FINAL DRAFT

INTERNATIONAL STANDARD

ISO/IEC FDIS 15444-17

ISO/IEC JTC **1**/SC **29**

Secretariat: **JISC**

Voting begins on:
2022-07-29

Voting terminates on:
2022-09-23

Information technology — JPEG 2000 image coding system —

Part 17:
Extensions for coding of discontinuous media

Reference number
ISO/IEC FDIS 15444-17:2022(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by ITU-T (as ITU-T - T.816) and drafted in accordance with its editorial rules, in collaboration with Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 15444 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

JPEG 2000 Parts 1 and 2 provide a suite of scalable coding technologies that are particularly suitable for photographic media, but less effective at coding media with hard discontinuities. An important example of such media is depth imagery, where each image sample is related to the length of the 3D line segment between the corresponding scene point and the camera. Depth imagery includes stereo disparity maps, where sample values are reciprocally related to depth. Another example of media with strong discontinuities is optical flow data, where each sample location is a two-dimensional vector. In these examples, discontinuities arise naturally at the boundaries of scene objects. Moreover, where this happens, intermediate values that might be obtained by bandlimited image resampling or interpolation operations have no physical meaning – i.e., they do not correspond to the depth or flow vector of any object in the original scene. The Discrete Wavelet Transform (DWT) employed in JPEG 2000 is not well suited to the coding of such media, both from the perspective of coding efficiency and considering the nature of distortions that result when the wavelet subband samples are quantized.

To address these challenges, this Recommendation | International Standard introduces alternate "breakpoint-dependent" spatial wavelet transforms that dependent on an auxiliary image component, known as a "breakpoint component." This Recommendation | International Standard also introduces scalable coding technologies for breakpoint components. Any non-initial component or components within the codestream can be designated as breakpoint components, allowing them to be used as the source of breakpoints for other components, or tiles thereof, which specify the use of breakpoint-dependent wavelet transforms.

This Recommendation | International Standard specifies two different types of breakpoint components, designated as "QuadBPT" and "TriBPT" components, with associated decoding and synthesis tools. Associated with the type of breakpoint component is a corresponding breakpoint-dependent wavelet transform, with its synthesis tools. The reconstruction procedures described in this document produce individual sample values. In the TriBPT case, it is possible instead to directly reconstruct a deformable triangular mesh, whose complexity is related to the number of non-zero wavelet coefficients and the number of decoded breaks, which are identified here as "vertices." In each case, breakpoints introduce tears in the mesh. This feature can be valuable in computer graphics applications, where the mesh elements provide a more convenient description of the data than individual samples.

The normative material of this Recommendation | International Standard is contained within the main body together with Annex A. Additionally, Annex B describes ways of encapsulating breakpoint data within a linear file structure, that can be used as a source for encoding and a target for decoding procedures.

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

1 Scope

This Recommendation | International Standard defines QuadBPT and TriBPT image components, collectively known as "breakpoint components," and specifies decoding and reconstruction procedures for recovering breakpoint component sample values from the codestream. This Recommendation | International Standard also specifies "breakpoint-dependent" spatial wavelet transforms that can be used in place of the transforms specified in Recommendation ITU-T T.800 | ISO/IEC 15444-1, for selected image components or tile-components. Extensions to the codestream syntax of Rec. ITU-T T.800 | ISO/IEC 15444-1 are specified to enable the identification of breakpoint components, of components that can use a breakpoint-dependent spatial wavelet transform, and the association of breakpoint components with such breakpoint-dependent wavelet transforms.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Recommendation ITU-T T.800 | ISO/IEC 15444-1, *Information technology — JPEG 2000 image coding system — Part 1: Core coding system*.

Recommendation ITU-T T.801 | ISO/IEC 15444-2, *Information technology — JPEG 2000 image coding system: Extensions*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Rec. ITU-T T.800 | ISO/IEC 15444-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
2-span
square configuration of width 2 and height 2, with 9 grid-points, such that the four corner grid-points all have coordinates that are divisible by 2

3.2
4-span
2×2 configuration of 2-spans (3.1), involving 25 grid-points, such that the four corner grid-points all have coordinates that are divisible by 4

3.3
arc
line segment connecting grid-points with even valued coordinates at any resolution of a breakpoint tile-component

3.4
ambivalent break
induced break (3.5) that has insufficient precision to determine whether the break occurs in the first or second half of the arc

3.5
break
explicitly decoded or inferred location within an arc (3.3) that indicates a boundary within an image component

Note 1 to entry: Breaks modify the behaviour of breakpoint-dependent transforms on that other image component.

Note 2 to entry: An arc has at most one break.

3.6
breakpoint
data structure consisting of break type, location and precision information for an arc (3.3)

Note to entry: Type-0 breakpoints have no break at all.

3.7
breakpoint tile-component
all the breakpoints of a given tile, within a breakpoint component (3.8)

3.8
breakpoint component
JPEG 2000 codestream image component that represents arc breakpoint (3.6) information

3.9
cell
2 × 2 configuration of grid-points within a resolution of a breakpoint tile-component (3.7)

Note to entry: Cells belong to a well-defined partition that is anchored at the global code-block anchor point.

3.10
CL Band
the sole subband associated with each resolution of a breakpoint tile-component (3.7) other than the lowest resolution

3.11
code-block anchor point
origin of the coding partitions, which is one of the locations (0,0), (0,1), (1,0) or (1,1)

[SOURCE: Rec. ITU-T T.801 | ISO/IEC 15444-2]

3.12
directly induced break
break (3.5) on an arc that is inferred from a break on a parent arc

3.13
extrapolation qualifier
2-bit quantity e_b which controls the way gradients are obtained for extrapolation within a TriBPT-dependent transformation

3.14
indefinite break
induced break (3.15) on an arc that is derived from one or more ambivalent breaks (3.4), such that there is insufficient precision to determine whether any break at all exists on the arc

3.15
induced break
break (3.5) on an arc that is inferred from breaks on other arcs

3.16
induction block
condition on an arc (3.3) that is explicitly recovered from the decoding of breakpoint code-blocks, indicating that no break (3.5) shall be induced on that arc

3.17
non-root arc
arc (3.3) that is not a root arc

3.18
parent arc
arc (3.3) at depth $d + 1$ in the breakpoint decomposition that contains an arc at depth d

Note to entry: At most two arcs at depth d can have the same parent at depth $d + 1$.

3.19
pass-complete
code-block within a breakpoint component for which one or more coding passes are found within the codestream packets and the last such coding pass is identified as completing the code-block's representation via the packet header signalling mechanisms

3.20
QuadBPT
breakpoint (3.6) arrangement involving only horizontal and vertical arcs (3.3)

3.21
root arc
arc (3.3) that is not contained within any arc projected from the next lower resolution of a breakpoint tile-component (3.7), regardless of whether that next lower resolution actually exists within the tile-component's resolution hierarchy

3.22
spatially induced break
induced break (3.15) that is inferred from breaks on non-parent arcs

3.23
tick-point
possible break (3.5) location along an arc

3.24
TriBPT
breakpoint (3.6) arrangement involving horizontal, vertical and diagonal arcs (3.3)

3.25
TriBPT-LR
TriBPT (3.24) breakpoint arrangement involving diagonal arcs that run from the top-left to the bottom-right of a 2-span within any resolution of a breakpoint tile-component (3.7)

3.26
TriBPT-RL
TriBPT (3.24) breakpoint arrangement involving diagonal arcs that run from the top-right to the bottom-left of a 2-span within any resolution of a breakpoint tile-component (3.7)

3.27
vertex
explicitly coded break (3.5) location

3.28
zero-complete
code-block within a breakpoint component (3.8) that makes no contribution to any codestream packet and is identified as complete by the first packet header of its precinct

4 Symbols and abbreviated terms

- **CBAP** $(\boldsymbol{z_x}, \boldsymbol{z_y})$ – code-block anchor point
- **MAX_WDG** – maximum search distance for the gradient extrapolation algorithm used during the TriBPT-dependent prediction step associated with a spatially induced arc.
- **BPT_INTER** – binary flag that is 1 if code-blocks of a breakpoint component use the inter-band coding mode and 0 if the code-blocks of a breakpoint component are coded without reference to any other code-block data.

5 Conformance

5.1 Part-17 codestream

A Part-17 Codestream shall conform to Annex A.

5.2 Part-17 decoder

A Part-17 Decoder shall process a Part-17 codestream as specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 together with any additional signalled capability, with the exception of breakpoint components and tile-components that identify the use of a breakpoint-dependent spatial wavelet transform, in which case the following shall apply:

- breakpoint components shall have the structure specified in Clause 6;
- tile-components that use a breakpoint-dependent spatial wavelet transform shall be processed in accordance with Clause 7; and
- breakpoint components shall be reconstructed from breakpoint code-blocks in accordance with Clause 0, where breakpoint code-blocks are decoded in accordance with Clause 8.

This Recommendation | International Standard is compatible with the coding technologies specified in both Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2. However:

1. Breakpoint components, as specified herein, are not compatible with the multiple component transformations specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2, nor are they compatible with the non-linear transformations specified in Rec. ITU-T T.801 | ISO/IEC 15444-2; and

2. Breakpoint code-blocks, as specified herein, are not compatible with the region of interest coding and extraction techniques specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2.

6 Breakpoint component structure

6.1 Breakpoint components and the reference grid

Breakpoint components have a "hierarchical data type," meaning that they are described at multiple resolutions, in a dyadic hierarchy. Breakpoint components are identified via the HDT marker segment defined in A.4. In particular, component c is a breakpoint component if Ihdt^{c} lies in the range 1 to 3.

Breakpoint component c is defined with respect to the same reference grid as other JPEG 2000 components, in terms of the global image dimensions $(\mathrm{Xsiz}, \mathrm{Ysiz})$, image offset $(\mathrm{XOsiz}, \mathrm{YOsiz})$ and sampling factors $(X\mathrm{Rsiz}^{c}, Y\mathrm{Rsiz}^{c})$ that are recorded in the SIZ marker segment. The *grid-points* of component c consist of all integer coordinates in the rectangle with upper left hand corner at (x_o, y_o) and lower right hand corner at $(x_l - 1, y_l - 1)$, where

$$x_o = \left\lceil \frac{\mathrm{XOsiz}}{X\mathrm{Rsiz}^{c}} \right\rceil,\ x_l = \left\lceil \frac{\mathrm{Xsiz}}{X\mathrm{Rsiz}^{c}} \right\rceil,\ y_o = \left\lceil \frac{\mathrm{YOsiz}}{Y\mathrm{Rsiz}^{c}} \right\rceil,\ y_l = \left\lceil \frac{\mathrm{Ysiz}}{Y\mathrm{Rsiz}^{c}} \right\rceil. \quad (1)$$

These formulae are identical to the equations for x_o, x_l, y_o and y_l in Rec. ITU-T T.800 | ISO/IEC 15444-1, but note that the term *grid-points* is used here for the locations of a breakpoint component, rather than *samples*.

As with other components, breakpoint components are partitioned into tile-components, following the partitioning of the reference grid into tiles, so that the grid-points of a breakpoint tile-component, having component index c belong to the rectangle with upper left corner at (tcx_o, tcy_o) and lower right hand corner at $(tcx_l - 1, tcy_l - 1)$, where

$$tcx_o = \left\lceil \frac{tx_o}{X\mathrm{Rsiz}^{c}} \right\rceil,\ tcx_l = \left\lceil \frac{tx_l}{X\mathrm{Rsiz}^{c}} \right\rceil,\ tcy_o = \left\lceil \frac{ty_o}{Y\mathrm{Rsiz}^{c}} \right\rceil,\ tcy_l = \left\lceil \frac{ty_l}{Y\mathrm{Rsiz}^{c}} \right\rceil, \quad (2)$$

where the tile in question occies the rectangle with top left corner at (tx_o, ty_o) and lower right corner at $(tx_l - 1, ty_l - 1)$ on the reference grid. These formulae are identical to the equantions for tcx_o, tcx_l, tcy_o and tcy_l in Rec. ITU-T T.800 | ISO/IEC 15444-1, except that the breakpoint tile-component's locations are identified as *grid-points* rather than *samples*.

Like other JPEG 2000 components, each breakpoint tile-component has an associated number of *decomposition levels* N_L, that is identified via the applicable COD or COC marker segment, imputing the component with $N_L + 1$ distinct resolution levels, denoted $r = 0,1, \ldots, N_L$; these are the resolutions of the hierarchical breakpoint representation. Level $r = N_L$ is the full resolution of the tile-component, while $r = 0$ is the lowest resolution, also known as the tile-component's LL band. The *grid-points* associated with the resolution r of a breakpoint tile-component correspond to the integer-valued coordinates within the rectangle with upper left corner at (trx_o, try_o) and lower right hand corner at $(trx_l - 1, try_l - 1)$, where

$$trx_o = \left\lceil \frac{tcx_o}{2^{N_L - r}} \right\rceil,\ trx_l = \left\lceil \frac{tcx_l}{2^{N_L - r}} \right\rceil,\ try_o = \left\lceil \frac{tcy_o}{2^{N_L - r}} \right\rceil,\ try_l = \left\lceil \frac{tcy_l}{2^{N_L - r}} \right\rceil. \quad (3)$$

Each resolution r has a corresponding depth d within the hierarchical representation. The highest resolution $r = N_L$ corresponds to depth $d = 1$, while the resolution $r = 1$ corresponds to depth $d = N_L$.

Unlike other JPEG 2000 components, for which the LL band at resolution $r = 0$ has the same depth $d = N_L$ as resolution 1, the LL band of a breakpoint component has depth $d = N_L + 1$.

6.2 Division of breakpoint resolutions into cells, arcs and the CL band

Breakpoint components, their tile-components and their resolutions, do not have samples, but their grid-points may correspond to the sample locations of other components. Instead of samples, the coded information associated with a breakpoint component belongs to *arcs*.

There are two fundamental arrangements for these arcs, known as QuadBPT and TriBPT, and two mirror-image variants for the TriBPT arrangement, known as TriBPT-LR and TriBPT-RL. All three arrangements are defined in terms of 2×2 cells within each resolution of a breakpoint tile-component. To understand these arrangements, it is helpful to start by introducing the concept of a *2-span*, which consists of 9 grid-points. As shown in Figure 1, each arc runs between grid-points on the boundary of its 2-span. The central grid-point of a 2-span always has odd-valued coordinates in the resolution to which it belongs, while the corners of each 2-span have even-valued coordinates. Within any given resolution arcs have length 2, if we measure the length of diagonal arcs in the vertical direction only. This is equivalent to a length of 2^d at the component's full resolution.

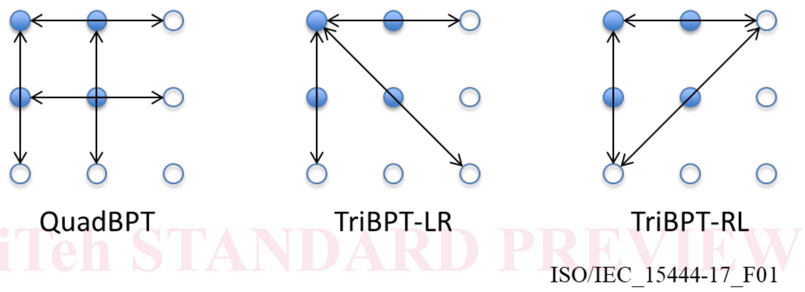

Figure 1: Arc arrangements within a single 2-span. A 2×2 cell is shown here with solid grid-points. See Figure 2 for other geometric relationships between cells and 2-spans.

Each 2-span is associated with a cell that contains only 4 unique grid-points. The grid-points of each resolution are partitioned into cells, such that the upper left corner of each cell has coordinates of the form $(2c_x + z_x, 2c_y + z_y)$ and (z_x, z_y) corresponds to the code-block anchor point (CBAP) identified by bits 3 and 4 of the Scod parameter in the extended COD syntax specified in Rec. ITU-T T.801 | ISO/IEC 15444-2. The four possible combinations of CBAP coordinates, (z_x, z_y) = (0,0), (0,1), (1,0) and (1,1), lead to four different cell partitions, that enable geometric manipulation (flipping and rotation) in the compressed domain. Figure 2 illustrates the four different relationships between cells and their 2-spans that result from the different anchor-points.

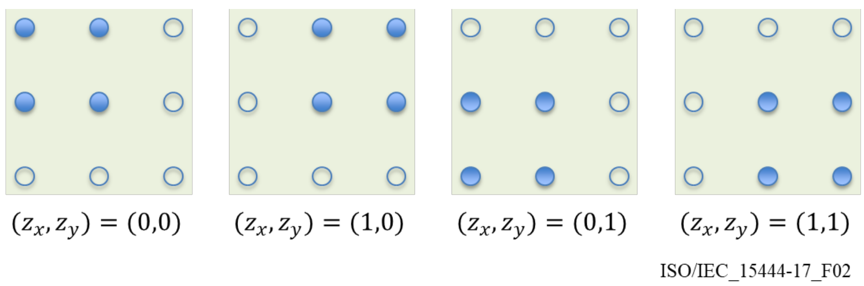

Figure 2: Cell-span geometry for each of the four possible code-block anchor points.

NOTE 1 - The code-block anchor point is also the anchor point for the precinct partitions in JPEG 2000. This means that the cell partition always aligns with the precinct partition for the same resolution of a tile-component.

The cells associated with a resolution of a breakpoint tile-component have coordinates (c_x, c_y), covering the rectangle with upper left corner at (clx_o, cly_o) and lower right hand corner at $(clx_l - 1, cly_l - 1)$, where

$$clx_o = \left\lceil \frac{trx_o - z_x}{2} \right\rceil, \; clx_l = \left\lceil \frac{trx_l - z_x}{2} \right\rceil, \; cly_o = \left\lceil \frac{try_o - z_y}{2} \right\rceil, \; cly_l = \left\lceil \frac{try_l - z_y}{2} \right\rceil. \qquad (4)$$

That is, the resolution's cells consist of all 2×2 elements from the partition that intersect with the resolution's region of support, regardless of whether the intersection involves 1, 2 or all 4 grid-points.

Ignoring the LL band (resolution 0) for the moment, each other resolution $r = 1, \ldots, N_L$ of a breakpoint tile-component is assigned a single "detail band," similar to the LH, HL and HH subbands described in Rec. ITU-T T.800 | ISO/IEC 15444-1. This single detail band is denoted the CL band. Each element of a CL band is a 2×2 cell (or a piece thereof, if the tile-component boundaries disect the cell). The hierarchical representation of the breakpoint tile-component then consists of one LL band and N_L CL bands. Each CL band is $clx_l - clx_o$ cells wide by $cly_l - cly_o$ cells high, where these quantities are derived from the corresponding resolution's dimensions using Formula (4). The set of cells that constitute a CL band determine the properties of its code-blocks and precincts, as explained next.

The dimensions of the LL band are expressed in grid-points rather than cells, and are identical to the dimensions of resolution 0, as derived from Formula (3) with $r = 0$. The LL band of a breakpoint tile-component is used to record breakpoints that would have been found in a CL band at depth $N_L + 1$, except that they are encoded with extra information to reveal whether or not the breakpoints should be considered to have been induced from even lower (uncoded) levels in the hierarchy. This is explained further in 8.

NOTE 2 - Unlike regular image components, the LL bands of breakpoint tile-components are commonly devoid of any significant information – i.e., they need signal no breaks at all – but this might not always be the case.

6.3 Division of breakpoint resolutions into precincts and code-blocks

Like all JPEG 2000 components, each resolution of a breakpoint component is partitioned into precincts, and the associated bands are partitioned into code-blocks, such that each code-block belongs to exactly one precinct. The number of precincts which span a breakpoint tile-component at resolution r is given by

$$precinctswide = \begin{cases} \left\lceil \frac{trx_l - z_x}{2^{\mathrm{PPx}}} \right\rceil - \left\lfloor \frac{trx_o - z_x}{2^{\mathrm{PPx}}} \right\rfloor & trx_l > trx_o \\ 0 & trx_l = trx_o \end{cases},$$

$$precinctshigh = \begin{cases} \left\lceil \frac{try_l - z_y}{2^{\mathrm{PPy}}} \right\rceil - \left\lfloor \frac{try_o - z_y}{2^{\mathrm{PPy}}} \right\rfloor & try_l > try_o \\ 0 & try_l = try_o \end{cases},$$

where PPx and PPy are signalled in COD and COC marker segments. These formulae for *precinctswide* and *precinctshigh* are the same as those found in Rec. ITU-T T.800 | ISO/IEC 15444-1, modified to account for the possibility of non-zero CBAP coordinates (z_x, z_y). However, the values of PPx and PPy associated with breakpoint components shall be no smaller than 1. Moreover, for resolutions $r > 0$, the values of PPx and PPy for breakpoint components with TriBPT arrangements, shall be no smaller than 2. These constraints are summarised in Table 1.

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

Breakpoint Arrangement	Resolution	PPx	PPy
QuadBPT	$r = 0$	≥ 1	≥ 1
QuadBPT	$r > 0$	≥ 1	≥ 1
TriLR or TriRL	$r = 0$	≥ 1	≥ 1
TriLR or TriRL	$r > 0$	≥ 2	≥ 2

Each CL band is divided into code-blocks which are organized into precincts in exactly the same way as regular subbands, except that each precinct for a breakpoint tile component at resolution $r > 0$ contains code-blocks from just the one CL band at that resolution, as opposed to HL, LH and HH subbands. Each nominal code-block at resolution r contains a rectangular array of cells that is $2^{xcb'}$ cells wide and $2^{ycb'}$ cells high, except at the boundaries of the tile-component, where code-blocks contain only those cells that intersect with the CL band region, as given by Formula (4). The code-block partition is anchored at the CBAP location (z_x, z_y) introduced above (code-block anchor point), and consists of all code-blocks that have a non-empty intersection with the CL band region. This means that each CL code-block's cell indices occupy a rectangle whose upper-left corner has coordinates (bx_o, by_o) and whose lower-right corner has coordinates $(bx_l - 1, by_l - 1)$, where

$$bx_o = \max\{k_x 2^{\mathrm{xcb'}}, clx_o\},\quad bx_l = \min\{(k_x + 1)2^{\mathrm{xcb'}}, clx_l\},$$

$$by_o = \max\{k_y 2^{\mathrm{ycb'}}, cly_o\},\quad by_l = \min\{(k_y + 1)2^{\mathrm{ycb'}}, cly_l\},$$

(k_x, k_y) are the integer-valued code-block indices, and the set of code-blocks corresponds to all (k_x, k_y) such that $bx_l > bx_o$ and $by_l > by_o$. In the above,

$$\mathrm{xcb}' = \begin{cases} \min\{\mathrm{xcb}, \mathrm{PPx} - 1\} & r > 0 \\ \min\{\mathrm{xcb}, \mathrm{PPx}\} & r = 0 \end{cases} \qquad \mathrm{ycb}' = \begin{cases} \min\{\mathrm{ycb}, \mathrm{PPy} - 1\} & r > 0 \\ \min\{\mathrm{ycb}, \mathrm{PPy}\} & r = 0 \end{cases}, \tag{5}$$

where xcb and ycb are signalled in COD and COC marker segments. These formulae are identical to the formulae for xcb′ and ycb′ in Rec. ITU-T T.800 | ISO/IEC 15444-1.

NOTE 1 - The CL band region, given by Formula (4), can differ slightly from the regions associated with LH, HL or HH bands of a non-breakpoint tile-component at the same resolution, as given by determined in accordance with Rec. ITU-T T.800 | ISO/IEC 15444-1. It turns out that every precinct of a breakpoint tile-component is non-empty, in the sense that it contains at least one code-block that contains at least one cell. By contrast, it is possible for a precinct of a non-breakpoint tile-component to be empty, having no code-block from any LH, HL or HH subband, as explained in Rec. ITU-T T.800 | ISO/IEC 15444-1.

The LL band of a breakpoint tile-component is also partitioned into code-blocks, where the partition is anchored at the CBAP location (z_x, z_y), with code-blocks of width $2^{\mathrm{xcb'}}$ and height $2^{\mathrm{ycb'}}$, where xcb' and ycb' are still found using Formula (5). However, LL code-blocks are measured in grid-points rather than cells, and code-blocks on the boundary of the tile-component contain only those grid-points that belong to the region associated with resolution 0, as given by Formula (3) with $r = 0$. Specifically, each LL code-block's grid-points have indices that lie within a rectangle whose upper-left corner has coordinates (Bx_o, By_o) and whose lower-right corner has coordinates $(Bx_l - 1, By_l - 1)$, where

$$Bx_o = \max\{z_x + k_x 2^{\mathrm{xcb'}}, trx_o\},\quad Bx_l = \min\{z_x + (k_x + 1)2^{\mathrm{xcb'}}, trx_l\},$$

$$By_o = \max\{z_y + k_y 2^{\mathrm{ycb'}}, try_o\},\quad By_l = \min\{z_y + (k_y + 1)2^{\mathrm{ycb'}}, try_l\},$$

(k_x, k_y) are the integer-valued code-block indices, and the set of LL code-blocks corresponds to all (k_x, k_y) such that $Bx_l > Bx_o$ and $By_l > By_o$.

NOTE 2 - The number and sizes of LL code-blocks from a breakpoint tile-component are identical to those of an identically dimensioned non-breakpoint component with the same N_L, PPx, PPy, xcb and ycb parameters.

Even though LL code-blocks of a breakpoint tile-component are dimensioned in terms of grid-points, their coded information is based on 2×2 cells. Each LL code-block's grid-points are separately partitioned into cells, such that the upper-left corner of the top-left cell in the code-block, and the lower-right corner of the bottom-right cell in the code-block have coordinates

$$\left(2LLx_o + z_x, 2LLy_o + z_y\right) \text{ and } \left(2LLx_l + z_x - 1, 2LLy_l + z_y - 1\right),$$

respectively, where the inclusive lower and exclusive upper bounds on the block's cell coordinates are

$$LLx_o = \left\lfloor \frac{Bx_o - z_x}{2} \right\rfloor, LLy_o = \left\lfloor \frac{By_o - z_y}{2} \right\rfloor, LLx_l = \left\lceil \frac{Bx_l - z_x}{2} \right\rceil \text{ and } LLy_l = \left\lceil \frac{By_l - z_y}{2} \right\rceil$$

This means that the left half of each cell on the left boundary of the LL code-block lies outside the block if $Bx_o \bmod 2 \neq z_x$, a condition that can only occur for code-blocks on the left edge of the LL band. Similarly, the upper half of each cell on the top boundary of the code-block lies outside the block if $By_o \bmod 2 \neq z_y$,