

Package ‘x3ptools’

April 5, 2018

Type Package

Title Tools for Working with 3D Surface Measurements

Version 0.0.1

Date 2018-03-21

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), digest (>= 0.6.15), xml2 (>= 1.2.0), rgl (>= 0.99.9), zoo (>= 1.8.1)

Suggests knitr, rmarkdown

License MIT + file LICENSE

LazyData true

RoxygenNote 6.0.1

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

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

NeedsCompilation no

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

Repository CRAN

Date/Publication 2018-04-05 11:22:21 UTC

R topics documented:

| | |
|---------------------------|---|
| addtemplate_x3p | 2 |
| df_to_x3p | 2 |
| image_x3p | 3 |
| interpolate_x3p | 4 |
| read_x3p | 4 |

| | |
|-------------------------|---|
| rotate_x3p | 5 |
| sample_x3p | 6 |
| transpose_x3p | 6 |
| write_x3p | 7 |
| x3p_to_df | 8 |
| y_flip_x3p | 8 |

Index 10

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)
```

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
```

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

| | |
|-----------|------------------------------------|
| image_x3p | <i>Plot x3p object as an image</i> |
|-----------|------------------------------------|

Description

Plot x3p object as an image

Usage

```
image_x3p(x3p, file = NULL, col = "#cd7f32", 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). |
| col | color specification |
| 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")

## 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 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

| | |
|---------------------|---|
| <code>x3p</code> | x3p object |
| <code>resx</code> | numeric value specifying the new resolution for the x axis. |
| <code>resy</code> | numeric value specifying the new resolution for the y axis. |
| <code>maxgap</code> | integer variable used in <code>na.approx</code> 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
```

read_x3p *Read an x3p file into an x3p object*

Description

Read an x3p file into an x3p object

Usage

```
read_x3p(file)
```

Arguments

file The file path to the x3p file

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 | <i>Rotate an x3p object</i> |
|------------|-----------------------------|

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 | <i>Sample from an x3p object</i> |
|------------|----------------------------------|

Description

Sample from an x3p object

Usage

```
sample_x3p(x3p, m = 2)
```

Arguments

| | |
|-----|---|
| x3p | x3p object |
| m | integer value - every mth value is included in the sample |

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 | <i>Transpose an x3p object</i> |
|---------------|--------------------------------|

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)
```

Arguments

| | |
|------|--|
| x3p | x3p object |
| file | path to where the file should be written |

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_to_df | <i>Convert an x3p file into a data frame</i> |
|-----------|--|

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 | <i>Flip the y coordinate of an x3p image</i> |
|------------|--|

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

`addtemplate_x3p`, 2

`df_to_x3p`, 2

`image_x3p`, 3

`interpolate_x3p`, 4

`read_x3p`, 4

`rotate_x3p`, 5

`sample_x3p`, 6

`transpose_x3p`, 6

`write_x3p`, 7

`x3p_to_df`, 8

`y_flip_x3p`, 8