VisionX v4 Image Formats: Index

1. Index
2. VisionX V4 Image Formats Overview
3. Conventional Image Formats
4. VisionX Image Formats (two-dimensional)
5. with multiple channels
6. VisionX Image Formats (three-dimensional)
7. VisionX V4 Image Sequence (video)
8. Visionx File Formats (Meta header example)
9. Visionx File Formats (2D image example)
10. VisionX File Formats (3D image example)
11. Notes on 3D and 4D images
12. Graphical image formats
13. VisionX Objects

VisionX v4: Image Formats

- Visionx v4 employs a tagged image file structure (similar to tiff or DICOM) such that a file consists of a set of labeled data elements.
- VisonX extends conventional image formats by providing for a five dimensional model

<frame><z><y><x><c>

<frame> is an image sequence marker

<z> z-dimension index

<y> y-dimension index

<x> x-diemnsion index

<c> number of channels per pixel.

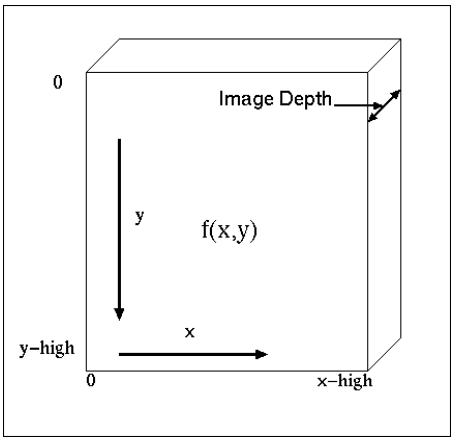
- In addition VisionX provides support for a wide range of pixel data types

- In general only four dimensions are supported and the <frame> and <z> values follow a simple convention (see panel 10)

Conventional Image Formats

Most conventional image formats employ a two-dimensional array with the index origin at the top right corner.

1. The x index increases from left to right
2. The y index increases from top to bottom
3. Grey (black and white) images require a single element per pixel; color images require three elements per pixel (R,G, B)
4. Standard image formats have 1-8 bits per element. Recent extended formats have 16-bits per element

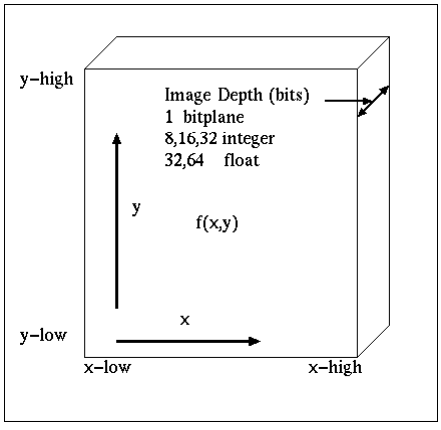


Some conventional image formats: gif, tiff, xbm, png, eps, pdf, ps, jpeg, mpeg, mjpeg, avi, pbm, ppm, pnm, pgm, xpm, etc...

VisionX v4 Image Formats

VisionX image formats employ an extended indexing scheme and allow for additional element typs

1. The x index increases from left to right
2. The y index increases from **bottom to top.** This provides a more convenient mapping to graphical data representations.
3. The image lowest indices (x-low and y-low) may have values other than zero; they may be positive or negative. This provides additional support for sub images that are cut from larger images.

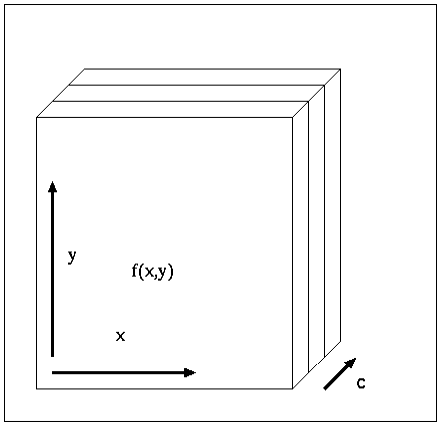


4. VisionX has support for many different element types up to 64-bit floating point
5. VisionX elements may have an arbitrary number of elements per pixel (channels).

VisionX v4 Multi-Channel formats

- VisionX support an unlimited number of channels per pixel

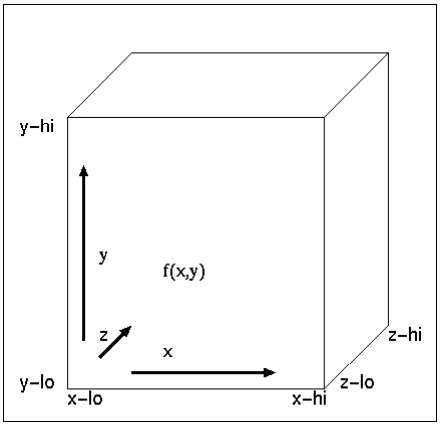
1. This provides direct support for multispectral images that may contain many more than the three conventional colors channels.
2. Support for complex-number pixels is provided by setting the number of channels to two. Either Real and Imaginary or magnitude and Phase.
3. Tensor images are also supported.



VisionX v4 3-dimesional (3D) Images

- VisionX supports up to three dimensions of image indexing

1. At the programming level an additional “z” index is supported
2. In the image file structure a 3D image is represented as a consecutve set of two dimensional (2D) images

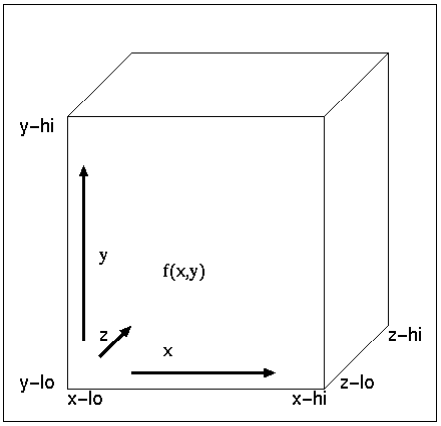


3. The index range of a VisionX image is defined by a six element bounding box x-lo, x-hi, y-lo, y-hi, z-lo, z-hi

VisionX v4 Image Sequence (video)

- A VisionX multi-frame image sequence or a video movie has each frame identified by start and end frame markers.

1. The frame is usually labeled with the value 1.
2. At the programming level a small number of frames may be read at a time facilitating the processing of very large “video” files that may be too large to be read into memory.
3. Typically 3D images have both the z indices set in the bounding box and frame markers between the 2D images



VisionX v4 File Formats

The elemental tagged structure of a VisionX file may be revealed with the vpr command. The following shows the output from vpr for the file mickey-vx

	Tag Name	Size	Contents	Standard Meta Data
Meta Data	[File Name]:	[10]	mickey-vx	Current file name
	[File Command]:	[24]	vxfile of=mickey-vx f=1	Last Command
	[File Machine]:	[6]	win32	Machine and OS ID
	[File User ID]:	0		User ID
	[File Date]:	1280028021		Date in internal format
	[File History]:	[8]	vpnmtovx	Previous commands
Image Data	[Image Frame]:	1		
	[Bounding Box]:	[6]		
		0 640 0 400 0 0		
	[Pixels (byte)]:	[256000]		
	[End Frame]:	1		

File Date and File User ID are scalar values and therefore have no size

VisionX v4 File Formats

The image data for the file shows the standard tags for a 2D image

Image Data

```
[Image Frame    ]: 1
[Bounding Box   ]: [6]
          0        640          0        400          0          0
[Pixels (byte)  ]: [256000]
[End Frame      ]: 1
```

1. The optional image frame markers are included in this file
2. The required bounding box (BBX) indicates a 640 x 400 2D image
   for a 2D image the last two elements of the BBX are usually set to 0
3. The valid image index ranges are 0-639, 0-399 and 0-0
4. The required byte pixels tag contains 640 x 400 = 256000 bytes
   (vpr does not print the contents of very large elements by default

VisionX-V4 File Formats

VisionX v4 File Formats:3D image

```
[Image Frame    ]: 1
[Bounding Box   ]: [6]
          0          3          0          2          0          1
[Pixels (byte)  ]: [6]
   0   0   0   0   0   0
[End Frame      ]: 1
[Image Frame    ]: 2
[Bounding Box   ]: [6]
          0          3          0          2          1          2
[Pixels (byte)  ]: [6]
   0   0   0   0 255   0
[End Frame      ]: 2
[Image Frame    ]: 3
[Bounding Box   ]: [6]
          0          3          0          2          2          3
[Pixels (byte)  ]: [6]
   0   0   0   0   0   0
[End Frame      ]: 3
```

The image data for a 3D image with frame markers is shown

1. Note that frame 1 has a z-index range of 0 this is the **standard VisionX convention**
2. The image size is 3x2x3 pixels
3. All pixel values are shown. One pixel has the max value 255 all others have the value zero

VisionX-V4 File Formats

VisionX v4 Image Formats

The 4D image

- A 4D image in VisionX is represented by a framed sequence of 3D images. That is, each frame contains one 3D image. In the future, an extended version of the 3D image structure could be used for 4D images; however, while such a file can be created now, there are currently no programming tools to directly support this data structure.
- Commands are available for reorganizing the dimensions of 4D images so that they can be processed with standard 2D and 3D commands.

3D Image Structures

- There are two 3D image structures 3D images and image sequences (z-index and frames). At this time, some commands will only operate on one not both of these formats; the best current practice is to use both mechanisms for 3D entities (see panel 10).

VisionX v4 Image Formats

- **3D Graphical Representations**
- 3D graphics in VisionX is based on files containing a set of 3D polygons (preceded by a single 3D bounding box). A set of polygons may be grouped into a single "object" using the object component mentioned above such that a file may contain a collection of "objects". In addition a polygon or set of polygons may be preceded with other attributes such as a face color and a a boundary color. An important feature of the VisionX system is the matching of coordinate systems of both images and polygons which makes possible the mixed rendering of both image and polygon surface data.
- Programming tools are available for the rendering of polygon files that contain just 3 and 4sided polygons. A number of utility commands are available for manipulating 3D polygon files in the above format.

VisionX-V4 File Formats

VisionX v4 Image Formats

- **Objects**
- Objects are collections or groups of components. Objects are delineated by the object component. Components between two object components are considered to comprise of a single object. Object groupings are nested within and do not cross frame boundaries.
- Objects are very useful in grouping say a set of polygons to a single entity. Other attributes (such as color) may then be included in the group.