



The OpenVX™ XML Schema User Guide

The Khronos® OpenVX Working Group, Editors: Jesse Villarreal

Version 1.1 (provisional), Mon, 10 Dec 2018 23:43:03 +0000

Table of Contents

1. XML Schema Extension User Guide	2
1.1. Purpose	2
1.2. Schema	2
2. XML Description	3
2.1. OpenVX Element	3
2.2. OpenVX Graph Objects	3
2.2.1. Nodes	4
2.2.2. Graph Parameters	5
2.2.3. Virtual Data Objects	5
2.3. OpenVX Data Objects	5
2.3.1. Array	5
2.3.2. Convolution	8
2.3.3. Delay	8
2.3.4. Distribution	8
2.3.5. Image	9
2.3.6. LUT	10
2.3.7. Matrix	11
2.3.8. Object Array	11
2.3.9. Pyramid	12
2.3.10. Remap	12
2.3.11. Scalar	13
2.3.12. Threshold	14
2.4. User Defined Elements	15
2.4.1. Library	15
2.4.2. Struct	15
3. XML Example	16
3.1. example.xml	16



Copyright 2013-2018 The Khronos Group Inc.

This specification is protected by copyright laws and contains material proprietary to Khronos. Except as described by these terms, it or any components may not be reproduced, republished, distributed, transmitted, displayed, broadcast or otherwise exploited in any manner without the express prior written permission of Khronos.

This specification has been created under the Khronos Intellectual Property Rights Policy, which is Attachment A of the Khronos Group Membership Agreement available at www.khronos.org/files/member_agreement.pdf. Khronos Group grants a conditional copyright license to use and reproduce the unmodified specification for any purpose, without fee or royalty, EXCEPT no licenses to any patent, trademark or other intellectual property rights are granted under these terms. Parties desiring to implement the specification and make use of Khronos trademarks in relation to that implementation, and receive reciprocal patent license protection under the Khronos IP Policy must become Adopters and confirm the implementation as conformant under the process defined by Khronos for this specification; see <https://www.khronos.org/adopters>.

Khronos makes no, and expressly disclaims any, representations or warranties, express or implied, regarding this specification, including, without limitation: merchantability, fitness for a particular purpose, non-infringement of any intellectual property, correctness, accuracy, completeness, timeliness, and reliability. Under no circumstances will Khronos, or any of its Promoters, Contributors or Members, or their respective partners, officers, directors, employees, agents or representatives be liable for any damages, whether direct, indirect, special or consequential damages for lost revenues, lost profits, or otherwise, arising from or in connection with these materials.

Khronos is a registered trademark, and OpenVX is a trademark of The Khronos Group Inc. OpenCL is a trademark of Apple Inc., used under license by Khronos. All other product names, trademarks, and/or company names are used solely for identification and belong to their respective owners.

Contributors

- Jesse Villarreal, Texas Instruments

Chapter 1. XML Schema Extension User Guide

1.1. Purpose

The purpose of this user guide is to supplement the OpenVX Schema extension with additional schema element descriptions and examples corresponding to a specific version of the OpenVX standard.

1.2. Schema

The XML Schema described in this user guide is available at the following url:

- <https://www.khronos.org/registry/vx/schema/openvx-1-1.xsd>

Chapter 2. XML Description

The XML Schema provides the necessary specification required to validate a OpenVX conformant XML file, but it alone is not sufficient to ensure full compatibility with OpenVX compliant implementations. Therefore, this section is intended supplement the schema by providing more details and examples of a conformant xml document.

The rest of this section will discuss each xml element in more detail. The examples contained in this section are a sample of a larger example xml file which is included in section [XML Example](#) "XML Example"

Each element described in this section will also reference the schema type name in order to help cross-reference the schema. For all elements which can be references, an optional "name" field can be used so that the import program can reference a reference by name.

2.1. OpenVX Element

Schema type: openvx.type

The 'openvx' tag encapsulates the entire OpenVX element. The 'references' attribute informs the parser as to the the number of OpenVX references that are contained in the xml file. Each data and graph object that corresponds to an OpenVX object will have a unique reference number in the xml file. The range of valid reference numbers should be between 0 and ["references"-1]. These references may appear in any order. Below is an example of the openvx element which indicates that the xml file has 124 objects enumerated.

```
<openvx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="https://www.khronos.org/registry/vx/schema"
        xsi:schemaLocation="https://www.khronos.org/registry/vx/schema openvx-1-1.xsd"
        references="127">
```

2.2. OpenVX Graph Objects

Schema type: graph.type

Graph elements each have their own reference number (and optional name), and may have child elements consisting of nodes, graph parameters, and virtual objects associated with the graph. The following example can be used as a reference for the discussion on graph objects.

```

<graph reference="50" name="GRAPH1" >
  <node reference="62" bordermode="CONSTANT">
    <kernel>org.khronos.openvx.sobel_3x3</kernel>
    <parameter index="0" reference="0" />
    <parameter index="1" reference="51" />
    <parameter index="2" reference="52" />
    <borderconst>#00000005</borderconst>
  </node>
  <node reference="64">
    <kernel>org.khronos.openvx.convertdepth</kernel>
    <parameter index="0" reference="51" />
    <parameter index="1" reference="1" />
    <parameter index="2" reference="63" />
    <parameter index="3" reference="61" />
  </node>
  <node reference="66" is_replicated="true">
    <kernel>org.khronos.openvx.add</kernel>
    <parameter index="0" reference="36" replicate_flag="true" />
    <parameter index="1" reference="41" replicate_flag="true" />
    <parameter index="2" reference="65" replicate_flag="false" />
    <parameter index="3" reference="46" replicate_flag="true" />
  </node>
  <parameter index="0" node="62" parameter="0" />
  <parameter index="1" node="64" parameter="1" />
  <image reference="51" width="0" height="0" format="VIRT" />
  <image reference="52" width="0" height="0" format="S016" />
  <image reference="53" width="320" height="240" format="VIRT" />
  <image reference="54" width="640" height="480" format="U008" />
  <pyramid reference="55" width="0" height="0" format="VIRT" scale="0.500000"
levels="4" />
  <pyramid reference="56" width="640" height="480" format="VIRT" scale="0.500000"
levels="4" />
  <pyramid reference="57" width="640" height="480" format="U008" scale="0.500000"
levels="4" />
  <array reference="58" capacity="0" elemType="VX_TYPE_INVALID" />
  <array reference="59" capacity="0" elemType="VX_TYPE_KEYPOINT" />
  <array reference="60" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
</graph>

```

2.2.1. Nodes

Schema type: node.type

Each node element of the graph has its own reference number and one child element specifying the kernel name. The node element has the following optional attributes: name, bordermode, and is_replicated. Each node element contains "n" number of parameter child elements, where "n" is equal to the number of parameters corresponding to the kernel specified. The index attribute of each parameter should monotonically increment. The reference attribute associated with each parameter should correspond to the reference number of the data object which is enumerated

either as a virtual object reference within the graph, or elsewhere in the xml document. The parameter elements have the optional attribute: `replicate_flag`.

2.2.2. Graph Parameters

Schema type: `graph.parameter.type`

Graph parameters do not have a reference number. The index attribute of each parameter should monotonically increment. The graph parameter identifies which parameter of which node is being exported as a graph parameter at the corresponding index.

2.2.3. Virtual Data Objects

Schema type: `virtual.image.type`, `virtual.array.type`, `virtual.pyramid.type`

Typically, data objects are enumerated outside of the graph element. However, virtual data objects (such as virtual arrays, virtual images, and virtual pyramids) are always listed as child elements of the graph element that they are associated with. Since virtual objects may not have accessible memory to the user, the xml does not specify the data of virtual objects. The format of the virtual data objects will be discussed further in [OpenVX Data Objects](#).

2.3. OpenVX Data Objects

Each of the data objects specified in OpenVX can be represented in xml representation. For each data object, the minimum information required to create the data object in a context is required. Optionally, the actual data included in the data object may also be represented in child elements of the object element in the xml. This way, during xml import into a context, the import utility should create all listed data objects, and if data is also specified for any data objects, then this data should be initialized within the imported data object. During xml export, if a data object has been created but not yet written to, then it may not be necessary to export the uninitialized data into the xml file for the associated data object.

Since virtual objects (virtual arrays, virtual images, or virtual pyramids) may not have accessible memory to the user, the xml does not specify the data of virtual objects.

2.3.1. Array

Schema type: `array.type`

Each array element below is a valid example of a virtual arrays within a graph element:

```
<array reference="48" capacity="0" elemType="VX_TYPE_INVALID" />
<array reference="49" capacity="0" elemType="VX_TYPE_KEYPOINT" />
<array reference="50" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
```

Each array element is a valid example of array objects. Please refer to the schema under `array.type` for all supported data types and formats of arrays:

```

<array reference="96" capacity="10" elemType="VX_TYPE_UINT8">
  <uint8>2 3 4 5 6 7 8 9 10 11 </uint8>
</array>
<array reference="97" capacity="20" elemType="VX_TYPE_CHAR">
  <char>a 13,.,^-</char>
</array>
<array reference="98" capacity="4" elemType="VX_TYPE_ENUM">
  <enum>-1 0 45057 </enum>
</array>
<array reference="99" capacity="4" elemType="VX_TYPE_DF_IMAGE">
  <df_image>RGB2 U008 VIRT </df_image>
</array>
<array reference="100" capacity="3" elemType="VX_TYPE_KEYPOINT">
  <keypoint>
    <x>0</x>
    <y>0</y>
    <strength>2.300000</strength>
    <scale>6.555550</scale>
    <orientation>0.905900</orientation>
    <tracking_status>5</tracking_status>
    <error>3.545500</error>
  </keypoint>
  <keypoint>
    <x>400</x>
    <y>235</y>
    <strength>5.222200</strength>
    <scale>1.221000</scale>
    <orientation>0.569500</orientation>
    <tracking_status>8</tracking_status>
    <error>462.500000</error>
  </keypoint>
</array>
<array reference="101" capacity="5" elemType="VX_TYPE_RECTANGLE">
  <rectangle>
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>640</end_x>
    <end_y>320</end_y>
  </rectangle>
  <rectangle>
    <start_x>65</start_x>
    <start_y>32</start_y>
    <end_x>128</end_x>
    <end_y>362</end_y>
  </rectangle>
</array>
<array reference="102" capacity="6" elemType="VX_TYPE_COORDINATES2D">
  <coordinates2d>
    <x>1</x>
    <y>2</y>
  </coordinates2d>

```



```

    <coordinates2d>
      <x>0</x>
      <y>55</y>
    </coordinates2d>
  </array>
  <array reference="103" capacity="6" elemType="VX_TYPE_COORDINATES3D">
    <coordinates3d>
      <x>1</x>
      <y>2</y>
      <z>3</z>
    </coordinates3d>
    <coordinates3d>
      <x>55</x>
      <y>66</y>
      <z>77</z>
    </coordinates3d>
  </array>
  <array reference="104" capacity="8" elemType="VX_TYPE_INT8">
    <int8>5 0 -3 -8 </int8>
  </array>
  <array reference="105" capacity="6" elemType="VX_TYPE_INT16">
    <int16>200 100 0 -100 -200 </int16>
  </array>
  <array reference="106" capacity="6" elemType="VX_TYPE_INT32">
    <int32>200000 100000 0 -100000 -200000 </int32>
  </array>
  <array reference="107" capacity="3" elemType="VX_TYPE_BOOL">
    <bool>>true false true </bool>
  </array>
  <array reference="108" capacity="4" elemType="VX_TYPE_SIZE">
    <size>8000 24000 </size>
  </array>
  <array reference="109" capacity="2" elemType="VX_TYPE_FLOAT64">
    <float64>1235.255660 -563.256700 </float64>
  </array>
  <array reference="110" capacity="8" elemType="VX_TYPE_UINT64">
    <uint64>9000000000 8000000000 7000000000 6000000000 </uint64>
  </array>
  <array reference="111" capacity="6" elemType="VX_TYPE_UINT16">
    <uint16>290 100 0 100 260 </uint16>
  </array>
  <array reference="112" capacity="6" elemType="VX_TYPE_UINT32">
    <uint32>200000 100000 0 100000 200000 </uint32>
  </array>
  <array reference="113" capacity="2" elemType="VX_TYPE_FLOAT32">
    <float32>1235.255615 -563.256714 </float32>
  </array>
  <array reference="114" capacity="8" elemType="VX_TYPE_INT64">
    <int64>9000000000 8000000000 -7000000000 -6000000000 </int64>
  </array>
  <array reference="120" capacity="4" elemType="USER_STRUCT_0">

```

```
<user>32 0 0 0 158 127 0 0 154 153 153 153 153 153 9 64 </user>
<user>64 0 0 0 158 127 0 0 154 153 153 153 153 153 25 64 </user>
</array>
```

2.3.2. Convolution

Schema type: convolution.type

Below is a valid example of a convolution object:

```
<convolution reference="31" rows="3" columns="3" scale="16">
  <int16 row="0" column="0">-1</int16>
  <int16 row="0" column="1">0</int16>
  <int16 row="0" column="2">1</int16>
  <int16 row="1" column="0">-2</int16>
  <int16 row="1" column="1">0</int16>
  <int16 row="1" column="2">1</int16>
  <int16 row="2" column="0">-1</int16>
  <int16 row="2" column="1">0</int16>
  <int16 row="2" column="2">1</int16>
</convolution>
```

2.3.3. Delay

Schema type: delay.type

Delay object types can be one of many of the other data types. For list of supported delay types, please refer to the schema under delay.type. Below is a valid example of a delay object:

```
<delay reference="55" count="3">
  <image reference="56" width="6" height="4" format="U008">
  </image>
  <image reference="57" width="6" height="4" format="U008">
  </image>
  <image reference="58" width="6" height="4" format="U008">
  </image>
</delay>
```

2.3.4. Distribution

Schema type: distribution.type

Below is a valid example of a distribution object:

```
<distribution reference="34" bins="16" offset="0" range="256">
  <frequency bin="0">0</frequency>
  <frequency bin="1">1</frequency>
  <frequency bin="2">2</frequency>
  <frequency bin="3">3</frequency>
  <frequency bin="4">4</frequency>
  <frequency bin="5">5</frequency>
  <frequency bin="6">6</frequency>
  <frequency bin="7">7</frequency>
  <frequency bin="8">8</frequency>
  <frequency bin="9">7</frequency>
  <frequency bin="10">6</frequency>
  <frequency bin="11">5</frequency>
  <frequency bin="12">4</frequency>
  <frequency bin="13">3</frequency>
  <frequency bin="14">2</frequency>
  <frequency bin="15">1</frequency>
</distribution>
```

2.3.5. Image

Schema type: image.type

Below are valid examples of image objects:

```

<image reference="0" width="6" height="4" format="U008">
  <roi reference="124" start_x="2" start_y="1" end_x="4" end_y="2">
    <roi reference="126" start_x="0" start_y="0" end_x="1" end_y="1" />
  </roi>
  <roi reference="125" start_x="2" start_y="1" end_x="4" end_y="2" />
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <uint8 x="0" y="0">0</uint8>
      <uint8 x="1" y="0">1</uint8>
      <uint8 x="2" y="0">2</uint8>
      <uint8 x="3" y="0">3</uint8>
      <uint8 x="4" y="0">4</uint8>
      <uint8 x="5" y="0">5</uint8>
      <uint8 x="0" y="1">6</uint8>
      <uint8 x="1" y="1">7</uint8>
      <uint8 x="2" y="1">8</uint8>
      <uint8 x="3" y="1">9</uint8>
      <uint8 x="4" y="1">10</uint8>
      <uint8 x="5" y="1">11</uint8>
      <uint8 x="0" y="2">12</uint8>
      <uint8 x="1" y="2">13</uint8>
      <uint8 x="2" y="2">14</uint8>
      <uint8 x="3" y="2">15</uint8>
      <uint8 x="4" y="2">16</uint8>
      <uint8 x="5" y="2">17</uint8>
      <uint8 x="0" y="3">18</uint8>
      <uint8 x="1" y="3">19</uint8>
      <uint8 x="2" y="3">20</uint8>
      <uint8 x="3" y="3">21</uint8>
      <uint8 x="4" y="3">22</uint8>
      <uint8 x="5" y="3">23</uint8>
    </pixels>
  </rectangle>
</image>
<image reference="1" width="6" height="4" format="U008">
</image>
<image reference="2" width="64" height="32" format="RGB2">
  <uniform>
    <rgb>#00ff80</rgb>
  </uniform>
</image>

```

2.3.6. LUT

Schema type: lut.type

Below is a valid example of a lut object:

```
<lut reference="32" count="256" elemType="VX_TYPE_UINT8">
  <uint8 index="0">255</uint8>
  <uint8 index="1">0</uint8>
  <uint8 index="2">1</uint8>
  <uint8 index="3">2</uint8>
  <uint8 index="4">3</uint8>
  <uint8 index="5">4</uint8>
  <uint8 index="6">5</uint8>
  <uint8 index="7">6</uint8>
  <uint8 index="8">7</uint8>

  // skipping indicies 9-253 to save space

  <uint8 index="252">251</uint8>
  <uint8 index="253">252</uint8>
  <uint8 index="254">253</uint8>
  <uint8 index="255">254</uint8>
</lut>
```

2.3.7. Matrix

Schema type: matrix.type

Below is a valid example of a matrix object:

```
<matrix reference="14" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
  <float32 row="0" column="0">-3.141593</float32>
  <float32 row="0" column="1">0.000000</float32>
  <float32 row="0" column="2">3.141593</float32>
  <float32 row="1" column="0">-6.283185</float32>
  <float32 row="1" column="1">0.000000</float32>
  <float32 row="1" column="2">6.283185</float32>
  <float32 row="2" column="0">-3.141593</float32>
  <float32 row="2" column="1">0.000000</float32>
  <float32 row="2" column="2">3.141593</float32>
</matrix>
```

2.3.8. Object Array

Schema type: object_array.type

Object Array types can be one of many of the other data types. For list of supported object array types, please refer to the schema under `object_array.type`. Below is a valid example of an `object_array` object:

```
<object_array reference="108" count="3">
  <image reference="109" width="6" height="4" format="U008">
  </image>
  <image reference="110" width="6" height="4" format="U008">
  </image>
  <image reference="111" width="6" height="4" format="U008">
  </image>
</object_array>
```

2.3.9. Pyramid

Schema type: pyramid.type

Below is a valid example of a pyramid object:

```
<pyramid reference="35" width="24" height="16" format="U008" scale="0.500000" levels="4">
  <image reference="36" width="24" height="16" format="U008">
  </image>
  <image reference="37" width="12" height="8" format="U008">
  </image>
  <image reference="38" width="6" height="4" format="U008">
  </image>
  <image reference="39" width="3" height="2" format="U008">
  </image>
</pyramid>
```

2.3.10. Remap

Schema type: remap.type

Below is a valid example of a remap object:

```
<remap reference="33" src_width="6" src_height="4" dst_width="6" dst_height="4">
  <point src_x="0.500000" src_y="0.500000" dst_x="0" dst_y="0" />
  <point src_x="1.000000" src_y="0.500000" dst_x="1" dst_y="0" />
  <point src_x="1.500000" src_y="0.500000" dst_x="2" dst_y="0" />
  <point src_x="2.000000" src_y="0.500000" dst_x="3" dst_y="0" />
  <point src_x="2.500000" src_y="0.500000" dst_x="4" dst_y="0" />
  <point src_x="3.000000" src_y="0.500000" dst_x="5" dst_y="0" />
  <point src_x="0.500000" src_y="1.000000" dst_x="0" dst_y="1" />
  <point src_x="1.000000" src_y="1.000000" dst_x="1" dst_y="1" />
  <point src_x="1.500000" src_y="1.000000" dst_x="2" dst_y="1" />
  <point src_x="2.000000" src_y="1.000000" dst_x="3" dst_y="1" />
  <point src_x="2.500000" src_y="1.000000" dst_x="4" dst_y="1" />
  <point src_x="3.000000" src_y="1.000000" dst_x="5" dst_y="1" />
  <point src_x="0.500000" src_y="1.500000" dst_x="0" dst_y="2" />
  <point src_x="1.000000" src_y="1.500000" dst_x="1" dst_y="2" />
  <point src_x="1.500000" src_y="1.500000" dst_x="2" dst_y="2" />
  <point src_x="2.000000" src_y="1.500000" dst_x="3" dst_y="2" />
  <point src_x="2.500000" src_y="1.500000" dst_x="4" dst_y="2" />
  <point src_x="3.000000" src_y="1.500000" dst_x="5" dst_y="2" />
  <point src_x="0.500000" src_y="2.000000" dst_x="0" dst_y="3" />
  <point src_x="1.000000" src_y="2.000000" dst_x="1" dst_y="3" />
  <point src_x="1.500000" src_y="2.000000" dst_x="2" dst_y="3" />
  <point src_x="2.000000" src_y="2.000000" dst_x="3" dst_y="3" />
  <point src_x="2.500000" src_y="2.000000" dst_x="4" dst_y="3" />
  <point src_x="3.000000" src_y="2.000000" dst_x="5" dst_y="3" />
</remap>
```

2.3.11. Scalar

Schema type: scalar.type

Below are valid examples of scalar objects:

```

<scalar reference="16" elemType="VX_TYPE_CHAR">
  <char>z</char>
</scalar>
<scalar reference="17" elemType="VX_TYPE_UINT8">
  <uint8>255</uint8>
</scalar>
<scalar reference="18" elemType="VX_TYPE_INT8">
  <int8>-128</int8>
</scalar>
<scalar reference="19" elemType="VX_TYPE_FLOAT32">
  <float32>3.141593</float32>
</scalar>
<scalar reference="20" elemType="VX_TYPE_FLOAT64">
  <float64>6.283185</float64>
</scalar>
<scalar reference="21" elemType="VX_TYPE_DF_IMAGE">
  <df_image>NV12</df_image>
</scalar>
<scalar reference="22" elemType="VX_TYPE_UINT16">
  <uint16>65535</uint16>
</scalar>
<scalar reference="23" elemType="VX_TYPE_INT16">
  <int16>-32768</int16>
</scalar>
<scalar reference="24" elemType="VX_TYPE_UINT64">
  <uint64>18446744073709551615</uint64>
</scalar>
<scalar reference="25" elemType="VX_TYPE_INT64">
  <int64>-9223372036854775808</int64>
</scalar>
<scalar reference="26" elemType="VX_TYPE_UINT32">
  <uint32>4294967295</uint32>
</scalar>
<scalar reference="27" elemType="VX_TYPE_SIZE">
  <size>100</size>
</scalar>
<scalar reference="28" elemType="VX_TYPE_BOOL">
  <bool>>true</bool>
</scalar>

```

2.3.12. Threshold

Schema type: threshold.type

Below is are valid examples of a threshold object:


```

<threshold reference="29" elemType="VX_TYPE_UINT8" true_value="0" false_value="0">
  <binary>240</binary>
</threshold>
<threshold reference="30" elemType="VX_TYPE_UINT8" true_value="0" false_value="0">
  <range lower="10" upper="240" />
</threshold>

```

2.4. User Defined Elements

The OpenVX XML Schema extension contains two additional optional elements which are intended to aid in portability of an OpenVX graph which is dependent on additional libraries or user structs.

2.4.1. Library

The 'library' element is simply a string type. If the OpenVX context is dependent on user kernels which may be defined in a separate library, then the name of this library can be listed here. This way, if the OpenVX implementation importing this XML file also has the library in the system, then it may know that it needs to load or link against the named library before parsing and loading the rest of the xml file. Because of this dependency, all library names should be listed in the xml file before any objects, as per the sequence constraint in the `openvx.type` of the schema. See an example below:

```

<library>openvx-debug</library>

```

2.4.2. Struct

The 'struct' element is intended to communicate to the parser that there is a user struct defined in the context. If there are array objects which have a user struct as an element type, then the array object of this type can not be created without knowing the size of the user struct. Therefore, this size needs to be communicated as part of the xml file as the "size" attribute in the struct element before any other objects are defined, as per the sequence constraint in the `openvx.type` of the schema. See an example below:

```

<struct size="16">USER_STRUCT_0</struct>

```

The schema expects the label used for user structs to always be "USER_STRUCT_*", where the wildcard can be any number. This way, multiple user structs can be exported/imported using unique suffix numbers. Now that this is defined at the top of the xml file, an array can reference the same label in the implementation:

```

<array reference="120" capacity="4" elemType="USER_STRUCT_0">
  <user>32 0 0 0 158 127 0 0 154 153 153 153 153 9 64 </user>
  <user>64 0 0 0 158 127 0 0 154 153 153 153 153 25 64 </user>
</array>

```

Chapter 3. XML Example

3.1. example.xml

The following is an OpenVX XML Schema compatible example xml file. It contains at least one example of each object type that the schema supports, and can be used as a reference to for programming compatible import and export utilities.

```
<?xml version="1.0" encoding="utf-8"?>
<openvx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="https://www.khronos.org/registry/vx/schema"
  xsi:schemaLocation="https://www.khronos.org/registry/vx/schema openvx-1-0.xsd"
  references="140">
  <library>openvx-debug</library>
  <struct size="16">USER_STRUCT_0</struct>
  <image reference="0" width="6" height="4" format="U008" name="INPUT_IMG" >
    <roi reference="136" start_x="2" start_y="1" end_x="4" end_y="2">
      <roi reference="138" start_x="0" start_y="0" end_x="1" end_y="1" />
    </roi>
    <roi reference="137" start_x="2" start_y="1" end_x="4" end_y="2" />
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <uint8 x="0" y="0">0</uint8>
      <uint8 x="1" y="0">1</uint8>
      <uint8 x="2" y="0">2</uint8>
      <uint8 x="3" y="0">3</uint8>
      <uint8 x="4" y="0">4</uint8>
      <uint8 x="5" y="0">5</uint8>
      <uint8 x="0" y="1">6</uint8>
      <uint8 x="1" y="1">7</uint8>
      <uint8 x="2" y="1">8</uint8>
      <uint8 x="3" y="1">9</uint8>
      <uint8 x="4" y="1">10</uint8>
      <uint8 x="5" y="1">11</uint8>
      <uint8 x="0" y="2">12</uint8>
      <uint8 x="1" y="2">13</uint8>
      <uint8 x="2" y="2">14</uint8>
      <uint8 x="3" y="2">15</uint8>
      <uint8 x="4" y="2">16</uint8>
      <uint8 x="5" y="2">17</uint8>
      <uint8 x="0" y="3">18</uint8>
      <uint8 x="1" y="3">19</uint8>
      <uint8 x="2" y="3">20</uint8>
      <uint8 x="3" y="3">21</uint8>
      <uint8 x="4" y="3">22</uint8>
    </pixels>
  </rectangle>
</openvx>
```

```

        <uint8 x="5" y="3">23</uint8>
    </pixels>
</rectangle>
</image>
<image reference="1" width="6" height="4" format="U008" name="OUTPUT_IMG" >
</image>
<image reference="2" width="6" height="4" format="S016">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <int16 x="0" y="0">-32768</int16>
            <int16 x="1" y="0">-32767</int16>
            <int16 x="2" y="0">-32766</int16>
            <int16 x="3" y="0">-32765</int16>
            <int16 x="4" y="0">-32764</int16>
            <int16 x="5" y="0">-32763</int16>
            <int16 x="0" y="1">-32762</int16>
            <int16 x="1" y="1">-32761</int16>
            <int16 x="2" y="1">-32760</int16>
            <int16 x="3" y="1">-32759</int16>
            <int16 x="4" y="1">-32758</int16>
            <int16 x="5" y="1">-32757</int16>
            <int16 x="0" y="2">-32756</int16>
            <int16 x="1" y="2">-32755</int16>
            <int16 x="2" y="2">-32754</int16>
            <int16 x="3" y="2">-32753</int16>
            <int16 x="4" y="2">-32752</int16>
            <int16 x="5" y="2">-32751</int16>
            <int16 x="0" y="3">-32750</int16>
            <int16 x="1" y="3">-32749</int16>
            <int16 x="2" y="3">-32748</int16>
            <int16 x="3" y="3">-32747</int16>
            <int16 x="4" y="3">-32746</int16>
            <int16 x="5" y="3">-32745</int16>
        </pixels>
    </rectangle>
</image>
<image reference="3" width="6" height="4" format="U016">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint16 x="0" y="0">65535</uint16>
            <uint16 x="1" y="0">65534</uint16>
            <uint16 x="2" y="0">65533</uint16>
            <uint16 x="3" y="0">65532</uint16>

```

```

<uint16 x="4" y="0">65531</uint16>
<uint16 x="5" y="0">65530</uint16>
<uint16 x="0" y="1">65529</uint16>
<uint16 x="1" y="1">65528</uint16>
<uint16 x="2" y="1">65527</uint16>
<uint16 x="3" y="1">65526</uint16>
<uint16 x="4" y="1">65525</uint16>
<uint16 x="5" y="1">65524</uint16>
<uint16 x="0" y="2">65523</uint16>
<uint16 x="1" y="2">65522</uint16>
<uint16 x="2" y="2">65521</uint16>
<uint16 x="3" y="2">65520</uint16>
<uint16 x="4" y="2">65519</uint16>
<uint16 x="5" y="2">65518</uint16>
<uint16 x="0" y="3">65517</uint16>
<uint16 x="1" y="3">65516</uint16>
<uint16 x="2" y="3">65515</uint16>
<uint16 x="3" y="3">65514</uint16>
<uint16 x="4" y="3">65513</uint16>
<uint16 x="5" y="3">65512</uint16>
</pixels>
</rectangle>
</image>
<image reference="4" width="6" height="4" format="S032">
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <int32 x="0" y="0">-2147483648</int32>
      <int32 x="1" y="0">-2147483647</int32>
      <int32 x="2" y="0">-2147483646</int32>
      <int32 x="3" y="0">-2147483645</int32>
      <int32 x="4" y="0">-2147483644</int32>
      <int32 x="5" y="0">-2147483643</int32>
      <int32 x="0" y="1">-2147483642</int32>
      <int32 x="1" y="1">-2147483641</int32>
      <int32 x="2" y="1">-2147483640</int32>
      <int32 x="3" y="1">-2147483639</int32>
      <int32 x="4" y="1">-2147483638</int32>
      <int32 x="5" y="1">-2147483637</int32>
      <int32 x="0" y="2">-2147483636</int32>
      <int32 x="1" y="2">-2147483635</int32>
      <int32 x="2" y="2">-2147483634</int32>
      <int32 x="3" y="2">-2147483633</int32>
      <int32 x="4" y="2">-2147483632</int32>
      <int32 x="5" y="2">-2147483631</int32>
      <int32 x="0" y="3">-2147483630</int32>
      <int32 x="1" y="3">-2147483629</int32>
      <int32 x="2" y="3">-2147483628</int32>
    </pixels>
  </rectangle>
</image>

```

```

        <int32 x="3" y="3">-2147483627</int32>
        <int32 x="4" y="3">-2147483626</int32>
        <int32 x="5" y="3">-2147483625</int32>
    </pixels>
</rectangle>
</image>
<image reference="5" width="6" height="4" format="U032">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint32 x="0" y="0">2147483647</uint32>
            <uint32 x="1" y="0">2147483646</uint32>
            <uint32 x="2" y="0">2147483645</uint32>
            <uint32 x="3" y="0">2147483644</uint32>
            <uint32 x="4" y="0">2147483643</uint32>
            <uint32 x="5" y="0">2147483642</uint32>
            <uint32 x="0" y="1">2147483641</uint32>
            <uint32 x="1" y="1">2147483640</uint32>
            <uint32 x="2" y="1">2147483639</uint32>
            <uint32 x="3" y="1">2147483638</uint32>
            <uint32 x="4" y="1">2147483637</uint32>
            <uint32 x="5" y="1">2147483636</uint32>
            <uint32 x="0" y="2">2147483635</uint32>
            <uint32 x="1" y="2">2147483634</uint32>
            <uint32 x="2" y="2">2147483633</uint32>
            <uint32 x="3" y="2">2147483632</uint32>
            <uint32 x="4" y="2">2147483631</uint32>
            <uint32 x="5" y="2">2147483630</uint32>
            <uint32 x="0" y="3">2147483629</uint32>
            <uint32 x="1" y="3">2147483628</uint32>
            <uint32 x="2" y="3">2147483627</uint32>
            <uint32 x="3" y="3">2147483626</uint32>
            <uint32 x="4" y="3">2147483625</uint32>
            <uint32 x="5" y="3">2147483624</uint32>
        </pixels>
    </rectangle>
</image>
<image reference="6" width="6" height="4" format="RGB2">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <rgb x="0" y="0">#000102</rgb>
            <rgb x="1" y="0">#010203</rgb>
            <rgb x="2" y="0">#020304</rgb>
            <rgb x="3" y="0">#030405</rgb>

```

```

    <rgb x="4" y="0">#040506</rgb>
    <rgb x="5" y="0">#050607</rgb>
    <rgb x="0" y="1">#060708</rgb>
    <rgb x="1" y="1">#070809</rgb>
    <rgb x="2" y="1">#08090a</rgb>
    <rgb x="3" y="1">#090a0b</rgb>
    <rgb x="4" y="1">#0a0b0c</rgb>
    <rgb x="5" y="1">#0b0c0d</rgb>
    <rgb x="0" y="2">#0c0d0e</rgb>
    <rgb x="1" y="2">#0d0e0f</rgb>
    <rgb x="2" y="2">#0e0f10</rgb>
    <rgb x="3" y="2">#0f1011</rgb>
    <rgb x="4" y="2">#101112</rgb>
    <rgb x="5" y="2">#111213</rgb>
    <rgb x="0" y="3">#121314</rgb>
    <rgb x="1" y="3">#131415</rgb>
    <rgb x="2" y="3">#141516</rgb>
    <rgb x="3" y="3">#151617</rgb>
    <rgb x="4" y="3">#161718</rgb>
    <rgb x="5" y="3">#171819</rgb>
  </pixels>
</rectangle>
</image>
<image reference="7" width="6" height="4" format="RGBA">
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <rgba x="0" y="0">#00010203</rgba>
      <rgba x="1" y="0">#01020304</rgba>
      <rgba x="2" y="0">#02030405</rgba>
      <rgba x="3" y="0">#03040506</rgba>
      <rgba x="4" y="0">#04050607</rgba>
      <rgba x="5" y="0">#05060708</rgba>
      <rgba x="0" y="1">#06070809</rgba>
      <rgba x="1" y="1">#0708090a</rgba>
      <rgba x="2" y="1">#08090a0b</rgba>
      <rgba x="3" y="1">#090a0b0c</rgba>
      <rgba x="4" y="1">#0a0b0c0d</rgba>
      <rgba x="5" y="1">#0b0c0d0e</rgba>
      <rgba x="0" y="2">#0c0d0e0f</rgba>
      <rgba x="1" y="2">#0d0e0f10</rgba>
      <rgba x="2" y="2">#0e0f1011</rgba>
      <rgba x="3" y="2">#0f101112</rgba>
      <rgba x="4" y="2">#10111213</rgba>
      <rgba x="5" y="2">#11121314</rgba>
      <rgba x="0" y="3">#12131415</rgba>
      <rgba x="1" y="3">#13141516</rgba>
      <rgba x="2" y="3">#14151617</rgba>
    </pixels>
  </rectangle>
</image>

```

```

        <rgba x="3" y="3">#15161718</rgba>
        <rgba x="4" y="3">#16171819</rgba>
        <rgba x="5" y="3">#1718191a</rgba>
    </pixels>
</rectangle>
</image>
<image reference="8" width="6" height="4" format="UYVY">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="1" y="0">1 2</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="3" y="0">3 4</yuv>
            <yuv x="4" y="0">4 5</yuv>
            <yuv x="5" y="0">5 6</yuv>
            <yuv x="0" y="1">6 7</yuv>
            <yuv x="1" y="1">7 8</yuv>
            <yuv x="2" y="1">8 9</yuv>
            <yuv x="3" y="1">9 10</yuv>
            <yuv x="4" y="1">10 11</yuv>
            <yuv x="5" y="1">11 12</yuv>
            <yuv x="0" y="2">12 13</yuv>
            <yuv x="1" y="2">13 14</yuv>
            <yuv x="2" y="2">14 15</yuv>
            <yuv x="3" y="2">15 16</yuv>
            <yuv x="4" y="2">16 17</yuv>
            <yuv x="5" y="2">17 18</yuv>
            <yuv x="0" y="3">18 19</yuv>
            <yuv x="1" y="3">19 20</yuv>
            <yuv x="2" y="3">20 21</yuv>
            <yuv x="3" y="3">21 22</yuv>
            <yuv x="4" y="3">22 23</yuv>
            <yuv x="5" y="3">23 24</yuv>
        </pixels>
    </rectangle>
</image>
<image reference="9" width="6" height="4" format="YUYV">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="1" y="0">1 2</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="3" y="0">3 4</yuv>

```

```

    <yuv x="4" y="0">4 5</yuv>
    <yuv x="5" y="0">5 6</yuv>
    <yuv x="0" y="1">6 7</yuv>
    <yuv x="1" y="1">7 8</yuv>
    <yuv x="2" y="1">8 9</yuv>
    <yuv x="3" y="1">9 10</yuv>
    <yuv x="4" y="1">10 11</yuv>
    <yuv x="5" y="1">11 12</yuv>
    <yuv x="0" y="2">12 13</yuv>
    <yuv x="1" y="2">13 14</yuv>
    <yuv x="2" y="2">14 15</yuv>
    <yuv x="3" y="2">15 16</yuv>
    <yuv x="4" y="2">16 17</yuv>
    <yuv x="5" y="2">17 18</yuv>
    <yuv x="0" y="3">18 19</yuv>
    <yuv x="1" y="3">19 20</yuv>
    <yuv x="2" y="3">20 21</yuv>
    <yuv x="3" y="3">21 22</yuv>
    <yuv x="4" y="3">22 23</yuv>
    <yuv x="5" y="3">23 24</yuv>
  </pixels>
</rectangle>
</image>
<image reference="10" width="6" height="4" format="IYUV">
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <uint8 x="0" y="0">0</uint8>
      <uint8 x="1" y="0">1</uint8>
      <uint8 x="2" y="0">2</uint8>
      <uint8 x="3" y="0">3</uint8>
      <uint8 x="4" y="0">4</uint8>
      <uint8 x="5" y="0">5</uint8>
      <uint8 x="0" y="1">6</uint8>
      <uint8 x="1" y="1">7</uint8>
      <uint8 x="2" y="1">8</uint8>
      <uint8 x="3" y="1">9</uint8>
      <uint8 x="4" y="1">10</uint8>
      <uint8 x="5" y="1">11</uint8>
      <uint8 x="0" y="2">12</uint8>
      <uint8 x="1" y="2">13</uint8>
      <uint8 x="2" y="2">14</uint8>
      <uint8 x="3" y="2">15</uint8>
      <uint8 x="4" y="2">16</uint8>
      <uint8 x="5" y="2">17</uint8>
      <uint8 x="0" y="3">18</uint8>
      <uint8 x="1" y="3">19</uint8>
      <uint8 x="2" y="3">20</uint8>
    </pixels>
  </rectangle>
</image>

```



```

        <uint8 x="3" y="3">21</uint8>
        <uint8 x="4" y="3">22</uint8>
        <uint8 x="5" y="3">23</uint8>
    </pixels>
</rectangle>
<rectangle plane="1">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
        <uint8 x="0" y="0">0</uint8>
        <uint8 x="2" y="0">2</uint8>
        <uint8 x="4" y="0">4</uint8>
        <uint8 x="0" y="2">12</uint8>
        <uint8 x="2" y="2">14</uint8>
        <uint8 x="4" y="2">16</uint8>
    </pixels>
</rectangle>
<rectangle plane="2">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
        <uint8 x="0" y="0">0</uint8>
        <uint8 x="2" y="0">2</uint8>
        <uint8 x="4" y="0">4</uint8>
        <uint8 x="0" y="2">12</uint8>
        <uint8 x="2" y="2">14</uint8>
        <uint8 x="4" y="2">16</uint8>
    </pixels>
</rectangle>
</image>
<image reference="11" width="6" height="4" format="YUV4">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>

```

```

    <uint8 x="4" y="1">10</uint8>
    <uint8 x="5" y="1">11</uint8>
    <uint8 x="0" y="2">12</uint8>
    <uint8 x="1" y="2">13</uint8>
    <uint8 x="2" y="2">14</uint8>
    <uint8 x="3" y="2">15</uint8>
    <uint8 x="4" y="2">16</uint8>
    <uint8 x="5" y="2">17</uint8>
    <uint8 x="0" y="3">18</uint8>
    <uint8 x="1" y="3">19</uint8>
    <uint8 x="2" y="3">20</uint8>
    <uint8 x="3" y="3">21</uint8>
    <uint8 x="4" y="3">22</uint8>
    <uint8 x="5" y="3">23</uint8>
  </pixels>
</rectangle>
<rectangle plane="1">
  <start_x>0</start_x>
  <start_y>0</start_y>
  <end_x>6</end_x>
  <end_y>4</end_y>
  <pixels>
    <uint8 x="0" y="0">0</uint8>
    <uint8 x="1" y="0">1</uint8>
    <uint8 x="2" y="0">2</uint8>
    <uint8 x="3" y="0">3</uint8>
    <uint8 x="4" y="0">4</uint8>
    <uint8 x="5" y="0">5</uint8>
    <uint8 x="0" y="1">6</uint8>
    <uint8 x="1" y="1">7</uint8>
    <uint8 x="2" y="1">8</uint8>
    <uint8 x="3" y="1">9</uint8>
    <uint8 x="4" y="1">10</uint8>
    <uint8 x="5" y="1">11</uint8>
    <uint8 x="0" y="2">12</uint8>
    <uint8 x="1" y="2">13</uint8>
    <uint8 x="2" y="2">14</uint8>
    <uint8 x="3" y="2">15</uint8>
    <uint8 x="4" y="2">16</uint8>
    <uint8 x="5" y="2">17</uint8>
    <uint8 x="0" y="3">18</uint8>
    <uint8 x="1" y="3">19</uint8>
    <uint8 x="2" y="3">20</uint8>
    <uint8 x="3" y="3">21</uint8>
    <uint8 x="4" y="3">22</uint8>
    <uint8 x="5" y="3">23</uint8>
  </pixels>
</rectangle>
<rectangle plane="2">
  <start_x>0</start_x>
  <start_y>0</start_y>

```

```

<end_x>6</end_x>
<end_y>4</end_y>
<pixels>
  <uint8 x="0" y="0">0</uint8>
  <uint8 x="1" y="0">1</uint8>
  <uint8 x="2" y="0">2</uint8>
  <uint8 x="3" y="0">3</uint8>
  <uint8 x="4" y="0">4</uint8>
  <uint8 x="5" y="0">5</uint8>
  <uint8 x="0" y="1">6</uint8>
  <uint8 x="1" y="1">7</uint8>
  <uint8 x="2" y="1">8</uint8>
  <uint8 x="3" y="1">9</uint8>
  <uint8 x="4" y="1">10</uint8>
  <uint8 x="5" y="1">11</uint8>
  <uint8 x="0" y="2">12</uint8>
  <uint8 x="1" y="2">13</uint8>
  <uint8 x="2" y="2">14</uint8>
  <uint8 x="3" y="2">15</uint8>
  <uint8 x="4" y="2">16</uint8>
  <uint8 x="5" y="2">17</uint8>
  <uint8 x="0" y="3">18</uint8>
  <uint8 x="1" y="3">19</uint8>
  <uint8 x="2" y="3">20</uint8>
  <uint8 x="3" y="3">21</uint8>
  <uint8 x="4" y="3">22</uint8>
  <uint8 x="5" y="3">23</uint8>
</pixels>
</rectangle>
</image>
<image reference="12" width="6" height="4" format="NV12">
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <uint8 x="0" y="0">0</uint8>
      <uint8 x="1" y="0">1</uint8>
      <uint8 x="2" y="0">2</uint8>
      <uint8 x="3" y="0">3</uint8>
      <uint8 x="4" y="0">4</uint8>
      <uint8 x="5" y="0">5</uint8>
      <uint8 x="0" y="1">6</uint8>
      <uint8 x="1" y="1">7</uint8>
      <uint8 x="2" y="1">8</uint8>
      <uint8 x="3" y="1">9</uint8>
      <uint8 x="4" y="1">10</uint8>
      <uint8 x="5" y="1">11</uint8>
      <uint8 x="0" y="2">12</uint8>
      <uint8 x="1" y="2">13</uint8>

```

```

    <uint8 x="2" y="2">14</uint8>
    <uint8 x="3" y="2">15</uint8>
    <uint8 x="4" y="2">16</uint8>
    <uint8 x="5" y="2">17</uint8>
    <uint8 x="0" y="3">18</uint8>
    <uint8 x="1" y="3">19</uint8>
    <uint8 x="2" y="3">20</uint8>
    <uint8 x="3" y="3">21</uint8>
    <uint8 x="4" y="3">22</uint8>
    <uint8 x="5" y="3">23</uint8>
  </pixels>
</rectangle>
<rectangle plane="1">
  <start_x>0</start_x>
  <start_y>0</start_y>
  <end_x>6</end_x>
  <end_y>4</end_y>
  <pixels>
    <yuv x="0" y="0">0 1</yuv>
    <yuv x="2" y="0">2 3</yuv>
    <yuv x="4" y="0">4 5</yuv>
    <yuv x="0" y="2">12 13</yuv>
    <yuv x="2" y="2">14 15</yuv>
    <yuv x="4" y="2">16 17</yuv>
  </pixels>
</rectangle>
</image>
<image reference="13" width="6" height="4" format="NV21">
  <rectangle plane="0">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
      <uint8 x="0" y="0">0</uint8>
      <uint8 x="1" y="0">1</uint8>
      <uint8 x="2" y="0">2</uint8>
      <uint8 x="3" y="0">3</uint8>
      <uint8 x="4" y="0">4</uint8>
      <uint8 x="5" y="0">5</uint8>
      <uint8 x="0" y="1">6</uint8>
      <uint8 x="1" y="1">7</uint8>
      <uint8 x="2" y="1">8</uint8>
      <uint8 x="3" y="1">9</uint8>
      <uint8 x="4" y="1">10</uint8>
      <uint8 x="5" y="1">11</uint8>
      <uint8 x="0" y="2">12</uint8>
      <uint8 x="1" y="2">13</uint8>
      <uint8 x="2" y="2">14</uint8>
      <uint8 x="3" y="2">15</uint8>
      <uint8 x="4" y="2">16</uint8>

```

```

        <uint8 x="5" y="2">17</uint8>
        <uint8 x="0" y="3">18</uint8>
        <uint8 x="1" y="3">19</uint8>
        <uint8 x="2" y="3">20</uint8>
        <uint8 x="3" y="3">21</uint8>
        <uint8 x="4" y="3">22</uint8>
        <uint8 x="5" y="3">23</uint8>
    </pixels>
</rectangle>
<rectangle plane="1">
    <start_x>0</start_x>
    <start_y>0</start_y>
    <end_x>6</end_x>
    <end_y>4</end_y>
    <pixels>
        <yuv x="0" y="0">0 1</yuv>
        <yuv x="2" y="0">2 3</yuv>
        <yuv x="4" y="0">4 5</yuv>
        <yuv x="0" y="2">12 13</yuv>
        <yuv x="2" y="2">14 15</yuv>
        <yuv x="4" y="2">16 17</yuv>
    </pixels>
</rectangle>
</image>
<matrix reference="14" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
    <float32 row="0" column="0">-3.141593</float32>
    <float32 row="0" column="1">0.000000</float32>
    <float32 row="0" column="2">3.141593</float32>
    <float32 row="1" column="0">-6.283185</float32>
    <float32 row="1" column="1">0.000000</float32>
    <float32 row="1" column="2">6.283185</float32>
    <float32 row="2" column="0">-3.141593</float32>
    <float32 row="2" column="1">0.000000</float32>
    <float32 row="2" column="2">3.141593</float32>
</matrix>
<matrix reference="15" elemType="VX_TYPE_INT32" rows="3" columns="3">
    <int32 row="0" column="0">-23</int32>
    <int32 row="0" column="1">904</int32>
    <int32 row="0" column="2">29</int32>
    <int32 row="1" column="0">782</int32>
    <int32 row="1" column="1">823</int32>
    <int32 row="1" column="2">288</int32>
    <int32 row="2" column="0">848</int32>
    <int32 row="2" column="1">828</int32>
    <int32 row="2" column="2">994</int32>
</matrix>
<scalar reference="16" elemType="VX_TYPE_CHAR">
    <char>z</char>
</scalar>
<scalar reference="17" elemType="VX_TYPE_UINT8">
    <uint8>255</uint8>

```

```

</scalar>
<scalar reference="18" elemType="VX_TYPE_INT8">
  <int8>-128</int8>
</scalar>
<scalar reference="19" elemType="VX_TYPE_FLOAT32">
  <float32>3.141593</float32>
</scalar>
<scalar reference="20" elemType="VX_TYPE_FLOAT64">
  <float64>6.283185</float64>
</scalar>
<scalar reference="21" elemType="VX_TYPE_DF_IMAGE">
  <df_image>NV12</df_image>
</scalar>
<scalar reference="22" elemType="VX_TYPE_UINT16">
  <uint16>65535</uint16>
</scalar>
<scalar reference="23" elemType="VX_TYPE_INT16">
  <int16>-32768</int16>
</scalar>
<scalar reference="24" elemType="VX_TYPE_UINT64">
  <uint64>18446744073709551615</uint64>
</scalar>
<scalar reference="25" elemType="VX_TYPE_INT64">
  <int64>-9223372036854775808</int64>
</scalar>
<scalar reference="26" elemType="VX_TYPE_UINT32">
  <uint32>4294967295</uint32>
</scalar>
<scalar reference="27" elemType="VX_TYPE_SIZE">
  <size>100</size>
</scalar>
<scalar reference="28" elemType="VX_TYPE_BOOL">
  <bool>>true</bool>
</scalar>
<threshold reference="29" elemType="VX_TYPE_UINT8" true_value="255" false_value="
0">
  <binary>240</binary>
</threshold>
<threshold reference="30" elemType="VX_TYPE_UINT8" true_value="255" false_value="
0">
  <range lower="10" upper="240" />
</threshold>
<convolution reference="31" rows="3" columns="3" scale="16">
  <int16 row="0" column="0">-1</int16>
  <int16 row="0" column="1">0</int16>
  <int16 row="0" column="2">1</int16>
  <int16 row="1" column="0">-2</int16>
  <int16 row="1" column="1">0</int16>
  <int16 row="1" column="2">1</int16>
  <int16 row="2" column="0">-1</int16>
  <int16 row="2" column="1">0</int16>

```

```
<int16 row="2" column="2">1</int16>
</convolution>
<lut reference="32" count="256" elemType="VX_TYPE_UINT8">
  <uint8 index="0">255</uint8>
  <uint8 index="1">0</uint8>
  <uint8 index="2">1</uint8>
  <uint8 index="3">2</uint8>
  <uint8 index="4">3</uint8>
  <uint8 index="5">4</uint8>
  <uint8 index="6">5</uint8>
  <uint8 index="7">6</uint8>
  <uint8 index="8">7</uint8>
  <uint8 index="9">8</uint8>
  <uint8 index="10">9</uint8>
  <uint8 index="11">10</uint8>
  <uint8 index="12">11</uint8>
  <uint8 index="13">12</uint8>
  <uint8 index="14">13</uint8>
  <uint8 index="15">14</uint8>
  <uint8 index="16">15</uint8>
  <uint8 index="17">16</uint8>
  <uint8 index="18">17</uint8>
  <uint8 index="19">18</uint8>
  <uint8 index="20">19</uint8>
  <uint8 index="21">20</uint8>
  <uint8 index="22">21</uint8>
  <uint8 index="23">22</uint8>
  <uint8 index="24">23</uint8>
  <uint8 index="25">24</uint8>
  <uint8 index="26">25</uint8>
  <uint8 index="27">26</uint8>
  <uint8 index="28">27</uint8>
  <uint8 index="29">28</uint8>
  <uint8 index="30">29</uint8>
  <uint8 index="31">30</uint8>
  <uint8 index="32">31</uint8>
  <uint8 index="33">32</uint8>
  <uint8 index="34">33</uint8>
  <uint8 index="35">34</uint8>
  <uint8 index="36">35</uint8>
  <uint8 index="37">36</uint8>
  <uint8 index="38">37</uint8>
  <uint8 index="39">38</uint8>
  <uint8 index="40">39</uint8>
  <uint8 index="41">40</uint8>
  <uint8 index="42">41</uint8>
  <uint8 index="43">42</uint8>
  <uint8 index="44">43</uint8>
  <uint8 index="45">44</uint8>
  <uint8 index="46">45</uint8>
  <uint8 index="47">46</uint8>
```

```
<uint8 index="48">47</uint8>
<uint8 index="49">48</uint8>
<uint8 index="50">49</uint8>
<uint8 index="51">50</uint8>
<uint8 index="52">51</uint8>
<uint8 index="53">52</uint8>
<uint8 index="54">53</uint8>
<uint8 index="55">54</uint8>
<uint8 index="56">55</uint8>
<uint8 index="57">56</uint8>
<uint8 index="58">57</uint8>
<uint8 index="59">58</uint8>
<uint8 index="60">59</uint8>
<uint8 index="61">60</uint8>
<uint8 index="62">61</uint8>
<uint8 index="63">62</uint8>
<uint8 index="64">63</uint8>
<uint8 index="65">64</uint8>
<uint8 index="66">65</uint8>
<uint8 index="67">66</uint8>
<uint8 index="68">67</uint8>
<uint8 index="69">68</uint8>
<uint8 index="70">69</uint8>
<uint8 index="71">70</uint8>
<uint8 index="72">71</uint8>
<uint8 index="73">72</uint8>
<uint8 index="74">73</uint8>
<uint8 index="75">74</uint8>
<uint8 index="76">75</uint8>
<uint8 index="77">76</uint8>
<uint8 index="78">77</uint8>
<uint8 index="79">78</uint8>
<uint8 index="80">79</uint8>
<uint8 index="81">80</uint8>
<uint8 index="82">81</uint8>
<uint8 index="83">82</uint8>
<uint8 index="84">83</uint8>
<uint8 index="85">84</uint8>
<uint8 index="86">85</uint8>
<uint8 index="87">86</uint8>
<uint8 index="88">87</uint8>
<uint8 index="89">88</uint8>
<uint8 index="90">89</uint8>
<uint8 index="91">90</uint8>
<uint8 index="92">91</uint8>
<uint8 index="93">92</uint8>
<uint8 index="94">93</uint8>
<uint8 index="95">94</uint8>
<uint8 index="96">95</uint8>
<uint8 index="97">96</uint8>
<uint8 index="98">97</uint8>
```



```
<uint8 index="99">98</uint8>
<uint8 index="100">99</uint8>
<uint8 index="101">100</uint8>
<uint8 index="102">101</uint8>
<uint8 index="103">102</uint8>
<uint8 index="104">103</uint8>
<uint8 index="105">104</uint8>
<uint8 index="106">105</uint8>
<uint8 index="107">106</uint8>
<uint8 index="108">107</uint8>
<uint8 index="109">108</uint8>
<uint8 index="110">109</uint8>
<uint8 index="111">110</uint8>
<uint8 index="112">111</uint8>
<uint8 index="113">112</uint8>
<uint8 index="114">113</uint8>
<uint8 index="115">114</uint8>
<uint8 index="116">115</uint8>
<uint8 index="117">116</uint8>
<uint8 index="118">117</uint8>
<uint8 index="119">118</uint8>
<uint8 index="120">119</uint8>
<uint8 index="121">120</uint8>
<uint8 index="122">121</uint8>
<uint8 index="123">122</uint8>
<uint8 index="124">123</uint8>
<uint8 index="125">124</uint8>
<uint8 index="126">125</uint8>
<uint8 index="127">126</uint8>
<uint8 index="128">127</uint8>
<uint8 index="129">128</uint8>
<uint8 index="130">129</uint8>
<uint8 index="131">130</uint8>
<uint8 index="132">131</uint8>
<uint8 index="133">132</uint8>
<uint8 index="134">133</uint8>
<uint8 index="135">134</uint8>
<uint8 index="136">135</uint8>
<uint8 index="137">136</uint8>
<uint8 index="138">137</uint8>
<uint8 index="139">138</uint8>
<uint8 index="140">139</uint8>
<uint8 index="141">140</uint8>
<uint8 index="142">141</uint8>
<uint8 index="143">142</uint8>
<uint8 index="144">143</uint8>
<uint8 index="145">144</uint8>
<uint8 index="146">145</uint8>
<uint8 index="147">146</uint8>
<uint8 index="148">147</uint8>
<uint8 index="149">148</uint8>
```

<uint8 index="150">149</uint8>
<uint8 index="151">150</uint8>
<uint8 index="152">151</uint8>
<uint8 index="153">152</uint8>
<uint8 index="154">153</uint8>
<uint8 index="155">154</uint8>
<uint8 index="156">155</uint8>
<uint8 index="157">156</uint8>
<uint8 index="158">157</uint8>
<uint8 index="159">158</uint8>
<uint8 index="160">159</uint8>
<uint8 index="161">160</uint8>
<uint8 index="162">161</uint8>
<uint8 index="163">162</uint8>
<uint8 index="164">163</uint8>
<uint8 index="165">164</uint8>
<uint8 index="166">165</uint8>
<uint8 index="167">166</uint8>
<uint8 index="168">167</uint8>
<uint8 index="169">168</uint8>
<uint8 index="170">169</uint8>
<uint8 index="171">170</uint8>
<uint8 index="172">171</uint8>
<uint8 index="173">172</uint8>
<uint8 index="174">173</uint8>
<uint8 index="175">174</uint8>
<uint8 index="176">175</uint8>
<uint8 index="177">176</uint8>
<uint8 index="178">177</uint8>
<uint8 index="179">178</uint8>
<uint8 index="180">179</uint8>
<uint8 index="181">180</uint8>
<uint8 index="182">181</uint8>
<uint8 index="183">182</uint8>
<uint8 index="184">183</uint8>
<uint8 index="185">184</uint8>
<uint8 index="186">185</uint8>
<uint8 index="187">186</uint8>
<uint8 index="188">187</uint8>
<uint8 index="189">188</uint8>
<uint8 index="190">189</uint8>
<uint8 index="191">190</uint8>
<uint8 index="192">191</uint8>
<uint8 index="193">192</uint8>
<uint8 index="194">193</uint8>
<uint8 index="195">194</uint8>
<uint8 index="196">195</uint8>
<uint8 index="197">196</uint8>
<uint8 index="198">197</uint8>
<uint8 index="199">198</uint8>
<uint8 index="200">199</uint8>

```
<uint8 index="201">200</uint8>
<uint8 index="202">201</uint8>
<uint8 index="203">202</uint8>
<uint8 index="204">203</uint8>
<uint8 index="205">204</uint8>
<uint8 index="206">205</uint8>
<uint8 index="207">206</uint8>
<uint8 index="208">207</uint8>
<uint8 index="209">208</uint8>
<uint8 index="210">209</uint8>
<uint8 index="211">210</uint8>
<uint8 index="212">211</uint8>
<uint8 index="213">212</uint8>
<uint8 index="214">213</uint8>
<uint8 index="215">214</uint8>
<uint8 index="216">215</uint8>
<uint8 index="217">216</uint8>
<uint8 index="218">217</uint8>
<uint8 index="219">218</uint8>
<uint8 index="220">219</uint8>
<uint8 index="221">220</uint8>
<uint8 index="222">221</uint8>
<uint8 index="223">222</uint8>
<uint8 index="224">223</uint8>
<uint8 index="225">224</uint8>
<uint8 index="226">225</uint8>
<uint8 index="227">226</uint8>
<uint8 index="228">227</uint8>
<uint8 index="229">228</uint8>
<uint8 index="230">229</uint8>
<uint8 index="231">230</uint8>
<uint8 index="232">231</uint8>
<uint8 index="233">232</uint8>
<uint8 index="234">233</uint8>
<uint8 index="235">234</uint8>
<uint8 index="236">235</uint8>
<uint8 index="237">236</uint8>
<uint8 index="238">237</uint8>
<uint8 index="239">238</uint8>
<uint8 index="240">239</uint8>
<uint8 index="241">240</uint8>
<uint8 index="242">241</uint8>
<uint8 index="243">242</uint8>
<uint8 index="244">243</uint8>
<uint8 index="245">244</uint8>
<uint8 index="246">245</uint8>
<uint8 index="247">246</uint8>
<uint8 index="248">247</uint8>
<uint8 index="249">248</uint8>
<uint8 index="250">249</uint8>
<uint8 index="251">250</uint8>
```

```

<uint8 index="252">251</uint8>
<uint8 index="253">252</uint8>
<uint8 index="254">253</uint8>
<uint8 index="255">254</uint8>
</lut>
<remap reference="33" src_width="6" src_height="4" dst_width="6" dst_height="4">
  <point src_x="0.500000" src_y="0.500000" dst_x="0" dst_y="0" />
  <point src_x="1.000000" src_y="0.500000" dst_x="1" dst_y="0" />
  <point src_x="1.500000" src_y="0.500000" dst_x="2" dst_y="0" />
  <point src_x="2.000000" src_y="0.500000" dst_x="3" dst_y="0" />
  <point src_x="2.500000" src_y="0.500000" dst_x="4" dst_y="0" />
  <point src_x="3.000000" src_y="0.500000" dst_x="5" dst_y="0" />
  <point src_x="0.500000" src_y="1.000000" dst_x="0" dst_y="1" />
  <point src_x="1.000000" src_y="1.000000" dst_x="1" dst_y="1" />
  <point src_x="1.500000" src_y="1.000000" dst_x="2" dst_y="1" />
  <point src_x="2.000000" src_y="1.000000" dst_x="3" dst_y="1" />
  <point src_x="2.500000" src_y="1.000000" dst_x="4" dst_y="1" />
  <point src_x="3.000000" src_y="1.000000" dst_x="5" dst_y="1" />
  <point src_x="0.500000" src_y="1.500000" dst_x="0" dst_y="2" />
  <point src_x="1.000000" src_y="1.500000" dst_x="1" dst_y="2" />
  <point src_x="1.500000" src_y="1.500000" dst_x="2" dst_y="2" />
  <point src_x="2.000000" src_y="1.500000" dst_x="3" dst_y="2" />
  <point src_x="2.500000" src_y="1.500000" dst_x="4" dst_y="2" />
  <point src_x="3.000000" src_y="1.500000" dst_x="5" dst_y="2" />
  <point src_x="0.500000" src_y="2.000000" dst_x="0" dst_y="3" />
  <point src_x="1.000000" src_y="2.000000" dst_x="1" dst_y="3" />
  <point src_x="1.500000" src_y="2.000000" dst_x="2" dst_y="3" />
  <point src_x="2.000000" src_y="2.000000" dst_x="3" dst_y="3" />
  <point src_x="2.500000" src_y="2.000000" dst_x="4" dst_y="3" />
  <point src_x="3.000000" src_y="2.000000" dst_x="5" dst_y="3" />
</remap>
<distribution reference="34" bins="16" offset="0" range="256">
  <frequency bin="0">0</frequency>
  <frequency bin="1">1</frequency>
  <frequency bin="2">2</frequency>
  <frequency bin="3">3</frequency>
  <frequency bin="4">4</frequency>
  <frequency bin="5">5</frequency>
  <frequency bin="6">6</frequency>
  <frequency bin="7">7</frequency>
  <frequency bin="8">8</frequency>
  <frequency bin="9">7</frequency>
  <frequency bin="10">6</frequency>
  <frequency bin="11">5</frequency>
  <frequency bin="12">4</frequency>
  <frequency bin="13">3</frequency>
  <frequency bin="14">2</frequency>
  <frequency bin="15">1</frequency>
</distribution>
<pyramid reference="35" width="24" height="16" format="U008" scale="0.500000"
levels="4">

```

```

<image reference="36" width="24" height="16" format="U008">
</image>
<image reference="37" width="12" height="8" format="U008">
</image>
<image reference="38" width="6" height="4" format="U008">
</image>
<image reference="39" width="3" height="2" format="U008">
</image>
</pyramid>
<pyramid reference="40" width="24" height="16" format="U008" scale="0.500000"
levels="4">
<image reference="41" width="24" height="16" format="U008">
</image>
<image reference="42" width="12" height="8" format="U008">
</image>
<image reference="43" width="6" height="4" format="U008">
</image>
<image reference="44" width="3" height="2" format="U008">
</image>
</pyramid>
<pyramid reference="45" width="24" height="16" format="U008" scale="0.500000"
levels="4">
<image reference="46" width="24" height="16" format="U008">
</image>
<image reference="47" width="12" height="8" format="U008">
</image>
<image reference="48" width="6" height="4" format="U008">
</image>
<image reference="49" width="3" height="2" format="U008">
</image>
</pyramid>
<graph reference="50" name="GRAPH1" >
  <node reference="62" bordermode="CONSTANT">
    <kernel>org.khronos.openvx.sobel_3x3</kernel>
    <parameter index="0" reference="0" />
    <parameter index="1" reference="51" />
    <parameter index="2" reference="52" />
    <borderconst>#00000005</borderconst>
  </node>
  <node reference="64">
    <kernel>org.khronos.openvx.convertdepth</kernel>
    <parameter index="0" reference="51" />
    <parameter index="1" reference="1" />
    <parameter index="2" reference="63" />
    <parameter index="3" reference="61" />
  </node>
  <node reference="66" is_replicated="true">
    <kernel>org.khronos.openvx.add</kernel>
    <parameter index="0" reference="36" replicate_flag="true" />
    <parameter index="1" reference="41" replicate_flag="true" />
    <parameter index="2" reference="65" replicate_flag="false" />
  </node>
</graph>

```

```

    <parameter index="3" reference="46" replicate_flag="true" />
  </node>
  <parameter index="0" node="62" parameter="0" />
  <parameter index="1" node="64" parameter="1" />
  <image reference="51" width="0" height="0" format="VIRT" />
  <image reference="52" width="0" height="0" format="S016" />
  <image reference="53" width="320" height="240" format="VIRT" />
  <image reference="54" width="640" height="480" format="U008" />
  <pyramid reference="55" width="0" height="0" format="VIRT" scale="0.500000"
levels="4" />
  <pyramid reference="56" width="640" height="480" format="VIRT" scale="
0.500000" levels="4" />
  <pyramid reference="57" width="640" height="480" format="U008" scale="
0.500000" levels="4" />
  <array reference="58" capacity="0" elemType="VX_TYPE_INVALID" />
  <array reference="59" capacity="0" elemType="VX_TYPE_KEYPOINT" />
  <array reference="60" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
</graph>
<scalar reference="61" elemType="VX_TYPE_INT32">
  <int32>7</int32>
</scalar>
<scalar reference="63" elemType="VX_TYPE_ENUM">
  <enum>40961</enum>
</scalar>
<scalar reference="65" elemType="VX_TYPE_ENUM">
  <enum>40961</enum>
</scalar>
<delay reference="67" count="3">
  <image reference="68" width="6" height="4" format="U008">
</image>
  <image reference="69" width="6" height="4" format="U008">
</image>
  <image reference="70" width="6" height="4" format="U008">
</image>
</delay>
<delay reference="71" count="2">
  <pyramid reference="72" width="24" height="16" format="U008" scale="0.500000"
levels="4">
    <image reference="73" width="24" height="16" format="U008">
</image>
    <image reference="74" width="12" height="8" format="U008">
</image>
    <image reference="75" width="6" height="4" format="U008">
</image>
    <image reference="76" width="3" height="2" format="U008">
</image>
  </pyramid>
  <pyramid reference="77" width="24" height="16" format="U008" scale="0.500000"
levels="4">
    <image reference="78" width="24" height="16" format="U008">
</image>

```

```

    <image reference="79" width="12" height="8" format="U008">
    </image>
    <image reference="80" width="6" height="4" format="U008">
    </image>
    <image reference="81" width="3" height="2" format="U008">
    </image>
  </pyramid>
</delay>
<delay reference="82" count="2">
  <lut reference="83" count="256" elemType="VX_TYPE_UINT8">
  </lut>
  <lut reference="84" count="256" elemType="VX_TYPE_UINT8">
  </lut>
</delay>
<delay reference="85" count="2">
  <matrix reference="86" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
  </matrix>
  <matrix reference="87" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
  </matrix>
</delay>
<delay reference="88" count="2">
  <convolution reference="89" rows="3" columns="3" scale="1">
  </convolution>
  <convolution reference="90" rows="3" columns="3" scale="1">
  </convolution>
</delay>
<delay reference="91" count="2">
  <distribution reference="92" bins="16" offset="0" range="256">
  </distribution>
  <distribution reference="93" bins="16" offset="0" range="256">
  </distribution>
</delay>
<delay reference="94" count="2">
  <threshold reference="95" elemType="VX_TYPE_UINT8" true_value="255"
false_value="0">
    <binary>0</binary>
  </threshold>
  <threshold reference="96" elemType="VX_TYPE_UINT8" true_value="255"
false_value="0">
    <binary>0</binary>
  </threshold>
</delay>
<delay reference="97" count="2">
  <threshold reference="98" elemType="VX_TYPE_UINT8" true_value="255"
false_value="0">
    <range lower="0" upper="0" />
  </threshold>
  <threshold reference="99" elemType="VX_TYPE_UINT8" true_value="255"
false_value="0">
    <range lower="0" upper="0" />
  </threshold>

```

```

</delay>
<delay reference="100" count="4">
  <scalar reference="101" elemType="VX_TYPE_UINT8">
    <uint8>0</uint8>
  </scalar>
  <scalar reference="102" elemType="VX_TYPE_UINT8">
    <uint8>0</uint8>
  </scalar>
  <scalar reference="103" elemType="VX_TYPE_UINT8">
    <uint8>0</uint8>
  </scalar>
  <scalar reference="104" elemType="VX_TYPE_UINT8">
    <uint8>0</uint8>
  </scalar>
</delay>
<delay reference="105" count="2">
  <remap reference="106" src_width="6" src_height="4" dst_width="6" dst_height=
"4">
    </remap>
  <remap reference="107" src_width="6" src_height="4" dst_width="6" dst_height=
"4">
    </remap>
</delay>
<array reference="108" capacity="10" elemType="VX_TYPE_UINT8">
  <uint8>2 3 4 5 6 7 8 9 10 11 </uint8>
</array>
<array reference="109" capacity="20" elemType="VX_TYPE_CHAR">
  <char>a 13,.,;^-</char>
</array>
<array reference="110" capacity="4" elemType="VX_TYPE_ENUM">
  <enum>-1 0 45057 </enum>
</array>
<array reference="111" capacity="4" elemType="VX_TYPE_DF_IMAGE">
  <df_image>RGB2 U008 VIRT </df_image>
</array>
<array reference="112" capacity="3" elemType="VX_TYPE_KEYPOINT">
  <keypoint>
    <x>0</x>
    <y>0</y>
    <strength>2.300000</strength>
    <scale>6.555550</scale>
    <orientation>0.905900</orientation>
    <tracking_status>5</tracking_status>
    <error>3.545500</error>
  </keypoint>
  <keypoint>
    <x>400</x>
    <y>235</y>
    <strength>5.222200</strength>
    <scale>1.221000</scale>
    <orientation>0.569500</orientation>

```



```

        <tracking_status>8</tracking_status>
        <error>462.500000</error>
    </keypoint>
</array>
<array reference="113" capacity="5" elemType="VX_TYPE_RECTANGLE">
    <rectangle>
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>640</end_x>
        <end_y>320</end_y>
    </rectangle>
    <rectangle>
        <start_x>65</start_x>
        <start_y>32</start_y>
        <end_x>128</end_x>
        <end_y>362</end_y>
    </rectangle>
</array>
<array reference="114" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    <coordinates2d>
        <x>1</x>
        <y>2</y>
    </coordinates2d>
    <coordinates2d>
        <x>0</x>
        <y>55</y>
    </coordinates2d>
</array>
<array reference="115" capacity="6" elemType="VX_TYPE_COORDINATES3D">
    <coordinates3d>
        <x>1</x>
        <y>2</y>
        <z>3</z>
    </coordinates3d>
    <coordinates3d>
        <x>55</x>
        <y>66</y>
        <z>77</z>
    </coordinates3d>
</array>
<array reference="116" capacity="8" elemType="VX_TYPE_INT8">
    <int8>5 0 -3 -8 </int8>
</array>
<array reference="117" capacity="6" elemType="VX_TYPE_INT16">
    <int16>200 100 0 -100 -200 </int16>
</array>
<array reference="118" capacity="6" elemType="VX_TYPE_INT32">
    <int32>200000 100000 0 -100000 -200000 </int32>
</array>
<array reference="119" capacity="3" elemType="VX_TYPE_BOOL">
    <bool>true false true </bool>

```

```

</array>
<array reference="120" capacity="4" elemType="VX_TYPE_SIZE">
  <size>8000 24000 </size>
</array>
<array reference="121" capacity="2" elemType="VX_TYPE_FLOAT64">
  <float64>1235.255660 -563.256700 </float64>
</array>
<array reference="122" capacity="8" elemType="VX_TYPE_UINT64">
  <uint64>9000000000 8000000000 7000000000 6000000000 </uint64>
</array>
<array reference="123" capacity="6" elemType="VX_TYPE_UINT16">
  <uint16>290 100 0 100 260 </uint16>
</array>
<array reference="124" capacity="6" elemType="VX_TYPE_UINT32">
  <uint32>200000 100000 0 100000 200000 </uint32>
</array>
<array reference="125" capacity="2" elemType="VX_TYPE_FLOAT32">
  <float32>1235.255615 -563.256714 </float32>
</array>
<array reference="126" capacity="8" elemType="VX_TYPE_INT64">
  <int64>9000000000 8000000000 -7000000000 -6000000000 </int64>
</array>
<delay reference="127" count="4">
  <array reference="128" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    </array>
  <array reference="129" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    </array>
  <array reference="130" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    </array>
  <array reference="131" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    <coordinates2d>
      <x>1</x>
      <y>2</y>
    </coordinates2d>
    <coordinates2d>
      <x>0</x>
      <y>55</y>
    </coordinates2d>
  </array>
</delay>
<array reference="132" capacity="4" elemType="USER_STRUCT_0">
  <user>32 0 0 0 42 127 0 0 154 153 153 153 153 153 9 64 </user>
  <user>64 0 0 0 1 1 1 1 154 153 153 153 153 153 25 64 </user>
</array>
<delay reference="133" count="2">
  <array reference="134" capacity="4" elemType="USER_STRUCT_0">
    </array>
  <array reference="135" capacity="4" elemType="USER_STRUCT_0">
    </array>
</delay>
<image reference="139" width="64" height="32" format="RGB2">

```

```
<uniform>
  <rgb>#00ff80</rgb>
</uniform>
</image>
</openvx>
```