INTERNATIONAL STANDARD

ISO/IEC 23001-16

First edition
2021-11

Information technology — MPEG systems technologies —

Part 16: Derived visual tracks in the ISO base media file format

Technologies de l'information — Technologies des systèmes MPEG —

Partie 16: Pistes visuelles dérivées au format ISO de base pour les fichiers médias

Reference number
ISO/IEC 23001-16:2021(E)

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

Contents

Page

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. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

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 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 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 23001 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

Derived visual tracks are designed to enable defining a timed sequence of visual transformation operations to be applied to input still images and/or samples of timed sequences of images in the same presentation. It is built using tools defined in the ISO base media file format (ISO/IEC 14496-12). This document specifies the core design and an initial base set of transformation operations.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — MPEG systems technologies —

Part 16:
Derived visual tracks in the ISO base media file format

1 Scope

This document defines a storage format for derived visual tracks and an initial base set of related transformation operations. The format defined in this document enables the interchange, editing, and display of timed sequences of images that result from transformation operations applied to input still images or samples of timed sequences of images in the same presentation.

This format defines normative structures used to contain the description of transformation operations, how to link that transformation operations to the inputs, and defines how to process those transformation operations to obtain a timed sequence of video frames.

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.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO Base Media file format*

ISO/IEC 23001-10, *Information technology — MPEG systems technologies — Part 10: Carriage of timed metadata metrics of media in ISO base media file format*

ISO/IEC 23008-12, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 14496-12 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
derivation operation
container box representing an operation applying a *derivation transformation* (3.2) on an ordered list of *inputs* (3.5)

3.2
derivation transformation
visual transformation operation identified by a 32-bit value and a set of parameters that transforms *inputs* (3.5) into *visual outputs* (3.8)

Note 1 to entry: The 32-bit value is also known as a four-character code in ISO/IEC 14496-12.

3.3
derived sample
sample containing an ordered list of *derivation operations* (3.1)

3.4
derived visual track
video or picture track that contains a timed sequence of *derived samples* (3.3)

3.5
input
parameter input (3.6) or *visual input* (3.7)

3.6
parameter input
metadata from an input item or track that is used as input for a *derivation transformation* (3.2) of a *derivation operation* (3.1)

Note 1 to entry: The parameter input is either an input metadata item from file-level `MetaBox` or an interval of an input metadata track (possibly spanning multiple samples).

3.7
visual input
video or still image that is used as input for a *derivation transformation* (3.2) of a *derivation operation* (3.1)

Note 1 to entry: The visual input is either an input image item from file-level `MetaBox`, an interval of an input track (possibly spanning multiple samples), the visual output of a preceding *derivation operation* (3.1) or the default input fill picture signalled in the configuration record of the *derived visual track* (3.4).

3.8
visual output
one video frame or a sequence of video frames that is output from a *derivation transformation* (3.2) of a *derivation operation* (3.1)

4 Derived visual tracks, design principles

A derived visual track describes a timed sequence of derived samples composed of an ordered list of derivation operations, each derivation operation applying a derivation transformation for the duration of the derived sample on an ordered list of inputs represented in the same presentation.

A derived visual track shall be either a video track (with the `'vide'` handler type in the `HandlerBox` of the `MediaBox` as defined in ISO/IEC 14496-12) or a picture track (with the `'pict'` handler type in the `HandlerBox` of the `MediaBox` as defined in ISO/IEC 23008-12). A derived visual track is identified by its containing sample entry of type `'dtrk'` `DerivedVisualSampleEntry`. Each sample described by a `DerivedVisualSampleEntry` is a derived sample.

A derived visual track shall include a `TrackReferenceTypeBox` with `reference_type` equal to `'dtrk'` referring to all the inputs. Each reference shall be one of:

a) the `track_ID` of a track used by derived samples in the track, or, if unified IDs are in use as defined by ISO/IEC 14496-12, a `track_group_id`;

b) the `item_ID` of an image item, in the file-level `MetaBox`, used by derived samples in the track.

An ID value in the track references is resolved to a `track_ID` whenever the file contains a track with such ID, is resolved to a `track_group_id` whenever unified IDs are in use and the file contains a track group with such ID, and is resolved to an `item_ID` otherwise.

NOTE 1 A `track_ID` can be an ID of a derived visual track.

If a referenced track is a member of an alternate group or switch group, or if the reference is to a track group, then the reader should pick a track from the group as the input to the derived visual track.

NOTE 2 The `TrackSelectionBox` can be used to provide guidance on the selection between members of an alternate group or switch group.

Similarly, if a referenced image item is a member of an alternate group (which may contain both tracks and images), then the reader should pick one member of the group as the input to the derived visual track.

A derived sample contains an ordered list of the derivation operations to be performed, each derivation operation applying a derivation transformation on an ordered list of inputs. The `layer` syntax element in `TrackHeaderBox` has no impact on ordering the inputs for derived samples.

The four-character codes of derivation transformation from all derivation operations used by the derived samples in the track are listed in the `DerivedVisualSampleEntry`, and also default inputs and parameter values can be supplied there. A derived sample in the track may use all or some of the derivation operations listed in the linked `DerivedVisualSampleEntry`, but derived samples shall not use a derivation operation not listed in the sample entry.

The derived sample durations document the time over which the derivation represented by the ordered list of derivation operations is active. Therefore, the number of samples defined in a derived visual track does not necessarily match 1:1 with the number of input image items or samples of input tracks that are being transformed. A single derivation duration may span multiple samples in the source track(s), and also derivation transformations in derived samples may have 'internal time structure' (e.g. a cross-fade) so the picture may change during the sample duration. This is in contrast to 'classic video'.

Derived visual tracks do not respect edit lists on inputs. They operate on the composition timeline (i.e. before the application of edit lists) of their input tracks (including on derived visual tracks when used as visual inputs). However, the input tracks shall not have edit lists. Any edit lists of the input tracks shall be ignored if present.

NOTE 3 When time-alignment adjustment between input tracks is needed, signed composition offsets in input tracks can be used.

NOTE 4 A derived visual track can have an edit list; thus, a derived visual track using the identity transform, and with an edit-list, can provide a visual output that is a temporal re-mapping of the input track.

The inputs for a derivation operation in a derived sample can be either input image items from file-level `MetaBox` or intervals (possibly spanning multiple samples) of input video tracks, image sequence tracks, metadata items or metadata tracks, the visual output of a preceding derivation operation or a default input fill picture.

Transformative item properties or transformations (e.g. clean aperture, track matrix etc...) associated with input image items or samples of input tracks are always applied before performing the derivation operation.

NOTE 5 If a derived sample needs to refer to one explicit sample value in a referred track (other than the time-aligned sample value), an item can be created and referred to that has the same data as the desired sample value.

The visual inputs in a derived sample shall have consistent pixel aspect ratio and bit depth. The input image items, samples of input tracks or derived samples may have various width and height. When differences in width and height result in pixels that never get 'painted' by a derivation operation, those empty pixels are filled according to the value of `default_derivation_input` parameter signalled in `DerivedVisualTrackConfigRecord` (black, white or grey pixels). When differences in width and height result in pixels that end up outside the visual output size by a derivation operation, those pixels are cropped. This default behaviour may be overridden by derivation operation specifications.

A derived sample is reconstructed by performing the specified derivation operations in sequence. Some derivation operations can be marked as non-essential which indicates that the derivation operation may be skipped by the reader. However, the operations marked as essential shall be used in order to obtain a valid derived sample.

When more than one derivation operation is listed in a derived sample, the derivation operation that is not first in the list may include the output result (e.g. the visual output) of any of the previous derivation operations, only new inputs, or a combination of both.

In many cases the source tracks pointed to by the `'dtrk'` track reference are not intended for display. When a track is not intended for display, `track_in_movie` shall be equal to 0 for that track.

The visual output of a derived sample is the output from the last derivation operation in the sample. If there is no derivation operation, an empty derived sample (i.e. sample size of 0) is equivalent to an empty edit, i.e., there is no visual output from the derived visual track at that time.

Using derived visual tracks, it is possible to build either a chain of derivation operations on one single derived visual track or a hierarchy of multiple derived visual tracks when they are used as a visual input to another derived visual track. The latter should only be used when each derived visual track in the hierarchy is also needed on its own.

5 Derivation operation

5.1 Definition

Box Type: `'dimg'`
Container: derived sample or `DerivedVisualTrackConfigRecord` in a `DerivedVisualSampleEntry`
Mandatory: Yes, in a `DerivedVisualTrackConfigRecord`, and No in a derived sample
Quantity: At least one in a `DerivedVisualTrackConfigRecord`, and Zero or more in a derived sample

A derivation operation in either a derived sample entry or derived sample is represented by a container box of type `'dimg'` that always carries a derivation transformation box inherited from VisualDerivationBase, and can carry a `VisualDerivationInputs` providing the inputs for the derivation transformation.

A derivation transformation in a derivation operation is identified by a 32-bit value, also known as a four-character code in ISO/IEC 14496-12, unless that code is `'uuid'`, whereupon a UUID identifies a vendor-specific derivation transformation.

A derivation transformation's parameters shall

a) be single, countable

b) have defined default values in the specification

For both inputs and parameters, there is a bit-mask in the sample entry and a bit-mask in the derived sample that uses the derivation operation, defining whether the parameter value or input is supplied there.

Each parameter takes the value defined in that derived sample, if any, or else the value defined in the sample entry, if any, or else the default value for that parameter defined in the derivation transformation specification.

Each input takes, in precedence order:

a) the input listed in the derived sample if present;

b) the default input listed in the corresponding derivation operation in the configuration record in the sample entry, if present;

c) or the default input fill picture defined in the configuration record for the derived visual track.

Each input can be:

a) an index into the track reference box for the derived visual track (itself indicating either an input image item from the file-level `MetaBox`, an input track, or, if unified IDs are in use as defined by ISO/IEC 14496-12, a track group);

b) a relative index of a previous derivation operation in the same derived sample containing the derivation operation that uses it;

NOTE 1 A relative index declared in the configuration record is pointing to a derivation operation within the derived sample that uses it.

c) or the default input fill picture defined in the configuration record for this derived visual track.

The default input fill picture is signalled in `VisualDerivationDefault` in the `DerivedVisualTrackConfigRecord` in the derived visual track sample entry (e.g. either a full black, mid grey or full white picture).

NOTE 2 The size of the default input fill picture is given by `width` and `height` in the sample entry.

If no inputs are defined in a derivation operation listed in the sample entry, `VisualDerivationInputs` may be absent there; if none are defined in a derivation operation listed in the derived sample (presumably at least some are defined in the sample entry) then `VisualDerivationInputs` may be absent there.

Bits in the masks are assigned from least-significant (first input or parameter) upwards. Their default value is 0 (i.e. if the operation has 9 inputs and only 8 bits are supplied, the 9th bit is assumed to be zero).

The version of `VisualDerivationBase` is currently constrained to be 0; only one flag is defined, the low-order bit. When set to 1, it indicates that the operation is essential. If any derived sample sets this but for a given derivation transformation type (code), then the corresponding derivation operation listed in the sample entry shall set this bit. A parser shall not process a derived visual track that contains a derivation operation marked in the sample entry as essential that is not recognized or not supported by the parser. If a non-essential derivation operation is not supported, the derived sample containing it may be processed as if this non-essential derivation operation was a null derivation operation, i.e. the visual output of a non-essential and unsupported derivation operation is the visual output from the previous derivation operation in the sample. If there is no previous derivation operation, the visual output of the unsupported derivation operation is the default input fill picture defined in the configuration record for the derived visual track.

5.2 Syntax

```
aligned(8) class VisualDerivationBase
extends FullBox (code, version = 0, flags, optional unsigned int(8)[16] uuid_code){
   // the box may be empty and terminate before the following field;
   unsigned int(16) highest_param_idx;
   unsigned int(floor((highest_param_idx+7)/8))*8) parameter_defined_flags;
   // the remaining bytes are the values of parameters signalled by parameter_defined_
flags;
}

aligned(8) class VisualDerivationInputs
extends FullBox ('dinp', version = 0, flags = 0){
   unsigned int(16) highest_input_idx;
   unsigned int(floor((highest_input_idx+7)/8))*8) input_present_flags;
   unsigned int(16) reference_index[];
}

aligned(8) class VisualDerivation extends Box ('dimg'){
   VisualDerivationBase()   derivation_transformation;    // actually an instance of a
derived class
   VisualDerivationInputs   inputs;    // optional
}
```

5.3 Semantics

`highest_param_idx` provides the index of the parameter with highest index present in the derivation transformation box. Default value is 0 (i.e. no parameter is present).

NOTE 1 The index of a parameter is given by its order of declaration in derivation transformation definition. The first parameter starts with index value to 1.

`parameter_defined_flags` is a bit-mask that indicates which parameter is present in the derivation transformation box. If the N^{th} bit is set to 1, the N^{th} parameter shall be present, otherwise if the N^{th} bit is set to 0, the N^{th} parameter shall not be present. Default value is 0.

NOTE 2 When there is no parameter present in the derivation transformation box, `highest_param_idx` and `parameter_defined_flags` can be omitted (box size equals to empty box size would indicates there is no attribute in the box).

`highest_input_idx` provides the index of the input with the highest index present in the `VisualDerivationInputs` box. Default value is 0 (i.e. no input is present).

NOTE 3 When there is no input present in the box, the `VisualDerivationInputs` box can be omitted.

NOTE 4 The index of an input is given by its order of declaration in the derivation transformation definition. The first input starts with index value to 1.

`input_present_flags` is a bit-mask that indicates which input is present in the box. If the N^{th} bit is set to 1, the N^{th} input shall be present in the array `reference_index`, otherwise if the N^{th} bit is set to 0, the N^{th} input shall not be present in the array `reference_index`. Default value is 0.

`reference_index` is an array (of size equals to the number of bits set to 1 in `input_present_flags`) that provides the N^{th} input to the derivation operation when the N^{th} bit of `input_present_flags` is set to 1 as described below:

— When the value is less than 0x8000, the value is the 1-based index to the `TrackReferenceTypeBox` of type `'dtrk'` that designates the ID of the input track or input image item to use as input.

— When the value is more than 0x8000, the value minus 0x8000 provides the 1-based relative index to the preceding derivation operation whose visual output is used as input.

— The value 0 indicates the default input fill picture as signalled in `DerivedVisualTrackConfigRecord`.

— Value 0x8000 is reserved.

The value of the M^{th} entry in `reference_index` corresponds to the M^{th} bit of `input_present_flags` having the value 1.

6 Sample entry and configuration definition

6.1 Sample entry definition

The sample entry for a derived visual track is defined as follows:

```
aligned(8) class DerivedVisualSampleEntry
extends VisualSampleEntry ('dtrk'){
   DerivedVisualTrackConfigRecord   derived_config;   // mandatory
}
```

The sample entry documents all the possible derivation operations used in the derived samples of the derived visual track. Derivation operations that occur in derived samples shall have a matching entry in the configuration record box. Readers may discard any other derivation operations.

6.2 Derived visual track configuration record

6.2.1 Definition

Box Type: `DerivedVisualTrackConfigRecord`
Container: `DerivedVisualSampleEntry`
Mandatory: Yes, in a track containing a `DerivedVisualSampleEntry`
Quantity: One

When `derivation_method` is equal to 1, the derived visual track shall include one `TrackReferenceTypeBox` with `reference_type` equal to `'ctln'` indicating one input track providing the composition timeline for this derived visual track.

A derivation operation may supply in its definition the value of the `default_derivation_input` attribute to define its own default input.

6.2.2 Syntax

```
aligned(8) class VisualDerivationDefault()
extends FullBox ('dtrD', version = 0, flags = 0) {
   unsigned int(2) default_derivation_input;
   unsigned int(3) derivation_method;
   unsigned int(3) reserved;
}
aligned(8) class DerivedVisualTrackConfigRecord() extends Box ('dtrC') {
   VisualDerivationDefault   derived_default_config;
   VisualDerivation          derivation_operations[];   // to fill box
}
```

6.2.3 Semantics

`default_derivation_input` indicates the default visual input. Value 0 corresponds to a full black picture, value 1 corresponds to a full white picture and value 2 corresponds to a mid-grey picture. Value 3 is reserved.

`derivation_method` indicates the derivation method to be used for determining the timing and number of output video frames relatively to the composition timeline (i.e. before the application of the edit lists of the derived visual track).

— When equal 0, the timing and number of output video frames correspond to the temporal combination of samples of any of all input tracks or of the derived visual track itself (default behaviour). (That is, there is conceptually a new output video frame at least whenever any input changes).

— When equal 1, the timing and number of output video frames correspond to the temporal combination of samples of the input track provided by track reference of type `'ctln'`.

— When equal 2, the timing and number of output video frames are aligned with the samples of the derived visual track.

— When equal 3, the timing and number of output video frames is not mandated and are derived by the client (e.g. it matches the display refresh rate).

— Other values are reserved.

NOTE The derived visual track can have edit lists whatever is the derivation method. When present, they provide the timing and number of output video frames of the derived visual track on the presentation timeline. If the derived visual track does not have an edit-list then its presentation timeline is equal to its composition timeline.

7 Sample format

7.1 General

The sample format for the sample entry `DerivedVisualSampleEntry` consists of a set of derivation operations to be applied in sequence, starting with the first in the sample.

7.2 Syntax

```
aligned(8) class DerivedSample(){
   VisualDerivation   derivation_operations[];   // to fill sample
}
```

8 Derivation transformations

8.1 Overview

Table 1 gives the list of basic derivation transformations defined by this document. Further derivation transformations may be defined in derived specifications or added in future editions. The four-character codes of derivation transformations can and should be registered as defined in ISO/IEC 14496-12. Vendor-specific derivation transformation may also be defined by using the code `'uuid'` and an extended type UUID identifying the vendor-specific derivation transformation.

Examples of derivation operations usage are illustrated in Annex A.

Table 1 — List of basic derivation transformations

Operation	Inputs	Parameters
Identity	1	(none)
sRGB Fill	0	1. Red fill 2. Green fill 3. Blue fill 4. Opacity 5. Output width 6. Output height
Dissolve	2 (from, to)	1. Initial proportion 2. Final proportion
Crop	1	1. Width numerator 2. Width denominator 3. Height numerator 4. Height denominator 5. X offset numerator 6. X offset denominator 7. Y offset numerator 8. Y offset denominator
Rotate	1	1. Angle (0°, 90°, 180°, 270°)
Mirror	1	1. Horizontal or Vertical mirroring

Table 1 *(continued)*

Operation	Inputs	Parameters
Scaling	1	1. Target width numerator 2. Target width denominator 3. Target height numerator 4. Target height denominator
Region-of-interest	2 (visual track, ROI meta-data)	(none)
Grid composition	N	1. Output width 2. Output height 3. Rows 4. Columns
Overlay composition	2 (input, backdrop)	1. horizontal_offset 2. vertical_offset

8.2 Identity

8.2.1 Definition

Box Type: `'idtt'`
Mandatory (per sample): No
Quantity (per sample): Zero or one
Inputs: One

The `Identity` derivation transformation reproduces the visual input. When a derivation operation with the `Identity` derivation transformation is present in a derived sample, no other derivation operations should be present in the same derived sample.

8.2.2 Syntax

```
aligned(8) class Identity()
extends VisualDerivationBase ('idtt', flags) {
}
```

8.3 sRGB Fill

8.3.1 Definition

Box Type: `'cfil'`
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: None

The `SRGBFill` derivation transformation generates a visual output of a single colour with visual size `output_width` and `output_height` pixels.

8.3.2 Syntax

```
aligned(8) class SRGBFill
extends VisualDerivationBase ('cfil', flags) {
   unsigned int(16)    red_fill_value      = 0;                    // parameter 1
   unsigned int(16)    green_fill_value    = 0;                    // parameter 2
   unsigned int(16)    blue_fill_value     = 0;                    // parameter 3
   unsigned int(16)    opacity_value       = 65535;                // parameter 4
```

```
   unsigned int(32)    output_width         = sample entry width;    // parameter 5
   unsigned int(32)    output_height        = sample entry height;   // parameter 6
}
```

8.3.3 Semantics

`red_fill_value` indicates the pixel value for the red channel according to sRGB colour space as defined in IEC 61966-2-1.

`green_fill_value` indicates the pixel value for the green channel according to sRGB colour space as defined in IEC 61966-2-1.

`blue_fill_value` indicates the pixel value for the blue channel according to sRGB colour space as defined in IEC 61966-2-1.

`opacity_value` indicates the opacity value ranging from 0 (fully transparent) to 65535 (fully opaque).

`output_width`, `output_height` specify the width and height, respectively, of the visual output in pixels.

8.4 Dissolve

8.4.1 Definition

Box Type:	`'dslv'`
Mandatory (per sample):	No
Quantity (per sample):	Any
Inputs:	(1) starting visual input A, (2) ending visual input B

The `Dissolve` derivation transformation provides smooth blending of two visual inputs gradually fading from the first visual input to the second visual input. The visual output co-located pixel value, `O(x,y)` is computed by the weighted summation of the two visual inputs where the weights are time-based transitions of the proportions provided by `start_weight` and `end_weight`. In the following equations, `T` transitions linearly from 0 at the time of the derived sample start to 1 at the time of the derived sample end.

The sizes of the visual inputs are normalized to the size of the larger one before the operation.

```
O(x,y) = A(x,y) * (T * (end_weight - start_weight) + start_weight) / 256 +
         B(x,y) * ((1-T) * (end_weight - start_weight) + start_weight) / 256,
```

where '/' is a division by truncation. `O(x,y)` is saturated to the dynamic range of the pixel values.

8.4.2 Syntax

```
aligned(8) class Dissolve
extends VisualDerivationBase ('dslv', flags) {
   unsigned int(8) end_weight     = 1;     // parameter 1
   unsigned int(8) start_weight   = 255;   // parameter 2
}
```

8.4.3 Semantics

`end_weight`, `start_weight` A value between 1 and 255 that gives the initial and final weights to be multiplied to collocated input pixels. Default value for `end_weight` is 1, and default value for `start_weight` is 255. The value 0 is reserved.

NOTE The identity derivation transformation can be used in a preceding or following derived sample, if a copy of either visual input is needed.

8.5 Crop

8.5.1 Definition

Box Type: `'crop'`
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: One

The `SampleCrop` derivation transformation defines a cropping transformation of the visual input.

8.5.2 Syntax

```
aligned(8) class SampleCrop
extends VisualDerivationBase ('crop', flags) {
   unsigned int(32) cleanApertureWidthN = width of the visual input;
                                                  // parameter 1
   unsigned int(32) cleanApertureWidthD = 1;      // parameter 2

   unsigned int(32) cleanApertureHeightN = height of the visual input;
                                                  // parameter 3
   unsigned int(32) cleanApertureHeightD = 1;     // parameter 4

   unsigned int(32) horizOffN     = 0;            // parameter 5
   unsigned int(32) horizOffD     = 1;            // parameter 6

   unsigned int(32) vertOffN      = 0;            // parameter 7
   unsigned int(32) vertOffD      = 1             // parameter 8
}
```

8.5.3 Semantics

The semantics of the syntax elements within the `SampleCrop` derivation transformation are the same as those specified for the syntax elements of `CleanApertureBox` as defined in ISO/IEC 14496-12.

8.6 Rotation

8.6.1 Definition

Box Type: `'srot'`
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: One

The `SampleRotation` derivation transformation rotates the visual input in anti-clockwise direction in units of 90 degrees.

8.6.2 Syntax

```
aligned(8) class SampleRotation
extends VisualDerivationBase ('srot', flags) {
   unsigned int (6) reserved = 0;  // not a parameter
   unsigned int (2) angle = 0;    // parameter 1
}
```

8.6.3 Semantics

`angle` * 90 specifies the angle (in anti-clockwise direction) in units of degrees.

8.7 Mirror

8.7.1 Definition

Box Type: 'smir'
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: One

The SampleMirror derivation transformation mirrors the visual input about either a vertical or horizontal axis.

8.7.2 Syntax

```
aligned(8) class SampleMirror
extends VisualDerivationBase ('smir', flags) {
   unsigned int (7) reserved = 0;   // not a parameter
   unsigned int (1) axis = 0;    // parameter 1
}
```

8.7.3 Semantics

axis specifies how the mirroring is performed: 0 indicates that the top and bottom parts of the image are exchanged (i.e. the mirroring is vertical); 1 specifies that the left and right parts are exchanged (i.e. the mirroring is horizontal).

8.8 Scaling

8.8.1 Definition

Box Type: 'sscl'
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: One

The SampleScaling derivation transformation scales the visual input to a target size.

The computation of the target size from the syntax elements is the same as the one specified for ImageScaling as defined in ISO/IEC 23008-12.

8.8.2 Syntax

```
aligned(8) class SampleScaling
extends VisualDerivationBase ('sscl', flags) {
   unsigned int (16) target_width_numerator      = 1;    // parameter 1
   unsigned int (16) target_width_denominator    = 1;    // parameter 2
   unsigned int (16) target_height_numerator     = 1;    // parameter 3
   unsigned int (16) target_height_denominator   = 1;    // parameter 4
}
```

8.8.3 Semantics

The semantics of the syntax elements within the SampleScaling derivation transformation shall be as those specified for the syntax elements of ImageScaling defined in ISO/IEC 23008-12.

8.9 Region of interest (ROI) selection

8.9.1 Definition

Box Type: '2dcc'
Mandatory (per sample): No

Quantity (per sample): Any
Inputs: (1) visual track, (2) ROI timed metadata track

The number of inputs shall be equal to 2 for this derivation transformation, and the inputs for this derivation transformation shall be a visual track and a ROI timed metadata track carrying 2D cartesian coordinates as defined in ISO/IEC 23001-10.

The `ROISelection` derivation transformation uses the 2D Cartesian coordinates carried in the second input track to crop the samples of the first input track. In other words, the ROI timed metadata track carrying 2D cartesian coordinates is applied prescriptively to the visual track that is the input for the ROI selection derivation transformation. The visual output of the derivation transformation contains only the rectangle specified by the 2D cartesian coordinates of the ROI timed metadata track.

8.9.2 Syntax

```
aligned(8) class ROISelection
extends VisualDerivationBase ('2dcc', flags) {
}
```

8.10 Grid composition

8.10.1 Definition

Box Type: `'gdcp'`
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: Exactly rows*columns. First visual input is the top-left cell of the grid, last visual input is the bottom-right cell of the grid.

The `GridComposition` derivation transformation provides a composition of visual inputs in a given grid order.

The visual inputs are inserted in row-major order, top-row first, left to right, in the order they are listed as the derivation operation inputs. The number of visual inputs shall be equal to `rows*columns`. All visual inputs (possibly after being resized) shall have exactly the same width and height, `width` and `height`. The visual inputs, when composed together, completely "cover" the visual output of the transformation according to the grid, where `width*columns` is equal to `output_width` and `height*rows` is equal to `output_height`. In other words, the visual output of the transformation is formed by tiling the visual inputs into a grid with a column width equal to `width` and a row height equal to `height`, without any gap or overlap.

8.10.2 Syntax

```
aligned(8) class GridComposition
extends VisualDerivationBase ('gdcp', flags) {
   unsigned int(8)   rows_minus_one       = 0;                    // parameter 1
   unsigned int(8)   columns_minus_one    = 0;                    // parameter 2
   unsigned int(32)  output_width         = sample entry width;   // parameter 3
   unsigned int(32)  output_height        = sample entry height;  // parameter 4
}
```

8.10.3 Semantics

`output_width`, `output_height` specify the width and height, respectively, of the reconstructed visual output on which the visual inputs are placed.

`rows_minus_one`, `columns_minus_one` specify the number of rows and columns in the grid; the value is one less than the number of rows or columns respectively. Visual inputs populate the top row first, followed by the second and following, in the listing order of the derivation operation inputs.

8.11 Overlay composition

8.11.1 Definition

Box Type: 'sovl'
Mandatory (per sample): No
Quantity (per sample): Any
Inputs: (1) overlay visual input (2) backdrop visual input

The `OverlayComposition` derivation transformation provides a composition of a visual input over another visual input representing the backdrop.

The overlay visual input is copied over the backdrop visual input at the `horizontal_offset, vertical_offset` offsets. The size of the reconstructed visual output is equal to the size of the backdrop visual input. Pixels of the overlay visual input that end up outside the backdrop visual input size by the copy operation are cropped. When a visual input has an associated alpha plane, alpha blending shall be performed on the visual input.

A chain of derivation operations can be used to create a sequence of multiple `OverlayComposition` derivation transformations. This allows overlaying multiple visual inputs over an initial backdrop visual input in sequence order. The backdrop visual input of the first `OverlayComposition` derivation transformation in the sequence represents the initial backdrop visual input. In subsequent `OverlayComposition` derivation transformation in the sequence, the backdrop visual input should designate the visual output of the preceding `OverlayComposition` derivation transformation in the sequence. Therefore, the first `OverlayComposition` derivation transformation in the sequence represents the overlay composition of the bottom-most overlay visual input over the initial backdrop visual input. The last `OverlayComposition` derivation transformation in the sequence represents the overlay composition of the top-most overlay visual input over the reconstructed visual output of preceding overlay derivation transformations in the sequence.

NOTE The visual output of a sRGB fill derivation transformation can be used as an initial backdrop visual input.

8.11.2 Syntax

```
aligned(8) class OverlayComposition
extends VisualDerivationBase ('sovl', flags) {
  signed int(32)  horizontal_offset   = 0;   // parameter 1
  signed int(32)  vertical_offset     = 0;   // parameter 2
}
```

8.11.3 Semantics

`horizontal_offset, vertical_offset` specify the offset, from the top-left corner of the backdrop visual input, to which the overlay visual input is located. Pixel locations with a negative offset value are not included in the reconstructed visual output. Horizontal pixel locations greater than or equal to the width of the backdrop visual input are not included in the reconstructed visual output. Vertical pixel locations greater than or equal to the height of the backdrop visual input are not included in the reconstructed visual output.