Figure 1
Prior Art

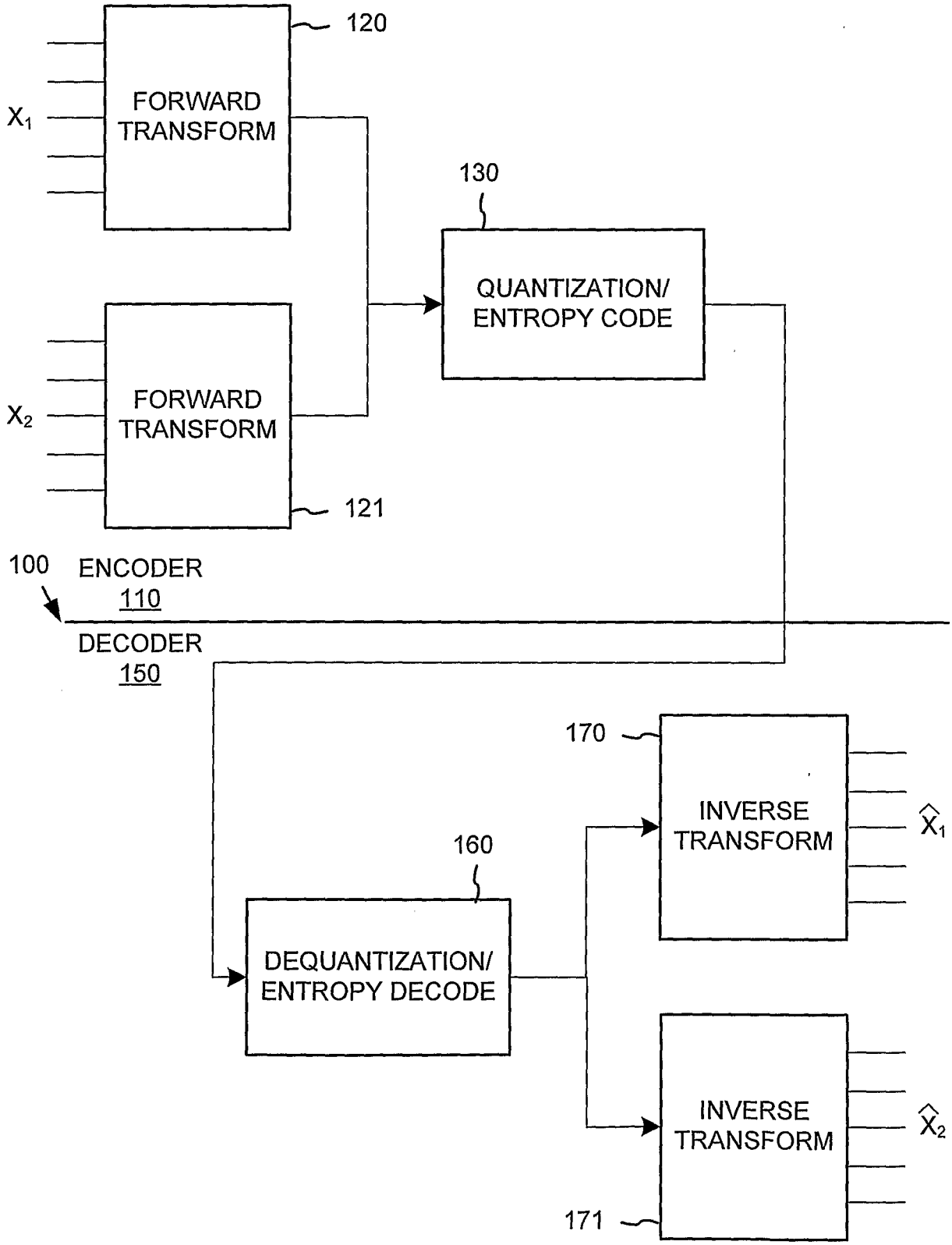


Figure 2

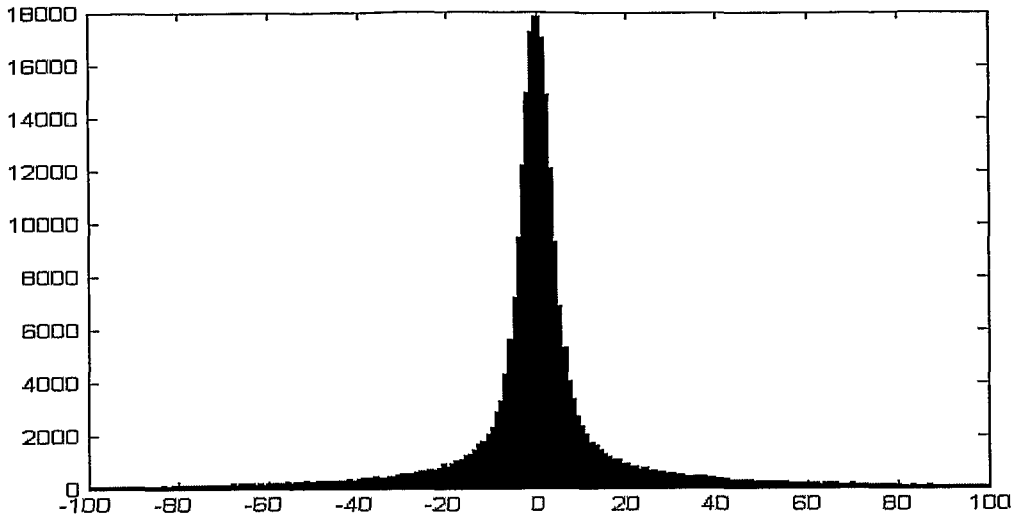


Figure 3

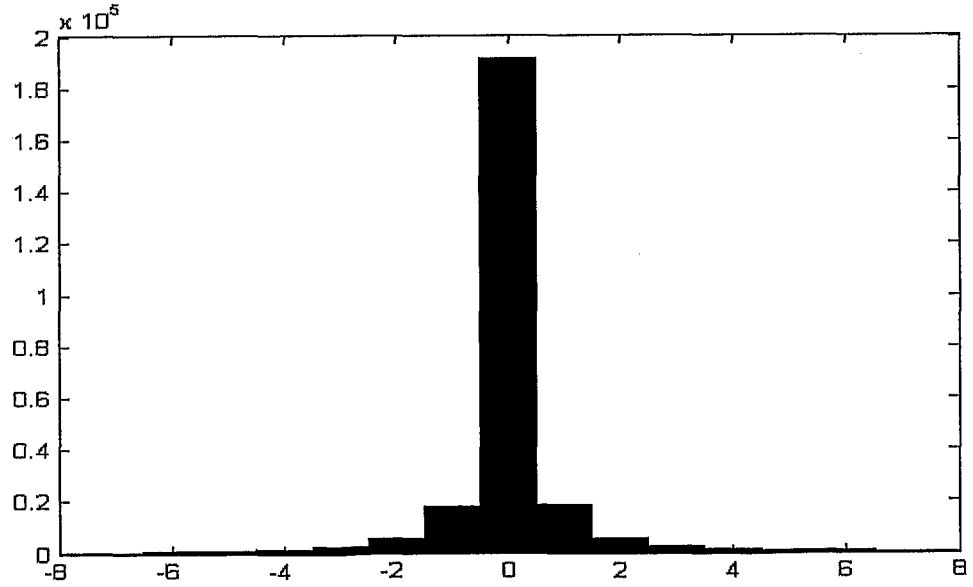


Figure 4

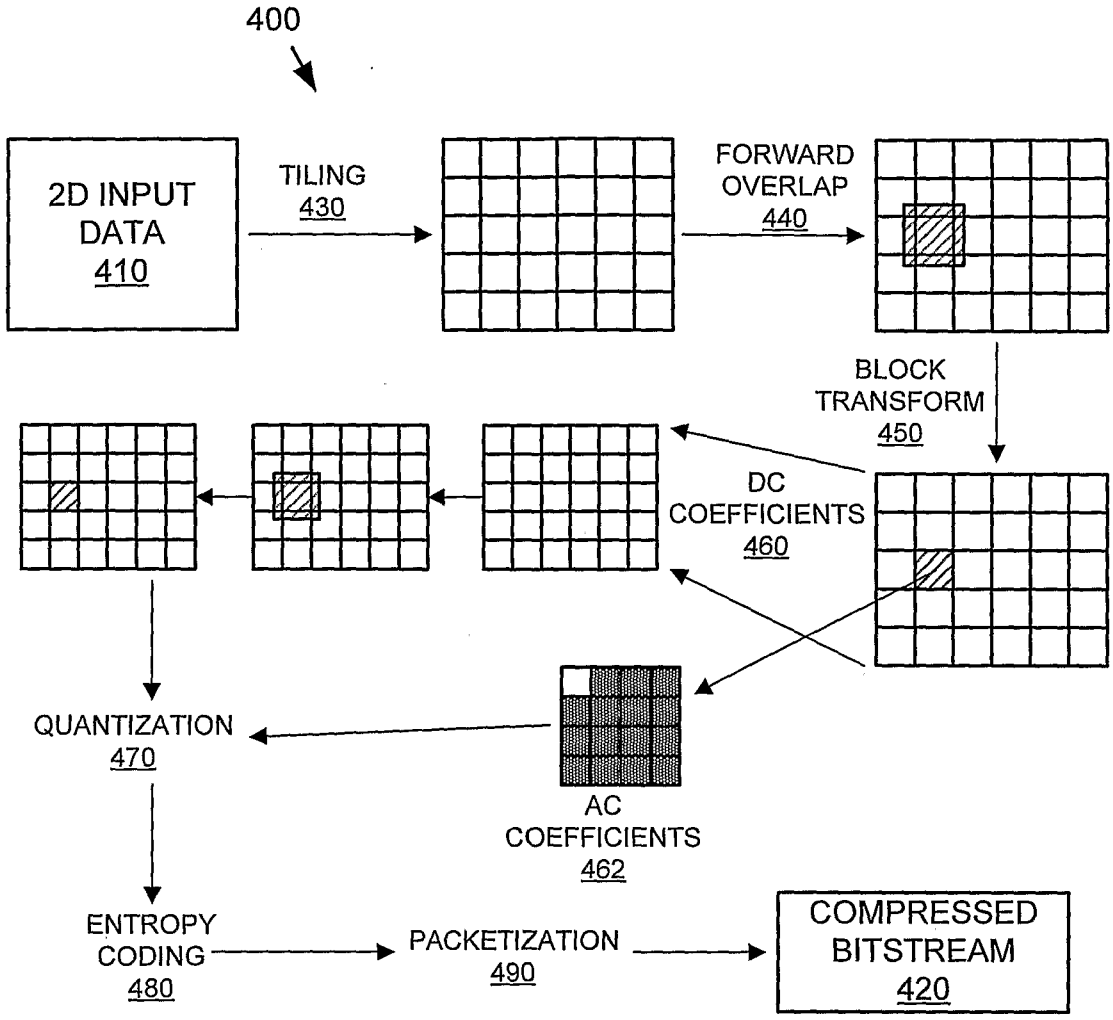

400
2D INPUT DATA 410
TILING 430
FORWARD OVERLAP 440
BLOCK TRANSFORM 450
DC COEFFICIENTS 460
QUANTIZATION 470
AC COEFFICIENTS 462
ENTROPY CODING 480
PACKETIZATION 490
COMPRESSED BITSTREAM 420

Figure 5

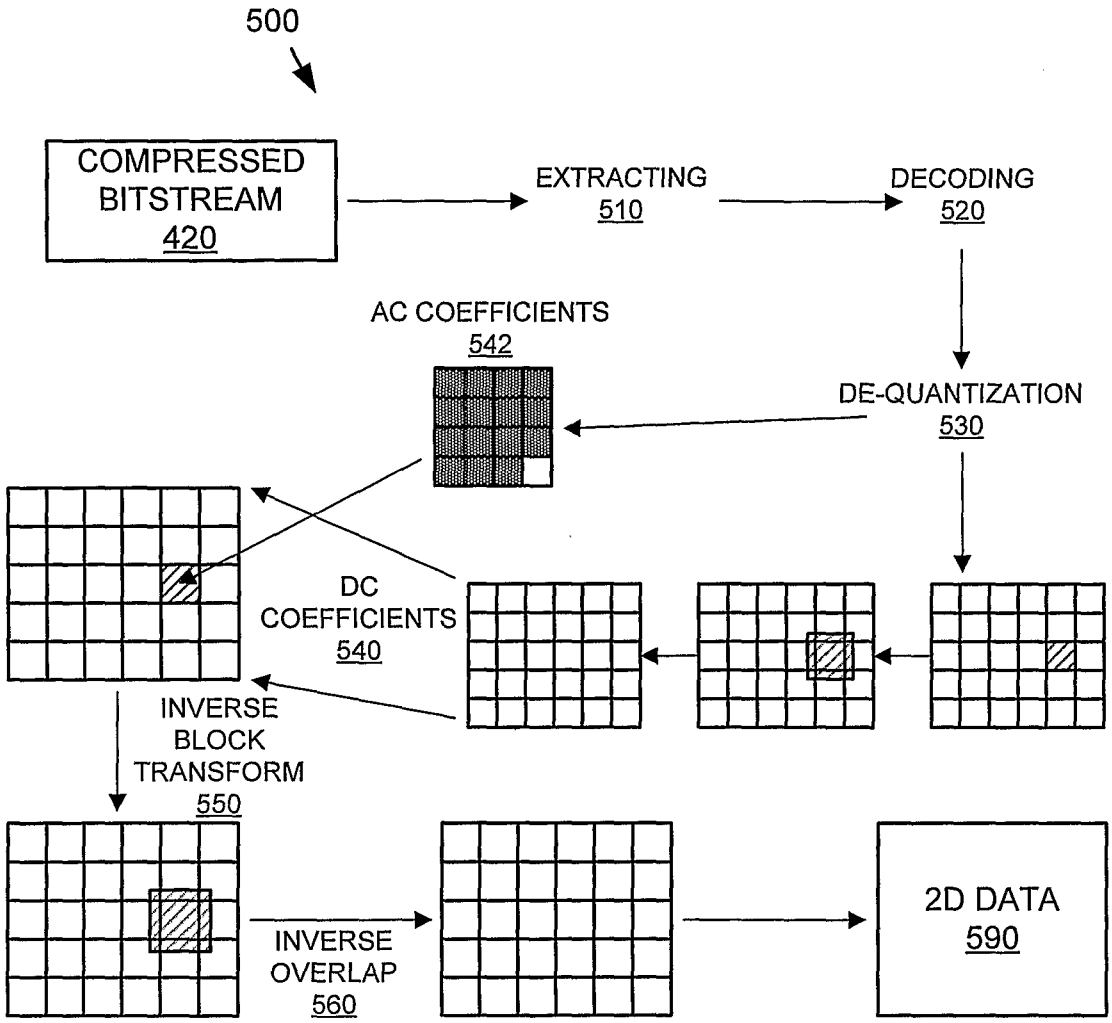

500
COMPRESSED BITSTREAM 420
EXTRACTING 510
DECODING 520
AC COEFFICIENTS 542
DE-QUANTIZATION 530
DC COEFFICIENTS 540
INVERSE BLOCK TRANSFORM 550
INVERSE OVERLAP 560
2D DATA 590

Figure 6

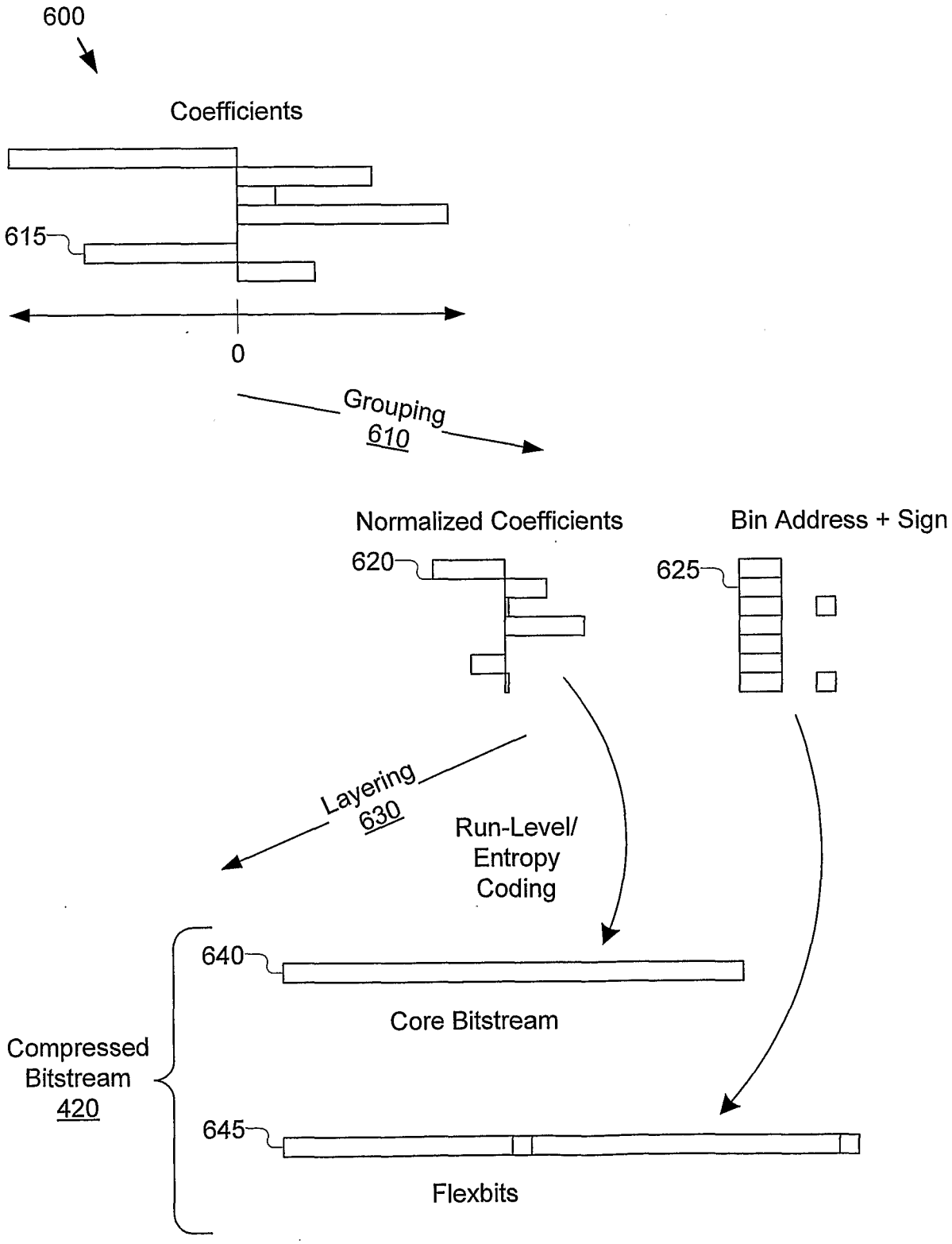

600
Coefficients
615
0
Grouping
610
Normalized Coefficients
620
Bin Address + Sign
625
Layering
630
Run-Level/
Entropy
Coding
640
Core Bitstream
Compressed
Bitstream
420
645
Flexbits

Figure 7

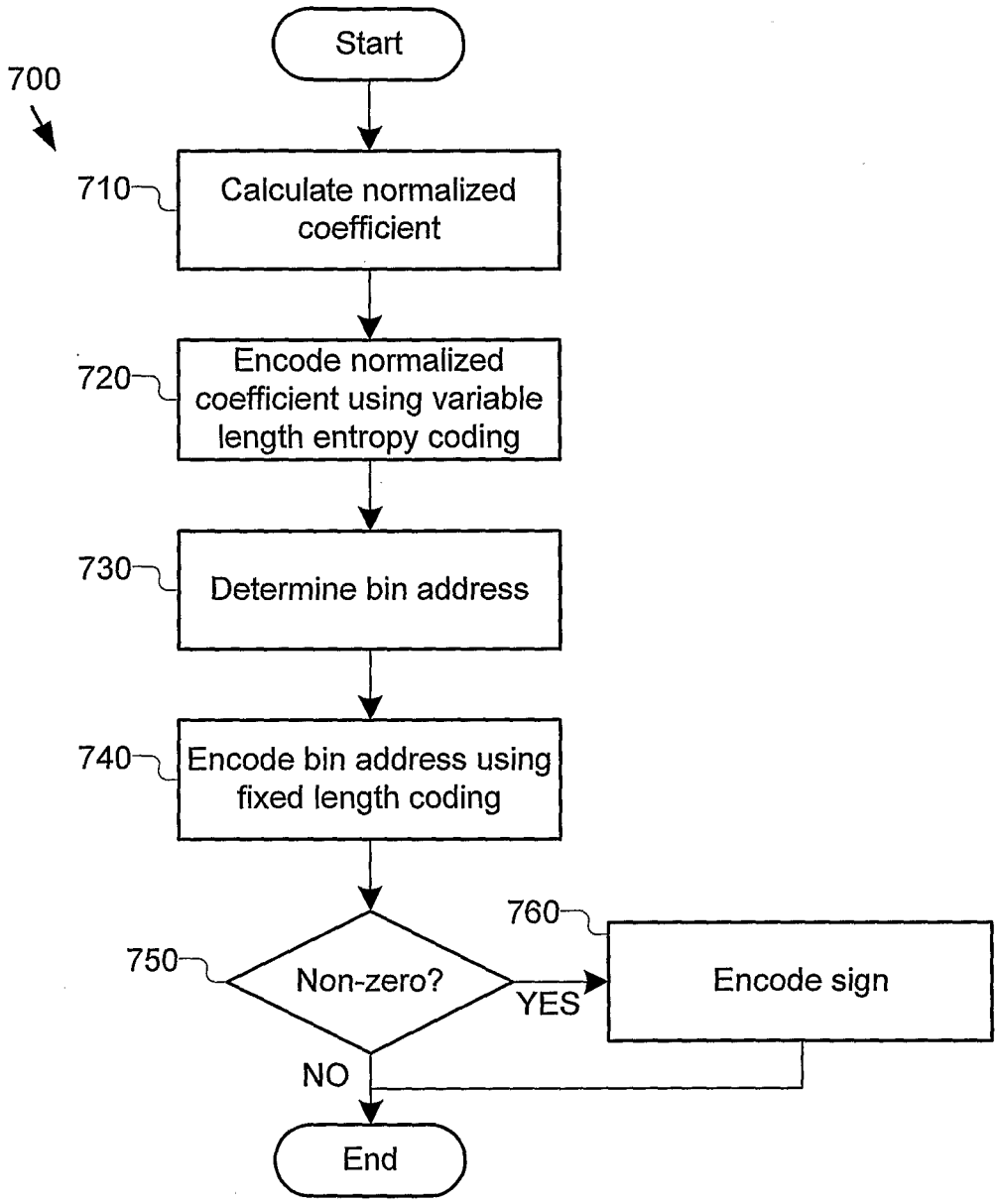

Start
700
710
Calculate normalized coefficient
720
Encode normalized coefficient using variable length entropy coding
730
Determine bin address
740
Encode bin address using fixed length coding
750
Non-zero?
YES
760
Encode sign
NO
End

Figure 8

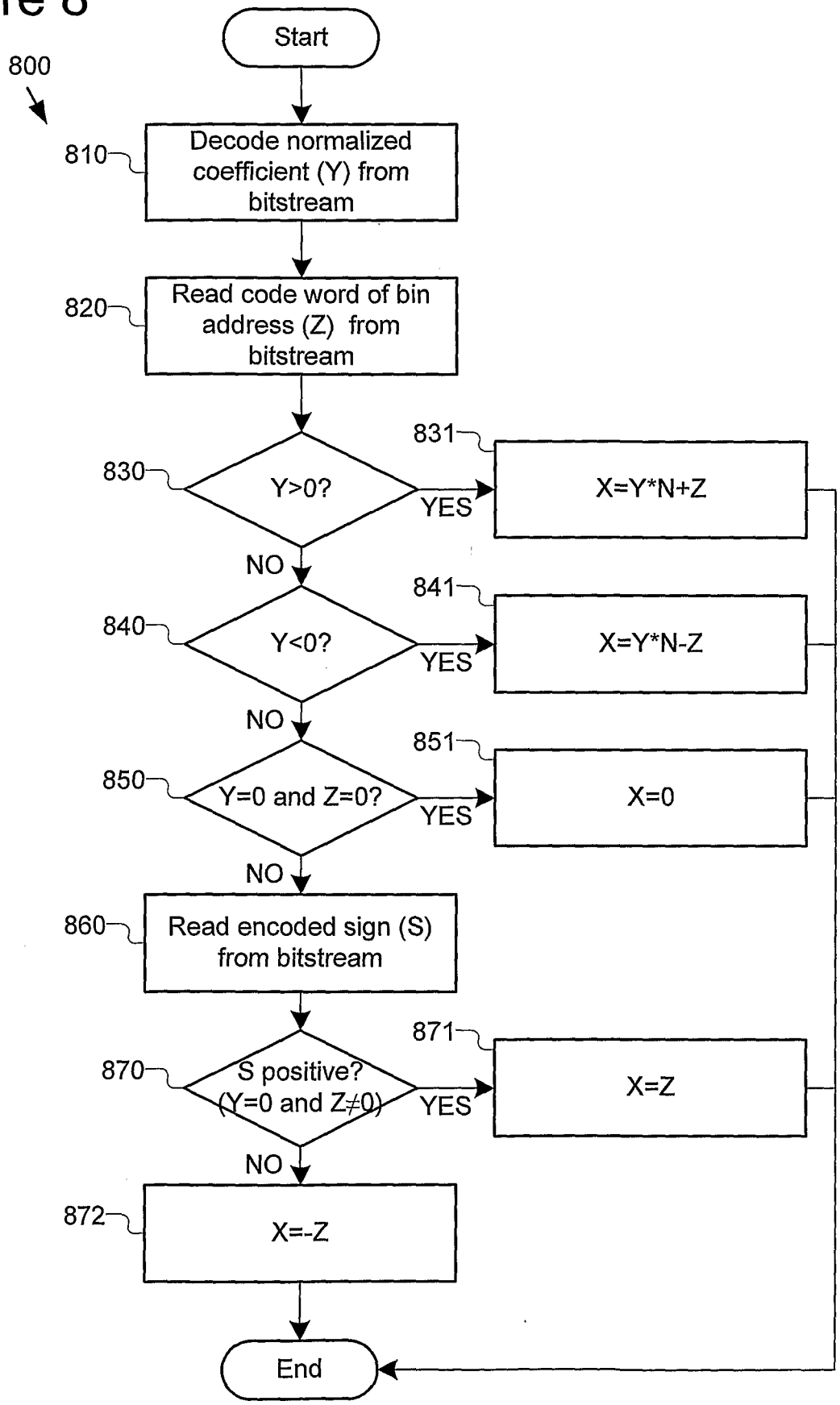

800
Start
810
Decode normalized coefficient (Y) from bitstream
820
Read code word of bin address (Z) from bitstream
830
Y>0?
YES
831
X=Y*N+Z
NO
840
Y<0?
YES
841
X=Y*N-Z
NO
850
Y=0 and Z=0?
YES
851
X=0
NO
860
Read encoded sign (S) from bitstream
870
S positive? (Y=0 and Z≠0)
YES
871
X=Z
NO
872
X=-Z
End

Figure 9

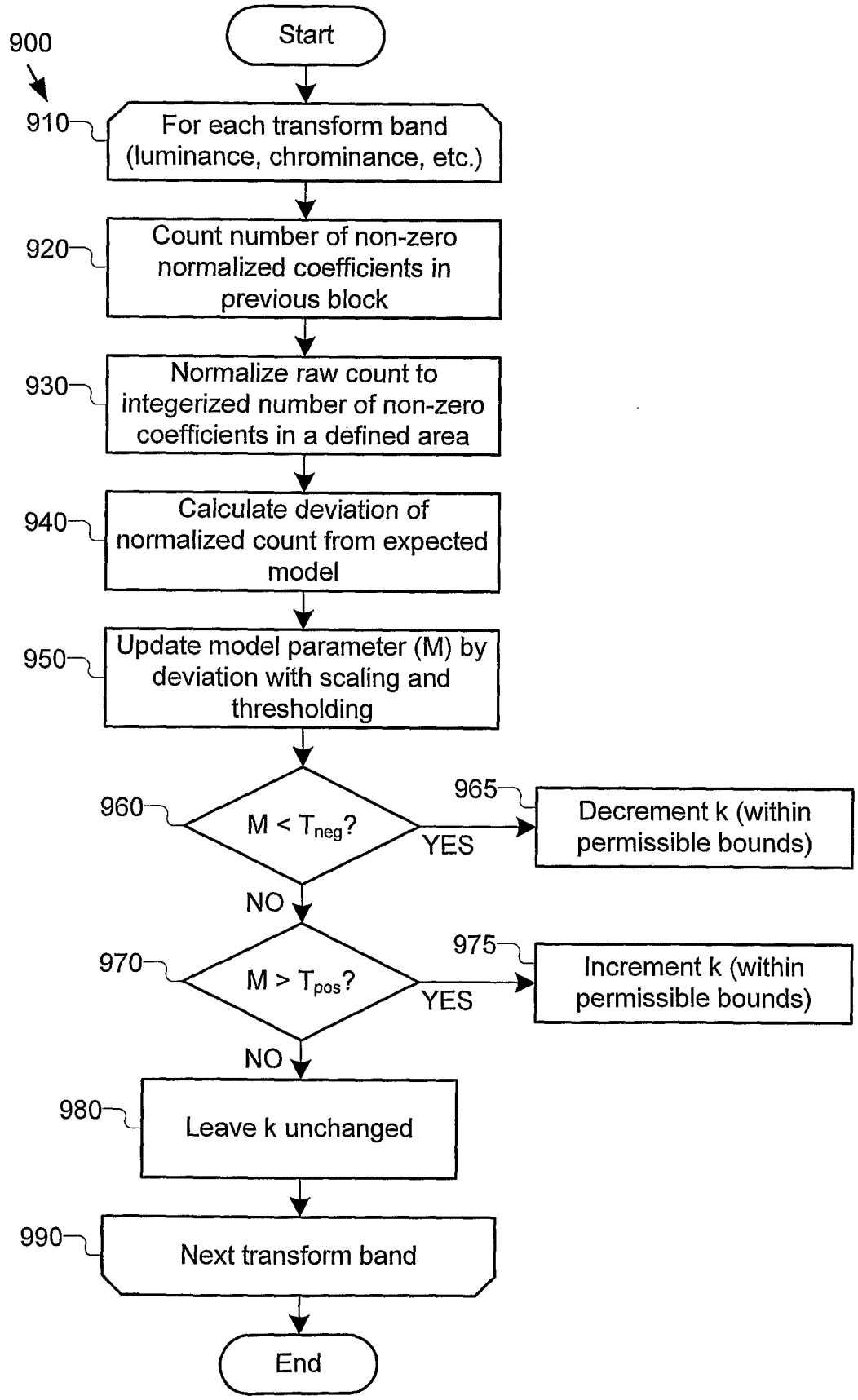

900
Start
910
For each transform band (luminance, chrominance, etc.)
920
Count number of non-zero normalized coefficients in previous block
930
Normalize raw count to integerized number of non-zero coefficients in a defined area
940
Calculate deviation of normalized count from expected model
950
Update model parameter (M) by deviation with scaling and thresholding
960
M < Tneg?
YES
965
Decrement k (within permissible bounds)
NO
970
M > Tpos?
YES
975
Increment k (within permissible bounds)
NO
980
Leave k unchanged
990
Next transform band
End

Figure 10

1000

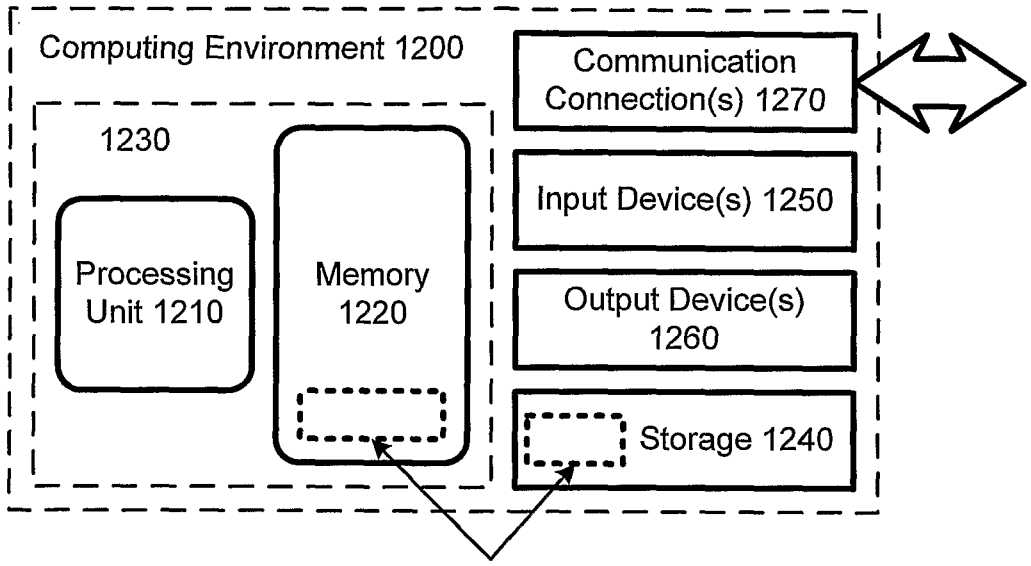
```
Constants:
    MODELWEIGHT = 70
    Weight0[3] = { 240/*DC*/, 12/*LP*/, 1/*HP*/ }
    Weight1[3][16] = {  /* upto 16 channels permitted */
        { 0,240,120,80, 60,48,40,34, 30,27,24,22, 20,18,17,16 },
        { 0,12,6,4,     3,2,2,2,     2,1,1,1,     1,1,1,1 },
        { 0,16,8,5,     4,3,3,2,     2,2,2,1,     1,1,1,1 }
    }
    Weight2[6] = { 120,37,2/*YUV 420*/,
                   120,18,1/*YUV 422*/ }


Model data structure:
    CAdaptiveModel {
        FLCState  /* internal state running from -8 to 7*/
                  /* FLCState is initialized to zero */
        FLCBits   /* k */
                  /* FLCBits is initialized to 0 for highpass,
*/
                  /* 4 for lowpass and 8 for DC */
    }


Function UpdateModelMB (
    Int Band,      /* band 0/1/2 => DC/LP/HP */
    COLORFORMAT cf,/* image color format */
    Int Channels,  /* number of chroma channels <= 16 */
    Int Count[],   /* number of coefficients in luma & chroma
bands */
    CAdaptiveModel Model)
{

/* Normalization */
    iLaplacianMean[0] *= Weight0[Band]
    if (cf == YUV_420) {
        iLaplacianMean[1] *= Weight2[Band]
    }
    else if (cf == YUV_422) {
        iLaplacianMean[1] *= aWeight2[3 + Band]
    }
    else {
        iLaplacianMean[1] *= Weight1[Band][iChannels - 1]
        if (Band == HP)
            iLaplacianMean[1] >>= 4
    }
...
```

Figure 11

1000 (cont.)

```
...

/* Loop over luma & chroma */
    for (j = 0; j < 2; j++) {
        iLM = Count[j]
        iMS = Model.FLCState[j]

/* Calculate deviation from baseline */
        iDelta = (iLM - MODELWEIGHT) >> 2

/* Too few coefficients in the previous MB ? */
        if (iDelta < -8) {
            iDelta += 4  /* hysteresis */
            if (iDelta < -16)
                iDelta = -16
            iMS += iDelta
            if (iMS < -8) {
                if (Model.FLCBits [j] == 0)
                    iMS = -8
                else {
                    iMS = 0
                    Model.FLCBits[j]-- /* bump k down */
                }
            }
        }
/* Too many coefficients in the previous MB ? */
        else if (iDelta > 8) {
            iDelta -= 4  /* hysteresis */
            if (iDelta > 15)
                iDelta = 15
            iMS += iDelta
            if (iMS > 8) {
                iMS = 0
                Model.FLCBits [j]++   /* bump k up */
                if (Model.FLCBits [j] > 16)
                    Model.FLCBits [j] = 16
            }
        }

        Model.FLCState[j] = iMS
    }
}
```

Figure 12


Computing Environment 1200
1230
Processing Unit 1210
Memory 1220
Communication Connection(s) 1270
Input Device(s) 1250
Output Device(s) 1260
Storage 1240
Software 1280 Implementing Adaptive Wide-range Coefficient Coding/Decoding