Photoshop FITS Liberator Plug-in v.2.0 Specifications

By
Lars Lindberg Christensen, ESA/ESO/HST (lars@eso.org)
Robert Hurt, IPAC/SIRTF
Lars Holm Nielsen
Zolt Levay, STScI/OPO/HST
Kaspar Kirstein Nielsen
Teis Johansen

Bugfixes etc are not included here and can be monitored on http://fits.hankat.dk . Write to lars@eso.org for access.

Core functionality

These elements represent the minimum functionality that would be needed for v.2.0 of the ESA/ESO/NASA Photoshop FITS Liberator.

1. **Retain the previous import settings** if the same file as last time is read in again (e.g. make it possible to repeat the import and easily tweak the settings if you have to do it again). These parameters should somehow be written to disk, to also retain the import settings even if Photoshop is restarted.

   There are two types of settings:
   1. Global User Settings: Preview on/off, mark pixels in preview, Y-axis scale in histogram and freeze (see next point)

   2. FITS File Settings: For an individual image/plane in a FITS file – background_level, white/black level, import channels, null values, stretch function. For an individual FITS File we also retain the selected image and plane.

   We also need a **reset button** that will reset the FITS File Settings (means that we will make a new default guess).
2. **Freeze settings:** Have an option to retain the settings regardless of which files are opened. This would probably be a checkbox saying something like "Freeze settings". If the checkbox is check-marked, then the next time you try to load a file, the initial settings will be the settings that were frozen, and checkbox will be check-marked. The reset button should probably be disabled, or if you try to click at it, you will get an error message saying something like "Freeze settings is currently on. You must disable this feature to be able to reset the settings."
3. **Retain the settings of one stretch function, when changing to another**, and then changing back to the original stretch function. (e.g. retain Lin settings, when testing log, and the returning to Lin).
   Should probably be solved with some sort of caching. This would make it easy to quickly blink between to e.g. log and lin stretches.
4. **Option to set undefined values and null values to 0 (black) or "transparent"**. Null values, are values that a blank in the FITS file. Undefined values are values that become undefined e.g. due to a log stretch.

It will not be possible to set the null and undefined values independently of each other. So either are both black or both transparent.

5. **Mouse cursor and toolbox controls**. By default when you are moving the mouse over the preview image, the cursor should change to crosshair to allow more precise readings of pixel values. By pressing space down, the cursor will change to a hand to allow moving the preview around. To make a more Photoshop-like solution, we should have a toolbox with the three tools mentioned below (three icons somewhere in the GUI).

   1. Crosshair (eyedropper): Reading out pixel values.

   2. Hand (move): Moving the preview around.

   3. Magnifier glass: Click once to zoom in. Holding down the mouse button while dragging, would make you select the area to zoom in on. Zoom buttons should still be available in the bottom of the preview. Holding down e.g. Ctrl and clicking once will zoom out.

   Each tool would have a common drop-down menu with the following options: Fit to preview, Zoom to 100%, Zoom in, Zoom out, Centre preview.

   Furthermore we need two **eyedroppers for inserting pixel values into the black/white level text fields**. This will be two icons next to the black/white level text fields.
6. **Zooming:** Larger zoom span (minus to plus) and better handling of large zooms (faster).
7. **Be able to browse FITS header:** Read the header info and make it possible to browse it, in a separate tab behind the main window (tab number 3).
8. **Extracting relevant header info and write Virtual Repository metadata.** Extraction of relevant header information into a short list of standardised keywords (TBC) and show it in a separate tab behind the main window (tab number 2). User can edit most fields. Some fields will be won't be editable as they are updated by the Liberator itself (e.g. black/white level). This metadata should be saved with the tiffs, jpegs etc. and will therefore – In a future vision – be searchable by some third party tools (provided that a central (IAU) repository delivers a) the list of standardised keywords, b) the list of image archives signed up, c) operational instructions for the tools/robots for how to query the individual archives).
   Metadata keywords:
   - **FILE:**
     1. Product type ("image") [image/video/text]***
     2. File format ("tiff") [tiff/jpeg/avi/mpeg-2/]
     3. Original dimensions (=NAXIS1/2) ("2100 x 2304 pixels")*
   - **ID:**
     4. IDs (e.g. "heic0412a, opo0420b")
     5. Data provider ("41: Hubble European Space Agency Information Centre")**
     6. Observatory ("1: Hubble Space Telescope")**
     7. Instrument ("WFPC2")**
     8. Dataset names (VO compliant if possible): ("ivo://ESO.HST/U2JZ0607B, ivo://ESO.HST/U2JZ0603B, ivo://ESO.HST/ U2JZ0607B, ivo://ESO.HST/U2JZ0605B") [ivo://AuthorityID/ResourceKey]

9. Image release date ("02.01.1995")
10. Author ("Lars Lindberg Christensen")**
11. Credit ("ESA & NASA")**

- **PROCESSING:**

12. White level (z1)*
13. Black level (z2)*
14. Stretch function*
15. Scale factor*
16. Offset*

- **INFO:**

17. Quality ("2")
18. Further information link ("http://hubblesite.org/newscenter/newsdesk/archive/releases/1995/45/image/a")**
19. Comment ("This spectacular color panorama of the center the Orion nebula is one of the largest pictures ever assembled from individual images taken with the Hubble Space Telescope. The picture, seamlessly composited from a mosaic of 15 separate fields, covers an area of sky about five percent the area covered by the full Moon.")

- **ASTRO:**

20. Wavelength range ("502-658 nm")****
21. Centre coordinate (ra, dec, Epoch 2000) "(04 12 10, -05 04 30)"
22. Corner coordinates (ra, dec, Epoch 2000) "(04 12 12, -05 04 32) (04 12 04, -05 04 32) (04 12 10, -05 07 32) (04 12 04, -05 07 32)" ****
23. Creation type ("real") [real/simulated/artwork]***
24. Target name ("M 42")
25. Other Number of exposures ("4") ****
26. Exposure times in seconds ("320, 300, 700, 900") ****
27. Object class/subclass ("nebula")("emission"):***
    - Solar System
        - Venus
        - Mars
        - Jupiter
        - Saturn
        - Uranus
        - Neptune
        - Pluto
        - Planetary Moon
        - Planetary Ring
        - Weather / Atmosphere
        - Minor Body
        - Asteroid
        - Comet
        - Kuiper Belt Object
    - Star
        - Binary Star
        - Brown Dwarf
        - Constellation
        - Massive Star
        - Neutron Star
        - Nova

        - Protostellar Jet
        - Protoplanetary Disk
        - Pulsar
        - Star Field
        - Star with Planet
        - Supernova
        - Variable Star
        - White Dwarf
    - Star Cluster
        - Globular
        - Open
    - Nebula
        - Dark
        - Emission
        - Planetary
        - Reflection
        - Supernova Remnant
    - Galaxy
        - Cluster
        - Dwarf
        - Elliptical
        - Interacting
        - Irregular
        - Magellanic Cloud
        - Quasar / Active Nucleus
        - Spiral
    - Exotic
        - Black Hole
        - Dark Matter
        - Gamma Ray Burst
        - Gravitational Lens
    - Cosmology
        - Distant Galaxies
        - Intergalactic Gas
        - Universe: Age / Size
    - Facilities
    - Miscellaneous

* Provided by Liberator & is not editable.
** Will be taken from last file as a default, but is editable.
*** Drop-down box.
**** Provided from Liberator from the header

9. **Help bubbles**.
10. **Separation of text and plug-in code:** Makes it possible to release localized versions.
11. **Code:** Optimise the code and make a better code design.
12. **Arrow decrements** (1/10 of a pixel + shift -> one pixel). Investigate if it can be solved this time (didn't make it to version 1).
13. **Better installation:** E.g. get 7.0 and Elements installation file paths correct and improvements of the OS X installation.
14. **Calculation of median**.

15. **Replace the "a" constant with two new optional parameters**: background_level (default = 0), scale (default = 1.0) where image_scaled = scale * (image - background_level).
16. **Make automated guesses for the values of scale and background_level:** We need to check the feasibility of this: E.g. background_level = [the pixel value at the peak of the histogram], and scale = [65565/the pixel value at 1/100 of the peak height of the histogram]. Or for instance background_level = min, scale = 65565/max. The user should be able to change these at will. Possibly the user should have a tool to calculate background_level the mean of a 100 pixel aperture by clicking once at the background.
17. **New formulae for stretch functions**

    - Linear = image_scaled
    - Log = log[image_scaled + 1]
    - Sqrt = sqrt[image_scaled]
    - Asinh = asinh[image_scaled]
    - Cube root = cube_root[image_scaled]
    - LogLog = log[scale2 * log[image_scaled + 1] + 1] (scale2 to be hard-coded to a useful value to be determined during development)
    - SqrtLog = sqrt[log[image_scaled + 1]]
18. **Fix file name restriction** with () [] or {} (contact CFITSIO development, NASA, Bill Pence).
19. **Show the values for both *image* and *image_scaled***, when reading out image statistics in the preview window.
20. **Open document title:** The open document in Photoshop should have a title that includes the plane and image name. It will also allow you to open the same file more than once.
21. **"Enter" key problem (known bug):** When you press “enter” in e.g. a text field, it shouldn’t load the image file. This is problematic due to ADM’s odd behaviour.
22. **Spectra:** Be able to open and render spectral information.
23. **Clipping:** If a pixel is clipped, then render all interpolated pixels containing this pixel in the zoom window as clipped (show it also as clipped even though the average may not be clipped). Or, if any of the pixels that went into the calculation of this pixel are clipped, clip this pixel.

Value-Added Features

The following elements represent very useful functionalities but may entail programming overheads that may make some number of them impractical to implement. They are ordered roughly in increasing complexity.

1. **Batch operation:** To be able to use the Liberator in unattended or non-interactive mode. This can be done either (simple) by you setting some default values or (advanced) by the user calling the Liberator (from where?) with values for z1, z2, stretch etc.
2. **PaintShop Pro version** of the plug-in if feasible. PSP can probably only use Filter plug-ins.
3. **Zooming on the histogram window**. There are two different ways of looking at this. One is to manually type in the lower and upper cuts, meaning that you type in the actual range of the histogram you want to se and then it scales to that (or some nearby rounded values). The other is some kind of

percent marker much like zooming an image. This should probably be a magnifying glass mouse pointer. Both ways could be implemented at the same time. The latter could be interpreted as a more graphical and automated tool, where as the former method is a more precise and maybe a little less user-friendly one. It should also be obvious that the y-axis should be invariant to range transformations since differently scaled areas of the same histogram would confuse people more than help them, e.g. we preserve direct comparison capability.

4. (IF keyword COLPLANE = RGB THEN interpret FITS image as a colour RGB and show this in PS) (TBD – as help for MaxIm DL users)

Timeline

15.03.05	Beta 1 release
01.06.05	Beta 2 release
20.06.05	Beta 3 release
25.06.05	Beta 4 release
30.06.05	Release candidate 1
01.07.05	v.2 Final Release

Updated 10/05/2005

Appendix A: FITS Liberator 2.0 Design: Revisions to Stretch Function Handling

NOTE: The term “zero_level” used below has been replaced by the word “background_level” above.

Robert Hurt, 15 July 2004

The Issue

Due to the complexities of applying image transfer functions, the current 1.0 implementation that employs an additive offset “a” results in inconsistent results between transfer functions and is somewhat non-intuitive in its application. Moreover, there are some issues with how log transforms are handled that could be improved.

Uniform Treatment of offsets/scaling

My key recommendation is to replace the "a" constant with two new optional parameters:

_ zero_level (default = 0)

_ scale (default = 1.0)

where

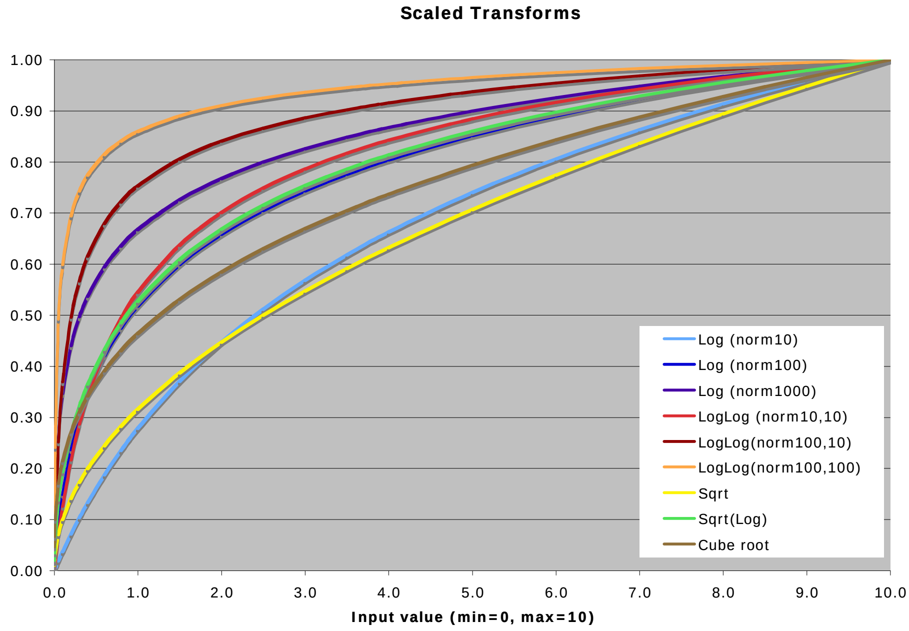
_ image_scaled = scale * (image - zero_level)

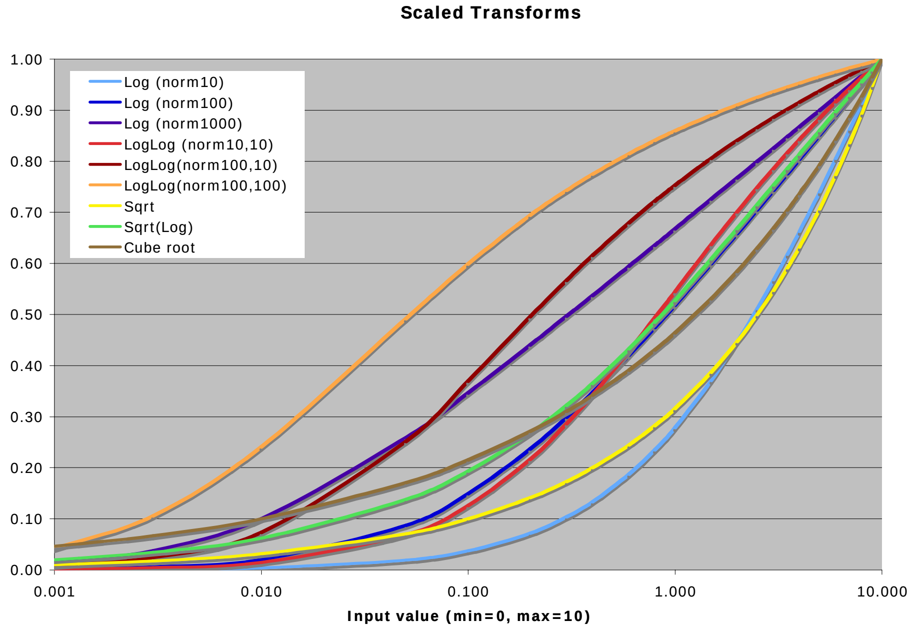
This transformation should be applied in all stretch functions. This serves several purposes. Most importantly the inclusion of a “zero_level” turns the abstract “a” offset into a physically meaningful number (the data value that represents the zero level of the image), something that is critical for properly applying the stretch functions that are undefined for negative values and it is important to define the zero level independent of where the black level may be set on the histogram.

The purpose of the scale is twofold. First, it can make it easier to work with datasets with extremely large or small data values. This had been a special problem with, for instance, the MSX FITS files. By applying an appropriate scale factor, the value readouts can be shifted to more easily handled ranges. Second, the scale can serve a very useful purpose for log transformations.

Stretch Functions

I’ve done a little research into the effective results of various stretch functions. The basic need for these is to meaningfully compress dynamic range in the final image. As such, one can effectively characterize them by how they behave at the extreme ends of the data ranges. Below are two plots I’ve compiled to demonstrate this point. For each curve the input ranges run from 0 to 10 (displayed on linear and log axes) and the resulting stretch has been normalized to run between 0 and 1. The log stretches are a bit more involved and will be discussed below.





Square & Cube Root Stretches

The behavior of the root stretches shows a useful property of increasing rapidly at the faint end. This early jump is clearly useful for bringing out very faint structures in images with extreme dynamic ranges. The cube root is noticeably more extreme than the square root and could be a useful option to add.

Log Stretches

Handling log stretches is intrinsically more difficult than for square roots because of turning undefined for negative values, but also being open ended on both the small and large number sides.

As a case study, I checked the code of the IPAC package “skyview” which was developed specifically for the IRAS datasets. The developer in this case decided to produce a uniform result by taking the user’s min and max data values, then taking the log of the image data normalized to run between 1 (for min) and 10 (for max). Thus the restretched values fall between 0 and 1. In the charts above, this is represented by the “Log(norm10)” lines. For the more extreme loglog stretch, the 0 to 1 values are again renormalized to run from 1 to 10, and is presented as “LogLog(norm10,10)” in the charts.

The choice to rescale on a 1 to 10 scale is, however, arbitrary. To explore other behaviors I also computed logs where the data values were scaled to run from 1 to 100 (norm100) and from 1 to 1000 (norm1000). Obviously the greater the dynamic range, the more the curve shifts to the left, representing increasingly strong compressions of dynamic range.

For comparison I also computed a curve for which the inner and outer logs were both scaled to run from 1 to 100 (norm100,100), producing the strongest dynamic range compression of all.

Clearly the overall scale of the data are significant. I do note that in principle one does not need both an offset and scale to reproduce an arbitrary segment of the log curve. The scale factor could be dropped out and the curve reproduced by shifting the black and white levels on the histogram, but this requires rather non-intuitive values of the offset “a”. However, for the double-log, the current 1.0 implementation is impaired, lacking any form of offset and/or scale term.

Handling Log Zero Point

In fact, for the log transformation, the scale factor is mathematically redundant since the log of a product is the sum of the logs. Since log[s(x – zl)] = log[s] + log[x – zl/s], the scale factor really just shows up as an offset to the log. Understanding how to correctly choose the zp and scale values is somewhat opaque and obtaining consistent dynamic range compression is problematic due to the unbounded negative side of the log transform.

I would propose that in some sense the above handling of logs in “skyview” is in part a useful model for the FITS Liberator. Our cases are different since skyview uses only a min and max level and computes the log transform based on these cuts, while it is far more efficient for the Liberator to calculate the log transform once and let user set the black and white levels interactively. However, this means that these levels are not mathematically related to the skyview min/max.

If we adopt the scale & zero_level offsets for all transfers, then in principle the user should have a properly zeroed dataset after the image_scaled = scale * (image - zero_level) calculation. As such, I think I like the idea of taking the log of [image_scaled + 1], as that will return a stretched value of 0 at the point where the image should in principle be black. Thus the histogram black level for log should nominally be 0, which is a lot more intuitive than fishing around for some arbitrary negative level that represents where the data should turn black.

This also allows the user to be instructed in a relatively simple manner how to effectively use the scale factor to help them get reproducible dynamic range compressions across different datasets. In effect we can suggest scale be set so that the brightest desired features in the image be at least 10. By increasing the scale to levels of 100, 1000, or more, the stretch favors the fainter levels at the expense of the brighter ones. If they apply appropriate scales for datasets with wildly different levels such that they each peak around the same level, then they'll get comparable dynamic range compression, which may be very valuable in preserving multiband color combinations.

If the user sets the peak displayed values of their scaled data to reach levels of 10, 100, and 1000, then they will correspond to the scaled single log curves in the figures above.

For the log-log stretch, I would propose that the easiest thing to do would be to hide a lot of these issues from the user and come up with a default rescaling for the dataset that seems to work best for typical images. Thus the outer log would be something like log[scale2 * (inner log) + 1], again with the +1 offset so the zero level comes out zero in the histogram. If the scaling advice suggested above is followed, then we know the range of values of interest from the inner log will be something like 0 to ~2±1. I suspect a good value of scale2 would be somewhere in the range of 1 to 10.

Stretch Comparisons

To appreciate the effects of the different stretches one can most easily compare behaviors at the bright end on the top chart, and at the faint end on the bottom chart. Each stretch function does show potentially useful properties.

The root functions appear to be the best choices for very faint structures of interest in fields where it is OK to burn in the brightest features more quickly. The single logs sacrifice the faintest structure in favor of preserving dynamic range at the bright end.

The double logs preserve more of the high end dynamic range, but in principle are very similar to single logs applied to a larger range, and the same is largely true for the sqrt log. In fact, for all practical purposes, the sqrt(log(norm10), the loglog(norm10,10), and the log(norm100) curves are practically the same and would be hard to distinguish by eye.

There is clearly some level of redundancy among some of these curves but since they're basically in place I don't see a strong need to remove any because of this. But the cube root does show a much more distinct behavior and would be worth adding.

Stretch Recommendations for 2.0

Following the issues raised in the document, I suggest the following revisions to the handling of image stretch functions.

1. Replace “a” constant with *zero_level* and *scale* values that are universally applied to all stretches (including linear):
   image_scaled = scale * (image - zero_level)

2. In the preview window, when reading out image statistics, show the values for both *image* and *image_scaled*.

3. Use the following formulae for stretch functions

Linear = image_scaled

Log = log[image_scaled + 1]

Sqrt = sqrt[image_scaled]

Cube root = cube_root[image_scaled]

LogLog = log[scale2 * log[image_scaled + 1] + 1] (scale2 to be hard-coded to a useful value to be determined during development)

SqrtLog = sqrt[log[image_scaled + 1]]

APPENDIX B

```
SIMPLE  =                    T          / Standard FITS format (NOST-100.0)
BITPIX  =                   16          / # of bits storing pix values
NAXIS   =                    2          / # of axes in data array
NAXIS1  =                 2148          / # pixels/axis
NAXIS2  =                 4096          / # pixels/axis
ORIGIN  = 'ESO     '                    / European Southern Observatory
DATE    = '2004-06-18T13:40:57.331'     / Date this file was written
MJD-OBS =       52760.02516459          / MJD start (2003-05-01T00:36:14.221)
DATE-OBS= '2003-05-01T00:36:14.220'     / Date of observation
EXPTIME =            1799.9930          / Total integration time
TELESCOP= 'ESO-VLT-U3'                  / ESO Telescope Name
RA      =           146.587200          / 09:46:20.9 RA (J2000) pointing
DEC     =            -24.48429          / -24:29:03.4 DEC (J2000) pointing
EQUINOX =                2000.          / Standard FK5 (years)
RADECSYS= 'FK5     '                    / Coordinate reference frame
LST     =            37722.528          / 10:28:42.528 LST
UTC     =             2171.000          / 00:36:11.000 UTC
OBSERVER= 'VIMOS   '                    / Name of observer
INSTRUME= 'VIMOS   '                    / Instrument used
PI-COI  = '555555555'                   / Name of PI and COI
OBJECT  = '0943-242'                    / Target description
PCOUNT  =                    0          / Number of parameters per group
GCOUNT  =                    1          / Number of groups
CRVAL1  =              1.00000          / 00:04:00.0, RA at ref pixel
CRPIX1  =        -1997.0000000          / Reference pixel in X
CDELT1  =           1.00000000          / SS arcsec per pixel in RA
CTYPE1  = 'PIXEL   '                    / pixel coordinate system
CRVAL2  =              1.00000          / 01:00:00.0, DEC at ref pixel
CRPIX2  =            1.0000000          / Reference pixel in Y
CDELT2  =           1.00000000          / SS arcsec per pixel in DEC
CTYPE2  = 'PIXEL   '                    / pixel coordinate system
BSCALE  =                  1.0          / pixel=FITS*BSCALE+BZERO
BZERO   =              32768.0          / pixel=FITS*BSCALE+BZERO
ORIGFILE= 'VIMOS_IFU_OBS121_0002_A.3.fits' / Original File Name
ARCFILE = 'VIMOS.2003-05-01T00:36:14.223.fits' / Archive File Name
CHECKSUM= '35Ae323d32Ad323d'            / ASCII 1's complement checksum
UT      = '00:36:11.000'                / UT at start
ST      = '10:28:42.528'                / ST at start
AIRMASS =              1.01400          / Averaged air mass
IMAGETYP= 'OBJECT  '                    / Observation type
FILTER3 = 'OS-blue '                    / Filter 3 name
```