Package 'sunxplrr'

February 4, 2020

Type Package

Title Tools for exploring Digital Images of the Sun

Version 0.1.0

Date 2020-01-31

Author Thomas k. Friedli <thomas.k.friedli@bluewin.ch>

Maintainer Thomas K. Friedli <thomas.k.friedli@bluewin.ch>

Description Tools for solar image analysis, feature extraction, indices calculation and visualisation.

Depends dplyr (>= 0.7), tidyr

Imports ggplot2

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

`fun_calc_mode` *Calculates the mode of the provided data vector*

Description

Calculates the mode (most frequent value) of the provided data vector.

Usage

```
fun_calc_mode(x)
```

Arguments

x	data vector.

Value

mode value.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_center_image` *Centers the provided image*

Description

Centers the provided image by shifting the x-axis and the y-axis of the image appropriately. Do not center sdo images, since they are centered already. The algorithm is slightly adapted from Berry, R. and Burnell, J.: The Handbook of Astronomical Image Processing, 2nd edition 2005, p. 323.

Usage

```
fun_center_image(image, hdrlst, header, sdo.image = "FALSE")
```

Arguments

`image`	tibble image with i and j pixel coordinates and pixel values x. Additional columns are ignored and will be not returned.
`hdrlst`	list containing image FITS header keywords and values.
`sdo.image`	boolean switch for dummy use in the case of non sdo calcium images.

Value

tibble with centered image.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_clv_calibrate` *Calibrate the image*

Description

Calibrate the image by adding the rescaled fitted degree-5 clv function to the residuals from the multi-parameter fit of the image.

Usage

```
fun_clv_calibrate(x, sdo.image = "FALSE")
```

Arguments

`x`	tibble containing columns with pixel coordinates i and j, pixel values x from the original image, fill from the disc mask, sclv and residuals from the sclv image.
`sdo.image`	boolean switch for dummy use in the case of non sdo calcium images.

Value

tibble containing the calibrated image values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_clv_correct` *Flattens an image*

Description

Flattens a solar image according to a provided clv function.

Usage

```
fun_clv_correct(x, name.image = "calib", name.clv = "sclv")
```

Arguments

`x`	tibble containing columns with pixel coordinates i and j and columns with the pixel values of the image and the clv function.
`name.image`	name of the column containing the image pixel values.
`name.clv`	name of the column containing the clv function values.

Value

tibble with additional column to x containing the flattened image.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_clv_fitting` *Fits an up to eight-parameter center-to-limb-variation function*

Description

Fits a six parameter polynom to the clv function. On request, a 2 parameter plane may be fitted trough the center of the image disc. Dark sunspots and bright faculae may be eliminated by up to two elimination runs, where with the first iteration run all 99 second iteration run all 95

Usage

```
fun_clv_fitting(x, hdrlst, model = "poly_with_plane", run = 3,
  clip.resid.out = "FALSE", light.save = "FALSE")
```

Arguments

x	tibble containing columns with pixel coordinates i and j, pixel values x from the original image, border and fill from the disc mask, image from the extracted image disc, pixel coordinates xi and yj measured from the estimated disc center and the heliographic coordinates theta, phi, L and B.
hdrlst	list containing image FITS header keywords and values.
model	implemented are "poly" for polynomial fit with 6 coefficients, "poly_with_plane" for polynomial fit with 6 coefficients and an additional 2 parameter plane through the disc center and a third model named as "poly_with_plane_and nonparametric background" for polynomial fit with 6 coefficients, an additional 2 parameter plane and a nonparametric surface fit in a gam structure.
clip.resid.out	if "TRUE" the clipped residuals from the last fit iteration are returned. In batch mode this output is not needed.
light.save	if TRUE only a small selection of csv files are saved.
clip.resid.out	if "TRUE" the clipped residuals from the fit iterations are returned. In batch mode this output is not needed.

Value

list containing the tibble z with the fitted model, the clv function and the residuals, two vectors clv.coeff and fit.coeff containing the fitted coefficients of the fitted model and of the fitted clv function and two values with the stdev of the fit and the clip value for low and high signals, if applied.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_clv_function *Calculates the standardized center-to-limb-variation function*

Description

Calculates the center-to-limb-variation function according to formulae published by different authors, including Allen in Cox (2000), Pierce & Slaughter (1977) and Neckel & Labs (1994). Furthermore, a customly fitted clv may be calculated. Note, that this is the only available option for SDO images, since near the temperature minimum, a near flat clv is expected but the SDO images show a clear limb darkening.

Usage

```
fun_clv_function(x, clv.i0, sclv.method = "NL", sdo.image = "FALSE",
  clv.coeff = NULL)
```

Arguments

x	tibble containing columns with pixel coordinates i and j, pixel values x, pixel coordinates xi, yj and angle theta from an image disc.
clv.i0	central intensity of the fitted clv function of the image.
sdo.image	boolean switch for dummy use in the case of non sdo calcium images.
clv.coeff	vector with 6 coefficients from the fitted clv function.
method	implemented are "Allen", "PS", "NL" and "fit".

Value

z tibble with additional column to x containing the standardized clv function and the cf vector with coefficients of the standardized clv function.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_clv_rmplane *Subtracts plane from image*

Description

Subtracts a previously fitted plane from the provided image values.

Usage

```
fun_clv_rmplane(x, hdrlst, coeff)
```

Arguments

x	tibble containing columns with pixel coordinates i and j, pixel values x from the original image, fill from the disc mask, image from the extracted image disc and theta.
hdrlst	list containing image FITS header keywords and values.
coeff	containing the previously fitted coefficients of an eight-parameter fit function.

Value

tibble containing the corrected image values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_date2jd *Calculates Julian Date from a vector containing date with time objects*

Description

Calculates Julian Date from a vector containing date with time objects. Does not work for negative years. Requires 4 digits for year.

Usage

```
fun_date2jd(date_time)
```

Arguments

date_time	vector containing date and time in FITS format.

Value

vector with Julian Dates.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_date_grid *Provides a tibble with dates from start to end*

Description

Privides a tibble with dates and columns containing year, month and day. The granularity can be chosen from year month or day. Works for all years within the gregorian calendar.

Usage

```
fun_date_grid(start_date, end_date, granularity = "day")
```

Arguments

start_date	First date of grid.
end_date	Last date of grid.
granularity	Implemented are "year", "month" and "day".

Value

tibble containing date, and depending on the granularity also year, month and day.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_deltaT *Calculates delta T in seconds according to formulae by Fred Espenak*

Description

Calculates delta T = TD - UT in seconds according to formulae given by Fred Espenak. Before 2005, the polynomial expressions for delta T are from the Five Millennium Canon of Solar Eclipses: -1999 to +3000. Source: http://www.eclipsewise.com/help/deltatpoly2014.html

Usage

```
fun_deltaT(date_time)
```

Arguments

date_time	vector containing date and time in FITS format.

Value

value of delta T in seconds.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_disc_center` *Calculates disc center coordinates and radius of the indicator disc*

Description

Calculates disc center coordinates and the radius for a given disc indicator.

Usage

```
fun_disc_center(x, disc = "fill")
```

Arguments

`x`	tibble containing at least 3 columns wih pixel coordinates i and j and disc indicator values disk.
`disc`	name of variable in x with disc indicator values.

Value

tibble with intensity center coordinates x_i for image columns and y_j for image rows and radius of the disc indicator circle.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_disc_coordinates` *Determines heliografic coordinates*

Description

Determines heliografic coodinates, including the pixel coordinates from disc center, the angle theta, measured in degrees from the disc center to each disc pixel. For limb pixels theta = 90°. Provided are also the radius r, the angle phi, the heliografic latitude B, the heliografic longitude l and L as the Carrington rotation number for each pixel.

Usage

```
fun_disc_coordinates(x, hdrlst)
```

Arguments

x	tibble containing columns with pixel coordinates i and j and additional columns which were all passed to the output tibble z.
hdrlst	list containing image FITS header keywords and values.

Value

z tibble with additional columns to x containing the heliografic coordinate information.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_disc_grid *Calculates grid of heliografic longitudes and latitudes*

Description

Calculates positions of grid pixels for heliografic longitudes and latitudes in intervals of 10 degrees.

Usage

```
fun_disc_grid(x, hdrlst, grid_each_deg = 10,
  res_each_deg_on_grid = 0.01)
```

Arguments

x	tibble containing columns with pixel coordinates i and j and additional columns which were all passed to the output tibble z.
hdrlst	list containing image FITS header keywords and values.
grid_each_deg	num number of degrees between grid lines. Default value is one grid line every 10 heliografic degrees.
res_each_deg_on_grid	num indicates the resolution of points of one degree along the grid lines. Default is 0.01, e.g. 100 points per degree.

Value

z tibble with additional column to x containing the flag for grid pixels.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_disc_math *Implements mathematical image manipulations.*

Description

Implements three mathematical image manipulations, including multiplication, substraction and summation.

Usage

```
fun_disc_math(im1, values_1 = "x", im2, values_2 = "fill",
  method = "mult", values.name = "image")
```

Arguments

im1	tibble containing at least 3 columns wih pixel coordinates i and j and pixel values of first image.
values_1	name of variable in im1 containing the pixel values.
im2	tibble containing at least 3 columns wih pixel coordinates i and j and pixel values of second image.
values_2	name of variable in im2 containing the pixel values.
method	calculation method. Implemented are thre methods: "mult" for image multiplication, "diff" for image substraction and "summ" for image summation.
values.name	name of variable containing the calculated pixel values.

Value

a tibble with additional column containing the resulting image.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_disc_rings *Assigns ring number to disc pixels*

Description

Computes a table with the ring borders given the number of rings and assigns the number of the ring for each disc pixel.

Usage

```
fun_disc_rings(x, num.rings = 50)
```

Arguments

num.rings	number of rings counted from limb to disc center.

Value

z tibble with additional column to x containing the ring number.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_extract_col` *Extracts a single column of image frame*

Description

Extracts a single column of image frame.

Usage

```
fun_extract_col(x, col = NULL)
```

Arguments

`x`	tibble containing 3 columns wih pixel coordinates i and j and pixel intensity values x.
`row`	int index of column to be extracted.

Value

tibble containing 2 columns wih row numbers j and pixel intensity values x.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_extract_row` *Extracts a single row of image frame*

Description

Extracts a single row of image frame.

Usage

```
fun_extract_row(x, row = NULL)
```

Arguments

`x`	tibble containing 3 columns wih pixel coordinates i and j and pixel intensity values x.
`row`	int index of row to be extracted.

Value

tibble containing 2 columns wih column numbers i and pixel intensity values x.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_flip_image` *Flips the provided image*

Description

Flips the provided image by reversing the y-axis of the image (interchanging up and down). Do not flip for sdo images. Algorithm from Berry, R. and Burnell, J.: The Handbook of Astronomical Image Processing, 2nd edition 2005, p. 331f.

Usage

```
fun_flip_image(image, hdrlst, header, sdo.image = "FALSE")
```

Arguments

`image`	tibble image with i and j pixel coordinates and pixel values x.
`hdrlst`	list containing image FITS header keywords and values.
`header`	list containing image FITS header.
`sdo.image`	boolean switch for dummy use in the case of non sdo calcium images.

Value

tibble with flipped image, updated hdrlst and header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_flop_image` *Flops the provided image*

Description

Flops the provided image by reversing the x-axis of the image (interchanging left and right side). Do not flop for sdo images, which are oriented east to the left and north to the top. The algorithm is from Berry, R. and Burnell, J.: The Handbook of Astronomical Image Processing, 2nd edition 2005, p. 332.

Usage

```
fun_flop_image(image, hdrlst, header, sdo.image = "FALSE")
```

Arguments

image	tibble image with i and j pixel coordinates and pixel values x.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.
sdo.image	boolean switch for dummy use in the case of non sdo calcium images.

Value

tibble with flopped image, updated hdrlst and header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_gather_csv *Gathers output files of an index*

Description

Gathers all output files of an index within an out_data_path.

Usage

```
fun_gather_csv(out_data_path, index.name = "_total_indices")
```

Arguments

out_data_path	full path to output directory.
index.name	character string with full index name as contained in the file name after the image id and before the file extension.

Value

tibble containing the calibrated image values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_grid_plt *Adds grid to an image*

Description

Adds grid to an image, both provided by the same tibble.

Usage

```
fun_grid_plt(x, hdrlst)
```

Arguments

x	tibble containing columns with pixel coordinates i and j, pixel values x from the image and a binarised mask with the matching grid.
hdrlst	list containing image FITS header keywords and values.
theta	angle measured in degrees from the disc center to the limb.
num.rings	number of rings counted from limb to disc center.

Value

tibble with additional column containing the image with ring borders.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_hdr2list *Extracts image header to list*

Description

Extracts the image header to a list of keywords with values. COMMENT and HISTORY lines are omitted.

Usage

```
fun_hdr2list(h)
```

Arguments

h	FITS hdr from FITSio::readFITS(image)

Value

list containing FITS header keywords and corresponding values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_jd2date *Calculates date and time vector from Julian Dates vector*

Description

Calculates date and time vector from vector containing Julian Dates objects. Does not work for negative years.

Usage

```
fun_jd2date(JD)
```

Value

vector containing date with time in FITS format.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_mask_border *Marks the border pixels of a given mask*

Description

Marks the border pixels of a given mask.

Usage

```
fun_mask_border(x, mask = "th")
```

Arguments

x	tibble containing columns with pixel coordinates i and j and mask.
mask	name of the column in x containing the mask.

Value

z tibble with additional column to x containing the border pixels.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_mask_circle *Determines a circle given the center coordinates and the radius*

Description

Determines limb pixels of a circle with given center coordinates and radius and stores them in a mask. Optionally, additional border pixels may be added to the circle radius.

Usage

```
fun_mask_circle(x, disc.center, border.pix = 0)
```

Arguments

x	tibble containing columns with pixel coordinates i and j.
border.pix	number of pixels to be added to the radius of the circle.
disc_center	tibble with center coordinates x_i for image columns and y_j for image rows and radius of the circle.

Value

z tibble with additional column to x containing the limb pixels of the circle.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_mask_create *Marks all pixels above or equal a given threshold*

Description

Calculates binarised pixel mask according to a given thereshold.

Usage

```
fun_mask_create(x, hdrlst, threshold = 10, method = "relative")
```

Arguments

x	tibble containing 3 columns wih pixel coordinates i and j and pixel intensity values x.
hdrlst	tibble containing image FITS header keywords and values.
threshold	cutoff threshold in absolute or relative units.
method	If method = 'absolute' then the cutoff thereshold value is in ADU units. Otherwise it is the threshold / 100 fraction of the maximal ADU value given by 2^bitpix.

Value

list containing tibble with 4 columns for i, j pixel coordinates, intensity value x and binary mask th and tibble with mask.threshold value.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_mask_diff` *Marks all pixels between two given thresholds*

Description

Calculates binarised pixel mask for all pixels above or equal a given lower.threshold but lower than a given upper.threshold.

Usage

```
fun_mask_diff(x, hdrlst, lower.threshold = 5, upper.threshold = 10,
  method = "relative")
```

Arguments

`x`	tibble containing 3 columns wih pixel coordinates i and j and pixel intensity values x.
`hdrlst`	tibble containing image FITS header keywords and values.
`lower.threshold`	cutoff threshold in absolute or relative units.
`upper.threshold`	cutoff threshold in absolute or relative units.
`method`	If method = 'absolute' then the cutoff thereshold value is in ADU units. Otherwise it is the threshold / 100 fraction of the maximal ADU value given by 2^bitpix.

Value

list containing tibble with 4 columns for i, j pixel coordinates, intensity value x, binarised mask th and tibbles with mask.lower.threshold and mask.upper.threshold values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_mask_fill *Fills out a given mask*

Description

Fills out a given mask within their biggest contours.

Usage

```
fun_mask_fill(x, mask = "th")
```

Arguments

x	tibble containing columns with pixel coordinates i and j and mask.
mask	name of the column in x containing the mask.

Value

z tibble with additional column to x containing the filled mask.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_mat2tibbl *Umwandlung einer Matrix in ein tibble*

Description

Wandelt eine Matrix in ein data.frame oder ein tibble um.

Usage

```
fun_mat2tibbl(x, tibbl = TRUE)
```

Arguments

x	matrix with i columns and j rows.
tibbl	boolian If TRUE, the a tibble will be returned, otherwise a data.frame.

Value

a tibble or a data.frame containing 3 columns wih matrix indices i and j and values x.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_parse_header *Parses FITS header*

Description

Parses FITS header. COMMENT, HISTORY, empty and some preserved keywords are omitted.

Usage

```
fun_parse_header(headerName)
```

Value

Character vector with parsed FITS header entries.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_readFITSarray *Reads FITS binary array from disc*

Description

Reads 16 bit or 8 bit FITS binary array from disc.

Usage

```
fun_readFITSarray(zz, hdr)
```

Arguments

zz	binary port to file.
hdr	parsed header keyword and values vector.

Value

list with image matrix and parsed header vector.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_read_header *Reads FITS ASCII header from disc*

Description

Reads FITS ASCII header from disc.

Usage

```
fun_read_header(zz, maxLines = 5000)
```

Arguments

zz	binary port to file.
maxLines	maximal number of header lines to search for END statement. May be increased for very large headers.

Value

Character vector with FITS header card images.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_read_image *Reads FITS image from disc*

Description

Reads 16 bit or 8 bit FITS image from disc.

Usage

```
fun_read_image(filename = "sunviewr.fits", maxLines = 5000)
```

Arguments

maxLines	maximal number of header lines to search for END statement. May be increased for very large headers.
file	output path and file name.

Value

list with image matrix, parsed header and full header vectors.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_ring_get *Returns ring parameters given a theta value*

Description

Computes a table with the ring borders given the number of rings and returns the ring characteristics given a single theta value.

Usage

```
fun_ring_get(theta = 75, num.rings = 50)
```

Arguments

theta	angle measured in degrees from the disc center to the limb.
num.rings	number of rings counted from limb to disc center.

Value

table and a vector with ring characteristic for a given theta value.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

fun_ring_plt *Adds a ring to the image given a theta value*

Description

Computes the ring characteristics given a single theta value and adds the inner and outer ring borders to the image.

Usage

```
fun_ring_plt(x, hdrlst, theta = 75, num.rings = 50)
```

Arguments

x	tibble containing columns with pixel coordinates i and j, pixel values x from the original image, fill from the disc mask, image from the extracted image disc and theta.
hdrlst	list containing image FITS header keywords and values.
theta	angle measured in degrees from the disc center to the limb.
num.rings	number of rings counted from limb to disc center.

Value

tibble with additional column containing the image with ring borders.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_sdo_keywords` *Updates the header and hdrlst with sdo-specific keywords*

Description

Reconstructs the FITS keywords DATE-OBS, NAXISn and WAVELNTH from the provided sdo file name. Updates the header and hdrlst with some other missing keywords.

Usage

```
fun_sdo_keywords(file.name, header, hdrlst, sdo.image = "FALSE")
```

Arguments

`header`	list containing image FITS header.
`hdrlst`	list containing image FITS header keywords and values.
`sdo.image`	boolean switch for dummy use in the case of non sdo calcium images.
`filename`	original file name of sdo image.

Value

list with updated header and hdrlst.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`fun_sun_ephem` *Calculates ephemeris for physical coordinates of the Sun*

Description

Calculates ephemeris for physical coordinates of the Sun, including P0, B0, L0, the Sun's apparent diameter in arcsecs and the Carrington rotation number. Formulae from: Meeus,J., Astronomical Algorithms, Willmann-Bell, 1991.

Usage

```
fun_sun_ephem(hdrlst, sdo.image = "FALSE")
```

Arguments

`hdrlst`	list containing image FITS header keywords and values.
`sdo.image`	boolean switch for case of sdo image.

Value

tibble with P0, B0, L0, the Sun's apparent diameter in arcsecs and the Carrington rotation number.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_tibbl2mat *Umwandlung einer tibble oder eines data.frames in eine Matrix*

Description

Wandelt ein tibble oder ein data.frame in eine Matrix um.

Usage

```
fun_tibbl2mat(x)
```

Arguments

x	tibble with 3 columns for i, j pixel coordinates and intensity value x.

Value

matrix with i columns and j rows.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_write_header *Writes image header*

Description

Writes and closes the image header.

Usage

```
fun_write_header(x, type = "single", bscale = 1, bzero = 0,
  header = "")
```

Arguments

x	image data matrix.
type	implemented are "single" and "byte" for 16 bit and 8 bit images.
bscale	pixel_values = bzero + bscale * FITS_value.
bzero	pixel_values = bzero + bscale * FITS_value.
header	string vector with card images of the existing header and of new or modified card images.

Value

z tibble with additional column to x containing the flattened image.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

fun_write_image *Writes FITS image to disc*

Description

Writes 16 bit or 8 bit FITS image to disc.

Usage

```
fun_write_image(x, file, hdrlst, header)
```

Arguments

x	image data matrix.
file	output path and file name.
hdrlst	tibble with header keywords and values.
header	string vector with card images of the existing header and of new or modified card images.

Value

nothing.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_clv_correction *Fits the clv function, calibrate it and flattens the image*

Description

Fits a six parameter polynom to the limb-variation-function. Additionnally, a plane may be fitted trough the center of the image disc. Removes all fitted non clv variation from the image and flattens it to intensity contrasts. Be aware, that the resulting intensity contrast values should be multiplied with some appropriate factor before converting them back to FITS, since in the FITS image all values are treated as integers.

Usage

```
mod_clv_correction(x, hdrlst, header, model = "poly_with_plane",
  run = 3, clip.resid.out = "FALSE", sclv.method = "NL",
  sdo.image = "FALSE", light.save = "FALSE")
```

Arguments

x	tibble containing columns wih pixel coordinates i and j and all the calculated image information as provided by the disc extraction module.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.
model	implemented are "poly" for polynomial fit with 6 coefficients, "poly_with_plane" for polynomial fit with 6 coefficients and an additional 2 parameter plane through the disc center and a third model named as "poly_with_plane_and nonparametric background" for polynomial fit with 6 coefficients, an additional 2 parameter plane and a nonparametric surface fit in a gam structure.
run	indicates how many iterations for spot and faculae elimination should be run. No more than 3 iterations are implemented.
clip.resid.out	if "TRUE" the clipped residuals from the last fit iteration are returned. In batch mode this output is not needed.
sdo.image	boolean switch for dummy use in the case of non sdo calcium images.
light.save	if TRUE only a small selection of csv files are saved.

Value

tibble containing clv function, calib image and flat image.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_disc_image *Calculates disc center coordinates and radius and extracts the image*

Description

Calculates disc center coordinates and radius and extracts the image. If the original image contains additional text in the image corners, the approximate center coordinates and the approximate radius of the image disc are estimated and the image disk is extracted with an oversized mask. The size of the image and the number of bits per pixel are taken from the FITS header provided by hdrlst.

Usage

```
mod_disc_image(x, hdrlst, header, threshold = 10, method = "relative",
  cut.threshold = 20, cut.method = "relative", cut.border.pix = 100,
  add.border.pix = 0, image.values.name = "image",
  grid_each_deg = 10, res_each_deg_on_grid = 0.01)
```

Arguments

x	tibble containing 3 columns wih pixel coordinates i and j and pixel intensity values x.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.

threshold	cutoff threshold in absolute or relative units.
method	If method = 'absolute' then the cutoff thereshold value is in ADU units. Otherwise it is the threshold / 100 fraction of the maximal ADU value given by 2^bitpix.
cut.threshold	cutoff threshold for images with text in the corners in absolute or relative units.
cut.method	If cut.method = 'absolute' then the cutoff thereshold value for images with text in the corners is in ADU units. Otherwise it is the cut.threshold / 100 fraction of the maximal ADU value given by 2^bitpix.
cut.border.pix	number of pixels to be added to the radius of the circle to extract the image without the text in the corners.
add.border.pix	number of pixels to be added to the radius of the circle to extract the final image.
image.values.name	name of column with image values in final image.
grid_each_deg	num number of degrees between grid lines. Default value is one grid line every 10 heliografic degrees.
res_each_deg_on_grid	num indicates the resolution of points of one degree along the grid lines. Default is 0.01, e.g. 100 points per degree.

Value

list with disc.image, hdrlst and header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_feature_extraction

Extracts the plage, enhanced network and active network features

---

Description

Extracts the plage, enhanced network and active network features by applying cutoff thresholds. Calculates also the hemispheric corrected values for the extracted feature areas and contrasts.

Usage

```
mod_feature_extraction(x, hdrlst, header, plage.contrast = 1.35,
  en.contrast = 1.25, qn.contrast = 1.1)
```

Arguments

x	tibble containing 3 columns wih pixel coordinates i and j and flat image pixel values.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.
plage.contrast	cutoff threshold for plages.
en.contrast	cutoff threshold for enhanced network.
qt.contrast	cutoff threshold for quiet network.

Value

tibble with mask of the extracted features and their contrast values.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_fits_import *Imports a single FITS file*

Description

Imports single FITS file given the file path and name. Returns tibble with image, tibble with hdrlst and character vector header.

Usage

```
mod_fits_import(inp_data_path, inp_file_name, sdo.image = "FALSE",
  cut.image = "FALSE")
```

Arguments

inp_data_path	string with input data path.
inp_file_name	string with input file name.
sdo.image	if TRUE the image is originally an imported sdo jpg-file.
cut.image	if TRUE the image contains additional text in the corners.

Value

list with fitsim, hdrlst and header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_index_calculation *Calculates activity indices and provides data for images and charts*

Description

Calculates activity indices and provides input data for FITS images and charts.

Usage

```
mod_index_calculation(x, hdrlst, header)
```

Arguments

x	tibble containing the previously calculated images.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.

Value

list with images, hdrlst, header, indices, charts and syopsis.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

mod_load_param *Prepares the parameter list for STU and SDO image analysis*

Description

Prepares the parameter list for STU and SDO image analysis. The add.border.pix and contrast thresholds are set following the recommendation of Jannine Meier from 13.01.2020.

Usage

```
mod_load_param(inp_file_name, sdo.image, inp_data_path, out_data_path,
  rds.output = "FALSE", full.output = "FALSE", light.save = "FALSE",
  fits.save = "FALSE", jpg.save = "FALSE")
```

Arguments

inp_file_name	input file name with extension.
sdo.image	if TRUE, param.lst for SDO image analysis is chosen.
inp_data_path	full path to input directory.
out_data_path	full path to output directory.
rds.output	if TRUE all results are saved as R data file.
full.output	if TRUE the full image data table is saved as csv file.
light.save	if TRUE only a small selection of csv files are saved.
fits.output	if TRUE some fits images are saved.
jpg.output	if TRUE some daily charts are saved as jpg files.

Value

list containing param.lst.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

mod_output *Saves results to disk*

Description

Saves results to disk.

Usage

```
mod_output(images, hdrlst, header, total.indices, hemisphere.indices,
  latitude.indices, chart.indices, charts, synopsis.indices, synopsis,
  inp_file_name, rds.output = "FALSE", full.output = "FALSE",
  light.save = "FALSE", fits.save = "FALSE", jpg.save = "FALSE",
  out_data_path)
```

Arguments

images	tibble containing the previously calculated images.
hdrlst	list containing image FITS header keywords and values.
header	list containing image FITS header.
rds.output	if TRUE all results are saved as R data file.
full.output	if TRUE the full image data table is saved as csv file.
light.save	if TRUE only a small selection of csv files are saved.
out_data_path	full path to output directory.
fits.output	if TRUE some fits images are saved.
jpg.output	if TRUE some daily charts are saved as jpg files.

Value

list containing images, hdrlst, header, indices and charts.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

---

newKwv *Utilities for modifying FITS header entries*

Description

Utilities for modifying FITS header entries.

Usage

```
newKwv(keyw, val, note = "")
```

Value

string vector with modified header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

`wrap_mod_calcium` *Runs the sunviewr modules for the calcium image analysis*

Description

Runs the sunviewr modules for the exploration of calcium images and the determination of activity indices for one image.

Usage

```
wrap_mod_calcium(inp_file_name, sdo.image, inp_data_path, out_data_path,
  rds.output = "FALSE", full.output = "FALSE", light.save = "FALSE",
  fits.save = "FALSE", jpg.save = "FALSE")
```

Arguments

`inp_file_name`	input file name with extension.
`sdo.image`	if TRUE, param.lst for SDO image analysis is chosen.
`inp_data_path`	full path to input directory.
`out_data_path`	full path to output directory.
`rds.output`	if TRUE all results are saved as R data file.
`full.output`	if TRUE the full image data table is saved as csv file.
`light.save`	if TRUE only a small selection of csv files are saved.
`fits.output`	if TRUE some fits images are saved.
`jpg.output`	if TRUE some daily charts are saved as jpg files.

Value

list with resulting image and header.

Author(s)

[Thomas K. Friedli](mailto:thomas.friedli@bluewin.ch)

Index