Package 'x3ptools'

March 27, 2019

Type Package

Title Tools for Working with 3D Surface Measurements

Version 0.0.2

Date 2019-02-28

Maintainer Heike Hofmann <hofmann@iastate.edu>

Description The x3p file format is specified in ISO standard 5436:2000 to describe 3d surface measurements. 'x3ptools' allows reading, writing and basic modifications to the 3D surface measurements.

Depends R (>= 3.3)

Imports digest (>= 0.6.15), xml2 (>= 1.2.0), rgl (>= 0.99.9), zoo (>= 1.8.1), png (>= 0.1-7), assertthat, grDevices

Suggests knitr, rmarkdown, testthat, covr, here, dplyr, magick (>= 2.0)

License MIT + file LICENSE

LazyData true

RoxygenNote 6.1.1

URL https://github.com/heike/x3ptools

BugReports https://github.com/heike/x3ptools/issues

Encoding UTF-8

NeedsCompilation no

Author Heike Hofmann [aut, cre],
Susan Vanderplas [aut],
Ganesh Krishnan [aut],
Eric Hare [aut]

Repository CRAN

Date/Publication 2019-03-27 16:50:06 UTC

R topics documented:

`addtemplate_x3p` *Add/change xml meta information in x3p object*

Description

Use the specified template to overwrite the general info in the x3p object (and structure of the feature info, if needed).

Usage

```
addtemplate_x3p(x3p, template = NULL)
```

Arguments

x3p	x3p object
template	file path to xml template, use NULL for in-built package template

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
# exchange meta information for general x3p information:
logo <- addtemplate_x3p(logo, template = system.file("templateXML.xml", package="x3ptools"))
logo$general.info
```

---

calculate_spacing *Calculate grid spacing*

Description

Helper function, not exported.

Usage

```
calculate_spacing(x3p, spaces, axis = "y")
```

Arguments

x3p	x3p file
spaces	space between grid lines
axis	axis to calculate, as character

Value

vector of line locations

---

df_to_x3p *Convert a data frame into an x3p file*

Description

Convert a data frame into an x3p file

Usage

```
df_to_x3p(dframe)
```

Arguments

dframe	data frame. dframe must have the columns x, y, and value.

Value

x3p object

`head.x3p` *Show meta information of an x3p file*

Description

`head.x3p` expands the generic head method for x3p objects. It gives a summary of the most relevant 3p meta information and returns the object invisibly.

Usage

```
## S3 method for class 'x3p'
head(x, n = 6L, ...)
```

Arguments

`x`	x3p object
`n`	number of rows/columns of the matrix
`...`	extra parameters passed to head.matrix()

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
head(logo)
```

`image_x3p` *Plot x3p object as an image*

Description

Plot x3p object as an image

Usage

```
image_x3p(x3p, file = NULL, col = "#cd7f32", crosscut = NA,
  ccParam = list(color = "#e6bf98", radius = 5), size = c(750, 250),
  zoom = 0.35, multiply = 5, ...)
```

Arguments

x3p	x3p object
file	file name for saving, if file is NULL the opengl device stays open. The file extension determines the type of output. Possible extensions are png, stl (suitable for 3d printing), or svg.
col	color specification
crosscut	crosscut index
ccParam	list with named components, consisting of parameters for showing crosscuts: color and radius for crosscut region
size	vector of width and height
zoom	numeric value indicating the amount of zoom
multiply	exaggerate the relief by factor multiply
...	not used

Examples

```
## Not run:
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
image_x3p(logo, file = "logo.png", crosscut = 50*.645e-6)
# alternative to crosscut
logoplus <- x3p_add_hline(logo, yintercept = 50*.645e-6, color = "#e6bf98", size = 5)
image_x3p(logoplus, size = c(741, 419), zoom=0.5)

## End(Not run)
```

---

interpolate_x3p *Interpolate from an x3p object*

Description

An interpolated scan is created at specified resolutions resx, resy in x and y direction. The interpolation is based on na.approx from the zoo package. It is possible to create interpolations at a higher resolution than the one specified in the data itself, but it is not recommended to do so. interpolate_x3p can also be used as a way to linearly interpolate any missing values in an existing scan without changing the resolution.

Usage

```
interpolate_x3p(x3p, resx = 1e-06, resy = resx, maxgap = 1)
```

Arguments

x3p	x3p object
resx	numeric value specifying the new resolution for the x axis.
resy	numeric value specifying the new resolution for the y axis.
maxgap	integer variable used in na.approx to specify the maximum number of NAs to be interpolated, defaults to 1.

Value

interpolated x3p object

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
# resolution:
logo$header.info$incrementX
# change resolution to 1 micron = 1e-6 meters
logo2 <- interpolate_x3p(logo, resx = 1e-6)
logo2$header.info$incrementX
```

---

print.x3p *Show meta information of an x3p file*

Description

print.x3p expands the generic print method for x3p objects. It gives a summary of the most relevant 3p meta information and returns the object invisibly.

Usage

```
## S3 method for class 'x3p'
print(x, ...)
```

Arguments

x	x3p object
...	ignored

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
print(logo)
```

---

read_x3p *Read an x3p file into an x3p object*

Description

Read file in x3p format. x3p formats describe 3d topological surface according to ISO standard ISO5436 – 2000. x3p files are a container format implemented as a zip archive of a folder consisting of an xml file of meta information and a binary matrix of numeric surface measurements.

Usage

```
read_x3p(file, size = NA, quiet = T)
```

Arguments

file	The file path to the x3p file, or an url to an x3p file
size	size in bytes to use for reading the binary file. If not specified, default is used. Will be overwritten if specified in the xml meta file.
quiet	for url downloads, show download progress?

Value

x3p object consisting of a list of the surface matrix and the four records as specified in the ISO standard

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
```

---

rotate_x3p *Rotate an x3p object*

Description

Rotate the surface matrix of an x3p object. Also adjust meta information.

Usage

```
rotate_x3p(x3p, angle = 90)
```

Arguments

x3p	x3p object
angle	rotate counter-clockwise by angle degrees given as 90, 180, 270 degree (or -90, -180, -270).

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
dim(logo$surface.matrix)
## Not run:
image_x3p(logo)

## End(Not run)
# rotate the image by 90 degrees clock-wise:
logo90 <- rotate_x3p(logo, 90)
dim(logo90$surface.matrix)
```

```
## Not run:
image_x3p(logo90)

## End(Not run)
```

`sample_x3p` *Sample from an x3p object*

Description

Sample from an x3p object

Usage

```
sample_x3p(x3p, m = 2, mY = m, offset = 0, offsetY = offset)
```

Arguments

`x3p`	x3p object
`m`	integer value - every mth value is included in the sample
`mY`	integer value - every mth value is included in the sample in x direction and every mYth value is included in y direction
`offset`	integer value between 0 and m-1 to specify offset of the sample
`offsetY`	integer value between 0 and mY-1 to specify different offsets for x and y direction

Value

down-sampled x3p object

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
dim(logo$surface.matrix)
# down-sample to one-fourth of the image:
logo4 <- sample_x3p(logo, m=4)
dim(logo4$surface.matrix)
## Not run:
image_x3p(logo)
image_x3p(logo4)

## End(Not run)
```

transpose_x3p *Transpose an x3p object*

Description

Transpose the surface matrix of an x3p object. Also adjust meta information.

Usage

```
transpose_x3p(x3p)
```

Arguments

x3p	x3p object

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
dim(logo$surface.matrix)
## Not run:
image_x3p(logo)

## End(Not run)
#  transpose the image
logotp <- transpose_x3p(logo)
dim(logotp$surface.matrix)
## Not run:
image_x3p(logotp)

## End(Not run)
```

write_x3p *Write an x3p object to a file*

Description

Write an x3p object to a file

Usage

```
write_x3p(x3p, file, size = 8, quiet = F)
```

Arguments

x3p	x3p object
file	path to where the file should be written
size	integer. The number of bytes per element in the surface matrix used for creating the binary file. Use size = 4 for 32 bit IEEE 754 floating point numbers and size = 8 for 64 bit IEEE 754 floating point number (default).
quiet	suppress messages

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
# write a copy of the file into a temporary file
write_x3p(logo, file = tempfile(fileext="x3p"))
```

---

x3p_add_grid *Add a grid of helper lines to the mask of an x3p object*

Description

Add a grid of lines to overlay the surface of an x3p object. Lines are added to a mask. In case no mask exists, one is created.

Usage

```
x3p_add_grid(x3p, spaces, size = c(1, 3, 5), color = c("grey50",
  "black", "darkred"))
```

Arguments

x3p	x3p object
spaces	space between grid lines, doubled for x
size	width (in pixels) of the lines
color	(vector of) character values to describe color of lines

Value

x3p object with added vertical lines in the mask

Examples

```
## Not run:
logo <- read_x3p(system.file("csafe-logo.x3p", package = "x3ptools"))
# ten vertical lines across:
logoplus <- x3p_add_grid(logo,
  spaces = 50e-6, size = c(1, 3, 5),
  color = c("grey50", "black", "darkred")
)
image_x3p(logoplus, size = c(741, 419), zoom = 0.5)

## End(Not run)
```

`x3p_add_hline` *Add horizontal line to the mask of an x3p object*

Description

Add horizontal lines to overlay the surface of an x3p object. Lines are added to a mask. In case no mask exists, one is created.

Usage

```
x3p_add_hline(x3p, yintercept, size = 5, color = "#e6bf98")
```

Arguments

`x3p`	x3p object
`yintercept`	(vector of) numerical values for the position of the lines.
`size`	width (in pixels) of the line
`color`	(vector of) character values to describe color of lines

Value

x3p object with added vertical lines in the mask

Examples

```
## Not run:
logo <- read_x3p(system.file("csafe-logo.x3p", package = "x3ptools"))
color_logo <- magick::image_read(system.file("csafe-color.png", package = "x3ptools"))
logoplus <- x3p_add_mask(logo, as.raster(color_logo))
# five horizontal lines at equal intervals:
logoplus <- x3p_add_hline(logo, seq(0, 418 * 6.4500e-7, length = 5), size = 3)
image_x3p(logoplus, size = c(741, 419), zoom = 0.5)

## End(Not run)
```

x3p_add_legend *Add legend to active rgl object*

Description

Add the legend for colors and annotations to the actuve rgl window.

Usage

```
x3p_add_legend(x3p, colors = NULL)
```

Arguments

x3p	x3p object with a mask
colors	named character vector of colors (in hex format by default), names contain annotations

Examples

```
x3p <- read_x3p(system.file("sample-land.x3p", package="x3ptools"))
## Not run:
image_x3p(x3p) # run when rgl can open window on the device
x3p_add_legend(x3p) # add legend

## End(Not run)
```

x3p_add_mask *Add/Exchange a mask for an x3p object*

Description

Create a mask for an x3p object in case it does not have a mask yet. Masks are used for overlaying colors on the bullets surface.

Usage

```
x3p_add_mask(x3p, mask = NULL)
```

Arguments

x3p	x3p object
mask	raster matrix of colors with the same dimensions as the x3p surface. If NULL, an object of the right size will be created.

Value

x3p object with added/changed mask

Examples

```
x3p <- read_x3p(system.file("sample-land.x3p", package="x3ptools"))
# x3p file has mask consisting color raster image:
x3p$mask[1:5,1:5]
## Not run:
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
color_logo <- png::readPNG(system.file("csafe-color.png", package="x3ptools"))
logoplus <- x3p_add_mask(logo, as.raster(color_logo))
image_x3p(logoplus, multiply=50, size = c(741, 419),zoom = 0.5)

## End(Not run)
```

---

x3p_add_vline *Add vertical line to the mask of an x3p object*

Description

Add vertical lines to overlay the surface of an x3p object. Lines are added to a mask. In case no mask exists, one is created.

Usage

```
x3p_add_vline(x3p, xintercept, size = 5, color = "#e6bf98")
```

Arguments

x3p	x3p object
xintercept	(vector of) numerical values for the position of the lines.
size	width (in pixels) of the line
color	(vector of) character values to describe color of lines

Value

x3p object with added vertical lines in the mask

Examples

```
## Not run:
logo <- read_x3p(system.file("csafe-logo.x3p", package = "x3ptools"))
logo_color <- magick::image_read(system.file("csafe-color.png", package = "x3ptools"))
logoplus <- x3p_add_mask(logo, as.raster(logo_color))
# ten vertical lines across:
logoplus <- x3p_add_vline(logo, seq(0, 740 * 6.4500e-7, length = 5), size = 3)
image_x3p(logoplus, size = c(741, 419), zoom = 0.5)
```

```
## End(Not run)
```

---

x3p_darker *Darken active rgl object*

Description

Makes the currently active rgl object darker by removing a light source. Once all light sources are removed the object can not be any darker.

Usage

```
x3p_darker()
```

Examples

```
x3p <- read_x3p(system.file("sample-land.x3p", package="x3ptools"))
## Not run:
image_x3p(x3p) # run when rgl can open window on the device
x3p_darker() # remove a light source

## End(Not run)
```

---

x3p_delete_mask *Delete mask from an x3p object*

Description

Deletes mask and its annotations from an x3p file.

Usage

```
x3p_delete_mask(x3p)
```

Arguments

x3p	x3p object

Value

x3p object without the mask

x3p_get_scale *Check resolution of a scan*

Description

Scans in x3p format capture 3d topogographic surfaces. According to ISO standard ISO5436 – 2000 scans are supposed to be captured in meters. For microscopic images capture in meters might be impractical.

Usage

```
x3p_get_scale(x3p)
```

Arguments

x3p	object

Value

numeric value of resolution per pixel

x3p_lighter *Lighten active rgl object*

Description

Make the currently active rgl object lighter. Adds a light source. Up to eight light sources can be added. Alternatively, any rgl light source can be added (see `light3d`).

Usage

```
x3p_lighter()
```

Examples

```
x3p <- read_x3p(system.file("sample-land.x3p", package="x3ptools"))
## Not run:
image_x3p(x3p) # run when rgl can open window on the device
x3p_lighter() # add a light source

## End(Not run)
```

x3p_mask_legend *Get legend for mask colors*

Description

Retrieve color definitions and annotations from the mask. If available, results in a named vector of colors.

Usage

```
x3p_mask_legend(x3p)
```

Arguments

x3p	x3p object with a mask

Value

named vector of colors, names show annotations. In case no annotations exist NULL is returned.

Examples

```
x3p <- read_x3p(system.file("sample-land.x3p", package="x3ptools"))
x3p_mask_legend(x3p) # annotations and color hex definitions
```

x3p_modify_xml *Modify xml elements meta information in x3p object*

Description

Identify xml fields in the meta file of an x3p object by name and modify content if uniquely described.

Usage

```
x3p_modify_xml(x3p, element, value)
```

Arguments

x3p	x3p object
element	character or integer. In case of character, name of xml field in the meta file. Note that element can contain regular expressions, e.g. "*" returns all meta fields. In case of integer, element is used as an index for the meta fields.
value	character. Value to be given to the xml field in the meta file.

Value

x3p object with changed meta information

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
x3p_show_xml(logo, "creator")
x3p_modify_xml(logo, "creator", "I did that")
x3p_show_xml(logo, 20)
x3p_modify_xml(logo, 20, "I did that, too")
```

---

x3p_m_to_mum *Convert x3p header information to microns from meters*

Description

ISO standard 5436_2 asks for specification of values in meters. For topographic surfaces collected by microscopes values in microns are more readable. Besides scaling the values in the surface matrix, corresponding increments are changed to microns as well.

Usage

```
x3p_m_to_mum(x3p)
```

Arguments

x3p	x3p file with header information in meters

Value

x3p with header information in microns

---

x3p_scale_unit *Scale x3p object by given unit*

Description

x3p objects can be presented in different units. ISO standard 5436_2 asks for specification of values in meters. For topographic surfaces collected by microscopes values in microns are more readable. This functions allows to convert between different units.

Usage

```
x3p_scale_unit(x3p, scale_by)
```

Arguments

x3p	object in x3p format, 3d topographic surface.
scale_by	numeric value. Value the surface to be scaled by. While not enforced, values of scale_by make most sense as multiples of 10 (for a metric system).

Value

x3p with header information in microns

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
logo # measurements in meters
x3p_scale_unit(logo, scale_by=10^6) # measurements in microns
```

---

x3p_show_xml *Show xml elements from meta information in x3p object*

Description

Identify xml fields by name and show content.

Usage

```
x3p_show_xml(x3p, element)
```

Arguments

x3p	x3p object
element	character or integer (vector). In case of character, name of xml field in the meta file. Note that element can contain regular expressions, e.g. "*" returns all meta fields. In case of integer, element is used as an index vector for the meta fields.

Value

list of exact field names and their contents

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
x3p_show_xml(logo, "creator") # all fields containing the word "creator"
x3p_show_xml(logo, "axis")
x3p_show_xml(logo, "CZ.AxisType")
# show all fields:
x3p_show_xml(logo, "*")
# show first five fields
x3p_show_xml(logo, 1:5)
```

x3p_to_df *Convert an x3p file into a data frame*

Description

An x3p file consists of a list with meta info and a 2d matrix with scan depths. fortify turns the matrix into a data frame, using the parameters of the header as necessary.

Usage

```
x3p_to_df(x3p)
```

Arguments

x3p	a file in x3p format as returned by function read_x3p

Value

data frame with variables x, y, and value and meta function in attribute

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
logo_df <- x3p_to_df(logo)
head(logo_df)
```

y_flip_x3p *Flip the y coordinate of an x3p image*

Description

One of the major changes between the previous two ISO standards is the way the y axis is defined in a scan. The entry (0,0) used to refer to the top left corner of a scan, now it refers to the bottom right corner, which means that all legacy x3p files have to flip their y axis in order to conform to the newest ISO norm.

Usage

```
y_flip_x3p(x3p)
```

Arguments

x3p	x3p object

Value

x3p object in which the y coordinate is reversed.

Examples

```
logo <- read_x3p(system.file("csafe-logo.x3p", package="x3ptools"))
dim(logo$surface.matrix)
## Not run:
image_x3p(logo)

## End(Not run)
# flip the y-axis for the old ISO standard:
logoflip <- y_flip_x3p(logo)
dim(logoflip$surface.matrix)
## Not run:
image_x3p(logoflip)

## End(Not run)
```

Index