Managing multichannel and color images using VisionX programming tools.

Multichannel images in Visionx are implemented by overloading the x-dimension of the image array structure. Multichannel images are identified by the x-range of the index being set to an exact multiple of the x-range in the bounding box.

For example, consider the following image declarations:

```c
VisXimage_t im1, im3, krn1, krn3;
VisX3dim_t im3d, im3d3c;
float imbbx[4] = {0, 5, 0, 8};
float krn1bbx[6] = {-2, 3, -2, 3, 0, 0};
float krn3bbx[6] = {0, 5, 0, 8, 0, 0};
float im3dbbx[6] = {0, 5, 0, 8, 0, 4};
float im3d3cbbx[6] = {-1, 4, 0, 8, -1, 2};
VXmakeimage ( &im1, VX_PBYTE, imbbx, 1);
VXmakeimage ( &im3, VX_PBYTE, imbbx, 3);
VXmakeimage ( &krn1, VX_PBYTE, krn1bbx, 1);
VXmakeimage ( &krn3, VX_PBYTE, krn3bbx, 3);
VXmake3dim ( &im3d, VX_PBYTE, im3dbbx, 1);
VXmake3dim ( &im3d3c, VX_PBYTE, im3d3cbbx, 3);
```

The image structures will have the following values:

<table>
<thead>
<tr>
<th>Structure Element</th>
<th>name</th>
<th>im1</th>
<th>im3</th>
<th>krn1</th>
<th>krn3</th>
<th>im3d</th>
<th>im3d3c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounding box</td>
<td>bbx</td>
<td>0,5,0,8,0,0</td>
<td>0,5,0,8,0,0</td>
<td>-2,3,-2,3,0,0</td>
<td>-2,3,-2,3,0,0</td>
<td>0,5,0,8,0,0</td>
<td>-1,4,0,8,0,8,0</td>
</tr>
<tr>
<td>x low index</td>
<td>xlo</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-6</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>x high index</td>
<td>xhi</td>
<td>4</td>
<td>14</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>y low index</td>
<td>ylo</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>y high index</td>
<td>yhi</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

All parameters are the same for a multichannel image except for xlo and xhi, which are multiplied by the number of channels:

For the 3-channel image we have that:

- `im1.u[0][0]` corresponds to the first channel of the first pixel
- `im1.u[0][1]` corresponds to the second channel of the first pixel.
- `im1.u[0][3]` corresponds to the first channel of the second pixel.

The function VXpchan may be used to identify the number of channels in an image if is called with a pixel element as an argument.
For example, VXpchan(im3.list->next) will return the value 3.