

Oma2 Image File Formats

The following image formats can be read by the oma2 software:

.o2d	oma2 binary file format
.dat	oma binary file format (the predecessor of oma2)
.hobj	raw data from ACME's color data camera
.nef	Nikon lossless raw format
.crw	Cannon raw format
.cr2	Cannon raw format
.hdr	High dynamic range format
.csv	Comma separated value text files with the image data in a 2D matrix
.raw	Binary data whose format is specified with the GETBINARYFILE command

In addition, these standard image file formats can be read

.png
.tif or .tiff
.jpg

Files can be opened using the following methods:

Drag the file onto the Status window – this will automatically display the loaded file.

Use the *File->Open...* or *File->Open Recent* menu items – also displays the file.

Type the **GET <filename>** command in the Command window.

When using the **GET <filename>** command, the file prefix (e.g., to specify the file's path) and suffix (e.g., to specify the extension) specified in *oma2->Preferences...* are added to the typed file name.

Decoding .hobj Files

The raw data from ACME's color data camera is encoded in a proprietary format of the commercial HALCON software from MVTec Software GmbH. The oma2 and QtOma software can decode these files and save them in a number of different standard file formats. Several oma2 commands related to the handling of these files are outlined below.

HOBJSSETTINGS <decodeFlag> <demosaicSetting> <clearBadPixelsFlag>

With no arguments, this command gives information on how .hobj files will be read in. If arguments are given, the decodeFlag, demosaicSetting, and clearBadPixelsFlag are specified. A nonzero decodeFlag will cause automatic decoding of .hobj files when they are read in. The demosaicSetting has the following values/meanings: 0 = no demosaiicing done (appropriate for monochrome images); 1 = treat as a color image and demosaic using DOC2RGB 0 1 1 2; 2 = treat as a color image and demosaic using

the bilinear algorithm with the red pixel at 0,0; 3 = treat as a color image and demosaic using the Malvar algorithm with the red pixel at 0,0. If the clearBadPixelsFlag is set, bad pixels previously specified with FINDBAD or READBAD will be set to the average of their eight nearest neighbors (of the same color -- assumes this is a raw color image).

FINDBADPIX Counts [TargetValue] [Passes]

Searches the current image buffer for pixels whose value is more than "Counts" above that of its nearest eight neighbors. Those pixels are tagged as hot pixels. This is best applied to a dark image. If the optional TargetValue is specified, a different algorithm is used that will get values from 1 to 8 neighboring pixels that are within the specified number of counts of the target value. A reasonable TargetValue would be the average A/D offset, e.g., from a dark frame. As currently implemented, the alternate algorithm is not appropriate for raw images from a color camera. The information is saved in an internal buffer. See also CLEARBADPIX, CCLEARBADPIX, WRITEBADPIX, and READBADPIX.

CLEARBADPIX

Sets pixels tagged as bad (e.g., with the FINDBADPIX or READBADPIX commands) to the value of their 8 nearest neighbors. This will not work well if there are contiguous bad pixels. For that, consider using the FILBOX or FILMSK commands. See also FINDBADPIX, CCLEARBADPIX, WRITEBADPIX, and READBADPIX.

CLEARBADPIX

Sets pixels tagged as bad (with the FINDBADPIX command) to the value of their 8 nearest neighbors. This will not work well if there are contiguous bad pixels. For that, consider using the FILBOX or FILMSK commands. See also FINDBADPIX, CCLEARBADPIX, WRITEBADPIX, and READBADPIX.

CCLEARBADPIX

For un-demosaiced color images, sets pixels tagged as bad (e.g., with the FINDBADPIX or READBADPIX commands) to the value of their 8 nearest same-color neighbors. This will not work well if there are contiguous bad pixels. For that, consider using the FILBOX or FILMSK commands. See also FINDBADPIX, CLEARBADPIX, WRITEBADPIX, and READBADPIX.

WRITEBADPIX filename

Save bad pixel data (e.g., from FINDBADPIX or READBADPIX commands) to a text file.

Format is:

of bad pixels

detector_width detector_height

bad_pix1_x bad_pix1_y

bad_pix2_x bad_pix2_y

READBADPIX filename

Read in bad pixel data from a text file (e.g., from a file saved with WRITEBADPIX).

Format is:

of bad pixels

detector_width detector_height

bad_pix1_x bad_pix1_y

bad_pix2_x bad_pix2_y

The hobjExample.zip file contains a folder with three files:

back.hobj A dark image

flame.hobj A flame image

hobjReadAndCorrect.o2m

A text file containing oma commands that illustrate reading in, correcting, and writing out hobj files. The macro file can be loaded into oma2 by dropping it onto the Status window. To run the macro, type rmac (or just mac for no command echo) into the Command window. The text from the file is included below:

```
; Before running the macro, open oma2->Preferences... and set the Get Data Files and
;   Save Data Files prefixes to the path of the hobjExamples folder
;
; set hobj readin parameters to decode hobj files but not demosaic or clear bad pixels
hobj 1 0 0
; set the display saturation value to be 25% of the maximum data value to emphasize
;   hot pixels
dsaturate .25
; get the dark frame data
get back.hobj
display
; save this as a temporary image
stemp dark
; find pixels more than 50 counts from their neighbors
findbadpix 50
;
; save this data for later use
writebadpix badpixfile
; get the raw flame image
get flame.hobj
display
; save this to a temporary image
stemp rawflame
; subtract the dark image
subtmp dark
; clear bad pixels for the undemosaiced image
cclear
display
; demosaic and display
demosaic
display
; save the corrected color image to a temporary image
```

```

stemp correctedflame
; for comparison, display the uncorrected image
gtemp rawflame
demosaic
display
;
; reset the display saturation value to 1.0
dsaturate 1.0
;
; now get the color image from the temporary image buffer and save it to different
formats
;
gtemp correctedflame
display
;
; native oma2 format
save colorFlame
;
; jpg (only 8 bits -- loses information, saves the currently displayed screen image)
savejpg colorFlame.jpg
;
; pdf (only 8 bits -- loses information, saves the currently displayed screen image)
savepdf colorFlame.pdf
;
; csv file, from the image buffer, has full resolution
fwdatm colorFlame ,
;
; 16-bit tif scaled from the 12-bit raw to 16 bits, no negative values
satiffscaled 0 4095 colorFlame16

```