

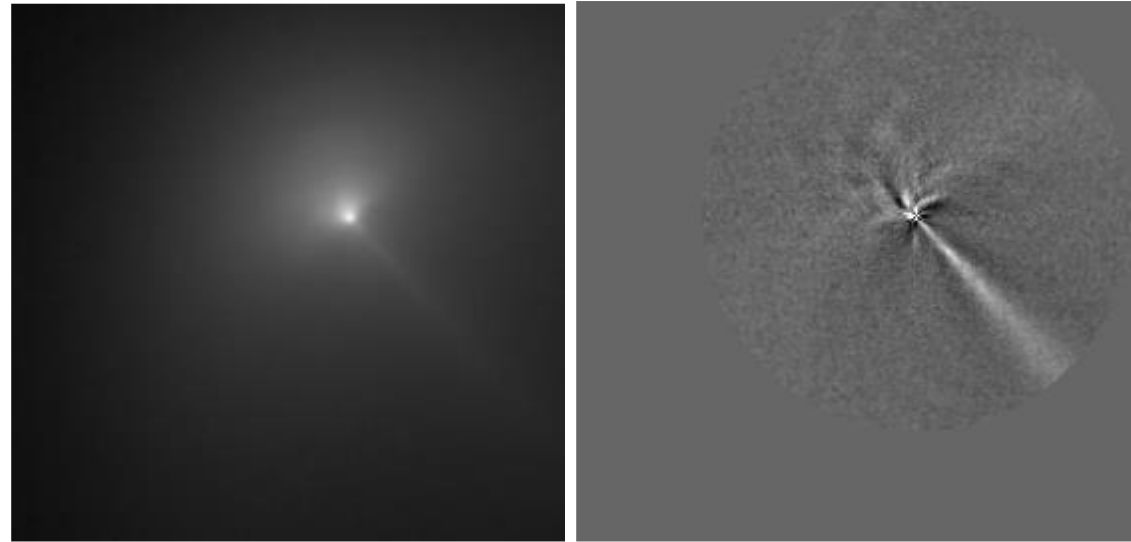
No	Function name	Parameters	Brief Description
	<a href="#">AAVSO_VAR</a>		Unknown.
1	<a href="#">abs</a>		Computes the absolute value of the pixel intensity.
2	<a href="#">acq</a>	[integration time]	Obtains an exposure from the connected camera with certain duration.
3	<a href="#">acq1/acq2/acq3</a>	[integration time]	-//-
4	<a href="#">ACQ_DSLR</a>	[exposure]	Takes an image with a Canon EOS series Digital SLR. The [EXPOSURE] time is given in seconds. The result of the exposure is loaded and displayed. Isn't functional in IRIS 5.59
5	<a href="#">add</a>	[name]	Adds the image in memory to the image designated by name on the disk (the image must be in the current directory).
6	<a href="#">add_max</a>	[name]	<p>Suppose the intensity <math>I_1(x, y)</math> of a pixel in the image <math>I_1</math> at coordinates <math>(x, y)</math> and the intensity <math>I_2(x, y)</math> of a pixel in the image <math>I_2</math> at the same coordinates <math>(x, y)</math>. <a href="#">add_max</a> produces a new image, <math>I</math>, where the intensity of pixel <math>(x, y)</math> is:</p> $I(x, y) = I_1(x, y) \text{ if } I_2 \leq I_1$ $I(x, y) = I_2(x, y) \text{ if } I_2 > I_1$ <p>In other word, a pixel of image <math>I_1</math> is replaced by a pixel of image <math>I_2</math> if the local intensity of <math>I_2</math> is superior to <math>I_1</math>.</p> <p>The usage of <a href="#">add_max</a> is simple:</p> <pre>load I1      (load in-memory the I1 image) add_max I2   (compute substitution) save I       (save the result)</pre> <p><a href="#">add_max</a> is compatible with 16-bits format (gray level images) and 48-bits format (true-color images).</p>
7	<a href="#">add_max2</a>	[name] [number]	<p>Function very similar to <a href="#">add_max</a> but processes many images simultaneously. For example, for stacking images I1, I2, I3, I4, I5, enter the command:</p> <pre>add_max2 I 5</pre> <p>This function is very efficient for constructing long exposure star-trails images. About this method read the excellent paper of <a href="#">Peter Michaud</a> (Gemini observatory) in the Marsh 2004 issue of Sky and Telescope.</p>
8	<a href="#">add_mean</a>	[name] [number]	Calculates the average of a sequence of images.
9	<a href="#">add_min</a>	[name]	Same as <a href="#">add_max</a> but keeps minimum pixel values.
10	<a href="#">add_min_max</a>	[name] [number]	Calculates the average of a sequence of images with rejection of minimum & maximum values.
11	<a href="#">add_min2</a>	[name] [number]	Same as <a href="#">add_min</a> but for sequence of images.
12	<a href="#">add_norm</a>	[name] [number]	Same command as <a href="#">add2</a> but normalizes intensity to 32700 if value of one or many pixels are higher than 32768 after loading.

13	<i>add_norm2</i>	[name] [number]	Same command that <i>add_norm</i> (addition of a sequence of images and normalize the most intense pixel to 32767 if necessary), but the zone where normalize is computed is selected with the mouse. This gives flexibility for some situation cases to avoid saturating a specified part of the image.
14	<i>add_seq</i>	[seq1] [seq2] [result] [coefficient] [number]	Adds matching elements in original sequences and multiplies the result by the factor of [COEFFICIENT].
15	<i>add2</i>	[name] [number]	Adds [NUMBER] images in the sequence of images having the generic name [NAME]. Example:  <i>add2</i> I 3  Adds the images I1.PIC, I2.PIC & I3.PIC.
16	<i>add3</i>	[name] [fwhm] [number]	When using the command <i>register</i> with deep-sky images, a file FWHM.LST is created on your hard drive. In this file, the first column contains an image index and the second one corresponds to the largest FWHM (along either X or Y axis, whichever is greater) of stars within the image. Images indexes appear according to increasing FWHMs. So, it is possible to determine the best images in the series by a simple look at this file. The command <i>add3</i> is essentially the same as <i>add2</i> , except that only the images with FWHM better than [FWHM] will be added. <i>Add3</i> uses the file FWHM to do so. Then you may add only the best images of the series.
17	<i>addnoise</i>	[sigma] [gain (e-/ADU)]	Adds simulated photon noise to each pixel according to its current brightness.
18	<i>addnoise2</i>	[intensity]	Adds uniform noise with selected amplitude.
19	<i>af3</i>	[coef]	Adaptive filtering of the noise in an image. Adaptive filtering consists of adjusting the strength of the filter as a function of local statistical criteria. The filter will be most active where the signal to noise ratio is low. This type of filter reduces the noise while conserving a maximum of details in the image. The parameter [COEF] contains a value that fixes the global strength of the filter. The filter does not act if [COEF]=0, and the filter gets stronger as the value of [COEF] increases. Typically, [COEF] is between 0.1 and 5.  <i>af3</i> uses a zone of 3x3 pixels, centered on the pixel being processed, for the statistical calculation. The <i>af3</i> command is a remarkably effective tool for reducing the noise in an image while preserving a maximum of details. This command (or <i>af5</i> ) is often chosen for this type of application instead of the filters whose action is isotropic (like those provided in the <i>gauss</i> command). Correctly works only with greyscale images otherwise will process only Red channel.
20	<i>af5</i>	[coef]	Same command as <i>af3</i> with a 5x5 array.
21	<i>ang_filter</i>	[xc] [yc] [filter strength]	Performs a low-pass filter on rings centered to $([x_c], [y_c])$ . The algorithm computes the average of pixels in the ring in sectors of [SIZE] degrees. The size of the computation relative to the center $(x_c, y_c)$ is a circular area of dimension [RADIUS]. The <i>ang_filter</i> command general use is to enhance radially structured features in images, such comet or solar jets visible during a total solar eclipse.

```

load HYAKUTA
ang_filter 171 164 108 31
save I
load HYAKUTA
ang_filter 171 164 108 0
sub I 1000
visu 1200 900

```



Result of the angular processing on the right.  
See also [rgradient](#).

22 [anim\\_plot](#)

[data] [out] [x-dim] [y-dim]  
[ymin] [ymax] [title] [number]

Saves a series of graphics images calculated with the data present in sequences of file of generic name [DATE] (the extension of the file is .DAT). These data files are text type and contain two columns (axes X and Y respectively). They are produced for example with command [data\\_anim](#). Graphics are saved in the form of images of generic names [OUTPUT] and of size in pixel [X-DIM]×[Y-DIM]. The range along the Y-axis is defined with the parameters [YMIN] and [YMAX]. The number of data files in the sequence is indicated in the parameter [NUMBER]. The parameter [TITLE] is a character string which will be displayed on the top of each plot. The space character is the symbol "\_".  
Example:

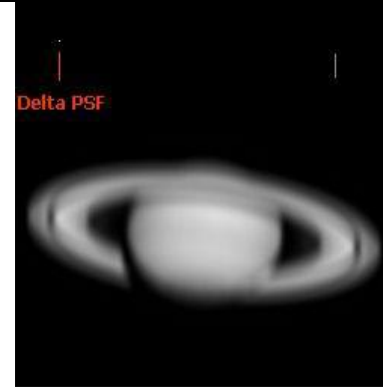
```

anim_plot SPECT GRAPH 300 400 800 20000 It_is_a_spectrum 23

```

			See also command <i>plot2</i> which displays only one graph in a similar way and which makes it possible to test <i>anim_plot</i> . <i>anim_plot</i> is often exploited in partnership with command <i>data_anim</i> for the dynamic study of the spectra.
23	<i>apn</i>	[exposure time]	Unknown.
24	<i>ascale</i>		Enlarge the factor of two the current image (image in memory). This function preserves the intensity per unit of area. This command is useful for precise aperture photometry ( <i>phot</i> and <i>photm</i> commands). Enlarge first the image (apply many time <i>ascale</i> is necessary), then measure the stellar image with large aperture circles.
25	<i>ascale2</i>	[in] [out] [number]	Same as <i>ascale</i> but for a sequence of images.
26	<i>asinh</i>	[alpha] [intensité]	<p>This command is for processing red, green and blue (RGB) composites from three-band astronomical images. <i>asinh</i> stretches the image to show faint objects, while simultaneously preserving the structure of bright objects of the field. The color contrast is boosted by the application a non-linear stretch: the Arc Sinus Hyperbolic function. This method permits to reveal an enormous amount of information: index color of stars, faint nebulae, galaxies having a distinctive color (see for example, the Hubble Deep-Space images of the HST, many times processed with a function very similar to <i>asinh</i>. The arcsinh is a new manner of defining the magnitudes scale, see R. Lupton, <i>Astronomical Journal</i>, 118, 1406-1410. This scale magnitudes has properties very interesting when one applies to color images colors because it boost the colors index of the objects (see R. Lupton &amp; all, <i>PASP</i>, 116,133-137). The colors contrast is very strongly accentuated whereas the noise increase is contained.</p> <p>The parameter [ALPHA] permit to adjust the non-linearity factor. A null value corresponds to a standard linear scale. Typical values are between 0.001 to 0.1. The parameter [INTENSITY] adjust the intensity of the final image. Typical values for this parameter go from 1 to 50 (carry out tests and exploit the visualization thresholds).</p> <p>If the original non-stretched image spans from 0 to 32767 level then in order to use this range without oversaturation the following equation helps to calculate the [INTENSITY] parameter in terms of [ALPHA]:</p> $\text{INTENSITY} = 6441.3 \times \text{ALPHA} + 1.4583$ <p>For example:</p> <p><i>asinh</i> 0.001 7.9</p>
27	<i>base</i>	[input sequence] [output file] [disk #] [generic name] [observatory] [number]	Unknown.
28	<i>best_strehl</i>	[name] [number]	Ordering of the must resolved stellar images in a sequence (Strehl ratio criteria). For details <a href="#">click here</a> .
29	<i>bestof</i>	[name] [number]	Ordering of the must resolved images in a sequence.
30	<i>bestof2</i>	[name] [number]	Same function that <i>bestof</i> but more particularly adapted to objects presenting a high contrast.

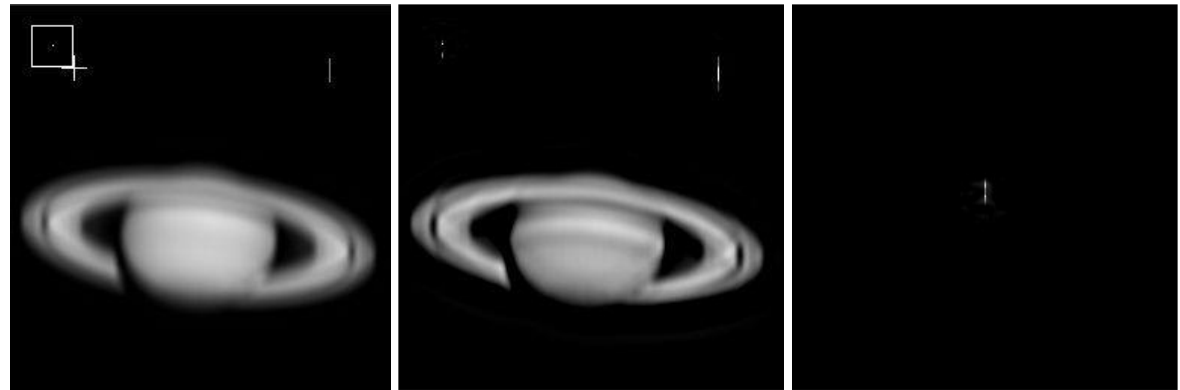
31	<i>bg</i>		Returns the background level of the image in memory.
32	<i>bgnoise</i>		Returns the background noise level.
33	<i>bin_down</i>	[value]	The pixels having an intensity lower than [VALUE] take value 255. The other pixels take value 0. This command becomes handy when one wants to create masks for example for gradient removing procedures. See <i>subgradient</i> , <i>subsky3</i> .
34	<i>bin_up</i>	[value]	The pixels having an intensity higher than [VALUE] take value 255. The other pixels take value 0.
35	<i>binx</i>	[binning factor]	Compute the binning of the along X axis.
36	<i>binxy</i>	[binning factor]	Compute the numerical binning of the in-memory image (sum of the pixels 2x2, 3x3..., like the analogic binning of CCD camera). Same as <b>Binning...</b> command of <b>Geometry</b> menu.
37	<i>binxy2</i>	[in] [out] [binning factor] [number]	Bins a sequence of images.
38	<i>biny</i>	[binning factor]	Computes the binning of the along Y axis.
39	<i>black</i>		To restore just colors of tri-color images sometimes it is necessary to the balance the white. Two new commands make it possible to carry out this operation with precision and speed. The first is the function <i>black</i> which bring the sky background to zero in a zone selected with the mouse, and do it simultaneously on the 3 colors plans in memory. The command returns the background levels of sky in the zone for the three plans. These levels are automatically subtracted and the result is displayed. See also <i>white</i> , <i>white2</i> .
40	<i>black2</i>	[input] [output] [number]	Same for the sequence of images
41	<i>blind</i>	[iteration #] [k-parameter] [regularization]	<p>Since V5.34 IRIS implements a (very) experimental blind deconvolution algorithm. Blind deconvolution consists to find an original image without requiring prior determination of the instrumental PSF. Iris uses an iterative blind deconvolution based on the Wiener filtering and includes non-negativity, support constrains and noise relaxation. The operation is complex, a problem is ill-posed and time consuming. The algorithm can be instable, the convergence not guaranteed and more than one solution is possible for the observed image or/and the PSF. The selection of the initial PSF model is critical. Reserve tests to small sub-images (256x256 or 512x512 pixels). But blind deconvolution is a fascination field!</p> <p>Add at the motion blurred image of Saturn an artificial star consisting to a single point (a delta function). The corresponding PSF is far from the real PSF.</p> <p><i>load</i> BLUR3 <i>put</i> 30 230 8000 <i>save</i> BLUR4</p>



The [RELAXATION] parameter limit the importance of the noise. The typical values of relaxation parameter are between 0 and 2. In most general situation RELAXATION=0. The PSF solution is stored in the image @PSF.

Select the isolated point-like PSF, then enter

*blind* 200 0.2 0



Left — the original blurred image. Middle — result of blind deconvolution (200 iterations) Right — the recovered PSF.

42

*blink*

[in1] [in2] [delay]

Compares two images by displaying them successively and cyclically on the screen. The names of the two images are in the parameters [IN1] and [IN2]. The images may have different sizes. It is recommended to adjust the dynamic and offset of images to minimize flickering effects between the images (*scalecolor* is a good command for this operation).

The blinking time may be adjusted with the [DELAY] parameter that contains the visualization time of an image in milliseconds.


			<p>During blinking it is possible to adjust visualization threshold, color palette, use some processing command like <i>trans</i> for register dynamically the two images Processing concern the image [name1]. For example, try the commands:</p> <pre> blink M51 M51 200 offset 100 trans 1 0 </pre> <p>To stop blinking enter the command: <i>blinkoff</i>. Correctly works only with greyscale images.</p>
43	<i>blink2</i>	[in1] [in2] [in3] [delay]	<p>Same command as <i>blink</i> but for 3 images instead of 2 (that allows sometimes better identification of moving objects). For example:</p> <pre> Blink2 ASTER1 ASTER2 ASTER3 200 </pre>
44	<i>blinkoff</i>		Stops blinking procedure
45	<i>blur</i>	[coef]	Apply a low-pass filtering to an image. The value of the parameter [COEF] is the force of the filter (between 0 and 1).
46	<i>bmp2pic</i>	[in] [out] [number]	Converts sequence 8-bits BMP 8 images in a sequence of FITS or PIC images.
47	<i>border</i>	[x width] [y width]	Adds edges to the existing image with 0 values.
48	<i>capture</i>		Captures the image from web camera.
49	<i>cavi</i>	[AVI file] [output file] [x0] [wide]	Simulates an "electronic slit" for a spectroheliograph instruments by performing the vertical cropping of video-file where [AVI FILE] is the name of file to be processed. [OUTPUT FILE] represents the prefix of the resulting series of images. [X0] is the cropping starting position and the last parameter [WIDE] sets the width of final images.
50	<i>cavi2</i>	[AVI file] [output file] [x0] [wide] [pivot] [angle] [radius]	Unknown.
51	<i>cdg/cog</i>		Returns the coordinates of the center of gravity in the area selected with the mouse.
52	<i>CENTER</i>		Draws cross at the center of into the current image.
53	<i>CFA</i>		<p>Extracts the RGB components from a CCD composed with a Color Filter Array (CFA), like Kodak KAF-0400C CCD. The <b>Iris</b> CFA command is dedicated to a Bayer array with the following aspect:</p> <pre> G R G R B G B G G R G R B G B G </pre> <p>The coordinates of the first red pixel in the bottom left corner of the image have to be set in the variables CFAX &amp; CFAY in the IRIS.INI file (this file is located in the windows directory). Isn't functional in IRIS 5.59 see <i>split_cfa</i>.</p>

54	<i>cfa2pic</i>	[in] [out] [number]	Analog to <i>cfa2rgb</i> but for the sequence of images.
55	<i>CFA2BW</i>		Converts Color-filtered array to a greyscale image.
56	<i>CFA2RGB</i>		Converts Color-filtered array to an RGB image
57	<i>circle</i>	[threshold]	Carries out the binarization of the current image to the pointed threshold then calculates the best circle which passes by contour thus definite. The software returns the coordinates of the center of the circle and its radius. The command is ideal for registration of sun or planetary images.
58	<i>circle2</i>	[threshold]	Computes the center and the radius of a circular object (planet, sun, moon, ...). The radius value is computed for a given intensity threshold in the image. <i>circle2</i> differs from <i>circle</i> command by the method used for identify the object. <i>circle</i> use a dragged rectangular area, <i>circle2</i> defines the rectangle from two clicked points (more useful for large images).
59	<i>clipmax</i>	[old] [new]	All the pixels with an intensity greater than [OLD] are assigned the value [NEW]. Examples:  <i>clipmax</i> 200 0  The pixels whose intensity is greater than 200 are assigned the value 0.  <i>clipmax</i> 4095 4095  The pixels with a value over 4095 are set to 4095. The <i>clipmax</i> command allows you to control the maximum intensity of the pixels in an image. It can be used, for example, when pixels with a high intensity may cause a calculation error in certain processes, or when you wish to reduce a 16 bit image to an 8 bit image.
60	<i>clipmin</i>	[old] [new]	All the pixels with an intensity less than [OLD] are assigned the value [NEW]. The <i>clipmin</i> command is practical for making the contents of an image strictly positive. For example:  <i>clipmin</i> 0 0
61	<i>cmy2rgb</i>	[c] [m] [y] [r] [g] [b]	Convert tricolor images Cyan, Magenta, Yellow (CMY) to tricolor images Red, Green, Blue (RGB).
62	<i>cn</i>	[lambda] [temperature]	Calculates emissivity of a black body at a given wavelength:  $\varepsilon_{\lambda}(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{kT\lambda}} - 1}$ [LAMBDA] is set it angströms and the [TEMPERATURE] in Kelvins. Return answer in units $\frac{W}{m^2 \cdot sr \cdot \text{\AA}}$ .
63	<i>cn2</i>	[entrée] [sortie] [temperature]	Returns numerical value of the integrated Plank formula mentioned above like so:  $I = \int_{\lambda_1}^{\lambda_2} \varepsilon_{\lambda}(\lambda, T)$ [ENTRÉE] asks user to set the name of the .DAT file for input. Data in the file must be in following form:

			<p>5000 10 6000 10 .....</p> <p>First number represents lower bound of the integral and the second sets the area required. Both values are in angstroms. For example, first line will set limits of the integral: (5000,5010). Results will be written in file [SORTIE].DAT. Parameter [TEMPERATURE] is in Kelvins.</p>
64	<i>coastro</i>	[name1] [name2] [mag. adjust (0 or 1)]	Fast version of the command <i>coastros</i> , using a direct projection plane to plane method (see "Fast Direct Plane-to-Plane Coordinate Transformation", D. Makovoz, PASP, 116, 971, Oct. 2004). The command is appropriate only when image scales are similar and for a gnomonic projection.
65	<i>coastro2</i>	[in] [out] [mag. adjust (0 or 1)] [number]	Applied <i>coastro</i> command to a sequence of images. From here: <a href="http://www.astrosurf.com/buil/iris/new553/new553_us.htm">http://www.astrosurf.com/buil/iris/new553/new553_us.htm</a>
66	<i>coastros</i>	[name1] [name2] [mag. adjust (0 or 1)]	The two starting images [NAME1] and [NAME2] are gnomonic projected. The <i>coastros</i> geometrically transform the image [NAME2] into the [NAME1] reference. The result is displayed on the screen and can be saved on the disk. If the Adjust magnitude flag = 0, the intensity of the image 2 is not affected. If the Adjust magnitude flag = 1, the intensity of all pixels of the image 2 is multiplied by a factor to equalize the constant magnitude between image #1 and image #2. This command is ideal for two superimpose wide field deep sky images for search variables objects. For example, image 1 and 2 image can be subtracted to detect the differences, you can use animation function. ( <b>View</b> menu) or <i>blink</i> and <i>blinkoff</i> commands.
67	<i>coastros2</i>	[in] [out] [mag. adjust (0 or 1)] [number]	Applies <i>coastros</i> command to a sequence of images.
68	<i>col2bw or COLOR2BW</i>	[in] [out] [number]	Converts a 48-bits true-colors image sequence into a black and white sequence (simple addition of the RGB layer). For a single file use <i>tri2mono</i> .
69	<i>Composit Or composit0</i>	[name] [sigma] [#iter] [flag max] [#images]	<i>Composit</i> is a powerful command to perform automatic combination of a sequence of images that were registered before. The simplest way to combine the images is of course to add them. <i>Composit</i> proceeds in that way, but will reject the pixels that have values significantly bad, i.e. for which the difference with respect of the mean of the values in all the images is greater that [SIGMA] times the standard deviation of the values. Moreover, the process may be iterative: at each iteration, a new analysis of the pixels statistics is made with the left ones. This method is called sigma-clipping. To be very efficient, it is necessary to have a large number of images to combine (at least 5). Try [SIGMA] values between 1.5 & 5. The [NAME] parameter contains the generic name of the sequence, and the [#IMAGE] parameter contains the number of images in the sequence.

		<p>The [SATURATE] parameter is a flag. If saturate=1 the max intensity of the coadded image is normalized to 32700 if the level is upper to 32767. This control is not realized if saturate=0. Example:</p> <p><i>Composit</i> M33- 2.5 2 0 7</p> <p>Combines the images M33-1.PIC, M33-2.PIC...M33-7.PIC with a rejection level of 2.5 sigma. Two iterations are performed.</p> <p>The <i>Composit</i> command is a powerful tool that gathers efficiency of the simple addition of images in terms of signal to noise ratio, and the power of median combination in terms of rejection of aberrant pixels (cosmic rays, satellites, ...).</p>
70	<i>composit2</i>	<p>[name] [flag max] [#images]</p> <p><i>Composit2</i> method use a robust average image using a continuous adaptive weighting scheme that is derived from the data themselves — see Artificial Skepticism Stacking algorithm — Stetson 1989, V Advanced School of Astrophysics [Univerisidade de Sao Paulo], p.1. See also: <a href="http://archive.stsci.edu/hst/wfpc2/pipeline.html">http://archive.stsci.edu/hst/wfpc2/pipeline.html</a> and <a href="http://archive.eso.org/archive/hst/wfpc2_asn/3sites/WFPC2_Newsletter.pdf">http://archive.eso.org/archive/hst/wfpc2_asn/3sites/WFPC2_Newsletter.pdf</a>.</p> <p>The given parameters are only the generic name of the input image sequence, the normalized flag (0 or 1, see <i>composit</i>) and the number of input image (an unlimited number).</p> <p>The weights of the pixel values are computed by the equation:</p> $w_i = \frac{1}{\sigma_i^2} \frac{1}{1 + \left(\frac{ r_i }{\sigma_i}\right)^2}$ <p>where <math>w_i</math> is the weight of the <math>i^{th}</math> pixel value, <math>\sigma_i</math> is the sigma of the <math>i^{th}</math> pixel in the stack, derived from the readout noise and camera gain. The <math>r_i</math> term, which is the residual between the current average pixel value and the value of the <math>i^{th}</math> pixel, is computed at each iteration. This version of <i>composit2</i> uses classical and internally coded value for CCD readout noise and camera gain (noise of 15 electrons RMS of 2 e-/ADU). <i>Composit2</i> is a simple command to use and efficient for bad pixels rejection.</p> <p>Important, before use commands like <i>smedian</i>, <i>composit</i> and <i>composit2</i> it is necessary to have the same sky background level for all the images of the sequence. Use the command <i>noffset</i> if is not the case (or <i>noffset2</i> for select a specific region for the harmonization of the sky level). Similar, if the exposure time is not the same the images should be scaled before stacking (<i>mult</i>, <i>mult2</i>, <i>ngain2</i> commands for example).</p> <p>The choice of the most optimal combining algorithm will depend on the nature of the data and on the exposure type. For produce a clean flat-field or a master dark frame the appropriate command is</p>

			<p><i>smedian</i> (or <i>smedian2</i>). For deep-sky imaging the classical sigma-clipping is a good choice for the best conservation of signal to noise (the median lose 30% in signal to noise typically relative to simple sum and the <i>composit/composit2</i> commands). The <i>composit2</i> command is now a useful and fast alternative to the sigma-clipping scheme.</p>
71	<i>COMPUTE</i>		<p>Draw a rectangle around a star with the mouse. Then type <i>compute</i>. Iris reads the files POLX.LST &amp; POLY.LST created by <a href="#">Astrometry/Photometry</a> dialog box and returns the equatorial coordinates and the magnitude of the star.</p>
72	<i>compute_trichro1</i>	[master] [r] [g] [b] [size] [select nb.] [total nb.]	<p>This command carries out the automatic processing of RGB images of planets. It connects commands <i>bestof</i>, <i>select</i>, <i>pregister</i>, <i>add_norm</i>, for the three colors channels. At the end of the processing the trichromatic image appears on the screen (you can then save it on the disc with command <i>savebmp</i>, <i>savejpg</i>, ... or adjust its chromatic balance with the command <b>White balance...</b> from menu <b>View</b> for example).</p> <p>Moreover, several sequences of images are created on the hard disk. The sequence @r1, @r2... @rn (<i>n</i> is the total number of image treated) contains the registered images of the red color channel and sorted by order of decreasing spatial resolution. The sequences @g1, @g2... @gn and @b1, @b2... @bn contain same information for the green and blue channels.</p> <p><i>compute_trichro1</i> also automatically produces the 3 images @r, @g, @b (without indices) which represent the addition of the best <i>n'</i> images for the 3 colors channel (<i>n'</i> parameters being provided by the operator).</p> <p>Command <i>compute_trichro1</i> uses function <i>pregister</i> to carry out the registration of the images (centering of the images of a sequence compared to the first image of this sequence), it is rather intended for the images being able not to have a contour of revolution, the such planet Saturn or lunar surface. Moreover, the use of <i>pregister</i> imposes that the operator enters in parameter of <i>compute_trichro1</i> the size of the window for the calculation of registration (it must be equal to a power of two because the technique used for registration is the intercorrelation in the Fourier space). The parameters are:</p> <p>[MASTER] is the generic name of a sequence of image from which IRIS will make sorts it better images (as command <i>bestof</i> and will calculate the parameters of registration for the 3 colors plans. The master sequence of images must contain images well exposed posed if possible and well resolved. Generally, in the case of the use of a Webcam camera, one will choose the images corresponding to the plan of green color.</p> <p>[R], [G], [B] is the generic names of red, green and blue channel images respectively.</p> <p>[SIZE] is the size of the zone for the calculation for registration (choose among values 128, 256, 512 for example).</p> <p>[SELECT NB.] is the number of images added during the final composite. It is a number equal or lower than the total number of images to treat. This value is dependent on the degree of turbulence.</p>

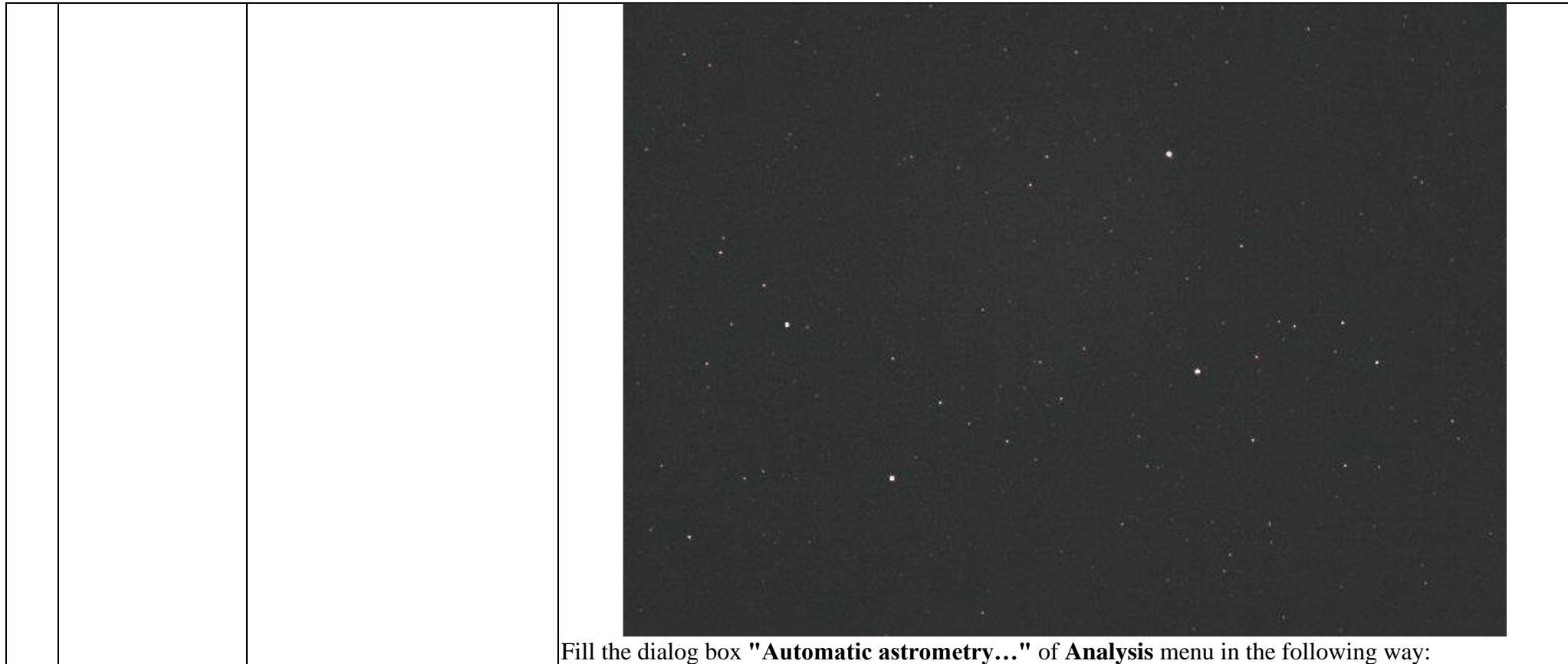
			<p>For example, for a sequence of 200 input images it is not abnormal to add only 50 images with final (they will be 50 best images).  [TOTAL NB.] is the total number of images to be treated.  Let us suppose that you extracted from a film AVI (command <b>Conversion AVI...</b> from <b>File</b> menu) or functions of Webcam acquisitions from IRIS (<b>Webcam</b> menu) from the sequences from 300 images whose generic names are R, G and B for respectively the plans colors red, green and blue. You make then for example:</p> <p><i>compute_trichro1</i> G R G B 128 40 300</p> <p>The size of the window for calculation (here 128) must be higher (but not much higher, if not calculation can be very long) than the diameter of the planet disc of planet. It is necessary moreover for run the command to surround planet by a rectangle while dragging with the mouse (press left button).  It is significant that the rectangle thus defined either centered on the center of the planetary disc. It is necessary moreover that its dimension makes it possible to include the images of planet of the beginning and at the end of the sequence. The size and the position of the selection box are less critical in the case of the lunar images (but for this type of images the three-color process in general presents well little interest and it is by far preferable to treat monochromic images).  Calculation can be relatively long. If a problem appears in the course of processing it is always possible to stop this one while click on the stop key  of the bar of tools.</p>
73	<i>compute_trichro2</i>	[master] [r] [g] [b] [threshold] [select nb.] [total nb.]	<p>This command is very similar to <i>compute_trichro1</i> except it uses of function <i>cregister</i> for the registration of the images instead of <i>pregister</i>. Function <i>cregister</i> determines the position of planet by adjusting a circle on the circumference of the limb. It is necessary to provide to <i>compute_trichro2</i> the value of the threshold of intensity from which the calculation of the circle is carried out (<a href="#">click here</a> for more information). Taking into account these characteristics, it is necessary to reserve the use of this command to planets having a good symmetry of revolution (Jupiter and generally Mars).  The parameter [THRESHOLD] define the level of threshold which will be used to adjust a circle around planet. For example:</p> <p><i>compute_trichro2</i> G R G B 80 60 300</p> <p>Tip: to carry out images having a good chromatic balance it should be checked that the level of the sky background is homogeneous between the three colors plan. It is not rare with the images coming from the Webcam camera that the level of the sky is higher in blue than in the red and the green. Thus to bring back the level of the bottom of sky in the sequence blue you will make:</p> <p><i>Noffset3</i> B B 0 300</p>

			<p>after having to draw a selection box in one of the images of the sequence. You can also use for that the <b>Offset normalization of the of a sequence...</b> command of <b>Processing</b> menu after having to notch the option <i>On a zone</i>.</p> <p>To note that command <i>compute_trichro2</i> is appreciably faster than <i>compute_trichro1</i> and if you have the choice, it is <i>compute_trichro2</i> which you will use preferably.</p>
74	<i>convert_index</i>	[in] [out] [number]	Converts image name format in0001, in0002, in0003, ... to out1, out2, out3, ...
75	<i>convertbmp</i>	[input] [output] [number]	Converts an 8-bits BMP input sequence [in] into an output sequence [out] in the current file format (fix the File type in the <b>Settings</b> dialog box - see the <b>File</b> menu). The number of images in the input sequence is [number]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.
76	<i>convertbmp24</i>	[input] [output] [number]	Converts a 24-bits BMP input sequence [IN] into three output generated sequences [R] [G] [B] in the current file format (fix the File type in the Settings dialog box - see the File menu). The number of images in the input sequence is [NUMBER]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.
77	<i>convertbmp24bw</i>	[input] [output] [number]	Converts a 24-bits BMP input sequence [IN] into the mean of the RGB planes and copy the result in the output sequence [OUT]. The number of images in the input sequence is [NUMBER]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.
78	<i>convertdsi</i>		Converts a raw image of the camera Meade DSI already in memory into an image color.
79	<i>convertdsi2</i>	[in] [out] [number]	Converts a sequence of RAW image Meade DSI (generic name [IN]) into a sequence of color 48-bits images (generic name [OUT]). The number of images in the sequence is [NUMBER]. The geometrical transformations are not carried out by this command.
80	<i>convertsx</i>	[in] [out] [number]	Converts a sequence of unsigned 16-bits images into a series of signed 16-bits images compatible with Iris (dynamic range between 0 and 65535). The level of the pixels is multiplied by 0.5 to respect dynamics 0...32767.
81	<i>convertsx2</i>	[in] [out] [number]	Converts a sequence of unsigned 16-bits images into a series of signed 16-bits images compatible with Iris. The level of the pixels is not modified, but the images are truncated for intensities higher than 32767. The final level lies between 0 and 32767.
82	<i>convertsx3</i>	[in] [out] [number]	Converts a sequence of unsigned 16-bits images into a series of signed 16-bits images compatible with Iris. Value 32767 is subtracted from all the pixels. The final level lies between -32768 and 32767.
83	<i>converttiff</i>	[input] [output] [number]	Convert an 8-bits uncompressed TIFF input sequence [IN] into an output sequence [OUT] in the current file format (fix the File type in the <b>Settings</b> dialog box - see the <b>File</b> menu). The number of images in the input sequence is [NUMBER]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.

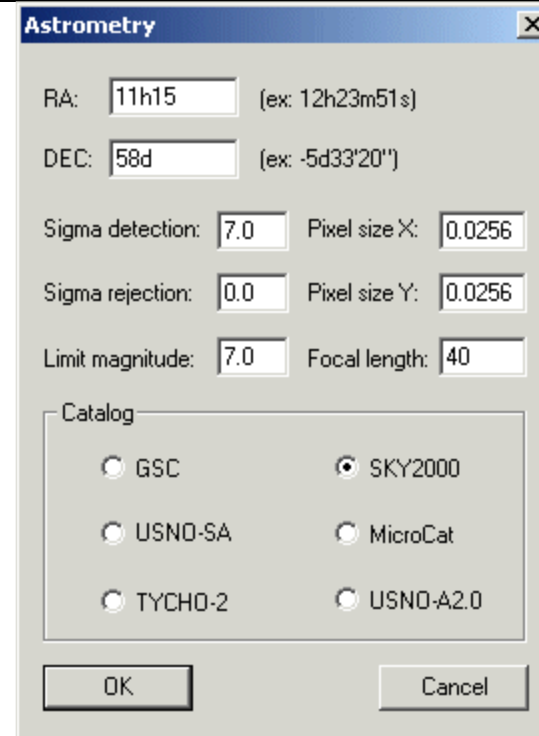
84	<i>converttiff24</i>	[input] [r] [g] [b] [number]	Converts a 24-bits uncompressed TIFF input sequence [IN] into an output sequence of RGB planes in the current file format (fix the File type in the Settings dialog box — see the <b>File</b> menu). The number of images in the input sequence is [NUMBER]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.
85	<i>converttiff24bw</i>	[input] [output] [number]	Converts a 24-bits uncompressed TIFF input sequence [IN] into the mean of the RGB planes and copy the result in the output sequence [OUT]. The number of images in the input sequence is [NUMBER]. You can set the base of the first index in the input sequence (see the <i>setbase</i> command). The first index of the output sequence is always 1.
86	<i>copy</i>	[name] [x1] [y1] [x2] [y2]	Copy in the current image the portion of the image [NAME] in the disk delimited by the coordinates $(x_1, y_1) - (x_2, y_2)$ .
87	<i>copyadd</i>	[in] [out] [total number] [nb_add]	This command adds the [NBADD] first images with the generic name [IN] and saves the result with the name [OUT] with the index 1. Then the result of adding the next [NB_ADD] images of generic name [IN] is saved with the name [OUT] with the index 2 and so on up to the image [IN] with the index [NUMBER]. One of the interests of this command is that the acquisition time put in the header of the added images is the barycenter of the individual images. As a consequence, the dating accuracy contained in the input images is transferred to the added images.
88	<i>copyfwhm</i>	[in] [out] [fwhm] [number]	Copy images of the sequence [IN] to a new sequence [OUT], but select only images where the FWHM is inferior to [FWHM] parameter. The command returns the selected image number. It is necessary to apply before the <i>register</i> command (see also <i>add3</i> ). <i>Copyfwhm</i> is perfect for isolating good images, before compositing, for example:  <i>copyfwhm</i> M51- I 1.6 12
89	<i>copymed</i>	[in] [out] [total number] [nb_median]	Same command as <i>copyadd</i> , but uses median averaging instead of adding the [NB_MEDIAN] images.
90	<i>copyx</i>	[x_ori] [x_dest]	Copy the column of position $[x_{ori}]$ to the column $[x_{dest}]$ .
91	<i>copyy</i>	[y_ori] [y_dest]	Copy the line of position $[y_{ori}]$ to the line $[y_{dest}]$ .
92	<i>coregister</i>	[name1] [name2]	Performs geometric transforms on [IN2] so that it can be superimposed on [IN1]. The output file is the new [IN2] file. Example:  <i>coregister</i> N266_1 N266_2  For a typical application, <a href="#">click here</a> . Also it is possible to change the order of the polynomial for registration by <i>setregister</i> function. See also <i>setspline</i> , <i>setnbsstar</i> , <i>setfindstar</i> , <i>setmatch</i> functions.
93	<i>coregister2</i>	[in] [out] [number]	Same function as <i>coregister</i> but for sequences.
94	<i>COREGISTER3</i>	[nom1] [name2] [size of research area]	Registration of the images [NAME1] and [NAME2] by applying a translation $(dx, dy)$ calculated automatically starting from three zones of the images.

95	<i>COREGISTER31</i>	[nom1] [name2] [size of research area]	Registration of the images [NAME1] and [NAME2] by applying a spline translation ( $dx, dy$ ) calculated automatically starting from three zones of the images.
96	<i>coregister4</i>	[in] [out] [wide] [number]	Registration of a set of images by applying a translation ( $dx, dy$ ) calculated automatically starting from three zones of the images.
97	<i>cosme</i>	[file list]	<p>Applies the local mean to a set of pixels on the in-memory image (cosmetic correction). The coordinates of these pixels are in an ASCII file [LIST FILE]. <i>Cosme</i> is adapted to correct residual hot and cold pixels after preprocessing (the coordinate of this points is constant for a given CCD). For example, if the goal is to correct pixels of coordinates:</p> <pre>(120,310) (9,501) (232, 140) and line (100) and column (20)</pre> <p>Create the following text file (use your favorite word processing):</p> <pre>P 120 310 P 9 501 P 232 140 L 100 0 C 20 0</pre> <p>Save under the name CORRECT.LST (for example, but the extension .LST is important). The file is saved in the working path (see Settings... dialog box of the File menu). Now, load the image to correct, then:</p> <pre><i>cosme</i> correct</pre> <p>You can correct up to 500 pixels (i.e. 500 lines max in the .lst file).</p>
98	<i>cosme_cfa</i>	[file list]	Same function that <i>cosme</i> but applying to RAW images (Bayer matrix images). All the types of RAW files recognized by Iris can be treated (Canon, Nikon...). The processing is distinct for red, green and blue pixels of CFA matrix). The file whose name is given in argument contains the list of the hot pixels. Those can be found automatically with command <i>find_hot</i> .
99	<i>cosme_cfa2</i>	[input] [output] [file list] [number]	Same function that <i>cosme_cfa</i> , but applying to a sequence of images.
100	<i>cosme2</i>	[input] [output] [file list] [number]	<p>Same command as <i>cosme</i> but for a sequence of [NUMBER] images. [INPUT] is the generic name of the input sequence and [OUTPUT] is the generic name of the corrected sequence. For example:</p> <pre><i>cosme2</i> M51- I correct 4</pre> <p>processes the sequence: M51-1, M51-2, M51-3, M51-4 and produces the sequence I1, I2, I3, I4</p>

101	<i>cosmic</i>	[threshold]	Trying to search for cosmic rays in the image. Applicable only to greyscale images.
102	<i>cosmic2</i>	[in] [out] [threshold] [number]	Same as <i>cosmic</i> but for a sequences.
103	<i>count_down</i>	[value]	Returns the number of pixels having an intensity lower than [VALUE].
104	<i>count_up</i>	[value]	Returns the number of pixels having an intensity lower than [VALUE].
105	<i>cpu</i>	[time (s)]	Measures the frequency of clock of the CPU.
106	<i>cregister</i>	[in] [out] [threshold] [number]	Carries out the registration of a sequence of image from the coordinates of the center of a circle determined from a contour defined for the intensity [THRESHOLD] in each image. <a href="#">Click here</a> for an example.
107	<i>crisp/sharp</i>	[coef]	Applies a high-pass filtering to an image. The value of the parameter [COEF] is the force of the filter (between 0 and 1). The synonymous name is <i>sharp</i> . See <a href="http://www.astrosurf.com/buil/iris/tutorial1/doc5_fr.htm">http://www.astrosurf.com/buil/iris/tutorial1/doc5_fr.htm</a>
108	<i>cvign</i>	[a] [b] [c]	Apply the vignetting correction with the given polynomial coefficients obtained from command <i>evign</i>
109	<i>cvign2</i>	[image to correct] [flat-field]	[IMAGE] is the name of the image file to correct. [FLAT] is the file-name of a uniform screen taken in the same conditions (same lens, same aperture). Iris adjusts a second order polynomial function ( $r_1 = ar^2 + br + c$ ) onto the "flat" image and applies the inverse polynomial function to the image to be corrected. The result is displayed on the screen after processing.
110	<i>d_alpha</i>	[ra] [intensity]	Displays a great right ascension circle in the in-memory image after an astrometric reduction. [alpha] is the right ascension and [intensity] is the intensity of the drawing.
111	<i>d_delta</i>	[dec] [intensity]	Displays a declination circle in the in-memory image after an astrometric reduction. [DELTA] is the declination angle and [INTENSITY] is the intensity of the drawing. Here an image of the head of Ursa Major constellation realized with a EOS 20D and a Canon zoom lens used at the 40 mm focal length (20 seconds exposure @ ISO400). The image is downscaled to 25% of its original.



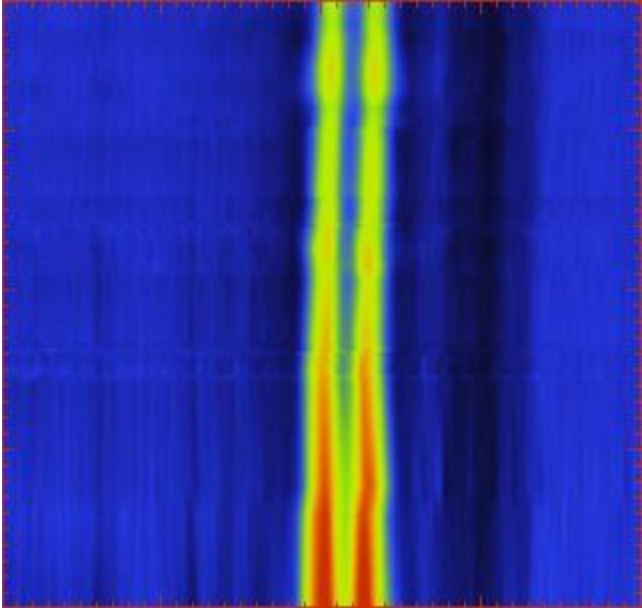
Fill the dialog box "**Automatic astrometry...**" of **Analysis** menu in the following way:



The catalogue SKY2000 is used considering the image is of wide-field type. The approximate equatorial coordinates of the center of images are provided (Alpha = 11h15m, Delta = +58°). The pixels size of EOS 20D detector is 6.4 microns, but taking into account the reduction factor of the image, their equivalent size is  $4 \times 6.4 = 25.6 \text{ microns} = 0.0256 \text{ mm}$ . Click OK to carry out the automatic reduction. IRIS automatically finds the correspondence between stars present in the image and those of the catalogue. At this stage the equatorial co-ordinates of an object in the image can be obtained by surrounding this one with a rectangle, then by typing *compute* command from the console or by run "**Compute**" command since the contextual menu. Of course stellar catalog and path to it should be set beforehand in menu **File > Settings**.

112	<i>data_anim</i>	[input] [output] [x1] [x2] [step]	<p>Powerful function allowing for example to interpolate at the same time along wavelength and temporally a whole of spectral data in order to carry out an animation. The parameter [IN] is the name of a text file having the extension LST. It contains two columns. The first give a spectral file name having extension DAT, the second is the date of acquisition of these spectra in Julian day or reduced Julian day. Here contents characteristic of an input file:</p> <pre>290601 245678.345</pre>
-----	------------------	-----------------------------------	---

			<p>300701 245689.446  220801 245693.945  .....</p> <p>It indicates that spectral profile 290601.DAT was acquired the day Julian 245678.345, that spectral profile 300701.DAT was acquired the day Julian 245689.446, that spectral profile 220801.DAT was acquired the day Julian 245693.945, and so on.</p> <p>The parameter [OUT] is the name of a text file having the extension LST. It contains two columns. The first give the spectral file name (DAT extension) which will be interpolated, the second is the date of the files interpolated in Julian day or reduced Julian day. Here contents characteristic of a file of an output file:</p> <p>R1 245679.0  R2 245690.0  R3 245691.0  .....</p> <p>It indicates that command <i>data_anim</i> must produce spectral profiles in files of names R1.DAT, R2.DAT, R3.DAT, respectively for the dates in Julian days 245679.0, 245690.0, 245691.0. The parameters [X1] and [X2] define an interval in wavelength for the interpolation which an interpolation step of [STEP].</p> <p>The mode of interpolation along the temporal axis is linear. One uses an interpolation spline along the wavelengths axis.</p> <p>Once the interpolated profiles, you can display this in a graphic form with the software of your choice then to create animations of the evolution of the spectrum according to time. It is possible also to produce an image of the dynamic spectrum starting from command <i>data2image</i>.</p>
113	<i>data_anim2</i>	[input] [output] [date1] [date2] [date step]	
114	<i>data_bin</i>	[in] [out] [binning factor]	Carries out the binning of a data file, for example a file resulting from the <a href="#">photometric analysis</a> , in order to increase the signal to noise ratio. [BINNING FACTOR] is the factor of binning (typically value: 2 to 4).
115	<i>data_reject</i>	[in] [out] [coef]	Analyzes a data file (file DELTA.DAT coming from the automatic photometric analysis for example) and eliminates the points deviating of more than [COEF]. sigma of the average value (sigma is the standard deviation of the distribution). [IN] is the name of the input file (it must have extension DAT on the disc). [OUT] is the name of the output text file.
116	<i>data_resample</i>	[in] [out] [step]	Re-sampling points of a data file (use of the spline interpolation). Useful command for example to represent data spectral or an intensity distribution curve with an integer step.
117	<i>data_stat</i>	[data file]	Returns statistical information on the data file [DATA_FILE].

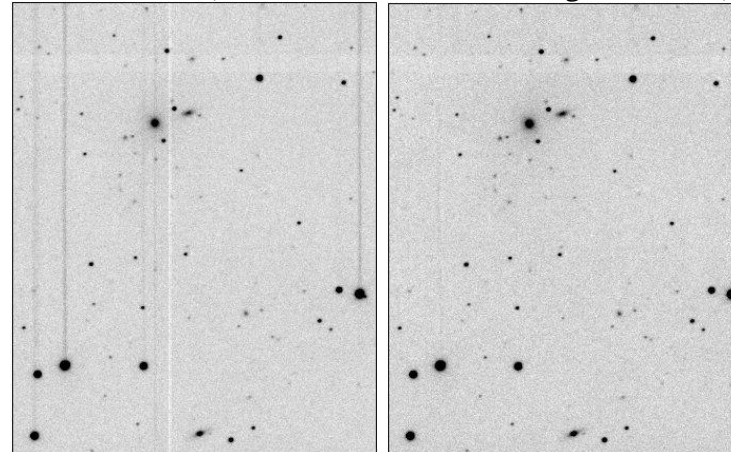
118	<i>data2image</i>	[input] [coefficient] [number]	<p>Creates an image in memory starting from a sequence of text file having extension DAT. The contents of the first file are used to produce the first line of the image. The contents of the second file build the second line of the image, and so on. The text files must contain two columns of real data. The image is built with information of the second column. With final, the size of the image along axis X is equal to the number of lines contained in files DAT and the size along the axis Y is equal to the number of file DAT. Parameters are:</p> <p>[INPUT] is the generic name of files DAT  [COEFFICIENT] is a multiplicative parameter by which one multiplies the second column of files DAT before assigning them to the pixels of the image in memory.  [NUMBER] is the number of files DAT</p> <p>The typical application of <i>data2image</i> is the creation of the image of a dynamic spectrum which shows the evolution of the spectral profile according to time. Here a characteristic image:</p>  <p>Each line of this image is an individual spectrum (here the region of the H-alpha line in the spectrum of Omega Orion). The interval of time following the vertical axis is regular. If needed, the spectral profiles are interpolated with command <i>data_anim</i>.</p>
119	<i>data2pic</i>	[data file name]	<p>Converts a text ASCII file with two columns into an image whose axis Y contains the values (standardized to 32767) contained in the second column of the file. This command is useful to import in IRIS spectral data.</p>
120	<i>DATE</i>		<p>Returns the current date.</p>
121	<i>date2jd</i>	[day] [month] [year]	<p>Converts date to Julian</p>

*date2jd* 27.76 08 2001

122 *deconvflat*

[coef]

Counters the smearing effect in an image exposed without obturator. [coefficient] is the ratio between the reading time of a CCD line and the exposure time.  
The readout time in a low noise CCD camera is relatively long. In case of the Audine camera the time needed for digitizing the image is 5 seconds in binning 2x2 and 15 seconds in binning 1x1 mode. During this time the CCD is still exposed to light, the objects in the field produce a parasitic artefact manifesting itself as trails. This phenomenon is called smearing. The traditional solution consists of preventing the light from reaching the CCD during reading, by mounting an electromechanical shutter, controlled by the camera electronics. The following images show an alternative that doesn't use a shutter at all. In this case treating the images afterwards with a compensating routine is sufficient to remove the smearing effect (the *deconvflat* command in Pisco or QMiPS32). This procedure is extremely effective in the vast majority of cases (it only fails locally in those parts of the image where a saturated star is in the field). It is economical (important in the Audine philosophy) and totally eliminates mechanical trouble (sometimes a shutter is a fragile element).

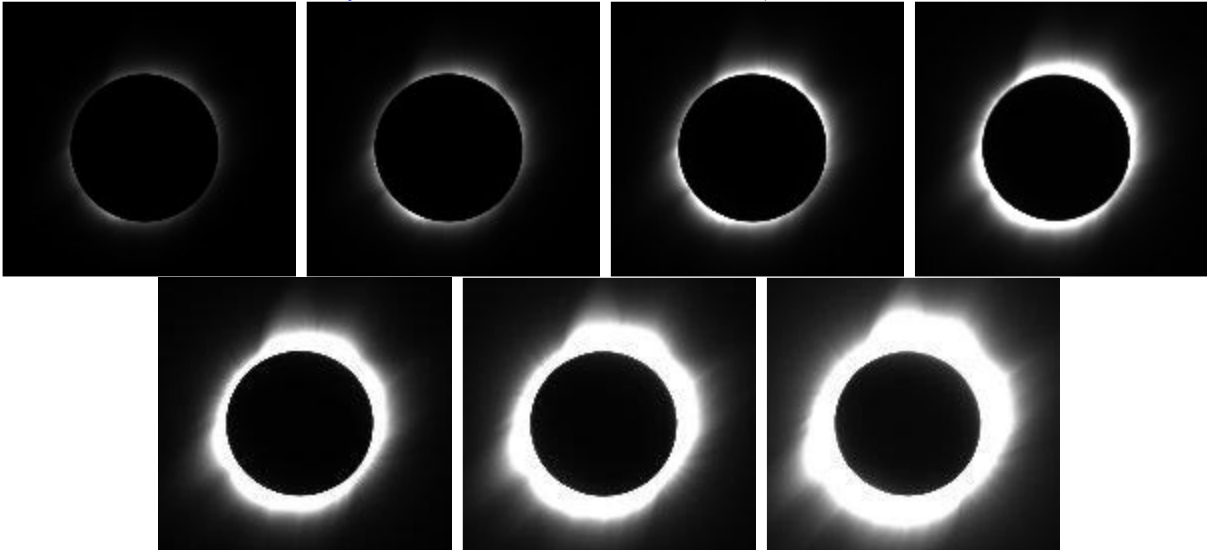


The image on the left is a composite of 5 exposures of 2 minutes of the neighbourhood of galaxy NGC 677 (190 mm telescope). The individual images were acquired without shutter in binning 1x1 mode. In this photonegative representation the smearing effect manifests itself as dark trails coming from the brightest stars. The white column is a defect of the CCD. On the right, the individual images are treated with the *deconvflat* command and composited afterwards. For erasing the vertical defect of the CCD we make use of the information in the adjacent columns. The smearing effect has disappeared almost completely and the resolution of the image stays intact.

123 *deconvflat2*

[in] [out] [coeff] [number]

Same for the sequence of images

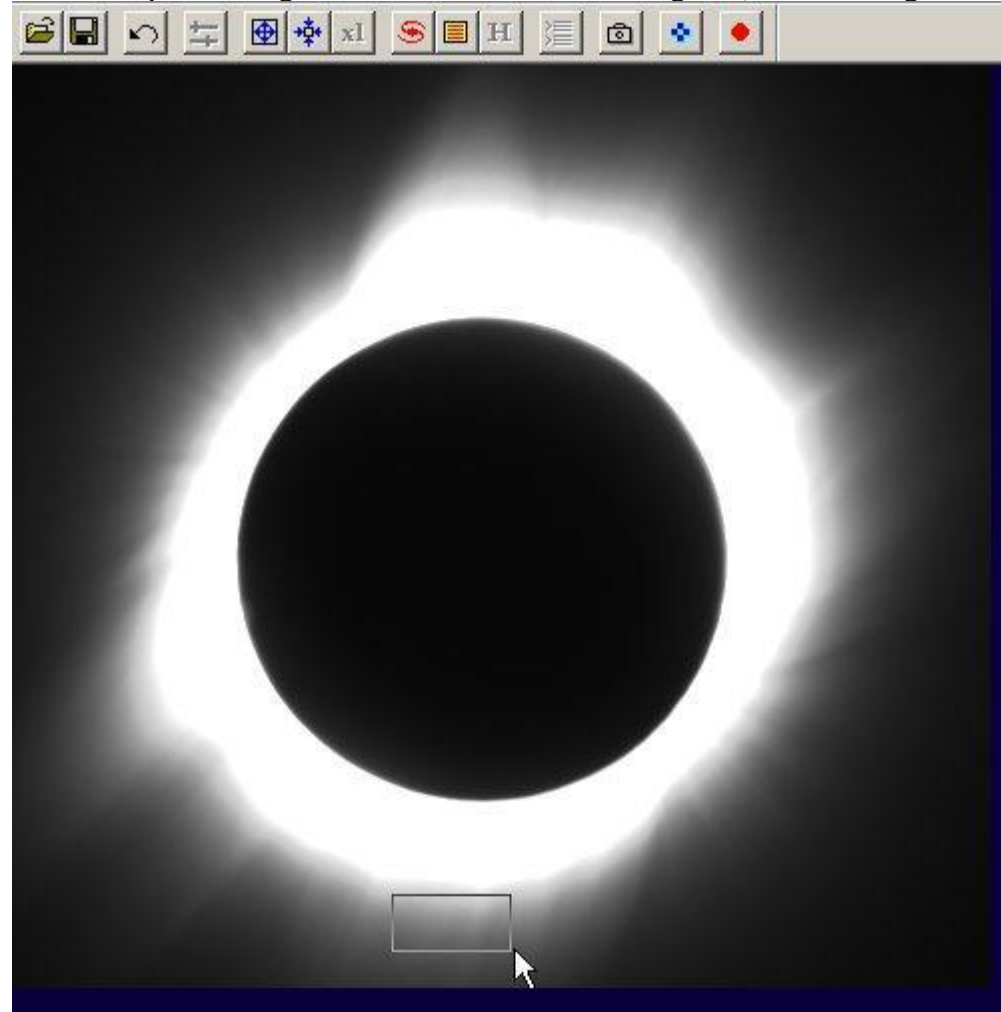
124	<a href="#">dedistor</a>	[polynom name]	<p>The argument of <a href="#">dedistor</a> command is the generic name of the polynomial files (we use here the default files POLX and POLY described <a href="#">here</a>). Note: you can define your proper transformation equations and to apply them to any image with command <a href="#">dedistor</a>.</p> <p>POLX.POL &amp; POLY.POL files: contain the coefficients of the polynomials (degree 2) and many more information that were used to calibrate the two fields both astrometrically and photometrically. The 14 first lines of a .POL file contain the coefficients of the polynomials. The 15th line contains the correlation coefficient. The lines 16 to 19 contain the coordinates of the image window used for the computation. The 20th corresponds to the degree of the polynomial. The 21st line is the magnitude constant. At last, lines 22 &amp; 23 contain the equatorial coordinates (in decimal degrees) of the center of the image.</p>
125	<a href="#">desc_hdr</a>	[image names] [description file] [number]	<p>The <a href="#">desc_hdr</a> command helps you to define the description file for the <a href="#">merge_hdr</a> command if the relative exposure time of the images is unknown or inaccurate. <a href="#">desc_hdr</a> evaluates the mean relative intensity of a user defined area of each image. The ratio of the computed intensities is the relative exposure time found in the produced description file.</p> <p>[IMAGE NAMES] is the generic name of the sequence. [DESCRIPTION FILE] is the name of the description file. [NUMBER] is the number of images in the sequence.</p> <p>Before running the command, define a small rectangle with the mouse pointer in an unsaturated part of all the set of images (use the most exposed image for identify the unsaturated region).</p> <p>Example. Here a sequence of seven progressive exposures of the 29 March 2006 eclipse (<a href="#">images Valérie Desnoux - Refractor Halley 70/400 mm and Nikon D70 DSLR</a>):</p> 

First, the Nikon RAW images are converted to PIC files. Second, the offset of the images is removed. It is important to define the true zero of the intensities scale. Use for example a command like:

```
sub2 ECLIPSE OFFSET ECLIPSE 0 7
```

where ECLIPSE is the generic name of the images to process and OFFSET is an offset master map of the camera.

Then, load the most exposed image and select an unsaturated region (but with a significant signal):

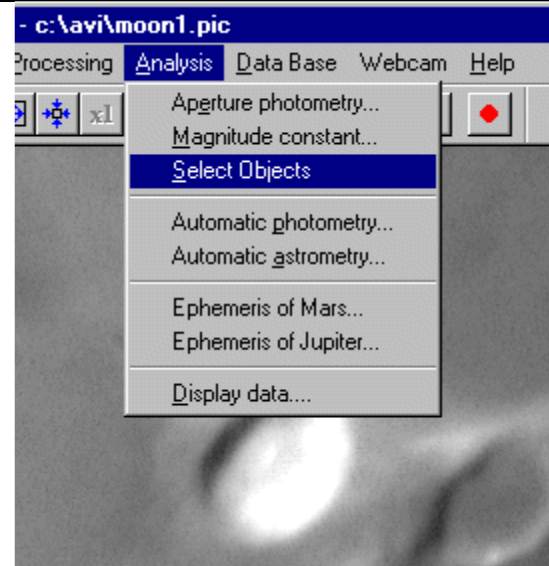


Run the command:

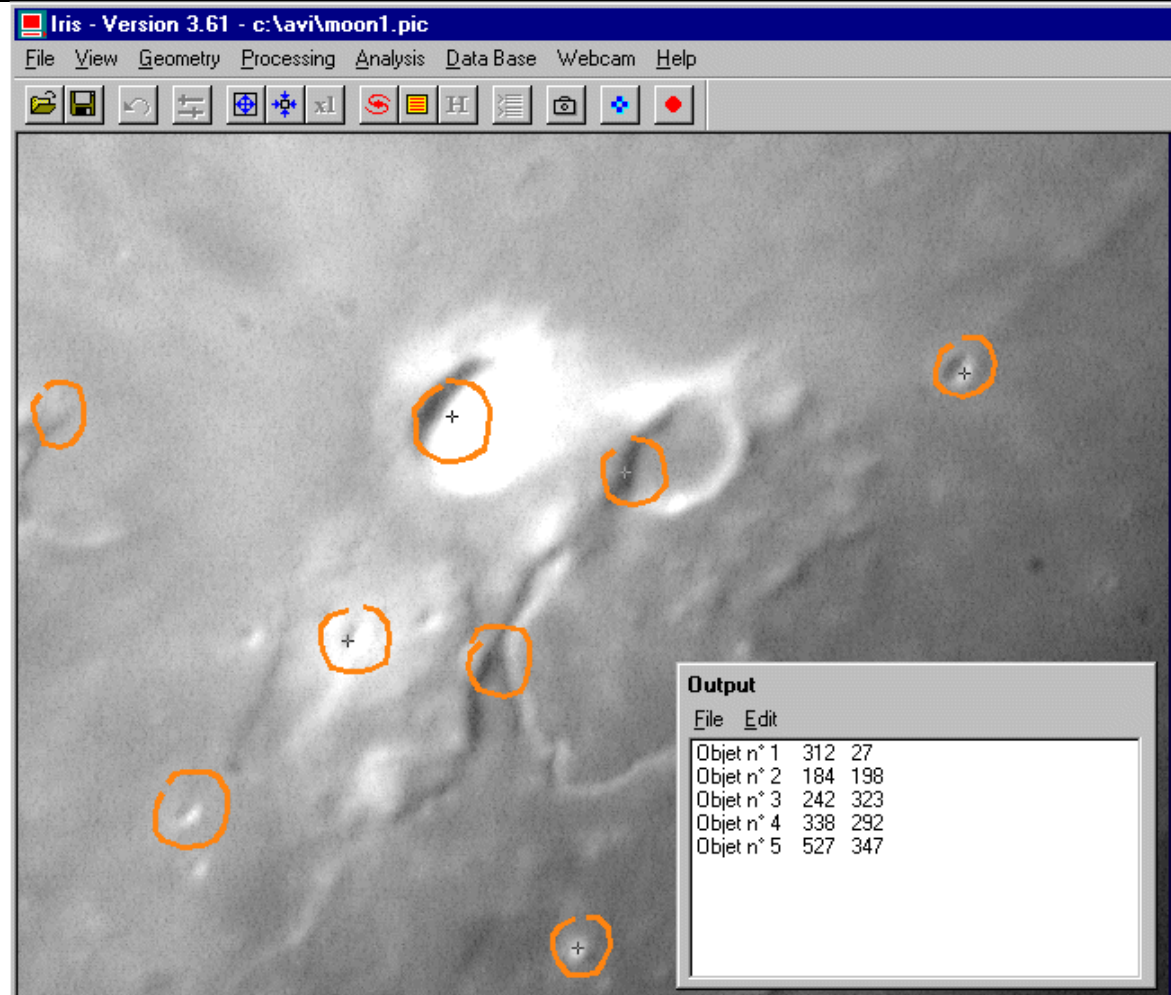
```
desc_hdr HDR ECLIPSE DESC 7
```

			<p>Iris generate the file DESC.LST in the working directory with the content:</p> <pre>eclipse1 0.016032 eclipse2 0.032064 eclipse3 0.066132 eclipse4 0.128257 eclipse5 0.254509 eclipse6 0.517034 eclipse7 1.000000</pre> <p>Rest of the merge process is described at <a href="#">merge_hdr</a> function.</p>
126	<a href="#">dilate</a>		Increases sizes of the brighter areas. Correctly works only with greyscale images. See also <a href="#">erode</a> .
127	<a href="#">disk1</a>	[x] [y] [r]	Draws a dark disk on the in-memory image. This function simulates for example a coronagraphic effect on Sun images. The parameters are the coordinates of the disk center and the radius.
128	<a href="#">disk2</a>	[x] [y] [r]	<p>This function is the opposite of the <a href="#">disk1</a> function: The outer part of the defined disk is masked. The simultaneous use of <a href="#">disk1</a> and <a href="#">disk2</a> is a solution for enhancing the aspects of prominences on the Ha image relatively to the disk.</p> <p>Example: Apply the <a href="#">disk1</a> command, multiply the result by 4 (try some value) and save the result:(give the name OUTER for example):</p> <pre><a href="#">disk1</a> 381.5 306.6 283.6 <a href="#">mult</a> 4 <a href="#">save</a> OUTER</pre> <p>Now, use the <a href="#">disk2</a> command to isolate only the sun disk image:</p> <pre><a href="#">disk2</a> 381.5 306.6 283.6</pre> <p>Finally, add the inner and the outer images:</p> <pre><a href="#">add</a> OUTER</pre>
129	<a href="#">disk3/ disk01/ disk02</a>	[x] [y] [r] [value]	Similar to a <a href="#">disk1</a> function but user can set the disk brightness.
130	<a href="#">dist</a>		Computes the distance between two stars in pixels.
131	<a href="#">distang</a>	[ad1] [dec1] [ad2] [dec2]	Returns angular distance between two points on the celestial sphere. Right ascension must be in degrees. For the conversion details see <a href="#">dms2dd</a> .
132	<a href="#">distor</a>	[name1] [name2] [degree]	Morphing is not just a special effect for artistic applications. Morphing techniques have various ranging from lens distortion correction, motion capture data interpolation, waves atmospheric turbulence correction, etc. This section describes the turbulence correction of planetary image (but the procedure is the same for correct optically distorted wide-field CCD images for examples).

		<p>The goal of the processing is to resample a target image relative to a reference image for minimize geometrical difference between the two. So, the blurring effect of the turbulence in the Earth's atmosphere is now partially compensated. If you stack the two images (or more), the spatial resolution is increased.</p> <p>The atmospheric distortion was calculated using <i>distor</i>.</p> <p>The parameters are:</p> <p>[NAME1] is the name of a good contrasted reference image on the disk.</p> <p>[NAME2] is the name to resample relative to the reference image.</p> <p>[ORDER] is the order of a 2-D quadratic equation used for fit the distortion. Range is between 1 (linear correction) and 5 (complex distortion). The recommended value is 3 or 4 for most the case.</p> <p><i>distor</i> use an iterative scheme for aberrant points elimination.</p> <p>Before running the <i>distor</i> command it is necessary to define point interactively with mouse in the reference image. If possible, this point mark contrasted details (light/shadow region on the moon surface, planetary limb, stars, ...). The geometric correction is only valid into the pointing area (outside this area Iris extrapolate).</p> <p>First, one should necessary to register the image at a pixel precision (it is a mean registration — remember the turbulence is the effect of the atmosphere). You can use for this command <i>pregister</i>, <i>cregister</i> (planetary applications) or <i>register</i>, <i>register2</i>, <i>coregister</i>, ... (deep-sky applications).</p> <p>For the Moon processing examples (sequence of 72 webcam frames MOON1... MOON72, Celestron 8 telescope) we select a contrasted area with mouse, then:</p> <p><i>pregister</i> MOON I 256 72</p> <p>Then select reference points into the reference image (MOON1 images for the example).</p>
--	--	--



This point serves to create the transform model between reference and processed image. The procedure fit polynomial equations using least squares criteria to model distortion. For points selection activate the option **Select Objects** of the **Analysis** menu. For an optimal result it is necessary to select contrasted feature in the reference image: Up to 100 points can be defined (more point = time consuming for the calculation). Try to distribute uniformly these points on the surface image. If you like you can now deselect the **Select Objects** option of the **Analysis** menu.



Compute the polynomial equations and produce the corrected image. The polynomial transform is a 1st to 5th order polynomial ([ORDER] parameter of the *distor* command). Example:

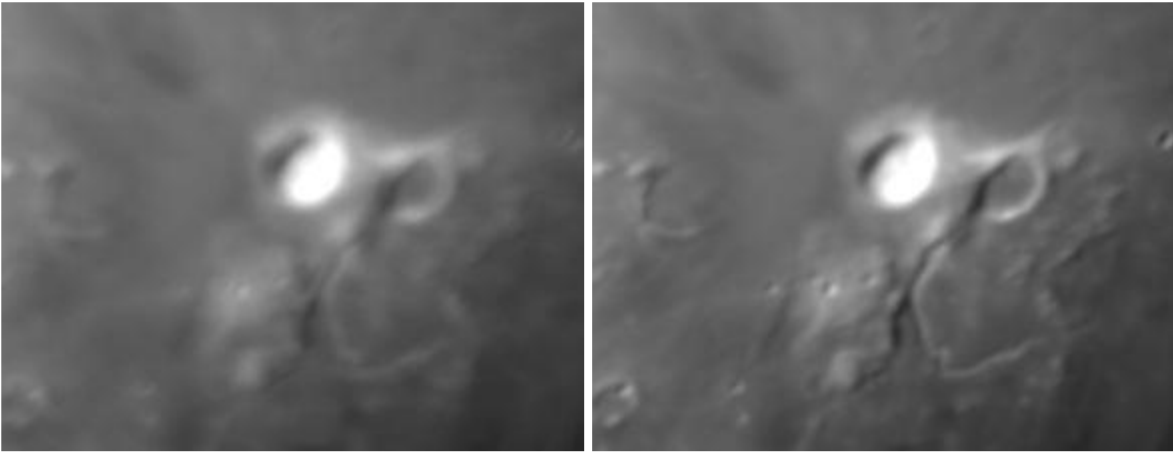
*distor* I1 I9 4

The image I9 is now registered relative to the I1 image. The result can now be saved on the disk:

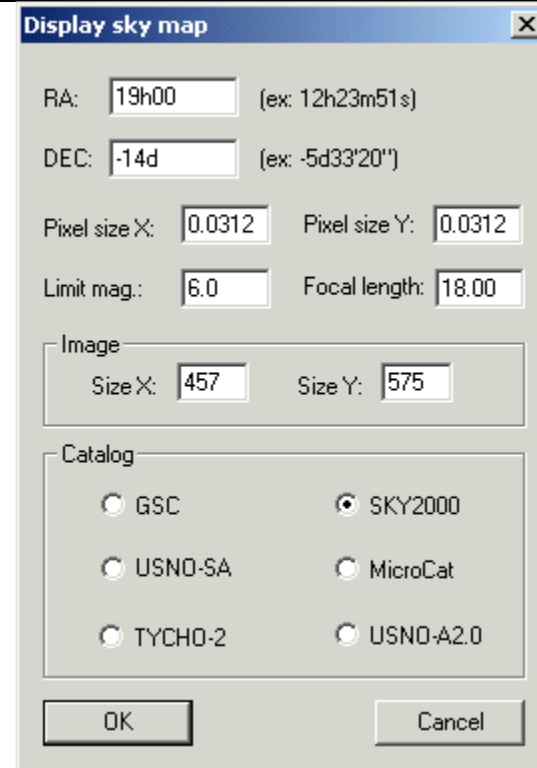
*save* R9

For check try some find like:

*load* I1

			<p><i>sub</i> R9 5000  <i>visu</i> 5100 4900</p> <p>Normally the difference is very near the 5000 level.</p>
133	<i>distor2</i>	[in] [out] [degree] [number]	<p>It is possible to process all the sequence images in one step by the use of the <i>distor2</i> command:</p> <p><i>distor2</i> [IN] [OUT] [DEGREE] [NUMBER]</p> <p>You can add the registered images:</p> <p><i>Add_norm</i> R 72</p> <p>or select the best ten resolved images before:</p> <p><i>bestof</i> R 72  <i>select</i> R T  <i>add_norm</i> T 10</p> <div style="display: flex; justify-content: space-around; align-items: center;">  </div> <p>Above: Left, stack of very bad aligned images. Right, stack of the same images after <i>distor</i>.</p>
134	<i>distortion or distorsion</i>	[x-center] [y-center] [A] [B] [C] [D] [scale]	Applies coefficients obtained from the command <i>fitdistor</i> .
135	<i>div</i>	[name] [coef]	Divides image in memory by the specified file. The resulting image is multiplied by the value [COEFFICIENT].
136	<i>div2</i>	[in] [operand] [out] [coef] [number]	Divides a sequence of images having the generic name [IN] by the image [OPERAND] and multiplies the result by [COEF] (see also <i>div</i> ). The [OUT] parameter is the generic name of the output images. The number of images to process is [NUMBER].

137	<i>dms2dd</i>	[angle] (12h23m34.3s or 30°34'12" or 30d34m12s)	<p>Converts angular distance represented in minute-second form into in decimal analog. Be careful when converting right ascension: hours are not converted in this function in the right way and stays the same. For example:</p> <p><i>dms2dd</i> 12h59m0s</p> <p>Returns Angle = 12.983333</p> <p>Which is correct only if you wanted to convert 30°34'12 or 30d34m12s into its decimal form. In order to avoid that and be able to use it to right ascension conversion you can multiply all values by factor of conversion 15 and enter the following expression:</p> <p><i>dms2dd</i> 180h885m0s</p> <p>Which correctly returns Angle = 194.750000</p> <p>Useful when one wants to determine the angular distance between two objects with <i>distang</i> function.</p>
138	<i>draw_airy</i>	[size x] [size y] [I0] [diameter (mm)] [obstruction]	<p>Allows you to create an image containing the representation in the form of an airy spot of an artificial star. The [SIZE] parameter contains the dimension of the new image in pixels (the format is square). The parameter [I0] contains the peak intensity of the Airy spot in ADU. The [DIAMETER] parameter contains the diameter of the telescope in millimeters. The [OBSTRUCTION] parameter contains the obstruction ratio of the main mirror (enter 0 for a telescope without secondary). The [LAMBDA] parameter contains the wavelength in microns. The [SCALE] parameter contains the scale of the size of the image spot in arc seconds per pixel. Note that the command produces the AIRY.DAT text file in the working directory which contains the profile of the Airy task in tabulated form.</p>
139	<i>draw_alpha</i>	[alpha0] [delta0] [focal length] [pixel size] [alpha] [intensity]	<p>[<math>\alpha_0</math>, <math>\delta_0</math>] are the approximate equatorial coordinates of the center of the field (the same used under display map dialog box). The focal length of the telescope and the size of the pixels of the sensor are provided in millimeters. The parameter [intensity] is the intensity of the circle of coordinated traced in the image. Finally (alpha, delta) are the coordinates of the drawn circle. But firstly, image must be properly oriented relative to an equatorial grid. It can be done by two different ways by first filling image in interest black and then through the <b>Display &gt; Display sky map</b> menu produce a map of required region:</p> <p><i>fill</i> 0</p>



IRIS automatically adds to the current image stars extracted the catalogues in form of red points. Then one can save this image and use manual alignment functions such as [qr3](#). Or try to use automated method of [coregister](#) function.

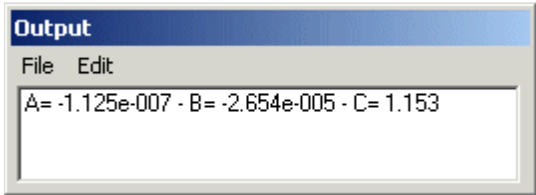
It is not well a great science, but this can help to illustrate some guiding principles in astronomy: the way in which an image is formed in the plan of the sensor, astrometry, etc.

The transformation employed is known as gnomonic: the celestial sphere is projected on a tangent point with the sphere. This point is line of sight at the coordinate Alpha = 19h00 and Delta = -14°.

When the orientation is set it is easy to display an equatorial grid in the image:

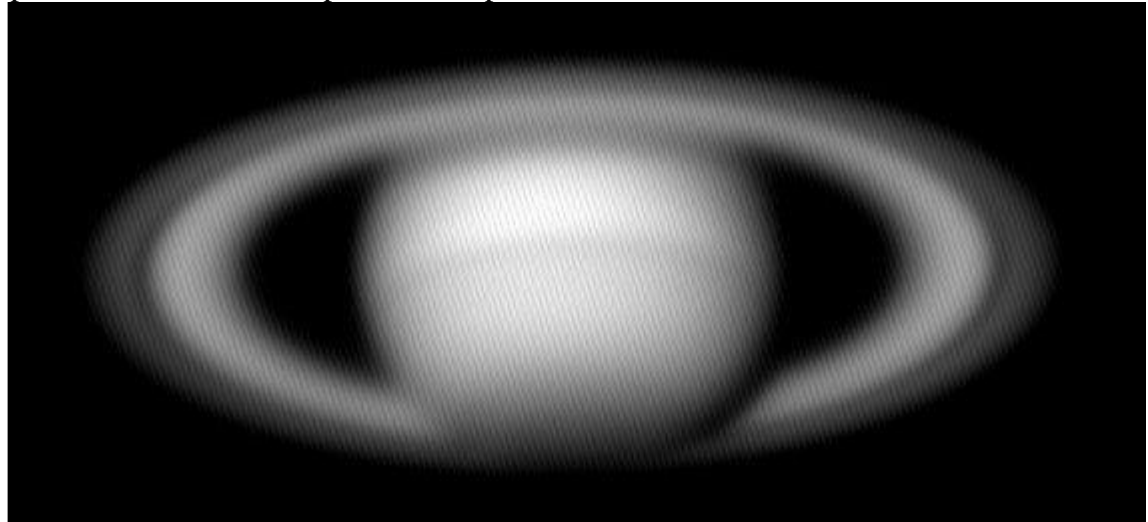
```
draw_alpha 19h -14d 18 0.0312 17h30m 10000
draw_alpha 19h -14d 18 0.0312 18h00m 10000
draw_alpha 19h -14d 18 0.0312 18h30m 10000
...
draw_delta 19h -14d 18 0.0312 10d 10000
draw_delta 19h -14d 18 0.0312 0d 10000
draw_delta 19h -14d 18 0.0312 -10d 10000
```

			Stellar catalog and path should be already set in menu <b>File &gt; Settings</b> .
140	<i>draw_delta</i>	[alpha0] [delta0] [focal length] [pixel size] [alpha] [intensity]	Works in correspondence <i>draw_alpha</i> with and have the same functionality.
141	<i>drizzle</i>	[name] [resolution] [number]	<p>This command performs an optimal adding of images as far as resolution is concerned. The principle is that, at sub-pixel level, shifts between individual input images are nearly randomly distributed. For example, a star in the first image may be centered perfectly in the middle of a pixel, whereas it will be across two pixels in the second one, and so on. Since it is easy to know the exact shift between the images, it is possible to create an output image with a finer sampling, in which resolution may be increased with respected to each input image. In fact, energy from each input pixel is dropped in the output image, and the whole process may be compared to a drizzle...</p> <p>The <i>drizzle</i> command is adapted to under-sampled images, for example when the telescope focal length is too short for the pixel size. One may consider that the system is under-sampled when FWHM is smaller than 2 pixels.</p> <p>Before using <i>drizzle</i>, it is necessary to know the shift between the images. We suppose that only a linear translation exists between images (with no distortion and no rotation). The shift values are in the file SHIFT.LST (to create this file, refer to the command <i>register</i>). You can also create this file manually, by measuring the shifts along X &amp; Y individually (the sign convention is image#1 - image#i).</p> <p>It is important that all the input images are acquired in the same conditions: same exposure time, same sky background level. If this is not the case, you have to adjust offset and gain prior to use <i>drizzle</i> (see commands <i>offset</i> and <i>mult</i>).</p> <p>The parameters of the command are:  [NAME] is the generic name of the input image.  [RESOLUTION] is the over-sampling factor with respect to the input images. A factor or 2 to 2.5 may be considered as a coherent objective when the number of images is between 5 and 10. If the number of images is much larger, this factor may be increased to values as high as 3 or more.  [NUMBER] is the number of images in the sequence. A minimum of 5 images is generally necessary.</p>
142	<i>dtrans</i>	[x] [y]	<p>Select a star in an image by drawing a small rectangle around it with the mouse. Then, the command <i>dtrans</i> translates the images with the values:</p> $DX = [x] - X_M$ $DY = [y] - Y_M$ <p>where <math>(X_M, Y_M)</math> is the centroid of the object in the rectangle.</p> <p>The <i>dtrans</i> command is very useful the register a sequence of images with respect to a reference star at coordinates ([x], [y]) in the reference image. Once again, [x] and [y] represent the coordinates of the star on the original first image and the amount of translation is estimated and applied automatically.</p>

143	<i>echo</i>	[message]	Print a message into the output windows during execution of a batch file ( <i>run</i> command). Space character are not authorized (use "_" character).
144	<i>edge</i>	[x1] [y1] [x2] [y2]	The coordinates [X <sub>1</sub> ], [Y <sub>1</sub> ] and [X <sub>2</sub> ], [Y <sub>2</sub> ] mark a frame outside of which the pixels of the image in memory are set to zero. Sometimes the edges of images contain no significant information or have defects (artefacts due to previous processing). In these cases, the EDGE command can be used to fix the image.
145	<i>edge2</i>	[lx] [ly]	Similar to <i>edge</i> but requires only two parameters which represent border width in pixels relative to original image size.
146	<i>ENTROPY</i>		Return the value of the entropy in a zone of the image selected with the mouse.
147	<i>equal</i>	[x1] [y1] [x2] [y2]	Fits a polynomial surface in the area defined by the point [X <sub>1</sub> ], [Y <sub>1</sub> ] and [X <sub>2</sub> ], [Y <sub>2</sub> ] , subtract this model (only the area is modified) and add the median level. Parameters of fitting is affected by the function <i>setsubsky</i> .
148	<i>EQUALIZE_CFA/ GREY_FLAT</i>		The function equalizes the median intensity of RGB layers in a CFA images.
149	<i>erase</i>		Cleans the information located inside a selection box. An interpolation of the central zone is carried out by using pixels located on the periphery. The texture of the erased area is preserved for a neutral rendering. This command can be used for example to erase residual dust umbra in the image.
150	<i>erf</i>	[x]	Returns value of the error function that is defined as: $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ But somewhat argument of this function in the Iris argument is divided by 1.201122 so in order to obtain true value of the error function user must divide the required value by this number like so: $\text{erf}\left(\frac{x}{1.201122}\right)$ . Why is that I have no clue.
151	<i>erode</i>		Increases sizes of the darker areas. Correctly works only with greyscale images.
152	<i>evign</i>		Load the "flat" image and run the <i>evign</i> command (no parameters). Iris return the polynomial coefficients in the output window:  Relative to the X distance from the image center, the vignetting is modeled by the function: $y = -1.125 \cdot 10^{-7}x^2 - 2.654 \cdot 10^{-5}x + 1.153$

153	<i>export</i>	[name] [header] [byte per pixel] [reverse (0 or 1)]	Exports images in a non-standard format. The program writes the header at the beginning of the file, whose length in bytes should be specified in [header]. This header is filled with zeroes. In the parameter [BYTE PER PIXEL], you must indicate whether the pixels are coded on one or two bytes. If the coding is on two bytes, you must also indicate in the parameter [reverse] the order of the bytes in the 16 bit word. If [reverse]=0, they will be in the INTEL format (most significant/least significant), while if [reverse]=1, they will be in the MOTOROLA format (least significant/most significant). See also: <i>import</i> . Example:  <i>export</i> FILE.IMG 256 2 0  Converts the image in memory as a free format image with the name FILE.IMG, and with a header length of 256 bytes and pixels coded on 16 bits in the INTEL format.
154	<i>export_asc</i>	[name]	Saves the images with an ASCII format in 3 rows. The two first rows contain the pixel coordinates (origin at (1,1)), whereas the third row contains the image intensity. Warning: the result file may be very big for large input images. See also <i>import_asc</i> .
155	<i>export2</i>	[name] [header] [byte number by pixel] [reverse (0 or 1)]	Same as <i>export</i> , but the natural internal coding of Iris [-32768..32767] is mapped into [0..65535].
156	<i>faint</i>	[coef]	Multiplies selected area by the [COEF]. Correctly works only for the greyscale images since on RGB it enhances only Red channel.
157	<i>fb1/fb2/fb3</i>		Functions performing low-pass type filtering on the image in memory. The effect is increasing going from <i>fb1</i> to <i>fb3</i> . Decreases high spatial frequencies. Opposite to <i>fh1/fh2/fh3</i> .
158	<i>fconv</i>	[image #1] [image #2]	Convolute [IMAGE #1] with [IMAGE #2] but for some reason it splits the original image on quarters and moves them... Image can be restored with the <i>permut</i> function but RGB channels must be processed separately.
159	<i>fcorrel</i>	[image #1] [image #2] [coef]	Compute the cross-correlation of [IMAGE #1] and [IMAGE #1]. The [COEF] coefficient is an intensity scale factor for the result. Example:  <i>fcorrel</i> MARS1 MARS2 1
160	<i>ffill/fill</i>	[value]	Uniform fill of a region defined by the mouse. Used to process an image in the Fourier (FFT) domain. Correctly works only on BW images. <i>ffill</i> is very similar to <i>fill2</i> command, but it fill with a given value two areas symmetric with regard to the x-axis and y-axis. Removes the principal parasitic peak. For more information see example of use <i>fftd</i> function.
161	<i>fftd</i>	[modulus] [phase]	IRIS can compute a FFT (Fast Fourier Transform) to evaluate fixed pattern noise of an image. First, the Direct (Forward) FFT is computed with FFTD. Next, the fixed pattern noise is corrected through boxcar filtering ( <i>ffill</i> ). Finally, the reverse FFT ( <i>ffti</i> ) is computed to display the corrected image. The

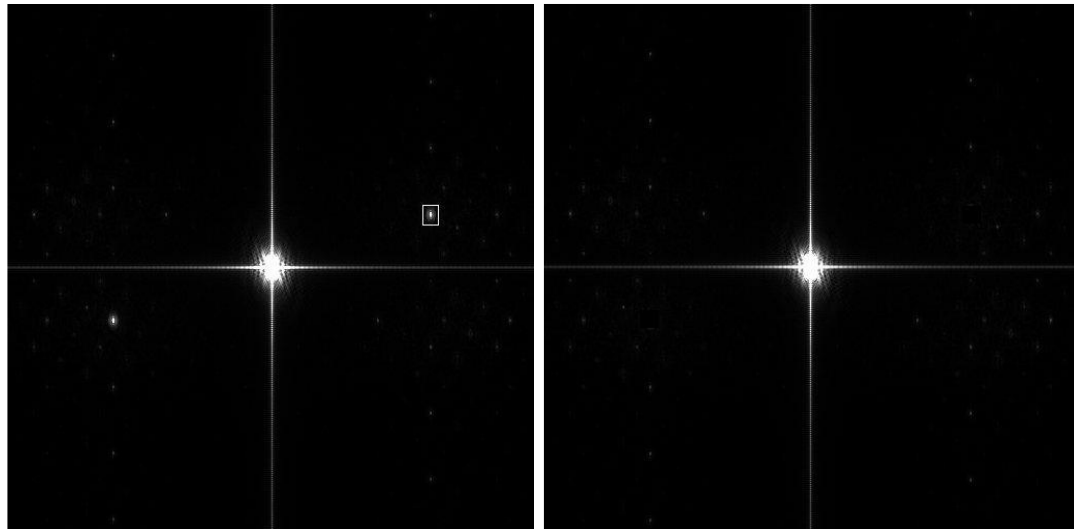
modulus (magnitude) of the FFT is stored in the output file [MODULUS] and the phase is stored in the output file [PHASE]. Example of use is presented below:



Periodic parasitic electrical noise is clearly visible  
First of all the direct FFT:

```
fft X Y
```

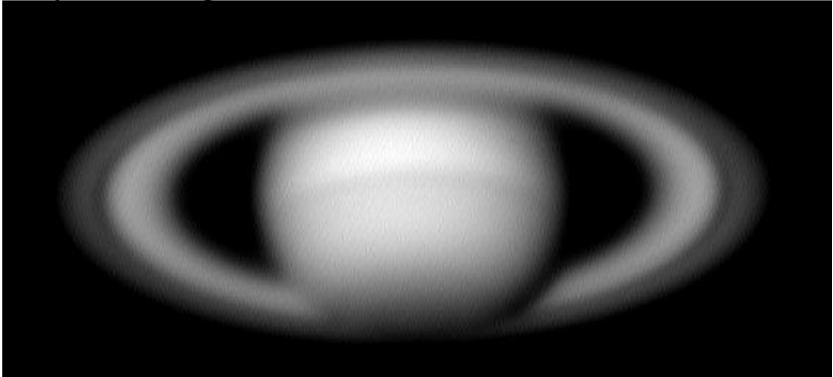
Now drag a small rectangle around one of the peaks and mask both noise peak by the use of *ffill* command.

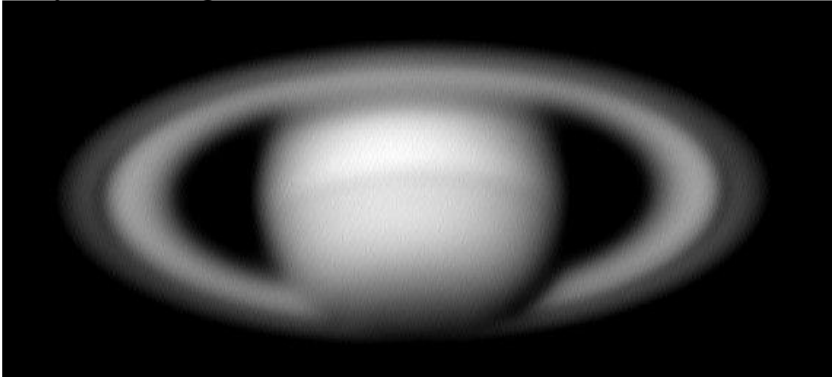


Save the modified module component of the FFT (important).

*save* X  
 And finally revert FFT:  
*fft* X Y

The result is now a very clean image:



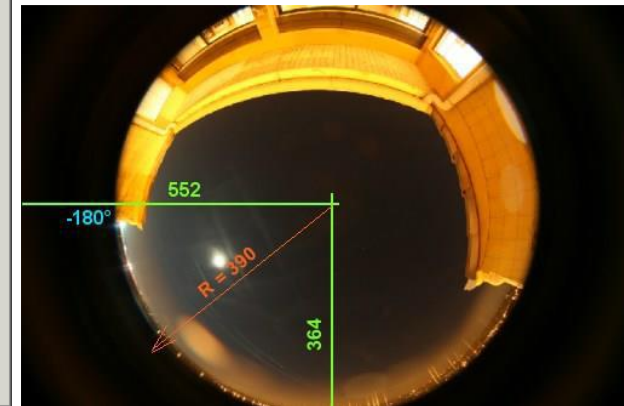
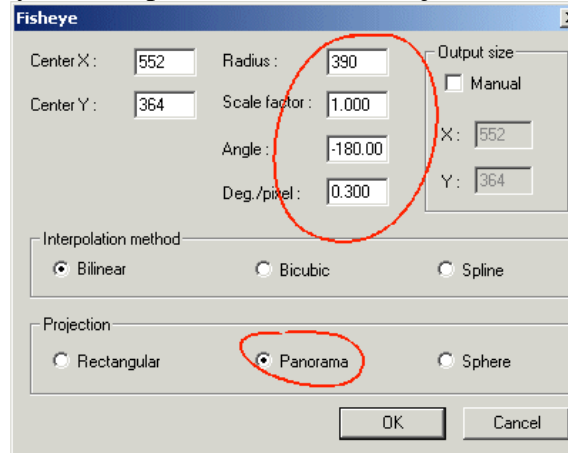
			<p><i>save</i> X          And finally revert FFT:  <i>fft</i> X Y</p> <p>The result is now a very clean image:</p> 
162	<i>fftd2</i>	[in] [modulus] [phase] [number]	Calculate the direct Fourier transform of a sequence of images.
163	<i>fft</i>	[modulus] [phase]	Compute an Inverse Fast Fourier Transform from the magnitude and phase data contained in files [MODULUS] and [PHASE].
164	<i>fft2</i>	[modulus] [phase] [out] [number]	Calculate the inverse Fourier transform of a sequence of images.
165	<i>fh1/fh2/fh3</i>	[modulus] [phase] [out] [number]	Functions performing low-pass type filtering on the image in memory. The effect is increasing going from <i>fh1</i> to <i>fh3</i> . Increases high spatial frequencies. Opposite to <i>fb1/fb2/fb3</i> .
166	<i>file_affine</i>	[in] [out] [number]	<p>Iris can register wide-field stellar images. For that, it is recommended to use the powerful <b>Three matching zones</b> function of <b>Stellar registration</b> dialog box (Processing menu) - (see details <a href="#">here</a>), or run command <i>coregister4</i> from the console.</p> <p>After the execution of the function, Iris created automatically in the working directory the file <i>affine.lst</i>. This text file contains the parameters of the transformation applied to each image of the sequence (a line by image). Each line contains the 6 values:</p> <p>A B C D E F</p> <p>such as</p> $x' = AX + BY + C$ $y' = DX + EY + F$ <p>with <math>(x, y)</math> the coordinates in the transformed image and <math>(x', y')</math> the coordinates in the starting image.</p>

			The interest of this file is the possibility to apply in a second time the same transformations to a sequence of images without having to remake the complete calculation of registration (see also command <i>file_trans</i> function, the partnership of <i>trans</i> command).
167	<i>file_conv</i>	[kernel name]	Convolution of the in memory image by a matrix whose coefficients are contained in image [KERNEL_NAME]. The value of the coefficients is multiplied by Iris by 0.001 before calculation itself. The image must be obligatorily square and of odd size. The maximum size is of 41x41 pixels. You can use command IMPORT_ASC to charge a matrix with convolution starting from a textual file, which you then save in the format PIC or FITS. Correctly works only with greyscale images.
168	<i>file_coregister</i>	[in] [out] [number]	After the execution of a <i>coregister2</i> command (registration of a sequence of image by simultaneously using geometrical transformations like translation, rotation and scaling), it is now possible to start again a registration with <i>file_coregister</i> which use parameters of transformation already calculated with <i>coregister2</i> . For example: <i>coregister2</i> A B 5 register the sequence A1..., A5 by producing the B1 sequence..., B5. So, then you type <i>file_coregister</i> A C the C1 sequence..., C5 is strictly identical to B1..., B5, but the result is obtained considerably more quickly because the transformations are not recomputed. <i>file_coregister</i> can also be used to register very large images after having calculated the parameters of registration in a cropped part of those. See also commands <i>file_trans</i> and <i>file_rot</i> , equivalent for the translation and a simple rotation.
169	<i>file_rot</i>	[in] [out] [number]	Allows either to re-run the rotational transformation which was performed with <i>rregister</i> function or user can manually input the required angles in the ROT.LST file. It's typical content: 72.472 79.068 0.000 72.472 79.068 1.498 72.472 79.068 2.189 Where first two columns represent the coordinates of the rotation center and the third one is the angle.
170	<i>file_trans</i>	[in] [out] [number]	Similar to <i>file_rot</i> but works with translations. All values are kept in SHIFT.LST file. Typical content might be: 0.000 0.000 5.160 -11.862

			17.969 -7.818 Commands <i>file_trans</i> and <i>file_rot</i> are particularly important when one wishes to register the components of a RGB image in the same way.
171	<i>fill</i>	[value]	Fills the whole current image with pixels having the [VALUE] intensity.
172	<i>fill_border</i>	[border size] [level]	Function similar both to border and edge2. Doesn't change dimensions of the file but creates frame around edges of the image with the given size in pixels. Intensity of that border is set in [LEVEL].
173	<i>fill_ellip</i> or <i>FELLIP</i>	[value]	Fills an ellipse defined by the mouse selection with given intensity.
174	<i>fill_inv</i>	[value]	Fills rest of unselected image with pixels given intensity
175	<i>fill2</i>	[value]	Fills selected area with given intensity
176	<i>fill3</i>	[x1] [y1] [x2] [y2] [value]	Same as <i>fill2</i> but coordinates can be set in command line.
177	<i>find_hot</i>	[list file] [threshold]	<p>The command makes it possible to generate a file lists (format text) in the working directory which contains the co-ordinates of the pixels which have an intensity higher than only one provided by the user.</p> <p>This file, known as cosmetic file, is then used by IRIS to correct certain systematic defects during the preprocessing of the deep-sky images. Thus, if command <i>find_hot</i> applies to an image of the dark signal, the produced file will contain coordinates of the pixels of affected by an abnormally high dark current (hot pixels). When this file is read by a function of preprocessing, the pixels in questions in then treated images are replaced by a computed value starting from the intensity of the close pixels.</p> <p>The two parameters are:</p> <p>[LIST FILE] the name of the file lists produced. The name is supplemented by extension .LST. [THRESHOLD] the threshold for the discrimination of the hot points.</p> <p>To note that IRIS turns over in the console the number of hot points found. For a normal CCD it is necessary to be arranged to adjust the threshold so as not to find much more than one ten hot points.</p> <p>For example:</p> <pre><i>load</i> DARK <i>find_hot</i> COSME 300</pre> <p>Here a typical content of the produced file (file COSME.LST in the example):</p> <pre>P 1086 1 P 402 7 P 1434 13 P 403 23 P 1372 27 C 468 0</pre>

			<p>The letter P indicates that it is necessary to correct only one pixel of the image. The coordinates of the first sick pixel are (1086, 1). Second is at the coordinates (402, 7) and so on.</p> <p>The last line starts with C and was added manually to the file (for example with an editor like WordPad). It indicates to IRIS that it will be necessary during the pretreatment to replace the column of row 468 (on the basis of the left) by the average value of the adjacent columns.</p> <p>In the present case the action of the last line of file COSME.LST is equivalent making:</p> <p>REPAIRX 468</p> <p>If the cosmetic file contains the line:</p> <p>L 34 0</p> <p>That means that it is necessary to replace the line of co-ordinates 34 (starting from the bottom of the image) by the value of the adjacent lines. To note that in the case of the correction of the lines and the columns the third value in a line of the cosmetic file must have a null value.</p> <p>To apply the cosmetic corrections registered in the file lists you can use commands <i>cosme</i> and <i>cosme2</i>. Some dialog boxes can now also take into account information of a cosmetic file.</p>
178	<i>findstar</i>		<p>Detects stars having a level greater than [sigma] times the level of the sky background noise (see the <i>setfindstar</i> command) and visualizes them. A file of the type STAR.LST is created on the drive. This file may be edited (e.g. EDIT command of MS-DOS). It contains parameters about detected stars (position, instrumental magnitude, FWHM). More precisely for each colon:</p> <p>star number,  X coordinate of the star in the image,  Y coordinate of the star in the image,  instrumental magnitude of the star,  astrometric right ascension (in decimal degrees),  declination (in decimal degrees),  true reduced magnitude,  isolation criteria for the star (1 is the most isolated),  the FWHM along X &amp; Y.</p> <p>Note that STAR.LST file is used with automatic astrometry and photometry command.</p>
179	<i>fisheye</i>	[x-center] [y-center] [focal length] [scale]	<p>Converts images obtained with the fisheye lens into rectangular projection.</p> <p>[x-center] X coordinate of the image center  [y-center] Y coordinate of the image center  [focal length] visible image radius in pixels  [scale] scale factor. Can be set less than one in order to show image corners.</p>

More options are available in **Geometry > Fisheye**. For example, it is possible to generate cylindrical panorama with the adjustable starting angle:



The resulting image will be:



For obtaining the same result from console commands, enter:

```
load FISHEYE_VIEW
rec2pol2 552 364 390 -80 0.3
mirrorxy
mirrorx
```

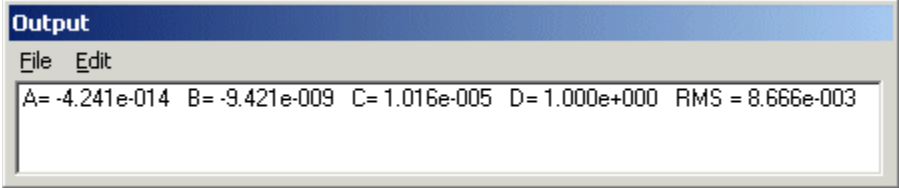
180 *fit*

[data file] [degree]

The command *fit* is a general command for fitting a polynomial function on the data saved in a text file (two columns format). For example:

```
Fit GRID 6
```

An O-C file is created automatically in the working directory (OC.LST)

181	<i>fitdistor</i>	[data file] [degree]	<p>Searches for optimal fitting function for the distortion correction. The first parameter is the name of the text file containing the coordinates measured in the image. The second parameter is the degree of the polynomial which is adjusted through the measured points. Here the result in our example:</p>  <p>RMS is the quadratic error of the fit of the second column of the file. More the error value is small, better is the adjustment. RMS is the quadratic error of the fit of the second column of the file. The smaller the error value, the better the adjustment.</p>
182	<i>fits2fit</i>	[nom] [number]	Convert a FITS extension sequence to a FIT extension sequence (example the file name M57-2.FITS is converted to M57-2.FIT).
183	<i>fits2pic</i>	[in] [out] [number]	<p>Converts a sequence of FITS image into a sequence with the proprietary PIC format. Parameters:</p> <p>[IN] is the generic name of the input sequence.</p> <p>[OUT] is the generic name of the output sequence.</p> <p>[NUMBER] is the number of images in the sequence.</p>
184	<i>fits2pic2</i>	[in] [out] [number]	Same as fits2pic but for sequences.
185	<i>flat</i>	[name] [hl] [ll] [level] [#iter] [#image]	<p>The <i>flat</i> command allows you to calculate a flat-field without having to take specific images at dusk (images without stars or other objects). The flat-field can be obtained from merely the images of the observed objects taken during the night.</p> <p>Other methods can also be used to obtain flat-field images from night images. They use the calculation of the median of a set of images (see the <i>smedian</i> command). However, the technique used in <i>flat</i> can resolve difficult cases:</p> <ul style="list-style-type: none"> <li>• you do not have a dusk flat-field and the night images contain dense star fields or extended objects.</li> <li>• you do not have a dusk flat-field and the night images are planetary images.</li> </ul> <p>In these two cases, correctly extracting a flat-field with a median set technique is very unlikely. The <i>flat</i> command is useful in these situations (the median set technique is still a good method in normal situations because it is faster and easier to implement). The command uses an iterative process and works with the logarithmic values of the images. It is recommended to have as many images as possible to get the best result (typically between 5 and 10 images, the maximum number allowed is 15).</p> <p>Several conditions must be respected:</p> <ul style="list-style-type: none"> <li>• The dark current and offset signal must have been subtracted from each image.</li> </ul>

- The images must have the same signal level (for the control, use the sky background level at the same place on the detector, or the level of a detail common to all the images). Use the *offset* command to adjust the levels.
- the images must be of the same celestial object and must be taken through the same filter if one is used (this last condition is a general rule for flat-fields, no matter which technique is used).
- The images must be displaced by some number of pixels along both the X and Y axes, with respect to each other. These shifts can be non integer pixel values (measure the displacements with the cursor, or commands such as *register*, *pregister*...). The shifts must be measured with respect to first image in the series. The only restriction on the shifts is that they not be colinear, nor have a common multiplier:
- If  $a_i$  is the displacement vector of the image "i", and  $a_j$  is the displacement vector of the image "j", there must not be a real constant  $k$  such that  $a_i = k \times a_j$ .

The displacement values do not affect the quality of the result. However, it is recommended to not displace the images too much, because the complete calculation of the flat-field image can only be done on the parts common to all the images. The parts of the flat-field outside the common area are still valid, but they are calculated with less precision because of the smaller number of images used in the calculation. Also, note that the calculation time grows with the displacement values. Large images are prohibited due to large computation time (up to 400 pixels size image are correct).

It is recommended to choose the reference image (first image) so that the relative displacements are as isotropic as possible for the set of images.

Before executing *flat* it is necessary to produce the file SHIFT.LST containing the relative displacements of the images with respect to the first image. Each line of this file contains the displacements with respect to the reference image (first the displacement along the X axis, then along the Y axis, with one or more blanks between the numbers). The first line of this file contains the elements of the first image (the reference image), that is (0,0). This file can be created with a word processing. Remember that automatic registration commands in Iris (*register*, *full\_pr*, *pregister*...) produce a shift file SHIFT.LST in the working directory.

The command arguments of *flat* are:

[NAME]: the generic name of the images to be processed. The generic name is the root of the name of the image, which will have a number and an extension added to it. The first number added is 1, and the last is [#IMAGE]. Thus, with the generic name "IMAGE" and [#IMAGE]=5, the processing will be done on the images:

```
IMAGE1.PIC
IMAGE2.PIC
IMAGE3.PIC
```

			<p>IMAGE4.PIC IMAGE5.PIC</p> <p>[#IMAGE]: the number of images to be used in the calculation (between 5 and 15). [#ITER]: the number of iterations. This number is chosen as a result of the previous tests. Typically, between 1 and 3 iterations are used. [HL]: eliminates pixels with a level over [hl] from the calculations. This can be useful when processing a field with saturated stars (the saturated parts do not contain radiometrically useful information). In general, [HL] should have the value of the maximum dynamic range of the image. [LL]: eliminates pixels with a level less than [LL] from the calculations. In particular, the value of [LL] can be adjusted to reject pixels with a negative value, thus [LL] is usually 0. [LEVEL]: the generated flat-field will have the average intensity [LEVEL]. Most of the time this value will be the same order of magnitude as the average level of the processed images. [LEVEL] must be greater than 1.</p>
186	<i>fpolrec</i>	[modulus] [phase]	<p>Transforms the frequency domain images from polar to rectangular. The two arguments are filenames containing the [MODULUS] and [PHASE] on input, and the [REAL] and [IMAGINARY] components on output, respectively. NOTE: this command rewrites the contents of the specified files! Make a copy to preserve the originals.</p>
187	<i>frecpol</i>	[real part] [imaginary part]	<p>Transform the frequency domain images from rectangular to polar. The two arguments are filenames containing the [REAL] and [IMAGINARY] components on input, and the [MODULUS] and [PHASE] on output, respectively. NOTE: this command rewrites the contents of the specified files! Make a copy to preserve the originals.</p>
188	<i>full_pr</i>	[in] [out] [dark] [offset] [flat] [number]	<p>Computes the automatic processing of a sequence of [NB] images having the generic name [IN], i.e. subtracts the offset signal, removes the dark current signal, and divides by the flat-field image. The images are then automatically registered. At last, the images are added.</p> <ul style="list-style-type: none"> <li>• The [DARK] parameter contains the name of the dark current map (note: this image must not contain the offset signal).</li> <li>• The [OFFSET] parameter contains the name of the offset image (or bias image).</li> <li>• The [FLAT] parameter contains the name of the flat-field image (note: this image must not contain the offset signal or the dark current signal).</li> </ul> <p>Example: we want to process a sequence of 3 raw images of the SH2-136 nebula. The image names are SH136_1.PIC, SH136_2.PIC and SH136_3.PIC. We have also the images OFFSET.PIC, DARK.PIC and FLAT.PIC. Load the first image in the sequence:</p> <p><i>load</i> SH136_1</p> <p>With the mouse, select a rectangle typically 50 pixels in width containing only one non-saturated star. This rectangle will be used for matching the images. Then type:</p>

			<p><i>full_pr</i> SH136_ RESULT dark offset flat 3</p> <p>The final image is fully preprocessed and corresponds to the combination of the 3 input images (simple addition).</p> <p>The <i>full_pr</i> command also created 3 images in the current directory: RESULT1, RESULT2 and RESULT3 in this example). These images are fully preprocessed. So, you may choose between several combination methods. for example, for a simple addition, you will do:</p> <p><i>add2</i> RESULT 3</p> <p>For a median combination, you will type:</p> <p><i>smedian</i> RESULT 3</p> <p>And of a sigma-clipping combination, you will do:</p> <p><i>composit</i> RESULT 1.5 1 3</p> <p>You can also use the <i>add3</i> command that selects the finest images for the addition.</p>
189	<i>full_spec</i>	[in] [out] [offset] [dark] [flat] [flag] [line width] [number]	<p>Automatic processing of a sequence of spectra. Computes the preprocessing, corrects orientation, registers the sequence to the first spectral image, and finally adds the images.</p> <ul style="list-style-type: none"> <li>- [INPUT]: is the generic name of the series to be processed.</li> <li>- [OUTPUT]: is the name of the pre-processed and realigned images.</li> <li>- [OFFSET]: is the name of the offset image.</li> <li>- [BLACK]: is the name of the image of black.</li> <li>- [FLAT]: is the name of the flat image.</li> <li>- [FLAG]: is a marker, it is at 1 if we want to recenter with a line in emission, and at 0 if we want to recenter with a line in emission.</li> <li>- [STRIP WIDTH]: width of the spectral line in pixels.</li> <li>- [NUMBER]: is the number of images in the series.</li> </ul> <p>Before launching the command surround a spectral line correctly marked in the spectrum. First, draw a rectangle in the current image with the mouse around a significant spectral line then run the command <i>full_spec</i>.</p>
190	<i>full_spec2</i>	[in] [out] [offset] [dark] [flat] [number]	Same as <i>full_spec</i> but doesn't require marker and width of the spectral input.
191	<i>gamma</i>	[coefR] [coefG] [coefB]	Applies a transformation to the level of RGB layers of a true color images according to a power function (correction known as "gamma"). If the image in memory is greyscale, it is converted into an RGB image. The levels of a given layer are accentuated if the corresponding coefficient has a great value. The characteristic excursion of the coefficients lies between 0,1 and 5. See also the command <b>Gamma adjustment...</b> of <b>Visu</b> menu.

			<p>For correction functioning it is important to set visual limits at least not smaller than the lowest and highest levels contained in the image! For example, as:</p> <p><i>visu</i> 32767 0</p> <p>Or the pixels with the brightness higher than these values become truncated.</p>
192	<i>gauss</i>	[sigma]	<p>Convolve the current image by a two-dimensional Gaussian whose width is given by the parameter [SIGMA].</p> <p>The convolution of an image by a Gaussian with a small sigma (less than 1) can be used to attenuate the noise. A high value for sigma produces a fuzzy effect.</p>
193	<i>gauss2</i>	[sigma]	<p>Same command as <i>gauss</i>, but the processing here is done for the whole image, including the sides (that makes this command slower than the <i>gauss</i>).</p>
194	<i>gauss3</i>	[sigma] [border size]	<p>Same function as the <i>gauss</i> but while excluding from calculation a border of the image having a size in pixel of equal to [size edge]. This makes it possible to filter images with minimum artefacts. Works only with RGB images.</p>
195	<i>gauss4</i>	[radius] [delta]	
196	<i>gc</i>	[ad1] [dec1] [ad2] [dec2]	<p>Draws a great circle between two points with equatorial coordinates <math>(ra_1, dec_1)</math> et <math>(ra_2, dec_2)</math> — after astrometric reduction of course.</p>
197	<i>gen_out</i>	[file] [text] [x1] [x2] [step]	<p>Utility function to generate an output file for command <i>data_anim</i>, this last allowing to produce dynamic spectra.</p> <p>[NAME] is the name of output LST file.</p> <p>[TEXT] is the generic name of the first column of LST file.</p> <p>[X1] and [X2] are the date limits of the second column (normally the Julian day).</p> <p>[STEP] is the temporal step of the second column.</p> <p>For example:</p> <p><i>gen_out</i> OUT R 2310 2317 0.5</p> <p>Generates this file OUT.LST contains:</p> <pre>r01 2310.500000 r02 2311.000000 r03 2311.500000 r04 2312.000000 r05 2312.500000 r06 2313.000000</pre>
198	<i>get</i>	[x] [y]	<p>Returns the intensity of the pixel at coordinates ([x], [y]).</p>
199	<i>GET_DSLR</i>		<p>Loads the latest image stored in a Digital DSLR CompactFlash (Canon EOS compatible).</p> <p>Isn't functional in IRIS 5.59</p>

200	<a href="#">gnuplot</a>	[x1] [x2] [y1] [y2]	
201	<a href="#">gr</a>	[ordre] [lambda]	
202	<a href="#">gr2</a>	[ordre] [lambda]	
203	<a href="#">gradx</a>	[option]	Filters the current image with a gradient along the X axis. The option (1 or 2) allows to choose the orientation of the gradient (1=left or 2=right). For example:  <i>load</i> M51 <i>gradx</i> 1 <i>offset</i> 1000 <i>visu</i> 1200 800
204	<a href="#">gradx2</a>	[option]	Same command as <a href="#">gradx</a> , but with a stronger effect.
205	<a href="#">grady</a>	[option]	Filters the current image with a gradient along the Y axis. The option (1 or 2) allows to choose the orientation of the gradient (1=up or 2=down). For example:
206	<a href="#">grady2</a>	[option]	Same command as <a href="#">grady</a> , but with a stronger effect.
207	<a href="#">grady3</a>	[option]	Same command as <a href="#">grady</a> , but with even stronger effect.
208	<a href="#">grid</a>	[map file] [intensity]	The <a href="#">grid</a> command is a complement to the <a href="#">map</a> command. It draws on the image, coordinate grids (parallels and meridians) corresponding to an already existing map. The intensity of the grid of [INTENSITY]. This map has an associated file, [LIST]. For a description of the files and the maps, read the documentation for the <a href="#">map</a> command. Here, we will just describe the last 3 parameters of these files, since they are directly related to drawing the grids: <ul style="list-style-type: none"> <li>• LONGSTEP: the step size in longitude of the grid, in integer degrees. It is advisable not to use too small step in order to avoid overloading the screen and making the map unreadable. For the same reasons, GRID automatically stops drawing near the poles.</li> <li>• LATSTEP: the step size in latitude of the grid, in integer degrees. The same remarks as for LONGSTEP apply to the size of the step.</li> <li>• FLAG_T: a reserved parameter.</li> </ul>
209	<a href="#">gsc</a>	[ra] [dec] [x_dim] [y_dim] [x_pixel] [y_pixel] [focal length]	Isn't functional in IRIS 5.59.
210	<a href="#">histo</a>		Calculates the histogram of the in-memory image and produces file HISTO.DAT in the working directory. This function calculates also the cumulated histogram (file CUMUL.DAT) and the opposite cumulated histogram (file CUMUL_INV.DAT).
211	<a href="#">hsi2rgb</a>	[in_h] [in_s] [in_i] [out_r] [out_g] [out_b]	The <a href="#">hsi2rgb</a> command converts a color image defined by its color components Hue, Saturation and Intensity into a trichromatic image in Red, Green, and Blue. The parameters are: <ul style="list-style-type: none"> <li>• [H], [S], [I]: the names of the H, S, I components, respectively.</li> <li>• [R], [G], [B]: the name of the R, G and B components.</li> </ul> See also <a href="#">rgb2hsi</a> , <a href="#">trichro.t_tr</a> .

212	<i>identify</i>	[name]	Provides information about the device that created the file. Example:  <i>identify</i> DSCN2045.CRW
213	<i>image2spec</i>	[file] [line/mm] [dist] [p_zero] [pixel]	Carries out the spectral calibration of an image spectral profile when the position of grating zero order is accessible. Parameters are: [FILE]: the name of the file of the calibrated spectral profile which will be created on the disc. It is an ASCII file with two columns. In the first column one can find the wavelength in angströms and in the second the intensity of the spectrum. [LINE/MM]: the number of grooves by millimeter of the grating. [DIST]: the distance separating the grating from the CCD in millimeters. [P_ZERO]: the position in pixel of the center of the zero-order image along the horizontal axis of the image. [PIXEL]: size of the pixel along the horizontal axis of the image in millimeters.
214	<i>import</i>	[name] [x] [y] [header] [byte number by pixel] [reverse (0 or 1)]	Imports images with a nonstandard format. The program skips the header at the beginning of the file. The number of bytes in the header should be specified in the parameter [HEADER]. The [#BYTE] parameter indicates whether the pixels are coded on one or two bytes. If the coding is on two bytes, you must indicate the order of the bytes in the 16 bit word in the parameter [REVERSE]. If [REVERSE]=0, they are in the INTEL format (most significant/least significant), while if [REVERSE]=1, they are in the MOTOROLA format (least significant/most significant). Finally, the parameters [X] and [Y] contain the image format in pixels along the X and Y axes, respectively (the X axis is the one which is read more quickly in the file). Example:  <i>import</i> EXTERN.IMG 512 800 256 2 0  Imports the image EXTERN.IMG with the following characteristics: pixels coded in 16 bit INTEL format, 256 byte header, 512x800 format. See also <i>export</i> .
215	<i>import_asc</i>	[name]	Imports an image that was saved in a 3-row ASCII format. The two first rows contain the pixel coordinates (origin in (1,1)), whereas the third row contains the pixel intensity. The extension .ASC is added automatically. Note that this file can be very big for large images. See also <i>exportasc</i> .
216	<i>import_asc2</i>	[nom] [X-size] [Y-size]]	Loads an image stored in an ASCII file in the simple form of a single column of real numbers. The length of this vector must be equal to [SIZE X] x [SIZE Y].
217	<i>INFO</i>		Returns information about current image (size, date/time of exposure, integration time). For a FITS file the command return also principal keyword of the header.
218	<i>INFO_ASTRO</i>		Displays information about astrometric reduction.
219	<i>init_date</i>	[description file]	Modifies the date in the header of a sequence of images. The names of the images and the new corresponding dates are defined in a text file (list file, extension .LST). The typical is:

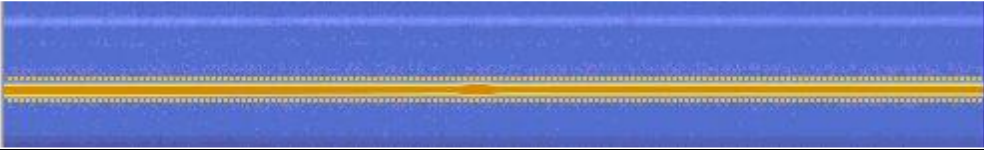
			<p>VAR1 13/12/2005 16:49:20.3  VAR2 13/12/2005 16:52:39.8  VAR3 13/12/2005 16:55:00.4</p> <p>Let us suppose that the name of this file is FILE.LST. If one makes  <i>init_date</i> FILE</p> <p>Iris loads the image VAR1 and defines the new date in the header, here December 13 2005 at 16 hours 49 minutes 20.3 seconds. The file image VAR1 is saved automatically, but with an updated header. Iris makes in the same modifications with the images VAR2, VAR3. It can be useful for <i>trans2</i> command, for example.</p>
220	<i>insert</i>	[in1] [in2] [mask] [value]	<p>For a given pixel of coordinate <math>(x, y)</math> in the [IN1], [IN2] and [MASK] images, if the intensity in the [MASK] image is equal to [VALUE] then the in-memory intensity image pixel is [IN2], else the intensity is [IN1]. Example:</p> <p><i>load</i> M51  <i>offset</i> -500  <i>clipmin</i> 0 0  <i>save</i> MASK  <i>insert</i> MASK M51 MASK 0</p> <p>In IRIS 5.59 correctly works only with greyscale images or crushes the program.</p>
221	<i>integer</i>	[flag]	<p>flag=0 -&gt; real translation - flag=1 -&gt; integer translation. When activated function <i>trans</i> applies only integer translations instead of real.</p>
222	<i>jd2date</i>	[julian day]	<p>Converts Julian day to date.</p>
223	<i>jpg2pic</i>	[in] [out] [first index] [number]	<p>Converts a sequence of JPEG file to PIC or FITS sequence (the final format is dependent of the choice in the setup dialog box of <b>File</b> menu).</p> <p>Consider the input sequence IM1.JPG, IM2.JPG and IM3.JPG. To convert to image R1.PIC, R2.PIC and R3.PIC (or R1.FIT, R2.FIT and R3.FIT), enter the command:</p> <p><i>jpg2pic</i> IM R 3</p>
224	<i>l_add</i>	[line1] [line2]	<p>Computes for each row of the current image sum of the lines between [LINE1] and [LINE2]. The sum is maximized to 32767. The "L_" commands (line commands) are adapted for spectra processing.  <a href="#">Click here for details.</a></p> <p>The result is represented as a new image with the same width as the input image and 20 identical lines in height. Each pixel corresponds to the mean of the input lines in the current row.</p>
225	<i>l_add_norm</i>	[name] [number]	

226	<i>l_add2</i>	[line1] [line2] [sky background level in ADU]	Same as <i>l_add</i> but by taking into account the noise for the calculation of a weight function. [SKY BACKGROUND] is the present sky level in ADU (Analog Digital Unit). [GAIN] is the camera gain (i.e 2 electrons/ADU). [READOUT NOISE] is the noise of camera in electrons (i.e. 18 electrons).
227	<i>l_bin</i>		Carries out an operation of binning along vertical axis on a spectrum whose dispersion axis is horizontal. The addition zone along the vertical axis is such as with final the result of the addition contains 93% of the information of the real spectrum. Iris calculates the optimal addition width zone for reduce noise in the result. The max intensity in the binned image is normalized to 32766. The result is an image of the spectral profile where this one is to duplicate 20 times along the vertical axis. Before run the command you must frame the spectrum of a rectangle to be traced with the mouse.
228	<i>l_bin2</i>		Same function that <i>l_bin</i> , but the normalization is carried out starting from the most intense pixel which is in the selection zone and not over the entire length of the spectrum.
229	<i>l_bin3</i>	[binning wide]	
230	<i>l_blur</i>	[wide]	
231	<i>l_calib</i>	[star] [neon] [first line position] (LHIRES 1200)	Similar to <i>l_calib2</i> but applies only to spectra acquired with the 1200 l/mm grating. It uses neon lines to perform the spectral calibration. The 5 neon lines are at wavelengths: 6506.528 6532.882 6598.953 6678.276 6717.043
232	<i>l_calib2</i>	[star] [first line position] (LHIRES 2400)	The command <i>l_calib2</i> uses the positions of water vapor telluric lines to perform the calibration. Atmospheric lines are more intense in the Halpha area. Warning, this command only works with spectra centered approximately on the Ha-line using the LHIRES III, with a 2400 l/mm grating. It is of course also important that the telluric lines are visible with good contrast (This is almost always the case in the summer season provided the star is not too faint). The command <i>l_calib2</i> is used by the dialog box Treatment spectra LHIRES III - 2400 during the spectral calibration. the wavelengths of the lines used (angstroms) are: 6532.359 6543.907 6548.622 6552.629 6572.086 6574.852 6580.785

			6586.570 The software automatically performs an adjustment using a second order polynomial. The software identifies the position of the telluric lines for you. The polynomial coefficients are returned and a level 1b spectrum is created in the working directory. Two parameters are needed to produce the level 1b spectrum. The filename (Iris adds the extension .dat to the filename), and the approximate position in pixels along horizontal of one of the telluric lines. The line in question is at the wavelength 6574.852 A. It is distinct and easy to identify unambiguously.
233	<i>l_correl</i>	[name]	Computes the shift in pixel unit along the x-axis of the in-memory spectrum and the [NAME] file spectrum. Before running <i>l_correl</i> define with the mouse a rectangle for the cross-correlation computation.
234	<i>l_count</i>		Computes the mean level of a 2D spectrum (in ADU or Analog/Digital Unit). The concerned part is selected with mouse.
235	<i>l_curve</i>	[line1] [line2] [radius]	Same as <i>l_add</i> but the binning is made along a curved spectrum. The radius of curvature of the spectrum is [RADIUS]. <a href="#">Click here</a> for details.
236	<i>l_curve_test</i>	[line1] [line2] [radius]	Draws line along the spectrum to test parameters of <i>l_curve</i> command.
237	<i>l_div</i>	[name] [coefficient]	Same principle as the <i>l_sub</i> command but dividing the image in memory by the image defined in the [NAME] parameter. The result is multiplied by the constant given in [COEFICIENT]. <i>l_div</i> is an important tool in the processing of spectral data or images obtained in drift-scan mode.
238	<i>L_EQUAL</i>		For each column of the image, the software calculates a median value specific of the intensity of the pixels and subtracts this median value of the whole column of pixels. The median is calculated between two vertical positions defined interactively after initiating the command using the mouse. Example: to remove the gradient parasites around the solar disc (phenomenon of diffusion of the light by optics and the atmosphere) so as to improve the observation of the weak protuberances. Note 1: To create a red version of the image:  <i>save R</i> <i>fill 0</i> <i>save G</i> <i>save B</i> <i>t_tr</i>  Note 2: the command <i>l_equal</i> can also be used to eliminate the darkening center/border from the disc.
239	<i>l_expand</i>	[size]	Creates a new image with [height] identical lines equal to the line obtained with, <i>l_median</i> , or <i>l_add</i>
240	<i>l_gauss</i>	[sigma]	Convolution of a vector image by a gaussian function. The sigma of the gauss function is given in parameter.
241	<i>l_median</i>	[line1] [line2]	Same command as <i>l_add</i> except that a median is applied instead of a mean.

242	<a href="#">l_median_curve</a>	[line1] [line2] [radius]	Useful for spectra processing. Same command as <a href="#">l_median</a> but along a curved spectrum. The radius of curvature in pixels is [RADIUS]. See examples <a href="#">here</a> .
243	<a href="#">l_merge</a>	[file #1] [file #2] [x1] [x2]	Merge two spectra [FILE #1] and [FILE #2]. The point of at the coordinate [X1] (pixels unit) into the spectrum #1 correspond to the point [X2] into the spectrum #2. The intensity is normalized at around this point. <a href="#">Click here</a> for an example.
244	<a href="#">l_merge2</a>	[file #1] [file #2] [x1] [x2]	Same as <a href="#">l_merge</a> . The only difference: the spectra are not normalized at the common point.
245	<a href="#">l_noise</a>		Compute the mean level and the RMS noise of a 1D spectrum - The wavelength limits of the computation are selected with the mouse.
246	<a href="#">l_norme</a>	[spectre] [lambda] [dlambda]	
247	<a href="#">l_opt</a>		An easy way to use optimal extraction function of the 1D spectrum from the 2D spectrum. The program calculates a 1D profile by adding the pixels in the 2D spectrum, column by column. The calculation uses pixels located between the upper and lower limits of the rectangle defined with the mouse. This calculation minimizes noise giving less importance to portions of the images that have received little starlight. See <a href="#">l_optbin</a> .
248	<a href="#">l_opt2</a>	[input] [output] [number]	Computes an optimized binning on a sequence of 2D spectra.
249	<a href="#">l_optbin</a>	[line1] [line2] [gain in e/ADU] [RON in electrons]	Optimal extraction for CCD spectroscopy. <a href="#">Click here</a> for a description.
250	<a href="#">l_optbin2</a>	[line1] [line2] [gain in e/ADU] [RON in electrons]	Optimal extraction for CCD spectroscopy. <a href="#">Click here</a> for a description.
251	<a href="#">l_ori</a>		Returns the orientation of a spectrum relative to the horizontal axis. At first draw a rectangle in the current image with the mouse around the spectra.
252	<a href="#">l_plot</a>	[Dimension]	Creates a new image with a plot of the line obtained with <a href="#">l_median</a> , or <a href="#">l_add</a> . The width of the plot is the line width, and its height is [height]. This command produces also the ASCII file PLOT.LST.
253	<a href="#">l_pos</a>	[flag] [width]	Computes the position of a spectral line. If FLAG=0 the line is in absorption. If FLAG=1, the line is in emission. WIDTH is the typical width of the line (FWHM). Select un area around the line with the mouse then execute <a href="#">l_pos</a> .
254	<a href="#">l_pos2</a>		Precise evaluation of the position of a spectral in a 1D spectrum. The line influence zone is selected with the mouse. The command return the position in pixel and also the FWHM of the line.
255	<a href="#">l_posy</a>		Return the vertical position of a 2D spectrum.
256	<a href="#">l_register</a>	[in] [out] [flag] [width] [number]	Registers a sequence of [NUMBER] spectral images with the aim of a spectral line. If FLAG=0 the line is in absorption. If FLAG=1, the line is in emission. [WIDTH] is the typical width of the line (FWHM). Select un area around the line with the mouse then execute <a href="#">l_register</a> .
257	<a href="#">l_register2</a>	[in] [out] [number]	

258	<a href="#">l_register3</a>	[in] [out] [number]	Registration of a sequence of 1D spectra along the spectral axis only, by using the intercorrelation method.
259	<a href="#">l_registry</a>	[in] [out] [number]	Registration of a sequence of 2D spectra along the space axis only (vertical axis).
260	<a href="#">l_registry2</a>	[in] [out] [number]	
261	<a href="#">l_sinc</a>	[factor]	Scale a spectral image by the coefficient [FACTOR]. This function uses the <i>sinc</i> interpolation. For an application example <a href="#">click here</a> .
262	<a href="#">l_sky</a>	[line1] [line2] [line3] [line4]	Computes the median value for each row of the current image between [LINE1] and [LINE2]. This gives the value $V_1$ . Computes the median value for each row of the current image between [LINE3] and [LINE4]. This gives the value $V_2$ . The value $(V_1 + V_2)/2$ is computed and subtracted to each row of the current image. <a href="#">l_sky</a> is useful to correct gradient background for spectral image. In the normal situation the area between [LINE1] and [LINE2] is upper the spectrum and the area between [LINE3] and [LINE4] is in the opposite side relative to the spectrum.
263	<a href="#">l_sky_curve</a>	[line1] [line2] [line3] [line4] [radius]	Same as <a href="#">l_sky</a> but along a curved spectrum. The [RADIUS] parameter contains the radius of curvature of the spectrum in pixels. <a href="#">Click here</a> for example of use.
264	<a href="#">l_sky_smt</a>	[line1] [line2] [line3] [line4]	
265	<a href="#">l_sky2</a>		<p>Command allowing to estimate the level of the sky background on both sides of a spectrum. The sky background is modeled by calculating for each column of the images taken independently the median value of the intensities of the pixels in two zones that one definite interactively. The operator defines these two zones by 4 clicks of the mouse as shown in the following image:</p> <p>The click order of the points is not critical. <a href="#">l_sky2</a> then subtract from all the columns the average of the two median intensities calculated in each one of them. The effect is to bring the level of the sky background to zero.</p> <p>It is an essential operation before being able to extract the spectral profile from a spectral image because it fixes the origin of the scale of the intensities. <a href="#">Click here</a> for an illustration.</p>
266	<a href="#">l_sky3</a>		Command, very similar to <a href="#">l_sky2</a> , allowing to estimate the level of the sky background on both sides of a spectrum. The sky background is modeled by fitting linear lines distinct for each column from the image. The pixels of the image which are used to calculate this fit are in two zones on both sides spectrum that the operator define by 4 clicks of the mouse


267	<i>l_sky4</i>		In the case of a difficult very irregular background, you can ask the adjustment to be made using a parabola (second order equation). Corresponding command is <i>l_sky4</i> . All these commands can be invoked via menu <b>Spectro &gt; Remove the sky to a 2D spectra...</b>
268	<i>l_sub</i>	[name]	Subtracts from each line of the current image the line in the [name] image obtained with <i>l_median</i> , or <i>l_add</i> .
269	<i>l_test_curve</i>	[line1] [line2] [radius]	Traces the contour of the binning zone on the image of the spectrum (see image below). The good radius of curvature is found when contours (dotted lines) follow the shape of the spectrum (see image below). 
270	<i>laplacian</i>		Calculates the Laplacian of the image in memory. Correctly works only with greyscale images.
271	<i>LED</i>	[val] [sleep]	
272	<i>lfill</i>	[x0] [value]	Masks the left part of an image relative to the [x0] coordinate. Iris give the level [value] to the masked area. See also command: <i>rfill</i>
273	<i>load</i>	[name]	Loads an image in memory from the current directory (defined in the item Current Directory from the <b>Settings</b> tab – <b>File</b> menu). You can also indicate in which directory to load a particular image by specifying the full path of the image. For instance: <i>load</i> c:\nuit7\m51
274	<i>loadbmp</i>	[name]	Loads an 8-bits BMP file in memory from the current directory.
275	<i>loadbmp24</i>	[name] [r] [g] [b]	Loads a 24-bits BMP file from the current directory and copy the RGB planes in the images [r], [g], [b] respectively.
276	<i>loadbmp24bw</i>	[name]	Load a 24-bits BMP file from the current directory in memory and convert the RGB planes into a B&W images.
277	<i>Loadcam or CONVERTCAM</i>	[name]	Converted RAW file coming from a digital camera into a color image which is displayed (see also <b>Load...</b> command of the <b>File</b> menu). Example: <i>loadcam</i> CRW_0347  Note that no offset is applied to the image. So, for correct white balance it should be subtracted first.
278	<i>loadcam</i>	[name] [r] [g] [b]	That synaxis is inactive in IRIS 5.59
279	<i>loadcam2</i>	[in] [out] [number]	Same as <i>loadcam</i> but for sequences.
280	<i>loadcambw</i>	[name]	Should load debayered BW version of the image like <i>loadcam</i> but it seems that nowadays cameras have to many megapixels for this function to handle. Try instead:

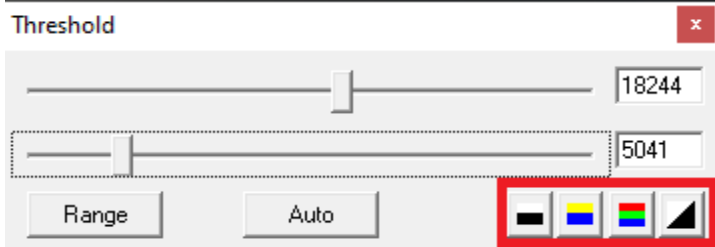
			<i>loadcam</i> your_image <i>cfa2bw</i>
281	<i>loadcanon</i>	[name] [r] [g] [b]	Deactivated in IRIS 5.59
282	<i>loadcfa</i>	[name]	Displays a CFA image extracted from a RAW file (CFA = matrix of colored filters covering the pixels of the electronic sensor).
283	<i>loaddsi</i>	[name]	Loads in memory a FITS image coming from the camera Meade Deep Sky Imager. <i>loaddsi</i> converted YCMG matrix into an 48-bits colors image corrects the ratio aspect induced by the rectangular pixels of the sensor ICX 404K which equips this camera (pixels of 9,6 microns X 7,5 microns). On the other hand, the command does not correct the white balance, an operation which must be carried out specifically according to the context (see the command " <b>RGB Balance...</b> " of the <b>Digital photo menu</b> — typically the layers red, green and blue must be multiplied respectively by 0,53, 1,00 and 2,23).
284	<i>loaddsi2</i>	[in] [out] [number]	Converted a sequence of RAW image Meade DSI (generic name [IN] and .fts extension) to a sequence of color 48-bits images (generic name [out] and .pic extension). The number of images in the sequence is [NUMBER]. The geometrical transformations are carried out by this command. See also the commands <i>loaddsi</i> , <i>convertdsi</i> and <i>convertdsi2</i> .
285	<i>loadjpg</i>	[name]	Loads a 24-bits jpg file from the current directory in memory.
286	<i>loadnikon</i>	[name] [r] [g] [b]	Deactivated in IRIS 5.59
287	<i>loadpng</i>	[name]	Loads a 24-bits png file from the current directory in memory.
288	<i>loadraw</i>	[name]	Analogous to <i>loadcam</i> but doesn't apply debayerisation and displays raw CFA data.
289	<i>loadraw2</i>	[in] [out] [number]	Same performance as <i>loadraw</i> but can save output data as a sequence of images.
290	<i>loadsb</i>	[name]	Loads an image in CCD SBIG (ST4, ST4X, ST6, ST7, ST8...) format into memory. This command also accepts compressed format.
291	<i>loadsx</i>	[name]	Loads into memory an unsigned 16-bits image (dynamic range between 0 and 65535). The level of the pixels is multiplied by 0.5 to bring final dynamics between 0 and 32767. See also: <i>convertsx</i> , <i>signed</i> .
292	<i>loadsx2</i>	[name]	Loads into memory an unsigned 16-bits image. The level of the pixels is not modified, but the images is truncated for intensities higher than 32767.
293	<i>loadsx3</i>	[name]	Loads into memory an unsigned 16-bits image. Value 32767 is subtracted from the intensity of all the pixels. The final level lies between -32768 and 32767.
294	<i>loadtiff</i>	[name]	Loads a 8-bits (since IRIS 5.59 version 48 bits also) uncompressed TIFF file in memory from the current directory.
295	<i>loadtiff24</i>	[name] [r] [g] [b]	Loads a 24-bits uncompressed TIFF file from the current directory and copy the RGB planes in the images [R], [G], [B] respectively.
296	<i>loadtiff24bw</i>	[name]	Load a 24-bits uncompressed TIFF file from the current directory in memory and convert the RGB planes into a B&W image. (In IRIS 5.59 it fails since image isn't BW but still has some colors.)

297	<i>log</i>	[norm]	<p>Calculates the base 10 logarithm of an image where [NORM] is a coefficient which adjusts the maximum dynamics of the output image and Max (input image) is the intensity of the brightest pixel in the input image.</p> <p>The logarithm of an image is used to display the range of intensity levels in one visualization. Often, during deep sky image processing, the <i>offset</i> command is used first to bring the sky background close to level 0. The depiction of faint details is then greatly improved.</p> <p>Let's calculate the logarithm of the image M51.FIT, whose sky background level is around 130:</p> <pre>load M51 offset -100 log 1000 visu 1000 400</pre> <p>With the <i>stat</i> command, you can verify that there are no pixels with intensities over 1000 in the final image.</p> <p>The <i>log</i> command allows you to transform the linear intensity scale into a magnitude scale, to within a constant. This can be very useful in photometry or in various kinds of image representation (for example isophotes visualization).</p> <p>Important note. This function proceeds each RGB channel separately. So, if your image doesn't contain oversaturated areas in which all three channels have the same maximum value and you want to preserve the set white balance then add such pixel somewhere in the corner of you image like that:</p> <pre>load M42 fill 0 save temp1 tri2mono      (it is necessary because put function works only with greyscale images) put 1 1 [REQUIRED_VALUE] mono2tri add M42 log [NORM]</pre>
298	<i>log2</i>	[in] [out] [norm] [number]	Calculates the logarithm of a sequence of images.
299	<i>lrgb</i>	[in_r] [in_g] [in_b] [in_l] [out_r] [out_g] [out_b]	<p>Give tri-color images IN_R, IN_G &amp; IN_B to the program to compute the RGB to HSI conversion, then replace the I image by the IN_L image (luminance image). Finally, the program performs the HSI to RGB conversion for producing the output images OUT_R, OUT_G &amp; OUT_B.</p> <p>See also: <i>rgb2hsi</i>, <i>hsi2rgb</i>, <i>rgb2pca</i>, <i>pca2rgb</i>, <i>trichro</i>.</p>
300	<i>lucam</i>	[exposure] [gain]	

301	<i>lucam_avi</i>	[AVI file] [exposure] [gain] [binning] [duration] [mode]	<p>The first parameter is the AVI file name (stored in the working directory, Iris add the extension .avi for you). The binning factor can be 1 or 2 (use binning=2 for very fast acquisition, up top 130 frames/sec). The parameter [duration] is the duration of the capture in seconds. If [mode] value is 0, a preview is also displayed. If [mode] value is 1, the video stream is captured without displaying it (the higher quality and frame rate mode). Example:</p> <p><i>lucam_avi</i> JUPITER 0.018 2 1 101</p> <p>The equivalent dialog box interface is available from the command <b>Lumenera AVI capture...</b> (<b>Video</b> menu)</p>
302	<i>lucam_set</i>	[exposure] [gain]	Modifies exposure time and gain during preview mode. See <i>lucam_start</i> .
303	<i>lucam_start</i>		Starts preview mode fir Lumenera camera.
304	<i>lucam_stop</i>		Stops the preview mode turned on by <i>lucam_start</i> function.
305	<i>lucam_video</i>	[generic name] [number] [exposure] [gain] [binning] [mode]	<p>Capture a fast sequence of 12 bits images. You can also invoke the dialog box <b>Lumenera video acquisition...</b></p> <p><i>lucam_video</i> test 50 0.1 2 1 1</p> <p>Here the sequence test1, test2, ..., test50 of individual images, corresponding to video frames, is saved into the hard disk.</p>
306	<i>lxalign</i>	[hour] [minute] [degree] [minute]	Apparently function of mount synchronization.
307	<i>lxgoto</i>	[hour] [minute] [degree] [minute]	Apparently function of mount goto request.
308	<i>map</i>	[input map file] [output map file]	<p>Creates an image in a given cartographic projection from an image defined in a different projection. Cartography is a method that allows you to represent, on a plane, a surface that is generally not flat, such as the earth or any other planetary surface. This science is thus necessarily inexact since local or global deformations of the surfaces to be represented are inevitable. The choice of a cartographic projection is generally based on a compromise between different desired properties (for example, global view of the planet, conservation of area, polar view, etc.).</p> <p>For examples and complete description see the section <a href="#">dedicated</a> to this topic.</p>
309	<i>map2rec</i>	[map file] [longitude] [latitude]	The <i>map2rec</i> command is used with the <i>map</i> command. It calculates the Cartesian coordinates of a point on the map, starting from the parameters of a cartographic projection and the planetocentric coordinates ([LONGITUDE], [LATITUDE]) of the point. [LIST] contains the name of the file with the parameters of the projection. For a description of the cartographic files, see the <i>map</i> command.
310	<i>max</i>	[pixel number]	Draw a rectangle in the current image with the mouse before running that command. This will replace the [PIXEL NUMBER] brightest pixels of the rectangle by the median value of the remaining pixels (the median value is computed at each iteration).

			This command is good for interactively removing cosmetic artefacts in an image, such as warm pixels. To remove a single pixel, you will type <i>max</i> 1. But you may also remove several pixels at a time, and even remove a star (e.g. <i>max</i> 30). Can be useful for creating COSME.LST files when the original dark frames are absent for some reasons.
311	<i>me</i>	[spectre #1] [spectre #2] [output]	
312	<i>median3</i>	[coef]	<p>Performs median filtering on the current image. The median value of the intensities of the pixels in a 3x3 matrix around a pixel is calculated for each pixel in the image. The corresponding point in the output image is set to this median value.</p> <p>[COEF] is a parameter that adjusts the strength of the filter (the action of the filter is maximum if [par]=0). If we number from n=1 to 9 the values of a 3x3 matrix sorted on increasing order, <math>I_n</math> represent the intensity of the ordered points in the matrix and if <math>I_{orig}</math> represents the intensity of the point at the center of the 3x3 matrix, <i>median3</i> performs as follows:</p> <ul style="list-style-type: none"> <li>- if the absolute value of <math>I_0 - I_5 &gt; [coef] \times (I_8 - I_2)</math> then the corresponding pixel <math>I_{orig}</math> in the output image will be given the median value <math>I_5</math>;</li> <li>- otherwise, the pixel <math>I_{orig}</math> of the output image will keep the value of <math>I_{orig}</math> from the input image.</li> </ul> <p>Median filtering is an excellent tool for eliminating impulse noise in an image (cosmic rays, interference in one or two pixels, etc.). The original version of the median filter is very energetic. Used without finesse, this filter may suppress useful information (faint stars) and give an artificial texture to the filtered image. This is why a weighting factor has been added to the Iris implementation of the median filter. For images that are not too noisy, the parameter [COEF] is typically between 0.80 and 0.99:</p>
313	<i>median5</i>	[coef]	Same command as <i>median3</i> but with a 5x5 matrix.
314	<i>medianf</i>	[size] [coef]	<p>Carries out same procedure as <i>median3</i> or <i>median5</i> but by using a kernel (zone in which the median is calculated) of adjustable [SIZE]. Size must be imperatively an odd value. For example,</p> <p><i>medianf</i> 9 0.8</p> <p>This command is practical to erase the details of relatively large size.</p>
315	<i>mem</i>	[#iter]	<p>Deconvolution using a Maximum Entropy Method. The process is iterative and contains [#iter] iterations. The size of the image must be square and equal to a power of 2 (see <i>window3</i> and <i>padding</i> commands). Before running the command, select an isolated star with the mouse. It has to have a good signal to noise ratio, but not too bright, in order to avoid saturation. Generally, 15 to 20 iterations are recommended. Try also to have a sky background close to zero (use the <i>offset</i> command) but strictly positive.</p>
316	<i>MEMORY</i>		
317	<i>merge</i>	[name1] [name2] [opacity]	<p>Calculates for each pixel of two images following sum:</p> $I_{new} = I_1 \times (1 - opacity) + I_2 \times opacity$

			Values of opacity higher than 1 and lower than zero are truncated.
318	<a href="#">merge_cfa</a>	[c1] [c2] [c3] [c4]	Recombine a single CFA image separated with <a href="#">split_cfa</a> command.
319	<a href="#">merge_cfa2</a>	[a] [b] [c] [d] [out] [number]	Merges a sequence of splitted CFA images (see <a href="#">split_cfa</a> command) into a unique CFA sequence of images (generic name [OUT]). For merge only one image use the <a href="#">merge_cfa</a> command.
320	<a href="#">merge_hdr</a>	[description file] [threshold]	<p>Generates an HDR image from the files previously used in <a href="#">desc_hdr</a> function.</p> <p><a href="#">merge_hdr</a> DESC 100</p> <p>The result is a 15-bits images of the corona:</p>  <p>In order to compress dynamic range, use the <a href="#">reduce_hdr</a> functions.</p>
321	<a href="#">merge_rgb</a>	[r] [g] [b]	Generates a true color 48-bits image from the R, G and B separate components. The command is synonymous to <a href="#">trichro</a> and <a href="#">tr</a> commands.

322	<i>merge_rgb2</i>	[r] [g] [b] [out] [number]	Merges sequences of the R, G and B separate component of colors images into a sequence of 48-bits sequence.
323	<i>microcat</i>	[ra] [dec] [x_dim] [y_dim] [x_pixel] [y_pixel] [focal length] [magnitude]	Probably used to show synthetic image of the sky with the given parameters from the Microcat catalog (special CD-ROM compiled by Alain Klotz for the Audine project). Now in version 5.59 isn't functional anymore. But all the functional still can be found in <b>Data Base&gt; Display a sky map</b> . At least that menu correctly works with Tycho-2 and USNO-A2.0 catalogs.
324	<i>min</i>	[pixel number]	Same command as <i>max</i> , but applies on pixels having a level lower than the median level of the area (e.g. for removing non active pixels).
325	<i>MIRRORX</i>		Rotates the image around a vertical axis.
326	<i>mirrorx2</i>	[in] [out] [number]	Same as <i>mirrorx</i> but for sequences.
327	<i>MIRRORXY</i>		Invert the X and Y axis.
328	<i>mirrorxy2</i>	[in] [out] [number]	Permutation of axes X and Y for a sequence of images.
329	<i>MIRRORY</i>		Rotates the image around a horizontal axis.
330	<i>mirrory2</i>	[in] [out] [number]	Same as <i>mirrory</i> but for sequences.
331	<i>mmse</i>	[sigma] [size]	Filters adaptatively noise by using the "Minimum Mean-Square Error" method. The [SIGMA] parameter contains the typical value of noise in the sky background. This value may be obtained for example with the STAT command. For example, load the M51.FIT. With the mouse, define a rectangle about 30 to 40 pixels in width, avoiding the galaxy and stars. then run the <i>stat</i> command (contextual menu). Iris returns a standard deviation around 8 (it is the estimation of the sky background). Then type:  <i>mmse 8 0</i>  Correctly works only with monochrome images.
332	<i>mmse2</i>	[level] [size]	Some kind of denoising algorithm
333	<i>modulo</i>	[value]	Computes for each pixel the value modulo [VALUE]. The result is the current image in memory. This command allows producing some isophote effects that may be useful to visualize images with large dynamic ranges. Select the false color palette to a better rendering.  
334	<i>mono2tri</i>	[value]	Converts greyscale image to RGB.

335	<i>mosa</i>	[name] [dx] [dy] [type] (1...5)	<p>Allows you to include a set of elementary images in one image. The images are merged two at a time. The first image is in memory. The second image is designed by the parameter [NAME]. The shift between two points on the images you want to merge is contained in the variables ([DX], [DY]).</p> <p>When the two images have points that overlap (which is not required), the value contained in the variable [type] allows you to choose the state of the overlapping zone in the final image:</p> <ul style="list-style-type: none"> <li>- If [TYPE] = 0: the second input image (the name in IN2) overwrites the first input image (the name in IN1) unless the pixels in the second input image have the intensity zero. In this case, the values from the first input image are used.</li> <li>- If [TYPE] = 1: the first input image overwrites the second input image unless the pixels in the first input image have zero intensity, in which case the values from the second input image are used.</li> <li>- If [TYPE] = 2: the pixels in the output image take the maximum value of the pixels from the two input images.</li> <li>- If [TYPE] = 3: the pixels in the output image take the minimum value of the pixels from the two input images.</li> <li>- If [TYPE] = 4: the pixels in the output image take the average value of the pixels from the two input images.</li> <li>- If [TYPE] = 5: a bilinear interpolation is performed on the common parts of the two input images.</li> </ul> <p>The format of the input images can be different. The <i>mosa</i> command is used primarily to put together several shots of the same object in order to have a single image of this object.</p>
336	<i>mouse_select</i>	[x1] [y1] [x2] [y2]	<p>The command simulates the selection of an area of the image with the mouse. Example, for compute the barycenter of an area limited by the pair of coordinates (20, 94)-(196, 287), enter:</p> <pre><i>mouse_select</i> 20 94 196 287</pre> <pre><i>cog</i></pre>
337	<i>mult</i>	[coef]	Multiplies all the current image pixels by the constant [COEF].
338	<i>mult2</i>	[in] [out] [coef] [number]	Multiplies all the pixels of a sequence of images by [COEF]. The generic name of the [NUMBER] input images is [IN], whereas the generic name of the output images is [OUT].
339	<i>new</i>	[x] [y]	Creates a new image filled with zeros. The image has a size of [X] x [Y].
340	<i>ngain</i>	[norm]	Normalizes the median level of an image to [NORM] by multiplying the image by a constant value. This command is generally used to change the median level of a flat-field image, in order to prepare a median sum.

341	<i>ngain2</i>	[in] [out] [norm] [number]	Normalizes the median level of a sequence of [NUMBER] images having a generic name [IN] to [NORM] by multiplying each image with a constant value. The generic name of the output images is [OUT].
342	<i>ngain3</i>	[in] [out] [number]	Normalization of a sequence of images. For the difference of <i>ngain2</i> it is necessary to define with the mouse the zone of the image in which the calculation is carried out.
343	<i>noffset</i>	[norm]	Normalizes the median level of an image to [NORM] by adding a constant value to the image.
344	<i>noffset2</i>	[in] [out] [norm] [number]	Normalizes the median level of a sequence of [NUMBER] images of generic name [IN] to [NORM] by adding a constant value to each image. The generic name of the output images is [OUT].
345	<i>noffset3</i>	[in] [out] [norm] [number]	<p>Normalizes the median background level of a series of [NUMBER] images having the generic name [IN] to [NORM]. The generic name of the output images is [OUT]. You have to select the area in which the background level will be estimated before running the command. Use the mouse for that. This command is very useful to normalize background levels in a series of planetary images. For example, the <i>pregister</i> command works best when the background level is close to zero, and when the image background is uniform including near the image edges (windowing is often necessary to avoid edge effects). So, a typical sequence to register a series of planetary images will be for example:</p> <pre> <i>window2</i> JUP I 5 5 170 170 9      (to suppress bad border) <i>noffset3</i> I J 0 9                (to make zero level background around the planet) <i>pregister</i> J K 256 9             (to register the images) <i>mult2</i> K K .3 9                 (to avoid integer overflow after summation) <i>add2</i> K 9                       (to composite the 9 images) </pre>
346	<i>number</i>	[generic name]	Returns the number of images of an image sequence and mid-date of the acquisition.
347	<i>offset</i>	[value]	Adds the constant [VALUE] to the current image. The constant can have a negative value.
348	<i>offset2</i>	[in] [out] [offset] [number]	Adds the value [OFFSET] to [NUMBER] images having the generic name [IN]. The parameter [OUT] contains the generic name of the output images.
349	<i>opacity</i>	[mask name] [coefficient]	<p>The opacity mask attenuates some part of the in-memory image proportionally to the mask intensity. The [COEFFICIENT] parameter adjusts the global intensity.</p> <p><a href="http://www.astrosurf.com/buil/iris/tutorial1/doc6_us.htm">http://www.astrosurf.com/buil/iris/tutorial1/doc6_us.htm</a></p>
350	<i>opt</i>	[dark name]	<p>In long exposure CCD imagery, one of the major difficulties posed during preprocessing is the dark current correction. This interference signal due to thermal charges is added to the signal produced by the observed objects. The problem is to suppress this interfering component of the image because it is noise that impairs the detectability.</p> <p>A classic solution is to take an exposure of the object under study with an integration time <math>T</math>, then to take another exposure of length <math>T</math> while placing the detector in total darkness. This last exposure is called the dark current map. This map is a constant, to within a coefficient, for a given CCD. To first</p>

order, the coefficient is proportional to the temperature of the CCD and to the integration time. In the preceding procedure, the dark current map is simply subtracted from the image of the object. This is, however, far from the ideal solution. In fact, this procedure implies that a dark current map must be taken after each image of the object. This is very constraining when the exposure time is several minutes or more.

Things seem to go better if the temperature of the CCD is perfectly stable. In this case, a priori only one map is necessary. It can be taken, for example, at the beginning of the observing session, and can be used to correct all the images. If the exposure time is not the same on the dark map and the image to be processed, the dark map must be multiplied by a coefficient before the subtraction. The coefficient is the ratio of the exposure time of the image to the exposure time of the dark map.

Besides the fact that is a delicate matter to maintain the detector at a fixed temperature, this method has the following inconvenience - the dark map has its own noise (readout noise) and when the map is subtracted from the images, this noise is actually added to the images.

There is a more efficient method:

- Take several (5 to 10) dark images with integration times that are not necessarily equal, but long enough to be sensitive to the dark current (they should have the same duration as the observation exposures). The CCD should be cooled to reduce the readout noise as much as possible.
- For each exposure in darkness, another one with a minimal integration time is taken. This provides the offset map.
- For each of the dark images, subtract the corresponding offset map. The resulting images then contain only the thermal component of the signal.
- Sum all the images from the previous step to obtain the dark map. This adds the thermal contributions from each image, but averages the readout noise. This map can be considered constant for a given CCD (it is still good to redo this procedure every 1 or 2 months to allow for any possible change in the electronic characteristics of the detector).
- Now a given image must be corrected. The difficulty is in finding a multiplicative coefficient, to apply to the dark map, which will optimally correct this image. This is what the command *opt* does.

Iris solves this problem nearly instantaneously using an analytical approach.

You have to select an area typically 30 to 100 pixels in width with the mouse. Then run the command. [DARK NAME] is the name of the dark current map. The program will return the coefficient to apply to the dark current map to create a new map that is optimal for the image to process. Example:

*opt* DARK

Calculates the optimal coefficient for the dark map DARK.PIC to correct the current image.

			The <i>opt</i> command should be systematically used to preprocess deep sky images. With this command in hand, you do not need to worry about taking dark images during the night. The result is quite good because the criteria chosen minimize the noise.
351	<i>opt_sub</i>	[name]	Convolution of the image in memory by a kernel calculated by the command so that the PSF of stars are most similar to the image [name]. The kernel is saved at ends under the name @k.
352	<i>opt2</i>	[in] [dark] [output] [number]	Same command as <i>opt</i> but applies on a sequence of [NUMBER] images having the generic name [IN]. [OUT] is the generic name of the output images.
353	<i>opt3</i>	[in] [dark] [output] [number]	Same command as <i>opt2</i> the with a more complete procedure (it is not necessary to select a zone in the image).
354	<i>optcfa/opt_cfa</i>	[dark name]	Analog of command <i>opt</i> but for CFA images supposedly.
355	<i>opte</i>	[dark name]	
356	<i>padding</i>	[xl] [yl]	Sets the size of the current image to [LX], [LY]. If the image is larger than ([LX], [LY]), it is truncated. If the image is smaller than ([LX], [LY]), it is completed by pixels of zero intensity. This command is generally useful for comparing images taking with different CCD or before using commands that perform Fast Fourier Transforms for which the image must have a size equal to a power of 2. For example, the command:  <i>padding</i> 256 256  guarantees that the image in memory has a size of 256x256 pixels.
357	<i>padding2</i>	[in] [out] [lx] [ly] [number]	Same function that <i>padding</i> , but applying to a sequence of images.
358	<i>pano_edge</i>	[threshold]	Check <a href="#">pano processing tutorial</a>
359	<i>pano_edge2</i>	[in] [out] [threshold] [number]	
360	<i>pano_max</i>	[name] [number]	
361	<i>pano_mean</i>	[name] [number]	
362	<i>para</i>	[pixel number]	
363	<i>para2</i>	[pixel number]	
364	<i>pca2rgb</i>	[c1] [c2] [c3] [r] [g] [b]	The <i>pca2rgb</i> command performs the reverse transformation of the <i>rgb2pca</i> command, that is, it goes from the space of the principal components to the space of the fundamental colors (R,G,B). To do this, this function needs the file containing the matrix of the eigenvectors of the covariance matrix of the three initial trichromatic images (pci.lst). For more information on this technique, see the <i>rgb2pca</i> command.
365	<i>permute</i>		Divides image on quarters and rearranges them with respect to center. First quadrant is interchanged with the third one and the second with the fourth respectively. Can negate consequences of <i>fconv</i> .
366	<i>perspective</i>	[alpha] [beta] [gamma] [scale]	Command analogous to <b>Perspective adjustment...</b> command of <b>Geometry</b> menu. Allows you to quickly verify how your adjustments would look like. If you want to manually set the interpolation algorithm and output image size use <b>Correct perspective...</b> of <b>Geometry</b> menu

367	<i>phot</i>	[radius1] [radius2] [radius3] [option (1, 2, 3)]	<p>Computes aperture photometry. After running the command, one or several circles appear in the field. By centering the circles on the stars and then clicking on the left button of the mouse the program returns information about the star intensity.</p> <p>If [OPTION]=1, a simple circle appears. The returned information is the sum of the intensities of the pixels inside the circle (i.e. the sum of the star intensity and the sky background).</p> <p>If [OPTION]=2, two circles appear. The signal of the star plus the sky background is integrated in the inner circle. In the area between the two circles, the mean sky background is measured. <i>Phot</i> then returns the signal of the star alone (sum of the intensities I, and instrumental magnitude M). The inner circle radius is [RADIUS1], and the outer circle radius is [RADIUS2].</p> <p>If [OPTION]=3, three circles appear. The most outer circle radius is [RADIUS3]. It is the same method as previously, except that the background level is calculated from the area between circles 2 &amp; 3. This allows sometimes to avoid close stars to the measured star, that may bias the measure of the sky background.</p> <p>To exit the <i>phot</i> mode, execute the command <i>photoff</i>.</p> <p>Examples:</p> <pre><i>phot</i> 6 0 0 1 <i>phot</i> 6 11 18 3</pre> <p>Note: it is important that a rather large number of pixels are used to determine the sky background. Moreover, as far as the inner circle is concerned, it has to be large enough to contain the whole star, but not too big in order to minimize noise level.</p>
368	<i>photm</i>	[radius1] [radius2] [radius3] [option (1, 2, 3)]	<p>This command is close to the <i>phot</i> command. The difference is that a median is used to compute the sky background level instead of a simple average. It may be useful to minimize the effect of feeble stars in the measure annulus. To exit the <i>phot</i> mode, execute the command <i>photoff</i>.</p>
369	<i>photoff</i>		<p>Terminates regime of photometry turned on with <i>phot</i> or <i>photm</i> commands.</p>
370	<i>pic_anim</i>	[input] [output]	<p>Function very close to <i>data_anim</i>. The latter calculates interpolations starting from data curves, in particular of spectra (click here for an example). <i>pic_anim</i> applies to 2-D images to improve fluidity of the animation of a sequences. For that of the intermediate images are calculated by simple linear interpolation starting from the acquired images.</p> <p>The parameter [INPUT] indicate the name of a text file which respectively contains on two columns the name of the acquired images and date of acquisition of these images (or all other identifying function of time, as for example an index value which goes into increasing).</p> <p>The parameter [OUTPUT] indicate the name of a text file which respectively contains two columns the name of the interpolated images and dates for which the interpolation is calculated (or an identifier function of time, in conformity with that used in the input file).</p> <p>Suppose 5 images to be interpolated with the names MET1, MET2, MET3, MET4 and MET5. We create in the working directory a text file of name IN.LST containing (use a text editor for that):</p>

met1 1  
met2 2  
met3 3  
met4 4  
met5 5

We create the output file OUT.LST

r1 1.00  
r2 1.25  
r3 1.50  
r4 1.75  
r5 2.00  
r6 2.25  
r7 2.5  
r8 2.75  
r9 3.00  
r10 3.25  
r11 3.50  
r12 3.75  
r13 4.00  
r14 4.25  
r15 4.50  
r16 4.75  
r17 5.00  
r18 4.50  
r19 4.75  
r20 5.00  
r21 4.50  
r16 4.75  
r17 5.00

The images R1 and R5 for example will be identical to images MET1 and MET2 (correspondence of the dates). But moreover, between the two images observed, command *pic\_anim* will generate the intermediate images R2, R3 and R4, and so on for the whole of the sequence.

Note: file OUT.LST can be created automatically with the assistance of command *gen\_out*, which is quite practical for long sequences. In the example one will make:

*gen\_out* OUT R 1 5 0.25

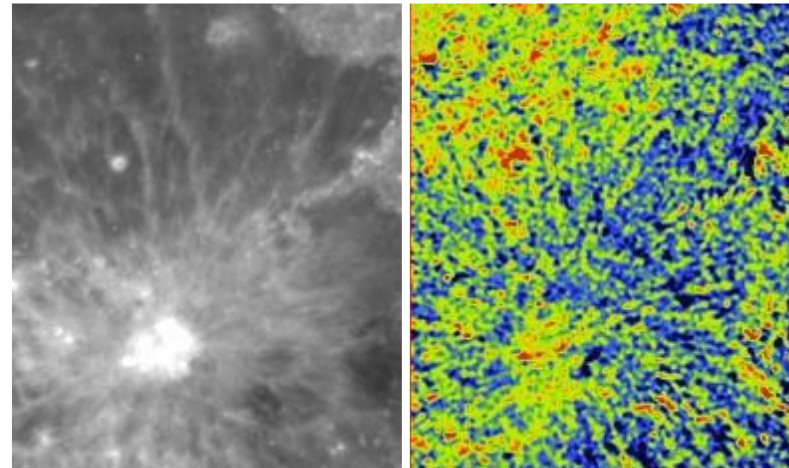
			<p>After having saved the file IN.LST and OUT.LST, we produce the interpolated sequence:</p> <p><i>pic_anim</i> IN OUT</p> <p>The sequence R1 ... R17 synthesized can be visualized with the Animation... command from <b>Visualisation</b> menu. You can also save the sequence in the form of BMP images for produce an animated GIF or a AVI film for example with the assistance of an adequate software:</p> <p><i>pic2bmp</i> R RR 17</p> <p>You have now on the disc a sequence RR1.BMP..., RR17.BMP.</p>
371	<i>pic2bmp</i>	[in] [out] [number]	Conversion of a sequence of FITS or PIC images to a sequence of 8-bits BMP images.
372	<i>pic2data</i>	[data file name]	Converts the values of the first line of the image in memory into a text file of name [DATA FILE NAME]. This command is particularly useful for the analysis of the spectral data starting from a spreadsheet or a program of display of curves.
373	<i>pic2data2</i>	[data file name] [first value] [step]	Converting a vector image to a two-column ASCII data file. The first value of the first column is provided as a parameter as well as the step between two successive values. This function is used to export calibrated wavelength spectra for example.
374	<i>pic2data3</i>	[data file name] [coef3] [coef2] [coef1] [lambda_ref] [x_ref] [lamda0] [pas]	
375	<i>pic2fits</i>	[in] [out] [number]	Converts a sequence of PIC images into a FITS sequence.
376	<i>plot2</i>	[data] [x-dim] [y-dim] [ymin] [ymax] [title]	Same as function <i>anim_plot</i> but applying to only one data file [DATA].
377	<i>pointoff</i>		Terminates command <i>pointon</i> .
378	<i>pointon</i>		<p>This command permits selection of any point on a sky background. The list of points will then be used by commands such as <i>poly</i> and <i>synthe</i> in order to produce synthetic sky backgrounds.</p> <p>Each time you click with the mouse, a small cross appears in the image, counter increments and the pixel coordinates and intensities are stored in memory. The maximum number of points is 5000.</p> <p>To come back to normal cursor mode type <i>pointoff</i>.</p>
379	<i>pol2rec</i>	[X] [Y] [R] [scale (deg./pixel)]	Converts a polar representation of an object to a Cartesian representation. The parameters are the same to the <i>rec2pol</i> command.
380	<i>pol2rec2</i>	[X] [Y] [R] [pos. angle] [scale (deg./pixel)]	Very similar to <i>pol2rec</i> but the added parameter [POS. ANGLE] offer the possibility to adjust the angle of origin in the rectangular representation. You can take into account the apparent orientation of rotation axis of the sun for example. The angular value is given in degrees (the default value of the <i>pol2rec</i> command is [POS. ANGLE] = -180°).
381	<i>polar</i>	[0°] [60°] [120°] [degree] [angle] [scale]	Computes the polarization angle and level from 3 images obtained through a polarizing filter at angles of 0°, 60° & 120°. The three parameters [0°], [60°], and [120°] contain the names of the three corresponding images. The two resultant images contain the linear polarization level (named with the

			<p>[DEGREE] parameter), and the polarization angle (named with the [ANGLE] parameter), respectively. The parameter [SCALE] allows normalizing the levels in the degree of polarization image. If [scale]=100, the image will contain the polarization degree in percent. The intensities in the polarization angle image correspond to degrees of polarization (between 0° and 180°). The angle origin corresponds to the filter with the 0° orientation. Angles are oriented counter-clockwise.</p> <p>Example:</p> <pre>polar P0 P60 P120 POL ANGLE 100</pre> <p>See also <a href="#">polar_carto</a>.</p>
382	<a href="#">polar_carto</a>	[degree] [angle] [step] [scale]	<p>The <a href="#">polar_carto</a> command creates a polarization map from an image containing the polarization magnitude and an image containing the polarization angle (in degrees). The output image is formed of small vectors whose length is proportional to the polarization degree and whose orientation is equal to the polarization angle. An angle of 0° corresponds to a vertical vector. The center of the vector is at the point where the calculation was done.</p> <p>[MAGNITUDE] is the polarization magnitude image name and [ANGLE] is the angle image name. The calculation of the vectors is performed at the intersection of a mesh whose step size, in pixels, is contained in the parameter [STEP]. Note that the displayed result is the average of the polarization degree and the polarization angles calculated on a zone of dimension [STEP] centered on the point. The parameter [SCALE] adjusts the length of the vectors as a function of the polarization rate. It is expressed in pixels per percent of polarization (if the polarization rate is itself in percent).</p> <p>Example:</p> <pre>polar_carto POL ANGLE 10 30</pre> <p>draws a polarization map from the images POL and ANGLE. The calculation steps size is 10 pixels, and each segment has a length of 30 pixels per polarization percentage (this requires that a polarization rate of 100% is equivalent to level 100 in the image POL.PIC).</p>
383	<a href="#">polar2</a>	[0°] [45°] [90°] [135°] [degree] [angle] [scale]	<p>Computes the polarization angle and degree from 4 images obtained through a polarizing filter at angles of 0°, 45°, 90° &amp; 135°. The four parameters [0°], [45°], [90°] and [135°] contain the names of the four corresponding images. The result consists of an image containing the polarization level (named with the [DEGREE] parameter), and an image containing the polarization angle (named with the [ANGLE] parameter). The parameter [SCALE] allows normalizing the levels in the polarization degree image. If [SCALE]=100, the image will contain the polarization degree in percent. The intensities in the polarization angle image correspond to degrees of polarization (between 0° and 180°).</p> <p>For an application, we are going to process a set of four images of the moon taken through a polarizing filter with four position angles 45° apart. This filter was placed just in front of a CCD detector. The exposure times are identical for the four images, which are called MP1.PIC, MP2.PIC,</p>

MP3.PIC, and MP4.PIC. The offset signal has been subtracted from each of these images (see the *sub* command) and they have been carefully centered to within a fraction of a pixel with respect to each other. Then:

```
Polar2 MP1 MP2 MP3 MP4 P A 100
load p
visu 5 0
```

The intensity levels in P.PIC express the polarization degree. For the moon the degree is often small as 5%. Comparing the polarization map and the albedo image is instructive. For the most part, the polarization rate is higher in the continental zones than in the seas. Nevertheless, there are some notable local exceptions. You will also note that some craters have a particularly high polarization rate.

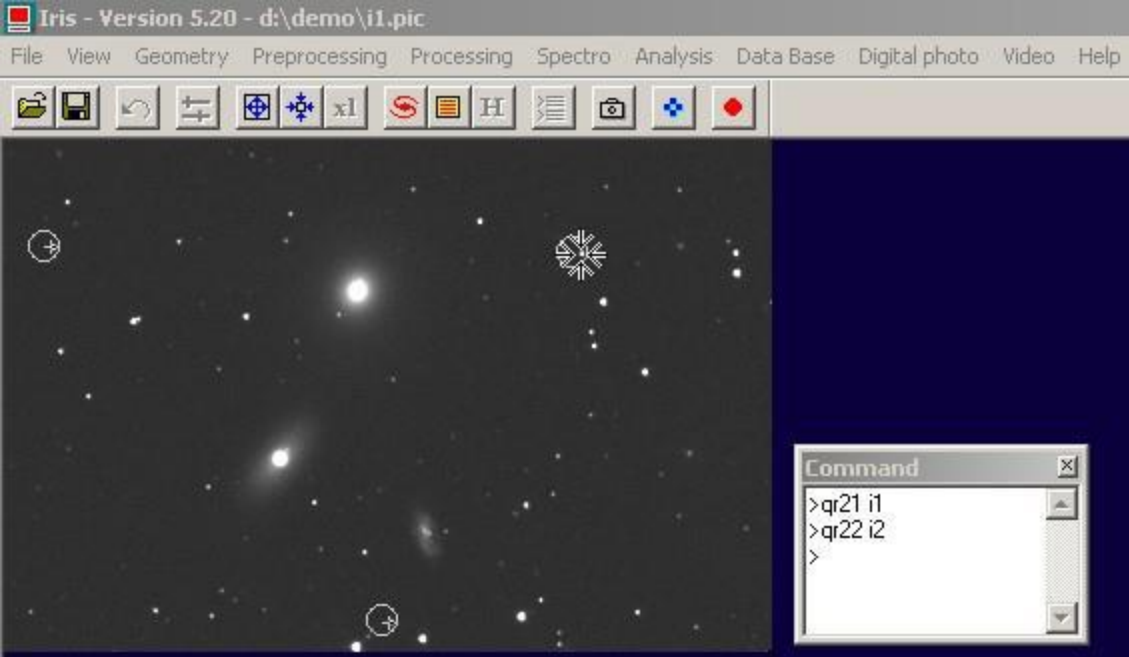



This type of polarization map provides information about the roughness of the lunar surface. The interpretation of this kind of document requires prudence (for example, the polarization rate at a given point on the moon is strongly dependent on the elevation of the sun at the point, and on the mineralogical makeup of the ground).

384	<i>polar3</i>	[0°] [45°] [90°] [degree] [angle] [scale]	Same as <i>polar2</i> but for only three angles: 0°, 45° and 90° of the polaroid analyzer.
385	<i>poly</i>	[order]	In many cases it is very important that the level of the sky background should be as uniform as possible, and with a given value (for low flux detection, photometry...). If the background is not uniform enough after preprocessing to perform such analyses, a solution may be to synthesize a background using mathematical functions that fits the real background. The synthetic background will be then subtracted from the original image.

			<p><i>Poly</i> calculates the coefficients of a polynomial by the method of least squares from data obtained by making pointings in the image background (see the command <i>pointon</i>).</p> <p>The parameter [ORDER] contains the order of the polynomial to calculate. The order can range from 0 to 5. For a fifth degree, the polynomial has the form:</p> $V = C_0 + C_1X + C_2Y + C_3 \cdot XY + C_4X^2 + C_5Y^2 + C_6 \cdot X^2Y + C_7 \cdot XY^2 + C_8X^3 + C_9Y^3 + C_{10}X^4 + C_{11}Y^4 + C_{12}X^5 + C_{13}Y^5$ <p>When a lower order polynomial is calculated, only the related coefficients are included (the others are set to zero). For example, for a second-degree polynomial, only the coefficients from <math>C_1</math> to <math>C_5</math>, as well as the constant, are significant.</p>
386	<i>power</i>	[value]	<p>Raises the intensity of the pixels of the current image to the power. The value of the power is provided by the user. Correctly works only with greyscale images.</p>
387	<i>pr</i>	[in] [dark + offset] [flat] [out] [number]	<p>The <i>pr</i> command (PR=PreProcessing) is an important command for the preprocessing of a sequence of [NUMBER] images. The [INPUT] parameter contains the generic name of the images to process. [DARK +OFFSET] is the name of the sum of the dark current image and of the offset signal. So, the dark image has either to be obtained in the same conditions as the images themselves (same exposure time, same temperature), or to be computed (see the OPT command). [FLAT] contains the flat-field image. [OUTPUT] is the generic name of the output images. See also: <i>full_pr</i></p> <p>The command <i>pr</i> produce preprocessed images that may be, for example, combined later after registration (see <i>register</i>, <i>add2</i>, <i>add3</i>, <i>drizzle</i>, <i>composit</i>, <i>smedian</i>, <i>full_pr</i> commands).</p>
388	<i>pregister</i>	[in] [out] [size] [number]	<p>Performs the registration of [NUMBER] planetary images having the generic name [IN] using an intercorrelation method. The size for the intercorrelation is given by [SIZE]. It must be a power of 2 (128, 256, 512...). The size should be larger than the planet diameter. [OUT] is the generic name of the output images. Before use <i>pregister</i> define a selection rectangle around the planetary disk (drag with the mouse). Note that the size of rectangle is not important (the rectangle mark only the center of interest of the image).</p> <p><i>pregister</i> MARS I 256 70</p> <p>To verify the efficiency of this command, you may subtract an image from another one, e.g.:</p> <p><i>load</i> MP1 MP2 MP3 MP4 P A 100  <i>sub</i> p  <i>visu</i> 3000 10000</p> <p>The observed residuals mainly come from the atmospheric turbulence.</p>
389	<i>pregister2</i>	[in] [out] [size] [number]	<p>Very similar function with <i>pregister</i> for the registration of the planetary images by the technique of the intercorellation in the Fourier domain. <i>Pregister</i> relative performs registration of each image of the sequence to the first image of this sequence. <i>Pregister2</i> on the other hand calculates the</p>

			intercorrelation of the image of row N relative with the image of row N-1. This is of an interest when the detail which is used to center the images changes in form notably (a solar protuberance for example).
390	<i>preregister</i>	[in] [out] [number]	Command for fast registering of a sequence image. The algorithm is fast (special cross correlation in the spatial domain) but less precise compared to <i>register</i> , <i>pregister</i> , or <i>cregister</i> for example ( <i>preregister</i> uses a registration at the nearest pixel for minimal degradation of image quality). <i>preregister</i> is used as a first step registration for difficult case (if traditional registration command are not applicable — situation of large de-registration for example). Apply command like <i>register</i> or <i>coregister</i> for a second pass (registration at a pixel fraction). Before run <i>preregister</i> select with the mouse a rectangle around a contrast detail. For important de-registration do not hesitate to select the whole image.
391	<i>prod</i>	[name] [coef]	Performs the multiplication pixel by pixel of the current image by the image [NAME]. The result is multiplied by [COEF]. Example: To calculate the square of the image M51.PIC with the maximum threshold of 10000:  <i>load</i> M51 <i>prod</i> M51 0.00032767 <i>visu</i> 32767 0  Value of [COEF] here was selected in such a way to remain in 0-32767 levels of Iris
392	<i>prompt</i>	[text]	Improved version of <i>prompt</i> command (batch mode, see <i>run</i> ). Now the command accepts an optional informative text. Use the character "_" for simulate space character.
393	<i>put</i>	[x] [y] [v]	Attributes the intensity [V] to the pixel at coordinates ([X], [Y]). See also: <i>get</i>
394	<i>qmosa/qm</i>	[name1] [name2] [type] (1...5)	Assembles the images [NAME1] and [NAME2] in a unique image. <i>qm</i> is optimized for stellar images: the common point between the two images is a star selected with the mouse (a simple click). <i>Qm</i> is very easy to use but the operation is rudimentary: only the relative translation between the images is considered, not the distortion for example. <i>Qm</i> is an interactive version of <i>mosa</i> command. The parameter [TYPE] defines the junction zone of the two images. <ul style="list-style-type: none"> <li>• If TYPE=1 the image 1 is on image 2,</li> <li>• If TYPE=2 a pixel in the common zone is the maximum of image 1 and 2,</li> <li>• If TYPE=3 a pixel in the common zone is the minimum of image 1 and 2,</li> <li>• If TYPE=4 the common zone is the mean intensity of image 1 and 2,</li> <li>• If TYPE=5 an interpolation is computed between images 1 and 2 for a more natural transition.</li> </ul> For the majority of situation, the type=5 option is preferred. <i>Qm</i> can process black and white images and true colors images.
395	<i>qmosa2/qm2</i>	[name1] [name2] [type] (1...5)	Same as <i>qm</i> but for non-stellar images. Iris uses the intercorellation technique around the clicked zone.

396	<a href="#">qregister/QR</a>	[name1] [name2]	Command for a quick and easy registering of deep-sky image pairs. The method used involves interactively identifying common point sources (stars) in overlapping images fields. These functions are compatible with 16-bits (black & white) and 48-bits images (true colors). [NAME1] and [NAME2] are the file name of the images to be register. The reference is the image [NAME1]. The <i>qr</i> command applied a sample translation to the image [NAME2] for superposition. For a spline version see <i>sqr</i> .
397	<a href="#">qregister2/QR2</a>	[name1] [name2]	Command for a quick and easy registering of deep-sky image pairs. The method used involves interactively identifying common point sources (stars) in overlapping images fields. These functions are compatible with 16-bits (black & white) and 48-bits images (true colors). [NAME1] and [NAME2] are the file name of the images to be register. The reference is the image [NAME1]. The <i>qr2</i> applied an affine transform to the image [NAME2] for superposition.
398	<a href="#">qregister21/QR21</a>	[name1]	QuickRegister method. Affine alternative procedure. Load a reference image (the first one of the sequence I1, I2, I3 for example) by using the command <i>qr21</i> . Iris invites you to select three stars. Next see <i>qr22</i> , <i>qr23</i> or <i>qr24</i> . For spline version see: <i>sqr2</i> .
399	<a href="#">qregister22/QR22</a>	[name2]	<p>After <i>qr21</i> run the command <i>qr22</i>. The parameter is the name of the second image of the sequence. Iris again invites you to select the same stars for an affine transform.</p>  <p>The screenshot shows the Iris software interface. The main window displays a star field with several stars. Two stars are marked with white circles and crosshairs, indicating they have been selected for registration. A snowflake-like registration marker is visible on the right side of the field. The software's menu bar includes File, View, Geometry, Preprocessing, Processing, Spectro, Analysis, Data Base, Digital photo, Video, and Help. A toolbar with various icons is located below the menu bar. In the bottom right corner, a 'Command' window is open, showing the following text:     <pre>&gt;qr21 i1 &gt;qr22 i2 &gt;</pre> </p> <p>After that save the result by: <i>save I2</i></p>

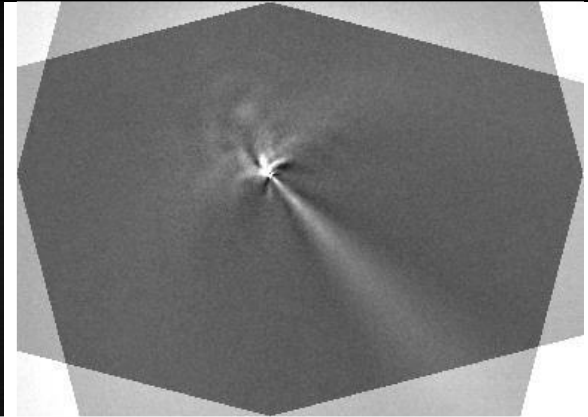
			<p>If you want to proceed further with the third image it is unnecessary to select stars in the first image again. <i>Qr21</i> command has taken in memory the star positions of the first image. So simply do:</p> <p><i>qr22</i> I3</p> <p>And so on.</p>
400	<i>qregister23/QR23</i>	[name2] [wide of the find zone]	<p>Another variation: the command <i>qr23</i> is similar to <i>qr22</i> (used after a <i>qr21</i> command), but Iris select the stars automatically. The parameters of <i>qr23</i> command are the name of the image to align and the wide of a zone where Iris find the corresponding stars. For example, after the command</p> <p><i>qr21</i> I1</p> <p>For an automatic registration of the image I2, enter:</p> <p><i>qr23</i> I2 30</p> <p>and so on for all the images of the sequence.</p>
401	<i>qregister24/QR24</i>	[in] [out] [wide of the find zone] [number]	<p>For quickly register a sequence of images with this same principle use command <i>qr24</i>.</p>
402	<i>qregister3/QR3</i>	[name1] [name2]	<p>Carries out the registration of stellar images [NAME1] and [NAME2] starting from a transformation of degree 2. It was pointed out that command <i>qr</i> applies a simple translation to superimpose the images (uses one star), whereas command <i>qr2</i> applies an affine transformation (translation, rotation, scaling and uses 3 stars). The interest of command <i>qr3</i> is to take into account some distortion of the images. In counterpart, it is necessary to work with a greater number of stars, at least 6. For example, <i>qr3</i> is adapted to register images carried out with photographic objectives lens with short focal length. <i>qr3</i> is adapted in the difficult cases of registration between images, when the geometrical transformation is not linear any more, and when the functions like <i>coregister</i> fails (difficulties for automatically matching star lists).</p> <p>The reference image is [NAME1] and Iris modifies the geometry of the image [NAME2] for which superimposes on [NAME1].</p> <p>The operation proceeds in two times. First, IRIS automatically loads in memory the image [NAME1]. You have to click with the mouse (right button of the mouse) on stars regularly distributed in the image (at least 6 and up to 50). Choose bright stars, unsaturated and insulated if possible. In order to stop the selection, click button . Select the same stars in the second image, which is also loaded automatically. At the end of the selection calculation is carried out. You must save the result manually.</p>
403	<i>r_color</i>	[red] [green] [blue] (ex: <i>r_color</i> 255 0 0=> set red)	<p>Sets the color of the reticle displayed in the video window. See also <i>r_start</i>, <i>r_stop</i>, <i>r_pos</i>, <i>r_size</i>.</p>
404	<i>r_pos</i>	[x] [y]	<p>When no input is present returns current position of reticle. When [X] and [Y] are set by user that coordinates are set to these numbers.</p>

405	<i>r_size</i>	[size]	Change the size of the reticle.
406	<i>r_start</i>	[size]	Starts to display a reticle with the set values.
407	<i>r_stop</i>	[size]	Erases the reticle from the video flow.
408	<i>radial_blur</i>	[xc] [yc] [filter strength] [method 0:spin, 1:zoom]	Applies radial blur to the image according to [METHOD] parameter; where 0 means rotational image smudging and 1 — zooming smudging.
409	<i>radial_weight</i>	[x] [y] [radius] [coefficient] [power]	Multiplies the intensity $I(r)$ of a given pixel image by a Lorentz function of the form: $I'(r) = \frac{I(r)}{1 + \left(\frac{r - r_0}{\text{coefficient}}\right)^{\text{power}}}$ where $r$ is the distance relative to a center of coordinate $(x_c, y_c)$ , $r_0$ is an offset radius, coefficient and power are adjustment parameters of the function shape. Normally, $\text{power} = 2$ .
410	<i>rainbow</i>	[name] [lambda1] [lambda2]	Useful for the representation of the spectral data. Colorize with the rainbow colors the image [NAME]. The colored distribution is realistic and takes into account which the first pixel on the left is with the wavelength [LAMBDA1] and which the last pixel on the right is with the wavelength [LAMBDA2].
411	<i>re</i>	[objet] [flat] [neon]	
412	<i>rec2map</i>	[map file] [x] [y]	The <i>rec2map</i> command is used with the <i>map</i> and <i>premap</i> commands. Starting from the parameters of a cartographic projection and the Cartesian coordinates $([x], [y])$ of a point on the map, it calculates the planetocentric coordinates of this point. Note that the point can be outside of the image (the Cartesian coordinates, for example, can be negative). The file of the projection parameters is named in the parameter [list]. For a description of the cartographic files, see the <i>map</i> command.
413	<i>rec2pol</i>	[xc] [yc] [radius] [scale (deg./pixel)]	Converts rectilinear image to its polar form. First three parameters are responsible for the center coordinates and visible radius of the image. Last parameter sets the horizontal resolution of the output image typically varying between 0.2-0.6.
414	<i>rec2pol2</i>	[X] [Y] [R] [pos. angle] [scale (deg./pixel)]	Very similar to <i>rec2pol</i> but the added parameter [POS. ANGLE] offer the possibility to adjust the angle of origin in the polar representation just as in <i>pol2rec2</i> case.
415	<i>rec2sky</i>	[x] [y]	Returns the equatorial coordinates of a point on the image whose Cartesian coordinates in pixels are in the parameters $([x], [y])$ , using the polynomial files POLX.POL and POLY.POL (created by Astrometry/Photometry dialog box). Example: <i>rec2sky</i> 12.67 321.12  Calculates the equatorial coordinates of the point with Cartesian coordinates (12.67, 321.12) from the parameters contained in the files POLX.POL and POLY.POL. See also the cartographic possibilities of IRIS ( <i>map</i> command).

416	<i>reduce_hdr1</i>	[gamma]	Compresses HDR image by applying gamma correction to the image.
417	<i>reduce_hdr2</i>	[gamma] [sharpness]	Compresses HDR image by applying gamma correction to the image along with unsharp masking.
418	<i>reduce_hdr3</i>	[gamma] [tone adjust.] [sharpness]	Compresses HDR image by applying gamma correction to the image along with color enhancement and unsharp masking.
419	<i>register</i>	[in] [out] [number]	Registers [NUMBER] images of a sequence having the generic name [IN] with respect to the first one. The generic name of the output images is [OUT]. The registration consists here in a simple translation. Draw a rectangle with the mouse around an isolated, non-saturated star. The registration of the images will be done with respect to that star. The size of the rectangle has to be large enough so that it contains the displacement of the star between two contiguous images (if this not the case, Iris could take a wrong star to perform the registration).
420	<i>reindex</i>	[input] [output] [first input index] [first output index] [number]	Reorganizes the indices of a sequence. Let us suppose a sequence: I1, I2, I3, I4. One wants to transform it into a sequence J5, J6, J7, J8. One will write: <i>reindex</i> I J 1 5 4 The input and the output sequences cannot have the same name. [NUMBER] is the number of images to be converted.
421	<i>remove@</i>		Delete all the files of the working directory starting with the character @. Equivalent with command DOS: DEL @*.*
422	<i>repairx</i>	[x]	Replaces the column of rank [X] by the average of the columns of rank [X]-1 and [X]+1. Example: <i>repairx</i> 66 Replaces the column of rank 66 by the average of columns 65 and 67. <i>repairx</i> is used for the cosmetic correction of images: suppressing electronic interference or a defective line in the detector.
423	<i>repairx2</i>	[in] [out] [x] [number]	Same as <i>repairx</i> but for an image sequence.
424	<i>repairy</i>	[y]	Replaces the line of rank [Y] by the average of the columns of rank [Y]-1 and [Y]+1.
425	<i>repairy2</i>	[in] [out] [y] [number]	Same as <i>repairy</i> but for an image sequence.
426	<i>reset</i>		Updates position of dialog box (command, threshold, ...). Useful when the screen resolution is modified.
427	<i>rfill</i>	[x0] [value]	Mask the right part of an image relative to the [X0] coordinate. See also command: <i>lfill</i>
428	<i>rgb2hsi</i>	[in_r] [in_g] [in_b] [out_h] [out_s] [out_i]	The <i>rgb2hsi</i> command converts an image defined by its Red, Green, and Blue components into an image defined by its Hue, Saturation and Intensity components. The starting element is a trichromatic image whose three components have been obtained in distinct spectral bands (not necessarily red, green, blue). The names of these images are in the parameters [R], [G], AND [B]. Starting from these three images, RGB2HSI produces three new images:


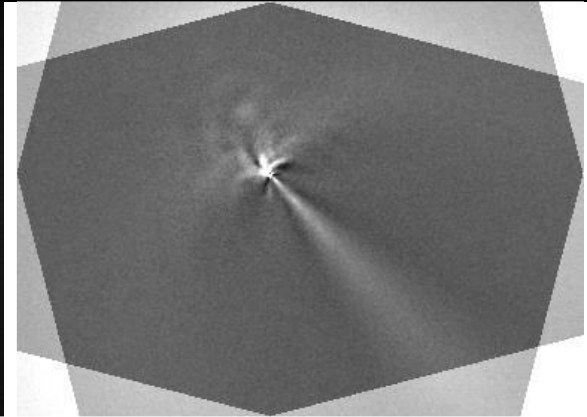
			<ul style="list-style-type: none"> <li>• An image [H] that expresses the hues of the trichromatic image in a gray scale. In this image, pixels that are predominantly red in the trichromatic image will be represented by high levels, pixels that are predominantly green will be represented by intermediate levels, and pixels that are predominantly blue will be represented by low intensity levels.</li> <li>• An image [S] that expresses the saturation of the colors in the trichromatic image in a gray scale. The areas of the trichromatic image where the colors are purest will be represented by the high levels in S.</li> <li>• An image [I] that expresses the average intensity of the three components in a gray scale. The I image is the one that most resembles each of the monochromatic components of the trichromatic image.</li> </ul> <p>See also: <a href="#">hsi2rgb</a>, <a href="#">lrgb</a>, <a href="#">rgb2pca</a>, <a href="#">pca2rgb</a>, <a href="#">trichro</a></p>
429	<a href="#">rgb2pca</a>	[r] [g] [b] [c1] [c2] [c3]	<p>The <a href="#">rgb2pca</a> command (PCA = "Principal Component Analysis") corresponds to a coordinate transformation of a trichromatic image that is represented in the space of fundamental colors (Red, Green, Blue). After the transformation, the axes are the eigenvectors of the covariance matrix of the three input images. The three resulting images are obtained by projecting the three starting axes (R,G,B) onto the three resulting axes. Without going into the mathematical details, it is interesting to choose this coordinate system because it defines three new images that are as uncorrelated from each other as possible, from the chromatic point of view. The matrix of eigenvectors, as well as the associated eigenvalues, are stored in a separate text file (pci.lst).</p> <p>The first axis, also called the principal axis, corresponds to the largest eigenvalue of the covariance matrix. Generally, this axis is very close to (but not coincident with) the "achromatic axis" (which is the axis of the [I] image in the <a href="#">rgb2hsi</a> command). This axis contains most of the intensity information and is often close to the average of the input images.</p> <p>The two other axes (ordered in decreasing eigenvalues) can thus be interpreted as linear combinations of the input images that lead to information that is not correlated to the first axis or to each other. The two corresponding images generally have much weaker dynamics, and are centered around zero. These images thus have a rather low signal to noise ratio, especially for deep sky images, and sometimes require a low pass filter (median type, for example) in order to be correctly visualized.</p> <p>The interest in this transformation is twofold:</p> <ol style="list-style-type: none"> <li>1. First, visualizing the three images in principal components allows a hierarchical classification of the information contained in the starting trichromatic image. This visualization can be done independently, or trichromatically (by putting the image of the first eigenvector in red, the second in green, and the third in blue). In this case, it is clear that the resulting image is not at all representative of the "true" colors of the image, nor is it very aesthetic, but it is the representation that gives the optimal visualization of the chromatic information in the image.</li> <li>2. Second, processing can be done in the space of principal components (primarily adaptations of the dynamics and filtering) and the results can then be brought back to the starting space (R,</li> </ol>

			<p>G, B) (with the <i>pca2rgb</i> command) to obtain a visual improvement in the original trichromatic image. It is worth noting that for this particular manipulation, the transformation is not as rigorous as the commands (<i>rgb2hsi</i>, <i>hsi2rgb</i>), and is trickier to use. It is generally reserved for images with a good signal to noise ratio (for example, planetary images or bright planetary nebulae).</p> <p>See also: <i>pca2rgb</i>, <i>rgb2hsi</i>, <i>hsi2rgb</i>, <i>lrgb</i>, <i>trichro</i>, <i>t_tr</i>.</p>
430	<i>rgbbalance</i>	[Rcoef] [Gcoef] [Bcoef]	<p>Multiplies components red, green and blue of the 48 bits in memory image respectively by the coefficient [Rcoef], [Gcoef] and [Bcoef]. This function is equivalent to RGB balances... command of <b>Digital Photo</b> menu or <b>White balance adjustment...</b> in <b>View</b> menu.</p>
431	<i>rgbbalance2</i>	[in] [out] [Rcoef] [Gcoef] [Bcoef] [number]	<p>Same function as <i>rgbbalance</i>, but applied to a sequence of images.</p>
432	<i>rgradient</i>	[xc] [yc] [dr] [alpha]	<p>Computes the rotational gradient of an image. Starting from an input image (in memory), RGRADIENT creates two images, with a radial shift ([<i>dr</i>] in pixels) and a rotational shift ([<i>dalpha</i>] in degrees) with respect to the point ([<i>x<sub>c</sub></i>], [<i>y<sub>c</sub></i>]). Between these two images, the shifts have the same amplitude, but opposite signs. The two images are then added to create the final image.</p> <p>In polar coordinates (<i>r</i>, <i>a</i>) with respect to the point ([<i>x</i>], [<i>y</i>]), we have:</p> $B'(a, r, da, dr) = 2B(a, r) - B(a - da, r - dr) - B(a + da, r + dr)$ <p>with:</p> <p><i>B</i> = the starting image  <i>B'</i> = the resulting image  <i>da</i> = the parameter [dalpha] of the command  <i>dr</i> = the parameter [dr] of the command</p> <p>The <i>rgradient</i> command can be executed also from a dialog box (<b>Processing</b> menu &gt; <b>Rotational gradient...</b>).</p> <p>The rotational gradient is used to observe poorly contrasted details in a bright object which exhibits a rotational symmetry (dust in an elliptical galaxy or jets in the tail of a comet). The function removes the rotationally symmetrical component with respect to point ([<i>x</i>], [<i>y</i>]).</p>



On the right is the original image of comet Hyakutake. On the right processed as follows:

```
load HYAKUTA
rgradient 171 164 0 15
offset 1000
visu 1700 600
```

			  <p>On the right is the original image of comet Hyakutake. On the right processed as follows:</p> <pre>load HYAKUTA rgradient 171 164 0 15 offset 1000 visu 1700 600</pre>
433	<i>ring_median</i>	[radius]	Uses a particular version of median filtering in order to eliminate from the image of large sizes. The value of the parameter [RADIUS] must be roughly the size in pixels of the objects which one wishes to erase.
434	<i>rl</i>	[#iter] [coef]	Restores an image using the Richardson-Lucy method. [#ITER] is the number of iterations to be performed (typically between 10 and 50). If [COEF] = 0, the original Lucy algorithm is applied. If [COEF] > 0, a relaxation method is used to reduce noise, but convergence is slower. Before running the command, lower the level of the sky background to a level close to zero (use the <i>offset</i> command, the <i>bg</i> command allows you to measure this level). Then choose with the mouse (small rectangle around a star) an isolated, non-saturated star, then run the command, e.g.:  <i>rl</i> 15 0  It is recommended (but not mandatory) that the image has a square size equal to a power of 2 (128, 256, 512...).
435	<i>rl2</i>	[#iter] [coef]	Modified version of the Richardson-Lucy deconvolution algorithm. The difference with command <i>rl</i> comes from the reduction of granular structure and ringing effect around the bright point-like objects. The reconstruction is effective only for pixels which have intensity higher than the computed sky background. Just like <i>rl</i> , <i>rl2</i> command use Fourier transforms and it is recommended to crop the image so that these with dimensions has a size equal to a power of 2 (128, 256, 512 pixels). Use for that <i>window3</i> command, especially designed to isolate a squared part of an image. <i>rl2</i> is used like <i>rl</i> (select with the mouse an unsaturated star before run it).

436	<i>rot</i>	[xc] [yc] [angle]	<p>Rotates the current image around the point <math>([x_c], [y_c])</math> by the angle [ANGLE]. The center of the rotation can be off the screen. Note that the coordinates of this center can be non-integer. [ANGLE] is in degrees.</p> <p>The rotation of an image is used to orient an image with respect to a reference direction (towards the north, for example). <i>Rot</i> is also used with <i>trans</i> and <i>scale</i> to register images taken with different instruments.</p> <p>In the following example, we will rotate the image M51.PIC, then go back to the initial orientation:</p> <pre>load M51 rot 100 130 22.3 rot 100 130 -22.3</pre>
437	<i>rot2</i>	[in] [out] [x] [y] [alpha] [number]	Same as <i>rot</i> but for a sequence of images. All images turn at the same angle.
438	<i>rotb</i>	[xc] [yc] [angle]	Spline version of <i>rot</i> function.
439	<i>rregister</i>	[in] [out] [size] [number]	<p>Command <i>rregister</i> carries out the registration of a sequence of images of the deep sky by taking of account an accidental field rotation. For spline version activate <i>setspline</i> function.</p> <p>The principle of the command rests on the pointing of two stars of the fields rather distant from each other but present in all the images of the sequence. The first selected star will be used to carry out registration in translation and will also be used as pivot at the time of the following phase of angular registration (the first star is the center of rotation). The position of second star pointed compared to first star is used to calculate the swing angle of field from one image to another. If possible, it is necessary that the stars are not saturated (i.e. too brilliant) to reach a maximum precision.</p> <p>The pointing of two stars is carried out by using the Select objects command of the Analysis menu. It is only after the two pointed stars operation (click with the mouse) that you can run command <i>rregister</i>.</p> <p>Command <i>rregister</i> generates two text files in the working directory: the file SHIFT.LST which contains the geometrical parameters in translation of the transformation of the images (identical to file SHIFT.LST produced by <i>register</i>) and the file ROT.LST which contains the parameters in rotation of the transformation (the position of the pivot in X and Y, then the swing angle in degree compared to the first image of the sequence).</p> <p>These files can be used by command <i>file_trans</i> and command <i>file_rot</i> (syntax identical to <i>file_trans</i>) for re-run the geometrical transformation. For that you must carry out in first <i>file_trans</i>, then <i>file_rot</i>.</p> <p>In our example, once command <i>rregister</i> carried out first, the registration of the three images can be obtained by doing something like:</p> <pre>file_trans I J 3 file_rot J J 3</pre>

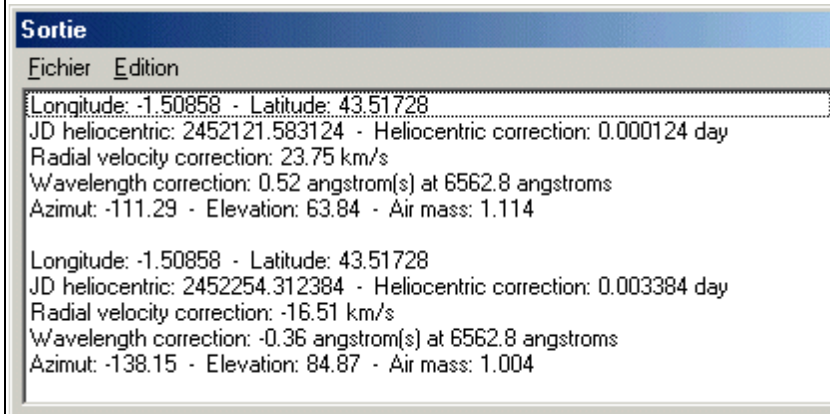
440	<i>rregister2</i>	[in] [out] [size] [number]	Slower but more precise version of <i>rregister</i> . In real images difference is virtually non-existent.
441	<i>rsurvey</i>	[alpha] [delta] [focal] [pixel]	
442	<i>run</i>	[batch file] [param1] [param2] ... [param5]	<p>Version 5.00 implements the possibility to run a succession of commands from a script file in text format. This script file (or batch file) must mandatory have the extension "pgm", for example "myprogram.pgm".</p> <p>Scripts are run with the RUN command from the console.</p> <p>The "batch" mode is very elementary. The execution is strictly linear. It is not possible to carry out conditional tests or loops for examples. The script is thus not a true program, but the process can be of a certain help for repetitive tasks.</p> <p>The minimal parameter of RUN command is the name of the batch file (no extension). If no path precedes the file name, Iris searches the file into the working directory.</p> <p>The name of the file can be followed by optional parameters which are provided to the script file. It is possible to pass up to 5 parameters by this method. In the script file the first argument is identified by the item "\$1", the second parameter by the item "\$2", and so on.</p> <p>For example, a script file which produces an effect of gradient ("bas-relief") in an image with a variable force can be written</p> <pre>load \$1 trans \$2 \$3 save tmp load \$1 sub tmp 0 visu 200 -200</pre> <p>Edit the file with word processing software, and save the result under the name test.pgm in the working directory.</p> <p>From the console, enter the command</p> <pre><i>run</i> TEST M51 1 1</pre> <p>The script which is executed is then equivalent to the typing:</p> <pre>load m51 trans 1 1 save tmp load m51 sub tmp 0 visu 200 -200</pre> <p>You can easily start again the batch by positioning cursor prompt on the command line and by modifying one or more parameters. For example:</p>

			<p><i>run</i> TEST M51 2 0</p> <p>The hundreds of Iris command can be used in a script "pgm" file:</p> <pre><i>load</i> \$1 <i>scale</i> 3 \$2 \$2 <i>mirrorxy</i> ...</pre> <p>Some commands require defining an area on the current displayed images. Just before this command, add in the script file the command <i>prompt</i>. <i>Prompt</i> freezes the execution of the batch file and open a "Prompt" dialog box. Select the area of the image, then, click the "OK" button of prompt dialog box. For example:</p> <pre><i>load</i> m51 <i>prompt</i> <i>window3</i> 200</pre> <p>In this example, during prompt, define a rectangle with the mouse. Iris crops a 200x200 pixels zone of the images M51 centered on the selected area.</p> <p><i>prompt</i> command is also useful for step-to-step run of the script.</p> <p>You can also add comment into the script file. If the first word of a sentence is not a valid command, the corresponding line is a comment line. For example:</p> <pre>===== My program load Messier 51 ===== load m51 End of my program</pre>
443	<i>rvc</i>	[day] [month] [year] [alpha (12h5m12s)] [delta (-5d32m39s)]	<p>For precise spectral calibration it is important to understand that the measured radial velocity of a star represents the relative motion of the star with respect to the observer at the time of the observation. Because the observer moves due to the Earth's rotation and revolution around the Sun, these motions have to be eliminated from the stellar radial velocity. This reduction step is called the heliocentric correction. For this, the command <i>rvc</i> of IRIS return the radial velocity of the observer relative to the observed object and the wavelength correction needed to correct the observed spectra (Doppler effect for the wavelength 6562.8 angstroms). The max correction for the Earth motion is roughly 30 km/s. The <i>rvc</i> command returns also the Heliocentric Julian Day. It is the Julian Day adjusted at the center of the Sun and depends on the standard Julian day of the observation and the celestial coordinates of the object observed. <i>Rvc</i> takes into account the light travel time for an event coming from a particular location on the sky to be observed at the center of the Sun. So the Heliocentric Julian Day removes the</p>

periodic parallactic time shift due to where the Earth was in its orbit when an event was observed. The correction can amount to as much as little over 16 minutes for observations taken 6 months apart. For example here Iris would return:

*rv* 31.083 07 2001 1h09.5m 47d14'

*rv* 10.809 12 2001 1h09.5m 47d14'



For the observation 1 the wavelength correction (for Ha line) is of 0.52 angstroms (spectral lines shifted towards the blue). For the observation 2 the correction is of -0.36 angstroms (spectral lines shifted towards the red).

The true wavelength  $\lambda$  is derived from the observed wavelength  $\lambda'$  from the relation (RV = radial velocity in km/sec):

$$\lambda = \lambda' \left( 1 + \frac{RV}{299792.458} \right)$$

444	<i>save</i>	[name]	Save the image in memory in the current directory (defined in the item Current Directory from the Settings tab). You can also indicate in which directory to save a particular image by specifying the full path of the image. For instance: <i>save</i> c:\nuit7\m51
445	<i>save_trichro</i>	[R] [G] [B]	Saves the colors layers of the current displayed true color image into three distinct file.
446	<i>savebmp</i>	[name]	Saves the image in memory in the current directory under the form of a bitmap file. The color palette used to create the file is the active palette. The BMP image is a 24-bit if a tri-color is loaded.
447	<i>savejpg</i>	[name] [quality]	Saves current image into a JPEG file. You have the possibility to adjust the compression factor. A value 1 for quality parameter offers best fidelity. A quality of 5 corresponds to the maximum compression ratio. It is also possible to produce a JPEG image from the <b>Save...</b> function of <b>File</b> menu. (quality = 2).
448	<i>savepng</i>	[name]	Saves current image into a 25 bit PNG file.

449	<i>saveppm</i>	[name]	Stores current image in the form of a PPM file (Portable Pixel Map - 24-bits format). You can save true colors images (24-bits) but also, black and white images (16-bits)
450	<i>savepsd</i>	[name]	Saves the in-memory image into a Photoshop PSD 48-bits file (16-bits by color plane). Note: The image must not contain any negative pixel values for this function to work correctly. See also commands: <i>clipmin</i> , <i>offset</i> , <i>mult</i> , and <i>savepsd2</i> .
451	<i>savepsd2</i>	[name]	Same as <i>savepsd</i> , but the natural internal coding of Iris [-32768..32767] is mapped into [0..65535].
452	<i>savetiff</i>	[name]	Saves the in-memory image into a TIFF 48-bits file (16-bits by color plane). Note: The image must not contain any negative pixel values for this function to work correctly.
453	<i>sblur</i>	[sigma] [gamma]	Command <i>sblur</i> (for Selective Blur i) generates a blur in the image with a force more higher since the objects are more intense. This function is used on star fields to produce an Akira Fujii effect, for example to reveal the contour of the constellations. The parameter SIGMA permit to adjust the degree of the blur (select values between 2 and 15) and the parameter GAMMA makes adjust the sensitivity to the brightness of stars (the characteristic value is between 4 and 10). For an optimal effect it should be taken care that the most brilliant stars are not saturated. This command boosts the colors and can be also used in combination with <i>asinh</i> .
454	<i>scale</i>	[option] [xf] [yf]	The <i>scale</i> command can be used to enlarge or reduce the current image. Each axis can have a different scale factor. The parameters are: - Type of interpolation: <ul style="list-style-type: none"> <li>- [option] = 1, for an enlargement, interpolation by pixel duplication; for a reduction, interpolation by undersampling.</li> <li>- [option] = 2, bilinear interpolation.</li> <li>- [option] = 3, spline interpolation (only for enlargements).</li> </ul> - [XF] = scale factor along the X axis - [YF] = scale factor along the Y axis. The <i>scale</i> command can be used to detail regions of an image by enlarging it. Some types of processing, like restoration, photometry, and astrometry, become more precise when the image is oversampled. You can artificially oversample with the <i>scale</i> command. Reducing images is also useful when they take up too much space, or to build a library of quick-look images (where only an approximate look is important). To see the influence of the different types of interpolation, execute the following commands: <i>load</i> M51 <i>scale</i> 1 3 3 Interpolation by pixel duplication conserves the sharpness of the image, but of course the result has an artificial look.

			<p><i>load</i> M51 <i>scale</i> 2 3 3</p> <p>Bilinear interpolation gives the image a smoother look than simple pixel duplication. Nevertheless, for large enlargement factors, some artificial geometric structures can appear around stars. The spline interpolation reduces these effects.</p> <p><i>load</i> M51 <i>scale</i> 3 3 3</p>
455	<i>scale_hr</i> <i>scalehr</i>	[option] [xf] [yf]	Similar to <i>sscale</i> but works with true color images. Correct description is unknown.
456	<i>scale2</i>	[in] [out] [option] [xf] [yf] [number]	Same function that <i>scale</i> but applies to a sequence of images.
457	<i>scalecolor</i>	[in] [out] [reference index] [number]	<p>Scales the input images sequence [IN] for color combining (or drizzling algorithm). For each image, a rescaled sky was subtracted and a gain adjusted relative to a reference image of the input sequence. The index of the reference image is [REFERENCE INDEX]. The generic name of the output sequence is [OUTPUT].</p> <p>Before <i>scalecolor</i> command select an unsaturated star with the mouse (the images are to be registered before applying <i>scalecolor</i>, see <i>register</i> or <i>coregister</i> for example). Ideally, for a good color balancing, select a solar type star (a G star).</p> <p>Example:</p> <p><i>scalecolor</i> I J 2 3</p> <p>Scale the images I1, I2, I3 relative to the image I2. This produces the scaled sequence J1, J2, J3. If J1 is the R component of a tricolor image, if J2 is the G component and J3 is the B component, then you can made:</p> <p><i>Trichro</i> J1 J2 J3</p>
458	<i>scalecolor2</i>	[R] [G] [B]	Adjust the respective levels of the images [R], [G], [B] so that their intensities are identical on average in a zone of the image defined as a preliminary with the mouse. More precisely, Iris multiplies the images [G] and [B] by a distinct coefficient and also adds constants to them so that the sky is really black. This command is very practical to carry out the chromatic balance of the planetary images (click here for an example). For the stellar images it is necessary to employ command <i>scalecolor</i> .
459	<i>scalecolor3</i>	[R] [G] [B]	
460	<i>scan</i>	[x1] [x2] [integration time] [line number]	Acquisition in scan mode (TDI) using an Audine camera. The first use of this command is the imagery in drift-scan mode: the telescope is stopped, the sky move on the surface of the CCD, but at the same time, the detector is read lines after lines with a speed which is synchronous with the diurnal

			<p>movement. If the lines of the CCD are correctly aligned compared to the celestial equator and if the reading rate is correct, one obtains steady image of the sky.</p> <p>The parameters [X1] and [X2] define the horizontal coordinates in pixels of the band effectively digitized during the scan. Indeed, on account of the effectiveness, it is not essential to acquire all the width of the CCD if one is interested only in one star or has a tight couple: one limits the occupation of the memory and it is the guarantee to obtain a very long tape.</p> <p>The time of integration is the duration in second which is passed between the acquisitions of two consecutive lines of the CCD. Here the exposure time is 0.13 second. It is selected by successive step tests and by examining the intensity level of trail (command <i>stat</i> for example). The detector should not especially be saturated, i.e. with Audine not to exceed the level of 32767 ADU.</p> <p>Finally [LINE NUMBER] is the number of lines in the final image once the completed scan. In the example one asked for 7000 lines. The time of observation of star is thus of 7000 X 0.13 = 700 seconds (15 minutes and 10 seconds).</p>
461	<i>scan_calib</i>	[name] [number]	<p>PC time synchronization method. Details are here:  <a href="http://www.astrosurf.com/buil/iris/new380/new380_us.htm">http://www.astrosurf.com/buil/iris/new380/new380_us.htm</a></p>
462	<i>scan2pic</i>	[name] [x0] [number]	<p>Selects the column of coordinate [X0] in the first image of a sequence of [NUMBER] images of generic name [NAME]. This column becomes the first column of a new image which is built in the memory of the computer. The column of coordinate [X0] in the second image of the sequence becomes the second column of the image in memory, and so on for all the images of the sequence. Finally, the image in memory, and which is displayed after the processing, has a horizontal format equal to the number of images of the sequence and a vertical format equal to the number of pixels along the axis Y in the images of the sequence.</p> <p>A use of <i>scan2pic</i> is the synthesis of a monochromatic image of the Sun while scanning its disc on the entrance slit of the spectrograph and by making a regular acquisition of images simultaneously. These images become the input sequence of <i>scan2pic</i>. The coordinate [X0] is for example the core of a spectral line (the H-alpha line in the red for example) or the close continuum if one wants an image standard photosphere.</p> <p>It is necessary, if possible, to adjust the frequency of acquisition of the images with the scanning speed if one wants that the scale of the image is the same following axes X and Y. The final improvement of a different scale along the axes is possible with <i>scale</i> command which acts independently as X and Y. Command <i>win_webcam</i> is useful to isolate the strictly necessary of the image during acquisition (some pixels on the right and on the left of a spectral line for example).</p>
463	<i>scan2pic2</i>	[name] [x0] [number]	<p>Differences between <i>scan2pic</i> are nowhere to be seen... Not even here:  <a href="http://www.astrosurf.com/rondi/obs/shg/spectroheliogramme.htm">http://www.astrosurf.com/rondi/obs/shg/spectroheliogramme.htm</a></p>
464	<i>scan2pic3</i>	[name] [x0] [nb col] [images number]	<p><i>Scan2pic3</i> command (a spectroheliogram construction family command - see <a href="#">Lhires 3</a> spectrograph documentation).</p>

465	<a href="#">select</a>	[in] [out]	Creates new sequence of images based on numeration in the SELECT.LST file obtained for example with the <a href="#">bestof</a> and <a href="#">bestof2</a> commands or you can fill it manually. The [ENTER] parameter contains the generic name of the starting series. The [OUTPUT] parameter contains the generic name of the re-indexed series.
466	<a href="#">set_date</a>	[date]	Modify the date of acquisition of an image. Example:  <a href="#">Set_date</a> 16/09/2000
467	<a href="#">set_fits</a>	[bscale] [bzero]	If necessary defines manually the parameters BSCALE and BZERO before loading a FITS image (for example for fit the dynamics to the authorized range for Iris [-32768..32767] for difficult case like no full compatible FITS header).
468	<a href="#">set_hour</a>	[hour]	Modifies the time of acquisition of an image. Example:  <a href="#">Set_hour</a> 20:05:45
469	<a href="#">setbase</a>	[base (0 or 1)]	Defines the first index for an input sequence for the commands <a href="#">convertbmp</a> , <a href="#">convertbmp24</a> , <a href="#">convertbmp24bw</a> , <a href="#">converttiff</a> , <a href="#">converttiff24</a> and <a href="#">convert24bw</a> . The default value is 1. The over typical value is 0. For this enter the command:  <a href="#">setbase</a> 0
470	<a href="#">setfindstar</a>	[sigma]	Defines the threshold above the noise for stars detection with <a href="#">findstar</a> and <a href="#">coregister</a> commands. The RMS noise in the background image is determined, then the threshold is: $[SIGMA] \times (RMS\ noise)$ . The default value for [SIGMA] is 7.
471	<a href="#">setmatch</a>	[method (1 or 2)]	IRIS has two algorithms to find correspondences between two star lists. This type of algorithm is in the heart of astrometric functions and registration commands like <a href="#">coregister</a> . Using  <a href="#">setmatch</a> 2  Selects the new improved algorithm (see <a href="#">new improved algorithm</a> for details) Using  <a href="#">setmatch</a> 1  Selects to original algorithm. This is the default setting.
472	<a href="#">setnbstar</a>	[number]	Sets the number of the brightest object used during the matching process of command like <a href="#">coregister</a> , full Deep-sky registration or astrometry. The default value is 30 objects, but some time it is necessary to increase this number (max value is 200) if the matching fails. If you run the command without argument, the actual default value is returned. Previously used value is conserved even if you restart the software.

473	<i>setregister</i>	[polynom order]	<p>Defines the order of the polynomial for registration with the <i>coregister</i> command. The default is a polynomial of degree one. If it is necessary to correct distortion between images you can increase this value. The max value is 5. Example:</p> <p><i>setregister 2</i></p> <p>set the polynomial to the order 2.</p>
474	<i>setspline</i>	[mode]	<p>If you carry out command <i>setspline 0</i> all operations of registrations run since that line of commands use the bilinear interpolation to calculate the registered images. If you carry out command <i>setspline 1</i> same the operations use the spline interpolation method. The advantage of the spine interpolation is that it doesn't smooth images like other algorithms and thus details are better preserved. On the other hand, the calculating time is longer. The commands concerned are for example <i>register</i>, <i>pregister</i>, <i>cregister</i>, <i>rregister</i>, <i>rregister2</i>, <i>coregister</i>, <i>coregister2</i>, For the simple translation or the simple rotation of the image in memory you can use commands <i>strans</i> and <i>srot</i> to profit from the spline interpolation spline (symmetrical functions the <i>trans</i> one and <i>rot</i> which use they a bilinear interpolation). The profit of the spline mode is particularly visible in deep sky imagery. The interpolation spline was already present in the preceding version of IRIS, but only for the accessible commands since the menus (for example the command <b>Registration of stellar images... of Processing menu</b>).</p>
475	<i>setsubsky</i>	[sigma] [polynomial degree]	<p>Defines the polynomial order for the fit of the sky with the <i>setsubsky</i> command and the threshold above the noise for background detection. The default values are standard for the majority of situations [(SIGMA)=5, [POLYNOMIAL DEGREE]=3). Example:</p> <p><i>setsubsky 8 3</i></p> <p>sets the fit order to three and the sigma coefficient to 8.</p> <p>Very useful function when the remaining gradient from the light pollution is still visible on the image after the flat field correction like in image of comet C/2004 Q2 (Machholz) obtained on January 11, 2005 with a camera Canon EOS 20D and a telephoto lens Canon of 400 mm at f/5.6 below:</p>



Three examples are shown next:



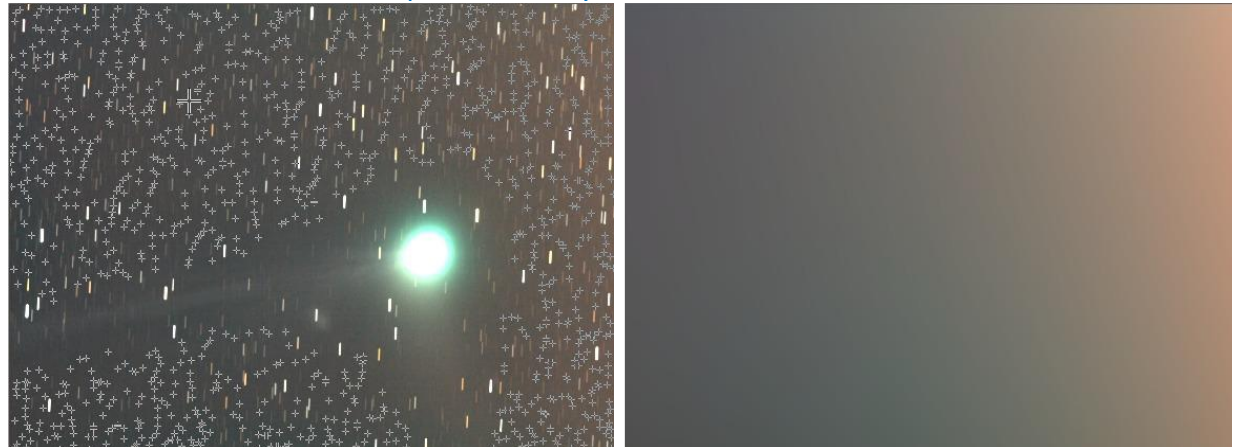
Image on the left is obtained with the couple of commands:

*setsubsky* 0.8 3  
*subsky*

Image on the right is obtained with following procedure:

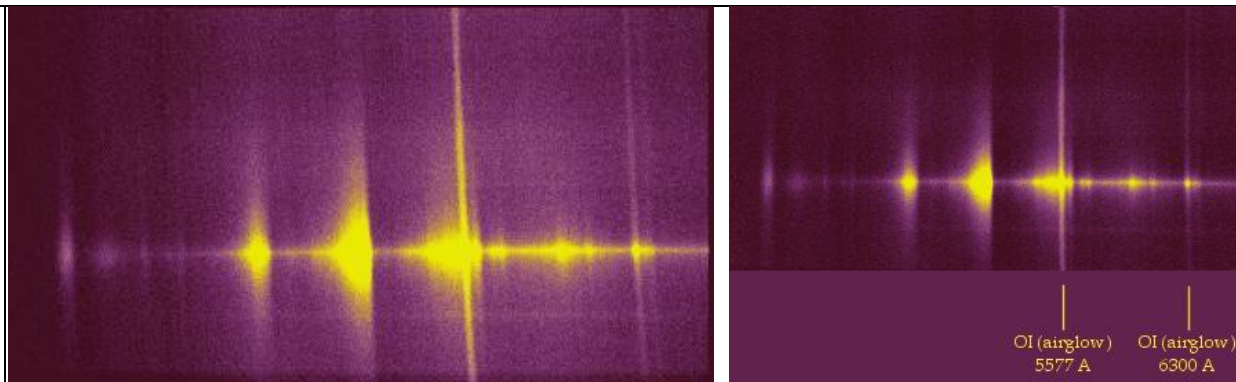
*pointon*  
*poly* 3  
*synthe*  
*pointoff*  
*save* SKY  
*load* COMET  
*sub* SKY 500

Where the results after commands *pointon* and *synthe* are shown below.

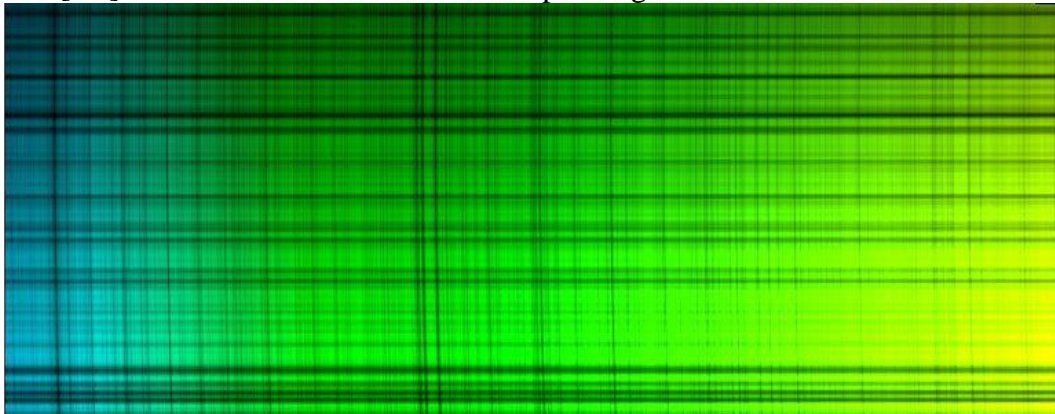


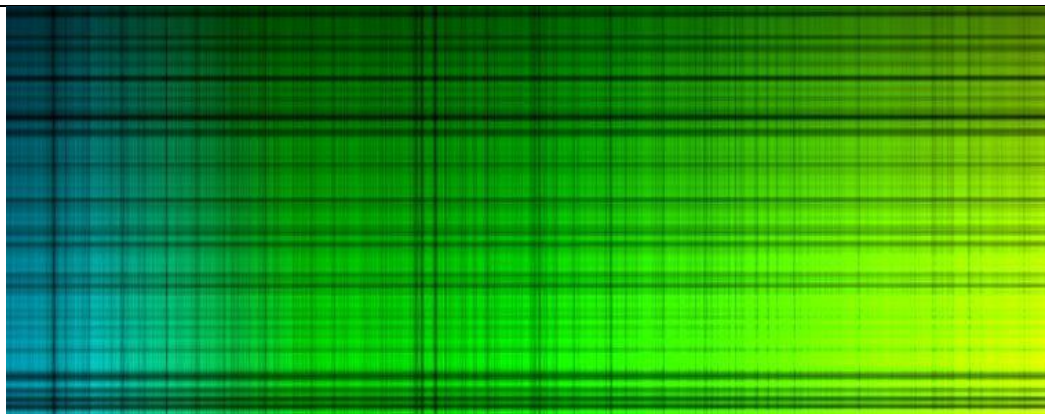
Points here were selected manually.

476	<i>shutter</i>	[val]	
477	<i>sigma_rejection</i>	[sigma]	
478	<i>signed</i>		Convert 16 bits unsigned in-image to a 16 bits signed image. Useful for some imported images. See also: <i>loadsx</i> , <i>convertsx</i> .
479	<i>sky2rec</i>	[ra] [dec]	Starting from the polynomial files POLX.POL and POLY.POL (created by Astrometry/Photometry dialog box), <i>sky2rec</i> returns the Cartesian coordinates of a point on the image whose equatorial coordinates are in the parameters ([ <i>ALPHA</i> ], [ <i>DELTA</i> ]). Example: <i>sky2rec</i> 8H34M20.3s -05d12'34"
480	<i>slant</i>	[y0] [alpha]	Rectification of tilted stellar spectra.



481	<i>slant2</i>	[input] [output] [y0] [alpha] [number]	Equivalent to <i>slant</i> , but for an images sequence.
482	<i>smax</i>	[name] [number]	For a given pixel in the output image, <i>smax</i> calculates the maximal of the intensities of the corresponding pixels in a set of images whose generic name is in the parameter [NAME]. The number of images in the stack is [NUMBER].
483	<i>smedian</i>	[name] [number]	<p>For a given pixel in the output image, <i>smedian</i> calculates the median of the intensities of the corresponding pixels in a set of images whose generic name is in the parameter [NAME]. The pixel in the output image is set to the value of the median intensity.</p> <p>The generic name is the root of the name of an image which is completed by a number and an extension. The first number added is 1 and the last is [NUMBER]. Thus, with the generic name "image" and [number]=5, the processing will act on the images:</p> <p>IMAGE1.PIC IMAGE2.PIC IMAGE3.PIC IMAGE4.PIC IMAGE5.PIC</p> <p>Recall the definition of the median. Start with the set of numbers:</p> <p>5, 9, 1, 0, 3</p> <p>Arranging these numbers in an increasing order gives:</p> <p>0, 1, 3, 5, 9</p> <p>The median value is the number 3.</p> <p>At the end of the calculations, <i>smedian</i> provides the percentage of the contribution from each input image to the final median image.</p> <p>The maximum number of images is 19 (see also: <i>smedian2</i>).</p>

			<p>The principal application of <i>smedian</i> is the calculation of a flat-field from a sequence of images containing stars. For the flat-field to be correct, the field of view should be different for each image in the sequence. This way, the median flat-field will not have stars; since there is little chance that a star would be on the same pixel on all the images in the sequence if the number of images is relatively high (greater than or equal to 5).</p> <p>On the other hand, <i>smedian</i> can be used on a set of centered deep sky images. In this case, you obtain an image with the same density as an individual image, but on which most of the artifacts have been eliminated (electronic interference, cosmic rays, satellite tracks, etc.).</p> <p>Before calculating the median image, it is important for each image in the set to have the same signal level (see <i>ngain2</i> and <i>noffset2</i> commands).</p> <p>It is good to compare this image with one of the individual images to see that in the flat-field image the stars have totally disappeared and the noise is noticeably attenuated.</p> <p>It is important to study the contribution of each input image to the median image. Ideally, the percentage for each image would be identical. In this case, since we have used 5 images, the percentage for each one should be about 20%. Any significant difference is a sign of an anomaly. This is one way to find that an image is not homogeneous with respect to the others (bad offset, for example).</p>
484	<i>smedian2</i>	[name] [number]	Same command as <i>smedian</i> , slower but unlimited as far as the number of images is concerned.
485	<i>smile</i>	[y0] [radius]	<p>Changes the curve of the spectral lines to compensate a distortion: optical defect of smile type, a traditional problem in spectrograph. The parameter [RADIUS] is the radius of curvature of the spectral lines. [Y0] is the vertical coordinates corresponding to the vertex of the curve.</p>  <p>The result of the command (radius of -7000 pixels find by successive try): smile 250 7000</p>



486	<i>smile2</i>	[ini] [out] [y0] [radius] [number]		Same command that <i>smile</i> but applying to a sequence of image. This makes it possible to correct the distortion of the spectral lines in order to make them quite right, which is in particular significant in a command like <i>scan2pic</i> during the synthesis of a spectroheliogramme of the Sun.
487	<i>smin</i>	[name] [number]		<i>Smin</i> calculates the minimal intensities of a stack of [number] images of generic name [name]. See here.
488	<i>split_cfa</i>	[c1] [c2] [c3] [c4]		This function concerns only RAW images of digital camera (DSLR). The command splits the CFA structure into four distinct files (one for each of the colors/positions in the periodic Bayer matrix). One image contains the intensity of red pixels, two images contain the intensity of green pixels, and one image contains the intensity of blue pixels. These four images can be processed individually, and then recombined in a new single image CFA with <i>merge_cfa</i> command.
489	<i>split_cfa2</i>	[in] [a] [b] [c] [d] [number]		Separates the pixels R, G and B of a sequence of images CFA (Bayer matrix pixels organization) in 4 distinct images (an image for the red pixels, an image for the blue pixels, two images for the green pixels).
490	<i>split_rgb</i>	[r] [g] [b]		Transforms a 48-bit image (true colors) into these R, G and B components in the form of separate image files (equivalent to RGB Separation ... from the Digital Photo menu).
491	<i>split_rgb2</i>	[in] [r] [g] [b] [number]		Same as <i>split_rgb</i> but for sequences.
492	<i>SQR/sqregister</i>			Spline version of <i>qregister/QR</i>
493	<i>sqr2</i>	[name1] [name2]		Spline version of <i>qregister2/QR2</i> .
494	<i>srot</i>	[xc] [yc] [angle]		Spline version of <i>rot</i> function.
495	<i>sscale</i>	[fx] [fy]		Acts very similar to function <i>scale_hr</i> with the interpolation option either 2 (bilinear) or 4 (bicubic). But not exactly identical... Works properly only with greyscale images.
496	<i>stat</i>			Returns global statistic of the image.
497	<i>stat2</i>	[x1] [y1] [x2] [y2]		Calculates the local statistics of an image. The coordinates of the calculation area are delimited by $(x_1, y_1) - (x_2, y_2)$ .

498	<i>stat3</i>	[in] [number]	Calculates global statistics of a sequence of images. The result is a text file named "STATS.LST" in the working directory with 6 columns:  Column 1: name of the image. Column 2: mean intensity. Column 3: maximum intensity. Column 4: minimal intensity. Column 5: standard deviation. Column 6: median intensity.
499	<i>stat4</i>	[in] [x1] [y1] [x2] [y2] [number]	Same function as <i>stat3</i> , but applying to a part of the images.
500	<i>strans</i>	[dx] [dy]	Command similar to <i>trans</i> but using the spline interpolation instead of the bilinear interpolation. <i>strans</i> makes it possible to obtain images having a factor of smoothing less significant than <i>trans</i> , which preserves resolution. Correctly works only with the greyscale images. For the true color ones use <i>transb</i> function.
501	<i>sub/soust</i>	[name] [offset]	Subtracts from the image in memory the image on disk designated by [NAME]. The constant value [OFFSET] is added to the result.
502	<i>sub2/soust2</i>	[in] [operand] [out] [offset] [number]	Subtracts from the [NUMBER] images of a sequence of images with the generic name [IN] the image [OPERAND] then adds the value [OFFSET] to these images (see the command <i>sub</i> ). [OUT] is the generic name of the output images.
503	<i>subgradient</i>	[mask name]	Equivalent to <i>subsky</i> but using the possibility of masking (see the procedure described in <i>setsubsky</i> function for removing a gradient). The <i>setsubsky</i> command can be used for the adjustment (see for example the command <b>Remove gradient (polynomial fit)...</b> of the <b>Processing</b> menu. The high, medium and low options for the sky background detection correspond respectively to values of "Sigma" equal to 6.0, 3.0, 1.5. My high, medium and low options for the adjustment are equal to 7, 3, 2 (degree of the adjustment polynomial. For example, for maximum sky background detection and intermediate background adjustment before launching <i>subgradient</i> the command <i>setsubsky</i> 6 will be written. All the intermediate values are of course possible.
504	<i>SUBSKY</i>		Subtracts synthesized background using the parameters set in <i>setsubsky</i> function.
505	<i>SUBSKY2</i>		Same command as <i>subsky</i> but applied to a series of images
506	<i>SUBSKY3</i>	[SIZE] [MASK NAME]	Subtracts synthesized background using another method using a local estimator of the sky background. A mask is necessary for identify bright objects and large nebulae because it is not easy for the software to determine areas containing only background. The function is also accessible from <b>Processing menu</b> and command <b>Remove gradient (local estimator)</b> . Mask can be obtained by using the function <i>bin_down</i> .
507	<i>subsky_ring</i>	[radius1] [radius2]	

508	<i>substitute</i>	[name1] [name2] [delta]	<p>Performs the pixel by pixel comparison of the images [NAME1] and [NAME2], and for each of these pixels:</p> <p>if <math>ABS([NAME1] - [NAME2]) &gt; [DELTA]</math> then</p> <ul style="list-style-type: none"> <li>• result = [NAME2]</li> </ul> <p>otherwise</p> <ul style="list-style-type: none"> <li>• result = [name1]</li> </ul> <p>(ABS signifies absolute value).</p> <p>The <i>substitute</i> command is used in conjunction with the image modeling commands (i.e. <b>Fit ellipses</b> command from the <b>Processing</b> menu) in order to suppress the stars in the original image. This creates a new image that is better analyzed after a second pass.</p> <p>There are three steps:</p> <ul style="list-style-type: none"> <li>• Build a rough model of the original image;</li> <li>• Apply <i>substitute</i> in order to produce an image that is close to the original, but where the zones that are difficult to model (stars, etc.) are replaced by their equivalent in the calculated model;</li> <li>• Calculate a second model from the image determined in the preceding step. This model is generally more satisfactory than the first.</li> </ul>
509	<i>sv0</i>	[file]	<p>The command <i>sv0</i> is a subset of the command <i>sv1</i> that insulates preprocessing section of the latter.<i>sv0</i> can be useful to carry out some checks, analyze the operating principle of command set <i>sv1</i>, <i>sv2</i>, <i>sv3</i> and allows the possibility of process your DSLR with a high degree of automation. Instructions concerning set of these commands can be found here.</p> <p><a href="http://astrosurf.com/buil/iris/new552/survey_us.pdf">http://astrosurf.com/buil/iris/new552/survey_us.pdf</a></p>
510	<i>sv1</i>	[file]	<p>Runs a script file (.lst extension) for develop sequence of RAW DSRL images, compute full preprocessing, find the sky background, select reference stars on the images, match with a reference catalogue, correct optical, and finally obtain plate solution via least square fit.</p>
511	<i>sv2</i>	[file]	<p>Runs a script file (.lst extension) for project sequence of astrometric reduced images on a selected cartographic map, correct difference on the magnitude constant between the images, and finally the images are merged on a common mosaic.</p>
512	<i>sv3</i>	[file]	<p>Runs a script file (.lst extension). Makes it possible to draw points, lines or of the circles in the cartographic image. The positions of these points, lines and circles are defined in equatorial coordinates (or galactic coordinates). Command <i>sv3</i> can be used to delimit fields, to trace the limits of the constellations, or to draw the trajectory of a mobile object directly in the image (comet, asteroid,</p>
513	<i>sym</i>	[xc] [yc]	<p>Copies a part of an image into another part of this image symmetrically with respect to the point ([cx], [cy]). The window to copy has to be drawn first with the mouse.</p> <p>Symmetry with respect to a point is used to suppress an undesired object in an image. The area containing the object is replaced by a similar area taken from the same image. The duplicated area should not, of course, contain undesirable objects itself. For a deep sky image, make sure that the sky</p>

background has the same value in the two zones. The *sym* command can be used to erase certain defects in an image (cosmic rays, etc.). It is often used in conjunction with the **fit ellipses** command to suppress the bright stars in the object to be modeled.

For example, let's suppress the star in front of the elliptical galaxy UGC 4170 (see here a tutorial) at the coordinates (50,91). Using the cursor, we find that the center of the galaxy is, to within a pixel, (66, 103). The process is the following:

```
load U4170  
visu 560 370
```

Drag the star

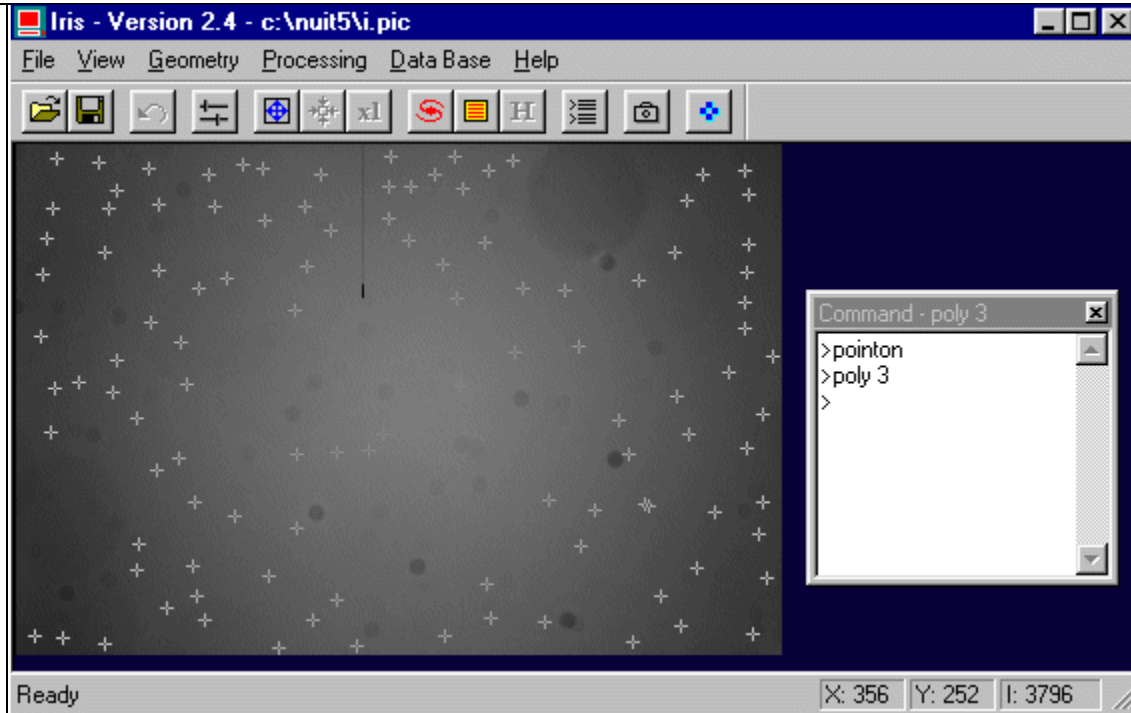
```
sym 66 103  
visu 560 370
```

514 *SYNTHE*

To calculate the synthetic sky, it is necessary to run command *synthe* (it does not have parameters). Example of the use can be seen here:

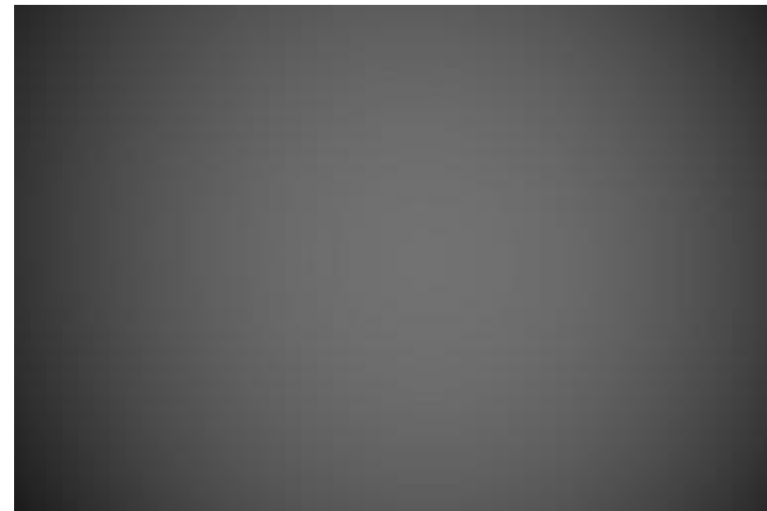


The figure shows a flat-field image with dust and dead column present. Enter the command *pointon* then, click on the image (mouse left button)

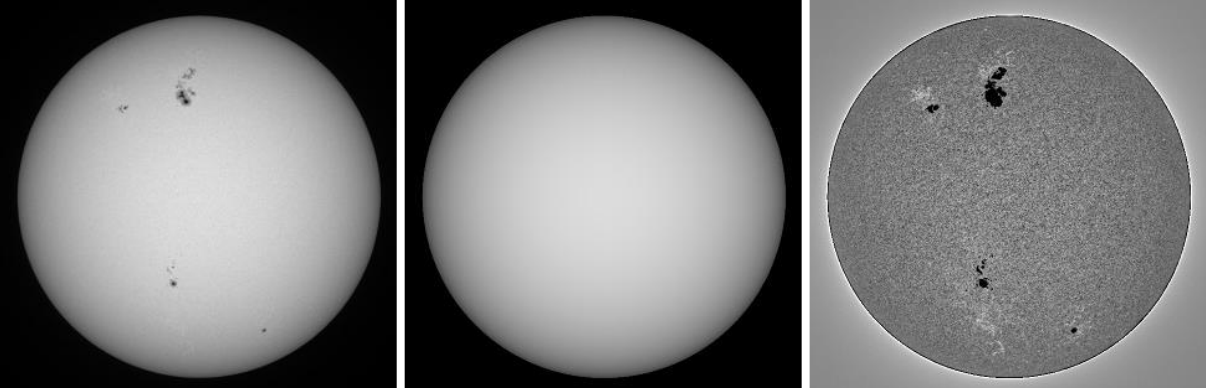


Define many points in the image (escape dust, stars, ...)

Compute a third-order polynomial fit of the test image: *poly 3*. Finally, synthesize a polynomial image.



Or you can extract an artificial gradient from the image. For details see the example of [setsubsky](#) use.

515	<a href="#">synthe_sun</a>	[x-center] [y-center] [radius] [lambda (nm)] [intensity]	<p>Produces a synthetic image of the sun disk taking into account a realistic the limb darkening. The algorithm is based on the H. Neckel model (see H. Neckel, Solar Physics, 229, 13-33, 2005). The input parameters are:</p> <p>(X-CENTER, Y-CENTER) = the coordinate in pixel (and fraction) of the disk center.</p> <p>(RADIUS) = the radius (in pixels) of the synthetic disk.</p> <p>(LAMBDA) = the wavelength in nanometers (the solar limb darkening is a function of wavelength). The accepted range is between 385 nm and 1100 nm.</p> <p>(INTENSITY) = the disk center intensity.</p> <p>Typical application is a dramatic enhancement of faint contrast features on solar images, i.e. after subtraction of an observed image and the synthetic image. See the example below.</p>  <p>Left, a sun image taken by Valerie Desnoux July 20, 2004 (120 mm Astrophysics refractor + Nikon D70 DSRL). See here for details about these observations (a top of Pic du Midi observatory). The measured center of the disk is X=665.7 and Y=654.0. The radius of the disk is 604.8 pixels (the commands CIRCLE or CIRCLE2 are useful for this measure). The intensity near the center is of 7040 counts.</p> <p>Center, the corresponding synthetic image for the wavelength 490 nm (try and error method is here the best solution for select the optimal wavelength because the band pass of the DSRL is very large). The command used is</p> <p><a href="#">synthe_sun</a> 665.7 654.0 604.8 490 7040</p>
516	<a href="#">t_add</a>	[R] [G] [B] [number]	<p>Addition of the images of each color planes (equivalent <a href="#">add2</a>). Operation carried out: (R1+R2+...+Rn, G1+G2+...+Gn, B1+B2+...+Bn) -&gt; [R], [G], [B]</p>
517	<a href="#">t_add_norm</a>	[R] [G] [B] [number]	<p>Even operation that <a href="#">t_add</a> but standardizes the images on level 32700 with the need if this level is exceeded (equivalent ADD_NORM).</p>
518	<a href="#">t_composit</a>	[R] [G] [B] [sigma] [#iter] [flag max] [#images]	<p>Produces three images [R] [G] and [B] by making an optimal addition of the images of the sequences R1... Rn, G1... Gn, B1... Bn (equivalent <a href="#">composit</a>). The procedure is iterative if [#ITER] is higher</p>

			than 1. The threshold of rejection of the bad values is adjusted by the variable [SIGMA] (typical real value between 2 and 3). Level 32700 after addition is not exceeded if [FLAG MAX] = 1 (a normalization of the intensities is done). Limited to 19 input frames.
519	<i>t_copy</i>	[in R] [in G] [in B] [out R] [out G] [out B] [number]	Duplicates a trichromatic sequence. Carry out the operation: (in R1... in Rn, G1... in Gn, B1... in Bn) -> (out R1... out Rn, out G1... out Gn, out B1... out Bn)
520	<i>t_cregister</i>	[threshold] [number]	Registration of planetary images by adjusting a circle on level [THRESHOLD] around planet limb (equivalent to <i>cregister</i> ).
521	<i>t_div</i>	[R] [G] [B] [number]	Divides the sequences R1... Rn, G1... Gn, B1... Bn by the images [R], [G] and [B] (equivalent has <i>div</i> , but the parameters of standardization is calculated automatically in <i>t_div</i> ). Carry out the operation: (R1/[R]... Rn/[R], G1/[G]... Gn/[G], B1/[B]... Bn/[B]) -> (R1... Rn, G1... Gn, B1... Bn)
522	<i>t_gauss</i>	[sigma]	Convolution by Gaussian of the trichromatic image defined by the file R, G and B. Equivalent to command <i>gauss2</i> .
523	<i>t_mult</i>	[coef R] [coef V] [coef B] [number]	Multiplies each image of the sequences R1... Rn, G1... Gn, B1... Bn by constants (equivalent with <i>mult2</i> ). Carry out the operation: (R1*[coef R]... Rn*[coef R], G1*[coef G]... Gn*[coef G], B1*[coef B]... Bn*[coef B]) -> (R1... RN, G1... Gn, B1... Bn)
524	<i>t_ngain</i>	[norm] [number]	Multiplies each image of the 3 sequences by a constant calculated by Iris in manner what the median level of each image is equal to [NORM] (equivalent with <i>ngain2</i> ). This command is in particular used for stack flat-field images before median composite.
525	<i>t_noffset</i>	[norm] [number]	Adds (or subtracts) to each image of the three sequences a constant calculated by Iris in manner what the median level of each image is equal to [NORM] (equivalent <i>noffset2</i> ). Useful to bring at the same level the sky background on deep-sky images.
526	<i>t_offset</i>	[offset R] [offset V] [offset B] [number]	Adds a constant to each image of the sequences R1... Rn, G1... Gn, B1... Bn (equivalent with <i>offset2</i> ). The constants can have negative values. Carries out the operation: (R1+[offset R]... +Rn+[offset R], G1+[offset G]+... +Gn+[offset G], B1+[offset B]+... +Bn+[offset B]) -> (R1... RN, G1... Gn, B1... Bn)
527	<i>t_pregister</i>	[size] [number]	Equivalent with <i>preregister</i> for the registration of the planetary images. Registration is calculated on the sequence G1... Gn then is the parameters of translation are applied to the sequences R1... Rn, B1... Bn. It is thus supposed that it is the green component of the three-color process which presents the best details and contrasts
528	<i>t_preregister</i>	[number]	Equivalent of <i>preregister</i> command.
529	<i>t_register</i>	[number]	Equivalent of <i>register</i> command. Ideal for the deep-sky images (uses the position of a reference star to be selected in the first images of one of the series, image G1.FIT for example).
530	<i>t_restore</i>		Carries out the opposite operation of <i>t_store</i> : restore in the files image R, G and B the contents of file # R, # G and # B

531	<i>t_scale</i>	[option] [xf] [yf]	Changes the scale of a trichromatic image defined by the files R, G and B. Same parameters as command <i>scale</i> .
532	<i>t_select</i>		Simultaneous sorting by decreasing quality the images in the red, green and blue planes (equivalent to <i>select</i> ). It is necessary to have run before command <i>bestof</i> on one of the components trichromatic (the green one for example)
533	<i>t_smedian</i>	[R] [G] [B] [number]	Median stack of the sequences R1... Rn, G1... Gn, B1... Bn by producing the images [R] [G] and [B] (equivalent with <i>smedian</i> ). Algorithm fast but limited to 19 images.
534	<i>t_smedian2</i>	[R] [G] [B] [number]	Even function that <i>t_smedian</i> , slightly slower but the number of images is unlimited (equivalent to <i>smedian2</i> ).
535	<i>t_store</i>		Copies the three images having for name R, G and B respectively in files # R, # G and # B
536	<i>t_sub/t_soust</i>	[R] [G] [B] [number]	Subtracts the images [R], [G] and [B] to the sequences R1... Rn, G1... Gn, B1... Bn (equivalent with <i>sub2</i> or <i>sub</i> ). Carry out the operation: (R1-[R]... Rn-[R], G1+[G]... Gn+[G], B1+[B]... Bn+[B]) -> (R1... Rn, G1... Gn, B1... Bn)
537	<i>t_trichro/t_tr</i>		Combines an RGB image from the files named R G B correspondingly (if they exist and have the same size)
538	<i>t_unsharp</i>	[sigma] [coef] [flag]	Displays a color image starting from the file's images R, G and B, but applies a unsharp masking filter with each component as a preliminary. Same parameters as command <i>unsharp</i> .
539	<i>t_unsharp2</i>	[sigma] [coef] [flag] [number]	Calculates the unsharp masking of each image of the three sequences (equivalent with <i>unsharp2</i> ).
540	<i>tcl</i>		Execute a Tcl script. [SCRIPT] is the name of the scripting file in the disk. [PARAMETER1], [PARAMETER2], etc is a variable number of parameters for the script. For details about the powerful TCL command see the <a href="#">ASTP protocol Web page</a> .
541	<i>test1</i>	[transfer number]	
542	<i>test2</i>	[read number]	
543	<i>text</i>	[text] [x] [y] [intensity]	Allows writing a text in the image. This function modifies the intensity of the pixels in the 16-bits image. You can retrogress while using the button "undo" tools bar. [TEXT] is the contents of the text. [X] and [Y] are the coordinates in pixels of the position of the beginning of the text. [INTENSITY] is the intensity of the text, a number ranging between 0 and 32767. Example: <i>text</i> Zeta_Tau____April_2002 30 5 32000 Note the use of the "_" character for spaces.
544	<i>th_cut</i>	[in] [out] [high level] [low level] [number]	Modifies the thresholds in the header of the images.
545	<i>tiff2pic</i>	[in] [out] [first index] [number]	Converts a TIFF sequence into a PIC or FITS.

546	<i>tilt</i>	[x0] [alpha]	Rectifies a spectrum whose axis of dispersion forms an angle [ALPHA] compared to the horizontal axis of CCD sensor. Calculation is done by vertically shifting each column of the adequate fraction of pixel. The pivot of rotation is located at the horizontal coordinate [X0] counted in pixels. The angle is in degrees and can be signed.
547	<i>tilt2</i>	[input] [output] [x0] [alpha] [number]	Same as <i>tilt</i> but for sequences.
548	<i>time</i>		Shows the time of obtaining the image
549	<i>track</i>	[name] [number]	Analyzes a star selected in a sequence of images and produces files DX.DAT, DY.DAT, and STAT.DAT allowing to observe the drift of a telescope according to time. The content of file STAT.DAT is column by column: <ul style="list-style-type: none"> <li>• The image index</li> <li>• The X-shift relative to the first image of the sequence</li> <li>• The Y-shift relative to the first image of the sequence</li> <li>• The absolute X coordinate of the selected object</li> <li>• The absolute Y coordinate of the selected object</li> <li>• The FWHM along X-axis</li> <li>• The FWHM along Y-axis</li> <li>• The integral of the stellar PSF</li> <li>• The local sky-background value</li> </ul>
550	<i>trail</i>	[y0] [y1] [y2]	Aligns the points of drift star trace or a spectrum.
551	<i>trail2</i>		Interactive version of <i>trail</i> , requiring only two mouse clicks on either side of a star trail or spectrum to get the straightness of that trail or spectrum.
552	<i>trans</i>	[dx] [dy]	Shifts the image by the entered amount.
553	<i>trans2</i>	[input] [output] [dy (pix./hour)] [dy (pix/hour)] [number]	Transform a sequence of image into another sequence by making a translation of a certain number of pixels depending on the hour of acquisition of the images and the shift in X and Y specified in parameters (in pixels per hour). The typical use is the centering of a sequence on the movement of a comet or of an asteroid in such a way that this object appears fixed after stacking. One generally proceeds in two times: registration on a star of the field ( <i>register</i> command for example), then, application of the command <i>trans2</i> command. Supposing that the ephemeris (or a direct measurement in the image) tell us that mobile object moves 0.230 <i>pixel/hour</i> in X and of -0.763 <i>pixel/hour</i> in Y in the sequence I1, I2, I3... I20. One obtains a new sequence J1, J2.... J20 where the movement of the object is cancelled while making:  <i>trans2</i> I J 0.230 -0.763 20
554	<i>transb</i>	[dx] [dy]	Spline version of <i>trans</i> function for RGB images.

555	<i>tri2mono</i>		Adds RGB planes and convert picture to greyscale. In order to prevent clipping you can multiply in-memory image by 1/3 before applying this command.
556	<i>trichro</i>	[r] [g] [b]	Combines three separate greyscale images into a RGB picture.
557	<i>unsharp</i>	[sigma] [coef] [flag]	<p>Performs filtering by unsharp masking on the current image. This type of filtering is high pass: it tries to eliminate the low frequencies in the image, accentuate the high frequencies and then add them, with a weighting factor, to the original image. The process is:</p> <ul style="list-style-type: none"> <li>• Convolve the image to be processed by a gaussian whose width is in the parameter [SIGMA] (see the <i>gauss</i> and <i>gauss2</i> commands).</li> <li>• Subtract the result of this convolution from the original image. The result is an image with an average level close to zero, and whose low frequencies (slow variations in the image) have been strongly attenuated. At this stage, a positivity constraint can be applied (this sets all the negative values to zero). If [FLAG]=0, the constraint is not applied. This option should be used for processing planetary images. If [FLAG]=1, the constraint is applied. This option should be used for processing deep sky images.</li> <li>• Multiply the result of the preceding step by the constant whose value is in the variable [COEF], and then add it to the original image. The result is the final image.</li> </ul> <p>Unsharp masking is a very simple, but nevertheless powerful tool for enhancing the contrast of an image. It is one of the basic tools for planetary image processing. With Iris, unsharp masking can easily be performed with the <i>gauss</i> (or <i>gauss2</i>), <i>sub</i>, <i>mult</i>, and <i>add</i> commands. The <i>unsharp</i> command simply puts all these operations into one command. The values of the variables [SIGMA] and [COEF] need to be adjusted by trial and error. As a general rule, a small [SIGMA] (&lt;1) and a large coefficient enhance the finest details, but then the noise may become overwhelming.</p>
558	<i>unsharp_trichro</i>	[R] [G] [B] [sigma] [coef] [flag]	<p>Carries out an operation of filtering of unsharp masking type on the three components of a RGB-image at the same time and display the result. Syntax is similar to command <i>unsharp</i> but it is necessary to provide the name of the three images correspondents to the red, green and blue channel. For example, for a planetary image one will do something which resembles:</p> <pre><i>unsharp_trichro</i> R G B 1.8 4 1</pre>
559	<i>unsharp2</i>	[in] [out] [sigma] [coef] [flag] [number]	Same as command <i>unsharp</i> but for a sequence of images.
560	<i>unsharp3</i>	[sigma] [coef] [edge tuning] [flag]	Has a similar effect to the <b>Unsharp Masking...</b> dialog box in the <b>Processing</b> menu.
561	<i>USB</i>	[valeur]	
562	<i>vancittert</i>	[fwhm] [#iter]	Restores an image with the Van-Cittert method. [#ITER] gives the number of iterations to be performed (typically between 5 & 20), whereas [FWHM] is the characteristic FWHM of the stars in the image to restore.

			<p>Before running the command, lower the sky background to a level close to zero (use the <i>offset</i> command for example).</p> <pre>load m51 offset -100 vancittert 1.8 5</pre>
563	<i>video</i>	[x1] [x2] [integration time] [cluster size] [cluster number]	<p>The parameters [X1] and [X2] define an area along the horizontal axis which will be used to produce the final image. The smaller the width, the faster the acquisition.</p> <p>The integration time is in seconds.</p> <p>The height of an elementary block is then given in [CLUSTER SIZE]. And the number of blocks is the number of thumbnails desired.</p> <p>But it should be noted that this command is applicable only when working with CCD cameras operating in video mode. Nowadays when cheap CMOS planetary cameras are almost a standard in amateur astronomy this approach is quite redundant.</p>
564	<i>video_extract</i>	[name] [high] [first index]	<p>The parameter [NAME] is the generic name of the extracted images. [HEIGHT] is the width of the sub-images. [FIRST INDEX] is the index of the first image of the sequence.</p>
565	<i>video_grid</i>	[size]	<p>Displays in the image a horizontal grid with a step in pixel equal to the value of the parameter size]. This grid is a help to position at the good place objects for the VIDEO mode acquisition (Audine CCD camera). You can also reveal this grid by typing with the keyboard simultaneously the combination of key Ctrl+F6 (you must have clicked in the image before).</p>
566	<i>view</i>	[nom] [high level] [low level] [number]	<p>Display a sequence of images of generic name [NAME] by using the visualization thresholds [HIGH] and [LOW]. See also the command <i>th_cut</i> (which modifies the thresholds in the header of the images).</p>
567	<i>visu</i>	[high] [low]	<p>Sets the visual thresholds</p>
568	<i>wavelet</i>	[out1] [out2] [scale]	<p>The WAVELET command performs a wavelet transform on an image. This analysis decomposes the current image into images that each show details of increasing scales. This amounts to multiresolution analysis of the initial image.</p> <p>The algorithm used in Iris is called "à trous" ("with holes"). It calculates an approximation to the input image by considering only the pixels on the crossings of a mesh whose step varies by a factor of 2 between two scales. In a way, as the scale increases, you see the objects as they would look if you moved away from them by a factor of 2 from one decomposition to the next.</p> <p>Points that are between the mesh crossings are approximated with an interpolation function - the wavelet. The interpolation is done with a 3x3 matrix. Many classes of wavelets can be defined, but they all have common characteristics (in particular, they are functions with a zero average). Wavelet analysis is a new method for interpreting the contents of images. It studies structures of different sizes in the image, and analyzes their relations. It is called analyzing the hierarchy of structures of the objects in the image.</p>

The decomposition of the image into structures with distinct scales allows the reconstruction of the initial image so that only the pertinent details remain (the algorithm programmed in Iris permits this reconstruction). This yields a very precise filtering of the image. Note also that wavelets are at the core of some image compression algorithms.

The parameter [OUT1] contains the generic name of the approximated images with increasing scales. The number of scales is contained in the variable [SCALE]. The image with the smallest scale has the index 1, the following 2, and so on.

The parameter [OUT2] contains the generic name of the images corresponding to the difference between two successive approximations (namely the wavelet coefficients). These images contain the details that disappear from one scale to the next (the image with index 1 contains the details from scale 1, the image with index 2 those from scale 2, the image with index 3 those from scale 4, the image with index 4 those from scale 8, and so on).

The number of scales analyzed is in [SCALE]. Typically scale=3 to 5.

Let's perform wavelet analysis on the image M51:

```
load M51
visu 800 40
```

Wavelet analysis will allow us to study the relationship between structures with different scales (groups, arms, nucleus, etc.). We will also attempt to enhance the appearance of the groups by reconstructing the image.

Perform the transform:

```
wavelet I J 6
```

The images I1...I6 contain the successive approximations of the image in increasing scales. The images J1...J6 are the wavelet coefficients for the successive scales 1, 2, 4, 8, 16, and 32.

Let's examine the wavelet coefficients for each of the scales:

```
load J1
visu 100 -100
```

It is difficult to recognize the galaxy on this image. In fact, at the scale 1, the image contains mostly noise, which is why it is so uncorrelated with the input image.

```
load J2
visu 100 -100
```

At this scale, the nodules in the arms are brought out.

```
load J3
visu 100 -100
```

The arms start to be continuous. Note the strong hierarchical relationship of the details between this scale and the previous one.

*load J4*  
*visu 100 -100*

The large-scale structures of the galaxy are becoming evident...

*load J5*  
*visu 200 -200*

This trend continues...

*load J6*  
*visu 300 -300*

At this last scale, only the massive central part of the galaxy is visible.

You can, of course, examine the corresponding approximations (images I1.PIC, I2.PIC, etc.). The last one is special since it is the residual of the transform. To understand the contents of this images, imagine how the galaxy would look if you moved away from it while you were observing it through a telescope with constant resolution.

It is possible to reconstruct the initial image by adding the set of wavelet coefficients and the residual:

*load J1*  
*add J2*  
*add J3*  
*add J4*  
*add J5*  
*add J6*  
*add I6*  
*visu 800 40*

or

*add2 J 6*  
*add I6*

However, since we have seen that the coefficients of the first scale only correspond to noise, it is wise to eliminate the details of this scale from the reconstruction. Also, we want to enhance the contrast of the stellar groups. To do this, we will assign a weight greater than 1 to the coefficients corresponding to the details of scale 2 and 4 (the weighting factors of 2 and 1.5 that we have used are arbitrary; it requires several trials to estimate the best values for a given application):

			<pre>load J2 mult 2 save K load J3 mult 1.5 add K add J4 add J5 add J6 add I6 visu 800 20</pre> <p>The noise has been noticeably reduced and the contrast increased. You have probably noticed that the previous process is very similar to traditional unsharp masking. However, one of the fundamental differences is that the unsharp mask is performed with only one scale while the <i>wavelet</i> command offers a multi-scale analysis that acts more finely on the contents of the final result. <a href="#">Click here for some example.</a></p>
569	<i>wavelet_filter</i>	[noise] [coef_sigma] [number]	<p>Applies a filter to the current image for noise suppression. For this, wavelet filter uses an evolved procedure that limits resolution degradation. [NOISE] parameter is the RMS noise in the background of the image (it can be measured with the command Statistics from the contextual menu. [COEF] is a rejection noise factor. Typically [COEF] value is between 2 et 4. [NUMBER] is the number of wavelet plane (see <i>wavelet</i> command). Typically [NUMBER] = 5.</p> <p>Example for M51 image:</p> <pre>load M51 wavelet_filter 7.8 3 5</pre> <p>Correctly works only with greyscale images otherwise will process only Red channel.</p>
570	<i>wavelet2</i>	[out1] [out2] [scale]	Same command as <i>wavelet</i> but using a 5x5 matrix
571	<i>wdate</i>		Writes the acquisition date in the image. The same effect is obtained while making: Ctrl+F8. For memory, the combination of Ctrl+F9 keys copies the image in the clipboard.
572	<i>white</i>		Multiplies RGB channels by such coefficients that would set the selected area neutrality white.
573	<i>white2</i>		The <i>white2</i> command harmonizes the three layers of a color image (48-bits) by equalizing the intensity of the same star in these layers (multiplication by suitable coefficient). The program carries out a Gaussian adjustment of star. The selected star must be surrounded by a small rectangle before run the command. Select for example a solar type star (type G2V). See also the <i>white</i> which uses the median value of an area selected with the mouse and <i>black</i> commands.

574	<i>wiener</i>	[coefficient]	<p>Applies Wiener inverse filter. The form of the filter is:</p> $Wiener\ filter = \frac{P^*}{ P ^2 + k}$ <p>where <math>P</math> is the Fourier transform of the PSF, <math>P^*</math> is its conjugate value and <math>k</math> is a constant. The constant <math>k</math> fixes the degree of low frequency in the restored image (and the importance of the noise). If <math>k = 0</math>, a pure inverse filter is applied, but the amplification of the noise can be considerable. Choose <math>k</math> between 1 and 0.0001.</p>
575	<i>wiener2</i>	[iteration #] [coefficient]	Same as <i>wiener</i> but allows several iterations.
576	<i>win</i>		Allows cropping the image by interactively selecting two points manually.
577	<i>win_webcam</i>	[x1] [y1] [x2] [y2]	<p>Defines coordinates of a window in the images resulting from a webcam source and acquired with the command <b>Images acquisition</b> of <b>Webcam</b> menu. Let us suppose for example that the format of the images is 320x240 pixels (<b>Size of the images...</b> from <b>Webcam</b> menu). If before launching acquisition you type in the console:</p> <pre><i>win_webcam</i> 100 1 120 240</pre> <p>then, all the acquired images (including in sequence mode of course) will have a size of 20x240 pixels, isolating in the full format image a sub-images delimited by the coordinates (100,120)-(1-240). To reset to the full format, you must in this example type in the console:</p> <pre><i>win_webcam</i> 1 1 320 240</pre> <p>(the effect is similar if you leave, then enter the program). The interest of <i>win_webcam</i> is a substantial saving in place on the disc if the object observed is of small size compared to the format of the image. The speed of acquisition is also increased. Another typical application is the acquisition of large number of images of a spectral line of the Sun whereas the disc of this one travels on the input slit of the spectrograph. It is in this manner possible to reconstitute a monochromatic image of the Sun. See also command: <i>scan2pic</i>.</p>
578	<i>window</i>	[x1] [y1] [x2] [y2]	Crops the image with selected coordinates.
579	<i>window2</i>	[in] [out] [x1] [y1] [x2] [y2] [number]	Same as <i>window</i> but for a sequence.
580	<i>window3</i>	[size]	Crops a square area with the entered size around selected area.
581	<i>window4</i>	[in] [out] [size] [number]	Same as <i>window3</i> but for a sequence.
582	<i>window5</i>	[X-size] [Y-size]	Crops an area with the entered size of [X] and [Y] pixels around center of the selected area
583	<i>work</i>	[path]	<p>Definition of the working path from the console. Example:</p> <pre><i>work</i> C:\MYIMAGES\SESSION21</pre>



## Drizzle

Dither algorithm performs an optimal adding of a sequence of images as far as resolution is concerned. The principle is that, at sub-pixel level, shifts between individual input images are nearly randomly distributed. For example, a star in the first image may be centered perfectly in the middle of a pixel, whereas it will be across two pixels in the second one, and so on. Since it is easy to know the exact shift between the images, it is possible to create an output image with a finer sampling, in which resolution may be increased with respect to each input image. In fact, energy from each input pixel is dropped in the output image, and the whole process may be compared to a drizzle.

Drizzling is adapted to undersampled images, for example when the telescope focal length is too short for the pixel size. One may consider that the system is undersampled when FWHM is smaller than 2 pixels. In this situation much of the information lost in undersample regime can be restored.

Before using drizzling technique, it is necessary to know the exact shift between the images. It is also very important that all the input images are acquired in the same conditions: same exposure time, same sky background level. If this is not the case, you have to adjust offset and gain prior applying drizzling algorithm (NGAIN and NOFFSET commands under Iris for example).

The drizzling algorithm step by step (see figure 8):

Step 1: Reduce or coarse (by calculation!) the size of the pixels in the starting image, but preserve the same interval between pixels.

Step 2: Project in the final image fine grid after a geometrical transformation (take into account, if necessary, shifts, rotations, optical distortions).

Step 3: Calculate the fraction of the pixel projected in each cell of the grid of the final image and add this fraction with the current value with the output pixel.

Step 4: Start again at step 1 for each input image.

The "shrink" pixel size at the step 1 is crucial. We define  $\text{pixfrac}$  as the ratio of the linear size of the coarse pixel to the original input pixel linear size. If  $\text{pixfrac}=0$  the drizzle algorithm is equivalent to interlacing, while the traditional shift-and-add is equivalent to  $\text{pixfrac}=1$ . One must choose a  $\text{pixfrac}$  value that is small enough to avoid degrading final image, but large enough that then all images are dropped, the coverage of the output image is fairly uniform. We choose typically  $\text{pixfrac}$  between 0.5 and 0.7.

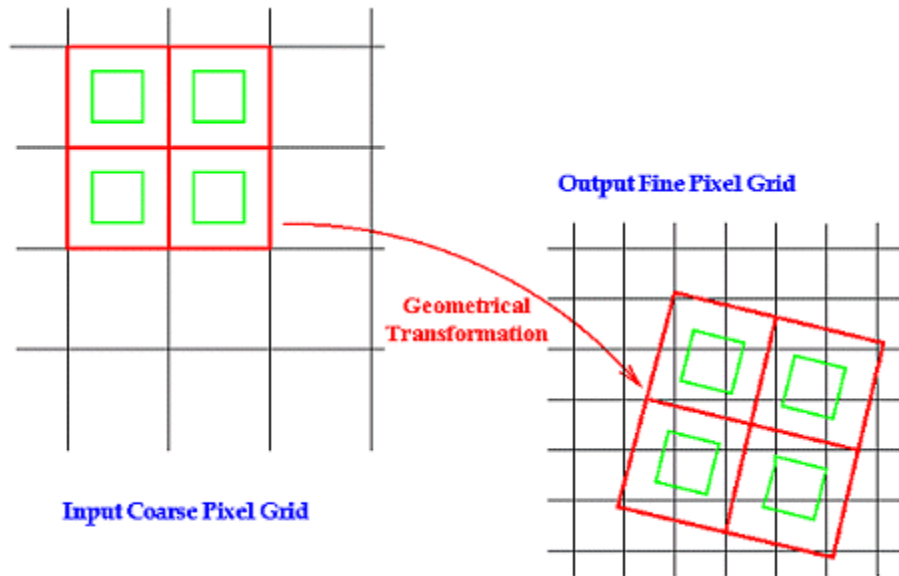


Figure 8. Schematic representation of drizzling technique. In this particular case, the central output pixel receives no information from the input image. It will not be necessary the case for the following images of the sequence, and so on. "Dark" output pixels are not a concern as long as there are enough input frames with different sub-pixel dither positions to fill in the output image. The ratio between the input grid size and the output grid size define the "scale factor" parameter.

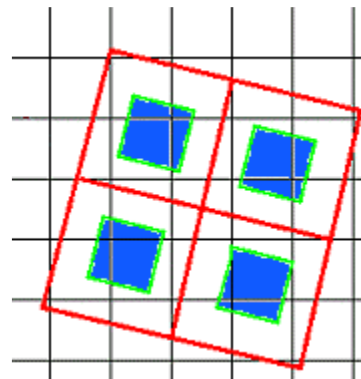


Figure 9. In blue color, area fraction of the input pixels dropped in the output pixels.

Mathematical formulation of drizzling:

if

- $i$  = intensity of the projected input pixel
- $w$  = weight of this pixel

- $a$  = fraction of the pixel projected in a cell of the output grid (fractional pixel overlap  $0 < a < 1$ )
- $I$  = current intensity in the output pixel
- $W$  = current average weight in the output pixel
- $I'$  = resulting intensity in this output pixel
- $W'$  = resulting weight of this output pixel

then

$$W' = a \cdot w + W$$

$$I' = (a \cdot I \cdot w + I \cdot W) / W'$$

The weight  $w$  of the pixel can be zero if it is a bad pixel (hot pixels, dead pixels, cosmic rays event, ...), or can be adjusted according to the local noise (the value is then inversely proportional to the variance maps of the input image).

Remember that algorithm is effective if the images are really undersampled (FWHM of 1 to 2 pixel). The displacement, and more generally, geometric distortion, between the individual input images must be perfectly well-known (to 1/10 of pixel precision typically). The number of input images must be large (10 or more) to avoid holes in the final image. Most important, displacement between the input images (dithering techniques) must be random on 2 axes. So, it is necessary to shift arbitrary the telescope between each exposure during deep-sky sessions. The amplitude of the shift can be of a few fractions of pixels in a random direction. At the processing stage the relative shifts between images are precisely determined by calculation of the centroid of stars (PSF fitting between common stars or cross-correlation between a reference image and the input images). The registration parameters are fundamental quantities for the drizzling method.

Main performances:

1. Resolution gain can be up to 2.
2. Combination of sequence images produce high resolution without sacrificing the final signal to noise ratio.
3. Conservation of photometric quality.
4. Preserve astrometric accuracy.
5. Effective removal of the bad pixels (cosmic rays, traps, etc).
6. Optimal compositing if the weight function is quite selected relative to the local noise.
7. Very good geometrical correction of the images (significant for photometry).

The Iris software implement a version of the drizzle procedure (DRIZZLE command). The algorithm was developed by Richard Hook and Andrew Fruchter to produce the The Hubble Deep Field, the deepest optical image of the universe yet taken. It is now used for many other fields.

## Map function and its parameters.

Creates an image in a given cartographic projection from an image defined in a different projection.

Cartography is a method that allows you to represent, on a plane, a surface that is generally not flat, such as the earth or any other planetary surface. This science is thus necessarily inexact since local or global deformations of the surfaces to be represented are inevitable. The choice of a cartographic projection is generally based on a compromise between different desired properties (for example, global view of the planet, conservation of area, polar view, etc.).

The types of projection can be classified into three large families of criteria:

a) according to their geometric properties. There are 4 projection categories following this criterion:

- Conformal projections. they conserve the angles between all points on the map. Used mostly in navigation, somewhat less in astronomy; a famous example is the MERCATOR projection.

- Equal-area projections. They conserve areas. Very useful when you are interested in a balanced representation of different parts of the body being mapped. Example: sinusoidal projection. A projection can be both equal-area and conformal, sometimes only in a limited domain of the representation, if you allow a certain margin of error.

- Equidistant projections: they conserve certain distance ratios, for example, only on the meridians, or on a particular parallel, etc.

- Aphylactic projections: this covers all the other projections, that is, those that more or less deform the surfaces and the angles. Some of them are commonly used anyway. Example: simple cylindrical projection.

b) according to how they are constructed. The types are:

- cylindrical projections: obtained by wrapping a cylinder around a sphere. The projection of the sphere onto the cylinder is usually normal to the cylinder, or normal to the sphere (example: MERCATOR projection).

- azimuthal projections: replace the cylinder by a plane, with the same type of projections. The important subcategory of orthographic projections are normal to the plane, which makes it seem as if the sphere is being looked at from very far away.

- conic projections: the sphere is projected onto a cone that intercepts the sphere in one or two circles, then the cone is rolled on a plane. the others, numerous and varied!

c) according to the central region represented. The types are:

- equatorial projections: these are cylindrical projections that surround the sphere along the equator, or of azimuthal projections that intercept the sphere in one point on the equator.

- polar projections: these are azimuthal projections that intercept the sphere at a pole, or conic projections for which the vertex of the cone is on the polar axis.

- transverse projections: this is the case, for example, of cylindrical projections whose intersection with the sphere is a meridian.

- oblique projections: for cylindrical projections, the intersection would be a great circle of the sphere. For example, a typical telescopic view (an image of Jupiter on a CCD matrix, etc.) corresponds to an oblique orthographic projection.

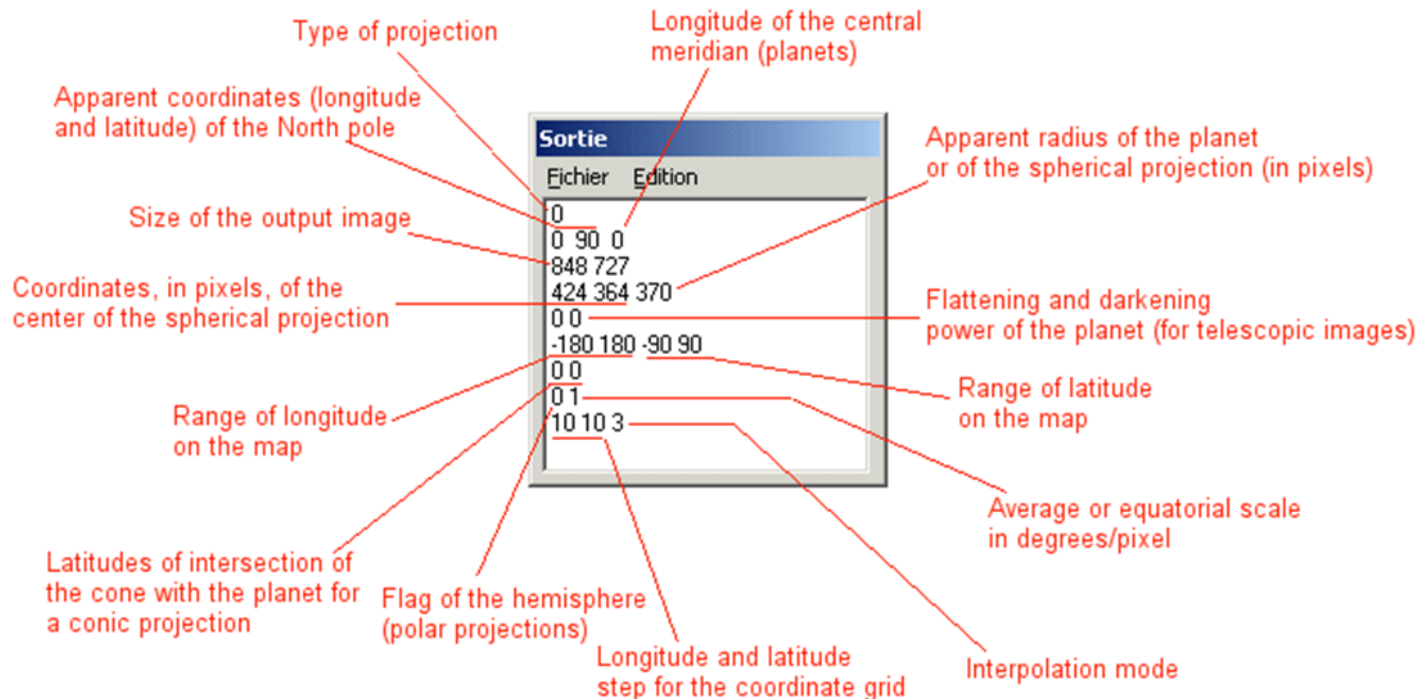
Note that the term sphere above comes from the simple application of these projections for the case of a spherical planet. The same methods are still valid for ellipsoids and are thus applied by Iris (with a more complex mathematical formulation) for non-spherical bodies (for example, Jupiter).

The essence of the MAP command is the transformation from one cartographic system to another one (or even the same one with different parameters). The input image and the output images must contain integer data.

The parameters [input list] and [output list] contain the names of text files in which there are 22 parameters that describe the cartographic system used for input and for output, respectively. The names of these files end with the extension .LST (the extension should not be specified in the command line). The files can be created with a text editor.

PARAMETER	VALIDITY
PROJ	[0,16]
LAMP	[-180,180]
PHIP	[-90,90]
MERI	[-180,180]
XW	[1,4096]
YW	[1,4096]
XC	[ ]
YC	[ ]
RC	[ ]
FL	[0,1]
POWER	0
LONG1	[-180,180]
LONG2	[-180,180]
LAT1	[-90,90]
LAT2	[-90,90]
L1	[-90,90]
L2	[-90,90]
FLHEM1	[0,1]
SCALE	>0
LONGSTEP	>0
LATSTEP	>0
INTERPOL	[1,6]

The order of the parameters in the files must be respected. There can be one or more parameters per line, with a space, tab, or carriage return between parameters. For example the default parameter arrangement is shown below:



Or in text form:

```

0
0 90 0
848 727
424 364 370
0 0
-180 180 -90 90
0 0
0 1
10 10 3

```

The files contain the values of the following parameters (we have indicated the range of validity for each parameter, but it is sometimes necessary to adjust this range depending on the type of projection used):

For a given map type you will not necessarily use all of the above parameters. A parameter that is not used should be set to 0 in the .LST file.

Here is the description of each of these parameters in the order they should have in the .LST file:

- PROJ: the type of projection used. The MAP command provides 17 projection systems (as input and output, thus 289 possible combinations!). These systems have been chosen as a function of their utility in astronomy and/or planetology. They include most of the projections in use except those that are specifically adapted to the earth, or to human activities. They are described later.

- (LAMP,PHIP): used only for a telescopic image. They are the apparent coordinates (longitude and latitude) of the North pole of the planet on the image. Apparent means that the x axis is directed towards the user (perpendicular to the screen), the y axis is horizontal (directed towards the right) and the z axis is vertical, directed upward. Note that in MAP, all angles are in degrees. Also, longitudes are between -180 and +180 degrees, and latitudes are between -90 and +90 degrees.

- MERI: this is the longitude of the central meridian of the planet at the time of the observation. This information is available in ephemerides, once a coordinate system has been chosen.

- (XW,YW): the size of the output image, in pixels, for telescopic images. These parameters are only required in [output list], and only for a telescopic projection.

- (XC,YC): the coordinates, in pixels, of the center of the planet in a telescopic projection. This center can be outside of the image itself.

- RC: the apparent equatorial radius of the planet, in pixels, for a telescopic projection. This radius can be smaller or larger than the size of the image.

All the preceding parameters are required for a telescopic projection.

- FL: the flattening of the planet (used only for telescopic projections), which can be obtained from many books on the planets.

- POWER: the darkening power of the planet (only for telescopic projections). In the current version of MAP, the planets are assumed to be without phase (full moon or "full Mars"), and the limb darkening law is a power law of the cosine of the apparent longitude angle. This power can be determined empirically. For example, depending on the filters used, the value for Jupiter is generally between 0.4 and 0.5.

- (LONG1, LONG2): the range of longitude on the map. For global cartography, these values would be -180 and 180 respectively, or for regional cartography, any other value.

- (LAT1, LAT2): the range of latitude on the map, with  $LAT1 < LAT2$ . For global cartography, these values would be -90 and 90, if the type of projection allows it, or otherwise, other appropriate values.

- (L1, L2): the latitudes of intersection of the cone with the planet for a conic projection (see ahead for more information).

- FLHEMI: a flag that indicates which hemisphere is used for polar projections (conic or azimuthal), with 0 for the northern and 1 for the southern hemisphere.

- SCALE: the average or equatorial scale, depending on the type of projection, in degrees/pixel. Pay attention to this variable for [output list] because it directly affects the size of the output image. For example, an image with a simple cylindrical projection from -180 to 180 degrees will have a horizontal size of 360 pixels for a scale of 1 degree/pixel, but 3600 pixels if you put 0.1 degrees/pixel by accident! Make sure you choose the value for this variable carefully.

- (LONGSTEP, LATSTEP): these variables are not used in the MAP command, they are only used for the GRID command (see GRID).

- INTERPOL: the algorithm used for interpolate the output image. You can select 6 value (1...6). 1 = nearest point method, not recommended. 2 = classical bilinear interpolator. 3 = bicubic method (a good balancing between the resolution of the images and the calculation speed - recommended). 4 = spline interpolation, very good result, but more time consuming. 5 = a more elaborate version of spline #4, but very slow. 6 = sinc interpolation (the better result in term of image quality but very long execution time, try on 16-bits images first, not true colors - not very recommended for normal work).

Here is the description of the different types of projections available with the MAP command:

PROJ = 0: perspective = telescopic view = oblique orthographic projection. This type of projection (aphylactic) is particularly important in astronomy because it corresponds to the images acquired with a telescope, that is, an

actual view of the planet in the sky. The required parameters are LAMP, PHIP, MERI, XC, YC, RC, FL, and POWER, plus XW and YW if this projection is used for output. Mathematically, the projection is normal to the plane in which the planet "sits" in the desired attitude. Thus, you can realize equatorial, transverse, polar, or oblique projections. As input, this projection can create any type of map from telescopic images. As output, you can simulate planetary telescopic images, including original views, such as a polar view of Jupiter!

- PROJ = 1: simple cylindrical projection - this is one of the most popular projections, with a regular grid that is identical in latitude and longitude. An entire planet can be represented. There are significant distortions in longitude near the poles. However, this representation is the most practical because of the linear relationship between the image pixels and the planetary coordinates. It is recommended as the basic projection. The required parameters are LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 2: LAMBERT equal-area cylindrical projection - this projection is normal to the cylinder. The poles are less deformed (at least in latitude) and the projection is equal-area, which can be useful. A global representation of the planet is possible. The required variables are LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 3: MERCATOR projection - this is one of the most famous cylindrical projections, and it has the advantage of being conformal, thus its utility in navigation (loxodromy,...). It is less interesting in astronomy, even though it is used as a standard in most planetary organizations. Note that this map dilates elevated latitudes quite a bit (the poles go to infinity), so it is strongly advised to not use this projection beyond latitude 60 or 65°. As an example, the zones at latitude 60° are dilated by a factor of 2 with respect to the equatorial scale, and the zones at 80°, by a factor of 33! The required variables are LONG1, LONG2, LAT1(>-90°), LAT2 (<90°), SCALE.

- PROJ = 4: transverse MERCATOR projection: this is the transverse form of the MERCATOR projection (the cylinder is in contact with both poles). It is still conformal, of course, and has the advantage of not distorting the poles. Part of the equator is now excessively dilated. For output, the variables XW and YW limit the extent of the map (which would otherwise go to infinity), and the meridian on which the cylinder is resting is automatically equal to half of the sum of LONG1 and LONG2. The required variables are LONG1, LONG2, LAT1, LAT2, SCALE, and, if this projection is used as output, XW and YW.

- PROJ = 5: sinusoidal projection, also called the SANSON-FLAMSTEED equal-area projection. This projection can be used for a global representation of the planet, it is equal-area, and deforms the poles less than the cylindrical projections, though the shapes are somewhat inexact, because of the pointed look of the map. The deformation is less near the central meridian, which you can vary if necessary. The required variables are MER1, LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 6: MOLLWEIDE equal-area projection (also known as BABINET) - this is a classical projection, in the form of a football. It can be used for a global representation and is more pleasant to the eye than the SANSON-FLAMSTEED. The deformations become larger if you choose a central meridian that is far from half the sum of LONG1 and LONG2. It is rather complicated mathematically (long calculation time). The required variables are MER1, LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 7: POSTEL polar azimuthal equidistant projection - this is an equidistant form of the polar projection. It is equidistant along the meridians. Note that by a mathematical extension of this projection, you can represent the entire planet (a polar projection is a priori limited to one hemisphere). The flag FLHEMI selects which pole is on the map. The chosen central meridian is displayed vertically on the screen, as in all the other polar projections. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, FLHEMI, SCALE.

- PROJ = 8: central polar azimuthal projection (also called gnomonic) - the center of the projection is the center of the planet and the projection is normal to the surface of the planet. The equator goes to infinity, so this projection is limited to the hemisphere in contact with the map (without the equator). The variables LAT1 and LAT2 should be chosen to be in agreement with the flag FLHEMI (MAP will send appropriate error messages otherwise). For example, if FLHEMI = 0 (representation of the northern hemisphere), the two extreme latitudes of the map must be strictly positive. This map is mostly useful for navigation, because the loxodromic routes are arcs of the circle, and the orthodromic routes (the shortest path between two points) are line segments between two points on the map. In astronomy, obviously, this is less interesting. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, FLHEMI, SCALE.

- PROJ = 9: LORGNA polar azimuthal equal-area projection - this is an equal-area form of the polar projection. By extension, the entire planet can be represented. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, FLHEMI, SCALE.

- PROJ = 10: a polar azimuthal stereographic projection (orthomorphic) - the center of the projection is the opposite pole and the projection is normal to the sphere. A global representation is possible, and is fairly representative of the shapes, at least in the hemisphere in contact with the map. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, FLHEMI, SCALE.

- PROJ = 11: LAMBERT polar azimuthal equal-area projection - this is another equal-area form of the polar projection, used a lot since it is more realistic than the LORGNA projection. By extension, the entire planet can be represented. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, FLHEMI, SCALE.

- PROJ = 12: ALBERS polar conic projection - this is a simple form of the conic projection. These projections are characterized by their fan-shaped form. They are generally used because they do not deform the represented objects very much, at least not near the center of the map. A conic projection is obtained by projecting the planet on a cone. There are then one or two circles of intersection of the cone with the planet (two circles are obtained when the cone crosses the surface of the planet, whereas a single circle is produced when the cone is tangent to the planet). Note that if the single circle approaches the equator, the vertex of the cone goes to infinity and you get a cylindrical projection. In general, the latitudes of the two circles are L1 and L2. L1 represents the latitude of intersection that is closest to the vertex of the cone. Choose L1 and L2 to be in agreement with FLHEMI. For example, if FLHEMI = 0 (north pole), you could have L1 = L2 = 50°, but not L1 = L2 = -50°. You could have L1 = 50° and L2 = 20°, or -40°, but not -60° (since a cone whose vertex is above the north pole cannot intercept the sphere at L1 = 50° and L2 = -60°). A little common sense is enough to avoid mistakes here. In addition, the error messages from MAP will help. The variable MERI determines the meridian that is displayed vertically on the map. The ALBERS projection can be used for a global representation of the planet. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, L1, L2, FLHEMI, SCALE.

- PROJ = 13: LAMBERT polar conformal conic projection - this is a conformal version of the conic projection, fairly practical. A global representation of the planet is possible. The same remarks as for the ALBERS projection apply. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, L1, L2, FLHEMI, SCALE.

- PROJ = 14: equatorial polyconic projection - with this projection, only one hemisphere at most can be represented, but there is minimal deformation. This is an aphyllactic projection, but with a good rendering of the shapes and relative areas. The map is centered on the central meridian indicated in MERI. This projection is rather complicated mathematically, and so takes a long time to calculate. The required variables are MERI, LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 15: HAMMER equal-area projection - this projection resembles the polyconic projection and the MOLLWEIDE projection. It has the advantage of being equal-area, and does not distort planetary details very much. A global representation of the planet is possible. Its only defect is a long calculation time (but not as long as the two previously mentioned projections). The required variables are MERI, LONG1, LONG2, LAT1, LAT2, SCALE.

- PROJ = 16: USGS-type projection - this projection is a grouping of several projections. There is an equatorial MERCATOR projection that is automatically limited to +60 and -60 degrees in latitude and 360 degrees in longitude starting from LONG1, plus 2 stereographic azimuthal polar projections that are displayed underneath it and cover the poles up to the 50th parallel. The map corresponding to the north pole is on the left, with the central meridian equal to 0°, and the map corresponding to the south pole is on the right, with a central meridian of 180°. This representation is obviously global, and is the standard used by the United States Geophysical Survey (USGS), which is the agency in charge of creating official planetary maps (Mars, Jupiter, satellites of the giant planets, ...). This map has been chosen because of the complementarity of the different types of projections, the limited distortion of the regions, and the conformity of the set. The only parameter to provide is LONG1 (which it is best to set to -180). In order to interpret the file FILE\_IN (or FILE\_OUT), it is advisable to set

LONG2 to 180 (even though it is not used in this projection).

Remark: Each time the projection is changed, the image undergoes numerous operations, notably bilinear interpolation on the pixels. This creates a slight smoothing of the image, thus it is advised to not iterate too many times so that the resolution of the maps does not deteriorate. Therefore, if you have produced a

map in projection "b" from a map in projection "a", and you now want a map in projection "c", it is better to start from the "a" projection than from the "b", as far as the resolution of "c" is concerned.

See also: GRID, REC2MAP, MAP2REC, SKY2MAP, MAP2SKY

Cartography is the base for all serious morphological studies in planetology. There are multiple applications, such as:

- Measurements on the maps (coordinates, distances, areas, etc.).
- Tracking phenomena (maps taken at different times are directly comparable when they are brought to an identical projection by division, subtraction, etc.).
- Morphological or structural analysis (classification, etc.).
- Merging sub maps.
- Presentation of results (MAP can even be used with trichromatic images!).
- Simulation (create telescopic views, for example).

Merging sub maps is the principal application, at least at first - this is what allows you to create a global map from elementary images. For example, start with a collection of 6 images of Jupiter taken with about one and a half hour intervals during one night. For each image, create an individual [input list] with the necessary information, and create as many output images as input images (but with a unique [output file] so that all the maps are identical). In this example, you would have 6 maps of Jupiter (for example, simple cylindrical) that can now be combined. Generally, you limit the map to a valid zone, on either side of the central meridian, that does not cover all that is visible on the image, because the edges of the image are too distorted. They are often over or under corrected for limb darkening as well. You can obtain the valid zone directly from the output map by adjusting LONG1 and LONG2. If you notice that the result is not as desired, you can also apply the EDGE command (see the EDGE command) to set to zero the areas on the map that you judge to be undesirable after the projection. In any case, you will finally have 6 maps where the non zero information is valid. Then you only need to execute the MOSA command map by map (see MOSA), using the output from one execution as the input for the next one. The shift between the maps is 0 in both x and y and the mosaic type can be chosen (the best is probably option 5, which considerably attenuates the overlap effects). After 5 executions of MOSA, the final map will exist. In this example, if each map had been limited to 40 degrees on each side of the central meridian (thus to zones definitely valid- try not to go past 60 degrees if possible), the result will be a global map of Jupiter obtained during one night. The operation is easiest on Jupiter because of its short rotation period. This method is still valid for example for Mars, though it takes several weeks of observations.

For the moon, global cartography is difficult from the earth, but, thanks to MAP and MOSA, you can create, for example for each lunar pole, a polar map of Luna Incognita. Little by little, as the observations accumulate, the new images fill in the gaps in the old ones. Using the options available in the MOSA command, you can even replace old data on the map with new, possibly better data.

## Temp links

<http://www.astrosurf.com/buil/iris-software.html>

[http://www.astrosurf.com/buil/iris/nav\\_pane/CommandsFrame.html](http://www.astrosurf.com/buil/iris/nav_pane/CommandsFrame.html)

<http://astrosurf.com/buil/us/iris/astro/astro.htm>

[http://www.astrosurf.com/buil/iris/tutorial3/doc13\\_us.htm](http://www.astrosurf.com/buil/iris/tutorial3/doc13_us.htm)

[http://www.astrosurf.com/buil/iris/nav\\_pane/commands\\_iris\\_v5.0.html#data2image](http://www.astrosurf.com/buil/iris/nav_pane/commands_iris_v5.0.html#data2image)

[http://www.astrosurf.com/buil/iris/tutorial19/doc42\\_us.htm](http://www.astrosurf.com/buil/iris/tutorial19/doc42_us.htm)

<http://astrosurf.com/vdesnoux/virtualeq/virtualeq.htm>

[http://www.astrosurf.com/buil/iris/new354\\_us.htm](http://www.astrosurf.com/buil/iris/new354_us.htm)

[http://www.astrosurf.com/buil/iris/new360\\_us.htm](http://www.astrosurf.com/buil/iris/new360_us.htm)

[http://www.astrosurf.com/buil/iris/new370/new370\\_us.htm](http://www.astrosurf.com/buil/iris/new370/new370_us.htm)

[http://www.astrosurf.com/buil/iris/new380/new380\\_us.htm](http://www.astrosurf.com/buil/iris/new380/new380_us.htm)

[http://www.astrosurf.com/buil/iris/new382/new382\\_us.htm](http://www.astrosurf.com/buil/iris/new382/new382_us.htm)

[http://www.astrosurf.com/buil/iris/new383/new383\\_us.htm](http://www.astrosurf.com/buil/iris/new383/new383_us.htm)

[http://www.astrosurf.com/buil/iris/new400/new400\\_us.htm](http://www.astrosurf.com/buil/iris/new400/new400_us.htm)

[http://astrosurf.com/buil/iris/new410/new410\\_us.htm](http://astrosurf.com/buil/iris/new410/new410_us.htm)

[http://astrosurf.com/buil/iris/new414/new414\\_us.htm](http://astrosurf.com/buil/iris/new414/new414_us.htm)

[http://www.astrosurf.com/buil/iris/new420/new420\\_fr.htm](http://www.astrosurf.com/buil/iris/new420/new420_fr.htm) pol2rec, rec2pol, QM QR SQR SQR2

[http://www.astrosurf.com/buil/iris/new430/new430\\_fr.htm](http://www.astrosurf.com/buil/iris/new430/new430_fr.htm)

[http://www.astrosurf.com/buil/iris/new437/new437\\_us.htm](http://www.astrosurf.com/buil/iris/new437/new437_us.htm) mainly gradient removal procedure

[http://astrosurf.com/buil/iris/new520/new520\\_us.htm](http://astrosurf.com/buil/iris/new520/new520_us.htm)

[http://astrosurf.com/buil/iris/new523/new523\\_us.htm](http://astrosurf.com/buil/iris/new523/new523_us.htm)

[http://astrosurf.com/buil/iris/new525/new525\\_us.htm](http://astrosurf.com/buil/iris/new525/new525_us.htm)

[http://www.astrosurf.com/buil/iris/new530/new530\\_us.htm](http://www.astrosurf.com/buil/iris/new530/new530_us.htm)

[http://www.astrosurf.com/buil/iris/new533/new533\\_us.htm](http://www.astrosurf.com/buil/iris/new533/new533_us.htm)

[http://www.astrosurf.com/buil/iris/new540/new540\\_us.htm](http://www.astrosurf.com/buil/iris/new540/new540_us.htm)

[http://astrosurf.com/buil/iris/new550/new550\\_us.htm](http://astrosurf.com/buil/iris/new550/new550_us.htm)

[http://astrosurf.com/buil/iris/new550/new5502\\_us.htm](http://astrosurf.com/buil/iris/new550/new5502_us.htm)

[http://astrosurf.com/buil/iris/new552/new552\\_us.htm](http://astrosurf.com/buil/iris/new552/new552_us.htm)

[http://astrosurf.com/buil/iris/new552/survey\\_us.pdf](http://astrosurf.com/buil/iris/new552/survey_us.pdf)

[http://www.astrosurf.com/buil/iris/new553/new553\\_us.htm](http://www.astrosurf.com/buil/iris/new553/new553_us.htm) coastro

[http://www.astrosurf.com/buil/us/spectro8/spaude32\\_us.htm](http://www.astrosurf.com/buil/us/spectro8/spaude32_us.htm) about spectra processing

[http://www.astrosurf.com/buil/iris/lecon25\\_us.htm](http://www.astrosurf.com/buil/iris/lecon25_us.htm) about audline cameras and progressive scanning techniques.

<http://www.astrosurf.com/rondi/obs/shg/spectroheliogramme.htm>

<http://www.astrosurf.com/buil/us/iris/fringe/fringe.htm>

[http://www.astrosurf.com/aras/spiris\\_en/spiris\\_en1.htm](http://www.astrosurf.com/aras/spiris_en/spiris_en1.htm) how to process spectra

<http://astrosurf.com/buil/us/iris/astro/astro.htm> photometry