ISO/IEC 15444-17:2023-06 (E)

Information technology - JPEG 2000 image coding system - Part 17: Extensions for coding of discontinuous media

Contents

Page

1 Scope 1
2 Normative references 1
3 Terms and definitions 1
4 Abbreviations 2
5 5 Conventions 3
6 Conformance 3
6.1 Codestream conformance 3
6.2 Decoder 3
7 Breakpoint component structure 4
7.1 Breakpoint components and the reference grid 4
7.2 Division of breakpoint resolutions into cells, arcs and the CL band 4
7.3 Division of breakpoint resolutions into precincts and code-blocks 5
7.4 Root and non-root arc breakpoint associations 7
7.5 Breakpoint values and vertices 8
8 Breakpoint-dependent spatial wavelet transformation 10
8.1 Overview 10
8.2 TriBPT-dependent irreversible transforms 11
8.2.1 Introduction 11
8.2.2 Arcs with tb = 0 12
8.2.3 Arcs with tb = 3 13
8.2.4 Arcs with tb = 2 13
8.2.5 Arcs with tb = 1 17
8.3 TriBPT-dependent reversible transforms 21
8.3.1 Arcs with tb = 0 22
8.3.2 Arcs with tb = 3 22
8.3.3 Arcs with tb = 2 22
8.3.4 Arcs with tb = 1 22
8.4 QuadBPT-dependent transforms 23
8.4.1 Introduction 23
8.4.2 BD_2D_SR phase 1 24
8.4.3 BD_2D_SR phase 2 26
9 Decoding of breakpoint code-blocks 28
9.1 Embedded bit-plane decoding of breakpoints 28
9.2 Inter-band coding mode (BPT_INTER) 29
9.3 Cell-based scanning patterns and CBAP-based block flipping 30
9.4 QuadBPT decoding procedures 32
9.4.1 MQ Coder contexts and initial states for QuadBPT decoding 32
9.4.2 Derivation of context labels for QuadBPT CL band significance coding passes 33
9.4.3 Non-root significance decoding for QuadBPT CL code-blocks 35
9.4.4 Root significance decoding for QuadBPT CL code-blocks 36
9.4.5 Root significance decoding for QuadBPT LL code-blocks 37
9.4.6 Position refinement decoding for QuadBPT code-blocks 37
9.5 TriBPT decoding procedures 38
9.5.1 MQ Coder contexts and initial states for TriBPT decoding 38
9.5.2 Derivation of context labels for TriBPT CL band significance coding passes 39
9.5.3 Non-root significance decoding for TriBPT CL code-blocks 42
9.5.4 Root significance decoding for TriBPT CL code-blocks 43
9.5.5 Root significance decoding for TriBPT LL code-blocks 44
9.5.6 Position refinement decoding for TriBPT code-blocks 45
9.6 Quality layers and packets for breakpoint components 45

10 Reconstruction of breakpoint components 47
10.1 Overview 47
10.2 Direct induction step 47
10.2.1 Introduction 47
10.2.2 Extrapolation Qualifier for TriBPT direct induction 48
10.3 QuadBPT spatial induction 49
10.3.1 Introduction 49
10.4 TriBPT spatial induction 51
10.4.1 TriBPT induction modes and corner notation 51
10.4.2 4-arc mode 51
10.4.3 3-arc mode 52
10.4.4 Extrapolation qualifier for TriBPT spatial induction 53
10.4.5 Rounding policies for TriBPT spatial induction 56
10.4.6 Image boundary handling for TriBPT 56

Annex A (This annex forms an integral part of this Recommendation | International Standard.) – Codestream syntax 58
A.1 General 58
A.2 SIZ marker segment 58
A.3 CAP marker segment 58
A.4 Hierarchical data type (HDT) marker segment 59
A.5 COD and COC marker segments 59

Annex B (This annex does not form an integral part of this Recommendation | International Standard) – Hierarchical breakpoint encapsulation and raw file format 62
B.1 General 62
B.2 Raster organization of breakpoints 62
B.3 16- and 32-bit packed breakpoint values 63
B.4 A raw file format for breakpoints 64

Bibliography 67

List of Tables

Page

Table 1 – Constraints on PPx and PPy precinct size parameters breakpoint components 6

Table 2 – MQ coder contexts for QuadBPT CL code-block decoding passes. State index values correspond to the Qe values and probability estimation transitions tabulated in Rec. ITU-T T.800 | ISO/IEC 15444-1 33

Table 3 – Context labels from Table 2 that are used for QuadBPT LL code-block decoding passes 33

Table 4 – MQ coder contexts for TriBPT CL code-block decoding passes. State index values correspond to the Qe values and probability estimation transitions tabulated in Rec. ITU-T T.800 | ISO/IEC 15444-1 38

Table 5 – Context labels from Table 4 that are used for TriBPT LL code-block decoding passes 38

Table A.1 – Extensions and constraints to marker segments specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 and new marker segments specified in this Recommendation | International Standard 58

Table A.2 – Ccap[17] Syntax and Semantics 58

Table A.3 – Hierarchical data type parameter values 59

Table A.4 – Allowed values for Ihdt[i] 59

Table A.5 – Scod and Scoc semantics for bit 5 60

Table A.6 – SXcod bit fields specified by this Recommendation | International Standard 60

Table A.7 – Allowed values for the SEcod field 61

Table A.8 – Transformation for the SPcod and SPcoc parameters when $Dc \neq 0$ 61

List of Figures

Page

Figure 1 – Arc arrangements within a single 2-span. A $\mathbf{2 \times 2}$ cell is shown here with solid grid-points. See Figure 2 for other geometric relationships between cells and 2-spans 5

Figure 2 – Cell-span geometry for each of the four possible code-block anchor points 5

Figure 3 – Root arc (dotted lines) and non-root arcs (solid lines) 8

Figure 4 – Possible break locations (tick-points) at the finest resolution, with $\boldsymbol{FB = 2}$, shown for the QuadBPT case. Also shows the mapping of vertices with precision $\boldsymbol{Pb}$ to tick-points 9

Figure 5 – Stages of the breakpoint dependent inverse discrete wavelet transform 10

Figure 6 – Interleaving of sub-band coefficients to a triangular grid for a given resolution *r* 12

Figure 7 – Update step examples for TriBPT-LR arrangement 12

Figure 8 – Predict step examples for a horizontal arc 12

Figure 9 – Example of direct induction of a breakpoint 14

Figure 10 – Example illustrating spatially induced breaks 14

Figure 11 – Example of a spatially induced breakpoint on a non-root arc for the TriBPT-LR arrangement 15

Figure 12 – Examples showing base-arc $\boldsymbol{bsa}$ and gradients $\boldsymbol{Gbsa}$ and $\boldsymbol{Gwdg}$ along with corresponding binary flags ORN and $\boldsymbol{DRCTNX_FLAG\ induced}$ for root arcs 16

Figure 13 – Example showing base-arc $\boldsymbol{bsa}$ and gradients $\boldsymbol{Gbsa}$ and $\boldsymbol{Gwdg}$ along with corresponding binary flags ORN and $\boldsymbol{DRCTNX_FLAG\ induced}$ for diagonal non-root arc containing a spatially induced break 16

Figure 14 – Examples of arcs $\boldsymbol{prl}$ parallel in orientation to root arcs centred at $\boldsymbol{p}$ and containing a spatially induced break. Gradient calculated across arc $\boldsymbol{prl}$ is denoted as $\boldsymbol{Gprl}$ 19

Figure 15 – Example of arc $\boldsymbol{prl}$ parallel to the non-root arc centred and at $\boldsymbol{p}$ and containing a spatially induced break. Corresponding gradient across arc $\boldsymbol{prl}$ is shown as $\boldsymbol{Gprl}$ 19

Figure 16 – Algorithm for determining the gradient $\boldsymbol{Gb}$ based on validity flags VALID_WDG and VALID_PRL and corresponding attributes 20

Figure 17 – Parent 4-span arrangement at resolution $\boldsymbol{rrt = r - 1}$ for arc $\boldsymbol{a}$ at resolution $\boldsymbol{r}$ and centred at $\boldsymbol{p}$ 21

Figure 18 – Algorithm for determining the gradient $\boldsymbol{Gb}$ for the reversible transform, based on validity flags VALID_WDG and VALID_PRL and corresponding attributes 23

Figure 19 – Interleaving of sub-band coefficients to a quadrilateral grid for a given resolution *r* 23

Figure 20 – Phase 1 update step for (even, even) location $\boldsymbol{2n, 2m}$ 24

Figure 21 – Phase 1 update step for (even, odd) location $\boldsymbol{2n, 2m + 1}$ and (odd, even) location $\boldsymbol{2n + 1, 2m}$ 25

Figure 22 – Phase 1 predict step for (odd, odd) location $\boldsymbol{2n + 1, 2m + 1}$ with source locations of $\boldsymbol{Ic, r} *$shown 26

Figure 23 – Phase 2 update step for (even, even) location $\boldsymbol{2n, 2m}$ 27

Figure 24 – Phase 2 predict step examples 28

Figure 25 – Canonical scanning pattern used for the significance coding passes of CL band code-blocks in a QuadBPT component. In this case, missing cells within a $\mathbf{2 \times 2}$ group are not included in the scan 30

Figure 26 – Canonical scanning pattern used for the significance coding passes of CL band code-blocks in TriLR components. Arcs belonging to cells that lie beyond the code-block boundaries are included in the significance coding passes 31

Figure 27 – Canonical scanning pattern used for the significance coding passes of CL band code-blocks in TriRL components. Arcs belonging to cells that lie beyond the code-block boundaries are included in the significance coding passes 31

Figure 28 – Neighbouring cells and arcs involved in forming coding contexts for QuadBPT significance decoding in CL band code-blocks 34

Figure 29 – Neighbouring cells and arcs involved in forming coding contexts for TriLR significance decoding in CL band code-blocks 39

Figure 30 – Neighbouring cells and arcs involved in forming coding contexts for TriRL significance decoding in CL band code-blocks 40

Figure 31 – Breakpoint synthesis stages 47

Figure 32 – Relationship between non-root and parent arcs during direct induction in synthesis stage $\boldsymbol{r}$, shown for the case of horizontal arcs 48

Figure 33 – Rounding policies used in QuadBPT spatial induction, showing different induction examples in three of the four 2-spans within a 4-span 50

Figure 34 – Arc and corner labels for the TriBPT-LR arrangement 51

Figure 35 – 4-arc mode spatial induction examples for the TriBPT-LR arrangement 52

Figure 36 – 3-arc mode spatial induction examples for the TriBPT-LR arrangement 53

Figure 37 – Example of intersection points, arcs and grid-lines required for determining the extrapolation qualifier for a spatially induced break 55

Figure 38 – Examples of rounding policy for 3-arc mode spatial induction when the inducing line intersects the root arc X at its centre. The direction of rounding is highlighted by the black arrows 56

Figure A.1 – Hierarchical data type syntax 59

Figure A.2 – Coding style default syntax 60

Figure A.3 – Coding style component syntax 60

Figure B.1 – Bit fields within 16- and 32-bit packed breakpoint representations. The most significant bit of the 16- or 32-bit integer, as appropriate, is shown at the top of the figure, while the least significant bit appears at the bottom 64