SMPTE STANDARD

Interoperable Master Format — Output Profile List — Common Image Definitions and Macros

Table of Contents

Approved
December 12, 2014

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 2067-101 was prepared by Technology Committee 35PM.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

This document specifies definitions, data structures and Macros applicable to image essence for use with the Interoperable Master Format (IMF) Output Profile List framework.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 377-1:2011, Material Exchange Format (MXF) — File Format Specification

Amendment 1:2012 to SMPTE ST 377-1:2011

Amendment 2:2012 to SMPTE ST 377-1:2011

SMPTE ST 2067-2:2013, Interoperable Master Format — Core Constraints

SMPTE ST 2067-100, Interoperable Master Format – Output Profile List

IEEE 754-2008, Standard for Floating-Point Arithmetic

IETF (2005, January). RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax

World Wide Web Consortium (W3C) (2004, October 28). XML Schema Part 1: Structures (Second Edition).

World Wide Web Consortium (W3C) (2004, October 28). XML Schema Part 2: Datatypes (Second Edition).

4 XML Schema Definitions

This section shall apply whenever a data structure is specified using XML schema definitions as specified in W3C XML Schema Part 1: Structures and W3C XML Schema Part 2: Datatypes.

In order to avoid duplication between text and schema, the cardinality and default values of elements are specified in the schema definitions only.

In the event of a conflict between schema definitions and the prose, the prose shall take precedence.

Note: The prefix associated with each namespace in an XML Schema definition document is arbitrary, and an Output Profile List instance can use other prefix value for the same namespace.

The target namespace of XML Schema definitions, i.e. the value of the `targetNamespace` attribute of the `schema` element, allows implementations to unambiguously identify the defining specification.

Specifications that modify the schema definitions or the semantics of the elements, including future versions of this specification, shall use a different target namespace and no two distinct schemas shall have the same target namespace.

5 Reference Image

For commonality across Macros operating on images, a Reference Image is defined.

Implementations need not implement support for Reference Image exactly as specified herein, but are expected to produce output substantially equivalent to what would have been produced by the operations against the reference image.

A reference image sequence shall consist of an ordered sequence of reference images, and, as illustrated in Figure 1, each reference image shall consist of a rectangular pixel array.

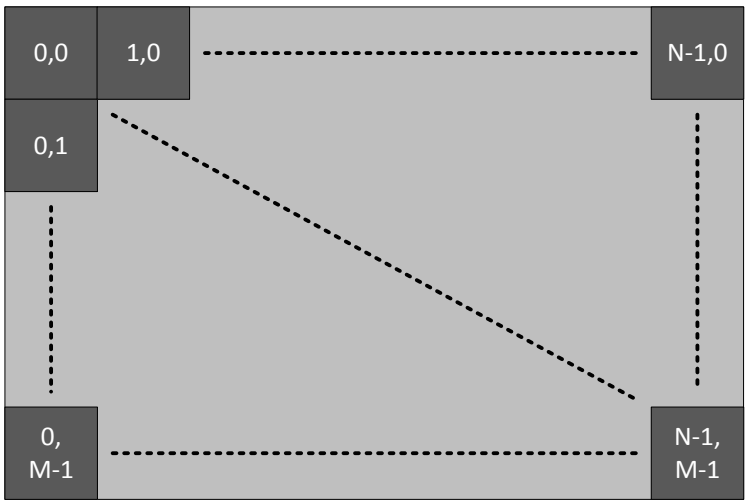

Figure 1 – Reference image

The pixel at the intersection of column i and row j of the rectangular array is denoted P(i, j), with P(0,0) denoting the topmost and leftmost pixel.

Each pixel P(i, j) shall consist of a 4-tuplet (P_1, P_2, P_3, P_4) of IEEE 754 binary32 values, as specified in IEEE 754

When mapping common image formats to Reference Images, P_1, P_2 and P_3 should each correspond to a component associated with a color primary, and P_4 to an alpha channel.

6 Pixel Color Schemes

For commonality across Macros operating on images, the following defines a framework for defining a pixel color scheme and representing a pixel color value in such a pixel color scheme. This specification does not specify concrete Pixel Color Schemes, which is left to other specifications

A Pixel Color Scheme shall specify the following:

- a URI (as specified in IETF RFC 3986) that uniquely and globally identifies the Pixel Color Scheme;
- a mapping from images using the Pixel Color Scheme to the Reference Image specified in Section 5;
- a mapping from Reference Image specified in Section 5 to images using the Pixel Color Scheme; and
- a structure for represented a single pixel in the Pixel Color Scheme, in the form of a concrete XML Schema type definition derived by extension from `ColorEncodingType`, as specified in Table 1.

Table 1 – ColorEncodingType schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-101/2014/color-
                    schemes"
  xmlns:oplc="http://www.smpte-ra.org/schemas/2067-101/2014/color-schemes"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType abstract="true" name="ColorEncodingType">
    <xs:complexContent>
     <xs:extension base="xs:anyType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

A Macro definition can then use ColorEncodingType as the type of an element intended to represent a pixel color value in any Pixel Color Scheme. An instance of such a Macro can then use a concrete pixel color as the value of the element.

EXAMPLE: The XML schema definition of a Macro wishing to represent a pixel color value using the `ColorEncodingType` can be declared as:

```
<xs:complexType name="SampleMacroType">
  <xs:complexContent>
    <xs:extension base="opl:MacroType">
      <xs:sequence>
<!-- other definitions -->
        <xs:element name="Color" minOccurs="0" type="oplc:ColorEncodingType"/>
<!-- other definitions -->
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

and a corresponding instance as:

```
<Macro xmlns:sm="smuri" xsi:type="sm:SampleMacroType">
<!-- other elements -->
  <sm:Color xsi:type="foo:REC709-RGB-8-ColorEncodingType">1 255 255</sm:Color>
<!-- other elements -->
</Macro>
```

where `REC709-RGB-8-ColorEncodingType` uniquely identifies the Pixel Color Scheme and the triplet `1 255 255` corresponds to a color in the Pixel Color Scheme.

7 Pixel Decoder Macro

7.1 Structure

The Pixel Decoder Macro maps source pixels to Reference Image pixels (see Section 5).

A Pixel Decoder Macro instance shall be a Macro instance with the type `PixelDecoderType` specified in Table 2.

Table 2 – PixelDecoderType schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace=" http://www.smpte-ra.org/schemas/2067-101/2014/pixel-
decoder"
  xmlns:pdec=" http://www.smpte-ra.org/schemas/2067-101/2014/pixel-decoder"
xmlns:opl="http://www.smpte-ra.org/schemas/2067-100/2014"
xmlns:dcml="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-100/2014" />
  <xs:import namespace="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/" />

  <xs:complexType name="PixelDecoderType">
    <xs:complexContent>
      <xs:extension base="opl:MacroType">
        <xs:sequence>
          <xs:element name="InputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
type="dcml:UserTextType"/>
                <xs:element name="Handle" type="opl:HandleType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="OutputReferenceImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
type="dcml:UserTextType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

7.1.1 InputImageSequence

7.1.1.1 Handle

The `Handle` element shall be a Handle that shall uniquely identify the input image sequence.

7.1.1.2 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the input images. It is intended to be displayed to the user.

7.1.2 OutputReferenceImageSequence

The `Annotation` element shall be a free-form, human-readable annotation describing the output reference image sequence. It is intended to be displayed to the user.

7.2 Processing Model

Each image of the input image sequence shall be transformed, in order, into an output reference image with the same rectangular pixel array dimensions. Each pixel of the input image shall be mapped into the

corresponding pixel of the output reference image according to the Pixel Color Scheme (see Section 6) of the input image sequence.

7.3 Output

A Pixel Decoder Macro instance defines a single output corresponding to the sequence of output reference images specified in Section 7.2. The Handle of the output in the context of the Macro instance shall be:

```
images
```

8 Pixel Encoder Macro

8.1 Structure

The Pixel Encoder Macro maps reference image pixels, as specified in 5, to encoded color pixels conforming to a scheme specified in the instance.

A Pixel Encoder Macro instance shall be a Macro instance with the type `PixelEncoderType` as specified in Table 3.

Table 3 – PixelEncoderType schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-101/2014/pixel-
encoder"
  xmlns:penc="http://www.smpte-ra.org/schemas/2067-101/2014/pixel-encoder"
xmlns:opl="http://www.smpte-ra.org/schemas/2067-100/2014"
xmlns:dcml="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-100/2014" />
  <xs:import namespace="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/" />
  <xs:complexType name="PixelEncoderType">
    <xs:complexContent>
      <xs:extension base="opl:MacroType">
        <xs:sequence>
          <xs:element name="InputReferenceImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
type="dcml:UserTextType"/>
                <xs:element name="Handle" type="opl:HandleType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="OutputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
type="dcml:UserTextType"/>
                <xs:element name="ColorEncoding" type="xs:anyURI"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

8.1.1 InputReferenceImageSequence

8.1.1.1 Handle

The `Handle` element shall be a Handle as specified in SMPTE ST 2067-100 and shall reference a sequence of Reference Images.

8.1.1.2 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the input reference images. It is intended to be displayed to the user.

8.1.2 OutputReferenceImageSequence

8.1.2.1 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the output image sequence. It is intended to be displayed to the user.

8.1.2.2 ColorEncoding

The `ColorEncoding` element is the color encoding of the output image pixels.

It shall be the URI of a Pixel Color Scheme conforming to Section 6.

8.2 Processing Model

Each image of the reference input image sequence shall be transformed, in order, into an output image with the same rectangular pixel array dimensions.

Each pixel of the input reference image shall be mapped into the corresponding pixel of the output image according to the image pixel color scheme specified by the `ColorEncoding` element.

8.3 Output

A Pixel Encoder Macro instance defines a single output corresponding to the sequence of output images specified in Section 8.2. The name of the output shall be:

```
images
```

9 Image Crop Macro

9.1 Structure

An Image Crop Macro instance shall be a Macro instance with the type `ImageCropMacroType` as specified in Table 4.

Table 4 – XML Schema root element definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-101/2014/crop-macro"
  xmlns:opl="http://www.smpte-ra.org/schemas/2067-100/2014"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  xmlns:icm="http://www.smpte-ra.org/schemas/2067-101/2014/crop-macro"
  xmlns:oplc="http://www.smpte-ra.org/schemas/2067-101/2014/color-schemes"
```

```
  xmlns:dcml="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-100/2014"/>
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-101/2014/color-
                  schemes"/>
  <xs:import namespace="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/" />

  <xs:complexType name="ImageCropMacroType">
    <xs:complexContent>
      <xs:extension base="opl:MacroType">
        <xs:sequence>
          <xs:element name="InputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Handle" type="opl:HandleType"/>
                <xs:element minOccurs="0" name="Annotation"
                  type="dcml:UserTextType"/>
                <xs:element name="ReferenceRectangle" type="icm:MXFRectangleEnum"/>
                <xs:element name="Inset" type="icm:RectanglePaddingType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="OutputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
                  type="dcml:UserTextType"/>
                <xs:element name="Padding" type="icm:RectanglePaddingType"/>
                <xs:element name="FillColor" type="oplc:ColorEncodingType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="RectanglePaddingType">
    <xs:sequence>
      <xs:element default="0" minOccurs="0" name="Left"
                  type="xs:nonNegativeInteger"/>
      <xs:element default="0" minOccurs="0" name="Right"
                  type="xs:nonNegativeInteger"/>
      <xs:element default="0" minOccurs="0" name="Top" type="xs:nonNegativeInteger"/>
      <xs:element default="0" minOccurs="0" name="Bottom"
                  type="xs:nonNegativeInteger"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="MXFRectangleEnum">
    <xs:restriction base="xs:token">
      <xs:enumeration value="Stored"/>
      <xs:enumeration value="Sampled"/>
      <xs:enumeration value="Display"/>
      <xs:enumeration value="Active"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

9.2 ImageCropMacroType

9.2.1 InputImageSequence

9.2.1.1 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the input images. It is intended to be displayed to the user.

9.2.1.2 Handle

The `Handle` element shall be a Handle as specified in SMPTE ST 2067-100 that reference the sequence of input images from which output images are generated.

9.2.2 ReferenceRectangle

The `ReferenceRectangle` element indicates a rectangle within the input image, as specified in Table 5.

Table 5 – MXF Rectangle Enum Values

Value	Meaning
Stored	Stored Rectangle as specified in SMPTE ST 377-1
Sampled	Sampled Rectangle as specified in SMPTE ST 377-1
Display	Display Rectangle as specified in SMPTE ST 377-1
Active	Active Area Rectangle as specified in SMPTE ST 2067-2

9.2.3 Inset

The `Inset` element specifies insets to be applied to the rectangle specified by the ReferenceRectangle element.

9.2.4 OutputImageSequence

9.2.4.1 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the output images. It is intended to be displayed to the user.

9.2.4.2 Padding

The `Padding` element specifies the padding to be applied to the pixels copied from the input image.

9.2.4.3 FillColor

The `FillColor` element indicates the background color of the output image.

It shall correspond to a Pixel Color Scheme conforming to Section 6.

9.3 Processing Model

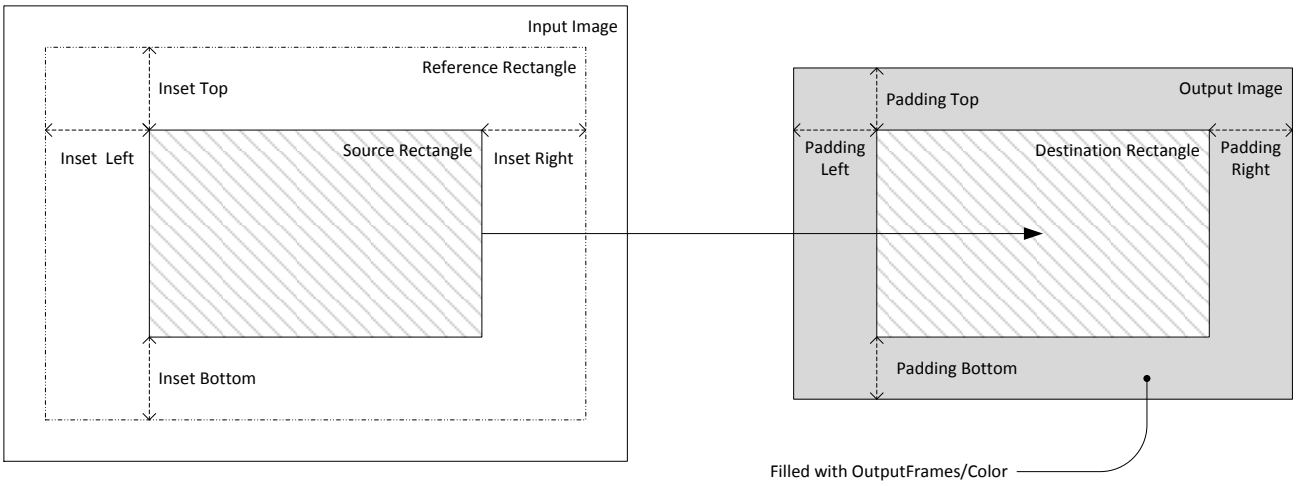

Figure 2 – Image Crop Macro

As illustrated in Figure 2, each input image in the sequence referenced by the `Handle` element (Section 9.2.1.2) shall be transformed, in the order in which they appear, into a single output image according to the following:

- All pixels within the Source Rectangle of the input image shall be copied without modification to the Destination Rectangle within the output image.
- The Source Rectangle shall be the rectangle specified by the `ReferenceRectangle` element (Section 9.2.2), inset by `Inset` (Section 9.2.3).
- The Destination Rectangle shall be a rectangle with width and height equal to that of the Source Rectangle.
- The stored, sampled, display rectangles of the output image shall be the destination rectangle padded by `Padding` (Section 9.2.4.2).
- The active rectangle of the output image shall be the destination rectangle.
- The pixels of the output image outside the destination rectangle shall be set to the color specified by the `FillColor` property (see Section 9.2.4.3).

Unless specified otherwise above, the characteristics of each output image, e.g. color space, shall be identical to the characteristics of the corresponding input image.

Note 1: Results are undefined if `FillColor` does not match the pixel encoding of input images.

Note 2: If none of the background of the output frame is exposed as part of the operation, the value of the `FillColor` element has no effect.

Note 3: No scaling is performed on the input frame. These and other transformations are left to other macros.

9.4 Output

An Image Crop Macro instance defines a single output corresponding to the sequence of output images specified in Section 9.3. The Handle of the output in the context of the Macro instance shall be:

```
images
```

10 Image Scale Macro

10.1 Structure

An Image Scale Macro instance shall be a Macro instance with the type `ImageScaleMacro` as specified in Table 6.

Table 6 – ImageScaleMacroType definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-101/2014/scale-
                  macro"
  xmlns:opl="http://www.smpte-ra.org/schemas/2067-100/2014"
  xmlns:ism="http://www.smpte-ra.org/schemas/2067-101/2014/scale-macro"
  xmlns:dcml="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-100/2014"/>
  <xs:import namespace="http://www.smpte-ra.org/schemas/433/2008/dcmlTypes/" />
  <xs:complexType name="ImageScaleMacroType">
    <xs:complexContent>
      <xs:extension base="opl:MacroType">
        <xs:sequence>
          <xs:element name="InputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
                  type="dcml:UserTextType"/>
                <xs:element name="Handle" type="opl:HandleType"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="OutputImageSequence">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" name="Annotation"
                  type="dcml:UserTextType"/>
                <xs:element name="Height" type="xs:positiveInteger"/>
                <xs:element name="Width" type="xs:positiveInteger"/>
                <xs:element ref="ism:BoundaryCondition"/>
                <xs:element name="Algorithm" type="ism:ScalingAlgorithmType">
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="BoundaryCondition">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="null"/>
```

```
        <xs:enumeration value="mirror"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:complexType abstract="true" name="ScalingAlgorithmType"/>
</xs:schema>
```

10.1.1 InputImageSequence

10.1.1.1 Handle

The `Handle` element shall be a Handle as specified in SMPTE ST 2067-100 and uniquely identify the sequence of reference input image frames.

10.1.1.2 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the input images. It is intended to be displayed to the user.

10.1.2 OutputImageSequence

10.1.2.1 Annotation

The `Annotation` element shall be a free-form, human-readable annotation describing the output images. It is intended to be displayed to the user.

10.1.2.2 Height

The `Height` element specifies the height of the output image frame.

10.1.2.3 Width

The `Width` element specifies the width of the output image frame.

10.1.2.4 BoundaryConditions

The `BoundaryConditions` element specifies filter behavior at the boundaries of the input image.

Given an input reference image frame with pixels $P^k(i,j)$, the value of the `BoundaryConditions` element shall determine the value of the pixel components $\hat{P}^k(i,j)$ according to Table 7.

Table 7 – Boundary Conditions

BoundaryConditions	$\hat{P}^k(i,j)$
mirror	$\hat{P}^k(i,j) = \begin{cases} P^k(\lvert i\rvert, \lvert j\rvert) \text{if } i < N \text{ and } j < M \\ P^k(\lvert i\rvert, 2M - j - 1) \text{if } i < N \text{ and } j \geq M \\ P^k(2N - i - 1, \lvert j\rvert) \text{if } i \geq N \text{ and } j < M \\ P^k(2N - i - 1, 2M - j - 1) \text{if } i \geq N \text{ and } j \geq M \end{cases}$
null	$\hat{P}^k(i,j) = \begin{cases} P^k(i,j) \text{if } 0 \leq i < N \text{ and } 0 \leq j < M \\ 0 \text{ otherwise} \end{cases}$

10.1.2.5 Algorithm

The `Algorithm` element specifies the algorithm and any associated parameters used to generate the output image sequence.

10.2 Processing Model

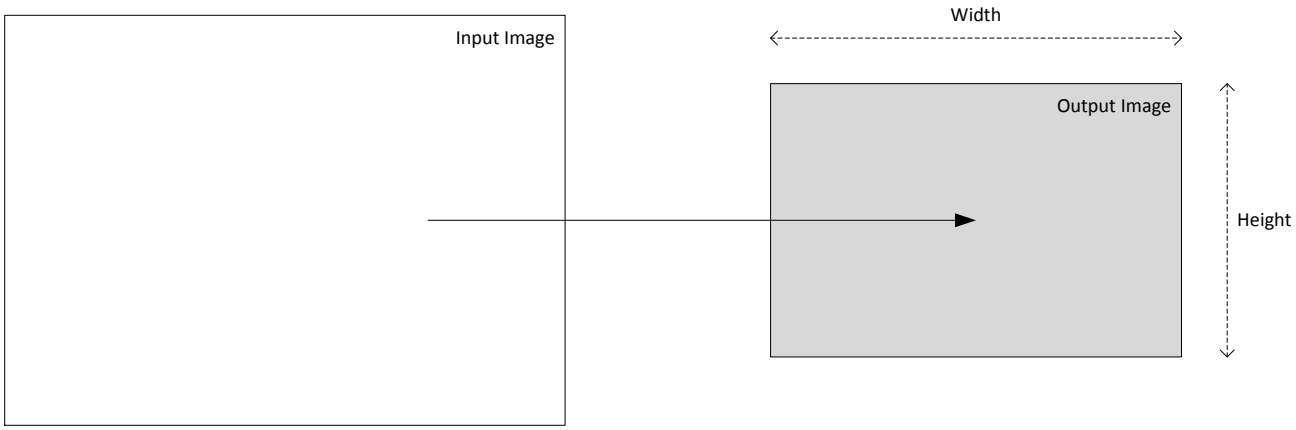

Figure 3 – Image Scale Macro

As illustrated in Figure 3, each input reference image frame in the sequence identified by the `Handle` element (Section10.1.1.1) shall be transformed, in the order in which they appear, into a single output reference image frame, as specified in Section 5, with horizontal and vertical dimensions specified by the `Height` and `Width` elements.

The type of the `Algorithm` element shall determine the scaling algorithm.

Annex A specifies Lanczos scaling. Other specifications may define additional scaling algorithms by defining for each an XML Schema type derived from `ScalingAlgorithmType`.

Note: This macro does not perform cropping of the input frame. This and other transformations are left to other macros.

10.3 Output

An Image Scale Macro instance defines a single output corresponding to the sequence of output images specified in Section 10.3. The name of the output shall be:

```
images
```

Annex A Lanczos Scaling (Normative)

This Annex specifies a scaling algorithm for use with the Image Scale Macro specified in Section 10.

The LanczosType type shall be specified by Table A.1.

Table A.1 – LanczosType definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.smpte-ra.org/schemas/2067-101/2014/lanczos"
  xmlns:lanc="http://www.smpte-ra.org/schemas/2067-101/2014/lanczos"
  xmlns:ism="http://www.smpte-ra.org/schemas/2067-101/2014/scale-macro"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.smpte-ra.org/schemas/2067-101/2014/scale-macro"/>
  <xs:complexType name="LanczosType">
    <xs:complexContent>
      <xs:extension base="ism:ScalingAlgorithmType">
        <xs:attribute name="parameterA" type="xs:positiveInteger"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

Let L(x) be the Lanczos kernel, with

$$\mathrm{L}(x) = \begin{cases} \mathrm{sinc}(\pi x)\mathrm{sinc}\left(\frac{\pi x}{a}\right) \text{ if } |x| < \mathrm{a} \\ 0 \text{ otherwise} \end{cases}$$

where a shall be the value of the `ParameterA` element, and

$$\mathrm{sinc}(x) = \frac{\sin(\mathrm{x})}{\mathrm{x}}$$

Note 1: The number of lobes of the Lanczos kernel is equal to $2a - 1$.

Note 2: Larger values of a generate sharper output, but tend to produce ringing artifacts around the edges.

The component $Q^k(i, j)$ of the output image frame shall be equal to

$$\mathrm{Q}^k(i,j) = \sum_{\mathrm{m}=-\infty}^{\infty} \sum_{\mathrm{n}=-\infty}^{\infty} \hat{P}^k(m,n)\, \mathrm{L}\big((\alpha \cdot i) - m\big)\, \mathrm{L}((\beta \cdot j) - n)$$

where $\hat{P}^k(m,n)$ is specified in Section 10.1.2.4, and α (β) is the ratio of the input image frame width (height) over the output image frame width (height).

Annex B Consolidated Schema (Informative)

This specification is accompanied by the following elements, each an XML schema document as specified in XML Schema Part 1: Structures.

st2067-101a-2014.xsd
st2067-101b-2014.xsd
st2067-101c-2014.xsd
st2067-101d-2014.xsd
st2067-101e-2014.xsd
st2067-101f-2014.xsd

These elements collect the XML schema definitions defined in this specification. They are informative and, in case of conflict, this specification takes precedence.