, such computer-readable media may comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that may be used to carry or store desired program code in the form of instructions or data structures and that may be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[00142] Although selected aspects have been illustrated and described in detail, it will be understood that various substitutions and alterations may be made therein without departing from the spirit and scope of the present invention, as defined by the following claims.

CLAIMS

What is claimed is:

1. A method of interpolating a pixel value, the method comprising:
locating a missing pixel;
determining a plurality of closest pixels;
determining a value for each of the plurality of closest pixels;
determining a distance between the missing pixel and each of the plurality of closest pixels;
classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel; and
determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.
2. The method of claim 1, wherein the plurality of closest pixels includes a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis.
3. The method of claim 2, wherein the distance between the missing pixel and each of the plurality of closest pixels includes a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis.
4. The method of claim 3, wherein the first pixel along the vertical axis includes a first value, the second pixel along the vertical axis includes a second value, the third pixel along the horizontal axis includes a third value, and the fourth pixel along the horizontal axis includes a fourth value.
5. The method of claim 4, wherein the first pixel along the vertical axis includes a first classification, the second pixel along the vertical axis includes a second classification, the third pixel along the horizontal axis includes a third classification, and the fourth pixel along the horizontal axis includes a fourth classification.

6. The method of claim 5, further comprising:
 - setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel;
 - setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel;
 - setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel; and
 - setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel.
7. The method of claim 6, further comprising:
 - determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (L_A * A * b + L_B * B * a) / (L_B * a + L_A * b)$, wherein Vert_Pred is the vertical predictor value, L_A is the first mask, A is the first value, b is the second distance, L_B is the second mask, B is the second value, and a is the first distance.
8. The method of claim 7, further comprising:
 - determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (L_C * C * d + L_D * D * c) / (L_D * c + L_C * d)$, wherein Horz_Pred is the vertical predictor value, L_C is the third mask, C is the third value, d is the fourth distance, L_D is the fourth mask, D is the fourth value, and c is the third distance.
9. The method of claim 8, further comprising:
 - determining an interpolated value for the missing pixel from $P = (\text{Vert_Pred} + \text{Horz_Pred}) / 2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.
10. The method of claim 1, wherein the non-edge pixel is a midpoint pixel.

11. The method of claim 10, wherein the midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof.
12. The method of claim 1, where the value of each of the plurality of closest pixels is selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.
13. A decoder device, the decoder device comprising:
 - means for locating a missing pixel;
 - means for determining a plurality of closest pixels;
 - means for determining a value for each of the plurality of closest pixels;
 - means for determining a distance between the missing pixel and each of the plurality of closest pixels;
 - means for classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel; and
 - means for determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.
14. The decoder device of claim 13, wherein the plurality of closest pixels includes a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis.
15. The decoder device of claim 14, wherein the distance between the missing pixel and each of the plurality of closest pixels includes a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis.
16. The decoder device of claim 15, wherein the first pixel along the vertical axis includes a first value, the second pixel along the vertical axis includes a second value, the third pixel along the horizontal axis includes a third value, and the fourth pixel along the horizontal axis includes a fourth value.

17. The decoder device of claim 16, wherein the first pixel along the vertical axis includes a first classification, the second pixel along the vertical axis includes a second classification, the third pixel along the horizontal axis includes a third classification, and the fourth pixel along the horizontal axis includes a fourth classification.

18. The decoder device of claim 17, further comprising:
 means for setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel;
 means for setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel;
 means for setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel; and
 means for setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel.

19. The decoder device of claim 18, further comprising:
 means for determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (L_A * A * b + L_B * B * a) / (L_B * a + L_A * b)$, wherein Vert_Pred is the vertical predictor value, L_A is the first mask, A is the first value, b is the second distance, L_B is the second mask, B is the second value, and a is the first distance.

20. The decoder device of claim 19, further comprising:
 means for determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (L_C * C * d + L_D * D * c) / (L_D * c + L_C * d)$, wherein Horz_Pred is the vertical predictor value, L_C is the third mask, C is the third value, d is the fourth distance, L_D is the fourth mask, D is the fourth value, and c is the third distance.

21. The decoder device of claim 20, further comprising:
 means for determining an interpolated value for the missing pixel from $P = (\text{Vert_Pred} + \text{Horz_Pred})/2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

22. The decoder device of claim 13, wherein the non-edge pixel is a midpoint pixel.

23. The decoder device of claim 22, wherein the midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof.

24. The decoder device of claim 13, where the value of each of the plurality of closest pixels is selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

25. A device, comprising:

a processor, wherein the processor is operable to:

locate a missing pixel;

determine a plurality of closest pixels;

determine a value for each of the plurality of closest pixels;

determine a distance between the missing pixel and each of the plurality of closest pixels;

classify each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel; and

determine a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

26. The device of claim 25, wherein the plurality of closest pixels includes a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis.

27. The device of claim 26, wherein the distance between the missing pixel and each of the plurality of closest pixels includes a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis.

28. The device of claim 27, wherein the first pixel along the vertical axis includes a first value, the second pixel along the vertical axis includes a second value, the third pixel along the horizontal axis includes a third value, and the fourth pixel along the horizontal axis includes a fourth value.

29. The device of claim 28, wherein the first pixel along the vertical axis includes a first classification, the second pixel along the vertical axis includes a second classification, the third pixel along the horizontal axis includes a third classification, and the fourth pixel along the horizontal axis includes a fourth classification.

30. The device of claim 29, wherein processor is further operable to:

- set a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel;
- set a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel;
- set a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel; and
- set a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel.

31. The device of claim 30, wherein processor is further operable to:

- determine a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (L_A * A * b + L_B * B * a) / (L_B * a + L_A * b)$, wherein Vert_Pred is the vertical predictor value, L_A is the first mask, A is the first value, b is the second distance, L_B is the second mask, B is the second value, and a is the first distance.

32. The device of claim 31, wherein processor is further operable to:

- determine a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (L_C * C * d + L_D * D * c) / (L_D * c + L_C * d)$, wherein Horz_Pred is the vertical predictor value, L_C is the third mask, C is the

third value, d is the fourth distance, L_D is the fourth mask, D is the fourth value, and c is the third distance.

33. The device of claim 32, wherein processor is further operable to:

determine an interpolated value for the missing pixel from

$P = (\text{Vert_Pred} + \text{Horz_Pred})/2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

34. The device of claim 25, wherein the non-edge pixel is a midpoint pixel.

35. The device of claim 34, wherein the midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof.

36. The device of claim 25, where the value of each of the plurality of closest pixels is selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

37. A computer program product, comprising:

a computer-readable medium, comprising:

at least one instruction for locating a missing pixel;

at least one instruction for determining a plurality of closest pixels;

at least one instruction for determining a value for each of the plurality of closest pixels;

at least one instruction for determining a distance between the missing pixel and each of the plurality of closest pixels;

at least one instruction for classifying each of the plurality of closest pixels as either an edge-pixel or a non-edge pixel; and

at least one instruction for determining a value of the missing pixel at least partially based on the value of each of the plurality of closest pixels, the distance between the missing pixel and each of the plurality of closest pixels, and a classification of each of the plurality of closest pixels.

38. The computer program product of claim 37, wherein the plurality of closest pixels includes a first pixel along a vertical axis, a second pixel along a vertical axis, a third pixel along a horizontal axis, and a fourth pixel along a horizontal axis.

39. The computer program product of claim 38, wherein the distance between the missing pixel and each of the plurality of closest pixels includes a first distance along the vertical axis, a second distance along the vertical axis, a third distance along the horizontal axis, and a fourth distance along the horizontal axis.

40. The computer program product of claim 39, wherein the first pixel along the vertical axis includes a first value, the second pixel along the vertical axis includes a second value, the third pixel along the horizontal axis includes a third value, and the fourth pixel along the horizontal axis includes a fourth value.

41. The computer program product of claim 40, wherein the first pixel along the vertical axis includes a first classification, the second pixel along the vertical axis includes a second classification, the third pixel along the horizontal axis includes a third classification, and the fourth pixel along the horizontal axis includes a fourth classification.

42. The computer program product of claim 41, wherein the computer-readable medium further comprises:

at least one instruction for setting a first mask equal to zero or one if the first classification is an edge pixel or a non-edge pixel;

at least one instruction for setting a second mask equal to zero or one if the second classification is an edge pixel or a non-edge pixel;

at least one instruction for setting a third mask equal to zero or one if the third classification is an edge pixel or a non-edge pixel; and

at least one instruction for setting a fourth mask equal to zero or one if the fourth classification is an edge pixel or a non-edge pixel.

43. The computer program product of claim 42, wherein the computer-readable medium further comprises:

at least one instruction for determining a vertical predictor value, wherein the vertical predictor value is determined from $\text{Vert_Pred} = (L_A * A * b + L_B * B * a) / (L_B * a + L_A * b)$, wherein Vert_Pred is the vertical predictor value, L_A is the first mask, A is the first value, b is the second distance, L_B is the second mask, B is the second value, and a is the first distance.

44. The computer program product of claim 43, wherein the computer-readable medium further comprises:

at least one instruction for determining a horizontal predictor value, wherein the horizontal predictor value is determined from $\text{Horz_Pred} = (L_C * C * d + L_D * D * c) / (L_D * c + L_C * d)$, wherein Horz_Pred is the vertical predictor value, L_C is the third mask, C is the third value, d is the fourth distance, L_D is the fourth mask, D is the fourth value, and c is the third distance.

45. The computer program product of claim 44, wherein the computer-readable medium further comprises:

at least one instruction for determining an interpolated value for the missing pixel from

$P = (\text{Vert_Pred} + \text{Horz_Pred})/2$, wherein Vert_Pred is the vertical predictor value and Horz_Pred is the horizontal predictor value.

46. The computer program product of claim 37, wherein the non-edge pixel is a midpoint pixel.

47. The computer program product of claim 46, wherein the midpoint pixel may be obtained from a base layer bitstream, an enhancement layer bitstream, or a combination thereof.

48. The computer program product of claim 37, where the value of each of the plurality of closest pixels is selected from the group consisting of: a luminance value, a color density value, a contrast value, or a gradient value.

49. A method of selectively refining image corners, the method comprising: scanning a region of an image;

classifying the region as either text-containing or non-text-containing;
identifying any missing corners within a text-containing region; and
adding the missing corners to an edge map.

50. The method of claim 49, further comprising:
obtaining an overlapping block of pixels within the image; and
plotting a histogram (HistN) of the pixels in the block, wherein the
histogram comprises a plurality of bins and each bin corresponds to one pixel
intensity.
51. The method of claim 50, further comprising:
determining a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$, wherein Max(HistN) is a maximum number of pixels
in the histogram and Tot_Blk_Pix is a total number of pixels in the block of
pixels.
52. The method of claim 51, further comprising:
determining a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1,
based on the histogram.
53. The method of claim 52, further comprising:
determining whether BPR is greater than a second threshold, Th2;
determining whether NC is less than a third threshold, Th3; and
classifying the block of pixels as a text-containing region if BPR is
greater than Th2 and NC is less than Th3.
54. The method of claim 53, further comprising:
performing pattern matching on the text-containing region.
55. The method of claim 54, further comprising:
locating a non-edge pixel;
considering the non-edge pixel in at least one corner position;

locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position; and

locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position.

56. The method of claim 55, further comprising:

determining whether any of the plurality of horizontal pixels and the plurality of vertical pixels is an edge pixel; and

labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel.

57. The method of claim 56, wherein the at least one corner position is selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

58. A device, comprising:

means for scanning a region of an image;

means for classifying the region as either text-containing or non-text-containing;

means for identifying any missing corners within a text-containing region; and

means for adding the missing corners to an edge map.

59. The device of claim 58, further comprising:

means for obtaining an overlapping block of pixels within the image; and

means for plotting a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity.

60. The device of claim 59, further comprising:

means for determining a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels.

61. The device of claim 60, further comprising:
means for determining a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram.
62. The device of claim 61, further comprising:
means for determining whether BPR is greater than a second threshold, Th2;
means for determining whether NC is less than a third threshold, Th3;
and
means for classifying the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.
63. The device of claim 62, further comprising:
means for performing pattern matching on the text-containing region.
64. The device of claim 63, further comprising:
means for locating a non-edge pixel;
means for considering the non-edge pixel in at least one corner position;
means for locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position; and
means for locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position.
65. The device of claim 64, further comprising:
means for determining whether any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel; and
means for labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel.

66. The device of claim 65, wherein the at least one corner position is selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.
67. A device, comprising:
a processor, wherein the processor is operable to:
scan a region of an image;
classify the region as either text-containing or non-text-containing;
identify any missing corners within a text-containing region; and
add the missing corners to an edge map.
68. The device of claim 67, wherein the processor is further operable to:
obtain an overlapping block of pixels within the image; and
plot a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity.
69. The device of claim 68, wherein the processor is further operable to:
determine a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$, wherein Max(HistN) is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels.
70. The device of claim 69, wherein the processor is further operable to:
determine a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram.
71. The device of claim 70, wherein the processor is further operable to:
determine whether BPR is greater than a second threshold, Th2;
determine whether NC is less than a third threshold, Th3; and
classify the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.
72. The device of claim 71, wherein the processor is further operable to:

perform pattern matching on the text-containing region.

73. The device of claim 72, wherein the processor is further operable to:
- locate a non-edge pixel;
 - consider the non-edge pixel in at least one corner position;
 - locate a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position; and
 - locate a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position.
74. The device of claim 73, wherein the processor is further operable to:
- determine whether any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel; and
 - label the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel.
75. The device of claim 74, wherein the at least one corner position is selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.
76. A computer program product, comprising:
- a computer-readable medium, comprising:
 - at least one instruction for scanning a region of an image;
 - at least one instruction for classifying the region as either text-containing or non-text-containing;
 - at least one instruction for identifying any missing corners within a text-containing region; and
 - at least one instruction for adding the missing corners to an edge map.
77. The computer program product of claim 76, wherein the computer-readable medium further comprises:
- at least one instruction for obtaining an overlapping block of pixels within the image; and

at least one instruction for plotting a histogram (HistN) of the pixels in the block, wherein the histogram comprises a plurality of bins and each bin corresponds to one pixel intensity.

78. The computer program product of claim 77, wherein the computer-readable medium further comprises:

at least one instruction for determining a background pixel ratio (BPR) using $BPR = \text{Max}(\text{HistN}) / \text{Tot_Blk_Pix}$, wherein $\text{Max}(\text{HistN})$ is a maximum number of pixels in the histogram and Tot_Blk_Pix is a total number of pixels in the block of pixels.

79. The computer program product of claim 78, wherein the computer-readable medium further comprises:

at least one instruction for determining a number of components (NC) using $NC = \text{Sum}(\text{HistN} > \text{Th1})$, wherein NC is a number of pixels that exceed a first threshold value, Th1, based on the histogram.

80. The computer program product of claim 79, wherein the computer-readable medium further comprises:

at least one instruction for determining whether BPR is greater than a second threshold, Th2;

at least one instruction for determining whether NC is less than a third threshold, Th3; and

at least one instruction for classifying the block of pixels as a text-containing region if BPR is greater than Th2 and NC is less than Th3.

81. The computer program product of claim 80, wherein the computer-readable medium further comprises:

at least one instruction for performing pattern matching on the text-containing region.

82. The computer program product of claim 81, wherein the computer-readable medium further comprises:

at least one instruction for locating a non-edge pixel;

at least one instruction for considering the non-edge pixel in at least one corner position;

at least one instruction for locating a plurality of horizontal pixels nearest to the non-edge pixel in the at least one corner position; and

at least one instruction for locating a plurality of vertical pixels nearest to the non-edge pixel in the at least one corner position.

83. The computer program product of claim 82, wherein the computer-readable medium further comprises:

at least one instruction for determining whether any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel; and

at least one instruction for labeling the non-edge pixel as an edge pixel when any of the plurality of horizontal pixels and any of the plurality of vertical pixels is an edge pixel.

84. The computer program product of claim 83, wherein the at least one corner position is selected from the group consisting of: top left corner position, a top right corner position, a bottom left corner position, and a bottom right corner position.

85. A method of inter-edge/inter-midpoint prediction, the method comprising:

receiving previous frame data;

initializing a search radius to be equal to one;

scanning around a co-located point in the previous frame data using the search radius; and

determining whether a reference point is found.

86. The method of claim 85, wherein the previous frame data and the reference point comprises at least one midpoint.

87. The method of claim 85, wherein the previous frame data and the reference point comprises at least one edge.

88. The method of claim 85, further comprising:
setting an inter-predicted value equal to a value of the reference point,
when the reference point is found.
89. The method of claim 85, further comprising:
increasing the search radius by one, when the reference point is not
found; and
determining whether the search radius is equal to a maximum range.
90. The method of claim 89, further comprising:
scanning around a co-located point in the previous frame data using the
search radius, when the search radius is not equal to the maximum range;
determining whether a reference point is found; and
setting an inter-predicted value equal to a value of the reference point,
when the reference point is found.
91. The method of claim 89, further comprising:
setting an inter-predicted value equal to a predetermined default value,
when the search radius is equal to the maximum range.
92. The method of claim 91, wherein the predetermined default value is a
middle value in a range of values of frame data.
93. A device, comprising:
means for receiving previous frame data;
means for initializing a search radius to be equal to one;
means for scanning around a co-located point in the previous frame data
using the search radius; and
means for determining whether a reference point is found.
94. The device of claim 93, wherein the previous frame data and the
reference point comprises at least one midpoint.

95. The device of claim 93, wherein the previous frame data and the reference point comprises at least one edge.
96. The device of claim 93, further comprising:
means for setting an inter-predicted value equal to a value of the reference point, when the reference point is found.
97. The device of claim 93, further comprising:
means for increasing the search radius by one, when the reference point is not found; and
means for determining whether the search radius is equal to a maximum range.
98. The device of claim 97, further comprising:
means for scanning around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range;
means for determining whether a reference point is found; and
means for setting an inter-predicted value equal to a value of the reference point, when the reference point is found.
99. The device of claim 97, further comprising:
means for setting an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range.
100. The device of claim 99, wherein the predetermined default value is a middle value in a range of values of frame data.
101. A device, comprising:
a processor, wherein the processor is operable to:
receive previous frame data;
initialize a search radius to be equal to one;

scan around a co-located point in the previous frame data using the search radius; and
determine whether a reference point is found.

102. The device of claim 101, wherein the previous frame data and the reference point comprises at least one midpoint.

103. The device of claim 101, wherein the previous frame data and the reference point comprises at least one edge.

104. The device of claim 101, wherein the processor is further operable to:
set an inter-predicted value equal to a value of the reference point, when the reference point is found.

105. The device of claim 101, wherein the processor is further operable to:
increase the search radius by one, when the reference point is not found;
and
determine whether the search radius is equal to a maximum range.

106. The device of claim 105, wherein the processor is further operable to:
scan around a co-located point in the previous frame data using the search radius, when the search radius is not equal to the maximum range;
determine whether a reference point is found; and
set an inter-predicted value equal to a value of the reference point, when the reference point is found.

107. The device of claim 105, wherein the processor is further operable to:
set an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range.

108. The device of claim 107, wherein the predetermined default value is a middle value in a range of values of frame data.

109. A computer program product, comprising:
computer-readable medium, comprising:
at least one instruction for receiving previous frame data;
at least one instruction for initializing a search radius to be equal to one;
at least one instruction for scanning around a co-located point in the
previous frame data using the search radius; and
at least one instruction for determining whether a reference point is
found.

110. The computer program product of claim 109, wherein the previous frame
data and the reference point comprises at least one midpoint.

111. The computer program product of claim 109, wherein the previous frame
data and the reference point comprises at least one edge.

112. The computer program product of claim 109, wherein the computer-
readable medium further comprises:
at least one instruction for setting an inter-predicted value equal to a
value of the reference point, when the reference point is found.

113. The computer program product of claim 109, wherein the computer-
readable medium further comprises:
at least one instruction for increasing the search radius by one, when the
reference point is not found; and
at least one instruction for determining whether the search radius is equal
to a maximum range.

114. The computer program product of claim 113, wherein the computer-
readable medium further comprises:
at least one instruction for scanning around a co-located point in the
previous frame data using the search radius, when the search radius is not equal
to the maximum range;
at least one instruction for determining whether a reference point is
found; and

at least one instruction for setting an inter-predicted value equal to a value of the reference point, when the reference point is found.

115. The computer program product of claim 113, wherein the computer-readable medium further comprises:

at least one instruction for setting an inter-predicted value equal to a predetermined default value, when the search radius is equal to the maximum range.

116. The computer program product of claim 115, wherein the predetermined default value is a middle value in a range of values of frame data.

117. A method of intra-edge/intra-midpoint prediction, the method comprising:

determining a plurality of sent pixels;

dividing an image into a plurality of pixel blocks having a predetermined size; and

determining whether any pixels within each pixel block is within the plurality of sent pixels.

118. The method of claim 117, wherein the plurality of sent pixels comprises a plurality of edge pixels and a plurality of midpoint pixels.

119. The method of claim 117, wherein the predetermined size is set equal to a scale factor in a last iteration.

120. The method of claim 117, further comprising:

setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels.

121. The method of claim 120, further comprising:

setting a predictor value equal to a mean of all sent pixel values within the block of pixels.

122. The method of claim 121, further comprising:
determining a residue of unsent pixels.
123. The method of claim 122, wherein the residue of unsent pixels comprises an actual value minus the predictor values.
124. A device, comprising:
means for determining a plurality of sent pixels;
means for dividing an image into a plurality of pixel blocks having a predetermined size; and
means for determining whether any pixels within each pixel block is within the plurality of sent pixels.
125. The device of claim 124, wherein the plurality of sent pixels comprises a plurality of edge pixels and a plurality of midpoint pixels.
126. The device of claim 124, wherein the predetermined size is set equal to a scale factor in a last iteration.
127. The device of claim 124, further comprising:
means for setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels.
128. The device of claim 127, further comprising:
means for setting a predictor value equal to a mean of all sent pixel values within the block of pixels.
129. The device of claim 128, further comprising:
means for determining a residue of unsent pixels.
130. The device of claim 129, wherein the residue of unsent pixels comprises an actual value minus the predictor values.

131. A device, comprising:
a processor, wherein the processor is operable to:
determine a plurality of sent pixels;
divide an image into a plurality of pixel blocks having a predetermined size; and
determine whether any pixels within each pixel block is within the plurality of sent pixels.
132. The device of claim 131, wherein the plurality of sent pixels comprises a plurality of edge pixels and a plurality of midpoint pixels.
133. The device of claim 131, wherein the predetermined size is set equal to a scale factor in a last iteration.
134. The device of claim 131, wherein the processor is further operable to:
set a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels.
135. The device of claim 134, wherein the processor is further operable to:
set a predictor value equal to a mean of all sent pixel values within the block of pixels.
136. The device of claim 135, wherein the processor is further operable to:
determine a residue of unsent pixels.
137. The device of claim 136, wherein the residue of unsent pixels comprises an actual value minus the predictor values.
138. A computer program product, comprising:
a computer-readable medium, comprising:
at least one instruction for determining a plurality of sent pixels;
at least one instruction for dividing an image into a plurality of pixel blocks having a predetermined size; and

at least one instruction for determining whether any pixels within each pixel block is within the plurality of sent pixels.

139. The computer program product of claim 138, wherein the plurality of sent pixels comprises a plurality of edge pixels and a plurality of midpoint pixels.

140. The computer program product of claim 138, wherein the predetermined size is set equal to a scale factor in a last iteration.

141. The computer program product of claim 138, wherein the computer-readable medium further comprises:

at least one instruction for setting a predictor value equal to a predetermined default value, when none of the pixels within the pixel block is within the plurality of sent pixels.

142. The computer program product of claim 141, wherein the computer-readable medium further comprises:

at least one instruction for setting a predictor value equal to a mean of all sent pixel values within the block of pixels.

143. The computer program product of claim 142, wherein the computer-readable medium further comprises:

at least one instruction for determining a residue of unsent pixels.

144. The computer program product of claim 143, wherein the residue of unsent pixels comprises an actual value minus the predictor values.

145. A method of refining midpoints, the method comprising:

receiving a plurality of midpoints;

scanning a plurality of pixels in a predetermined number of directions using a predetermined step angle from each of the plurality of midpoint pixels;

determining whether a current pixel location is greater than a maximum scan distance; and

determining whether a current pixel is an edge pixel if the current pixel

location is not greater than the maximum scan distance.

146. The method of claim 145, further comprising:

determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel; and

removing the current pixel from a midpoint map when the current pixel is a midpoint pixel.

147. The method of claim 145, wherein the predetermined number of directions comprises eight directions.

148. The method of claim 147, wherein the predetermined step angle is forty-five degrees (45°).

149. A device, comprising:

means for receiving a plurality of midpoints;

means for scanning a plurality of pixels in a predetermined number of directions using a predetermined step angle from each of the plurality of midpoint pixels;

means for determining whether a current pixel location is greater than a maximum scan distance and

means for determining whether a current pixel is an edge pixel if the current pixel location is not greater than a maximum scan distance.

150. The device of claim 149, further comprising:

means for determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel;

means for removing the current pixel from a midpoint map when the current pixel is a midpoint pixel.

151. The device of claim 149, wherein the predetermined number of directions comprises eight directions.

152. The device of claim 151, wherein the predetermined step angle is forty-five degrees (45°).

153. A device, the device comprising:
a processor, wherein the processor is operable to:
receive a plurality of midpoints;
scan a plurality of pixels in a predetermined number of directions
using a predetermined step angle from each of the plurality of midpoint pixels;
determine whether a current pixel location is greater than a maximum
scan distance; and
determine whether a current pixel is an edge pixel if the current pixel
location is not greater than the maximum scan distance.

154. The device of claim 153, wherein the processor is further operable:
determine whether the current pixel is a midpoint pixel if the current
pixel is not an edge pixel; and
remove the current pixel from a midpoint map when the current pixel is a
midpoint pixel.

155. The device of claim 153, wherein the predetermined number of
directions comprises eight directions.

156. The device of claim 155, wherein the predetermined step angle is forty-five degrees (45°).

157. A computer program product, comprising:
a computer-readable medium, comprising:
at least one instruction for receiving a plurality of midpoints;
at least one instruction for scanning a plurality of pixels in a
predetermined number of directions using a predetermined step angle from each
of the plurality of midpoint pixels;
at least one instruction for determining whether a current pixel location is
greater than a maximum scan distance; and
at least one instruction for determining whether a current pixel is an edge

pixel if the current pixel location is not greater than the maximum scan distance.

158. The computer program product of claim 157, wherein the computer-readable medium further comprises:

at least one instruction for determining whether the current pixel is a midpoint pixel if the current pixel is not an edge pixel; and

at least one instruction for removing the current pixel from a midpoint map when the current pixel is a midpoint pixel.

159. The computer program product of claim 157, wherein the predetermined number of directions comprises eight directions.

160. The computer program product of claim 159, wherein the predetermined step angle is forty-five degrees (45°).

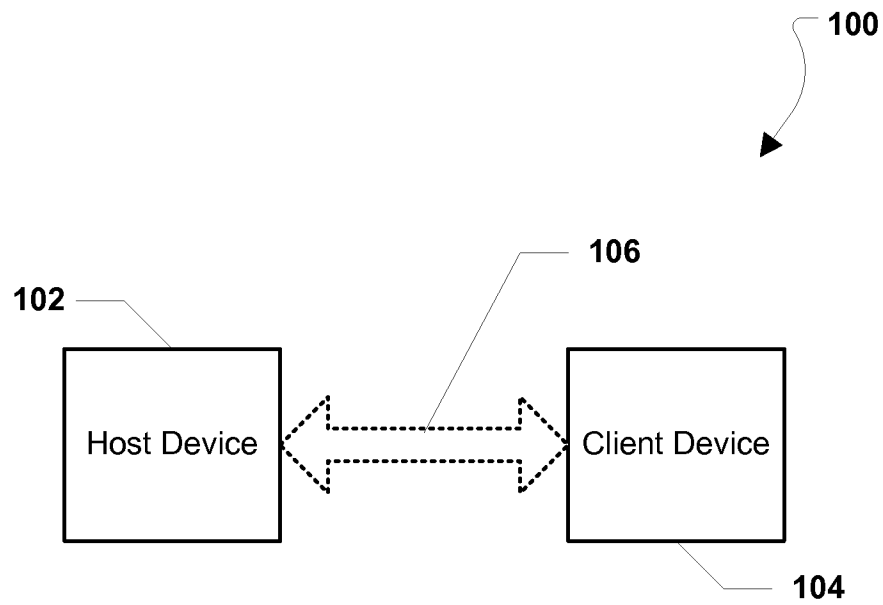


FIG. 1

2/20

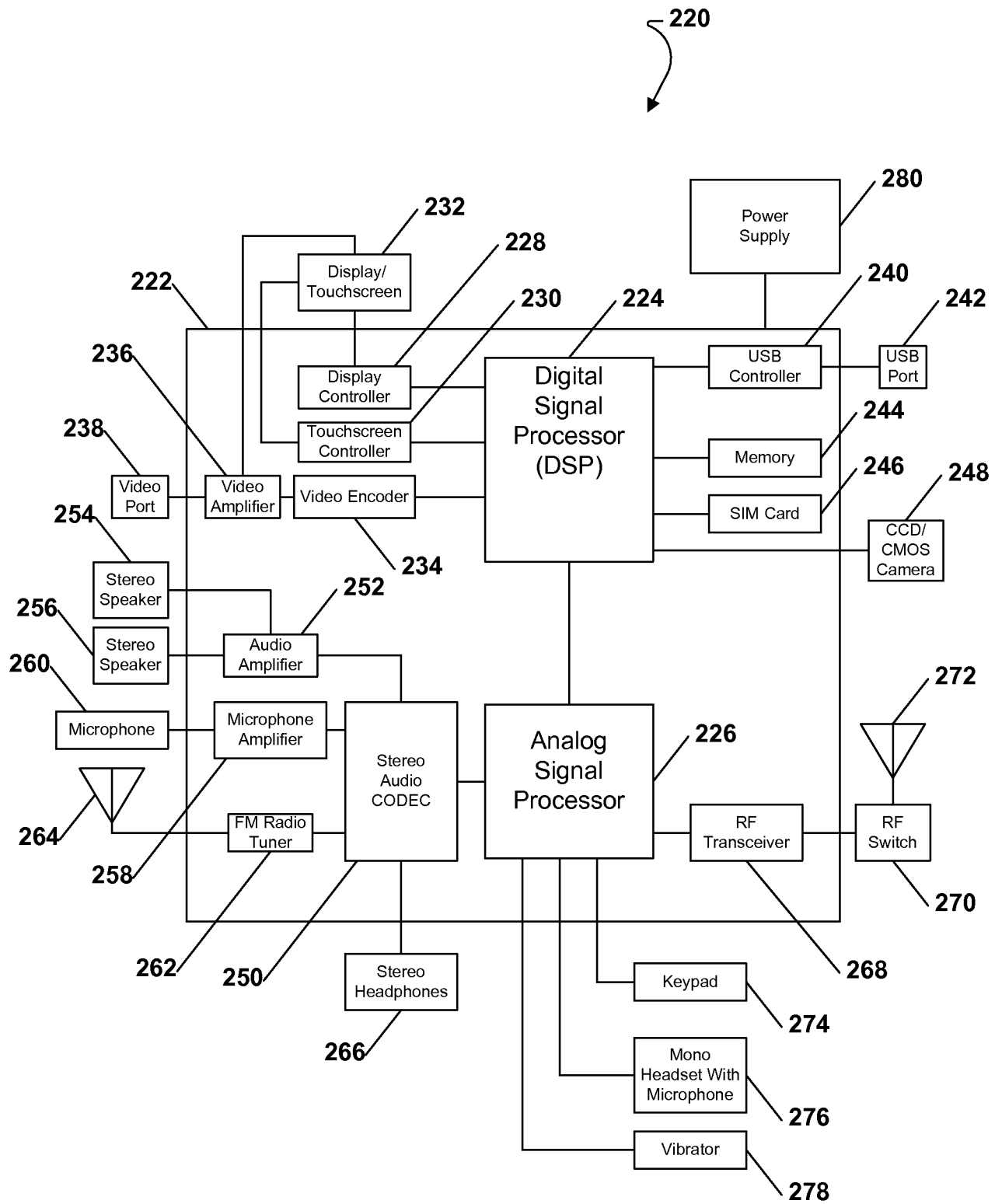


FIG. 2

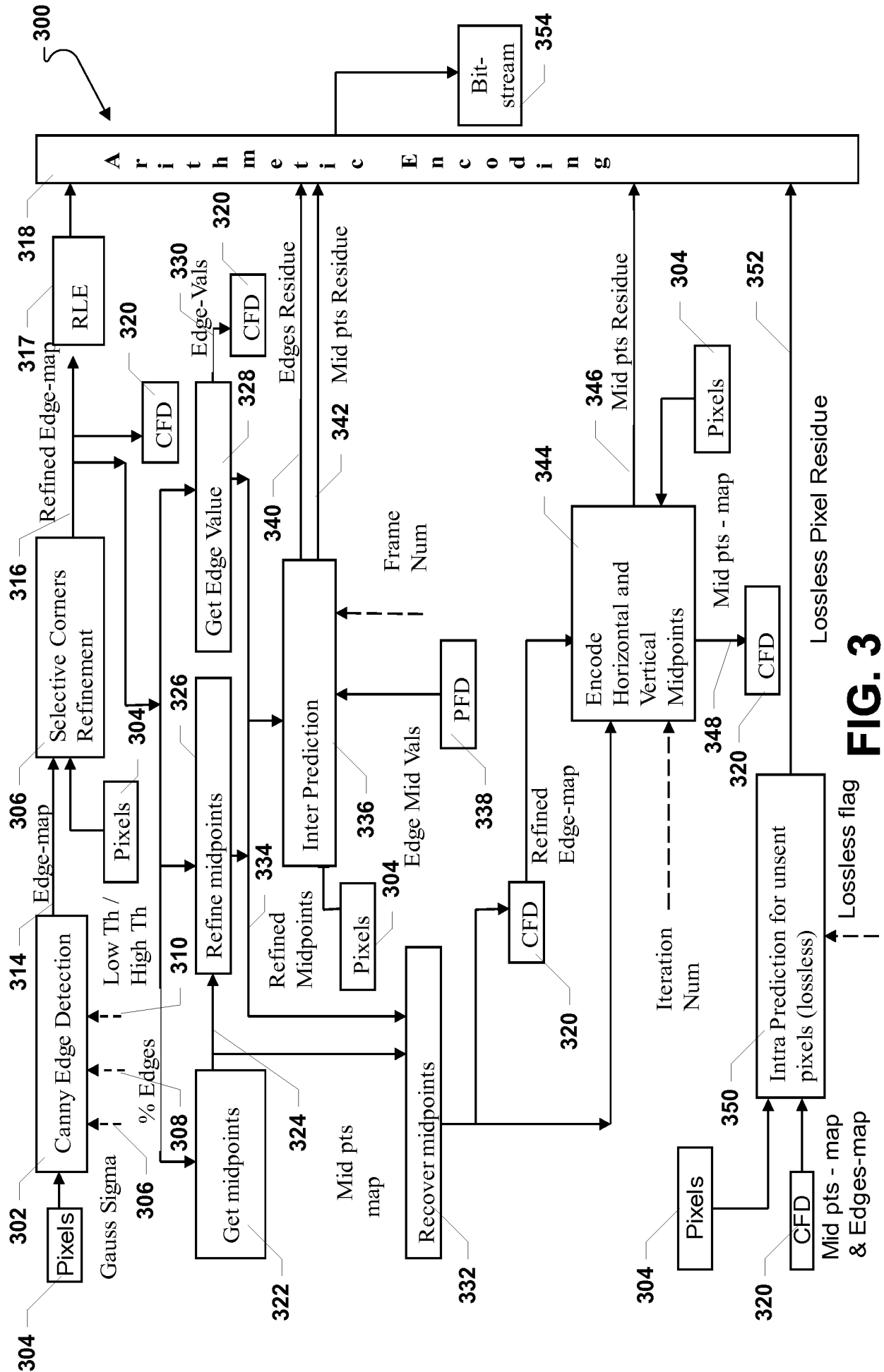
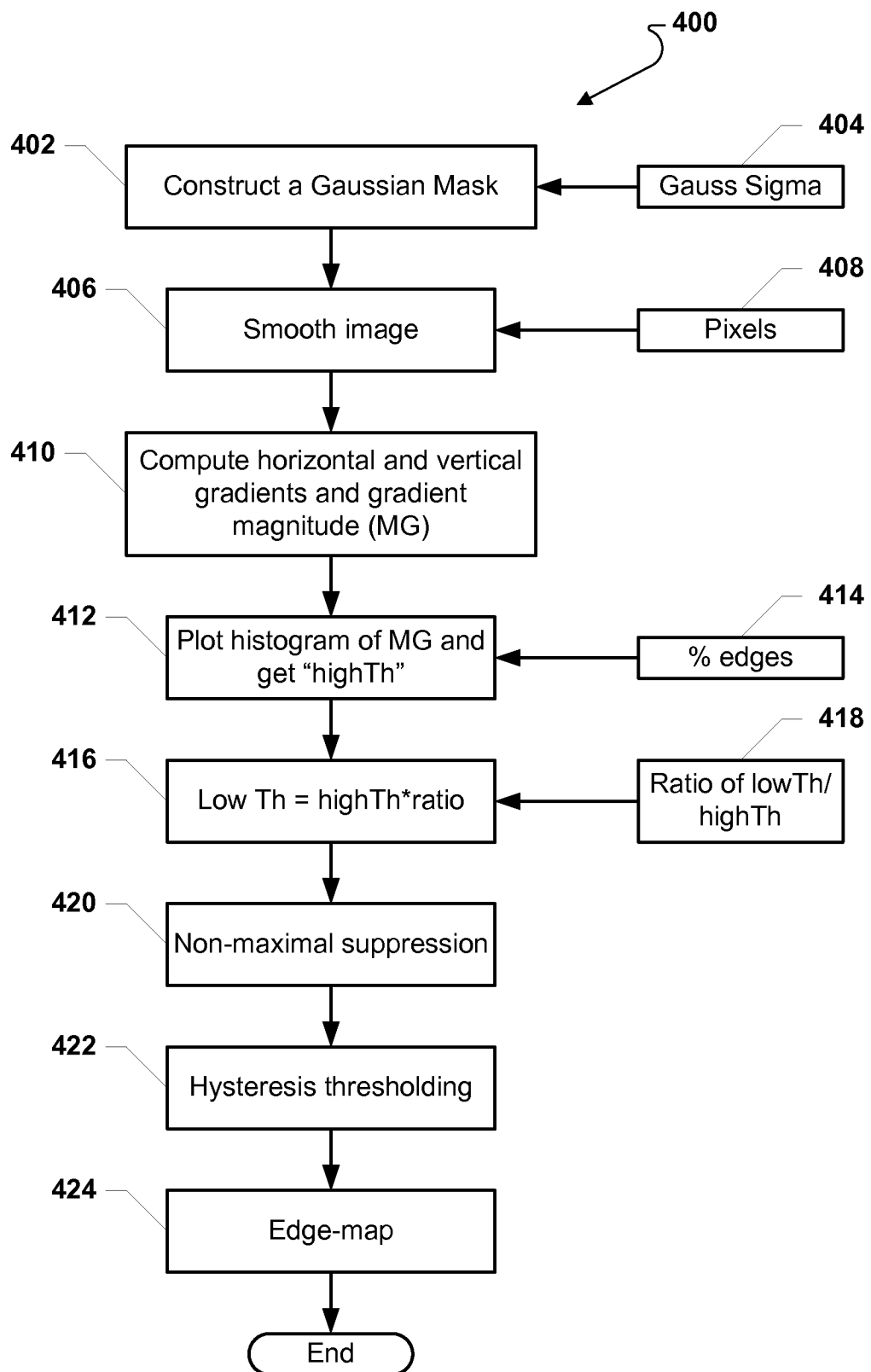
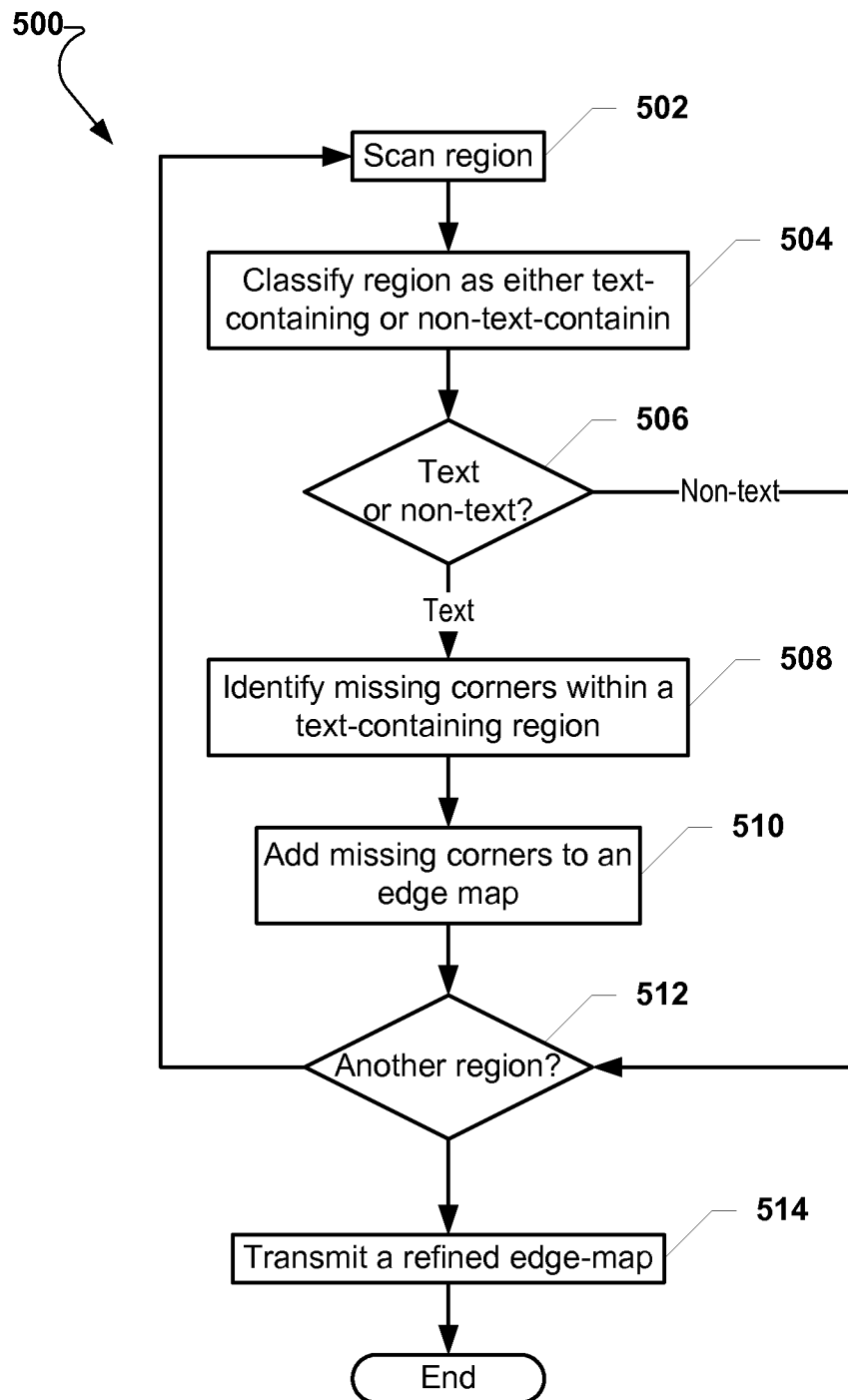


FIG. 3

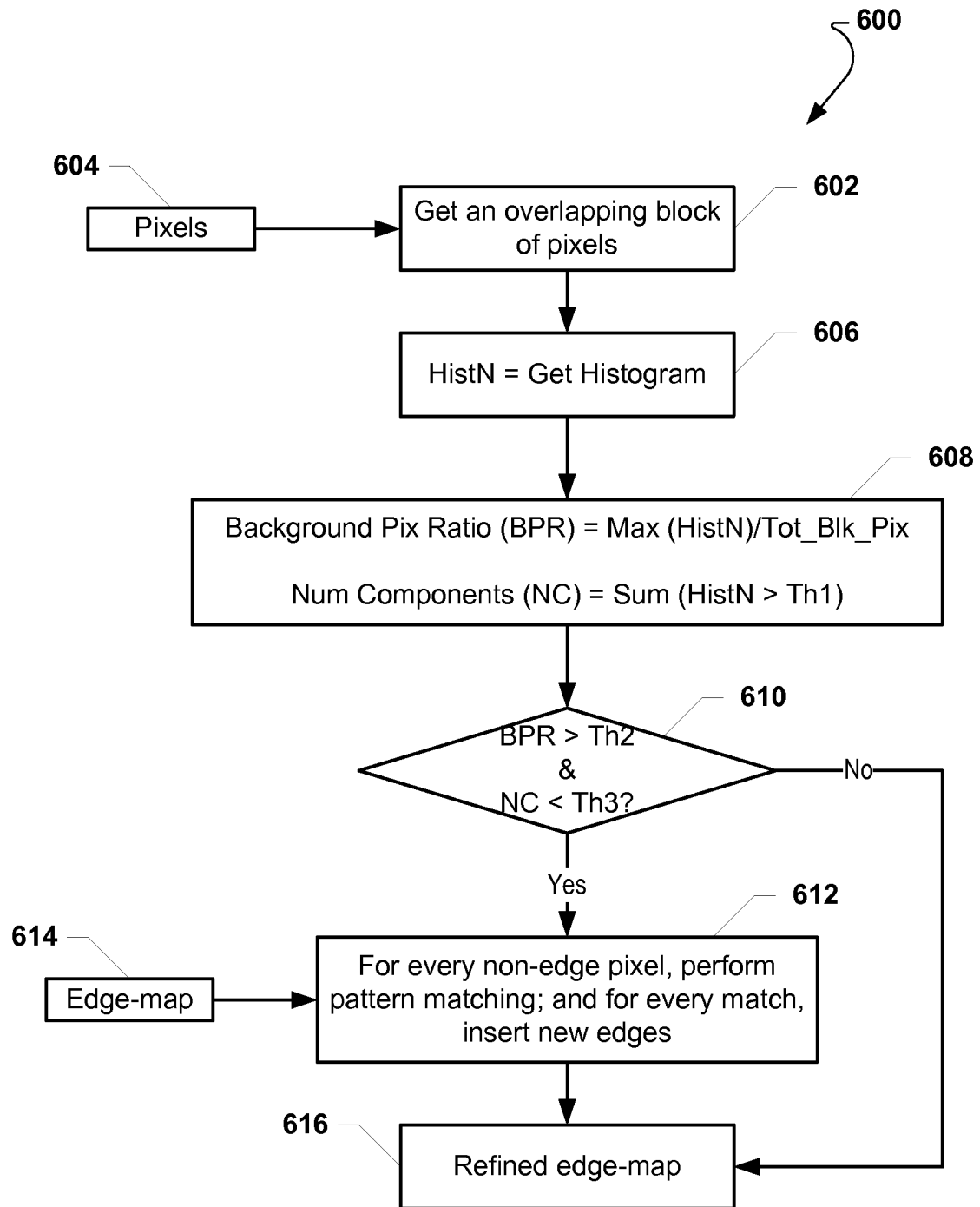
4/20

**FIG. 4**

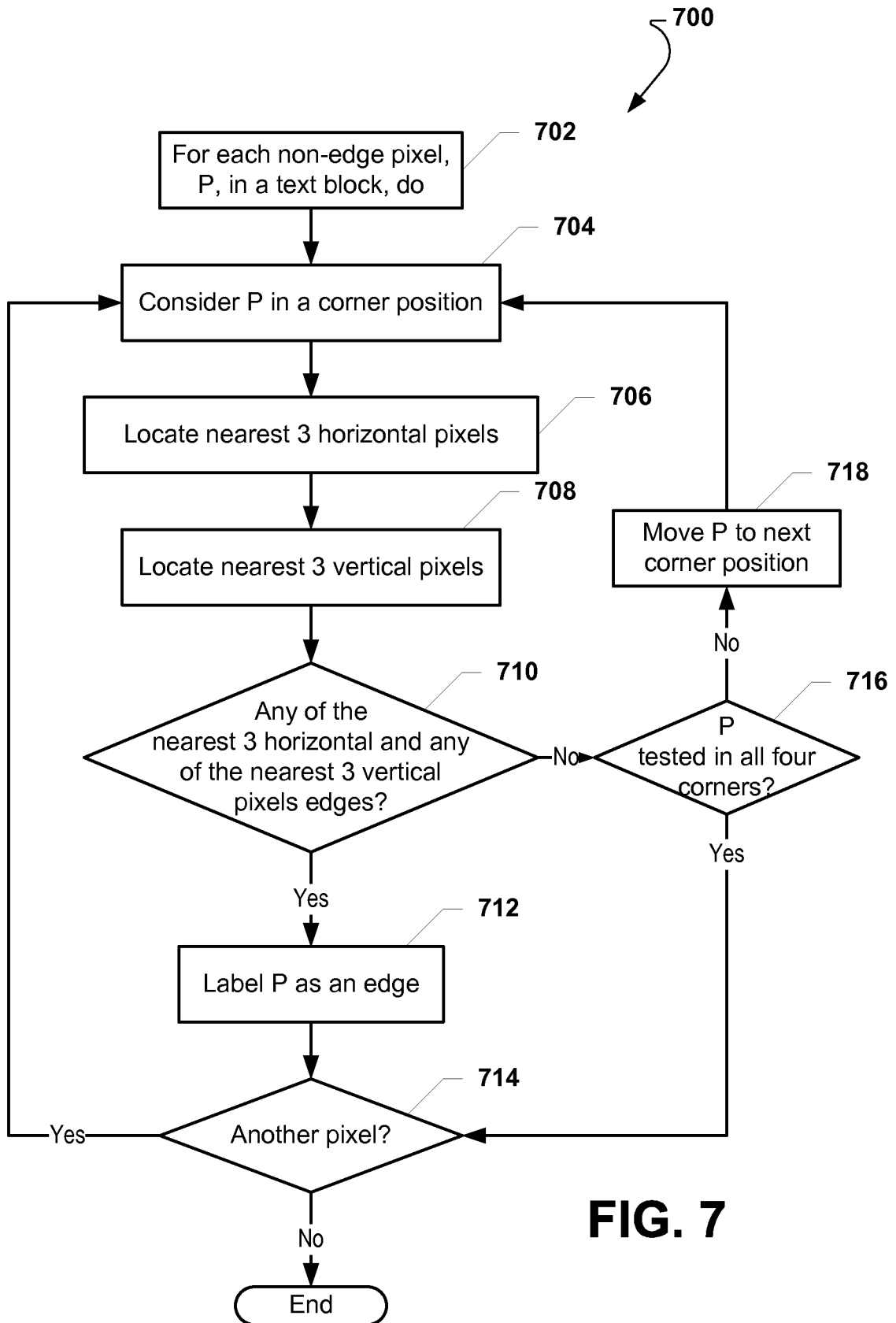
5/20

**FIG. 5**

6/20

**FIG. 6**

7/20

**FIG. 7**

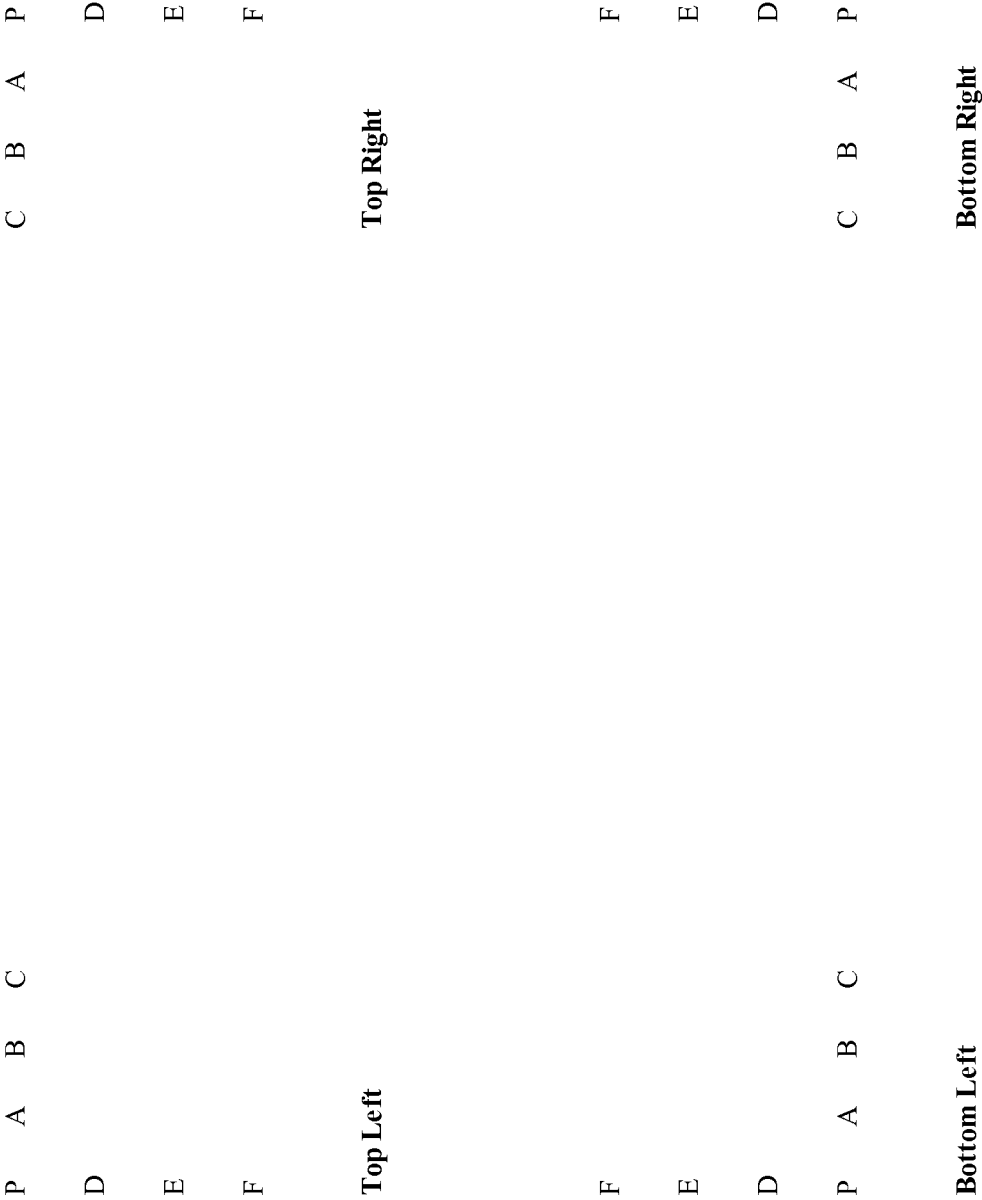
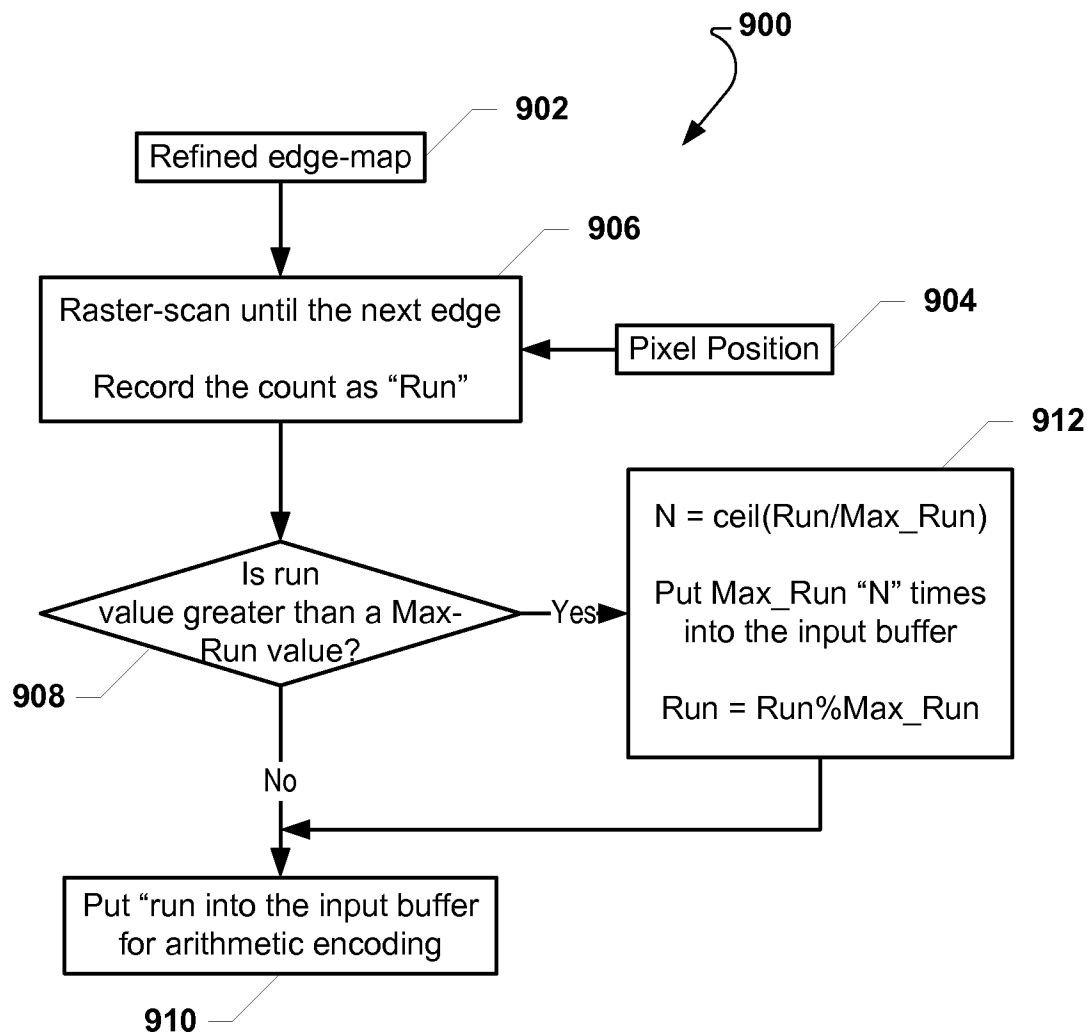
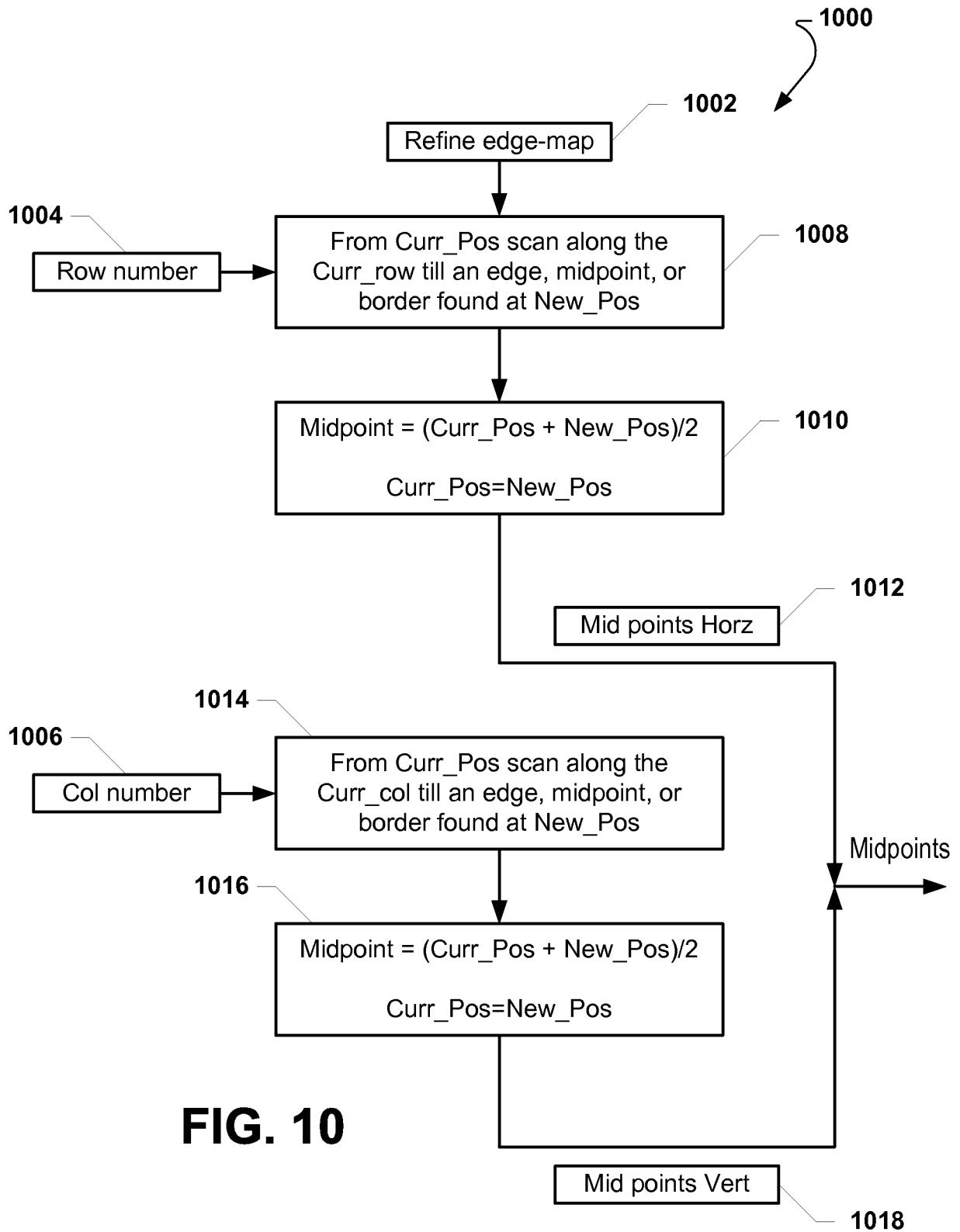


FIG. 8

9/20

**FIG. 9**

10/20

**FIG. 10**

11/20

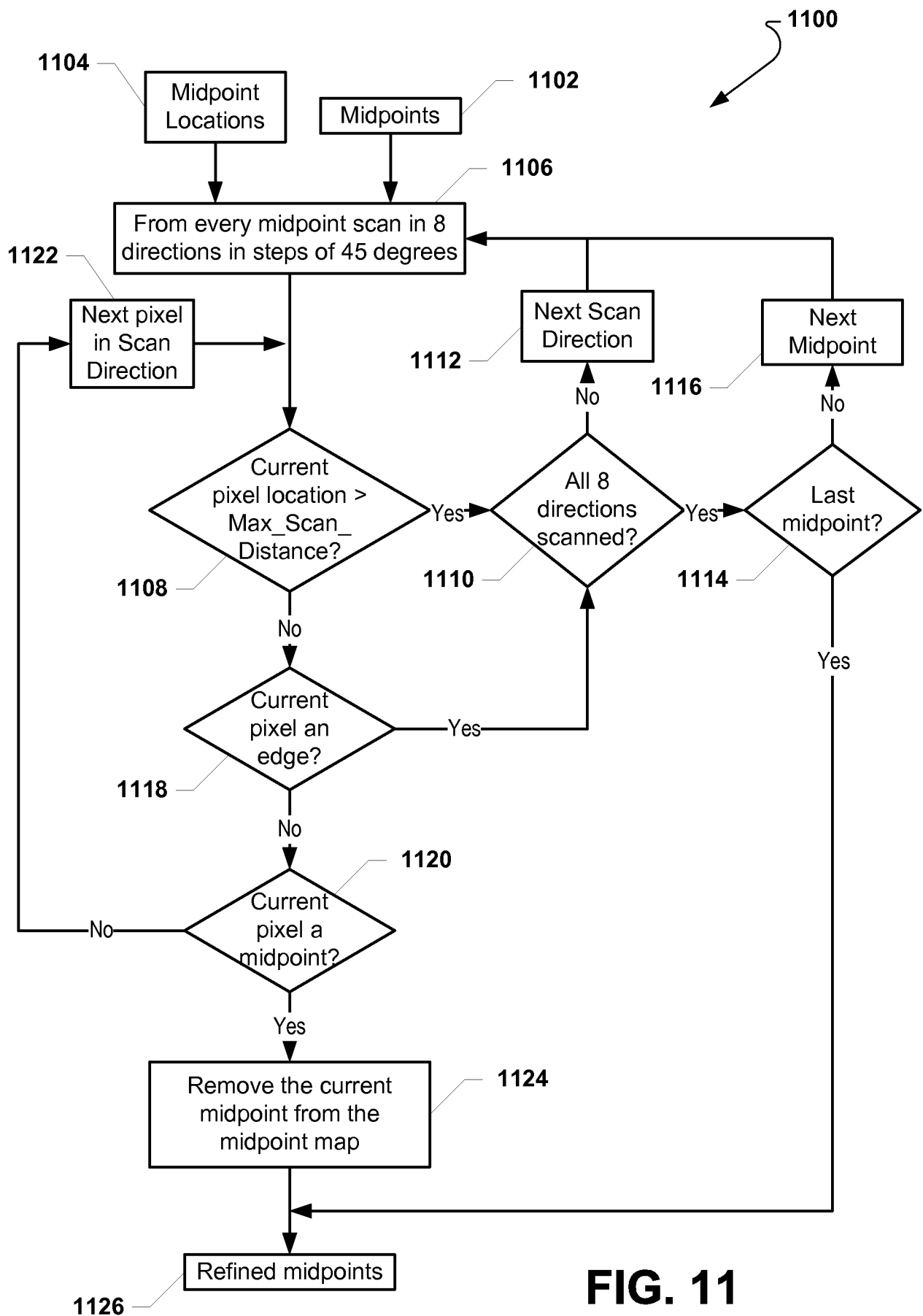


FIG. 11

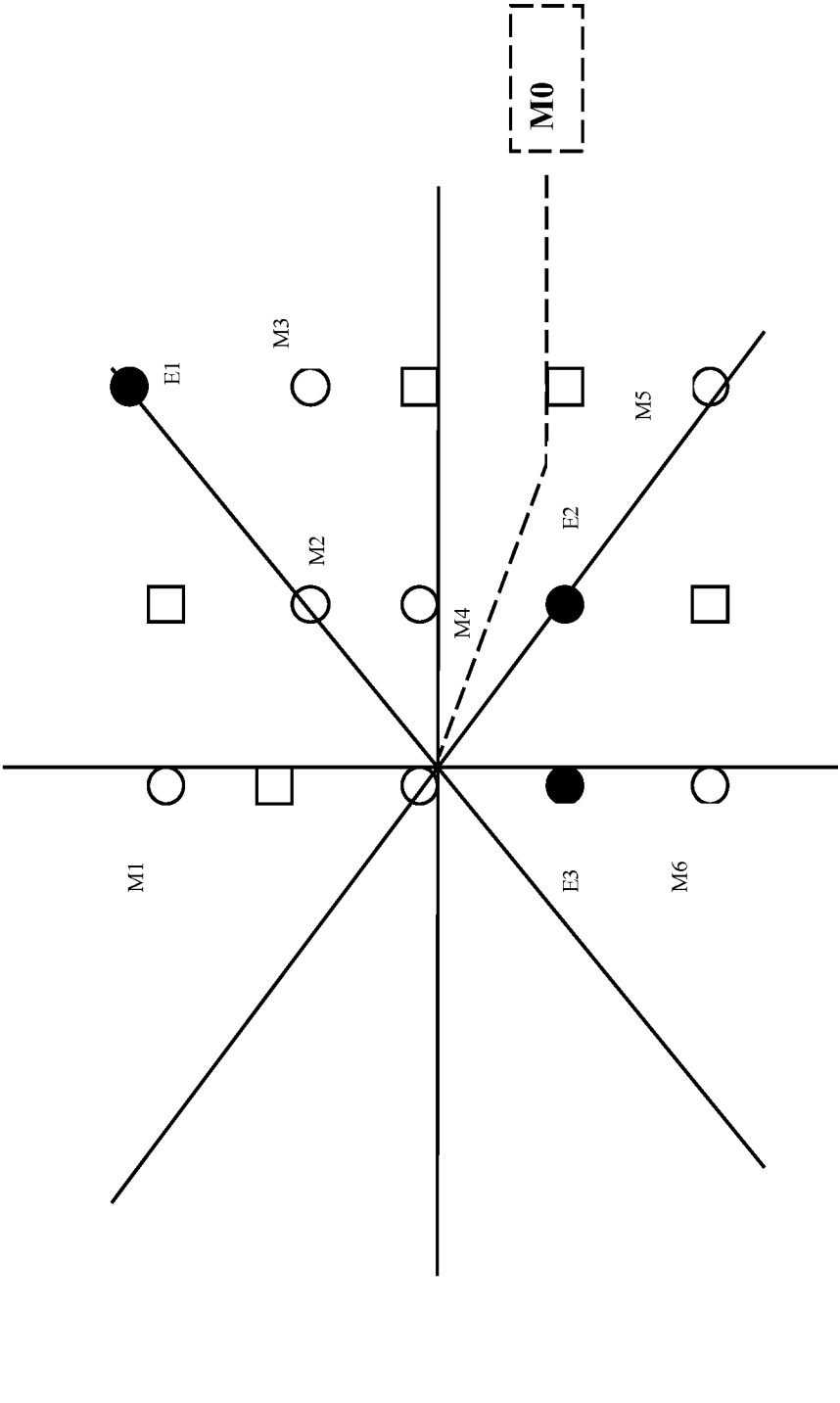
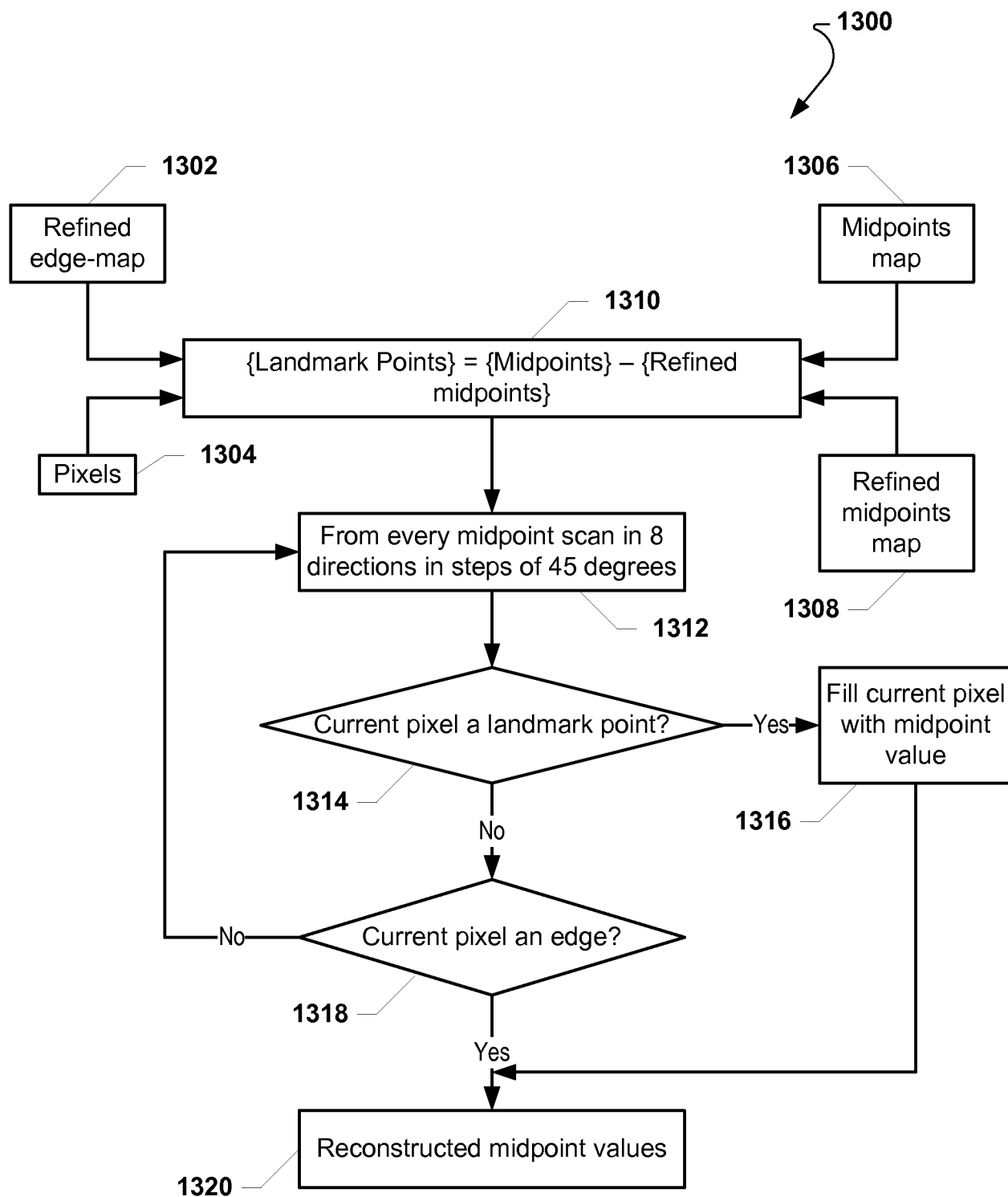
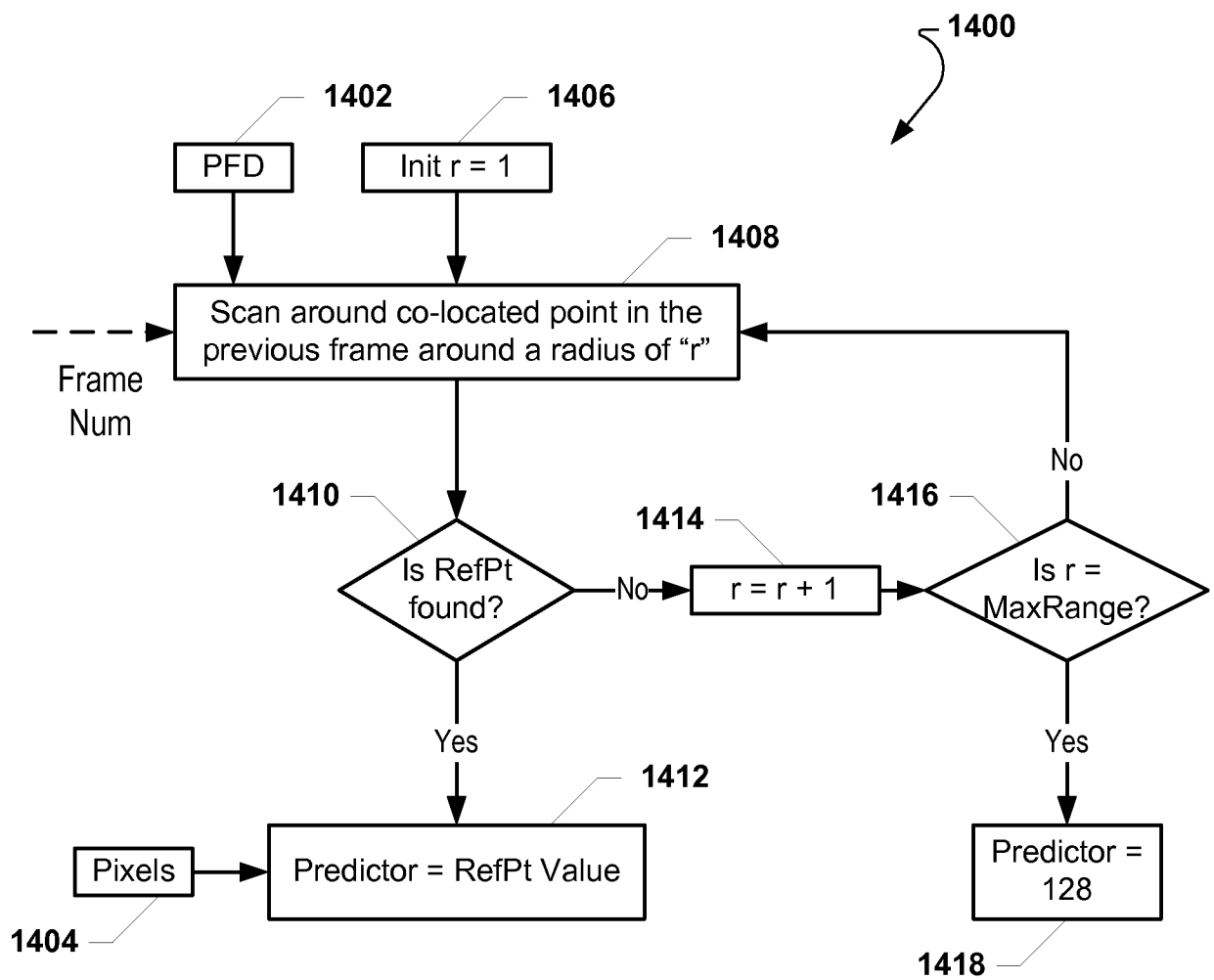


FIG. 12

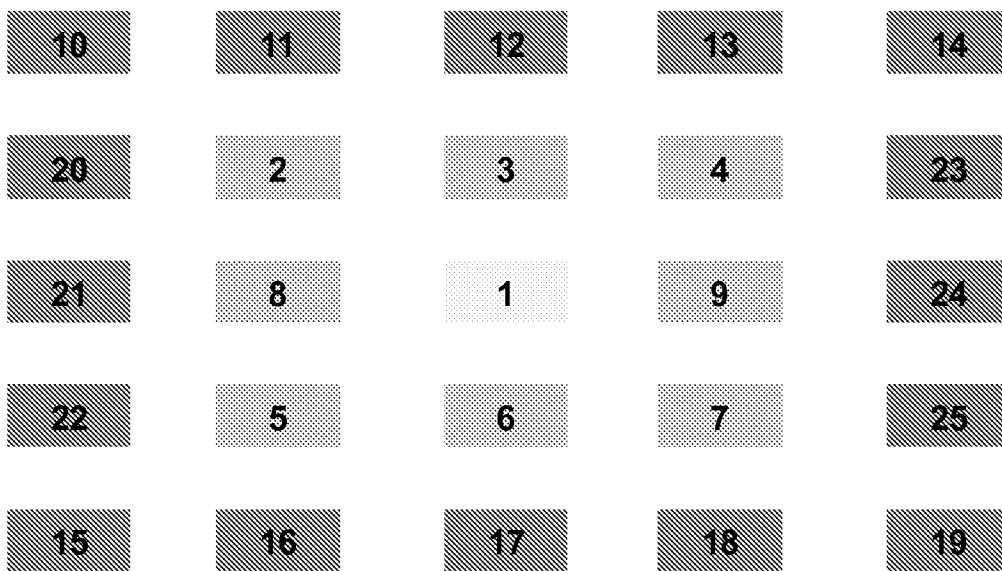
13/20

**FIG. 13**

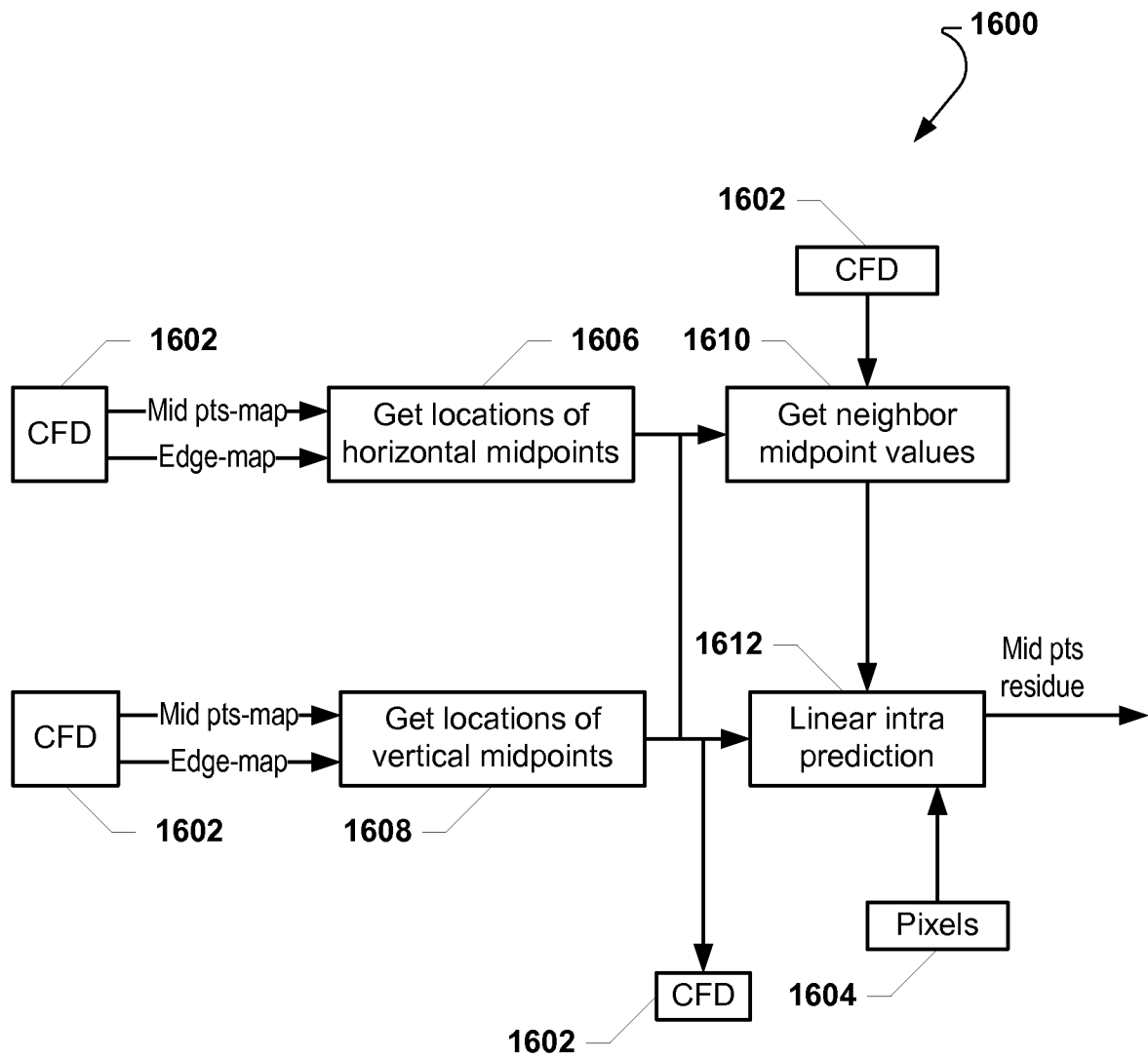
14/20

**FIG. 14**

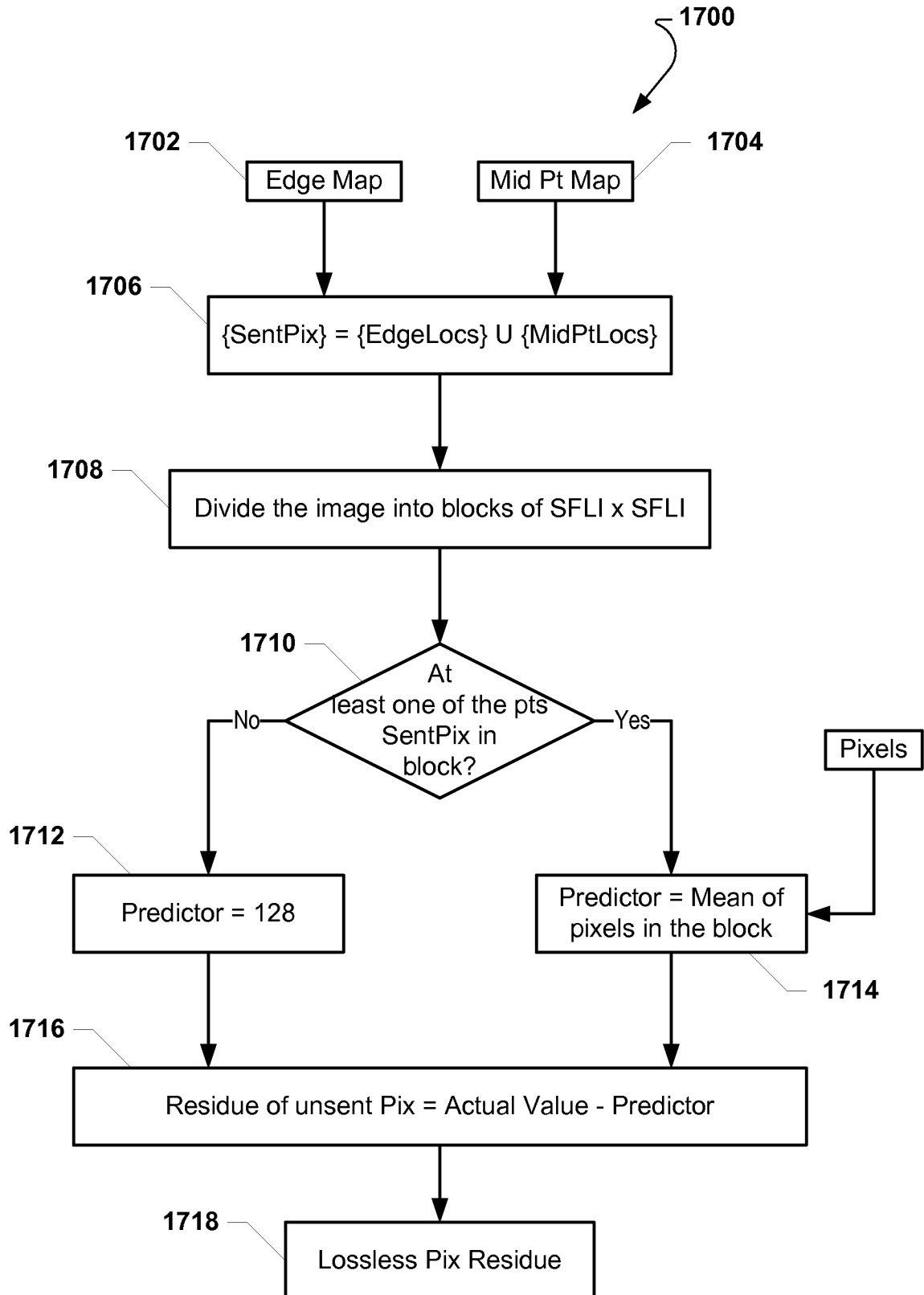
15/20

**FIG. 15**

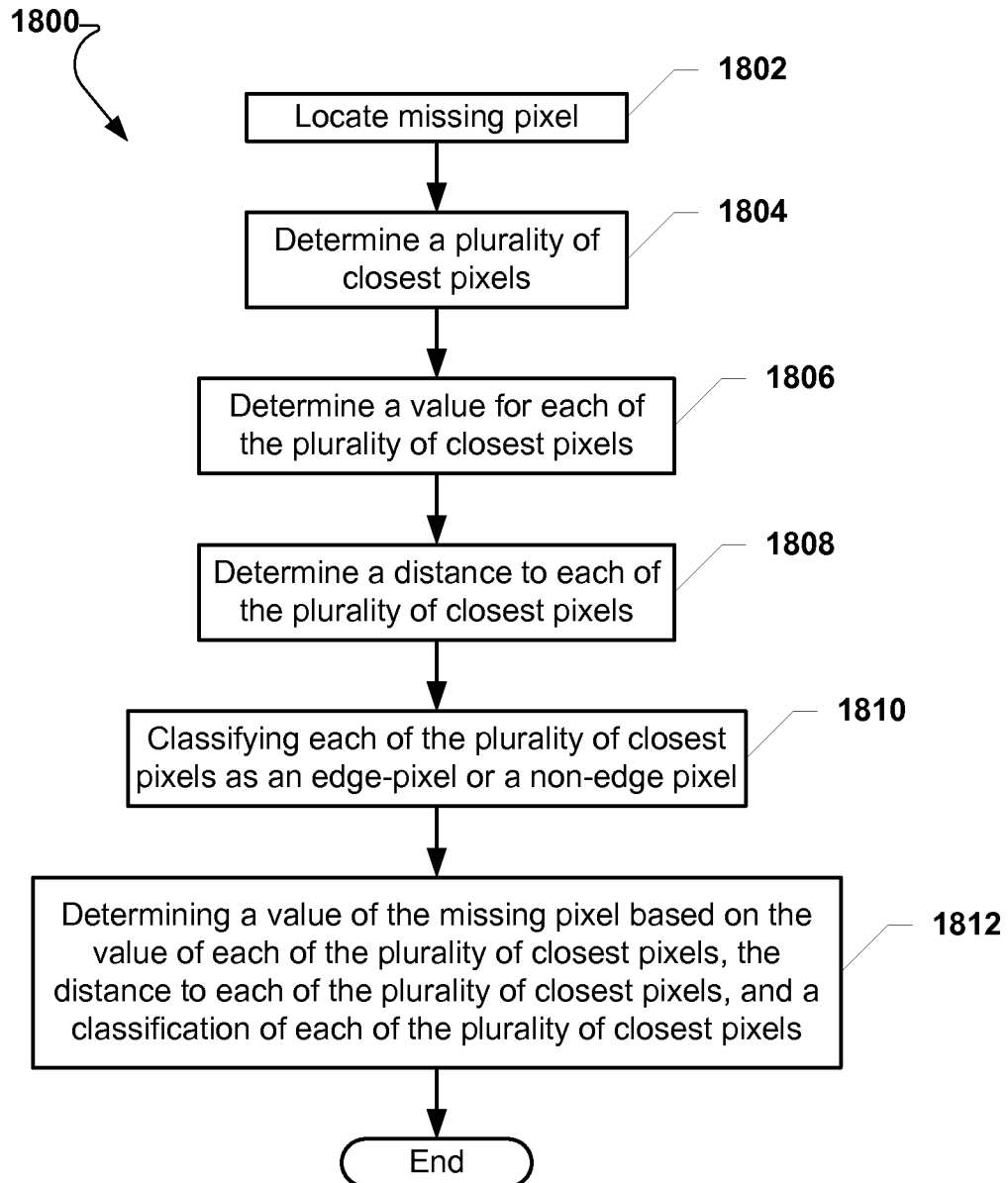
16/20

**FIG. 16**

17/20

**FIG. 17**

18/20

**FIG. 18**

19/20

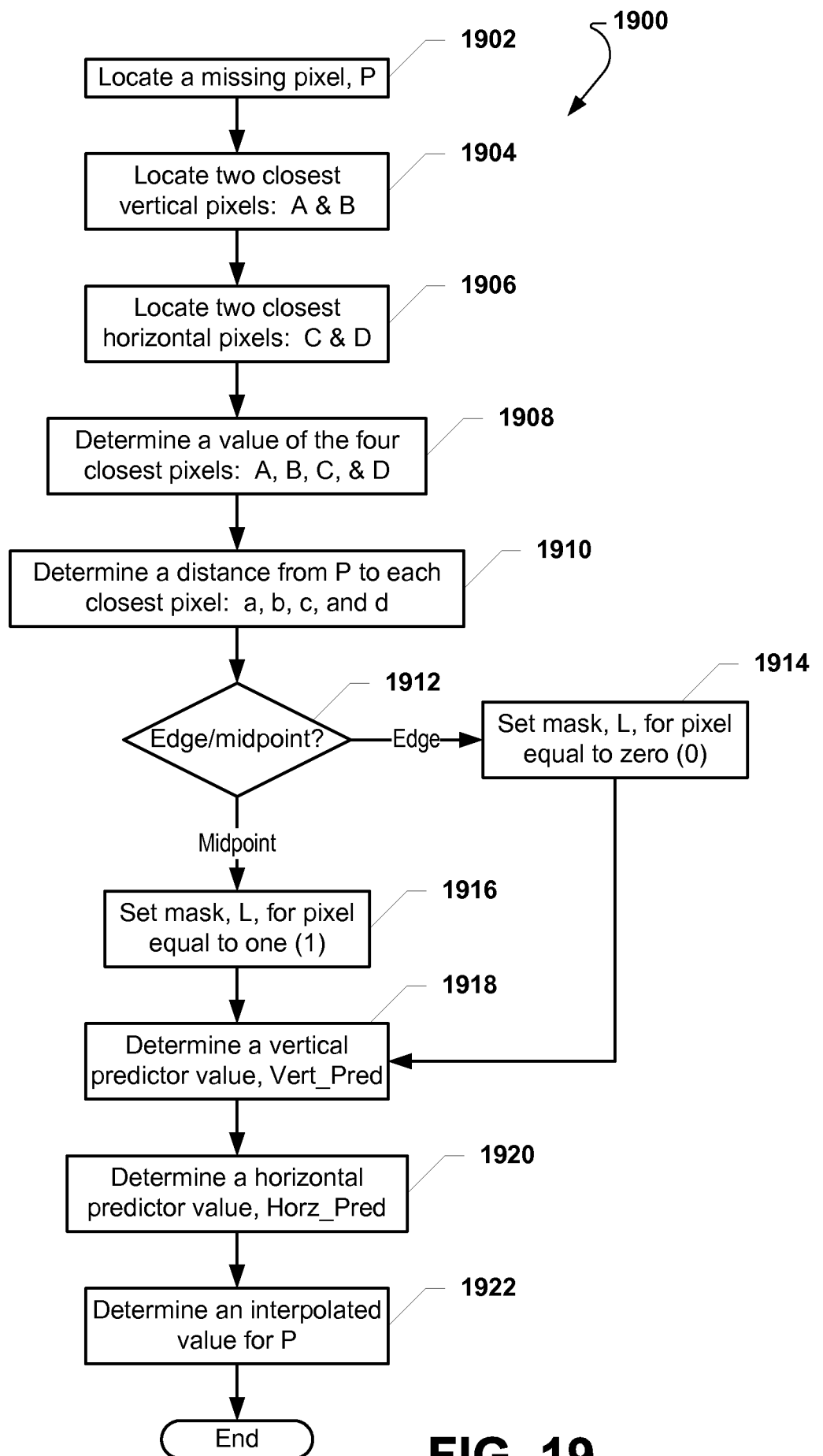


FIG. 19

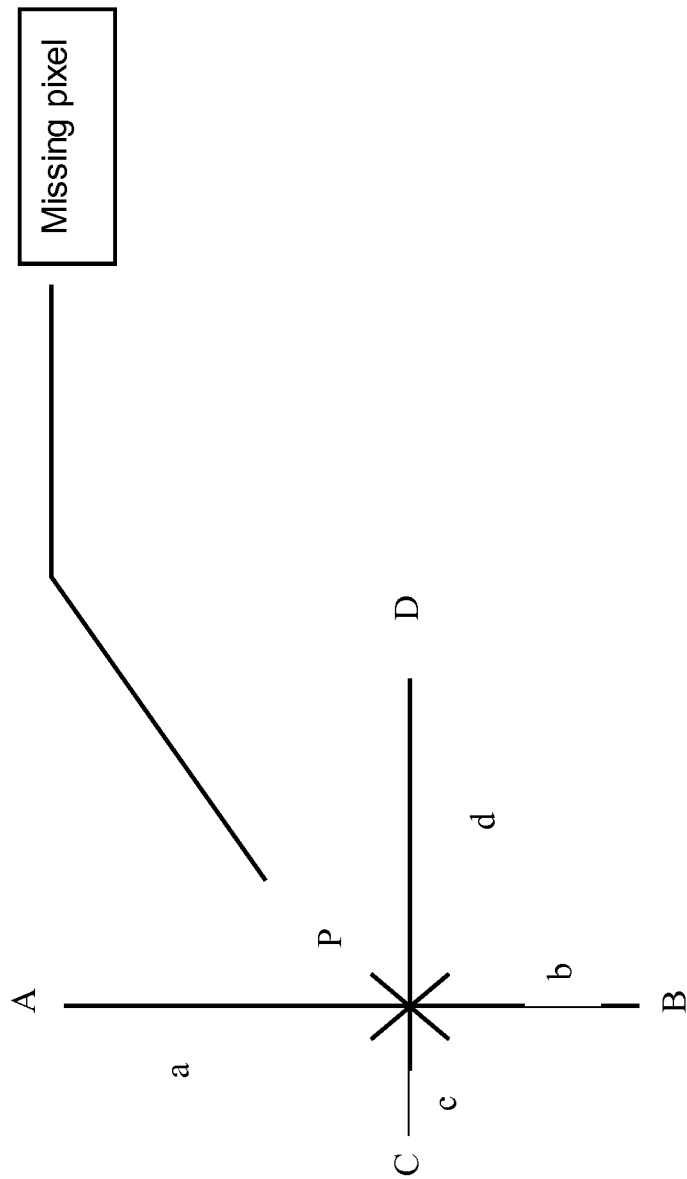


FIG. 20