In an Igor image plot the wave's column data is plotted horizontally while in HDFView and most other programs the row data is plotted horizontally. Therefore, without special handling, a regular image would appear rotated in Igor relative to most programs.

The **HDF5LoadImage** and **HDF5SaveImage** operations handle loading and saving formal images. These operations automatically compensate for the difference in image orientation.

If you are dealing with a regular image, you will use the **HDF5LoadData** and **HDF5SaveData** operations, or **HDF5LoadGroup** and **HDF5SaveGroup**. These operations have a /TRAN flag which causes 2D data to be transposed. When you use /TRAN with HDF5LoadData, images viewed in Igor and in programs like HDFView will have the same orientation but will appear transposed when viewed in a table.

The /TRAN flag works with 2D and higher-dimensioned data. When used with higher-dimensioned data (3D or 4D), each layer of the data is treated as a separate image and is transposed. In other words, /TRAN treats higher-dimensioned data as a stack of images.

Saving and Reloading Igor Data

The **HDF5SaveData** and **HDF5SaveGroup** operations can save Igor waves, numeric variables and string variables in HDF5 files. All of these Igor objects are written as HDF5 datasets.

The datasets saved from Igor waves are, by default, marked with attributes that store wave properties such as the wave data type, the wave scaling and the wave note. The attributes have names like IGORWaveType and IGORWaveScaling. This allows **HDF5LoadData** and **HDF5LoadGroup** to fully recreate the Igor wave if it is later read from the HDF5 file back into Igor. You can suppress the creation of these attributes by using the /IGOR=0 flag when calling HDF5SaveData or HDF5SaveGroup.

Wave text is always written using UTF-8 text encoding. See **HDF5 Wave Text Encoding** on page II-221 for details.

Wave reference waves and data folder reference waves are read as such when you load an HDF5 packed experiment but HDF5LoadData and HDF5LoadGroup load these waves as double-precision numeric. The reason for this is that restoring such waves so that they point to the correct wave or data folder is is possible only when an entire experiment is loaded.

The datasets saved by HDF5SaveGroup from Igor variables are marked with an "IGORVariable" attribute. This allows **HDF5LoadData** and **HDF5LoadGroup** to recognize these datasets as representing Igor variables if you reload the file. In the absence of this attribute, these operations load all datasets as waves.

The value of the IGORVariable attribute is the data type code for the Igor variable. It is one of the following values:

0:	Igor string variable
4:	Igor real numeric variable
5:	Igor complex numeric variable

See also **HDF5 String Variable Text Encoding** on page II-221.

Handling of Complex Waves

Igor Pro supports complex waves but HDF5 does not support complex datasets. Therefore, when saving a complex wave, **HDF5SaveData** writes the wave as if its number of rows were doubled. For example, HDF5SaveData writes the same data to the HDF5 file for these waves:

```
Make wave0 = {1,-1,2,-2,3,-3}                          // 6 scalar points
Make/C cwave0 = {cmplx(1,-1),cmplx(2,-2),cmplx(3,-3)}   // 3 complex points
```

When reading an HDF5 file written by HDF5SaveData, you can determine if the original wave was complex by checking for the presence of the IGORComplexWave attributes that HDF5SaveData attaches to the