
**Information technology — JPEG 2000
image coding system —**

**Part 17:
Extensions for coding of discontinuous
media**

*Technologies de l'information — Système de codage d'images JPEG
2000 —*

Partie 17: Extensions pour le codage des supports discontinus



Reference number
ISO/IEC 15444-17:2023(E)

Documents
Document Sharing Hub
© ISO/IEC 2023



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

INTERNATIONAL STANDARD ISO/IEC 15444-17
RECOMMENDATION ITU-T T.816 (V1)**Information technology – JPEG 2000 image coding system:
Extensions for coding of discontinuous media****Summary**

Rec. ITU-T T.816 | ISO/IEC 15444-17 provides extensions of the scalable image coding tools described in Rec. ITU-T T.800 | ISO/IEC 15444-1 and Rec. ITU-T T.801 | ISO/IEC 15444-2, of two types. First, new wavelet-like image transforms known as "breakpoint-dependent" transforms are defined, whose underlying basis functions can be discontinuous at defined locations within the image component to which they are applied. Second, new scalable coding tools are described for a new type of image component known as a "breakpoint component", which provides a successively refinable and hierarchical description of the breakpoint locations used by the breakpoint-dependent transforms. Any non-initial component or components within a codestream conforming to this Recommendation | International Standard can be breakpoint components and any of the components in the codestream other than breakpoint components can use a breakpoint-dependent transform that depends upon one of the breakpoint components in the same codestream. These new tools together allow for the scalable coding of imagery that naturally exhibits strong discontinuities in the spatial domain. An important example of such imagery is depth maps.

This Recommendation was developed jointly with ISO/IEC JTC 1/SC 29/WG 1 (JPEG) and is common text with ISO/IEC 15444-17.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T T.816 (V1)	2023-02-13	16	11.1002/1000/11830-en

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2023

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

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

	Page
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 ≠ 0.....	61

List of Figures

Page

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.....	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 $FB = 2$, shown for the QuadBPT case. Also shows the mapping of vertices with precision 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 bsa and gradients $Gbsa$ and $Gwdg$ along with corresponding binary flags ORN and DRCTNX_FLAG induced for root arcs.....	16
Figure 13 – Example showing base-arc bsa and gradients $Gbsa$ and $Gwdg$ along with corresponding binary flags ORN and DRCTNX_FLAG induced for diagonal non-root arc containing a spatially induced break	16
Figure 14 – Examples of arcs prl parallel in orientation to root arcs centred at p and containing a spatially induced break. Gradient calculated across arc prl is denoted as $Gprl$	19
Figure 15 – Example of arc prl parallel to the non-root arc centred and at p and containing a spatially induced break. Corresponding gradient across arc prl is shown as $Gprl$	19
Figure 16 – Algorithm for determining the gradient Gb based on validity flags VALID_WDG and VALID_PRL and corresponding attributes.....	20
Figure 17 – Parent 4-span arrangement at resolution $r_{rt} = r - 1$ for arc a at resolution r and centred at p	21
Figure 18 – Algorithm for determining the gradient 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 $2n, 2m$	24
Figure 21 – Phase 1 update step for (even, odd) location $2n, 2m + 1$ and (odd, even) location $2n + 1, 2m$	25
Figure 22 – Phase 1 predict step for (odd, odd) location $2n + 1, 2m + 1$ with source locations of Ic, r *shown.....	26
Figure 23 – Phase 2 update step for (even, even) location $2n, 2m$	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 2×2 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 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

Introduction

The JPEG 2000 core coding system specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 and its extensions specified in Rec. ITU-T T.801 | ISO/IEC 15444-2 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 sub-band samples are quantized.

To address these challenges, this Recommendation | International Standard introduces alternate "breakpoint-dependent" spatial wavelet transforms that are 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 Recommendation | International Standard 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.

**INTERNATIONAL STANDARD
ITU-T RECOMMENDATION**

**Information technology – JPEG 2000 image coding system:
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 Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- Recommendation ITU-T T.800 (2019) | ISO/IEC 15444-1:2019, *Information technology – JPEG 2000 image coding system: Core coding system*.
- Recommendation ITU-T T.801 (2021) | ISO/IEC 15444-2:2021, *Information technology – JPEG 2000 image coding system: Extensions*.

3 Definitions

3.1 Terms defined elsewhere

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

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

- ITU Terms and definitions database: available at <https://www.itu.int/go/terms>
- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

3.2 Terms defined in this Recommendation | International Standard

This Recommendation | International Standard defines the following terms:

- 3.2.1 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.
- 3.2.2 4-span:** 2×2 configuration of 2-spans (3.2.1), involving 25 grid-points, such that the four corner grid-points all have coordinates that are divisible by 4.
- 3.2.3 ambivalent break:** Induced break (3.2.5) that has insufficient precision to determine whether the break occurs in the first or second half of the arc.
- 3.2.4 arc:** Line segment connecting grid-points with even valued coordinates at any resolution of a breakpoint tile-component.

3.2.5 break: Explicitly decoded or inferred location within an arc (3.2.4) that indicates a boundary within an image component.

NOTE 1 – Breaks modify the behaviour of breakpoint-dependent transforms on that other image component.

NOTE 2 – An arc has at most one break.

3.2.6 breakpoint: Data structure consisting of break type, location and precision information for an arc (3.2.4).

NOTE – Type-0 breakpoints have no break at all.

3.2.7 breakpoint component: JPEG 2000 codestream image component that represents arc breakpoint (3.2.6) information.

3.2.8 breakpoint tile-component: All the breakpoints of a given tile, within a breakpoint component (3.2.7).

3.2.9 cell: 2×2 configuration of grid-points within a resolution of a breakpoint tile-component (3.2.8).

NOTE – Cells belong to a well-defined partition that is anchored at the global code-block anchor point.

3.2.10 CL band: The sole sub-band associated with each resolution of a breakpoint tile-component (3.2.8) other than the lowest resolution.

3.2.11 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]

3.2.12 directly induced break: Break (3.2.5) on an arc that is inferred from a break on a parent arc.

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

3.2.14 indefinite break: Induced break (3.2.15) on an arc that is derived from one or more ambivalent breaks (3.2.3), such that there is insufficient precision to determine whether any break at all exists on the arc.

3.2.15 induced break: Break (3.2.5) on an arc that is inferred from breaks on other arcs.

3.2.16 induction block: Condition on an arc (3.2.4) that is explicitly recovered from the decoding of breakpoint code-blocks, indicating that no break (3.2.5) shall be induced on that arc.

3.2.17 non-root arc: Arc (3.2.4) that is not a root arc.

3.2.18 parent arc: Arc (3.2.4) at depth $d + 1$ in the breakpoint decomposition that contains an arc at depth d .

NOTE – At most two arcs at depth d can have the same parent at depth $d + 1$.

3.2.19 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.

3.2.20 QuadBPT: Breakpoint (3.2.6) arrangement involving only horizontal and vertical arcs (3.2.4).

3.2.21 root arc: Arc (3.2.4) that is not contained within any arc projected from the next lower resolution of a breakpoint tile-component (3.2.8), regardless of whether that next lower resolution actually exists within the tile-component's resolution hierarchy.

3.2.22 spatially induced break: Induced break (3.2.15) that is inferred from breaks on non-parent arcs.

3.2.23 tick-point: Possible break (3.2.5) location along an arc.

3.2.24 TriBPT: Breakpoint (3.2.6) arrangement involving horizontal, vertical and diagonal arcs (3.2.4).

3.2.25 TriBPT-LR: TriBPT (3.2.24) 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 (3.2.8).

3.2.26 TriBPT-RL: TriBPT (3.2.24) 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 (3.2.8).

3.2.27 vertex: Explicitly coded break (3.2.5) location.

3.2.28 zero-complete: Code-block within a breakpoint component (3.2.7) that makes no contribution to any codestream packet and is identified as complete by the first packet header of its precinct.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

BD-IDWT Breakpoint-Dependent Inverse Discrete Wavelet Transformation

BGP	Base Grid Point
BIL	Break Inducing Line
BIP	Base Intersection Point
BLN	Base Line
BSA	Base Arc
CBAP	Code-Block Anchor Point
DIB	Directly Induced Breaks
HDT	Hierarchical Data Type
HGP	Horizontal Grid-Point
MPS	More Probable Symbol
RAC	Round Away from Centre
RTN	Round to Nearest
SBR	Source Breaks
SPB	Spatially induced Break
TPS	Tick-Points
VMR	Vertex Mapping Rule
VRT	Vertex
WDA	Wedge Area

5 Conventions

This Recommendation | International Standard uses the following conventions:

- **CBAP** (z_x, 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.

6 Conformance

6.1 Codestream conformance

A codestream conforming to this Specification shall conform to Annex A.

6.2 Decoder

A decoder conforming to this Specification shall process a codestream that conforms to this Specification in the manner specified in Rec. ITU-T T.800 | ISO/IEC 15444-1 together with any additional signalled capability specified in this Specification, 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 7;
- tile-components that use a breakpoint-dependent spatial wavelet transform shall be processed in accordance with clause 8; and
- breakpoint components shall be reconstructed from breakpoint code-blocks in accordance with clause 10, where breakpoint code-blocks are decoded in accordance with clause 9.

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.

7 Breakpoint component structure

7.1 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 Ihd^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 (X_{siz} , Y_{siz}), image offset (XO_{siz} , YO_{siz}) and sampling factors (XR_{siz}^c , YR_{siz}^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

$$x_o = \left\lceil \frac{XO_{\text{siz}}}{XR_{\text{siz}}^c} \right\rceil, \quad x_l = \left\lceil \frac{X_{\text{siz}}}{XR_{\text{siz}}^c} \right\rceil, \quad y_o = \left\lceil \frac{YO_{\text{siz}}}{YR_{\text{siz}}^c} \right\rceil, \quad y_l = \left\lceil \frac{Y_{\text{siz}}}{YR_{\text{siz}}^c} \right\rceil. \quad (1)$$

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

$$tcx_o = \left\lceil \frac{tx_o}{XR_{\text{siz}}^c} \right\rceil, \quad tcx_l = \left\lceil \frac{tx_l}{XR_{\text{siz}}^c} \right\rceil, \quad tcy_o = \left\lceil \frac{ty_o}{YR_{\text{siz}}^c} \right\rceil, \quad tcy_l = \left\lceil \frac{ty_l}{YR_{\text{siz}}^c} \right\rceil, \quad (2)$$

where the tile in question occupies 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 equations 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 $r = 0, 1, \dots, N_L$; 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

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

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 $r = 0$ 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 $d = N_L + 1$.

7.2 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.

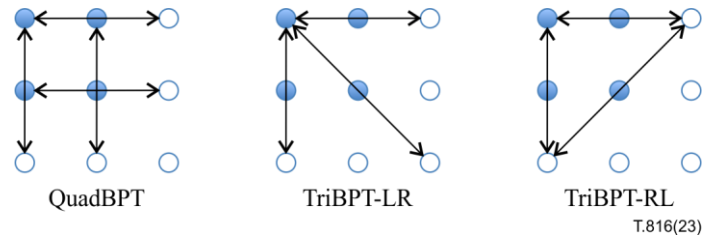


Figure 1 – Arc arrangements within a single 2-span. A 2×2 cell is shown here with solid grid-points. See Figure 2 for other geometric relationships between cells and 2-spans

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.

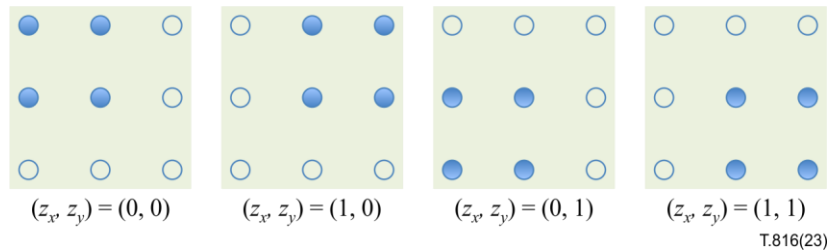


Figure 2 – Cell-span geometry for each of the four possible code-block anchor points

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

$$clx_o = \left\lfloor \frac{trx_o - z_x}{2} \right\rfloor, \quad clx_l = \left\lfloor \frac{trx_l - z_x}{2} \right\rfloor, \quad cly_o = \left\lfloor \frac{try_o - z_y}{2} \right\rfloor, \quad cly_l = \left\lfloor \frac{try_l - z_y}{2} \right\rfloor. \quad (4)$$

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 $r = 1, \dots, N_L$ of a breakpoint tile-component is assigned a single "detail band", similar to the LH, HL and HH sub-bands 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 dissect 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 clause 9.

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.

7.3 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

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

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

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 summarized 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 sub-bands, 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 sub-bands. 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

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

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

(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,

$$xcb' = \begin{cases} \min \{xcb, PPx - 1\} & r > 0 \\ \min \{xcb, PPx\} & r = 0 \end{cases} \quad ycb' = \begin{cases} \min \{ycb, PPy - 1\} & r > 0 \\ \min \{ycb, PPy\} & r = 0 \end{cases} \quad (5)$$

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 sub-band, 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 $2^{xcb'}$ and height $2^{ycb'}$, 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

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

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

(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

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

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

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

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 $By_o \bmod 2 \neq z_y$, a condition that can only occur for code-blocks on the upper edge of the LL band. The right half of cells on the right edge of a code-block will lie outside the block if $Bx_l \bmod 2 \neq z_x$, while the lower half of cells on the bottom edge of a code-block will lie outside if $By_l \bmod 2 \neq z_y$; these conditions can only occur for code-blocks on the right and bottom edges of the LL band, respectively. Regardless of whether all grid-points of a cell lie within the code-block to which it belongs, the block decoding procedures described in 9 regard all arcs associated with such grid-points as if they belong to the block.

The number of cells that are found within any code-block is given by

$$W_{\text{blk}} = bx_l - bx_o \text{ and } H_{\text{blk}} = by_l - by_o,$$

where bx_o , by_o , bx_l and by_l become LLx_o , LLy_o , LLx_l and LLy_l for LL band code-blocks.

7.4 Root and non-root arc breakpoint associations

Breakpoints are associated with the grid-points of a 2×2 cell, and serve to describe possible breaks along one of the arcs that span the corresponding 2-span.

For the QuadBPT arrangement shown in Figure 1, there are four breakpoints per cell, as follows:

- Horizontal *non-root arc* breakpoints are associated with each of a resolution's grid-points (p_x, p_y) for which p_x is odd and p_y is even, serving to describe possible breaks along the horizontal arc that is centred at (p_x, p_y) .
- Vertical *non-root arc* breakpoints are associated with each of a resolution's grid-points (p_x, p_y) for which p_x is even and p_y is odd, serving to describe possible breaks along the vertical arc that is centred at (p_x, p_y) .
- Horizontal *root arc* breakpoints and vertical *root arc* breakpoints are both associated with each of a resolution's grid-points (p_x, p_y) for which p_x and p_y are both odd, serving to describe possible breaks along the horizontal and vertical arcs that are centred at (p_x, p_y) .

The term *non-root arc* here is used to describe an arc that shares an end-point with an arc of the same orientation at the next lower resolution (whether it exists or not); each non-root arc thus corresponds to one half of an arc at the lower resolution. By contrast, the *root arcs* are those that are not subsets of any arc at lower resolutions.

For the TriBPT arrangements shown in Figure 1, there are three breakpoints per cell, as follows:

- Horizontal arc breakpoints are associated with each of a resolution's grid-points (p_x, p_y) for which p_x is odd and p_y is even, serving to describe possible breaks along the horizontal arc that is centred at (p_x, p_y) .
- Vertical arc breakpoints are associated with each of a resolution's grid-points (p_x, p_y) for which p_x is even and p_y is odd, serving to describe possible breaks along the vertical arc that is centred at (p_x, p_y) .
- Diagonal arc breakpoints are associated with each of a resolution's grid-points (p_x, p_y) for which p_x and p_y are both odd, serving to describe possible breaks along the diagonal arc that is centred at (p_x, p_y) . These diagonal arcs run from the top-left to bottom-right corners of the 2-span in the TriBPT-LR case, and from the top-right to the bottom-left corners of the 2-span in the TriBPT-RL case.

The TriBPT arrangements also have root and non-root arcs that can be observed across four 2-spans, known collectively as a *4-span*. The corners of a 4-span correspond to grid-points whose coordinates are divisible by 4. Figure 3 identifies the root and non-root arcs for all configurations.

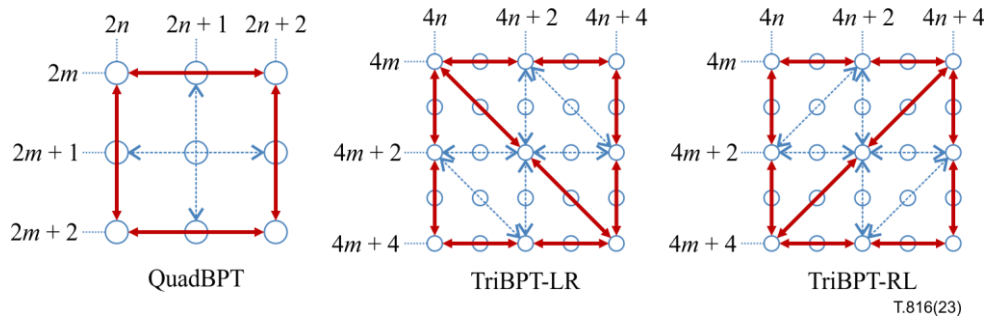


Figure 3 – Root arc (dotted lines) and non-root arcs (solid lines)

7.5 Breakpoint values and vertices

Each breakpoint tile-component has an associated parameter F_B that controls the cardinality of the set of possible break locations on each arc. Each arc at resolution r has length 2 in its own resolution, which is equivalent to a length of 2^d at the full resolution of the tile-component, where

$$d = N_L + 1 - r,$$

and each such arc has 2^{d+F_B} possible break locations, of the form

$$\text{breakposition} = (2k_b + 1)2^{-(F_B+1)} \quad (6)$$

Here, k_b is an integer in the range $0 \leq k_b < 2^{d+F_B}$ for an arc b . These possible break locations, indexed by k_b , are known here as *tick-points*. Note that diagonal arcs and their tick-points are projected onto the vertical axis for measurement purposes, so their lengths are still equal to 2^d .

The separation between tick-points, also known as the *tick-interval*, is 2^{-F_B} . The smallest possible value for F_B is 0, in which case the tick-interval is 1 and arcs at the finest resolution ($r = N_L$, $d = 1$) can have breaks at only two possible locations, corresponding to the centres of the first and second halves of the arc. This is minimally sufficient to allow the breakpoints to be used in guiding the prediction processes of a breakpoint-dependent transform, as discussed in clause 8. Larger values of F_B allow breaks to be more precisely located along an arc. Figure 4 illustrates the tick-points resulting from a typical value of $F_B = 2$, for the case of the QuadBPT arrangement, at the finest resolution level.

NOTE – The F_B parameter can be interpreted as the number of "fraction bits" associated with the break location index k_b .

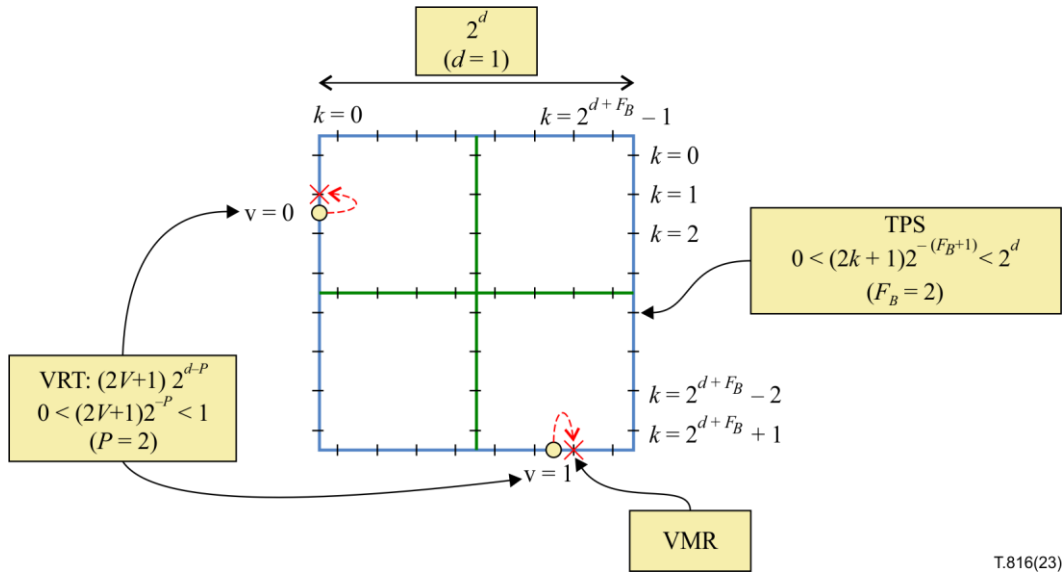
In addition to the break index k_b , each arc has a *break-type* value t_b in the range 0 to 3 and an integer *break-precision* value P_b that relates to the amount of information derived from the decoding of breakpoint code-blocks, from which breaks are inferred. For the TriBPT arrangement, breakpoints are also associated with an *extrapolation-qualifier* e_b that conveys information for performing extrapolation for arcs containing directly induced ($t_b = 2$) or spatially induced ($t_b = 1$) breaks. For all other break-type values the e_b parameter is irrelevant and can be ignored.

In following discussions, for the QuadBPT arrangement the parameters of a breakpoint on an arc b are denoted by (t_b, P_b, k_b) while for the TriBPT arrangement the breakpoint parameters are given by (t_b, P_b, k_b, e_b) . Each of these parameters and the values that they can represent are discussed below.

Break-type values are as follows:

- $t_b = 3$ means that there is an explicitly coded break on the arc, known as a *vertex*; vertices are the significant values recovered from decoded breakpoint code-blocks. Vertex decoding is discussed in clause 9.
- $t_b = 2$ means that there is a break on the arc that is directly induced from a vertex or another directly induced break at the next lower resolution; direct induction is possible only for non-root arc breakpoints. Direct induction is discussed in 10.2.
- $t_b = 1$ means that there is a spatially induced break on the arc; spatial induction occurs when breaks are induced onto root-arcs based on breaks found in surrounding arcs at the same resolution, after which the condition is passed on recursively (through direct induction) to sub-arcs in finer resolutions. Spatial induction is discussed in 10.3 and 10.4.
- $t_b = 0$ means that there is no break on the arc.

The meaning of the break-precision value P_b depends upon the break-type t_b , as discussed next.



T.816(23)

VRT vertex

TPS tick-points

VMR vertex mapping rule of mapping vertex to nearest tick-point "away from arc centre"

NOTE – This figure depicts the case where $d = 1$, $P = \text{precision bits} = 2$ and $F_B = 2$ with arc length given by 2^d .

Figure 4 – Possible break locations (tick-points) at the finest resolution, with $F_B = 2$, shown for the QuadBPT case. Also shows the mapping of vertices with precision P_b to tick-points

Vertex Precision: For vertices ($t_b = 3$), the break-precision P_b shall satisfy

$$1 \leq P_b - 1 \leq d + F_B \quad (7)$$

where $P_b - 1$ is the number of bits of the vertex V_b that were actually decoded. The value $(2V_b + 1)$ is a P_b -bit unsigned integer, and the relationship between the tick-point k_b and the decoded vertex value V_b is as follows:

$$k_b = (2V_b + 1)2^{d+F_B-P_b} + \begin{cases} 0, & \text{if } (2V_b + 1)2^{-P_b} > \frac{1}{2} \\ -1, & \text{if } (2V_b + 1)2^{-P_b} < \frac{1}{2} \end{cases} \quad (8)$$

Figure 4 illustrates this mapping of the $(P_b - 1)$ bit vertex value V_b to tick-point k_b . The quantity $(2V_b + 1)2^{-P_b} \in (0,1)$ can be interpreted as a relative vertex position. As seen in the figure, the role of the offset on the right hand side of Formula (8) is to round this relative position outwards (away from the centre of the arc at relative position $\frac{1}{2}$) to the nearest available tick-point.

Induced break precision: Directly induced breaks ($t_b = 2$) are obtained on non-root arcs, from vertices or other directly induced breaks at the next lower resolution, by decrementing P_b and discarding the most significant bit of k_b . Specifically, using the term "parent" to identify the corresponding arc in the next lower resolution,

$$P_{\text{induced}} = \max \{1, P_{\text{parent}} - 1\}, k_{\text{induced}} = k_{\text{parent}} \bmod 2^{d+F_B-1} \quad (9)$$

Spatially induced breaks ($t_b = 1$) are derived from breaks on surrounding arcs at the same resolution, and their precision is the smallest value of P_b associated with those surrounding arcs whose breaks are involved in the induction.

Induced breaks with $P_b = 1$ are said to be *ambivalent*, since insufficient information is preserved to determine whether the break occurs in the first or second half of the arc. Both directly and spatially induced breaks can be ambivalent.

The special value $P_b = 0$ is used to identify *indefinite* breaks; these are induced breaks whose existence is uncertain because one or more of the arcs used in the induction process has break-precision $P_b \leq 1$ that is too small to determine whether a break should exist. Both directly and spatially induced breaks can be indefinite.

No-break precision: When $t_b = 0$, meaning that there is no break on an arc, the value of k_b is meaningless, but the parameter P_b can be either 0 or 1. The combination $t_b = 0, P_b = 1$ means that the arc contains an *induction block*. Induction blocks prevent any break from being induced on the arc. Induction blocks are derived from decoding breakpoint code-blocks.

The extrapolation-qualifier e_b is comprised of two binary flags $e_{b,0}$ and $e_{b,1}$, such that $e_b = (e_{b,1}, e_{b,0})$. A break divides an arc into two parts and the binary value of $e_{b,0}$ conveys information required for performing extrapolation for the first

part while $e_{b,1}$ corresponds to information for the second part of the arc. These binary flags have specific and distinct meaning for directly induced ($t_b = 2$) and spatially induced ($t_b = 1$) breaks for the TriBPT arrangement as detailed below.

For a directly induced break, $e_{b,0} = \text{GRAD1_FLAG}_{\text{induced}}$ and $e_{b,1} = \text{GRAD2_FLAG}_{\text{induced}}$. The predict step for an arc containing a directly induced break is conditional on these flags as described in 8.2.4. The values for $\text{GRAD1_FLAG}_{\text{induced}}$ and $\text{GRAD2_FLAG}_{\text{induced}}$ can be derived during the breakpoint synthesis stage and this is detailed in 10.2.2.

If an arc contains a spatially induced break, then $e_{b,0} = \text{DRCTN1_FLAG}_{\text{spatial}}$ and $e_{b,1} = \text{DRCTN2_FLAG}_{\text{spatial}}$. As detailed in 8.2.5, these binary flags signify a direction of search for an external gradient and are utilized in the predict step for arcs containing spatially induced breaks. The values for $\text{DRCTN1_FLAG}_{\text{spatial}}$ and $\text{DRCTN2_FLAG}_{\text{spatial}}$ can be determined from the 4-span triangle and inducing line geometry as explained in 10.4.4. For both flags a value of 1 indicates a search in the increasing x-coordinate direction for horizontal arcs and in the increasing y-coordinate direction for vertical and diagonal arcs. Conversely, a value of 0 indicates a search in the decreasing x-coordinate direction for horizontal arcs and in the decreasing y-coordinate direction for vertical and diagonal arcs.

8 Breakpoint-dependent spatial wavelet transformation

8.1 Overview

To perform breakpoint-dependent inverse discrete wavelet transform (BD-IDWT), this Recommendation | International Standard uses the synthesized breakpoints, as described in clause 10, to adapt a two-dimensional reconstruction of sub-band samples obtained by processing the codestream in accordance with Rec. ITU-T T.800 | ISO/IEC 15444-1 together with any additional signalled capability.

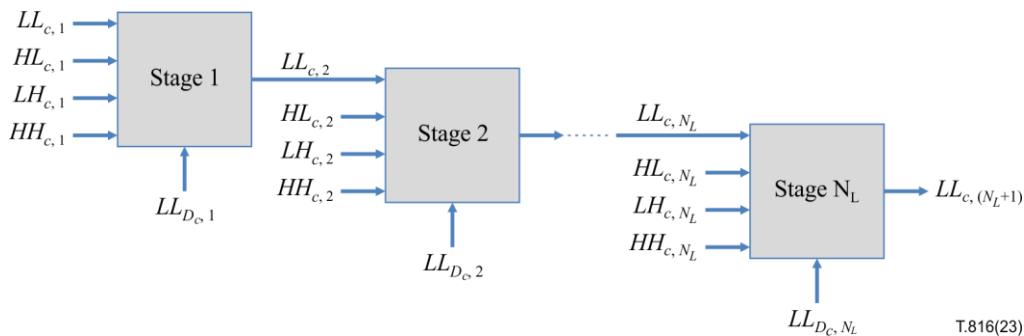
The BD-IDWT is performed in stages for a given tile t of a non-breakpoint component with index c , progressing from coarse to fine resolutions $r = 1, 2, \dots, N_L$. The BD-IDWT uses spatial sub-band samples found within that tile-component, that are obtained by processing the codestream in accordance with Rec. ITU-T T.800 | ISO/IEC 15444-1, together with any additional signalled capability.

The BD-IDWT also uses breakpoint data from the same tile t of a separate breakpoint component with index D_c , where the value D_c is communicated via the SEcod field specified in Annex A.

Each stage r utilizes the four spatial sub-bands $LL_{c,r}, HL_{c,r}, LH_{c,r}, HH_{c,r}$ that belong to resolution r of the non-breakpoint tile-component indexed by c , together with a corresponding breakpoint band $LL_{D_c,r}$ which contains all breakpoint information at resolution r for the designated breakpoint tile-component indexed by D_c . The breakpoint data provided by $LL_{D_c,r}$ corresponds to the (partially) reconstructed breakpoints that appear at the same resolution r as determined using the reconstruction procedure in clause 10. Stage r uses this information to synthesize a sub-band $LL_{c,(r+1)}$, for use in the next stage, if any. The iterative procedure is illustrated in

$LL_{D_c,i}$ denotes the breakpoint LL band for resolution $r=i$.

Figure 5 where the output of the final stage $LL_{c,(N_L+1)}$ constitutes the decoded samples of the tile-component c .



$LL_{D_c,i}$ denotes the breakpoint LL band for resolution $r=i$.

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

In the simplest case, where both components have the same number of decomposition levels N_L and identical sampling factors XRsiz^i and YRsiz^i reconstructed breakpoint data for resolution r is combined directly with the four spatial sub-bands $LL_{c,r}, HL_{c,r}, LH_{c,r}, HH_{c,r}$ to perform the reconstruction at stage $r = 1, 2, \dots, N_L$.

If the breakpoint component and breakpoint-dependent component have different sampling factors XR_{siz}^i and YR_{siz}^i , the transformation process is not well defined. Some decoders might be capable of producing meaningful results when this happens, but this Recommendation | International Standard does not define the behaviour under such circumstances.

If the breakpoint component provides Δ_L more decomposition levels N_L than the breakpoint-dependent component, the transformation processes described here shall be applied after renumbering the resolutions of the breakpoint component so that original resolution r is interpreted as resolution $r - \Delta_L$. This effectively discards the unnecessary lower resolutions of breakpoint information.

If the breakpoint component provides only $N_L - 1$ decomposition levels, for the breakpoint-dependent component's N_L decomposition levels, the transformation processes described here shall be performed after renumbering the resolutions of the breakpoint component so that original resolution r is interpreted as resolution $r + 1$. This leaves the breakpoint component's LL band at resolution 0 providing the breakpoint data for the first stage of the transform at $r = 1$. If the breakpoint component provides even fewer decomposition levels, the transformation process is not defined by this Recommendation | International Standard.

Each stage of the BD-IDWT is performed in two steps. First the four spatial sub-bands $LL_{c,r}, HL_{c,r}, LH_{c,r}, HH_{c,r}$ are subjected to a mapping procedure 2D_GP_MAPPING that produces a single interleaved output band $I_{c,r}$. A breakpoint dependent 2D sub-band reconstruction procedure BD_2D_SR is then applied to $I_{c,r}$ that depends on the corresponding breakpoint band $LL_{D_c,r}$.

8.2 TriBPT-dependent irreversible transforms

8.2.1 Introduction

For the TriBPT case, the 2D_GP_MAPPING procedure maps the spatial sub-bands $LL_{c,r}, HL_{c,r}, LH_{c,r}, HH_{c,r}$ to corresponding grid-point locations to form an interleaved band $I_{c,r}$ such that $I_{c,r} = 2D_GP_MAPPING(LL_{c,r}, HL_{c,r}, LH_{c,r}, HH_{c,r}, trx_o, trx_l, try_o, try_l)$ where (trx_o, try_o) specifies the upper left corner and $(trx_l - 1, try_l - 1)$ the lower right corner of the rectangular grid-point region at resolution r that corresponds to $I_{c,r}$. The mapping of sub-band samples to grid-point locations for a given resolution r is illustrated in Figure 6; while the specific example shown relates to the TriBPT-LR case, the mapping remains identical for TriBPT-RL.

NOTE – As shown in Figure 6, the grid-point mapping procedure associates the samples of the $HL_{c,r}, LH_{c,r}, HH_{c,r}$ spatial sub-bands at the same locations as horizontal, vertical and diagonal arcs from the breakpoint component D_c , respectively.

In the description that follows, it is convenient to identify different triangular elements within a 4-span (see Figure 3) by means of two variables, s_{LR} and s_{TB} . The first variable distinguishes between the mirror-image geometries TriBPT-LR and TriBPT-RL, with $s_{LR} = 1$ for the TriBPT-LR arrangement and $s_{LR} = -1$ for the TriBPT-RL arrangement. The second variable distinguishes between the upper and lower triangles within the 4-span, denoted TOP_Tri and BOT_Tri, with $s_{TB} = 1$ and $s_{TB} = -1$, respectively.

This clause defines the TriBPT-dependent irreversible transforms. The reversible version of these transforms are defined in 8.3.

The interleaved band $I_{c,r}$ produced by the 2D_GP_MAPPING procedure, is subject to 2D filtering BD_2D_SR, which consists of two lifting steps. The update step is applied first, modifying sub-band samples at grid-points (p_x, p_y) that are (even, even), based on sub-band samples and breakpoint values at connected neighbouring grid-points with (even, odd), (odd, even) and (odd, odd) locations. The update step is implemented by making the following assignment for each (even, even) location $(2n, 2m)$ of $I_{c,r}$.

$$I_{c,r}(2n, 2m) = I_{c,r}(2n, 2m) - \sum_{(\delta_n, \delta_m) \in \mathcal{N}_U} U(I_{c,r}, LL_{D_c,r}, 2n + \delta_n, 2m + \delta_m) \quad (10)$$

Here, $\mathcal{N}_U = \{(1, 0), (0, 1), (s_{LR}, 1), (-1, 0), (0, -1), (-s_{LR}, -1)\}$ identifies the update neighbourhood, and the update function U is given by

$$U(I_{c,r}, LL_{D_c,r}, p_x, p_y) = \begin{cases} \mu * I_{c,r}(p_x, p_y), & \text{if } t_b(p_x, p_y) = 0, \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where $\mu = \frac{1}{8}$ and $t_b(p_x, p_y)$ is the arc break-type found from $LL_{D_c,r}(p_x, p_y)$. As defined in clause 7, $t_b = 0$ means that there is no break on the arc. Figure 7 shows the operation of the update step for the TriBPT-LR arrangement, providing examples with no breakpoints and when breakpoints are present on neighbouring arcs.

After the update step, the (even, even) indexed sub-band samples located at end points of arcs at resolution r are utilized in performing the second (predict) lifting step. The predict step is implemented by making the following assignment for each (even, odd), (odd, even) and (odd, odd) location within $I_{c,r}$:

$$I_{c,r}(\mathbf{p}) = I_{c,r}(\mathbf{p}) + P(I_{c,r}, LL_{D_{c,r}}, \mathbf{p}), \text{ where } \mathbf{p} = (p_x, p_y) \text{ and } p_x, p_y \text{ are not both even.} \quad (12)$$

For a root or non-root arc centred at location $\mathbf{p} = (p_x, p_y)$, the predict function $P()$ is dependent on the corresponding break-type $t_b(p_x, p_y)$, as well as the location of the break $k_b(p_x, p_y)$, as given by $LL_{D_{c,r}}(p_x, p_y)$. There are in total five distinct cases that give rise to specific breakpoint dependent predict functions, as detailed in 8.2.2 through 8.2.5.

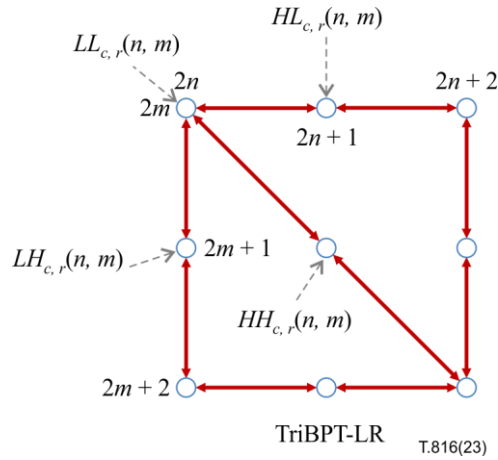
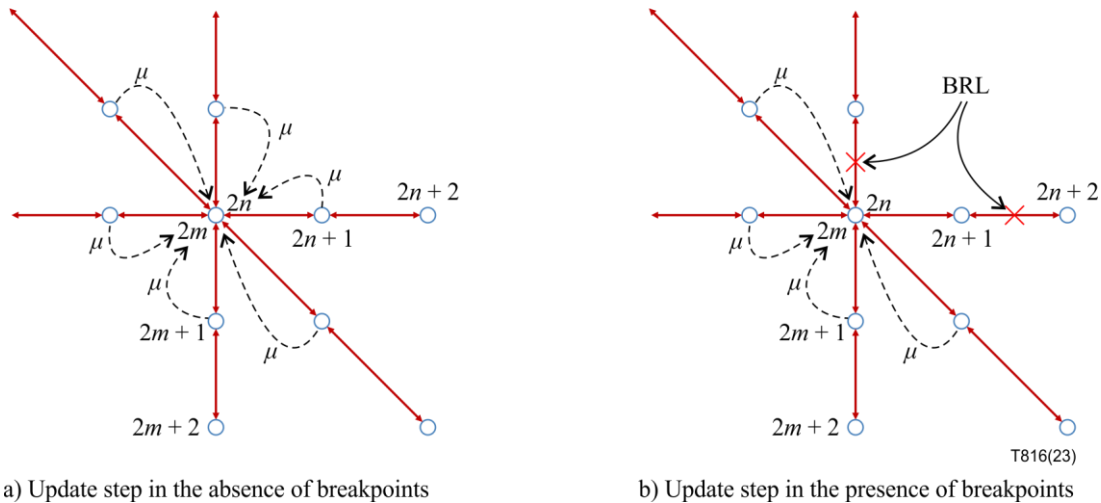
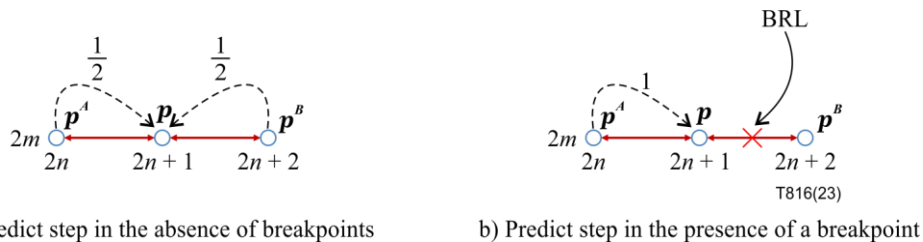


Figure 6 – Interleaving of sub-band coefficients to a triangular grid for a given resolution r



BRL breakpoint location

Figure 7 – Update step examples for TriBPT-LR arrangement



BRL breakpoint location

Figure 8 – Predict step examples for a horizontal arc

8.2.2 Arcs with $t_b = 0$

For arcs without any breaks ($t_b(\mathbf{p}) = 0$), the predict function is given by

$$P(I_{c,r}, LL_{D_{c,r}}, \mathbf{p}) = \frac{1}{2}(I_{c,r}(\mathbf{p}^A) + I_{c,r}(\mathbf{p}^B)) \quad (13)$$

where the prediction source locations $\mathbf{p}^A$ and $\mathbf{p}^B$ are given by

$$[\mathbf{p}^A, \mathbf{p}^B] = \begin{cases} [(2n, 2m), (2n + 2, 2m)] & \text{if } \mathbf{p} = (2n + 1, 2m) \\ [(2n, 2m), (2n, 2m + 2)] & \text{if } \mathbf{p} = (2n, 2m + 1) \\ [(2n, 2m), (2n + 2s_{LR}, 2m + 2)] & \text{if } \mathbf{p} = (2n + s_{LR}, 2m + 1) \end{cases} \quad (14)$$

An example involving a horizontal arc is shown in Figure 8 (left).

8.2.3 Arcs with $t_b = 3$

For arcs containing a vertex ($t_b(\mathbf{p}) = 3$), the predict function is given by

$$P(I_{c,r}, LL_{D_{c,r}}, \mathbf{p}) = \begin{cases} I_{c,r}(\mathbf{p}^A) & k_b(\mathbf{p}) > 2^{d+F_B-1} \\ I_{c,r}(\mathbf{p}^B) & k_b(\mathbf{p}) < 2^{d+F_B-1} \end{cases} \quad (15)$$

where the prediction source locations $\mathbf{p}^A$ and $\mathbf{p}^B$ are given by Formula (14). An example involving a horizontal arc is shown in Figure 8 (right).

8.2.4 Arcs with $t_b = 2$

For arcs containing a directly induced break ($t_b(\mathbf{p}) = 2$), the predict function is given by

$$P(I_{c,r}, LL_{D_{c,r}}, \mathbf{p}) = \begin{cases} I_{c,r}(\mathbf{p}^A) + G_b(\mathbf{p}, 2\mathbf{p}^A - \mathbf{p}) & k_b(\mathbf{p}) > 2^{d+F_B-1} \\ I_{c,r}(\mathbf{p}^B) + G_b(\mathbf{p}, 2\mathbf{p}^B - \mathbf{p}) & k_b(\mathbf{p}) < 2^{d+F_B-1} \end{cases} \quad (16)$$

Where the prediction source locations $\mathbf{p}^A$ and $\mathbf{p}^B$ are given by Formula (14) and $G_b(\mathbf{p}, \mathbf{p}^{nbr})$ refers to a gradient value

$$G_b(\mathbf{p}, \mathbf{p}^{nbr}) = \begin{cases} \frac{1}{2} \left[I_{c,r} \left(\mathbf{p}^{nbr} + \frac{\mathbf{p} - \mathbf{p}^{nbr}}{2} \right) - I_{c,r} \left(\mathbf{p}^{nbr} - \frac{\mathbf{p} - \mathbf{p}^{nbr}}{2} \right) \right] & \text{if } \text{SMOOTH}(\mathbf{p}, \mathbf{p}^{nbr}) \\ 0 & \text{otherwise} \end{cases}$$

Here, $\mathbf{p}^{nbr}$ denotes the location of a neighbouring arc within $LL_{D_{c,r}}$, to the one at location $\mathbf{p}$, and the boolean function $\text{SMOOTH}(\mathbf{p}, \mathbf{p}^{nbr})$ returns TRUE only if the following three conditions are satisfied:

- $\mathbf{p}^{nbr}$ belongs to $LL_{D_{c,r}}$ and
- $t_b(\mathbf{p}^{nbr}) = 0$ and
- $\text{GRADX_FLAG}_{induced}(\mathbf{p}) = 1$

where $\text{GRADX_FLAG}_{induced}$ for the arc at $\mathbf{p}$ is defined as shown below.

$$\text{GRADX_FLAG}_{induced}(\mathbf{p}) = \begin{cases} \text{GRAD1_FLAG}_{induced} & k_b(\mathbf{p}) > 2^{d+F_B-1} \\ \text{GRAD2_FLAG}_{induced} & k_b(\mathbf{p}) < 2^{d+F_B-1} \end{cases}$$

The values of the binary flags $\text{GRAD1_FLAG}_{induced}$ and $\text{GRAD2_FLAG}_{induced}$ are defined by the extrapolation-qualifier e_b for the arc as specified in 7.5.

An example of direct induction is shown in Figure 9. The example shows a vertex at resolution r^{anc} , directly inducing a break on the horizontal arc centred at $\mathbf{p} = (2n + 1, 2m)$ in resolution r . The neighbouring arc nbr at location $\mathbf{p}^{nbr} = (2n + 3, 2m)$ and ancestor arc (anc) are also shown. The extrapolation-qualifier for the second half of the current arc $\text{GRAD2_FLAG}_{induced} = 1$ to indicate that the neighbouring arc nbr is a sub-arc of the ancestor arc anc . Since nbr is free from breaks and a sub-arc of anc , the samples at both endpoints of nbr are used to determine the gradient value $G_b(\mathbf{p}, \mathbf{p}^{nbr})$. The extrapolation-qualifier for the first half of the current arc $\text{GRAD1_FLAG}_{induced} = 0$.

NOTE – As explained in clause 7, $t_b(\mathbf{p}) = 2$ means that the arc at $\mathbf{p}$ contains a break that is recursively induced from a decoded vertex on some ancestor arc at a coarser resolution $r^{anc} < r$. The predict function allows for linear extrapolation within the span of this ancestor arc.

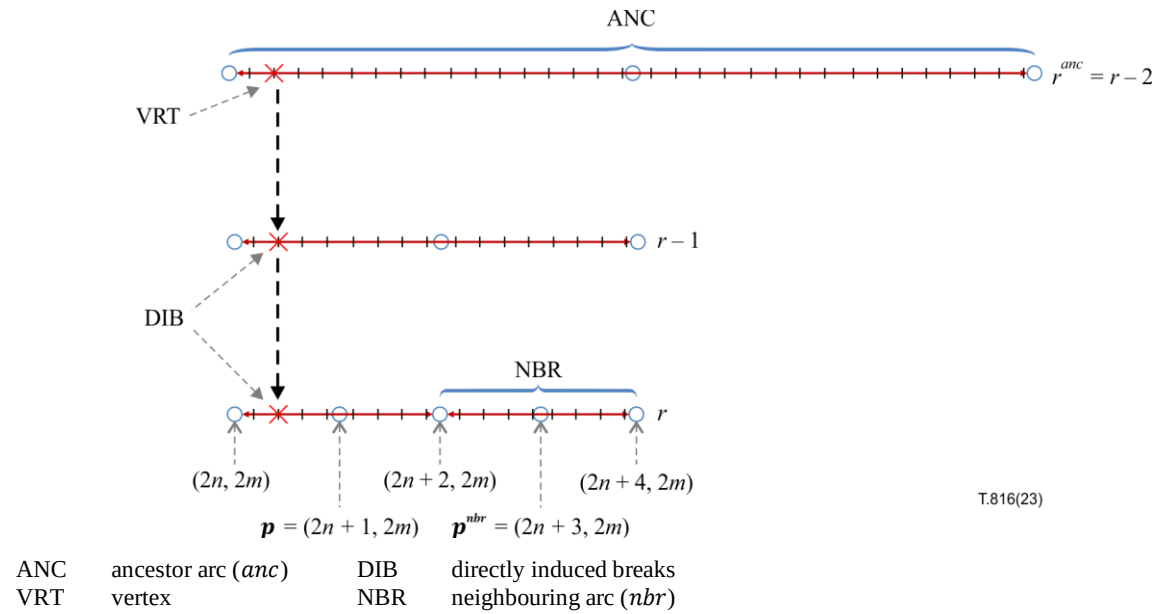


Figure 9 – Example of direct induction of a breakpoint

Figure 10 illustrates examples of spatially induced breaks on root arcs by source breaks on non-root arcs for the TriBPT-LR arrangement. The base-arc and base-grid-point along with the configuration of the corresponding wedge area for each case is shown.

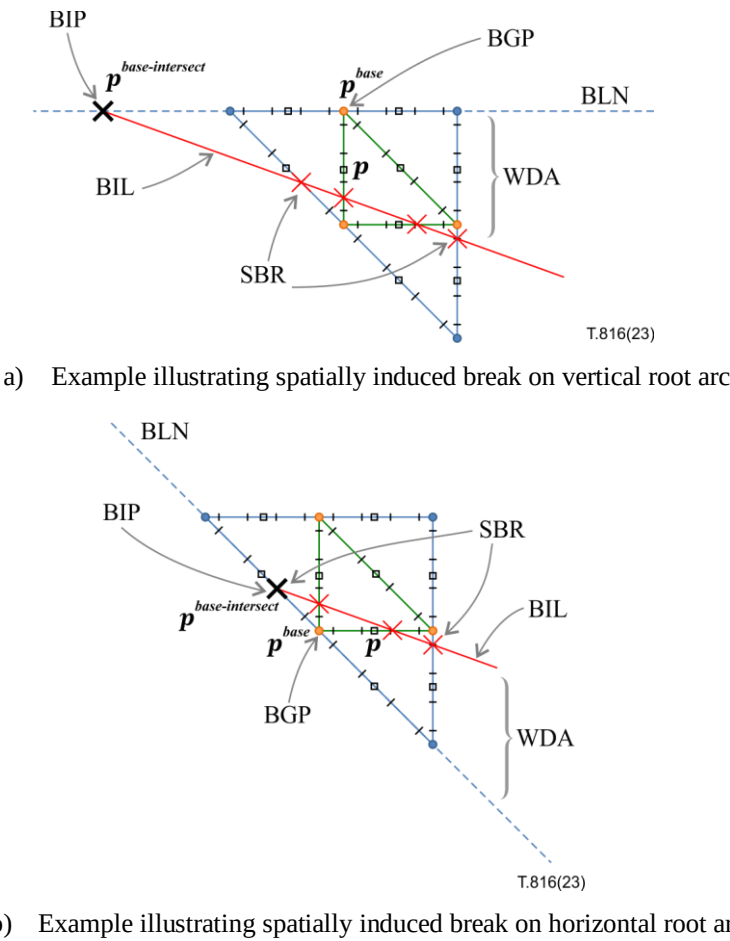


Figure 10 – Example illustrating spatially induced breaks