3MF Volumetric Extension

Specification & Reference Guide

Version	**0.2.0**
Status	Draft

Table of Contents

Preface

1.1. About this Specification

This 3MF volumetric specification is an extension to the core 3MF specification. This document cannot stand alone and only applies as an addendum to the core 3MF specification. Usage of this and any other 3MF extensions follow an a la carte model, defined in the core 3MF specification.

Part I, "3MF Documents," presents the details of the primarily XML-based 3MF Document format. This section describes the XML markup that defines the composition of 3D documents and the appearance of each model within the document.

Part II, "Appendixes," contains additional technical details and schemas too extensive to include in the main body of the text as well as convenient reference information.

The information contained in this specification is subject to change. Every effort has been made to ensure its accuracy at the time of publication.

This extension MUST be used only with Core specification 1.x.

Document Conventions

See the standard 3MF Document Conventions documentation.

Language Notes

See the standard 3MF Language Notes documentation.

Software Conformance

See the standard 3MF Software Conformance documentation.

Part I: 3MF Documents

Chapter 1. Overview of Additions

This document describes new elements, each of which is OPTIONAL for producers, but core features with some exceptions (see below, various types of 3d volumetric data types) MUST be supported by consumers that specify support for this volumetric extension of 3MF. Not all types of volumetric information make sense for all printing technologies and consumers may choose to ignore color, composite materials, or other properties that do not match their hardware. The proposed extensions enable the embedding of volumetric data within 3MF files. This will enable the representation of objects that are characterized by variable material properties throughout their volume, like opacity, color, strength etc.

Addition:

- explain how surface/volume properties work together
- Proposed language for volumetric / surface property language: (from Alan)

the

Volumetric content is always clipped to surface of the mesh that embedds it. If a property defined at the surface of an object conflicts with the property within the object defined by this extension, a surface layer should be defined with a thickness as small as possible to achieve the surface property on the outside of the object. Away from this thin surface region, the volumetric property should be applied everywhere within the object.

this may be tricky to implement!

The properties at surface regions that are not explicitly specified are given by the volumetric properties.

Part I: 3MF Documents

Chapter 1. Overview of Additions

I am not sure what this "MUST" means, as it does not refer to a specific item in the specification...

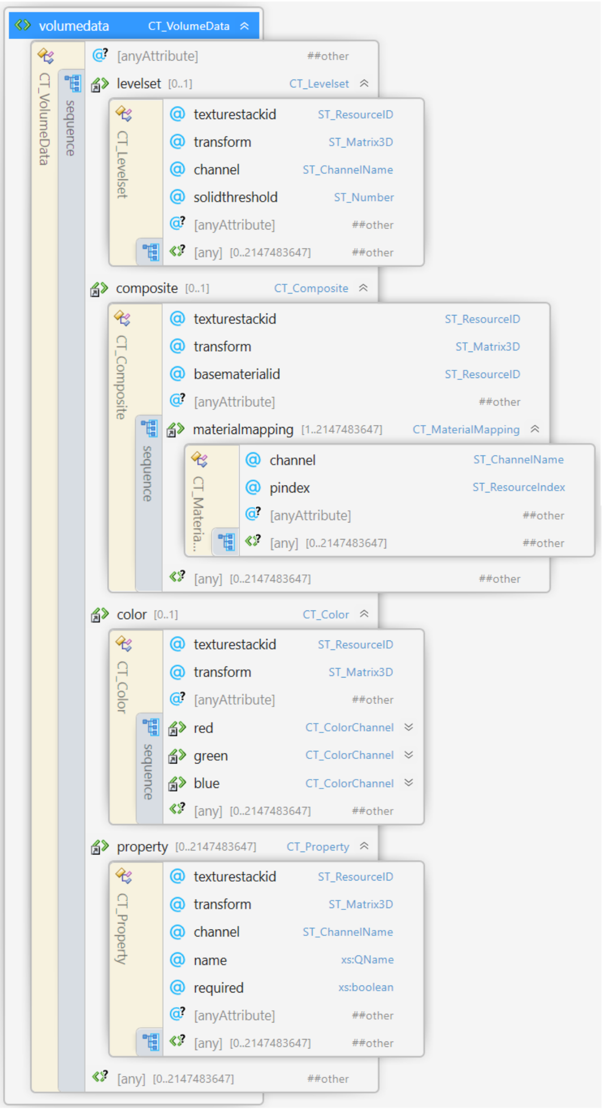
This document describes new elements, each of which is OPTIONAL for producers, but MUST be supported by consumers that specify support for this volumetric extension of 3MF.

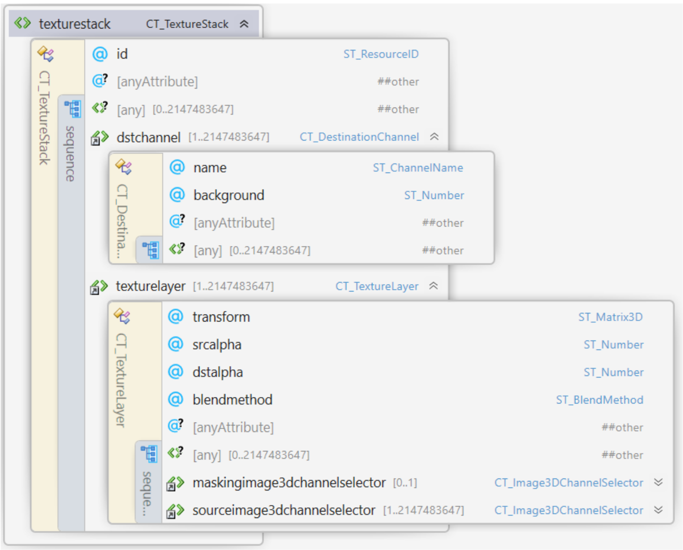
The central idea of this extension is to enrich the geometry notion of 3MF with volumetic elements that can represent spatially varying properties which are quite inefficient to handle with a mesh representation, especially in cases where the variation is continuous in space.

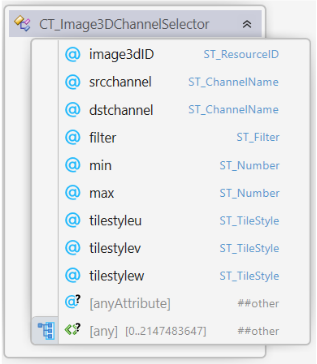
While this is meant to be an exact specification of geometric, material and in fact arbitary properties, and consumers MUST interpret it as such, the intent is also for applications in which editors can use the data structures for efficient interoperability and post processing the geometry in an intermediate step.

A producer using the volumetric specification MUST mark the extension as required, as described in the core specification.

Figure 2-1: Overview of model XML structure of 3MF with volumetric additions




texturestack CT_TextureStack
CT_TextureStack
id ST_ResourceID
[anyAttribute] ##other
sequence
[any] [0..2147483647] ##other
dstchannel [1..2147483647] CT_DestinationChannel
CT_Destina...
name ST_ChannelName
background ST_Number
[anyAttribute] ##other
[any] [0..2147483647] ##other
texturelayer [1..2147483647] CT_TextureLayer
CT_TextureLayer
transform ST_Matrix3D
srcalpha ST_Number
dstalpha ST_Number
blendmethod ST_BlendMethod
[anyAttribute] ##other
seque...
[any] [0..2147483647] ##other
maskingimage3dchannelselector [0..1] CT_Image3DChannelSelector
sourceimage3dchannelselector [1..2147483647] CT_Image3DChannelSelector



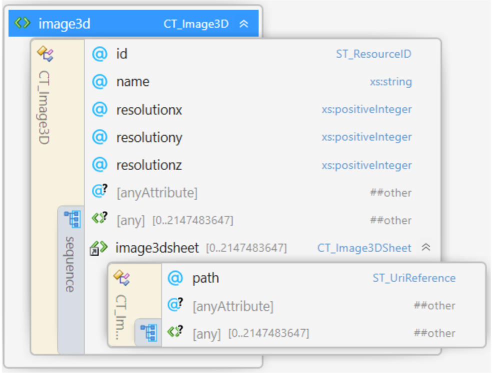
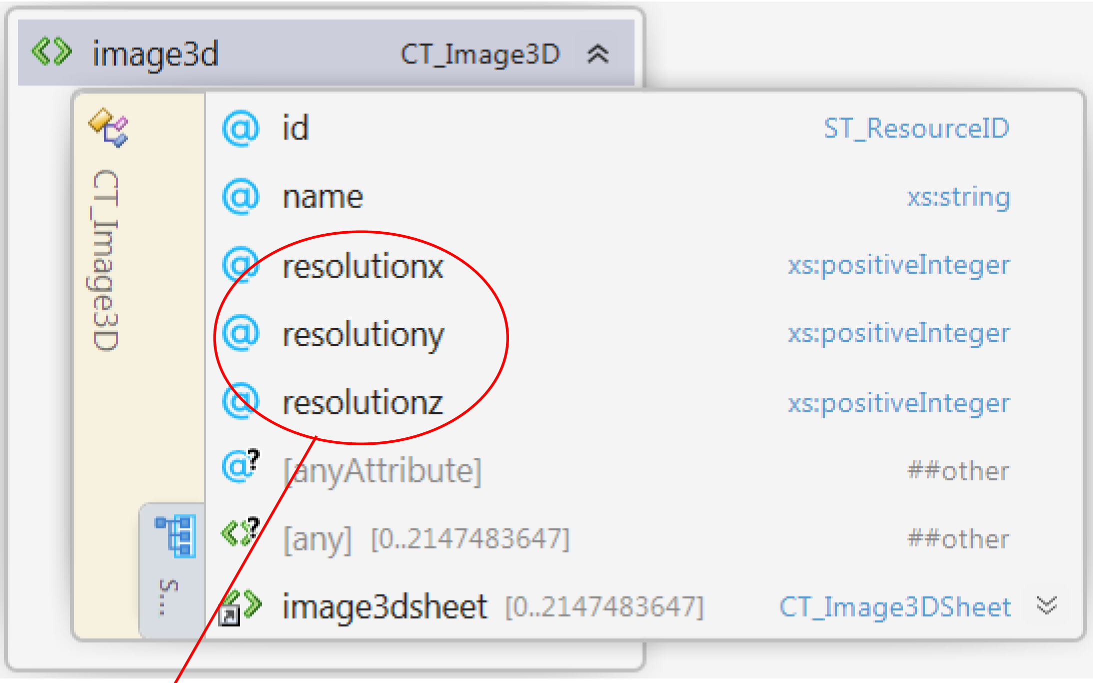


image3d CT_Image3D
CT_Image3D
id ST_ResourceID
name xs:string
resolutionx xs:positiveInteger
resolutiony xs:positiveInteger
resolutionz xs:positiveInteger
[anyAttribute] ##other
sequence
[any] [0..2147483647] ##other
image3dsheet [0..2147483647] CT_Image3DSheet
CT_Im...
path ST_UriReference
[anyAttribute] ##other
[any] [0..2147483647] ##other

Chapter 2. Additions to Resources

2.1 3D Image Resources

Element **<image3d>**



mismatch

Name	Type	Use	Annotation
id	ST_ResourceID	required	Specifies an identifier for this image3d resource.
name	xs:string	required	3d image resource name used for annotations purposes.
sizex	xs:positiveinteger	required	Size of all child <image3dsheet>-elements in first dimension in pixels.
sizey	xs:positiveinteger	required	Size of all child <image3dsheet>-elements in second dimension in pixels.
sheetcount	xs:positiveinteger	required	Number of <image3dsheet>-elements within this <image3d> element.

Volumetric data can be encoded as 3d images that consist of voxels. Each <image3d> element is assumed to represent a unit cube from which data can be sampled at any point. Volumetric images can be embedded inside a 3MF file using groups of PNG images that represent a stack of images.

All image3dsheets within an image3d MUST have the same x- and y-size that is specified in the sizex and sizey-attributes, respecitvely. sizex, sizey and sheetcount MUST not exceed 1024^3 each. There MUST be exactly sheetcount <image3dsheet>-elements under <image3d> that are implicitly ordered starting with index 0.

Yhis is GIGANTIC!

Image3D objects, and thus the underlying <image3dsheet> elements, SHOULD provide the channels "R", "G", "B" or "A". All image3dsheets within an image3d MUST provide the same channels, and each channel MUST have the same bit-depth accross all image3dsheets.

Specific rules apply if an image3dsheet does not provide these channels:

- If an image3dsheet does not provide a color channel "R", "B" or "G" but provides a greyscale channel, sampling any color channel will return the value of the greyscale channel.
- If an image3dsheet does not provide an alpha channel "A", sampling "A" will behave as if the image3dsheet contained a fully saturated alpha channel.

2.1.1 File Formats

PNG images can provide acceptable compression and bitdepth for the levelset-function, color information, material mixing ratios or arbitrary property information.

The following describes recomendations for the channel bit depth of PNG images used in this specification based on the nomenclature in the specification of PNG (https://www.w3.org/TR/PNG).

- A levelset-function can be deduced from an image with binary values, i.e. from images of image types "Greyscale" with bit-depth of 1 or an indexed-color with bit depths of 1.
- Color information, material mixing ratios and arbitrary properties can be deduced from PNG images with arbitrary color depth.

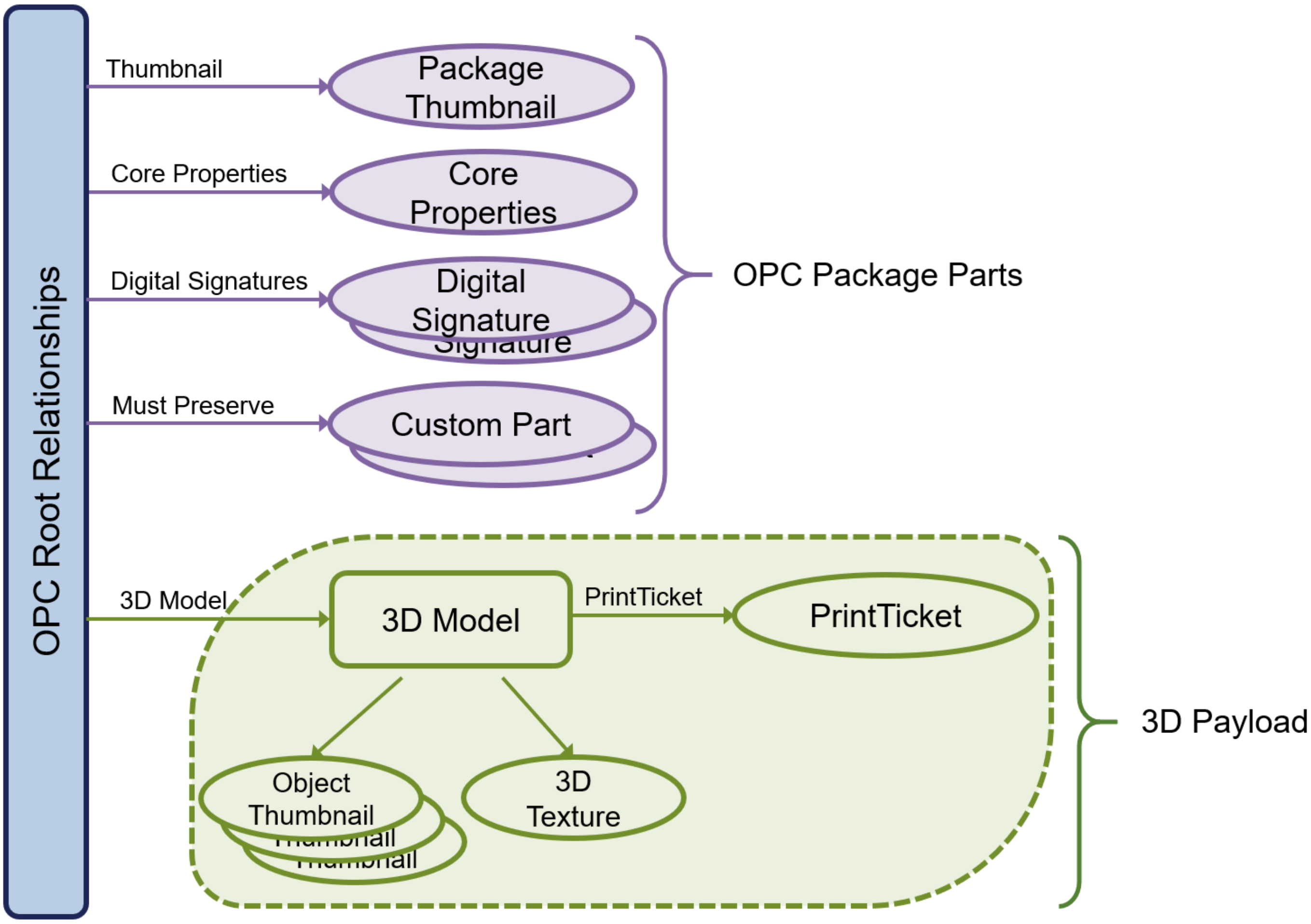
To achieve high accuracy, producers SHOULD store such information in image channels with bit depth of 16. Most professional image editing tools an standard implementations of the PNG format support channels with 16 bit.

2.1.2 OPC package layout

It is RECOMMENDED that producers of 3MF Documents use the following part naming convention:

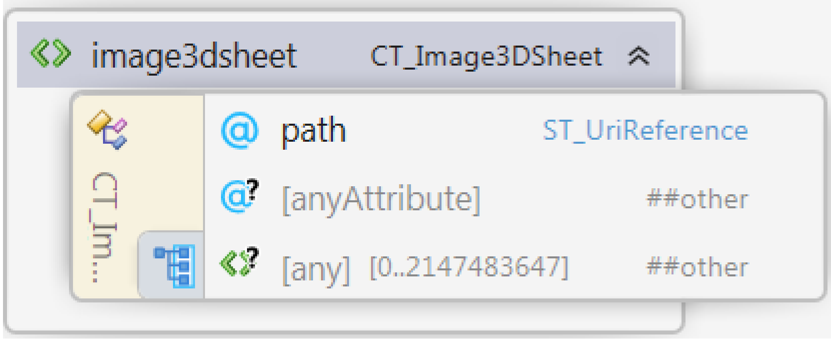
Paths of image3dsheet SHOULD consist four segments "/3D/volumetric/" as the first two segments, the name of a image3d-element that references this image3dsheet as third segment (for example "/3D/volumetric/mixingratios/", and the name of the image3dsheet as last segment (for example "sheet0001.png"). The 3D Texture part that is the image3dsheet MUST be associated with the 3D Model part via the 3D Texture relationship.

This implies that all image3dsheet parts for an image3d-object SHOULD be located in same OPC folder.



2.2 3D Image Sheet

Element **<image3dsheet>**



Name	Type	Use	Annotation
path	ST_UriReference	required	Specifies the OPC part name (i.e. path) of the image data file

Each <image3dsheet> element has one property which MUST be present. The path property determines the part name (i.e. path) of the 2D image data (see chapter 6 of the Materials & Properties Extension specification for more information).

2.3. 3D Image Channel Selector

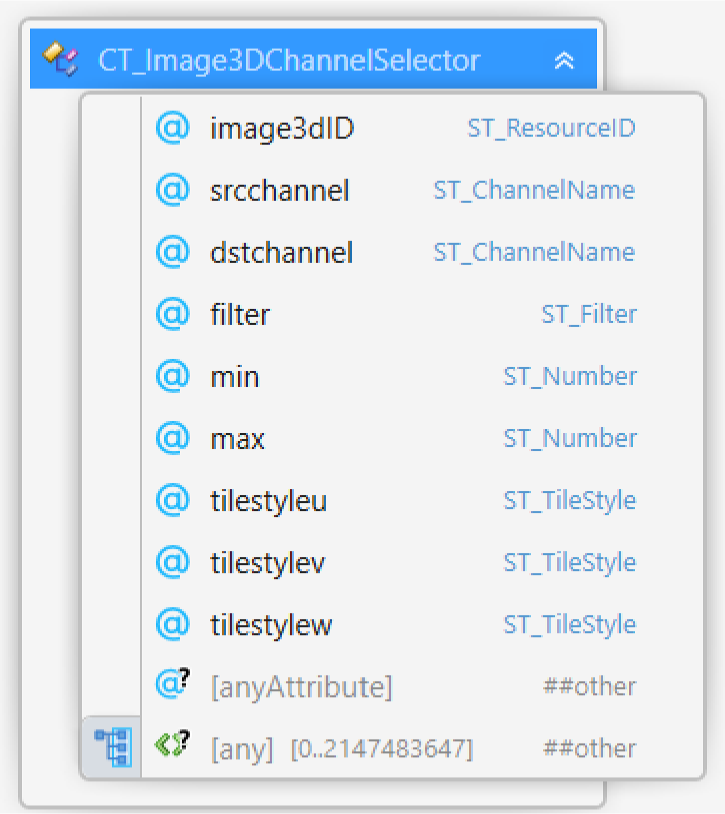
Elements **<sourceimage3dchannelselector>** and **<maskingimage3dchannelselector>**

sourceimage3dchannelselector CT_Image3DChannelSelector

maskingimage3dchannelselector CT_Image3DChannelSelector

of

Complex type **<CT_Image3DChannelSelector>**



CT_Image3DChannelSelector

image3dID	ST_ResourceID
srcchannel	ST_ChannelName
dstchannel	ST_ChannelName
filter	ST_Filter
min	ST_Number
max	ST_Number
tilestyleu	ST_TileStyle
tilestylev	ST_TileStyle
tilestylew	ST_TileStyle
[anyAttribute]	##other
[any] [0..2147483647]	##other

Name	Type	Use	Annotation
image3dID	ST_ResourceID	required	Specifies the id of the 3d image resource
srcchannel	ST_ChannelName	required	Specifies which channel to reference in the 3d image resource
dstchannel	ST_ChannelName	*srcchannel*	Specifies which channel the source channel should be mapped to during a sampling procedure. Will default to srcchannel if not given
minvalue	ST_Number	*0.0*	Specifies how the minimal possible value of the source channel is interpreted in the output.
maxvalue	ST_Number	*1.0*	Specifies how the maxmimal possible value of the source channel is interpreted in the output.
filter	ST_Filter	*linear*	"linear" or "nearest" neighbor interpolation
tilestyleu	ST_TileStyle	Required	Determines the behavior of the sampler for texture coordinate u outside the [0,1] range
tilestylev	ST_TileStyle	Required	Determines the behavior of the sampler for texture coordinate v outside the [0,1] range

Name	Type	Use	Annotation
tilestylew	ST_TileStyle	Required	Determines the behavior of the sampler for texture coordinate w outside the [0,1] range

Elements of type <CT_Image3DChannelSelector> define the way in which individual channels from volumetric image resources can be referenced inside the volumetric layer elements. Each channel reference MUST contain a resource id that maps to an actual <image3d> element.

In addition, the elements of type <CT_Image3DChannelSelector> MUST contain two string attributes which determine which channel to reference and how they should be mapped within the volumetric image. The channel name string can use any of the reserved channel names (i.e. "R", "G", "B", or "A").

For grayscale images "R", "G", and "B" are interchangeable and SHOULD always map to the same value. For images, which lack an alpha channel, any sampling operation on channel "A" should always return an alpha value of 1.0. The <image3d> resource MUST be defined before being referenced by an element of type <CT_Image3DChannelSelector> in the 3MF model document to simplify the parser.

tilestyle-u, -v or -w:

MUST be one of "wrap", "mirror", "clamp", and "none". This property determines the behavior of the sampler of this channel for 3d texture coordinates (u,v,w) outside the [0,1]x[0,1]x[0,1] cell. The different modes have the following interpretation (for s = u, s = v, or s = w):

1. "wrap" assumes periodic texture sampling. A texture coordinate s that falls outside the [0,1] interval will be transformed per the following formula:
   $s' = s - \text{floor}(s)$

   This gives an anomally, because the interval [0,1] is treated different than [3,4], for example. Snce it says above that it applies to "s that falls outside [0,1]", 1 maps to 1, but any other integer is mapped by this formula to zero.

2. "mirror" means that each time the texture width or height is exceeded, the next repetition of the texture SHOULD be reflected across a plane perpendicular to the axis in question following this formula:
   $s' = s$ ... TODO

3. "clamp" will restrict the texture coordinate value to the [0,1] range. A texture coordinate s that falls outside the [0,1] interval will be transformed according to the following formula:
   $s' = \min(1, \max(0,s))$

4. "none" will discard the channelselector's value if the 3d texture coordinate s falls outside the [0,1] range. This is useful if a 3d texture is used as a volumetric decal of sorts that affects only a limited region in the volume.

filter: The filter attribute defines the interpolation method.

- If the interpolation method of an elements of type <CT_Image3DChannelSelector> is "nearest", sampling it at an arbitrary (u,v,w) returns the floating point value defined by the closest point (u',v',w') to (u,v,w) which transforms back to a voxel center in the 3D image ressource.

- If the interpolation method of an elements of type <CT_Image3DChannelSelector> is "linear", sampling it at an arbitrary (u,v,w) returns the floating point defined by trilinearly interpolating between the eight closest points coordinates which transforms back to voxel centers in the 3D image ressource.

Function of instances of type <CT_Image3DChannelSelector>:

1. The referenced 3D Image gives a voxel grid of RGBA (RGB, Grey-Alpha, Grey) values distributed in a cuboid ([0..res_x] x [0..res_y] x [0..res_z]). The centers of each voxel (ix, iy, iz) are at the half integer positions (ix + 0.5, iy + 0.5, iz + 0.5).

2. The <CT_Image3DChannelSelector> selects one of those channels and gives integer values (between 0 and 2^bitdepth-1) at the half integer positions (ix + 0.5, iy + 0.5, iz + 0.5) with ix = 0..res_x – 1, iy = 0..res_y – 1, iz = 0..res_z – 1.

3. The **tilestyle** extends the voxel grid to infinity: they extend the pointwise defined pixel values to a mapping

   ρ:Z^3→Z through the rules 1-4 defined above.

The rules where defined for (continuous) floating point coordinates, and now we are talking about the discretization... Perhaps the standard should be more clear.

Also, notice that before the (integer) values of the texture where (correctly) assigned to centers of texels, but here the functions seem to be assigning integer values to integer coordinates (which is wrong).

4. The interpolation method defines a function on the full coordinate space. In mathematical terms, this defines a mapping

   ρ':R^3→Z, with values between 0 and 2^bitdepth-1.

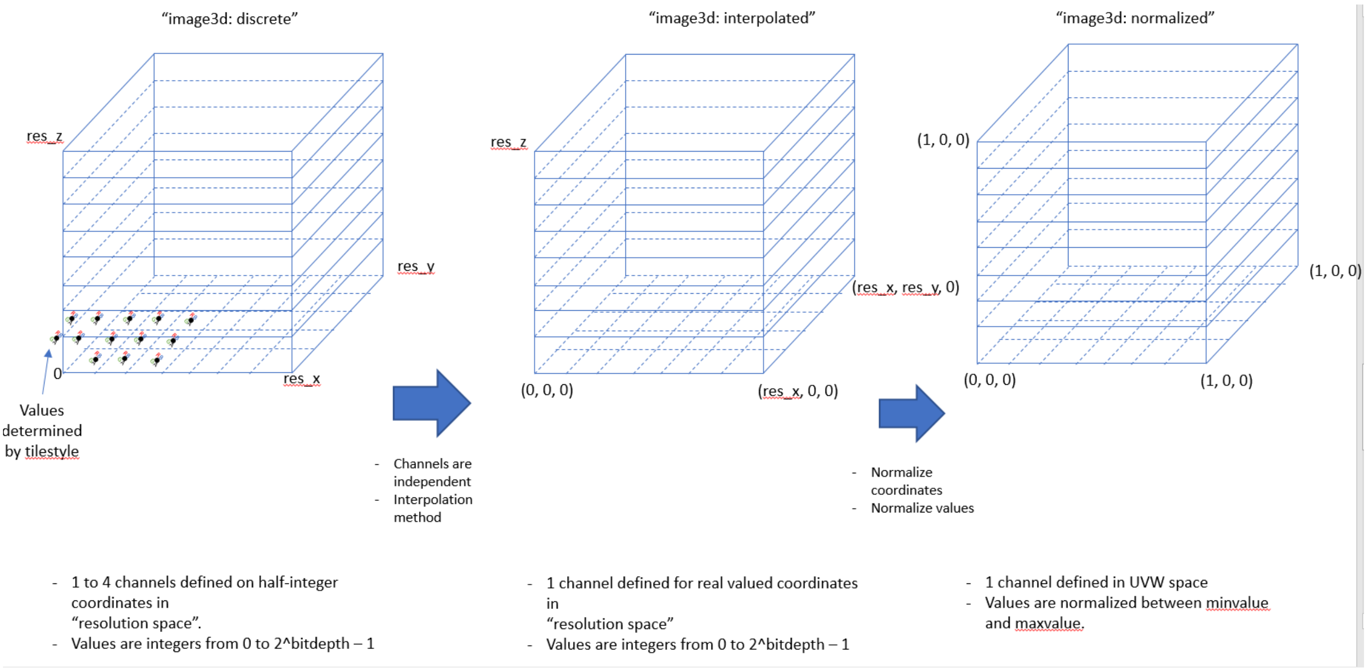
Again, teh image space should be R, not Z; for example all voxels store as value their x coordinate (an integer) and interpolation is linear, then at a point with u coordinate=sqrt(2), this should return sqrt(2)! Otherwise, aliasing,

5. In a final normalization step, the coordinates are mapped to the unit cube, and the values are mapped between the minvalue and maxvalue, thus giving a normalized function

   φ:R^3→R:

   (x,y,z) → minvalue + ρ'(x*res_x,y*res_y,z*res_z) *(maxvalue-minvalue)/(2^bitdepth-1)

The following image illustrates the channel selection process:



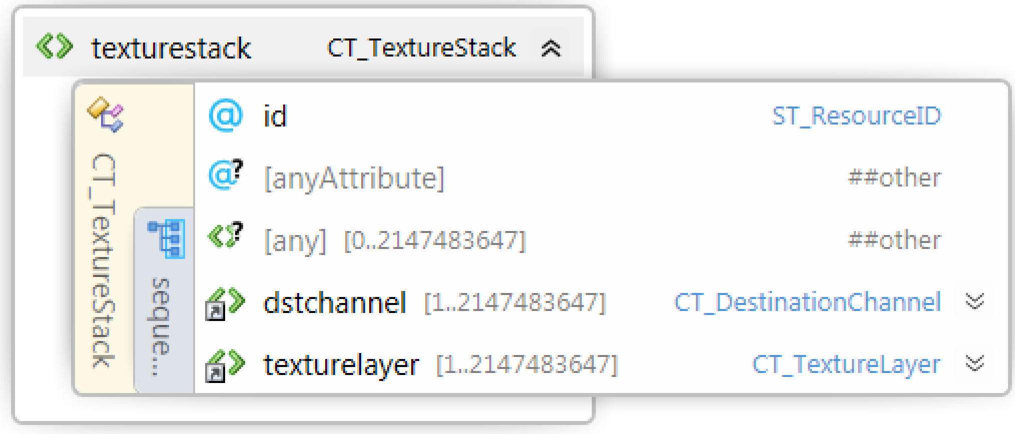
- 1 to 4 channels defined on half-integer coordinates in “resolution space”.
- Values are integers from 0 to 2^bitdepth – 1

- 1 channel defined for real valued coordinates in “resolution space”
- Values are integers from 0 to 2^bitdepth – 1

- 1 channel defined in UVW space
- Values are normalized between minvalue and maxvalue.

2.4. Volumetric Stack element

Element **<volumetricstack>**


texturestack
CT_TextureStack
CT_TextureStack
seque...
id ST_ResourceID
[anyAttribute] ##other
[any] [0..2147483647] ##other
dstchannel [1..2147483647] CT_DestinationChannel
texturelayer [1..2147483647] CT_TextureLayer

Name	Type	Use	Annotation
id	ST_ResourceID	required	Specifies the id of this volumetricstack

The volumetric stack is a resource within a 3MF model that defines how volumetric data from multiple <CT_Image3DChannelSelector> is composited to yield multiple custom scalar field (dstchannels) in 3d. This custom scalar field of a <volumetricstack> element can then be used to define volumetric properties inside the <volumedata>-element of an object, see . TODO

1. It defines multiple destination channels, <dstchannel>-elements. Each destinaton channel is a scalar field in 3d, whose values can be retrieved by sampling this volumetricstack.

2. The sampled values of each destination channel are built up by blending multiple layers, the <volumetriclayer>-elements. This allows e.g. boolean opeartions on the scalar fields provided by different <sourceimage3dchannelselector> elements.

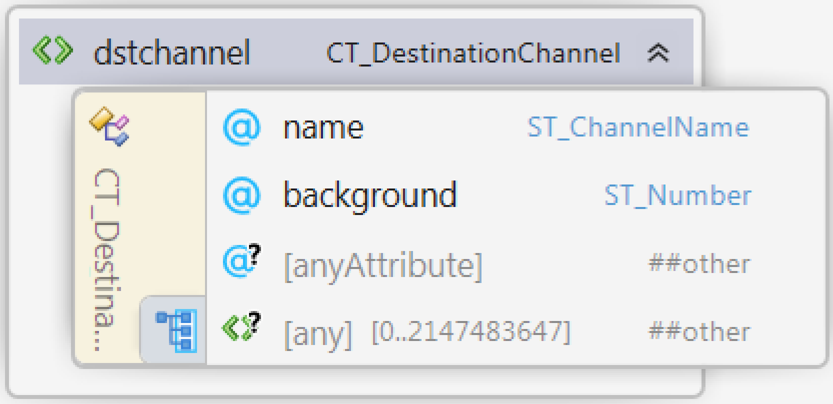
The volumetricstack element MUST contain at least one <dstchannel> child element and MUST NOT contain more 2^20 child-elements. The volumetricstack element MUST NOT contain more than 2^31-1 child-elements.

Which one!?
Again, a required minimum supported would be more fitting for a standard... These values seem exageratedsince they speak of a stack wiht a billion children!

Illustration of the composited value of 2 channels within a volumetricstack

2.4.1 Destination channel element

Element **<dstchannel>**



Name	Type	Use	Annotation

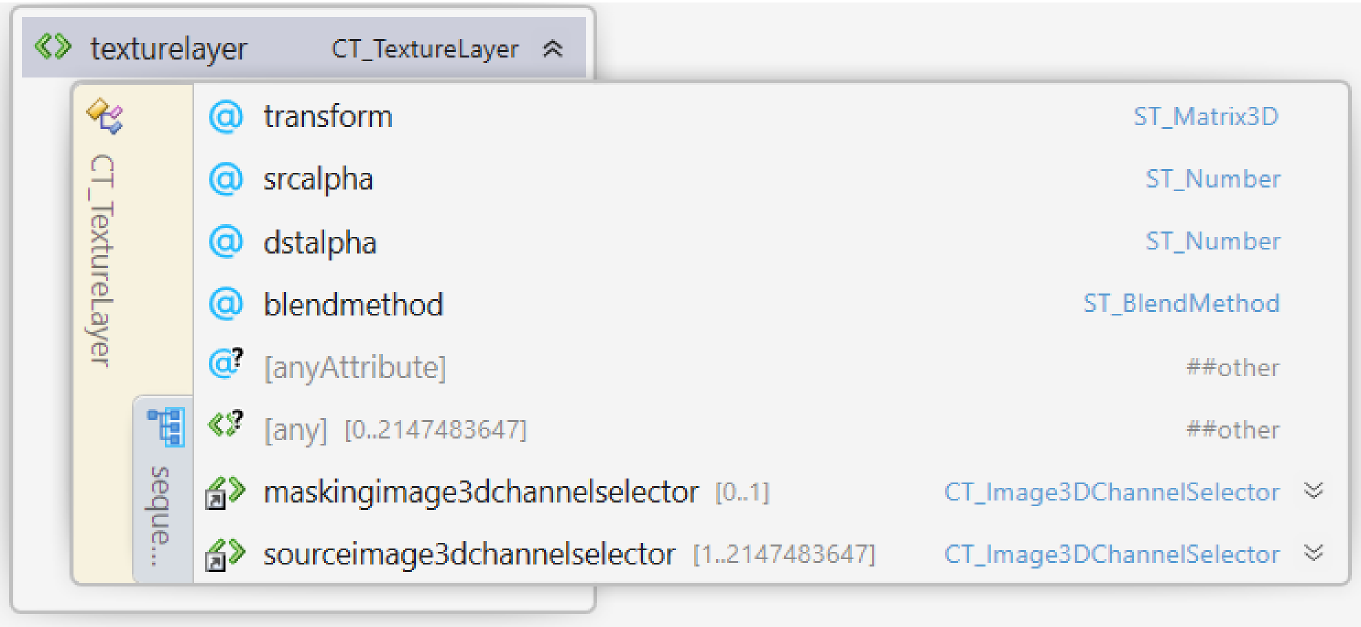
Name	Type	Use	Annotation
name	ST_ChannelName	required	Specifies the name of this destination channel
background	ST_Number	required	Specifies the background value of this channel

A destination channel specifies a name of a channel that can be sampled from a volumetricstack element. The background value is the value that serves as a base for the blending that takes place in the volumetriclayer elements within the <volumetricstack>-element.

The names of -elements must be unique within a <volumetricstack>-element.

2.4.2 Volumetric Layer element

Element **<volumetriclayer>**



Name	Type	Use	Annotation
transform	ST_Matrix3D	required	Transformation of the volumetricstack coordinate system into the volumetriclayer coordinate system
blendmethod	ST_BlendMethod	required	Determines how this layer is applied to its sublayers. Allowed values are "mix", "multiply" or "mask".
srcalpha	ST_Number	optional	Numeric scale factor [-1,1] for the source layer. Required if blendmethod is "mix".
dstalpha	ST_Number	optional	Numeric scale factor [-1,1] for the destination layer. Required if blendmethod is "mix".

Each -element modify the accumulated value of the destination channels of a volumetric stack. This modification is defined by the following attributes:

blendmethod: controls how the current layer (known as the source layer) is blended with the layers below it as well as with the stack's background value and potential overlapping objects. These functions either "add" or "multiply" the values of the destination layer with the corresponding values in the source layer.

Let "s" denote the value of the source channel, "d" the current value of the destination channel, then the modified value of the destination channel "d'" after blending is calculated according to the blendmethod:

- "mix":

  d' = src_alpha * s + dst_alpha * d

- "multiply":

  d' = s * d

- "mask":

  d' = m * s + (1 - m) * d

  Here, m is the value of the dstchannel of the <maskingimage3dchannelselector> element of this volumetriclayer. The blendmethod "mask" provides a means to use another 3d texture as a volumetric decal that only affects a region of complex shape within the volume.

srcalpha: is a scalar value that SHOULD be in the range [-1, 1] which is multiplied with the sampled values in the source layer during the blending process.

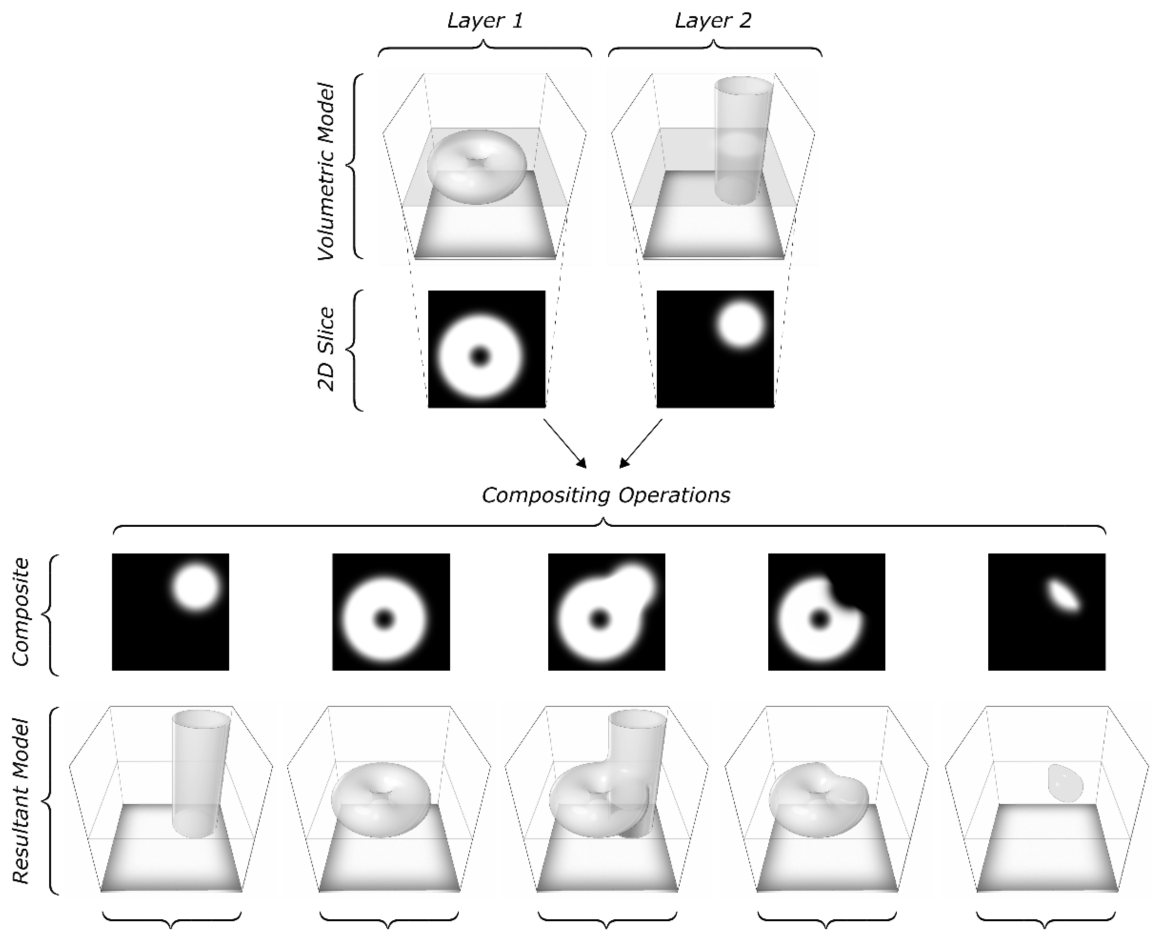
dstalpha: is a scalar value that SHOULD be in the range [-1, 1] which is multiplied with the sampled values in the destination during the blending process.

Figure 4-1 shows an example of two layers within a volumetric stack and the result using various blending functions with different source and destination alpha values.

If the blendmethod is "mask", a volumetriclayer MUST contain exactly one <maskingimage3dchannelselector> element.

A volumetriclayer MUST contain at least one <sourceimage3dchannelselector> element. The dstchannel attribute of the each <sourceimage3dchannelselector> within a volumetriclayer element MUST match a <dstchannel> element within this <volumetriclayer>. The name of each <dstchannel> element MUST occur at most once as dstchannel attribute in one of the <sourceimage3dchannelselector>.

Destination channels that are not mentioned in as dstchannel attribute in this list are not modifed by this <volumetriclayer>.

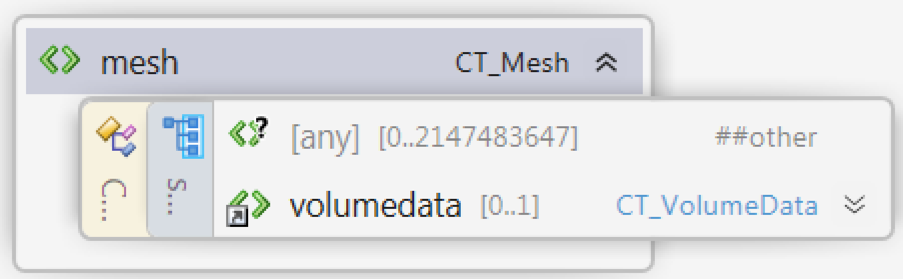


Compositing Function	Add	Add	Add	Add	Multiply
Source Layer Alpha	0.0	+1.0	+1.0	+1.0	+1.0
Destination Layer Alpha	+1.0	0.0	+1.0	-1.0	+1.0

Chapter 3 Additions to Mesh

3.1. Volumetric Data extension to Mesh

Element **<mesh>**

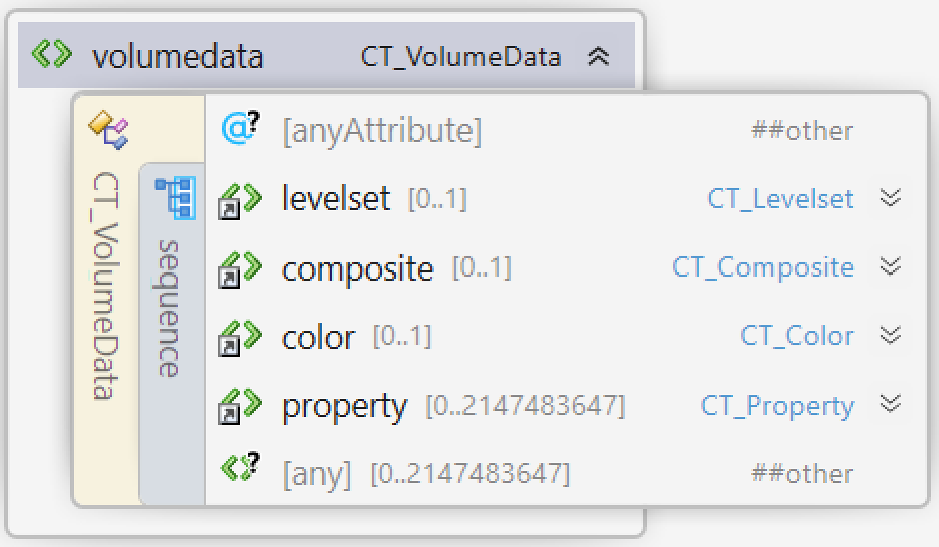

mesh CT_Mesh
[any] [0..2147483647] ##other
volumedata [0..1] CT_VolumeData

Name	Type	Use	Annotation
v:volumedata	CT_VolumeData	optional	The entry point to volumetric information of this specification.

The volumetric data <volumedata> element is a new OPTIONAL element which extends the root triangular mesh representation (i.e. <mesh> element).

3.2. Volumetric Data

Element **<volumedata>**

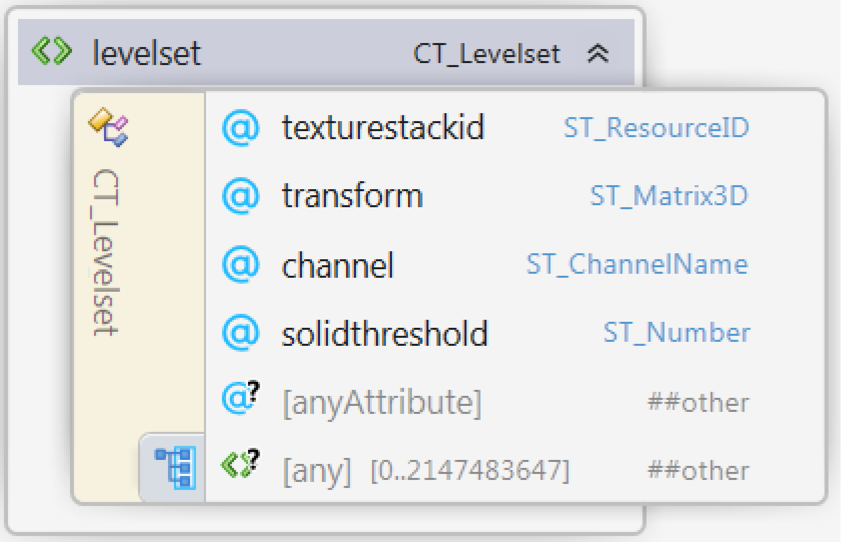

volumedata
CT_VolumeData
CT_VolumeData
sequence
[anyAttribute] ##other
levelset [0..1] CT_Levelset
composite [0..1] CT_Composite
color [0..1] CT_Color
property [0..2147483647] CT_Property
[any] [0..2147483647] ##other

The <volumedata> element references the volumetric data given by <volumetricstack>-elements and defines how their various channels are mapped to specific properties witin the interior volume of the enclosing mesh. The root mesh object determines the boundary geometry that acts as a trimming mesh for any volumetric data defined therein. Any data outside the mesh's bounds MUST be ignored. Volumedata MUST only be used in a mesh of object type "model" or "solidsupport".

The volumedata element can contain up to one child element, up to one child element, up to one <color> element, and an arbitray number of <property> elements.

3.2.1 Levelset element

Element **<levelset>**



Name	Type	Use	Annotation
volumetricstackid	ST_ResourceID	required	ResourceID of the volumetricstack that holds the levelset function
channel	ST_ChannelName	required	Name of the channel that holds the levelset function

this defines objects to be open, mathematically, which is ugly, since open sets are not manufacturable... pray to switch the equal sign to the definition of the solid!...

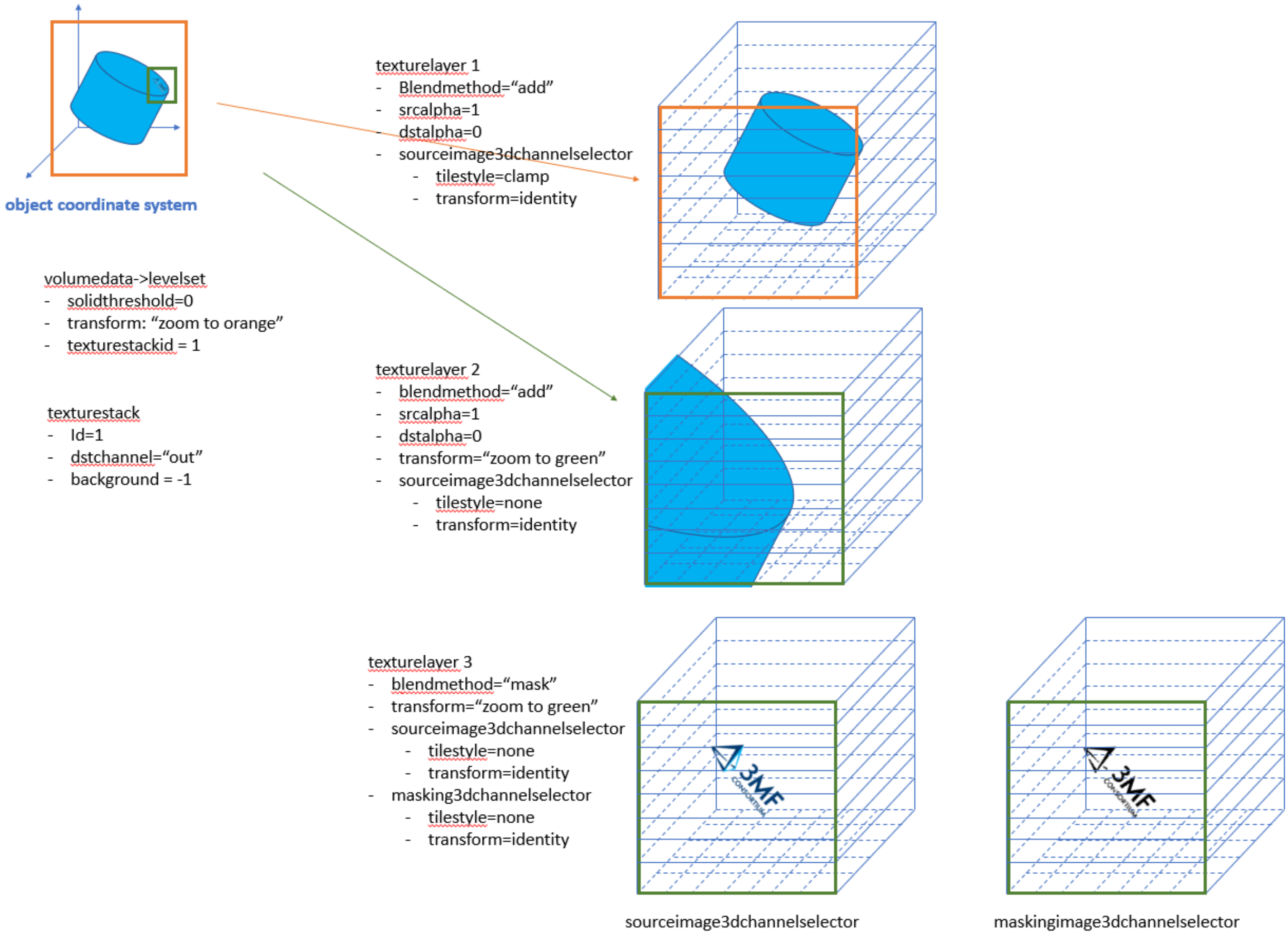
Name	Type	Use	Annotation
solidthreshold	ST_Number	*0.0*	All locations whose levelset function evaluates to a value < or >= than solidthreshold are consired within or outside of the specified object, respecively.
transform	ST_Matrix3D	required	Transformation of the object coordinate system into the volumetricstack coordinate system

The <levelset> element is used to define the boundary of the enclosing <object>-element as the set of locations where the value of the levelset function equals the solidthreshold attribute. Locations where the levelset function is smaller and larger than the solidthreshold indicate the interior and exterior of the object, respectively.

The levelset function is given by the "destination channel" within the <volumetricstack> with resource id matching the volumetricstackid-attribute and with name matching the "channel"-attribute of the <levelset>-element.

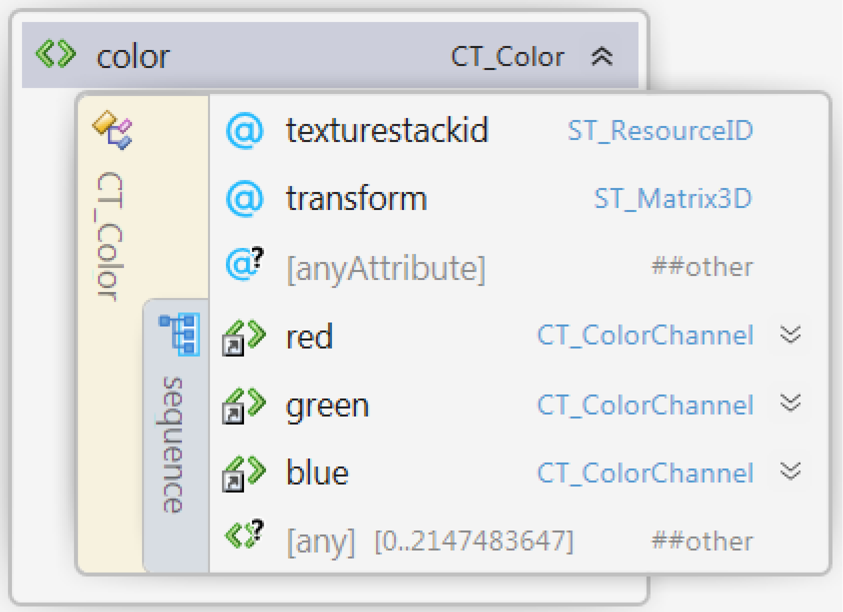
The mapping from object coordinates to the coordiante system of the corresponding volumetricstack is given by the transform attribute.

Figure 3-1: Illustration of different local coordinate systems and blendmethods



3.2.2 Color element

Element **<color>**



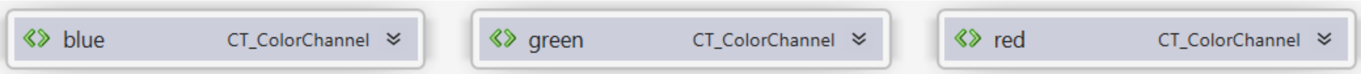
Name	Type	Use	Annotation
transform	ST_Matrix3D	required	Transformation of the object coordinate system into the volumetricstack coordinate system
volumetricstackid	ST_ResourceID	required	ResourceID of the volumetricstack that holds houses the channels to be used in the child color elements.

The <color> element is used to define the color of the object. The color format is RGB between normalized to the [0 - 1] range.

The <color>-element MUST contain exactly three <red>-, <green>- and <blue>-element.

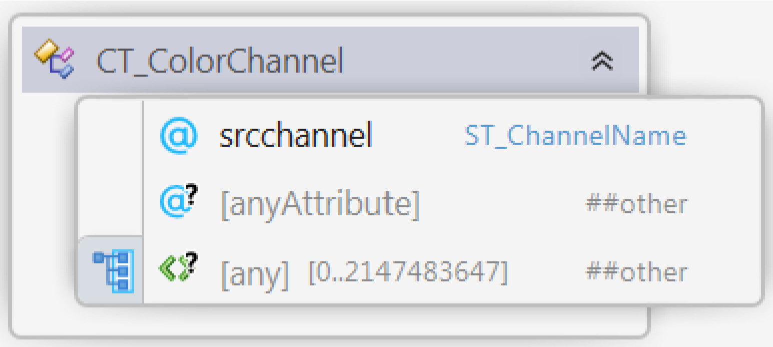
3.2.3 Color channel elements

Elements **<red>, <green> and <blue>**



of

Complex type **<colorchannel>**



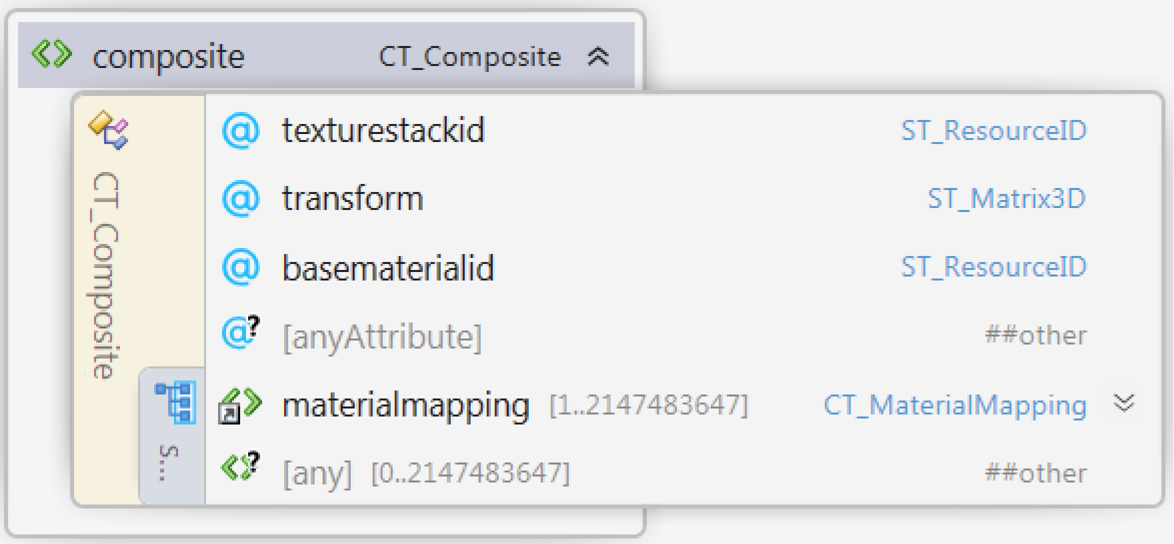
Name	Type	Use	Annotation
srcchannel	ST_ChannelName	required	Source channel for the values of this color channel

Each element instance of CT_ColorChannel MUST have an attribute "srcchannel" that references a destination channel from the <volumetricstack> with id matching the volumetricstackid of the parent <color> element.

If the value of the srcchannel of a <red>-, <green>- and <blue>-element is <0 or >1 it has to be truncated at 0 or 1, respectively.

3.2.4 Composite element

Element **<composite>**



composite CT_Composite

CT_Composite

texturestackid ST_ResourceID

transform ST_Matrix3D

basematerialid ST_ResourceID

[anyAttribute] ##other

materialmapping [1..2147483647] CT_MaterialMapping

[any] [0..2147483647] ##other

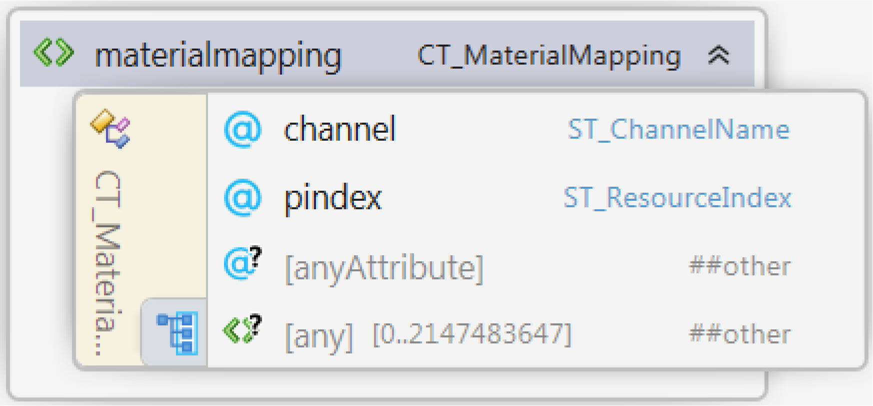
Name	Type	Use	Annotation
transform	ST_Matrix3D	required	Transformation of the object coordinate system into the volumetricstack coordinate system
volumetricstackid	ST_ResourceID	required	ResourceID of the volumetricstack that holds the channels used in the child <materialmapping>-elements
basematerialid	ST_ResourceID	required	ResourceID of the basematerial that holds the <base>-elements referenced in the child <materialmapping>-elements

The element describes a mixing ratio of printer materials at each position in space. The CONSUMER can determine the halftoning, mixing or dithering strategy that can be used to achieve these mixtures.

This element MUST contain at least one element, which will encode the relative contribution of a specific basematerial to the material mix.

3.2.5 Material mapping element

Element **<materialmapping>**



Name	Type	Use	Annotation
srcchannel	ST_ChannelName	required	Source channel for the values of this material
pindex	ST_ResourceIndex	required	ResourceIndex of the <base>-element within the parent's associated <basematerial>-element

The <materialmapping> element defines the relative contribution of a specific material to the mixing of materials in it's parent <composite>-element.

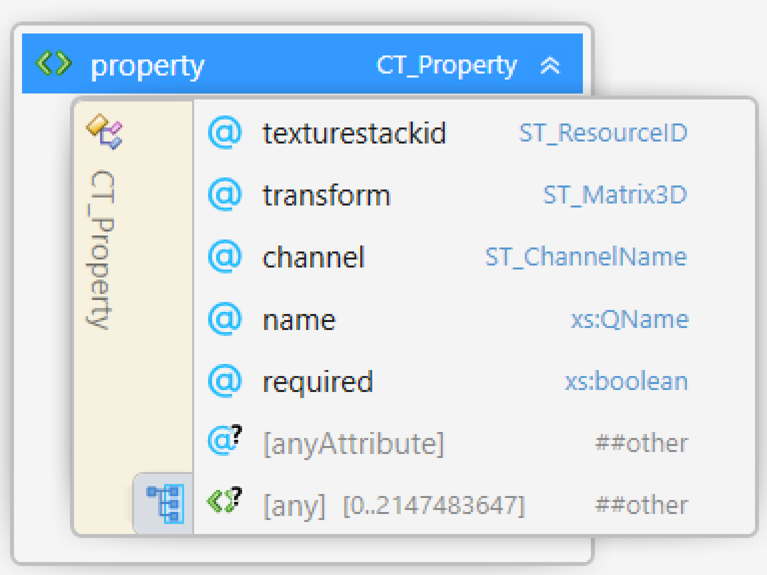
If the sampled value of a channel is <0 it must be evaluated as "0".

If the sum of all values in it's child <materialmapping>-elements is "0" ... TODO. PRODOCER should not do that, consumer MUST ... . Potentialy define a minimal value.

Each element instance of CT_MaterialMapping MUST have an attribute "srcchannel" that references a destination channel from the <volumetricstack> with id matching the volumetricstackid of the parent <composite> element.

3.2.4 Property element

Element **<property>**



Name	Type	Use	Annotation

Name	Type	Use	Annotation
transform	ST_Matrix3D	required	Transformation of the object coordinate system into the volumetricstack coordinate system
volumetricstackid	ST_ResourceID	required	ResourceID of the volumetricstack that holds the channel used by this property
channel	ST_ChannelName	required	Name of the channel that serves as source for this properties scalar value
name	xs:QName	required	Namespace and name of this property property
required	xs:boolean	optional	Indicator whether this property is required to process this 3MF document instance.

The <property> element allows to assign any point in space a scalar value of a freely definable property. This can be used to assign, e.g. opacity, conductivity, ...

The names of <property>-elements MUST be unique within a <volumedata>.

If a <property> is marked as `required`, and a consumer does not support it, it MUST warn the user or the appropriate upstream processes that it cannot process all contents in this 3MF document instance. Producers of 3MF files MUST mark all volumetric <properties> required to represent the design intent of a model as `required`.

TODO:

- rules for property versus composite (if they do not make sense together)
- rules for physical units
- rules for where to put namespace information ... ? Print ticket?

Part II. Appendixes

Appendix A. Glossary

See the standard 3MF Glossary.

Appendix B. 3MF XSD Schema

see: volumetric.xsd

```
```

Appendix C. Standard Namespace

Volumetric http://schemas.microsoft.com/3dmanufacturing/volumetric/2018/11

Appendix D: Example file

see: Examples/3dmodel.model

References

See the standard 3MF References.